УДК 519.65
DOI: 10.32626/2308-5916.2022-23.83-90

Д. М. Миронюк[*]**,** аспірант,
Б. Я. Благітко[*]**,** канд. техн. наук,
І. М. Заячук[**]**,** канд. техн. наук

[*]Львівський національний університет імені Івана Франка, м. Львів,
[**]Інститут прикладних проблем механіки і математики
імені Я. С. Підстригача НАН України, м. Львів

ВИЯВЛЕННЯ ОБ'ЄКТІВ В ПРОЦЕСІ РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ ЗА ДОПОМОГОЮ ТРАНСФОРМЕРІВ

Проаналізовано сучасні методи виявлення об'єктів в процесі розпізнавання зображень за допомогою технології трансформера.

Означено переваги та недоліки різних методів. Створено власну мережу на основі трансформера DETR від FAIR team, проаналізовано його роботу. Зроблено порівняння продуктивності трансформерних мереж із оптимізованими архітектурами згорткових нейронних мереж.

В процесі досліджень були використані засоби хмарних обчислень, графічні процесори, кластери Інтернету речей або вбудовані мікропроцесорні системи.

Для забезпечення високої точності детектора об'єктів та результатів виявлення в реальному часі на різних типах пристроїв розроблено ефективний детектор об'єктів і необхідна техніка для масштабування моделі.

Поетапно проілюстровано процес навчання моделі трансформера.

Ключові слова: *математичне моделювання, комп'ютерний зір, розпізнавання зображень, трансформер.*

Вступ. Техніка виявлення об'єктів на основі глибинного навчання має багато застосувань у нашому повсякденному житті [4]. Наприклад, аналіз медичних зображень, безпілотні транспортні засоби та ідентифікація обличчя покладаються на виявлення об'єктів. Обчислювальні засоби, необхідні для вищезазначених процедур, можуть бути засоби хмарних обчислень, графічні процесори, кластери Інтернету речей або вбудовані мікропроцесорні системи. Щоб розробити ефективний детектор об'єктів, необхідна техніка масштабування моделі, оскільки вона може забезпечити високу точність детектора об'єктів та забезпечити результат виявлення в реальному часі на різних типах пристроїв.

1. Огляд сучасних підходів для задачі виявлення об'єктів на зображенні. Однією із основних задач комп'ютерного зору є пошук та виділення об'єктів. Останні декілька років у цій задачі домінували під-

ходи, які базуються на основі двоетапного методу: спочатку відбувається виділення основних властивостей зображення, що є основою для подальшої роботи. Другим етапом є регресійний, який у свою чергу відповідає за класифікацію виділених властивостей зображення та виділення відповідної зони. Цей тип розпізнавальних рішень швидко витіснив інші типи з ринку та і досі лишається на передових позиціях у багатьох академічних змаганнях між розробниками. Флагманами розвитку цих підходів останнім часом виступали команди зі штучного інтелекту передових світових IT-гігантів (Microsoft, Facebook, Google та ін). Перевагами такого типу архітектур є їх точність на об'єктах різних розмірів, достатня стійкість системи та хороша швидкість, яка, однак, є недостатньою для сучасних систем детектування. Особливо цей недолік проявляється під час використання у системах на основі мобільних та менш потужних пристроїв. Такі архітектури нейронних мереж не дають змогу реалізувати знаходження об'єктів з достатньою швидкістю, що впливає на загальну швидкість алгоритмів. Приклади таких архітектур наведено нижче:

1. Fast-RCNN.
2. Faster- RCNN [3].

Архітектура дворівневої мережі Faster-RCNN [3] зображена на рис. 1.

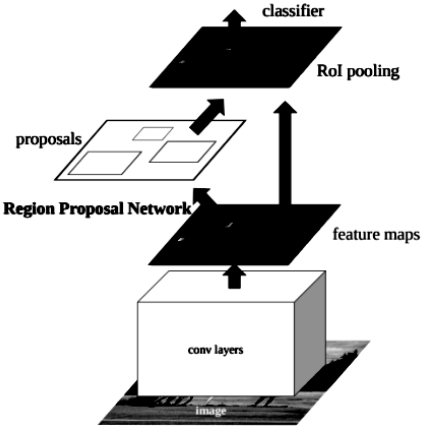

Рис. 1. *Архітектура дворівневої мережі Faster-RCNN [3]*

Наступним витком розвитку систем детектування можна вважати одноетапні системи на основі алгоритму детектування YOLO [2], які останніми роками стали швидким та досить точним рішенням для роботи на менш потужних платформах. На відміну від двоетапних систем, цей алгоритм використовує зонування зображень за один прохід (шля-

хом поділу на квадрати та роботою з ними окремо). Визначення об'єкта на зону проводиться оцінкою відстані від центра кожного квадрата до зони об'єкта (якоря що накладається на ціле зображення). Таким чином проводиться вибірка та оцінка найкращих зон для знаходжень об'єктів. Станом на 2022 рік така архітектура пережила 7 поколінь покращень і є однією із основних для детектування об'єктів на мобільних платформах. Архітектура різних варіацій YOLOv4 [2] зображена на рис. 2.

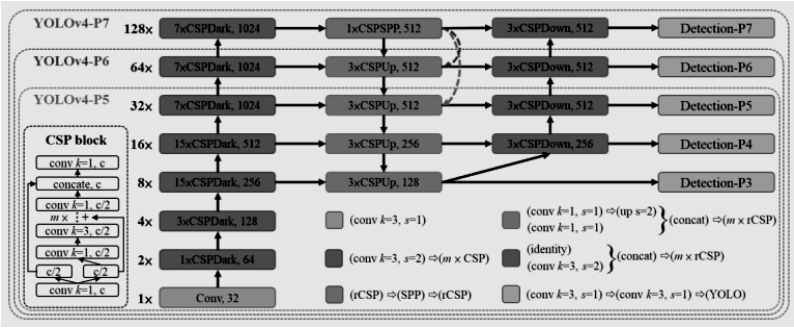

Рис. 2. *Архітектура різних варіацій YOLOv4 — large [2]*

Наступним логічним кроком була спроба використання архітектур-трансформерів, які дуже добре себе проявили як основний інструмент сучасної обробки тексту. Один із таких підходів — часткові трансформери, які у своїй суті є дворівневими мережами, де на першому рівні виокремлюються необхідні властивості зображень, які на наступному етапі розбиваються на зони та кодуються за допомогою мережі-кодувача. Для визначення об'єкта кодоване зображення пропускається через мережу — розкодовувач для виконання зворотної дії. Прикладом такої мережі є DETR (DEtection TRansformer) від Facebook AI research [1], яка використана для дослідження у цій статті. Принцип роботи мережі — трансформера DETR для розпізнавання об'єктів відображений на рис. 3.

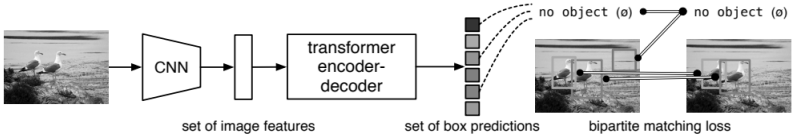

Рис. 3. *Принцип роботи мережі — трансформера DETR для розпізнавання об'єктів [1, 5]*

Підхід DETR можна поділити на такі блоки:

1. Виокремлення карт властивостей (Backbone). Початковим етапом обробки є звична для багатьох схем згорткова нейронна мережа, що перетворює вхідне зображення на карти властивостей з набагато меншою кількістю інформації.

2. Кодувач мережі — трансформера. На початковому етапі відбувається перемноження зі згорткою 1x1 для зменшення кольорової розмірності мап та подальше згортання до одноканальної схеми. Це потрібно для синхронізації входу з кодувачем, який очікує послідовність як вхідні дані. Кожен рівень кодера має стандартну архітектуру і складається з модуля самоконтролю з кількома головками та мережі прямого зв'язку (FFN). Оскільки архітектура трансформера є інваріантною до перестановок, то вона доповнюється її фіксованими позиційними кодуваннями, які додаються до вхідних даних кожного рівня уваги.
3. Декодувач мережі — трансформера. Декодер дотримується стандартної архітектури трансформатора, перетворюючи N вбудовувань розміру d за допомогою багатоголових механізмів звернення уваги сам і кодер-декодер. Модель декодує N об'єктів паралельно на кожному рівні декодера.
4. Повнозв'язна мережа для прогнозу класу та розміщення. Остаточний прогноз обчислюється за допомогою 3-шарового персептрона з функцією активації ReLU та прихованим розміром d, а також шару лінійної проекції. Повнозв'язна мережа прогнозує нормалізовані координати центру, висоту та ширину коробки вхідного зображення.
5. Лінійний шар прогнозує мітку класу за допомогою функції softmax. Оскільки ми передбачаємо набір фіксованого розміру з N обмежувальних прямокутників, де N зазвичай набагато більше, ніж фактична кількість цікавих об'єктів на зображенні, використовується додаткова спеціальна мітка класу ∅, щоб показати, що жоден об'єкт не виявлено. У класичних згорткових архітектурах такий клас позначається як «фон». Схема роботи мережі-трансформера DETR для розпізнавання об'єктів відображена на рис. 4.

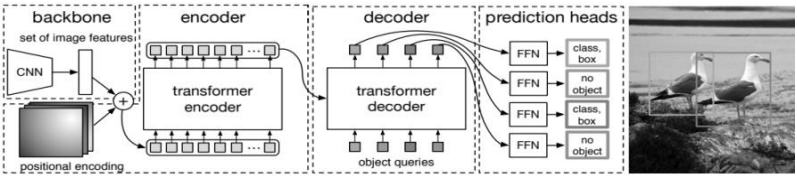

Рис. 4. *Детальна схема роботи мережі-трансформера DETR для розпізнавання об'єктів [5]*

2. Навчання моделі трансформера. ***Дані для навчання та параметри процесу.*** Для навчання моделі нейронної мережі, з метою використання у якості основного алгоритму розпізнавання робота маніпулятора, було використано власний набір даних, що містить 300 зображень 4 категорій, які були натреновані для розпізнавання об'єкта. Типи об'єктів набору даних приведені на рис. 5.

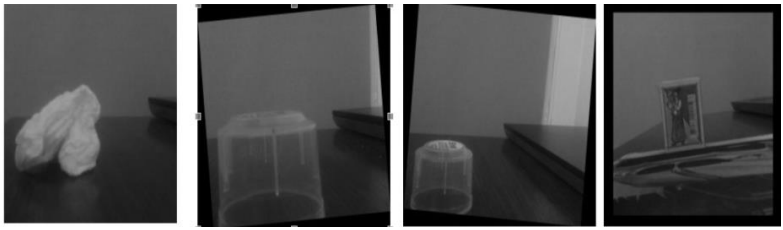

Рис. 5. *Типи об'єктів набору даних*

Загальні характеристики набору наведено у таблиці 1.

Таблиця 1

Характеристики навчального набору даних

№	Характеристика	Значення
1	Кількість унікальних зображень	300 + 30 (тренувальний та валідаційний набір даних)
2	Кількість класів	4
3	Аугментація	так (повороти, афінні перетворення, масштабування)
4	Розмір	224x224 RGB

Параметри навчання наведено у таблиці 2.

Таблиця 2

Параметри навчання

Backbone	ResNet50
Кількість параметрів	42M
Швидкість навчання	1e-4
Швидкість навчання BackBone	1e-5
Weight Decay	1e-4
Стратегія навчання	Transfer Learning
Стратегія зупинки навчання	Ліміт кроків
Loss function	bipartie matching
Оптимізатор	AdamW

Апаратні ресурси для навчання. Для навчання використовувались стандартні попередньо натреновані моделі на наборі даних ImageNet із пакету стандартних моделей, що поставляються командою FAIR разом із сирцевим кодом підходу DETR для програмного каркасу Pytorch. Для тестування також було використано фотографії валідаційного набору даних.

Як апаратну платформу було використано ноутбук на основі процесора Intel Core i5 сьомого покоління. Всі характеристика платформи наведено у таблиці 3.

Також було використано одноплатний комп'ютер спеціального призначення NVidia Jetson Nano, як платформу для тестування. Вибір платформи обґрунтовується можливим подальшим використанням такого типу моделі у якості основного алгоритму для роботи математичної моделі робота-маніпулятора. Характеристики платформи для тестування та камери розширення наведено у таблицях 4 та 5.

Таблиця 3

Характеристики апаратної платформи для навчання

Параметр	Значення
Процесор	Intel Core i5 7300 HQ 2.5-3.4 GHz
Оперативна пам'ять	8 GB
Пристрій зберігання	SSHD TOSHIBA MQ02ABD1 1 Tb
Графічний процесор	Nvidia GTX 1050 4 GB
CUDA	v. 11.8.89
Версія драйвера	v.520.56.06
Операційна система	Ubuntu 22.04
Кількість ядер CUDA	640

Програмні ресурси для навчання. Для виконання навчання була використана багатоцільова платформа для машинного навчання Anaconda на основі мови Python. Як основний програмний каркас було використано PyTorch. Перевагами такого підходу є:

- Динамічний обчислювальний граф.
- Широкий набір попередньо натренованих моделей.
- Типи даних, оптимізовані для GPU.
- Підтримка потужних інструментів візуалізації.

Для підготовки графіків навчання було використано інструмент TensorBoard. Для прискорення навчання та розпізнавання було також використано програмно-апаратну платформу CUDA, яка оптимізує обчислення шляхом перенесення однотипних операцій на графічну платформу. У цьому плані було уніфіковано підходи до навчання та розпізнавання, оскільки графічні процесори з підтримкою CUDA наявні як на платформі для навчання, так і на мобільній платформі для розпізнавання.

Nvidia CUDA — це програмно-апаратна платформа для виконання обчислень на графічних процесорах, яка була розроблена та підтримується компанією Nvidia. Містить у собі широкий набір інструментів для виконання різноманітних загальних обчислювальних задач, а також пакети для розпаралелювання та вирішення конкретних задач. У програмному вигляді вона представлена у вигляді розширення мови програмування C. Для трансляції коду з цього розширення використовується власний компілятор nvcc, який був створений на основі відкритого компілятора Open64.

Таблиця 4

Характеристики платформи Nvidia Jetson Nano

Процесор	Quad-core Cortex-A53 64-bit SoC @ 1.2GHz
Оперативна пам'ять. Пристрій зберігання	2GB LPDDR4 1600 MHz SDRAM Kingston MicroSDHC 32Gb Class 10 Canvas Select
Графічний процесор	Nvidia Maxwell architecture with 128 NVidia CUDA Cores
CUDA	11.8
Операційна система	Ubuntu Tegra OS, based on Ubuntu 18.04
Версії Tensorflow, PyTorch	Ver. 2.3.1 (TF), ver 1.8 (PyTorch)
Python	Ver. 3.8.2 Anaconda x64

Таблиця 5

Характеристики камери розширення Raspberry Pi camera ver. 1.3

Параметр	**Значення**
Роздільна здатність зображення	5MP Max 2592 x 1944
Інтерфейс підключення	Ribbon Cable
Розмір пікселя	1.4 x 1.4 μm
Лінза	f=3.6 mm, f/2.9
Кут огляду	54° x 41°
Максимальна роздільна здатність відео	1080p @ 30fps
Максимальна кількість кадрів за секунду	480p @ 90fps
Параметри вибору роздільної здатності відео	1080p @ 30fps, 720p @ 60fps, 480p @ 90fps
Розмір сенсора	3.67mm x 2.74mm (1/4" format)
Розміри модуля камери	25mm x 24mm (9mm thickness)

Основні характеристики CUDA:

- Основне уніфіковане рішення для виконання загальних задач з використанням графічних процесорів Nvidia.
- Великий набір рішень, що підтримуються.
- Великий набір стандартних бібліотек для чисельного аналізу (включно з BLAS та FFT).
- Оптимізована для ефективного обміну даних між центральним та графічним процесором.
- Взаємодія з графічним API OpenGL та DirectX.
- Можливість низькорівневої розробки.
- Підтримка широкого набору операційних систем.
- Висока документованість і широкий набір прикладів коду для початківців.

Результати навчання. В результаті роботи реалізованої мережі — трансформера було виявлено 59% об'єктів на плоских зображеннях у вигляді фотографій 250*250 пікселів.

Висновок: Результати тестування навчених моделей, які не були оптимізовані для роботи із спеціалізованою платформою Nvidia Jetson Nano є задовільними, хоч і далекими від розпізнавання у реальному часі. Використання оптимізованих моделей за допомогою бібліотеки TensorRT від Nvidia дає змогу підвищити продуктивність обчислень на мобільних платформах цього типу.

Список використаних джерел:

1. Carion N., Massa F., Synnaeve G., Usunier N., Kirillov A., Zagoruyko S. End-to-End Object Detection with Transformers. arXiv: 2005.12872v3 [cs.CV] 28 May 2020. P. 1-26. URL: https://arXiv.org/pdf/2005.12872.pdf.
2. Chien-Yao Wang, Alexey Bochkovskiy, Hong-Yuan Mark Liao. Scaled-YOLOv4: Scaling Cross Stage Partial Network. arXiv: 2011.08036v2 [cs.CV] 22 Feb 2021. P. 1-10. URL: https://arXiv.org/pdf/2011.08036.pdf.
3. Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. arXiv: 2016.1506.01497v3 [cs.CV] 6 Jan 2016. P. 1-14. URL: https://arXiv.org/pdf/1506.01497.pdf.
4. Goodfellow I., Bengio Yo., Courville A. Deep Learning. A MIT Press Book. *MIT Press.* 2016. 716 p.
5. DETR: End-to-End Object Detection with Transformers. URL: https://github.com/facebookresearch/detr

OBJECT DETECTION IN THE IMAGE RECOGNITION PROCESS USING TRANSFORMERS

Modern object detection methods in the image recognition process using transformer technology are analyzed.

The various methods advantages and disadvantages are identified. An own network was created based on the DETR transformer from the FAIR team, and its operation was analyzed. A comparison of the transformer networks performance with optimized architectures of convolutional neural networks is made.

The cloud computing tools, graphics processors, Internet of Things clusters or embedded microprocessor systems were used in the research process.

To ensure high object detector accuracy and real-time detection results on different types of devices, an efficient object detector and model scaling technique are required.

The transformer model learning is illustrated step-by-step process.

Key words: *mathematical modeling, computer vision, image recognition, transformer.*

Отримано: 24.10.2022