УДК 004.912

Аспірант Горба Д.О., д.т.н., професор Терейковський І. А.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

ПОСТАНОВКА ЗАДАЧІ РОЗРОБКИ НЕЙРОМЕРЕЖЕВИХ ЗАСОБІВ ДЕТЕКТУВАННЯ ОБ'ЄКТІВ У ВІДЕОПОТОЦІ

Abstract

Horba D., Tereikovskiy I.
Problem statement of developing neural network tools for detecting objects in a video stream.

This paper examines methods for designing and implementing neural network models capable of detecting and tracking objects in real-time video streams. It compares popular deep learning approaches such as YOLO, SSD, and Faster R-CNN in terms of accuracy, speed, and computational efficiency. The study also discusses optimization techniques for deployment on edge and cloud platforms, focusing on model compression and GPU acceleration. Applications in surveillance, autonomous systems, and industrial automation are explored, along with challenges related to scalability, latency, and data privacy.

Вступ

У сфері комп’ютерного зору задача виявлення та відстеження об’єктів у відеопотоці є фундаментальною для побудови інтелектуальних систем. Зростання обсягів відеоданих і потреба в обробці в реальному часі висувають суворі вимоги до точності, швидкодії та апаратної ефективності нейромережевих моделей. Поширення периферійних (edge) обчислень і хмарних сервісів відкриває нові можливості для масштабування, але одночасно створює виклики щодо затримок, конфіденційності та вартості обчислень. Порівняльний аналіз сучасних архітектур, таких як YOLO, SSD і Faster R-CNN, дозволяє визначити оптимальні підходи для конкретних сценаріїв і апаратних платформ.

Постановка задачі

Мета дослідження – сформулювати проблему розробки нейромережевих інструментів для виявлення об’єктів у відеопотоці, визначити критерії ефективності та проаналізувати можливі шляхи оптимізації. Основні задачі:

1. Проаналізувати сучасні архітектури детекції об’єктів (YOLO, SSD, Faster R-CNN) за показниками точності, швидкості та ресурсоємності.

2. Дослідити алгоритмічні та апаратні методи оптимізації (квантування, прунинг, GPU/TPU-акселерація) для розгортання на edge- та хмарних платформах.
3. Оцінити застосування моделей у системах відеоспостереження, автономних роботизованих комплексах і промисловій автоматизації.
4. Виявити ключові виклики, пов'язані з масштабованістю, низькою затримкою обробки та забезпеченням приватності даних.

Термінологія

Детектор об'єктів (Object Detector) – модель машинного навчання, яка визначає наявність і координати об'єктів на зображенні або у відеокадрі.
Inference Time – час, необхідний моделі для обробки одного кадру; критичний показник для систем реального часу.
Edge Computing – обробка даних на периферійних пристроях, близьких до джерела даних, для зменшення затримки та навантаження на мережу.
Model Compression – сукупність методів зменшення розміру та обчислювальної складності моделі (квантування, прунинг, distillation).
Latency – затримка між надходженням відеокадру та отриманням результату детекції; ключова характеристика для часових обмежених систем.
YOLO (You Only Look Once) – однопрохідна архітектура детекції об'єктів, яка одночасно виконує локалізацію та класифікацію в межах єдиної мережі.
SSD (Single Shot Multibox Detector) – метод детекції, що використовує багатомасштабні ознаки та одноетапне прогнозування боксів і класів, поєднуючи компроміс між точністю та продуктивністю.
GPU (Graphics Processing Unit) – багатоядерний процесор, оптимізований для паралельних обчислень, який прискорює навчання та inference глибинних нейронних мереж.
Faster R-CNN (Faster Region-based Convolutional Neural Network) – двоетапна архітектура детекції, яка спершу генерує регіони-гіпотези за допомогою регіональної мережі пропозицій, а потім виконує класифікацію та уточнення меж об'єктів.
Федеративне навчання (Federated Learning) – це метод розподіленого машинного навчання, за якого модель тренується на периферійних вузлах без передачі сирих даних на центральний сервер, що підвищує рівень конфіденційності.
Knowledge Distillation – це техніка передавання знань від великої, заздалегідь натренованої моделі ("вчителя") до компактнішої моделі ("учня"), яка вчиться відтворювати поведінку вчителя при меншій ресурсоємності.

Інференс (Inference) – це процес використання натренованої нейромережевої моделі для обробки нових даних з метою отримання прогнозів або детекцій.

Аналіз існуючих підходів та алгоритмів

Сучасні системи детекції об’єктів у відеопотоці спираються на глибокі згорткові нейромережі й варіюються між одноетапними та двоетапними архітектурами. Однопрохідні моделі, такі як сімейства YOLO та SSD, виконують локалізацію й класифікацію в межах одного проходу мережі, забезпечуючи прийнятну точність за реального часу обробки навіть на обмежених обчислювальних ресурсах [1,2]. Двоетапні підходи на кшталт Faster R-CNN реалізують попереднє генерування регіонів-гіпотез, після чого уточнюють класи й координати, що дає вищу точність на складних сценах, але збільшує inference time [3]. Узагальнені характеристики популярних детекторів наведено у таблиці 1.

Таблиця 1

Характеристики популярних моделей детекторів об’єктів

Модель	Ключові переваги	Обмеження	Типові платформи та сценарії
YOLO (v3–v8)	Інференс у реальному часі, моно-кадрова обробка, проста інтеграція	Падіння точності на дрібних об’єктах при низькій роздільній здатності	Відеоспостереження, робототехніка, UAV
SSD	Баланс точність/швидкодія, підтримка багатомасштабних ознак	Відстає по швидкості від YOLO та по точності від Faster R-CNN	Мобільні й edge-пристрої, AR/VR
Faster R-CNN	Висока точність, адаптивний backbone, стійкість до складних сцен	Значні обчислювальні витрати, вищі затримки	Промислова інспекція, офлайн-аналітика

Завдання відеоаналітики виходить за межі одиничного кадру, тому до детекторів додають трекери, які мінімізують пропуски та стабілізують траєкторії об’єктів. Поєднання детекції з алгоритмами на кшталт DeepSORT дозволяє підтримувати сталий ID-лічильник, компенсувати тимчасові втрати видимості й підвищувати загальну надійність системи [4]. Для

сценаріїв з розподіленою інфраструктурою дедалі частіше використовуються гібридні конвеєри, де периферійні вузли виконують первинну детекцію, а хмарні сервіси беруть на себе складнішу аналітику, агрегування даних та довготривале зберігання. Такий підхід дозволяє балансувати між латентністю, вартістю та конфіденційністю.

Подальша оптимізація моделей спрямована на зменшення обчислювального навантаження без критичного погіршення точності. Методи прунингу, квантування та knowledge distillation компактно кодують ваги й скорочують кількість операцій [6]. На етапі розгортання застосовують фреймворки типу TensorRT, які виконують поєднання шарів, перенумерацію тензорів та змішану точність, прискорюючи інференс на GPU й спеціалізованих акселераторах [7]. У таблиці 2 узагальнено рекомендації щодо вибору оптимізацій залежно від обмежень платформи. Дизайн систем реального часу також враховує архітектурні оптимізації (memory-aware scheduling, pipeline parallelism), адже навіть незначні затримки накопичуються при масштабуванні до десятків потоків [8].

Масштабованість та приватність даних стають визначальними чинниками для промислових і міських розгортань. Edge-засоби дозволяють попередньо фільтрувати й анонімізувати події, зменшуючи обсяг інформації, що потрапляє у хмару [5]. Індустрія впроваджує федеративне навчання та механізми диференційної приватності, щоб оновлювати моделі на численних вузлах без передачі сирих відеоданих [9]. Комплексне поєднання архітектурних, алгоритмічних й організаційних заходів забезпечує баланс між точністю, латентністю та безпекою, але потребує адаптації під конкретні сценарії використання.

Таблиця 2

Оптимізаційні техніки для розгортання детекторів

Техніка	Короткий опис	Очікуваний ефект	Доцільність застосування
Прунинг ваг	Видалення маловпливових каналів/фільтрів	Зменшення кількості операцій FLOPs (до 50%) при незначній втраті точності	Edge-пристрої, CPU/GPU
Квантування до INT8	Зниження розрядності ваг та активацій	Зменшення використання пам’яті та підвищення пропускної здатності	Edge TPU, TensorRT, мобільні SoC

Knowledge Distillation	Навчання компактної моделі за “вчителем”	Збереження точності при меншому розмірі та обчислювальній складності	Хмара → edge/мобільні пристрої
TensorRT / GPU-оптимізація	Fusion шарів, Mixed Precision, оптимізація графа	Підвищення швидкодії (у 2–4 рази) та зменшення затримки обробки	Центри обробки даних, відеоаналітика
Pipeline Parallelism	Рознесення етапів обробки між вузлами / потоками	Стабілізація затримки та забезпечення гнучкого масштабування	Багатопотокові системи, відеострічки
Federated Learning	Розподілене навчання без передачі сирих даних	Підвищення рівня приватності та адаптація до локальних даних	Мережі камер, корпоративні середовища

Висновки

Проведений аналіз засвідчив, що сучасні підходи до детекції об’єктів у відеопотоці охоплюють широке коло архітектур і технологій, які по-різному балансують між точністю, швидкодією та ресурсоємністю. Одноетапні моделі (YOLO, SSD) забезпечують високу продуктивність і придатні для сценаріїв реального часу та edge-розгортань, тоді як двоетапні рішення (Faster R-CNN) залишаються незамінними для застосувань із підвищеними вимогами до точності. Комплексна інтеграція детекторів із трекерами й гібридними edge–cloud конвеєрами дає змогу зменшити затримку, стабілізувати траєкторії об’єктів і оптимізувати використання мережевих ресурсів. Разом із тим ці рішення стикаються з низкою обмежень: висока вартість обробки відеопотоків у хмарі, складність масштабування на десятки й сотні камер, обмеження апаратних ресурсів периферійних пристроїв та вимоги до захисту персональних даних. Зазначені фактори актуалізують потребу в подальших дослідженнях, спрямованих на ефективніші методи компресії моделей, розвинені механізми федеративного навчання, стандартизовані конвеєри розгортання й комбіновані алгоритми детекції та відстеження, здатні адаптуватися до динамічних умов і різноманітних середовищ. Удосконалення цих підходів сприятиме створенню більш гнучких і надійних систем відеоаналітики для промисловості, безпеки та автономних платформ.

Література

1. Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. arXiv:1804.02767. https://arxiv.org/abs/1804.02767
2. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). SSD: Single Shot MultiBox Detector. In ECCV 2016. https://doi.org/10.1007/978-3-319-46448-0_2
3. Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In NeurIPS 2015. https://arxiv.org/abs/1506.01497
4. Wojke, N., Bewley, A., & Paulus, D. (2017). Simple Online and Realtime Tracking with a Deep Association Metric. IEEE ICIP 2017. https://doi.org/10.1109/ICIP.2017.8296962
5. Li, X., Wu, J., & Zhang, J. (2021). A Cloud-Edge Collaborative Inference System for Real-Time Video Analytics. IEEE Internet of Things Journal, 8(18), 14274–14286. https://doi.org/10.1109/JIOT.2021.3057780
6. Han, S., Mao, H., & Dally, W. J. (2016). Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. arXiv:1510.00149. https://arxiv.org/abs/1510.00149
7. NVIDIA Corporation. (2023). NVIDIA TensorRT Developer Guide. https://docs.nvidia.com/deeplearning/tensorrt/
8. Sze, V., Chen, Y.-H., Yang, T.-J., & Emer, J. S. (2020). Efficient Processing of Deep Neural Networks. Morgan & Claypool. https://doi.org/10.2200/S00979ED1V01Y202006AIM046
9. Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., et al. (2021). Advances and Open Problems in Federated Learning. Foundations and Trends® in Machine Learning, 14(1–2), 1–210. https://doi.org/10.1561/2200000083