

Сініцин Ігор Петрович*Інститут програмних систем НАН України, Київ*

ORCID 0000-0002-4120-0784

Дорошенко Анатолій Юхимович*Інститут програмних систем НАН України, Київ**Київський політехнічний інститут імені Ігоря Сікорського, Київ*

ORCID 0000-0002-8435-1451

Смірнов Валентин Євгенович*Інститут програмних систем НАН України, Київ*

ORCID 0009-0006-4022-5951

Яценко Олена Анатоліївна*Інститут програмних систем НАН України, Київ*

ORCID 0000-0002-4700-6704

ДОСЛІДЖЕННЯ ПРОДУКТИВНОСТІ НЕЙРОМЕРЕЖ YOLO ДЛЯ ВБУДОВАНИХ СИСТЕМ І ДРОНІВ

Анотація. У статті представлено комплексне дослідження продуктивності сучасних версій нейронних мереж YOLO (You Only Look Once) на широкому спектрі апаратних платформ — від настільних GPU до мобільних ARM-пристроїв і спеціалізованих AI-прискорювачів. У межах роботи оцінено ефективність моделей YOLOv4, YOLOv5, YOLOv8, а також новітніх YOLOv10n і YOLOv11n з урахуванням архітектурної складності, використання сучасних оптимізацій (FP16, INT8, TensorRT), затримки при інференсі, енергоспоживання та продуктивності в FPS. Для забезпечення об'єктивного міжплатформного порівняння запропоновано уніфіковану систему одиниць продуктивності, яка базується на еталонному виконанні YOLOv4 на GPU NVIDIA GTX 1070. Проведено понад 1000 запусків на реальному обладнанні, включаючи Orange Pi 5/5 Plus, Raspberry Pi 5 (з Hailo-8L AI HAT), серії Rockchip RK35xx, платформи Google Coral та лінійку NVIDIA Jetson Orin (Nano, NX, AGX). Всі вимірювання здійснено з урахуванням статистичної достовірності (50–100 прогонів), із забезпеченням довірчого інтервалу 90–95 %. Додатково враховано режим інференсу, тип квантизації та характер апаратної інтеграції. Результати дозволили виявити оптимальні конфігурації для різних умов застосування, зокрема в системах реального часу. Особливу увагу приділено сценаріям роботи YOLO на борту дронів, де ключовими є обмежені енергетичні ресурси, вага, компактність рішень, автономність і низька латентність. Розглянуто типові архітектури вбудованих рішень, а також оцінено ефективність розгортання моделей з урахуванням обмежень кожної платформи. Стаття містить узагальнені рекомендації щодо вибору моделей і апаратних рішень для автономної навігації, моніторингу, спостереження та робототехнічних систем в умовах обмеженого бюджету потужності. Запропоновані висновки можуть бути корисними розробникам і дослідникам у сферах вбудованих систем, Edge AI, IoT і автономних безпілотних платформ.

Ключові слова: YOLO, глибоке навчання, вбудовані системи, дрони, інференс, енергоефективність, GPU, ARM, Google Coral, NVIDIA Jetson Orin.

Sinitsyn Igor*Institute of Software Systems of the National Academy of Sciences of Ukraine, Kyiv*

ORCID 0000-0002-4120-0784

Doroshenko Anatoliy*Institute of Software Systems of the National Academy of Sciences of Ukraine, Kyiv**Igor Sikorsky Kyiv Polytechnic Institute, Kyiv*

ORCID 0000-0002-8435-1451

Smirnov Valentyn*Institute of Software Systems of the National Academy of Sciences of Ukraine, Kyiv*

ORCID 0009-0006-4022-5951

Yatsenko Olena

© Сініцин І.П., Дорошенко А.Ю., Смірнов В.Є., Яценко О.А.

2025

A STUDY OF YOLO NEURAL NETWORK PERFORMANCE FOR EMBEDDED SYSTEMS AND UAVS

Abstract. The article presents a comprehensive study of the performance of modern YOLO (You Only Look Once) neural network versions across a wide range of hardware platforms — from desktop GPUs to mobile ARM devices and specialized AI accelerators. The study evaluates the efficiency of YOLOv4, YOLOv5, YOLOv8, as well as the latest YOLOv10n and YOLOv11n models, taking into account architectural complexity, optimization techniques (FP16, INT8, TensorRT), inference latency, power consumption, and frames per second (FPS). To ensure objective cross-platform comparison, a unified performance unit system was proposed, based on the reference execution of YOLOv4 on an NVIDIA GTX 1070 GPU. More than 1000 runs were conducted on real hardware, including Orange Pi 5/5 Plus, Raspberry Pi 5 (with Hailo-8L AI HAT), Rockchip RK35xx series, Google Coral platforms, and the NVIDIA Jetson Orin lineup (Nano, NX, AGX). All measurements were performed with statistical significance (50–100 iterations) and a confidence interval of 90–95 %. Additionally, inference modes, quantization types, and hardware integration aspects were considered. The results identified optimal configurations for various real-time application scenarios. Special attention was given to YOLO deployment onboard drones, where limited energy resources, device weight, compactness, autonomy, and low latency are critical. Typical embedded architectures were analyzed, and the deployment efficiency of models under platform-specific constraints was evaluated. The article provides generalized recommendations for selecting models and hardware solutions for autonomous navigation, monitoring, surveillance, and robotics systems operating under strict power budgets. The findings can be valuable to developers and researchers in the fields of embedded systems, Edge AI, IoT, and autonomous unmanned platforms.

Keywords: YOLO, deep learning, embedded systems, drones, inference, energy efficiency, GPU, ARM, Google Coral, NVIDIA Jetson Orin.

1. Постановка проблеми

У сучасних умовах стрімкого розвитку систем комп'ютерного зору значну увагу привертають нейронні мережі типу YOLO (You Only Look Once) [1], які демонструють високу швидкість та ефективність у задачах виявлення об'єктів у реальному часі. Завдяки поєднанню точності та швидкодії, моделі YOLO знаходять широке застосування в автономних системах, зокрема у безпілотних літальних апаратах (дронах), вбудованих IoT-пристроях, мобільних застосунках та роботизованих платформах [2–4].

Разом із тим, інтеграція таких моделей у ресурсообмежені пристрої супроводжується низкою викликів. До них належать жорсткі вимоги до обчислювальних потужностей, обмеження енергоспоживання, необхідність забезпечення низької затримки інференсу та компактності моделей. В умовах обмежених ресурсів навіть незначне перевищення енергетичного або апаратного бюджету може призвести до непридатності рішення для практичного використання. Це особливо актуально для бортових систем дронів, де кожен грам ваги та кожен ват енергії критично впливають на тривалість польоту, стабільність роботи та безпеку виконання місій.

Попри значну кількість досліджень, у науковій та технічній літературі недостатньо систематизованих результатів комплексного порівняння різних версій YOLO на широкому спектрі апаратних платформ — від настільних GPU до спеціалізованих AI-прискорювачів і енергоефективних одноплатних комп'ютерів. Відсутність уніфікованих методик оцінки продуктивності ускладнює об'єктивний вибір архітектури для конкретного застосування, що може призводити до надмірного використання ресурсів або зниження якості розпізнавання.

Таким чином, актуальною науково-прикладною задачею є розробка підходів до комплексної оцінки продуктивності моделей YOLO з урахуванням архітектурної складності, енергоспоживання, затримки інференсу та масштабованості рішень. Теоретична значущість такої роботи полягає у формуванні єдиної системи метрик для міжплатформного порівняння моделей комп'ютерного зору, а практична — у визначенні оптимальних комбінацій “модель–апаратна платформа” для вбудованих систем і дронів. Результати подібних досліджень можуть слугувати основою для побудови ефективних автономних систем навігації, моніторингу та виявлення об'єктів у реальному часі, що особливо важливо для промислових, рятувальних та оборонних застосувань.

2. Аналіз останніх досліджень і публікацій

У контексті даного дослідження варто відзначити наявність низки праць, які мають подібну спрямованість. Зокрема, в роботі [5] представлено масштабний бенчмарк YOLOBench, який охоплює понад 550 конфігурацій моделей YOLO (від YOLOv3 до YOLOv8) на різноманітних вбудованих

платформах, включаючи ARM CPU, x86 CPU, GPU та NPU. Автори зібрали дані щодо точності та затримки інференсу для низки одноетапних детекторів на основі YOLO різного масштабу, забезпечивши фіксоване навчальне середовище (єдиний код та гіперпараметри тренування). Аналіз Парето-оптимальності показав, що при застосуванні сучасних архітектур і технік навчання навіть старі моделі (YOLOv3, YOLOv4) можуть демонструвати збалансоване співвідношення точності та затримки. Крім того, досліджувалися методи оцінки точності без повторного навчання (zero-cost proxies), що використовуються в задачах NAS. Автори показали, що хоча більшість таких методів поступаються простим базовим метрикам (наприклад, кількості MAC-операцій), окремі з них можуть бути корисними для передбачення Парето-оптимальних моделей.

Інше релевантне дослідження [6] фокусується на порівняльній оцінці глибоких моделей об'єктного розпізнавання на різних вбудованих пристроях. Автори провели тестування з п'ятьма рівнями квантування, два з яких дозволяють виконувати моделі на універсальних процесорах, а три — на спеціалізованих AI-співпроцесорах. Оцінювалися швидкість, точність (mAP), енергоспоживання та затримка, а також проведено аналіз переваг і недоліків вибраних платформ (i-MX8M-PLUS та Coral Dev Board). Отримані результати підкреслюють необхідність пошуку балансу між точністю та придатністю до розгортання в умовах обмежених ресурсів, а також демонструють виклики конвертації моделей у формати, сумісні з TensorFlow Lite.

Водночас існуючі роботи мають низку обмежень. Частина досліджень охоплює лише окремі версії YOLO або обмежується специфічним класом апаратних рішень, не надаючи цілісної картини міжплатформного порівняння. У більшості випадків недостатньо уваги приділено новітнім архітектурам (YOLOv10n, YOLOv11n), комплексній оцінці продуктивності з урахуванням енергоспоживання, латентності та можливостей оптимізації (FP16, INT8, TensorRT), а також формуванню уніфікованої системи продуктивності для різних платформ.

Таким чином, залишається невирішеною задача побудови узагальненої методики оцінювання продуктивності сучасних моделей YOLO на широкому спектрі апаратних платформ — від настільних GPU до low-power ARM/NPU рішень і спеціалізованих AI-прискорювачів. Її вирішення дозволить забезпечити об'єктивний вибір комбінацій «модель–апаратна платформа» для конкретних сценаріїв використання, зокрема у дронах та інших автономних системах.

3. Мета і задачі дослідження

Метою даного дослідження є комплексна оцінка продуктивності різних версій нейронних мереж YOLO на сучасних апаратних платформах із метою визначення оптимальних рішень для використання у вбудованих системах і безпілотних літальних апаратах, з урахуванням обмежень щодо енергоспоживання, обчислювальних ресурсів та латентності.

Для досягнення поставленої мети необхідно вирішити такі задачі:

- 1) проаналізувати архітектурні особливості та обчислювальні вимоги актуальних версій YOLO (YOLOv4–YOLOv11n), включаючи підтримку сучасних оптимізацій (FP16, INT8, TensorRT);
- 2) розробити уніфіковану методику оцінки продуктивності, що дозволяє порівнювати ефективність моделей на різних апаратних платформах, незалежно від типу процесора чи прискорювача;
- 3) провести експериментальні вимірювання FPS, латентності, енергоспоживання та завантаження апаратних ресурсів для різних моделей YOLO на широкому спектрі пристроїв — від настільних GPU до low-power ARM/NPU систем та спеціалізованих AI-прискорювачів;
- 4) виявити вузькі місця у продуктивності моделей при роботі на ресурсообмежених платформах та оцінити ефективність застосування оптимізаційних технік;
- 5) сформулювати рекомендації щодо вибору моделей YOLO та апаратних рішень для сценаріїв автономної навігації, моніторингу та розпізнавання об'єктів на борту дронів;
- 6) оцінити перспективність подальшого застосування запропонованої методики до інших архітектур комп'ютерного зору (MobileNet, EfficientDet тощо) та до адаптивних методів квантизації й енергоменеджменту.

4. Результати дослідження

У межах дослідження проаналізовано продуктивність актуальних версій YOLO (від YOLOv4 до YOLOv11n) [7–11], що відрізняються архітектурною складністю, вимогами до апаратного забезпечення та підтримкою оптимізацій (FP16, INT8, TensorRT [12] тощо). Для порівняння було введено універсальну систему одиниць продуктивності, яка дозволяє співставляти обчислювальну

ефективність моделей незалежно від платформи. Всі вимірювання проводилися з урахуванням статистичної значущості (50–100 прогонів, довірчий інтервал 90–95 %).

Особливу увагу приділено платформам із обмеженим енергоспоживанням, таким як Orange Pi, Raspberry Pi, Rockchip, Google Coral, а також лінійці NVIDIA Jetson Orin. Проведено детальний аналіз показників FPS, латентності, енергоспоживання та апаратної завантаженості, що дозволило сформулювати практичні рекомендації для вибору оптимальних рішень у галузі автономної навігації, моніторингу та розпізнавання об'єктів на борту дронів.

Отримані результати дозволяють не лише порівняти сучасні реалізації YOLO між собою, а й адаптувати вибір моделей та платформ під конкретні вимоги задач у реальному часі.

4.1. Методологія вимірів.

Оцінка продуктивності нейронних мереж YOLO [4] на різних апаратних платформах вимагає узгодженого підходу до проведення експериментів, щоб забезпечити об'єктивність та відтворюваність результатів. У цьому дослідженні використано уніфіковану методологію, що охоплює як стандартизовану систему продуктивності, так і чітко визначені умови проведення вимірювань.

Універсальна система оцінки продуктивності. З метою забезпечення можливості порівняння результатів між різними платформами та архітектурами, була розроблена система універсальних одиниць продуктивності. За базову одиницю прийнято продуктивність моделі YOLOv4 на графічному процесорі NVIDIA GTX 1070 із 8 ГБ відеопам'яті. Такий підхід дозволяє відносно оцінити ефективність інших моделей YOLO незалежно від використовуваного апаратного забезпечення.

Базова одиниця має такі характеристики:

- обчислювальна потужність: $\approx 50\text{--}70$ GFLOPS у режимі інференсу;
- об'єм доступної GPU-пам'яті: $\sim 4\text{--}6$ ГБ;
- еталонна платформа: NVIDIA GTX 1070 (8 ГБ);
- типове енергоспоживання під час інференсу: $\sim 15\text{--}20$ Вт.

Ця абстракція дозволяє порівнювати ефективність виконання нейронної мережі з урахуванням як обчислювального навантаження, так і енергетичних характеристик, що є критично важливим у контексті вбудованих та мобільних систем.

Методологія проведення експериментів. Для досягнення високої точності результатів та можливості порівняння між платформами були встановлені єдині експериментальні параметри. Всі вимірювання проводилися з урахуванням особливостей цільової архітектури та обмежень конкретного пристрою. Основні параметри вимірювання наведено в Табл. 1.

Таблиця 1

Параметри вимірювання для експериментів		
Параметр	Значення	Примітка
Тестова роздільна здатність	640×640 / 416×416	Для ARM-платформ
Розмір батчу	1–4	Обирається з урахуванням обмежень пам'яті
Точність	FP32 / FP16 / INT8	Використовується оптимальна для платформи
Кількість прогонів	50–100	Забезпечення статистичної значущості
Довірчий інтервал	90–95 %	Визначається на основі дисперсії замірів

У всіх експериментах моделі запускалися в умовах, наближених до реального сценарію застосування: інференс проводився на реальних відеокадрах або зображеннях, з урахуванням затримок, що виникають через обмеження пам'яті, частоти кадрів, а також апаратного доступу до NPU, GPU або CPU. Для зменшення впливу фонових процесів усі вимірювання проводились у виділених середовищах або при мінімальному системному навантаженні.

Загальна методика була спрямована на формування репрезентативного набору даних, який дозволяє провести якісний порівняльний аналіз та дати рекомендації для застосування моделей YOLO у вбудованих рішеннях, зокрема в автономних дронах і пристроях з обмеженим енергоспоживанням.

4.2. Продуктивність версій YOLO на настільних GPU.

Аналіз продуктивності нейронних мереж YOLO на потужних настільних графічних процесорах є необхідним етапом для розуміння базового рівня ефективності моделей, на який можна орієнтуватися під час перенесення рішень на менш потужні пристрої. У цьому розділі наведено результати тестування основних версій YOLO на сучасних GPU із достатнім обсягом відеопам'яті та підтримкою обчислень FP16/FP32, зокрема на платформі NVIDIA GTX 1070.

Відносні вимоги до продуктивності. Для оцінки відносної складності та обчислювального навантаження різних версій YOLO було використано показник кількості обчислень у GFLOPS, а також співвіднесено кожен версію з уніфікованими одиницями продуктивності, введеними у підрозділі 4.1. Це дозволяє зіставити ресурсоємність моделей у відносних величинах та оцінити доцільність їх використання в тій чи іншій системі (Табл. 2).

Як видно з таблиці, більш нові та спрощені версії YOLO (наприклад, YOLOv11n [11]) демонструють надзвичайно високу енергоефективність при низьких апаратних вимогах. Натомість старші або великі моделі (YOLOv5x, YOLOv8l) мають значне навантаження на GPU, проте забезпечують максимальну точність виявлення.

Продуктивність на NVIDIA GTX 1070. Відеокарта NVIDIA GTX 1070 із 8 ГБ пам'яті, незважаючи на свій вік, залишається однією з найпопулярніших платформ серед дослідників та розробників, завдяки поєднанню достатньої обчислювальної потужності (≈ 6.5 TFLOPS) та помірного енергоспоживання. В рамках дослідження вона використовувалась як еталонна система для тестування базової продуктивності моделей YOLO (Табл. 3).

Таблиця 2

Аналіз відносної складності та обчислювальних витрат різних версій YOLO

Версія YOLO	Одиниці продуктивності	GFLOPS (середнє $\pm$ відхилення)	Особливості
YOLOv4	1.0	65.2 ± 4.8	Базова версія
YOLOv5n	0.3	18.7 ± 2.1	Nano, мінімальні вимоги
YOLOv5s	0.7	42.3 ± 3.6	Small, оптимальний баланс
YOLOv5m	1.0	68.9 ± 5.2	Medium, висока точність
YOLOv5l	1.3	89.4 ± 6.8	Large, максимальна точність
YOLOv5x	1.8	124.8 ± 9.3	Extra Large, преміум
YOLOv8n	0.2	14.9 ± 1.6	Найбільш ефективна за ресурсами
YOLOv8s	0.5	35.7 ± 2.9	Популярний вибір
YOLOv8m	1.1	78.4 ± 5.8	Професійне застосування
YOLOv8l	2.4	165.2 ± 12.1	Максимальна продуктивність
YOLOv10n	0.18	12.3 ± 1.4	NMS-free архітектура
YOLOv11n	0.17	11.7 ± 1.2	Найефективніша версія за співвідношенням GFLOPS/FPS

Таблиця 3

Результати тестування базової продуктивності моделей YOLO

Версія YOLO	FPS	Розмір батчу	Рекомендація
YOLOv5s	45–60	1–2	Відмінно
YOLOv5m	30–40	1–2	Добре
YOLOv8n	60–80	2–4	Ідеально
YOLOv8s	40–50	2–4	Оптимально
YOLOv11n	65–85	2–4	Кращий вибір

Модель YOLOv11n показала найкращу продуктивність у режимі реального часу (до 85 FPS) при найменшому навантаженні на GPU. Такий результат є наслідком її компактної архітектури без використання NMS (non-maximum suppression), що знижує затримку інференсу. Модель YOLOv8n, своєю чергою, забезпечує стабільний FPS у діапазоні 60–80 при вищій точності, що робить її привабливим вибором для систем із помірними ресурсами.

Таким чином, аналіз продуктивності на настільних GPU дозволяє не лише виявити оптимальні конфігурації для high-end рішень, але й визначити базову лінійку порівняння для менш потужних платформ, таких як ARM, NPU або мобільні AI-прискорювачі. Це стає особливо важливим при проектуванні вбудованих систем, де кожен GFLOP і кожен ват енергії мають значення.

4.3. Продуктивність на мобільних ARM платформах.

Платформи на базі ARM-процесорів дедалі частіше застосовуються в системах комп'ютерного зору завдяки своїй низькій вартості, енергоефективності та компактності. Сучасні дослідження

підтверджують доцільність тестування моделей YOLO на ARM/NPU пристроях з урахуванням компромісу між точністю, FPS та споживанням енергії [5, 6]. У цьому розділі розглянуто продуктивність моделей YOLO на мобільних пристроях, таких як Orange Pi, Raspberry Pi та рішення на основі чипів Rockchip. Для кожної з платформ оцінювались три ключові параметри: продуктивність у GFLOPS, частота кадрів (FPS) та енергоспоживання.

Orange Pi сімейство. Одними з найбільш потужних серед ARM-рішень є платформи Orange Pi 5 та Orange Pi 5 Plus на базі SoC Rockchip RK3588 та RK3588S, які мають вбудований GPU Mali-G610 MP4. Ці пристрої забезпечують помірний рівень обчислювальної потужності при збереженні енергоефективності, що робить їх привабливими для застосування в робототехніці та edge-аналітиці (Табл. 4).

Таблиця 4

Результати вимірювання продуктивності на платформі Orange Pi

Пристрій	YOLO версія	GFLOPS	FPS	Енерго-споживання
Orange Pi 5 Plus (RK3588), Mali-G610 MP4, 8 ГБ LPDDR4X, 1000 МГц	YOLOv5n	3.6 ± 0.4	13–17	8–12 Вт
	YOLOv8n	3.3 ± 0.4	15–19	8–12 Вт
	YOLOv11n	3.1 ± 0.3	18–22	8–12 Вт
Orange Pi 5 (RK3588S), Mali-G610 MP4, 4 ГБ LPDDR4X, 1000 МГц	YOLOv8n	2.9 ± 0.3	14–18	6–10 Вт
	YOLOv11n	2.7 ± 0.3	16–20	6–10 Вт

Модель YOLOv11n продемонструвала найвищий FPS на обох модифікаціях Orange Pi, що пояснюється її компактною NMS-free архітектурою. Обидві платформи є придатними для побудови недорогих автономних систем реального часу.

Raspberry Pi 5. Платформа Raspberry Pi 5 вирізняється надзвичайно низьким енергоспоживанням, що робить її актуальною для компактних пристроїв і сенсорних вузлів. Проте базовий чип VideoCore VII має обмежені можливості, тому реальний FPS є відносно низьким (Табл. 5). Підключення Hailo-8L (13 TOPS) AI-прискорювача радикально змінює продуктивність, дозволяючи Raspberry Pi 5 досягти рівня, порівнянного з повноцінними edge AI-платформами.

Таблиця 5

Результати вимірювання продуктивності на платформі Raspberry Pi 5

Конфігурація	YOLO версія	GFLOPS	FPS	Енергоспоживання
Raspberry Pi 5 (без прискорювача), VideoCore VII, 4ГБ LPDDR4X, 800 МГц	YOLOv8n	1.2 ± 0.2	7–10	4–6 Вт
	YOLOv11n	1.1 ± 0.2	8–11	4–6 Вт
Raspberry Pi 5 + AI HAT (Hailo-8L), 13 TOPS AI прискорювач + VideoCore VII	YOLOv5n (опт.)	8.7 ± 0.9	35–45	12–15 Вт
	YOLOv8n (опт.)	7.9 ± 0.8	40–50	12–15 Вт
	YOLOv11n (опт.)	7.2 ± 0.7	45–55	12–15 Вт

Rockchip-платформи. Rockchip пропонує низку економічних AI-рішень із вбудованими нейронними прискорювачами (NPU), які забезпечують стабільну інференс-продуктивність при мінімальному енергоспоживанні (Табл. 6). Ці рішення оптимальні для недорогих пристроїв виявлення об'єктів.

Зокрема, RV1106 демонструє вражаючу енергоефективність, що робить його потенційно придатним для застосування в мікродронах і системах безперервного відеоспостереження.

Таблиця 6

Результати вимірювання продуктивності на платформі Rockchip

Платформа	Характеристики	YOLOv8n GFLOPS	FPS	Потужність
RK3576 (CM5)	Mali-G52 MC3, NPU 6 TOPS	2.5 ± 0.3	13–16	5–8 Вт
RK3566 (3B)	Mali-G52 MC2, NPU 1 TOPS	1.7 ± 0.2	9–12	3–5 Вт
RV1106 (AI плата)	NPU 2 TOPS, ARM Cortex-A7	3.8 ± 0.4	22–28	2–4 Вт

4.4. Продуктивність на Google Coral. Google Coral — це лінійка апаратних рішень, що базуються на спеціалізованому Edge TPU прискорювачі, оптимізованому виключно для INT8-квантизованих нейронних мереж. Незважаючи на обмеження щодо підтримуваних операцій і розміру

моделі, Coral-платформи забезпечують вражаючу продуктивність при надзвичайно низькому енергоспоживанні.

Особливості Edge TPU. Платформи Coral демонструють максимальну ефективність за умови використання повністю квантизованих моделей (INT8), адаптованих до фіксованої роздільної здатності вхідних зображень (300×300 пікселів). Також варто враховувати обмеження розміру моделі — не більше 8 МБ для стабільної роботи.

Основні характеристики TPU:

- підтримка лише INT8 моделей, конвертованих із TensorFlow;
- фіксована роздільна здатність: 300×300 пікселів;
- максимальна утилізація TPU: 78–92 % залежно від складності моделі;
- обмеження розміру моделі: ~8 МБ.

Порівняльна продуктивність Coral-пристроїв. Результати вимірювання продуктивності пристроїв Google Coral наведена в Табл. 7. Модулі Mini PCIe та M.2 A+E демонструють найкраще співвідношення між латентністю, енергоспоживанням і вартістю. Зокрема, M.2-пристрої ідеально підходять для вбудованих рішень з обмеженим простором, таких як безпілотники та смарт-камери.

Таблиця 7

Аналіз продуктивності пристроїв Google Coral

Пристрій	YOLOv8n GFLOPS	FPS	Латентність	Потужність	Ціна
Dev Board	14.2 ± 0.9	48–58	17–21 мс	2–3 Вт	\$175
USB Accelerator	13.4 ± 1.0	42–52	19–24 мс	2.5 Вт	\$75
Mini PCIe	15.3 ± 0.8	52–62	16–19 мс	2 Вт	\$70
M.2 A+E	14.6 ± 0.7	48–58	17–21 мс	2 Вт	\$70
Dev Board Micro	11.4 ± 0.8	38–48	21–26 мс	1.5 Вт	\$130

Переваги та обмеження Coral. До переваг Google Coral відносяться:

- висока стабільність інференсу без джиттера;
- надзвичайно низьке енергоспоживання (до 2–3 Вт);
- мінімальна латентність (менше 20 мс);
- прості засоби інтеграції через TensorFlow Lite.

Обмеження платформи є такими:

- підтримуються лише INT8-квантизовані моделі;
- фіксована роздільна здатність 300×300 пікселів;
- не всі операції YOLO підтримуються Edge TPU;
- максимальний розмір моделі — близько 8 МБ, що обмежує використання складніших YOLO-архітектур.

Таким чином, Google Coral є одним із найефективніших рішень для задач комп'ютерного зору на низькоресурсних edge-пристроях, за умови попередньої оптимізації моделі.

4.5. Продуктивність на NVIDIA Jetson Orin

Платформи NVIDIA Jetson Orin представляють найвищий клас вбудованих AI-рішень, орієнтованих на завдання глибокого навчання в реальному часі. Завдяки підтримці екосистеми CUDA та інтеграції з TensorRT [12], модулі Orin забезпечують виняткову продуктивність при збереженні відносно низького енергоспоживання порівняно з традиційними GPU. Ці пристрої активно застосовуються в автономних транспортних системах, промисловій аналітиці, дронах та робототехніці.

Лінійка Jetson Orin. У Табл. 8 наведено технічні характеристики доступних моделей Jetson Orin, які відрізняються кількістю CUDA- та Tensor-ядер, піковою AI-продуктивністю та споживаною потужністю.

Таблиця 8

Технічні характеристики доступних моделей Jetson Orin

Модель	CUDA ядра	Tensor ядра	AI продуктивність	Частота GPU	Потужність	Ціна
AGX Orin 64 ГБ	2048	64	275 TOPS	1300 МГц	60W	\$2000
AGX Orin 32 ГБ	1792	56	200 TOPS	1100 МГц	40W	\$1600
Orin NX 16 ГБ	1024	32	100 TOPS	918 МГц	25W	\$800
Orin NX 8 ГБ	1024	32	70 TOPS	765 МГц	20W	\$600
Orin Nano 8 ГБ	1024	32	40 TOPS	625 МГц	15W	\$500

Orin Nano 4 ГБ	512	16	20 TOPS	625 МГц	10W	\$400
----------------	-----	----	---------	---------	-----	-------

Jetson Orin охоплює повний спектр застосувань — від важких індустріальних завдань до мобільних і компактних систем, що дозволяє масштабувати рішення в межах єдиної архітектури.

Детальна продуктивність AGX Orin 64GB. В Табл. 9 наведено результати тестування продуктивності на топовій моделі AGX Orin 64GB, яка забезпечує повноцінну підтримку великих моделей YOLO та максимально використовує можливості TensorRT.

Таблиця 9

Результати тестування продуктивності на моделі AGX Orin 64GB

YOLO версія	GFLOPS	FPS	Латентність	Розмір батчу	GPU утилізація
YOLOV4	89.3 ± 6.2	42–52	19–24 мс	2	75–82 %
YOLOV5n	28.7 ± 2.1	185–220	4.5–5.4 мс	4	45–52 %
YOLOV5s	52.8 ± 3.8	95–115	8.7–10.5 мс	4	58–65 %
YOLOv5x	198.4 ± 14.2	24–30	33.3–41.7 мс	1	94–98 %
YOLOV8n	24.6 ± 1.8	205–245	4.1–4.9 мс	4	45–52 %
YOLOV8s	46.3 ± 3.2	108–128	7.8–9.3 мс	4	68–75 %
YOLOV8l	186.9 ± 13.1	32–38	26.3–31.3 мс	2	94–98 %
YOLOV11n	20.1 ± 1.4	250–290	3.4–4.0 мс	4	42–48 %

Моделі YOLOv11n та YOLOv8n демонструють максимальний FPS із мінімальною латентністю, що робить їх найкращим вибором для завдань, чутливих до часу.

Ефект TensorRT-оптимізації. Використання TensorRT [12] знижує латентність та підвищує FPS, особливо у випадку оптимізації в режимі FP16. Для YOLOv8s отримано покращення, наведене в Табл. 10.

Таблиця 10

Результати використання TensorRT для YOLOv8s

Конфігурація	GFLOPS	FPS	Прискорення
Без TensorRT	29.1 ± 2.8	45–55	1.0x
3 TensorRT FP16	46.3±3.2	108–128	2.4x

Цей приклад підтверджує доцільність використання TensorRT навіть на високопродуктивних системах, де ефект від оптимізації залишається суттєвим.

4.5.4. Порівняння моделей Orin (YOLOv8n). У Табл. 11. наведене порівняння моделей Orin для YOLOv8n.

Таблиця 11

Порівняння моделей Orin для YOLOv8n

Модель	GFLOPS	FPS	Латентність	Потужність	\$/FPS
AGX Orin 64 GB	24.6 ± 1.8	205–245	4.1–4.9 мс	60 Вт	\$8.2
AGX Orin 32 GB	20.8 ± 1.6	175–205	4.9–5.7 мс	40 Вт	\$8.9
Orin NX 16 GB	15.8 ± 1.2	138–165	6.1–7.2 мс	25 Вт	\$5.3
Orin Nano 8 GB	11.0 ± 0.9	88–105	9.5–11.4 мс	15 Вт	\$5.3
Orin Nano 4 GB	7.4 ± 0.6	58–72	13.9–17.2 мс	10 Вт	\$6.2

Модель Orin NX 16GB є оптимальним вибором за критерієм “вартість/продуктивність” серед середнього класу пристроїв.

4.6. Комплексний аналіз для дронів

Вибір апаратної платформи для дронів повинен враховувати не лише обчислювальну продуктивність, але й габарити, вагу, енергоспоживання та інтеграційні можливості. У цьому розділі проведено оцінку платформ на основі вагових коефіцієнтів, релевантних для безпілотних літальних апаратів.

Специфіка застосування YOLO на дронах потребує особливої уваги до параметрів, наведених у Табл. 12.

Критерії оцінки для дронів

Критерій	Вага (%)	Вимоги	Обґрунтування
Продуктивність	25	> 30 FPS	Реалізація real-time навігації та виявлення
Енергоефективність	30	< 20 Вт	Забезпечення тривалої автономної роботи (більше 30 хвилин)
Габарити/вага	20	< 200 г	Мінімізація впливу на аеродинаміку
Ціна	15	< \$500	Доступність для масових рішень
Простота інтеграції	10	Готові рішення	Скорочення часу розробки

Оптимальні рішення для дронів включають Google Coral Dev Board Micro та NVIDIA Jetson Orin Nano 4GB. Результати вимірювання наведені в Табл. 13.

Таблиця 13

Оптимальні рішення для дронів

Рішення	Параметр	Значення	Бал	Рекомендації
Google Coral Dev Board Micro – загальний бал 8.7/10	Продуктивність (YOLOv8n)	11.4 ± 0.8 GFLOPS, 38–48 FPS	7/10	YOLO версія YOLOv8n (квантизована INT8), роздільна здатність 300 × 300. Для малих дронів (менше 2 кг), завдань патрулювання, моніторингу зображень у реальному часі.
	Енергоспоживання	1.5W	10/10	
	Розмір/вага	65 × 30 мм, 40 г	10/10	
	Ціна	\$130	8/10	
	Інтеграція	TensorFlow Lite готові моделі	9/10	
NVIDIA Jetson Orin Nano 4GB – загальний бал 8.2/10	Продуктивність (YOLOv8n)	7.4 ± 0.6 GFLOPS, 58–72 FPS	8/10	YOLO версія YOLOv8n/YOLOv11n з TensorRT. Роздільна здатність 640 × 640. Для середніх дронів (2–10 кг), зокрема для пошуково-рятувальних або промислових застосувань.
	Енергоспоживання	10 Вт	8/10	
	Розмір/вага	70 × 45 мм, 70 г	9/10	
	Ціна	\$400	6/10	
	Інтеграція	Повна підтримка CUDA та TensorRT	10/10	

5. Висновки

У межах даного дослідження було проведено комплексну оцінку продуктивності сучасних версій нейронних мереж YOLO на різних апаратних платформах — від настільних GPU до вбудованих рішень для дронів. Основні висновки:

- моделі YOLOv8n та YOLOv11n є найбільш збалансованими з точки зору продуктивності та енергоспоживання і рекомендовані для застосування у вбудованих системах реального часу;
- платформи Google Coral забезпечують стабільну інференс-продуктивність при мінімальній потужності, проте мають обмеження щодо типу моделей;
- Jetson Orin демонструє найвищу гнучкість і масштабованість, що дозволяє адаптувати систему під різні класи задач — від простих до високонавантажених;
- TensorRT-оптимізація істотно підвищує ефективність моделей і повинна використовуватись у всіх випадках, де це можливо;
- для дронів оптимальними залишаються Coral Dev Board Micro та Orin Nano 4GB, які найкраще поєднують вимоги до продуктивності, розміру та енергоспоживання;

Надалі доцільним є розширення дослідження на інші архітектури (MobileNet, EfficientDet) та вивчення впливу адаптивної квантизації і динамічного управління енергоспоживанням у польових умовах.

Список використаної літератури

1. Gallagher J. E., Oughton E. J. Surveying You Only Look Once (YOLO) multispectral object detection advancements, applications and challenges. IEEE Access. 2025. Vol. 13. P. 7366–7395. <https://doi.org/10.1109/access.2025.3526458>
2. Bo C., Wei Y., Wang X., Shi Z., Xiao Y. Vision-based anti-UAV detection based on YOLOv7-GS in complex backgrounds. Drones. 2024. Vol. 8, № 7. P. 1–21. <https://doi.org/10.3390/drones8070331>
3. Luo X., Tian X., Zhang H., Hou W., Leng G., Xu W., Jia H., He X., Wang M., Zhang J. Fast automatic vehicle detection in UAV images using convolutional neural networks. Remote Sensing. 2020. Vol. 12, № 12. P. 1–15. <https://doi.org/10.3390/rs12121994>
4. Redmon J., Divvala S., Girshick R., Farhadi A. You only look once: unified, real-time object detection. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 27–30, 2016, Las Vegas, NV, USA. P. 779–788. <https://doi.org/10.1109/CVPR.2016.91>
5. Lazarevich I., Grimaldi M., Kumar R., Mitra S., Khan S., Sah S. YOLOBench: benchmarking efficient object detectors on embedded systems. 2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), October 2–6, 2023, Paris, France. P. 1161–1170. <https://doi.org/10.1109/ICCVW60793.2023.00126>
6. Cantero D., Esnaola Gonzalez I., Miguel Alonso J., Jauregi E. Benchmarking object detection deep learning models in embedded devices. Sensors. 2022. Vol. 22, № 11. <https://doi.org/10.3390/s22114205>
7. YOLOv5 by Ultralytics, GitHub repository, 2020. <https://github.com/ultralytics/yolov5>
8. Wang C.-Y., Bochkovskiy A.-I., Liao H.-Y. M. YOLOv7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 17–24, 2023, Vancouver, Canada. P. 7464–7475. <https://doi.org/10.1109/CVPR52729.2023.00721>
9. Selcuk B., Serif T. A comparison of YOLOv5 and YOLOv8 in the context of mobile UI detection. Mobile Web and Intelligent Information Systems. MobiWIS 2023. Lecture Notes in Computer Science. 2023. Vol. 13977. P. 161–174. https://doi.org/10.1007/978-3-031-39764-6_11
10. Lin T. -Y., Goyal P., Girshick R., He K., Dollár P. Focal loss for dense object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2020. Vol. 42, № 2. P. 318–327. <https://doi.org/10.1109/TPAMI.2018.2858826>
11. Khanam R., Hussain M. YOLOv11: an overview of the key architectural enhancements, arXiv:2410.17725v1. 2024. <https://doi.org/10.48550/arXiv.2410.17725>
12. NVIDIA, TensorRT developer guide, NVIDIA Corporation, 2024. <https://docs.nvidia.com/deeplearning/tensorrt/archives/tensorrt-825/developer-guide/index.html>