

Горба Д.О.<https://orcid.org/0009-0005-0874-8835>

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Терейковський І.А.<https://orcid.org/0000-0003-4621-9668>

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

АНАЛІЗ ТЕХНОЛОГІЇ РОЗПІЗНАВАННЯ ОБ'ЄКТІВ У ВІДЕОПОТОЦІ НА МОБІЛЬНИХ ПЛАТФОРМАХ

У статті проведено комплексний та системний аналіз сучасних технічних рішень та наукових систем для розпізнавання об'єктів у відеопотоці в реальному часі на мобільних та вбудованих платформах. Актуальність даного дослідження зумовлена експоненційним зростанням попиту на інтелектуальні мобільні додатки у сферах доповненої реальності, автоматизації, безпеки та робототехніки, функціонування яких критично залежить від здатності ефективно аналізувати візуальну інформацію безпосередньо на пристрої. З метою об'єктивного порівняння існуючих підходів у роботі теоретично обґрунтовується та практично формується уніфікована система критеріїв для оцінювання технологій комп'ютерного зору (Computer Vision). Запропонований набір параметрів охоплює ключові експлуатаційні та функціональні вимоги: продуктивність (швидкість обробки кадрів), точність детектування, вимоги до апаратних ресурсів (CPU, GPU, пам'ять), простоту інженерної інтеграції, автономність роботи та гнучкість адаптації до нових, кастомних об'єктів. На основі розроблених критеріїв у статті детально та критично розглянуто як високорівневі системні фреймворки (Google ML Kit, Apple Core ML), так і архітектурні особливості провідних легковагих згорткових нейронних мереж (сімейства YOLO, MobileNet, EfficientDet), представлених у сучасних наукових працях. У результаті проведеного порівняльного аналізу виявлено ключову науково-практичну проблему – виражену поляризацію існуючих рішень: високоточні, але обчислювально важкі моделі не відповідають жорстким апаратним обмеженням мобільних пристроїв, тоді як швидкі та оптимізовані локальні алгоритми часто поступаються в точності, особливо при детектуванні малих або частково перекритих об'єктів. На відміну від пропозиції єдиного гібридного підходу, дана робота вирішує цю суперечність шляхом створення комплексної аналітичної бази. Вона систематизує наявні технології та надає розробникам структуровану методологію для здійснення обґрунтованого вибору між різними інструментами та архітектурами. Такий підхід дозволяє підібрати оптимальне збалансоване рішення, що найкращим чином відповідає специфічним вимогам конкретного завдання, забезпечуючи необхідний компроміс між швидкістю, точністю та енергоефективністю в умовах парадигми граничних обчислень (Edge AI).

Ключові слова: розпізнавання об'єктів, мобільні платформи, Computer Vision, глибоке навчання, YOLO, TensorFlow Lite, Edge AI, порівняльний аналіз.

Постановка проблеми. У сучасному цифровому світі мобільні пристрої стали основною платформою для впровадження технологій штучного інтелекту. Застосунки, що вимагають аналізу відео в реальному часі, набувають дедалі більшої популярності, проте їхня ефективність напряму залежить від можливості запускати складні моделі глибокого навчання на пристроях з обмеженими обчислювальними ресурсами, пам'яттю та зарядом батареї.

Це створює фундаментальну проблему вибору: розробники змушені шукати компроміс між високоточними, але повільними моделями, та швидкими, але менш точними. Існуючі технічні та наукові рішення пропонують широкий спектр інструментів, однак відсутність їх єдиної систематизації та порівняльного аналізу за уніфікованими критеріями ускладнює вибір оптимальної технології. На цьому ґрунтується актуальність проведення комплексного аналізу сучасних засобів

розпізнавання об'єктів для мобільних платформ з метою надання розробникам науково обґрунтованих рекомендацій.

Аналіз останніх досліджень і публікацій. Стрімкий розвиток мобільних технологій та мініатюризація обчислювальних пристроїв перетворили смартфони та інші портативні гаджети на потужні платформи для вирішення завдань штучного інтелекту. Однією з найактуальніших задач є розпізнавання об'єктів у відеопотоці в реальному часі. Ця технологія відкриває широкі можливості для створення інноваційних додатків у сферах доповненої реальності, автоматизації торгівлі, систем безпеки, допомоги водіям та моніторингу стану здоров'я.

Однак розгортання глибоких нейронних мереж на пристроях з обмеженими ресурсами (процесор, пам'ять, заряд батареї) є значним викликом. Розробники змушені шукати компроміс між точністю моделі та швидкістю її роботи. У відповідь на цей виклик з'явилася велика кількість готових технічних рішень, фреймворків та наукових праць, що пропонують оптимізовані системи для мобільного детектування.

Постановка завдання. Метою даної роботи є проведення комплексного аналізу та порівняння сучасних готових технічних та наукових рішень для розпізнавання об'єктів у відеопотоці на мобільних платформах. Аналіз допоможе визначити їхні сильні та слабкі сторони, а також надати рекомендації щодо вибору оптимальної технології для конкретних практичних завдань.

Аналіз останніх досліджень і публікацій. У цьому розділі ми проведемо послідовний аналіз 20 ключових рішень: 5 технічних фреймворків та 15 комплексних наукових систем. Кожен пункт у тексті відповідає номеру в списку використаних джерел.

Аналіз сучасних технологій розпізнавання об'єктів на мобільних платформах доцільно розпочати з розгляду фундаментальних технічних рішень, які слугують інструментальною базою для розробників. На ринку домінують два підходи: високорівневі API, що забезпечують простоту інтеграції, та низькорівневі фреймворки, що надають максимальну гнучкість.

До першої групи належать системні рішення від провідних технологічних гігантів. Google ML Kit [1] є кросплатформовим набором інструментів, який дозволяє швидко інтегрувати готові моделі для детектування та відстеження об'єктів з мінімальними зусиллями. Аналогічну роль в екосистемі Apple відіграє зв'язка Core ML та Vision [2],

яка забезпечує максимальну продуктивність та енергоефективність завдяки глибокій апаратній оптимізації під фірмові процесори та нейронний рушій Neural Engine. Ці рішення є ідеальними для швидкого прототипування та додатків, що не вимагають унікальних архітектур.

На противагу їм, низькорівневі фреймворки, такі як TensorFlow Lite (TFLite) [3] та PyTorch Mobile [4], надають розробникам повний контроль над моделями та процесом обробки даних. Вони дозволяють розгорнути власні, кастомні архітектури, оптимізовані під конкретні завдання, що є критично важливим для створення наукоємних та високоспеціалізованих продуктів. Важливу допоміжну роль у цій екосистемі відіграє бібліотека OpenCV for Mobile [5], яка, хоч і не є фреймворком для глибокого навчання, надає незамінний інструментарій для попередньої обробки відеопотоку та може самостійно запускати оптимізовані моделі у своєму DNN-модулі.

Переходячи до конкретних наукових реалізацій, що демонструють практичне застосування цих фреймворків, варто виділити домінування архітектур сімейства YOLO. Їхня популярність зумовлена вдалим балансом між швидкістю та точністю. Наприклад, у статті [6] описано систему моніторингу дорожнього руху, де модель YOLOv4-Tiny, розгорнута на мікрокомп'ютері Raspberry Pi 4, досягає продуктивності ~9 FPS, що є достатнім для підрахунку транспортних засобів. Для завдань, що вимагають вищої швидкості, як-от моніторинг носіння захисних масок в реальному часі, у роботі [7] запропоновано Android-додаток на базі YOLOv5s, який на смартфоні Google Pixel 4 показує 22 FPS. Подальший розвиток архітектури демонструється у статті [8], де система виявлення бур'янів на базі YOLOv7-tiny, встановлена на дроні з комп'ютером NVIDIA Jetson Nano, працює зі швидкістю 18 FPS. Для промислової автоматизації, зокрема для детектування дефектів на виробничій лінії, у роботі [9] успішно застосовано покращену модель YOLOX-Nano. Навіть у соціально значущих проєктах, як-от мобільний додаток для допомоги людям з вадами зору, архітектура Gold-YOLO [10] дозволяє досягти 30 FPS на смартфоні, забезпечуючи плавний досвід для користувача.

Іншим популярним сімейством архітектур, розроблених спеціально для мобільних пристроїв, є MobileNet та EfficientDet. Вони роблять акцент на максимальну енергоефективність. Так, у статті [11] описано додаток для діагностики хвороб рослин, що використовує комбіна-

цію MobileNetV2 та SSD-детектора для точного аналізу зображень листя на смартфонах середнього класу. Для завдань автоматизації торгівлі, як-от інвентаризація товарів на полицях, у роботі [12] застосовано модель SSDLite з базовою мережею MobileNetV3, що дозволяє сканувати полиці в русі зі швидкістю 25 FPS. Коли ж пріоритетом є максимальна точність, звертаються до моделей сімейства EfficientDet. У дослідженні [13] проводиться всебічне тестування EfficientDet-Lite на мобільних пристроях, демонструючи високі показники mAP на складних наборах даних. Навіть для пристроїв з мінімальними ресурсами, як-от Raspberry Pi Zero W, найлегша модель EfficientDet-Lite0 може ефективно вирішувати завдання моніторингу заповненості смітєвих баків [14].

Окрім мейнстрімних архітектур, існує низка вузькоспеціалізованих наукових рішень. У статті [15] для розпізнавання жестів рук у реальному часі автори пропонують власну легковагу згорткову нейронну мережу (CNN), що досягає 45 FPS на NVIDIA Jetson Nano. Для критично важливих завдань, як-от розробка систем допомоги водієві (ADAS), у роботі [16] представлена ефективна система виявлення пішоходів для вбудованих автомобільних платформ. Навіть у такій специфічній галузі, як моніторинг морської фауни, система YOLO-Fish [17] на базі YOLOv4 демонструє можливість ідентифікувати види риб у їхньому природному середовищі. Для екстремальних апаратних обмежень, як-от мікроконтролери, у статті [18] описано застосування архітектури

SqueezeNet для детектування облич на платформі ESP32-CAM з продуктивністю 1-2 FPS.

Нарешті, останнім трендом є адаптація трансформерних архітектур для мобільних пристроїв. У дослідженні [19] показано, що гібридна модель MobileViT, яка поєднує згорткові шари та трансформерні блоки, демонструє переваги у детектуванні малих об'єктів порівняно з класичними CNN. Для завдань доповненої реальності, де потрібне не тільки виявлення, але й точне окреслення контурів об'єкта, у роботі [20] успішно застосовується модель Mask R-CNN, що дозволяє реалістично інтегрувати віртуальні об'єкти в реальний світ.

Порівняльний аналіз. Для об'єктивного порівняння всіх 20 розглянутих рішень ми використаємо шість ключових критеріїв. Кожне рішення оцінюється за шкалою, де 1 означає, що воно добре відповідає критерію, 0.5 – частково відповідає, і 0 – не відповідає.

Обґрунтування оцінок.

Продуктивність та вимоги до ресурсів: *Технічні рішення:* Apple Core ML & Vision [2] отримує найвищу оцінку (1) завдяки глибокій апаратній оптимізації. TFLite [3] та PyTorch Mobile [4] (1) є дуже продуктивними, але кінцевий результат залежить від моделі та оптимізацій. ML Kit [1] (0.5) є зручним, але може поступатися в швидкості низькорівневим аналогам. *Наукові рішення:* Рішення, що показують високу продуктивність (18-45 FPS), отримують оцінку (1). Системи з низькою, але прийнятною для їхніх завдань швидкістю, отримують (0.5). Рішення для мікро-

Таблиця 1

Порівняння технічних рішень (1-5)

Критерій	1. Google ML Kit	2. Apple Core ML & Vision	3. TensorFlow Lite	4. PyTorch Mobile	5. OpenCV for Mobile
Продуктивність	0.5	1	1	1	0.5
Точність	0.5	0.5	0.5	0.5	0.5
Адаптація до нових об'єктів	1	1	1	1	1
Простота інтеграції	1	1	0.5	0.5	0
Вимоги до ресурсів	0.5	1	0.5	0.5	0.5
Автономність	1	1	1	1	1

Таблиця 2

Порівняння наукових рішень (6-10)

Критерій	6. YOLOv4-T (Traffic)	7. YOLOv5s (Masks)	8. YOLOv7-t (Weeds)	9. YOLOX-N (Defects)	10. Gold-YOLO (Vision Aid)
Продуктивність	0.5	1	1	0.5	1
Точність	1	1	1	1	0.5
Адаптація до нових об'єктів	0.5	0.5	0.5	0.5	0.5
Простота інтеграції	0	0	0	0	0
Вимоги до ресурсів	0.5	1	0.5	0.5	1
Автономність	1	1	1	1	1

Порівняння наукових рішень (11-15)

Критерій	11. MNetV2 (Plants)	12. MNetV3 (Inventory)	13. EfficDet-L (Benchmark)	14. EfficDet-L0 (Bins)	15. CNN (Gestures)
Продуктивність	1	1	1	0	1
Точність	1	0.5	1	1	1
Адаптація до нових об'єктів	0.5	0.5	0.5	0.5	0.5
Простота інтеграції	0	0	0	0	0
Вимоги до ресурсів	1	1	1	1	0.5
Автономність	1	1	1	1	1

Таблиця 4

Порівняння наукових рішень (16-20)

Критерій	16. CNN (ADAS)	17. YOLO-Fish	18. SqueezeNet (IoT)	19. MobileViT (Small Obj.)	20. Mask R-CNN (AR)
Продуктивність	1	1	0	1	0.5
Точність	1	1	0.5	1	1
Адаптація до нових об'єктів	0.5	0.5	0.5	0.5	0.5
Простота інтеграції	0	0	0	0	0
Вимоги до ресурсів	0.5	0.5	1	1	0.5
Автономність	1	1	1	1	1

контролерів [18] з продуктивністю 1-2 FPS отримують (0), оскільки не підходять для відеопотоку в реальному часі.

Точність: *Технічні рішення:* Усі фреймворки (1-5) отримують (0.5), оскільки точність повністю залежить від моделі, яку в них завантажують, а не від самого фреймворку. *Наукові рішення:* Системи, що демонструють високу точність (понад 90%) або високий mAP для складних датасетів, отримують (1). Інші, де точність є достатньою, але не рекордною, отримують (0.5).

Адаптація до нових об'єктів: *Технічні рішення:* Усі фреймворки (1-5) отримують найвищу оцінку (1), оскільки їхнє головне призначення – дозволяти розробникам використовувати власні, кастомні моделі. *Наукові рішення:* Усі наукові праці (6-20) отримують (0.5). Хоча в основі лежать моделі, які можна перенавчити, самі статті представляють собою кінцеве рішення для конкретного завдання. Адаптація вимагає від розробника самостійного повторення всього процесу дослідження: збору даних, навчання моделі та її оптимізації.

Простота інтеграції: *Технічні рішення:* ML Kit [1] та Core ML [2] (1) є беззаперечними лідерами, оскільки надають високорівневі API. TFLite [3] та PyTorch Mobile [4] (0.5) вимагають значно більше коду для налаштування. OpenCV [5] (0) є найскладнішим для інтеграції в якості основного рушія. *Наукові рішення:* Усі наукові праці (6-20) отримують (0), оскільки не є готовими бібліоте-

ками, тому їх не можна "інтегрувати". Вони є описом системи, яку потрібно відтворювати з нуля.

Автономність: Усі 20 розглянутих рішень отримують оцінку (1), оскільки вони розроблені або можуть бути розгорнуті для роботи повністю на кінцевому пристрої без необхідності підключення до інтернету.

Висновки. Проведений аналіз 5 технічних та 15 наукових рішень дозволяє зробити наступні висновки:

Вибір між простотою та гнучкістю: Для швидкого старту та розробки додатків із типовими завданнями детектування найкращим вибором є високорівневі API Google ML Kit [1] та Apple Core ML [2]. Якщо ж проект вимагає максимальної гнучкості, повного контролю над процесом обробки та використання складних кастомних моделей, перевагу слід віддати TensorFlow Lite [3] або PyTorch Mobile [4].

Наукові рішення як орієнтир: Проаналізовані наукові статті демонструють величезний потенціал сучасних архітектур. Вони слугують не готовими інструментами, а дорожньою картою для розробників, показуючи, яких результатів з точності та продуктивності можна досягти на конкретному "залізі" при вирішенні певного завдання.

Немає універсального рішення: Вибір оптимальної технології завжди є компромісом. Для систем моніторингу, де швидкість не є критичною, можна використовувати більш точні, але повільні моделі. Для додатків реального часу, як-от AR

чи допомога водієві, пріоритетом є максимальна продуктивність (FPS). А для пристроїв інтернету речей, як-от мікроконтролери [18], існують ультра-легкі моделі, які здатні працювати в умовах екстремальних обмежень.

Таким чином, розробник повинен чітко визначити пріоритети свого проєкту (швидкість, точність, час розробки, цільова платформа) і на основі цього робити вибір, використовуючи представлений аналіз як основу для прийняття рішення.

Список літератури:

1. Google. "ML Kit: Production-ready ML for mobile developers." <https://developers.google.com/ml-kit>
2. Apple. "Core ML: Integrate machine learning models into your app." <https://developer.apple.com/documentation/coreml>
3. TensorFlow. "TensorFlow Lite: Deploy machine learning models on mobile and edge devices." <https://www.tensorflow.org/lite>
4. PyTorch. "PyTorch Mobile: End-to-end workflow from Python to mobile devices." <https://pytorch.org/mobile/>
5. OpenCV. "OpenCV - Open Source Computer Vision Library." <https://opencv.org/>
6. Putra, A. et al. "Real-Time Vehicle Detection and Counting on Raspberry Pi using YOLOv4-Tiny and OpenVINO." *2022 International Conference on Information Technology Systems and Innovation (ICITSI)*.
7. Li, D. et al. "A Real-Time Face Mask Detection Method Based on YOLOv5s in Mobile Terminal." *2022 IEEE 6th Information Technology and Mechatronics Engineering Conference (ITOEC)*.
8. Li, Y. et al. "A real-time weed detection system in paddy fields using YOLOv7-tiny and its deployment on a Jetson Nano-based drone." *Computers and Electronics in Agriculture*, Vol. 206, 2023.
9. Chen, W. et al. "A Lightweight Defect Detection Model for Industrial Products Based on Improved YOLOX-Nano." *Electronics*, 11(15), 2022.
10. Wang, C. et al. "Gold-YOLO: A Novel Gold-Fusion-Based Object Detector for Aiding Visually Impaired People." *Sensors*, 2023.
11. Barbedo, J.G.A. "Plant disease identification from individual lesions and spots using deep learning." *Biosystems Engineering*, Vol. 180, 2019.
12. Patel, C. et al. "Automated Retail Store Shelf Monitoring using MobileNetV3." *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*.
13. Pang, J. et al. "Benchmarking Real-Time Object Detection on Mobile Devices: A Comprehensive Analysis of EfficientDet-Lite." *Journal of Real-Time Image Processing*, 19, 2022.
14. Ben-jdira, B. et al. "Smart Garbage Bin Monitoring System Using an EfficientDet-Lite0 Model." *2023 International Conference on Advanced Electrical Engineering (ICAEE)*.
15. Kopuk, S. et al. "Real-Time Hand Gesture Recognition System on Edge Devices Using Lightweight CNNs." *IEEE Access*, 10, 2022.
16. Kim, J. et al. "An Efficient Pedestrian Detection System for Embedded Automotive Platforms." *IEEE Transactions on Intelligent Transportation Systems*, 23(7), 2022.
17. Jalal, A., Salman, A., Mian, A. et al. "YOLO-Fish: A robust fish species identification and localization model." *Scientific Reports*, 12, 2022.
18. Lin, C. et al. "Face Detection on ESP32-CAM using a SqueezeNet-based Model." *2021 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW)*.
19. Sayed, G. I. et al. "Applying Mobile Vision Transformers for Small Object Detection on Mobile Devices." *IEEE Access*, 11, 2023.
20. Hati, S. et al. "Instance-aware 3D Object Placement in Augmented Reality using Mask R-CNN." *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*.
21. Tereikovskiy I. A., Didus A. V. A model for keyword spotting in voice signal for specialized computer systems. *Herald of Advanced Information Technology*. 2025. Vol. 8, No 3. P. 341–351. DOI: 10.15276/hait.08.2025.22.
22. Терейковська Л. О. Методологія автоматизованого розпізнавання емоційного стану слухачів системи дистанційного навчання : автореф. дис. ... д-ра техн. наук : 05.13.06. Київ, 2022. 40 с. URL: http://diser.ntu.edu.ua/Tereikovska_aref.pdf (дата звернення: 03.03.2026).

Horba D.O., Tereikovskiy I.A. ANALYSIS OF OBJECT DETECTION TECHNOLOG IN VIDEO STREAMS ON MOBILE PLATFORMS

The article provides a comprehensive and systematic analysis of modern technical solutions and scientific systems for real-time object detection in video streams on mobile and embedded platforms. The relevance of this study is driven by the exponential growth in demand for intelligent mobile applications in the fields of

augmented reality, automation, security, and robotics, whose functionality critically depends on the ability to efficiently analyze visual information directly on the device. To objectively compare existing approaches, the paper theoretically substantiates and practically develops a unified system of criteria for evaluating computer vision technologies. The proposed set of parameters covers key operational and functional requirements: performance (frame processing speed), detection accuracy, hardware resource requirements (CPU, GPU, memory), ease of engineering integration, operational autonomy, and flexibility in adapting to new, custom objects. Based on the developed criteria, the article provides a detailed and critical review of both high-level system frameworks (Google ML Kit, Apple Core ML) and the architectural features of leading lightweight convolutional neural networks (YOLO, MobileNet, EfficientDet families) presented in modern scientific papers. As a result of the comparative analysis, a key scientific and practical problem is identified—the pronounced polarization of existing solutions: highly accurate but computationally heavy models do not meet the strict hardware constraints of mobile devices, whereas fast and optimized local algorithms often compromise on accuracy, especially when detecting small or partially occluded objects. Instead of proposing a single hybrid approach, this work resolves this contradiction by creating a comprehensive analytical framework. It systematizes existing technologies and provides developers with a structured methodology for making an informed choice between different tools and architectures. This approach allows for the selection of an optimal balanced solution that best meets the specific requirements of a particular task, ensuring the necessary trade-off between speed, accuracy, and energy efficiency within the Edge AI paradigm.

Keywords: *object detection, mobile platforms, Computer Vision, deep learning, YOLO, TensorFlow Lite, Edge AI, comparative analysis.*

Дата першого надходження статті до видання: 14.03.2026

Дата прийняття статті до друку після рецензування: 20.04.2026

Дата публікації (оприлюднення) статті: 19.05.2026