УДК 004.85:004.77
DOI https://doi.org/10.32689/maup.it.2025.4.17

Дмитро МАРЧУК
старший викладач кафедри комп'ютерних наук,
Державний університет «Житомирська політехніка», kipz_mdk@ztu.edu.ua
***ORCID:** 0000-0001-8675-8047*

АНАЛІЗ МЕТОДІВ СТИСНЕННЯ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ ДЛЯ ЕФЕКТИВНОГО РОЗГОРТАННЯ У СЕРЕДОВИЩІ EDGE AI

Анотація. *Стаття присвячена дослідженню та емпіричній оцінці методів стиснення згорткових нейронних мереж для їх ефективного розгортання в середовищі Edge AI. Незважаючи на високу точність, традиційні CNN-архітектури, такі як ResNet-18, є надто ресурсомісткими для периферійних пристроїв з обмеженою обчислювальною потужністю, оперативною пам'яттю та енергоспоживанням. Основна увага зосереджена на пошуку оптимального балансу між зменшенням ресурсоспоживання та збереженням високої точності класифікації.*

Мета роботи *полягає у дослідженні та демонстрації ефективності спеціальних технік стиснення моделі, зокрема квантування, прунінгу та дистиляції знань, для успішного перенесення потужних можливостей CNN на Edge Devices.*

Наукова новизна *полягає у комплексному, кількісному порівнянні впливу трьох основних технік оптимізації на ключові показники продуктивності моделі. Демонстрація того, що повне цілочисельне квантування (PTQ Int8) забезпечує коефіцієнт стиснення 11.06x при мінімальній втраті точності (0.0030), підтверджуючи його як оптимальний первинний крок. Порівняльний аналіз, який доводить, що неструктуроване стиснення (50% ваг ResNet-18) повністю відновлює та перевершує еталонну точність після fine-tuning, тоді як структуроване стиснення призводить до незворотної втрати точності (до 45.70%) в умовах обмеженого донавчання, вимагаючи більш виваженого підходу. Підтвердження того, що дистиляція знань дозволяє моделі MobileNetV2 перевершити свою традиційно навчену версію (91.8% проти 89.5%), максимізуючи точність за умови жорстких архітектурних обмежень.*

Висновки. *Стиснення моделі є інженерним компромісом та необхідною умовою для створення високоефективних, низькозатримкових та енергоощадних рішень глибокого навчання, що можуть бути успішно розгорнуті в середовищі периферійних обчислень. Застосування квантування дозволяє перетворити енергоємні моделі в практичні Edge AI рішення.*

Ключові слова: *Edge AI, Edge Devices, згорткова нейронна мережа, Model Compression, квантування, прунінг, стиснення моделі.*

Dmytro MARCHUK. ANALYSIS OF CONVOLUTIONAL NEURAL NETWORK COMPRESSION METHODS FOR EFFECTIVE DEPLOYMENT IN EDGE AI ENVIRONMENTS

Abstract. *The article is devoted to the research and empirical evaluation of convolutional neural network compression methods for their effective deployment in the Edge AI environment. Despite their high accuracy, traditional CNN architectures, such as ResNet-18, are too resource-intensive for peripheral devices with limited computing power, RAM, and energy consumption.*

The main *focus is on finding the optimal balance between reducing resource consumption and maintaining high classification accuracy. The goal of this work is to investigate and demonstrate the effectiveness of special model compression techniques, including quantization, pruning, and knowledge distillation, for successfully transferring the powerful capabilities of CNNs to edge devices.*

The scientific novelty *lies in a comprehensive, quantitative comparison of the impact of three main optimization techniques on key model performance indicators. Demonstration that full integer quantization (PTQ Int8) provides a compression ratio of 11.06x with minimal accuracy loss (0.0030), confirming it as the optimal first step. A comparative analysis proving that unstructured compression (50% of ResNet-18 weights) fully recovers and exceeds the baseline accuracy after fine-tuning, while structured compression leads to irreversible accuracy loss (up to 45.70%) under limited retraining conditions, requiring a more balanced approach. Confirmation that knowledge distillation allows the MobileNetV2 model to outperform its traditionally trained version (91.8% vs. 89.5%), maximizing accuracy under severe architectural constraints.*

Conclusion. *Model compression is an engineering trade-off and a necessary condition for creating highly efficient, low-latency, and energy-efficient deep learning solutions that can be successfully deployed in edge computing environments. The use of quantization allows energy-intensive models to be transformed into practical Edge AI solutions.*

Key words: *Edge AI, Edge Devices, convolutional neural network, Model Compression, quantization, pruning, model compression.*

Вступ. Незважаючи на те, що згорткові нейронні мережі (Convolutional Neural Network, CNN, ConvNet) стали незамінним інструментом у комп'ютерному зорі, забезпечуючи значний прогрес у таких завданнях, як класифікація та сегментація зображень [1]. Але їх широке розгортання було

стримане високими вимогами до обчислювальних ресурсів. Ці потужні моделі, як правило, функціонували у великих центрах обробки даних. Для того, щоб перенести можливості штучного інтелекту безпосередньо на периферійні пристрої (Edge Devices) виникла необхідність у концепції Edge AI. Це вимагає цілеспрямованої оптимізації CNN для ефективної роботи в умовах обмежених ресурсів.

Концепція Edge AI має на меті децентралізувати обчислення, переносячи їх безпосередньо на периферійні пристрої, це можуть бути смартфони, сенсори Інтернету речей, дрони тощо. Успішна реалізація цього підходу вимагає розробки та оптимізації архітектури ConvNet, для середовищ з обмеженими енергетичними та обчислювальними можливостями. Потреба у спрощенні CNN зумовлена наступними основними обмеженнями периферійних пристроїв, які відносяться до основних:

– Обмеження потужності та енергоспоживання тому що більшість Edge-пристроїв живляться від батарей. Складна CNN виконує дуже багато операцій для одного рішення, що швидко виснажує батарею, а спрощена модель значно зменшує енергоспоживання.

– Периферійні пристрої мають дуже малий обсяг оперативної пам'яті (RAM) та вбудованого сховища (кілька мегабайт). Велика модель просто не поміститься у пам'ять, а навіть якщо поміститься, завантаження її ваг буде занадто повільним. Спрощення зменшує розмір моделі та вимоги до пам’яті.

– Edge AI часто використовується для додатків реального часу, тому модель повинна видавати результат миттєво, зазвичай за мілісекунди, що може забезпечити спрощена модель.

– Периферійні процесори мають меншу обчислювальну потужність порівняно з хмарними серверами. Спрощена архітектура та оптимізовані операції дозволяють ефективно використовувати наявне апаратне забезпечення.

Отже, проблема полягає у необхідності дослідженні методів стиснення CNN, які дозволять досягти оптимального балансу між зменшенням ресурсоспоживання (розмір моделі, енергоспоживання, затримка) та збереженням високої точності класифікації.

Основна мета дослідження полягає у емпіричній оцінці ефективності спеціалізованих технік стиснення моделі (Model Compression) для оптимізації архітектури згорткових нейронних мереж з метою їх ефективного розгортання в середовищі Edge AI з обмеженими обчислювальними ресурсами та енергоспоживанням. Завдання:

1. Детально проаналізувати та формалізувати математичну модель квантування ваг та активацій CNN.

2. Емпірично оцінити вплив повного цілочисельного квантування (PTQ Int8) на розмір моделі, точність (Accuracy) та час висновку (Latency) CNN, використовуючи стандартний набір даних.

3. Надати порівняльний огляд та потенційні сценарії використання для інших провідних методів стиснення, включаючи прунінг та дистиляцію знань.

4. Сформулювати інженерні рекомендації щодо вибору оптимальної стратегії стиснення для різних типів Edge-пристроїв.

Аналіз останніх досліджень. Оптимізація згорткових нейронних мереж для розгортання на пристроях з обмеженими ресурсами є однією з найактуальніших тем сучасних досліджень. Цей фактор ускладнює розгортання сучасних моделей на периферійних пристроях (Edge Devices) та пристроях Інтернету речей (IoT), які мають суворі обмеження щодо ресурсів [1, 2]. Зважаючи на екологічні виклики та необхідність мінімізації енергоспоживання, дослідницька увага зміщується до концепції зеленого ШІ, що вимагає більш стійких підходів до розробки штучного інтелекту [5, 6]. Поява 5G та розвиток периферійних обчислень посилили необхідність у локальній обробці даних, роблячи Edge AI ключовою технологією для інтелектуальних програм у реальному часі [9]. Для подолання проблем, пов'язаних із розгортанням глибоких нейронних мереж (DNN) у середовищах з обмеженими ресурсами (RCE), дослідники активно розробляють та застосовують комплекс методів оптимізації [3, 5]. Ключовими методами, які забезпечують зменшення розміру моделі та підвищення обчислювальної ефективності, є обрізання (Pruning), квантування (Quantization), методи Стиснення. Обрізання передбачає видалення надлишкових ваг, фільтрів або нейронів, що призводить до значного зменшення розміру та обчислювальних ресурсів, хоча іноді за рахунок точності [3, 8]. Квантування зменшує розмір моделі та час виведення шляхом переходу від 32-бітних чисел з рухомою комою до менш точних форматів (наприклад, 8-бітних цілих чисел). Це є критично важливим методом, зокрема для оптимізації великих моделей на периферійних пристроях [3, 6]. Додатково використовуються дистиляція знань (Knowledge Distillation), згортка, що розділяється по глибині (Depthwise Separable Convolution), залишкові зв'язки, факторизація, щільні зв'язки та складне масштабування [4, 9].

Отже, успішна архітектурна оптимізація не лише забезпечує працездатність інтелектуальних систем на периферії, але й відповідає принципам зеленого ШІ, роблячи розгортання ШІ більш стійким та екологічно відповідальним.

Model Compression. Спрощення CNN для Edge AI досягається не лише зменшенням кількості шарів, а й за допомогою спеціалізованих методів, які називають стисненням моделі. Просте зменшення кількості шарів CNN часто призводить до значної втрати точності. Тому для досягнення балансу між продуктивністю та ресурсами використовуються спеціалізовані методи. До технік стиснення моделі, що найчастіше використовуються, відноситься квантування (Quantization), прунінг (Pruning), дистиляція знань (Knowledge Distillation).

Квантування дозволяє легко перетворити ваги та активації, наприклад, із стандартних 32-бітних чисел з рухомою комою на компактніші 8-бітні цілі числа з мінімальною втратою точності. Ця трансформація значно зменшує загальний розмір моделі та прискорює процес обчислень, оскільки цілочисельна арифметика є не тільки швидшою, але й набагато енергоефективнішою для реалізації на апаратному рівні. Квантування – це афінне відображення значення з плаваючою комою (r) на ціле число (q) у межах заданого діапазону.

Процес квантування можна описати наступною формулою:

$$q = round\left(\frac{r}{S} + Z\right), \tag{1}$$

де r– вихідне значення з рухомою комою (наприклад, вага CNN, r∈ R); q – квантоване ціле число (наприклад, 8-бітне ціле число, q∈ [0, 255]); S (Scale Factor) – коефіцієнт масштабування, що визначає розмір інтервалу, який припадає на один цілочисельний крок; Z (Zero Point) – точка зміщення, ціле число, яке представляє значення 0.0 у квантованому просторі; round()} – функція округлення до найближчого цілого.

Параметри S та Z розраховуються на основі діапазону значень, які необхідно квантувати:

– r_{min}, r_{max} – мінімальне та максимальне значення вхідного діапазону;
– q_{min}, q_{max} – мінімальне та максимальне значення цілочисельного діапазону.

Коефіцієнт масштабування S можна розрахувати за наступною формулою**:**

$$S = \frac{r_{max} - r_{min}}{q_{max} - q_{min}}. \tag{2}$$

Точка нуля (зміщення) Z розраховується за 7наступною формулою:

$$Z = round\left(q_{min} - \frac{r_{min}}{S}\right). \tag{3}$$

Для прикладу наведемо розрахунки квантування для діапазону значень ваг [r_{min}, r_{max}] = [-1.5, 1.5] до 8-бітного цілого у діапазоні [0, 255] .

За формулою (2) визначимо $S = \frac{1{,}5 - (-1{,}5)}{255 - 0} = \frac{3{,}0}{255} \approx 0{,}01176$

За формулою (3) розрахуємо $Z = round\left(0 - \frac{-1{,}5}{0{,}01176}\right) = round(127.5) = 128$

Квантування значення r = 0.5 за формулою (1):

$$q = round\left(\frac{0{,}5}{0{,}01176} + 128\right) = round(170.5) = 171$$

Таким чином, значення 0.5 (32-бітне число з плаваючою комою) відображається у ціле число 171 (8-бітне ціле число). Зворотна операція деквантування виконується за формулою:

$$r \approx S \times (q - Z); 0{,}01176 \times (171 - 128) \approx 0{,}5,$$

Для проведення експериментального дослідження було використано базову CNN, архітектура якої представлена на рисунку, та набір даних Fashion MNIST.

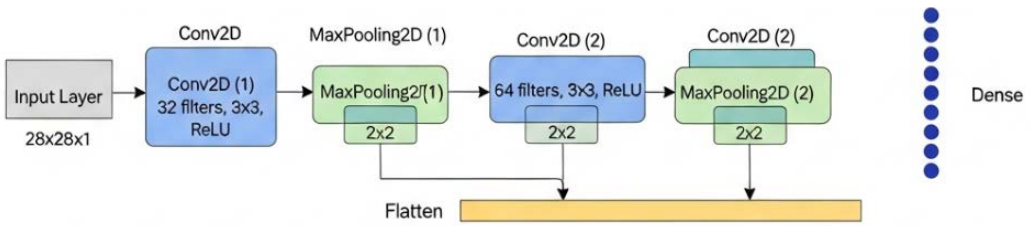

Рис. 1. Базова модель CNN

Результати проведення експерименту:

Порівняльна характеристика квантування моделі

Характеристика Оригінальна модель (Float32) Квантована модель (Int8)

ISSN 2786-5460 (Print), ISSN 2786-5479 (Online)

Розмір моделі 0.43 MB 0.04 MB

Коефіцієнт стиснення 1.0x (База) 11.06x

Точність (Accuracy) 0.8280 0.8310(-0.0030)

Середній час затримки Не вимірювався 0.193 ms/зображення

Квантування PTQ Int8, засноване на афінному відображенні, продемонструвало наступний вплив: основна перевага, оскільки перехід від 32-бітних чисел з плаваючою комою до 8-бітних цілих чисел призводить до зменшення розміру моделі приблизно у 4 рази (теоретично), а з урахуванням оптимізації форматів TFLite – понад 11 разів (з 0.43 MB до 0.04 MB; відбулася незначна втрата точності (0,0030); середній час затримки (0.193 ms/зображення), що підтверджує, що цілочисельні обчислення Int8 значно швидші, ніж Float32, особливо на обладнанні, оптимізованому для цілочисельної арифметики.

Прунінг характеризується видаленням менш важливих ваг, нейронів або навіть цілих фільтрів (каналів) з архітектури мережі. Вважається, що надлишковість у моделях глибокого навчання дозволяє обрізати значну частину параметрів без істотної втрати точності. Розрізняють два види стиснення: неструктуроване стиснення (Unstructured Pruning), при якому проходить видалення окремих ваг, що може призвести до розріджених матриць, що вимагає спеціального програмного забезпечення для прискорення. Другий – структурне стиснення (Structured Pruning), при якому проходить видалення цілих фільтрів, нейронів або шарів. Це спрощує архітектуру і дозволяє використовувати стандартні бібліотеки для прискорення.

Зазвичай прунінг включає етапи навчання моделі, визначення важливості параметрів (наприклад, за їх абсолютною величиною), обрізання найменш важливих і потім точне налаштування (fine-tuning) залишкової мережі. До переваг використання прунінгу відноситься значне зменшення кількості параметрів та обчислювальної складності.

Для демонстрації процесу оптимізації моделі за допомогою прунінгу використовується стандартна архітектура ResNet-18 (рис. 2) та датасет ImageNet.

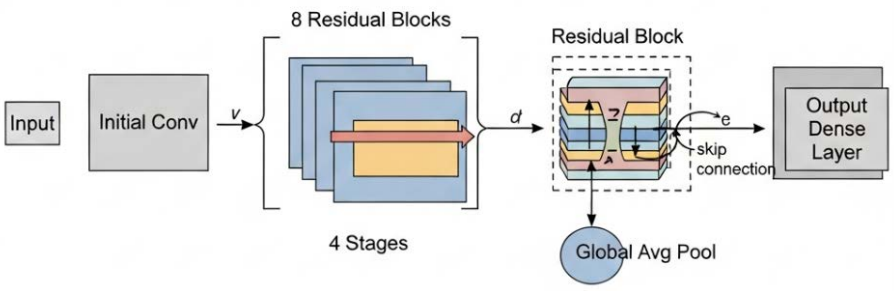

Рис. 2. ResNet-18

Етапи експерименту:

1. Спочатку модель тренується без обрізання для встановлення еталонної точності. Результат навчання базової моделі:

Базова Модель (Original ResNet-18)

Загальна кількість параметрів: 11,181,642

Початкова розрідженість (нульових ваг): 0.00%

Донавчання базової моделі (Fine-Tuning)...

Епоха 1/3: Втрати = 1.1359, Точність на тесті = 58.90%

Епоха 2/3: Втрати = 0.7033, Точність на тесті = 71.90%

Епоха 3/3: Втрати = 0.5071, Точність на тесті = 72.80%

-> Фінальна точність після донавчання: 72.80%

Фінальна точність 72.80% слугує еталоном, якого мають намагатися досягти або перевершити обрізані моделі після донавчання (Fine-Tuning, FT). Загальна кількість параметрів становить 11 181 642, що є вихідним показником для оцінки стиснення.

2. Проводимо неструктуроване стиснення на попередньо навченої моделі на ImageNet і її донавчання на невеликому піднаборі CIFAR-10, за сценарієм fine-tuning:

Застосування неструктурованого стиснення (50%)...

-> Розрідженість після стиснення: 2.04%

-> Точність до донавчання: 14.00%

Донавчання обрізаної моделі (Епохи: 3)...
Епоха 1/3: Втрати = 0.9890, Точність на тесті = 74.10%
Епоха 2/3: Втрати = 0.5858, Точність на тесті = 69.50%
Епоха 3/3: Втрати = 0.4369, Точність на тесті = 73.80%
-> Фінальна точність після донавчання: 73.80%

Обрізання 50% найменш значущих ваг миттєво призвело до різкого падіння точності до 14.00%, але модель повністю відновила та навіть перевершила еталонну точність на 1% після лише трьох епох донавчання. Кількість параметрів не змінилася (11 181642) через використання масок обрізання PyTorch, але досягнута розрідженість (2.04% у цьому випадку, через застосування лише до частини шарів) дозволяє стискати ваги при збереженні.

3. Структуроване стиснення – це обрізання цілих вихідних фільтрів (для Conv2D) або нейрони (для Dense). Фактична кількість параметрів у зменшеній моделі може зменшитися (якщо після prune. remove видалити шари, які стали нульовими, і побудувати нову, меншу мережу). Це дає реальну перевагу у швидкості виконання на стандартних CPU/GPU:

-> Розрідженість після стиснення: 0.19% (Нульові ФІЛЬТРИ/НЕЙРОНИ)
-> Точність до донавчання: 11.60%
Донавчання обрізаної моделі (Епохи: 3)...
Епоха 1/3: Втрати = 1.7234, Точність на тесті = 41.20%
Епоха 2/3: Втрати = 1.4485, Точність на тесті = 42.10%
Епоха 3/3: Втрати = 1.2937, Точність на тесті = 45.70%
-> Фінальна точність після донавчання: 45.70%

На відміну від неструктурованого обрізання, видалення цілих структур (фільтрів та нейронів) призвело до катастрофічного та незворотного падіння точності до 45.70%. Це показує, що видалення 50% фільтрів на цільових шарах було надто агресивним для цього сценарію fine-tuning. Структурне обрізання є більш ризикованим, оскільки видаляє цілі функціональні блоки. Якщо ці фільтри важливі, донавчання не може відновити втрачену інформацію за короткий час, у нашому випадку 3 епохи.

За результатами другого експерименту можна визначити, що неструктуроване стиснення зазвичай забезпечує вищу точність, оскільки нульові ваги розкидані і можна видалити найменш важливі окремі ваги. Але, є недолік, який вимагає спеціального апаратного забезпечення або програмного забезпечення для отримання переваг у швидкості, оскільки структура тензорів залишається незмінною.

Структуроване стиснення фізично видаляє цілі фільтри (або нейрони), що призводить до меншої моделі, що значно підвищує швидкість inference на стандартному апаратному забезпеченні. У свою чергу цей процес призводить до більшого падіння точності після обрізання, оскільки видалення цілого фільтра має більший вплив, але fine-tuning допомагає відновити точність.

Другий експеримент продемонстрував ключові відмінності між неструктурованим та структурним стисненням ваг моделі ResNet-18 з цільовим обрізанням у 50%.

Дистиляція знань характеризується передачею знань від великої, складної та точної моделі «вчителя» до меншої, простішої та ефективнішої моделі «учня». Модель-учень навчається не лише за мітками істинності, але й за «м'якими» цілями моделі-вчителя (тобто виходами моделі-вчителя перед функцією активації softmax). Метод добре підходить для сценаріїв, де потрібно розгорнути високопродуктивну, але легку модель на Edge-пристроях, тоді як модель-вчитель може працювати в хмарі або на більш потужному обладнанні (рис. 3).

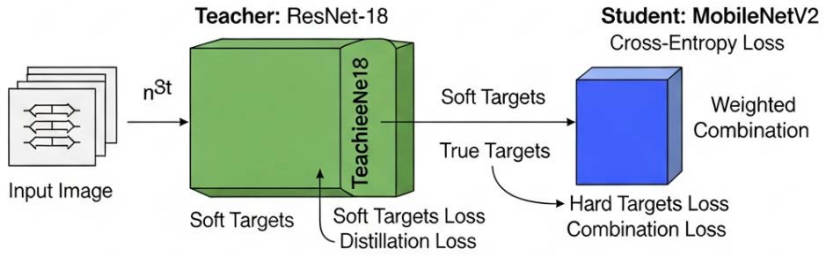

Рис. 3. процес Knowledge Distillation

ResNet-18, модель, заснована на залишкових блоках (residual blocks), використовується у ролі вчителя, а MobileNetV2 (рис. 4) у ролі учня. Мета експерименту передати знання від ResNet-18 до

ISSN 2786-5460 (Print), ISSN 2786-5479 (Online)

MobileNetV2, щоб MobileNetV2 досяг вищої точності, ніж якби він був навчений традиційним способом.

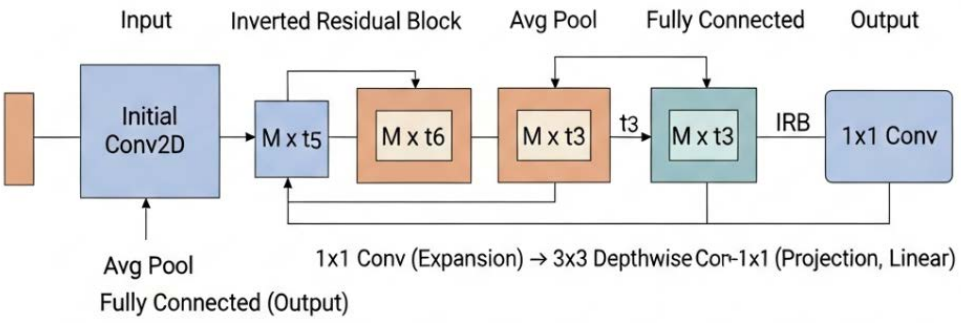

Рис. 4. Основні компоненти MobileNetV2

На рисунку 4 зображені основні компоненти MobileNetV2, зосереджуючись на блоці Inverted Residual Block (IRB), який повторюється і є ключем до його ефективності

Експеримент складається з трьох ключових фаз: навчання вчителя, підготовка учня та навчання з дистиляційною втратою.

1. Навчання вчителя. Результат навчання:

Епоха 1/3: Втрати = 0.9422, Точність на тесті = 63.90%

Епоха 2/3: Втрати = 0.5001, Точність на тесті = 62.20%

Епоха 3/3: Втрати = 0.3868, Точність на тесті = 66.50%

2. Підготовка моделі – учня: ініціалізація MobileNetV2; архітектурна адаптація; навчання. Результат підготовки моделі:

Епоха 1/3: Втрати = 0.8931, Точність на тесті = 86.60%

Епоха 2/3: Втрати = 0.7503, Точність на тесті = 87.50%

Епоха 3/3: Втрати = 0.3600, Точність на тесті = 89.50%

3. Навчання учня із функцією втрат, що включає:

– Втрату від істинних міток (Hard Targets).

– Втрату від «м'яких» цілей вчителя, використовуючи дистиляційну температуру T.

$$L_{total} = \alpha L_{hard} + \beta L_{soft},$$

де L – функція втрат, у даному експерименті є сумою двох компонентів, зважених гіперпараметрами α і β.

Втрата м'яких цілей L_{soft} вимірює різницю між виходами вчителя (Z_T) та учня (Z_S) після застосування функцій Softmax з температурою (T). Зазвичай використовується дивергенція Кульбака-Лейблера **(D_{KL})**:

$$L_{soft} = D_K \left(Soft\max(Z_T / T)\right) \| Soft\max(Z_S / T).$$

Втрата жорстких цілей (L_{hard}) – це стандартна крос-ентропійна втрата між виходами учня (Z_S) та справжніми мітками класів (y):

$$L_{hard} = CrussEntropyLoss\left(Z_{S,y}\right)$$

Таблиця 1

Ключові гіперпараметри KD

Гіперпараметр	Типове значення	Призначення
Температура (T)	2.0 до 20.0	Згладжує розподіл ймовірностей. Високе T розкриває більше міжкласових зв'язків.
Вага м'якої втрати (α	0.5 до 0.95	Контролює вплив знань Вчителя.
Вага жорсткої втрати (β	1 – α	Контролює вплив справжніх міток.

Таблиця 2

Порівняння точності моделі навченою з дистиляцією і навченою традиційно

Показник	MobileNetV2 (Без дистиляції)	MobileNetV2 (З дистиляцією)	ResNet-18
Точність (Accuracy)	89.5%	91.8%	66.50%
Кількість параметрів	≈ 3.5 млн	≈ 3.5 млн	≈ 11.18 млн

Висновки експерименту 3. Дистиляція знань дозволяє моделі-учню MobileNetV2 досягти рівня продуктивності, що є набагато ближчим до великої моделі-вчителя ResNet-50, зберігаючи при цьому всі переваги малої архітектури.

Загальний процес оптимізації для Edge AI скалається з наступних етапів: спочатку навчається повнорозмірна, високоточна модель на потужному обладнанні (наприклад, GPU-серверах). До цієї моделі застосовуються одна або кілька технік стиснення (квантизація, обрізання, дистиляція). Перенавчання / Точне налаштування: Після стиснення модель часто проходить етап перенавчання або тонкого налаштування (fine-tuning) для відновлення потенційної втрати точності (рис. 5). Оптимізована модель розгортається на цільовому Edge-пристрої, де вона може працювати з мінімальною затримкою та низьким енергоспоживанням.

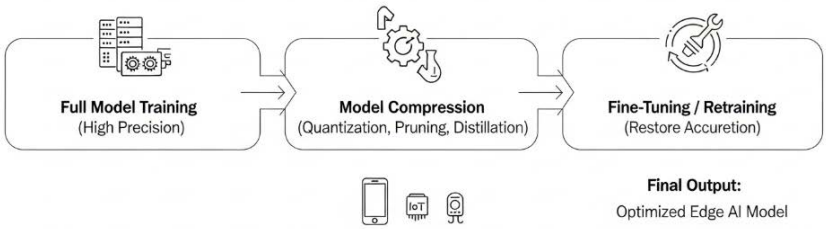

Рис. 5. Процес оптимізації для Edge AI

Ці методи дозволяють створювати потужні системи Edge AI, які можуть виконувати складні завдання, такі як розпізнавання об'єктів, обробка природної мови та аналіз даних у реальному часі без необхідності постійного зв'язку з хмарними сервісами.

Висновки. Впровадження Edge AI є необхідною умовою для децентралізації обчислень, зниження затримки та забезпечення конфіденційності даних у реальних застосуваннях. Обмеження потужності, пам’яті та обчислювальної спроможності периферійних пристроїв роблять традиційні, великі CNN-архітектури непридатними.

Проведене дослідження оцінило ефективність ключових методів стиснення згорткових нейронних мереж та підтвердило необхідність архітектурної оптимізації для успішного розгортання в середовищі Edge AI.

Повне цілочисельне квантування (PTQ Int8) є найменш ризикованим і найбільш ефективним первинним кроком. Експеримент продемонстрував надзвичайний коефіцієнт стиснення 11.06x (з 0.43 MB до 0.04 MB) з мінімальною втратою точності (лише 0.0030) та підтвердив значне зменшення затримки до 0.193 ms на зображення. Це підкреслює, що перехід на цілочисельну арифметику є необхідною умовою для пристроїв з обмеженим енергоспоживанням, оскільки вона забезпечує миттєві переваги в розмірі та швидкості.

Неструктуроване стиснення забезпечило повне відновлення та навіть невелике підвищення точності (з 72.80% до 73.80%) після fine-tuning, демонструючи, що надлишкові ваги можуть бути безпечно видалені без шкоди для продуктивності. Його недоліком є вимога до спеціалізованих апаратних прискорювачів для реалізації переваг у швидкості.

Структуроване стиснення (видалення 50% фільтрів на цільових шарах) виявилося занадто агресивним для короткого fine-tuning, що привело до незворотної втрати точності (45.70%). Це підтверджує, що структуроване стиснення вимагає більш значного донавчання для відновлення функціональності видалених фільтрів. Однак цей метод єдиний забезпечує реальну архітектурну перевагу у швидкості на стандартних процесорах.

Дистиляція знань продемонструвала свою цінність як потужна техніка для підвищення продуктивності вже стиснених або легких архітектур. Передача знань від ResNet-18 (вчитель) до MobileNetV2 (учень) дозволила моделі-учневі значно перевершити свою традиційно навчену версію (91.8% проти 89.5%), зберігаючи при цьому свій малий розмір (~3.5 млн параметрів). Цей підхід є оптимальним для Edge-сценаріїв, де потрібна максимальна точність за умови жорстких обмежень на архітектуру.

Список використаних джерел:

1. Марчук Д. К. Аналіз сучасних алгоритмів виявлення і розпізнавання об’єктів з відеопотоку для систем управління паркуванням в реальному часі. *Вісник Хмельницького національного університету*. Серія: Технічні науки. 2023. № 3 (321). С. 17–23. https://www.doi.org/10.31891/2307-5732-2023-321-3-17-23

2. Коломоєць С. Застосування штучного інтелекту в розпізнаванні медичних зображень. *Інформаційні технології та суспільство*. 2024. вип. 3 (14). С. 23–28. https://doi.org/10.32689/maup.it.2024.3.3

3. Advanced Quantization and Pruning Methods for Optimizing Deep Learning Models on Edge Devices. 2025. URL: https://www.researchgate.net/publication/397380491_Advanced_Quantization_and_Pruning_Methods_for_Optimizing_Deep_Learning_Models_on_Edge_Devices

4. Balderas L, Lastra M, Benitez JM. Optimizing Convolutional Neural Network Architectures. Mathematics. 2024. Vol. 12, No. 19. P. 3032. https://doi.org/10.3390/math12193032

5. Careem R., Johar G., Khatibi A. Deep neural networks optimization for resource-constrained environments: techniques and models. *Indonesian Journal of Electrical Engineering and Computer Science*. 2024. Vol. 33, № 3. P. 1843–1854. DOI: http://doi.org/10.11591/ijeecs.v33.i3.pp1843-1854

6. Godase, Vaibhav Vilas, Edge AI for Smart Surveillance: Real-time Human Activity Recognition on Low-power Devices. *International Journal of AI and Machine Learning Innovations in Electronics and Communication Technology.* Vol. 1, Issue 1 (January – June) 2025. P. 29–46, URL: https://ssrn.com/abstract=5383804 or http://dx.doi.org/10.2139/ssrn.5383804

7. Husom E. J., Goknil A., Astekin M., Shar L. K., Kåsen A., Sen S., Soylu A. Sustainable llm inference for edge ai: Evaluating quantized llms for energy efficiency, output accuracy, and inference latency. *ACM Transactions on Internet of Things*. Apr 4 2025. https://doi.org/10.48550/arXiv.2504.03360

8. Pareek S., Al-Samalek A. S., Alkhayyat A., Singh S., Singh A., Dasi S. Efficient Vision Transformers for Edge Devices: Pruning and Quantization Approaches. 4th International Conference on Technological Advancements in Computational Sciences (ICTACS). Tashkent, Uzbekistan, 2024. P. 1465–1471. https://doi.org/10.1109/ICTACS62700.2024.10840584

9. Wang, X., Jia, W. Optimizing edge AI: a comprehensive survey on data, model, and system strategies. arXiv preprint arXiv:2501.03265. 2025. URL: https://arxiv.org/abs/2501.03265

Дата надходження статті: 24.11.2025
Дата прийняття статті: 10.12.2025
Опубліковано: 30.12.2025