ПОЛТАВСЬКИЙ ДЕРЖАВНИЙ АГРАРНИЙ УНІВЕРСИТЕТ
Навчально-науковий інститут економіки, управління, права та інформаційних технологій
Кафедра інформаційних систем та технологій

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти бакалавр

на тему: **«Застосування акселераторів ШІ для прискорення нейронних мереж на Raspberry Pi»**

Виконав: здобувач вищої освіти
за освітньою програмою
Інформаційні управляючі системи
спеціальності 126 Інформаційні
системи та технології
ступеня вищої освіти бакалавр
групи 126ІСТ_бд_2021
Данилко Максим Геннадійович
Керівник: Слюсарь Ігор Іванович
Рецензент: Муравльов Володимир
Вячеславович

Полтава – 2025 року

ВСТУП

Актуальність теми кваліфікаційної роботи підтверджується необхідністю впровадження обчислювальних систем, призначених для ефективного виконання складних алгоритмів нейронних мереж, до яких відносяться рішення Edge AI. Критично важливими їх компонентами є акселератори штучного інтелекту (ШІ). Їх конструктивні особливості дозволяють забезпечити високий рівень паралелізму, низьку затримку та високу енергоефективність, що робить їх оптимальними для обробки великих даних у реальному часі. Інтеграція одноплатних комп'ютерів з технологіями Edge AI є перспективним напрямком, що забезпечує ефективне використання інтелектуальних систем у широкому спектрі практичних додатків. Однак через обмеження апаратних ресурсів питання реалізації рішень Edge AI потребує додаткових досліджень. Все це свідчить про актуальність теми роботи.

Метою кваліфікаційної роботи є підвищення ефективності рішень Edge AI за рахунок використання акселераторів ШІ.

Завданнями кваліфікаційної роботи є:

– обґрунтування вибору інструментарію для реалізації Edge AI;

– оцінка продуктивності рішень Edge AI на основі Hailo-8;

– формування рекомендацій щодо використання акселератору ШІ Hailo-8;

– техніко-економічне обґрунтування прийнятих рішень.

Об'єктом дослідження є процеси розгортання та функціонування моделей ШІ у комп'ютерних платформах з обмеженими апаратними ресурсами.

Предметом дослідження є інтеграція вбудованих комп'ютерних системах та акселераторів ШІ.

Методами дослідження є в рамках визначення інструментарію для реалізації Edge AI і техніко-економічного обґрунтування прийнятих рішень

використовувався аналітичний метод досліджень, а для порівняльної оцінки продуктивності рішень Edge AI на основі акселератора ШІ Hailo-8 – моделювання.

Інформаційна база кваліфікаційної роботи сформована з Інтернет-ресурсів, що місять інформацію про акселератора ШІ, одноплатні мікрокомп'ютери, нейронні мережі для впровадження Edge AI.

Практична значущість роботи полягає у розробці рекомендацій щодо використання акселератору ШІ Hailo-8 – можуть бути використані для подальших досліджень за даною тематикою та при проектуванні вбудованих комп'ютерних систем.

Апробація результатів відбувалася в рамках XX щорічної студентської наукової конференції «Сучасні інформаційні технології та інноваційні методики в економіці, менеджменті та бізнесі» Полтавського державного аграрного університету (16 квітня 2025 р., м. Полтава).

За результатами досліджень здійснено публікацію тез доповідей.

Структура кваліфікаційної роботи логічно пов'язана з завданнями досліджень і містить вступ, три розділи основної частини, висновки, список використаних джерел, додатки. Загальний обсяг пояснювальної записки кваліфікаційної роботи складає 63 сторінки формату А4. Вона містить 36 рисунків і 1 таблицю.

РОЗДІЛ 1
АНАЛІЗ ОСОБЛИВОСТЕЙ РЕАЛІЗАЦІЇ РІШЕНЬ EDGE AI

1.1 Загальні відомості про акселератори ШІ

Обчислювально важкий штучний інтелект (ШІ), що швидко розвивається, в першу чергу для завдань глибокого та машинного навчання, збільшив попит на обчислювальні ресурси. Більшість сьогоднішніх додатків штучного інтелекту є важкими для обчислень, що означає, що це ставить під сумнів доцільність традиційних алгоритмів машинного навчання (Machine Learning, ML) [1].

У результаті було розроблено спеціальне обладнання, а саме акселератори ШІ. Деякі прискорювачі розроблені для роботи з архітектурами паралельної обробки, які максимізують продуктивність з мінімальним використанням енергії, отже, виконують деякі функції, пов'язані зі штучним інтелектом, досить ефективно [2, 3].

Акселератори ШІ набули значного поширення в галузях, пов'язаних з автономними транспортними засобами, аналітикою даних у реальному часі, охороною здоров'я, Інтернет речей та ін. Це призводить до різкого збільшення обсягу даних у поєднанні з вдосконаленими алгоритмами з більшою обчислювальною потужністю, що каталізує розвиток ШІ. Найбільша увага приділяється нейронним мережам у дослідженнях ML та поширенню їх використання в системах і програмах.

Апаратні додатки, що призначені для покращення роботи машинного навчання на етапі висновків або навчання, прискорювачі ШІ використовуються для вищих швидкостей із меншою потужністю, у процесі розмиває оцінку різних прискорювачів. У той час як деякі з підходів – графічні та центральні процесори – залишаються домінуючими в деяких областях, розробка передових процесорів, що включають NPU і TPU, була зроблена, щоб бути більш ефективними в обробці деяких додатків ШІ.

Архітектура таких акселераторів ШІ кардинально відрізняється від архітектури звичайних процесорів (CPU). Акселератори ШІ відповідають за більшість операцій добутку матриць і тензорних операцій, які виконуються в нейронних мережах. Архітектури повинні забезпечувати розпаралелювання, низьку затримку, високу пропускну здатність для запуску моделей ШІ. Різноманітні прискорювачі, наприклад, графічні процесори (GPU), TPU, спеціальні мікросхеми ШІ, мають переваги. Тому вибір архітектури є дуже важливим, коли мова йде про проектування системи ШІ.

За своєю конструкцією акселератори постійно вдосконалюються, щоб задовольнити потреби в масштабуванні збільшення обчислювальної потужності, енергоефективності та масштабованості, з якими необхідно керувати робочими навантаженнями ШІ. Ці інфраструктури додатково руйнують звичайні архітектурні бар'єри та впроваджують нові парадигми з метою ефективного обчислення ШІ. У цьому напрямку деякі захоплюючі розробки включають нейроморфні обчислення та квантові прискорювачі, кожна з яких значно розширює межі інших технологій.

Нейроморфні обчислення черпають натхнення у структурі та функціональності людського мозку. Замість традиційних CPU, які виконують послідовність інструкцій, нейроморфні системи використовують масову паралельну архітектуру для обробки інформації подібно до біологічних нейронів. Завдяки використанню фотонних нейронних мереж (рис. 1.1) ці системи забезпечують ефективну обробку даних у діапазоні світлі, забезпечуючи швидку та енергоефективну передачу інформації, у якій оптичні сигнали служать як представлення даних (рис. 1.2) [4].

Сама конструкція нейроморфних процесорів спрямована на мінімізацію енергоспоживання при максимальному підвищенні ефективності обробки [4]. Таким чином, буде забезпечена продуктивність у режимі реального часу з рішеннями з низьким енергоспоживанням. Наприклад, в автономних системах, таких як дрони або роботи, де потужність і швидкість найбільше актуальні.

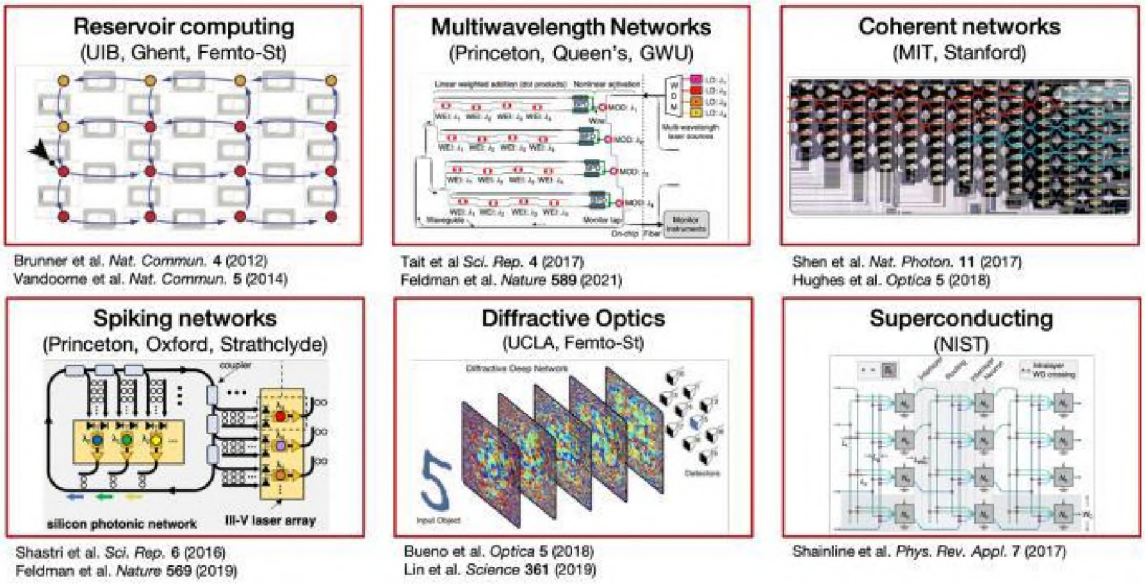

Рисунок 1.1 – Приклади нещодавно продемонстрованих архітектур фотонних нейронних мереж

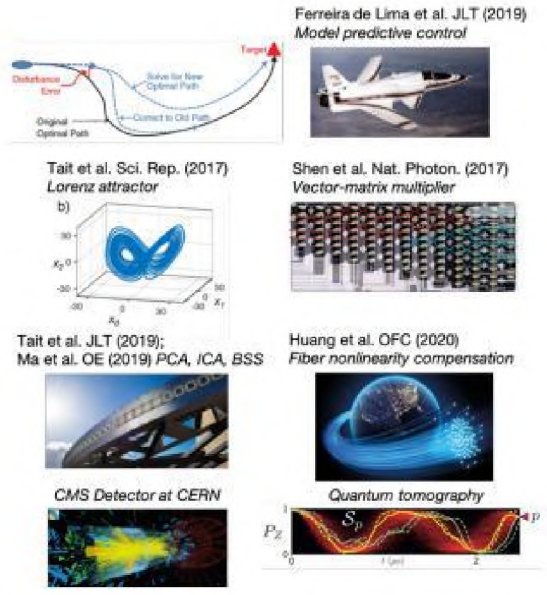

Рисунок 1.2 – Застосування фотонних нейронних мереж

Також нейроморфні обчислення все частіше використовуються в додатках Edge AI, особливо в тих випадках, коли обчислювальна потужність обмежена, а енергоефективність є вирішальною.

Такі процесори найкраще працюють із завданнями, які вимагають дуже швидкого процесу прийняття рішень під час обробки отриманої інформації, оскільки ці процесори керуються подіями.

У майбутньому, нейроморфні системи повинні бути ще більш досконалими для обробки постійно зростаючих складних навантажень ШІ. Перший багатообіцяючий напрямок майбутнього – це «навчання на чіпі», тобто пряма інтеграція алгоритмів навчання у нейроморфне обладнання.

Квантові обчислення – це наступний значний стрибок у обчислювальній потужності, і, очевидно, вони допоможуть вирішити завдання, які досі не вирішувались. Отже, квантові акселератори ШІ (рис. 1.3) покладатимуться на квантові біти (кубіти) при застосуванні за основними законами суперпозиції та заплутаності, щоб мати можливість протистояти декільком станам одночасно [5, 6].

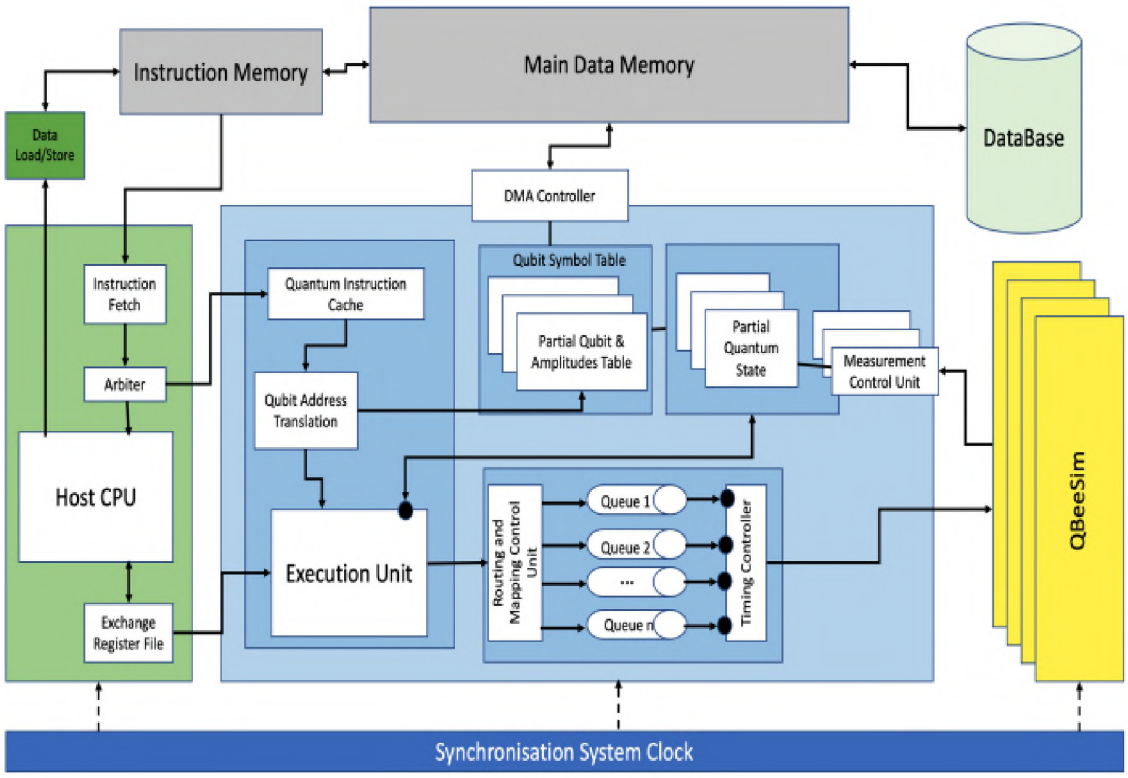

Рисунок 1.3 – Загальний приклад мікроархітектури квантового прискорювача

Це може призвести до квантових стрибків настільки, наскільки доступна потужність для виконання вибраних завдань ШІ: оптимізації, аналізу даних і складного моделювання та ін. Зручність квантових прискорювачів полягає в тих робочих навантаженнях, які передбачають

інтенсивні обчислення: навчання нейронної мережі, комбінаторні проблеми, оптимізація великих наборів даних.

Імовірно, квантові акселератори ШІ принесуть прорив лише в дуже специфічних підгалузях, де традиційні обчислення мають великі недоліки, наприклад створення нових ліків, фінансове моделювання та криптографія [5]. При цьому, квантові обчислення все ще є науковою сферою, що розвивається, з багатьма технічними складними завданнями, які ще потрібно вирішити, перш ніж вона стане остаточною.

Безумовно, стабільність кубітів, механізми виправлення помилок і дуже низькі температури, які підтримують квантові стани, – усе це потребує певних технологій. Однак, завдяки багатьом поточним дослідженням, вони неодмінно стануть новаторськими в наступні роки.

1.2 Аналіз ключових компонентів акселераторів ШІ

Надалі доцільно розглянуті архітектурні рішення акселераторів ШІ, включаючи блоки обробки, ієрархії пам'яті та з'єднання, на користь особливих переваг у продуктивності та ефективності завдань ШІ.

Сучасні GPU [7] складаються з значної кількості невеликих простих ядер, спеціально оптимізованих для виконання паралельної обробки.

Кожне ядро призначене для паралельного виконання основних арифметичних дій, а GPU виконує сотні потоків паралельно. Наприклад, GPU NVIDIA (рис. 1.4) складаються з кількох потокових мультипроцесорів (SM), кожен з яких містить кілька ядер потокового процесора (SP), як це реалізовано в архітектурі Fermi. Модель програмування CUDA дозволяє програмістам ефективно контролювати та керувати ресурсами GPU. Це особливо корисно під час виконання завдань, які містять дуже високий рівень паралелізму даних, як це спостерігається під час навчання глибоких нейронних мереж [8]. Потокові мультипроцесори (SM): GPU має багато SM,

кожен з яких містить численні ядра CUDA (у GPU NVIDIA) або потокові процесори (у GPU AMD). Ці SM виконують операції паралельних завдань.

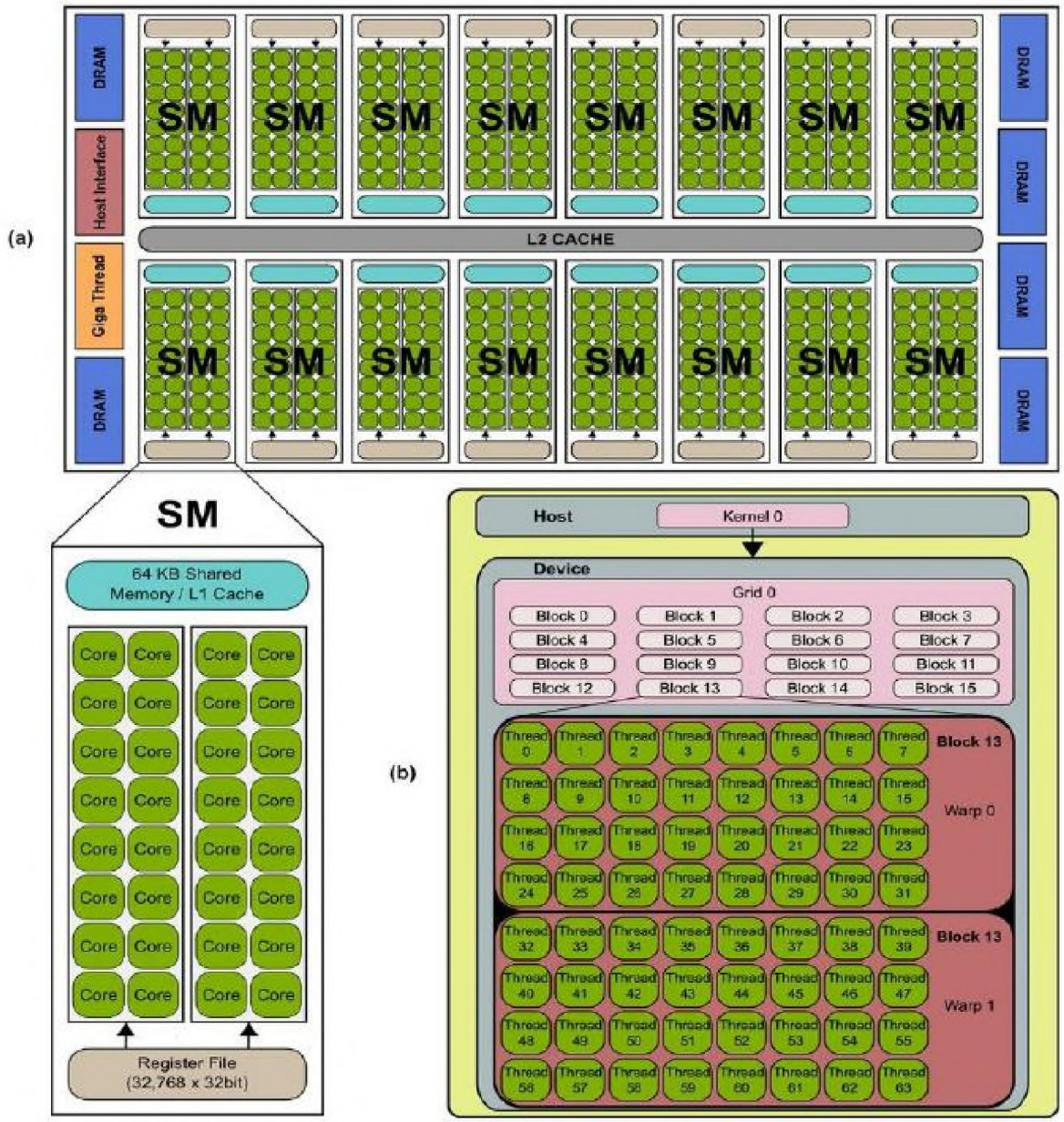

Рисунок 1.4 – GPU NVIDIA

Глобальна пам'ять – це величезні пули з високою пропускною здатністю, доступні для всіх ядер і використовуються для зберігання повних наборів даних і проміжних результатів.

Спільна пам'ять на чіпі між ядрами одного SM, яка використовується для тимчасового зберігання та зв'язку між потоками, прискорює обробку.

Текстурні блоки, зазвичай, використовуються у разі деяких специфічних і складних режимів доступу та роботи з текстурами, він надає спеціальні блоки, які можуть бути дуже корисними під час виконання програм ШІ для рендеринга графіки.

Блоки обробки тензорів (TPU) представляють доменні процесори, що розроблені к. Google для прискорення виконання тензорної обробки, яка є основною операцією в алгоритмах глибокого навчання (рис. 1.5).

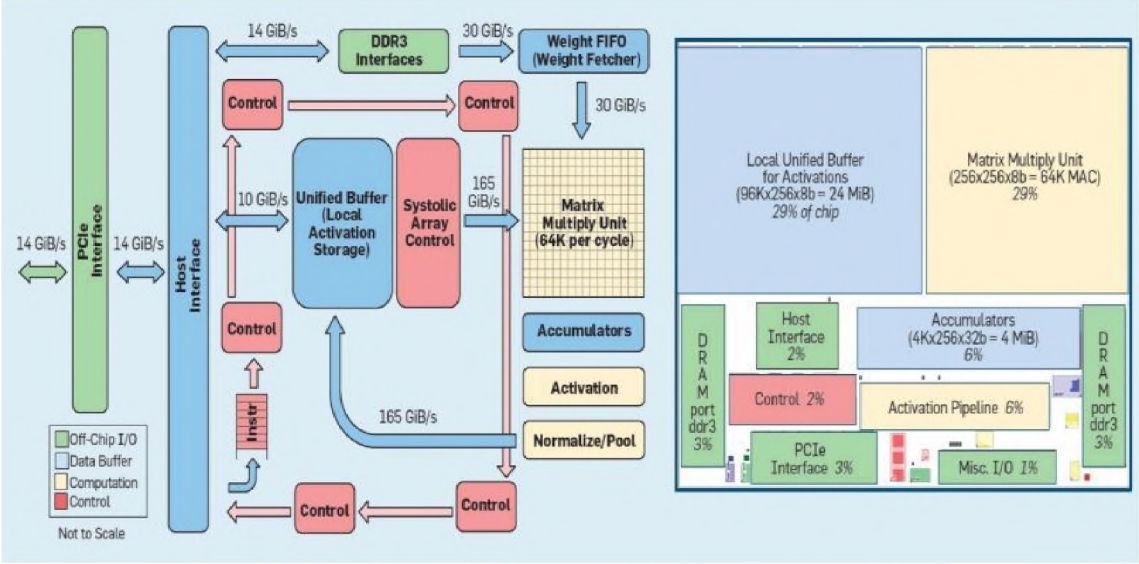

Рисунок 1.5 – Архітектура TPU від к. Google

У TPU використовується програмований блок обробки матриць під назвою Tensor Processing Core (TPC) для виконання дуже ефективного високопродуктивного множення матриць [9].

Блоки множення матриці – цей клас спеціалізованого обладнання виконує операції множення матриці в основі навчання нейронної мережі та виводу.

Інтегрована пам'ять з високою пропускною здатністю, що забезпечує швидкий і великий доступ до даних, зменшує вузькі місця, які, зазвичай, виникають під час передачі даних у звичайній архітектурі.

Функціональні одиниці активації – це пристрої, які швидко виконують обчислення функцій активації, вирішальних для обчислень нейронної мережі.

Блок нейронної обробки (NPU) – це процесор спеціального призначення для оптимізації обчислень у нейронних мережах, які здебільшого виконуються на межі, де виникають обмеження потужності та розміру (рис. 1.6). Оптимізовані архітектури для деяких операцій нейронної мережі, таких як згортка та множення матриць, вступають у дію в NPU.

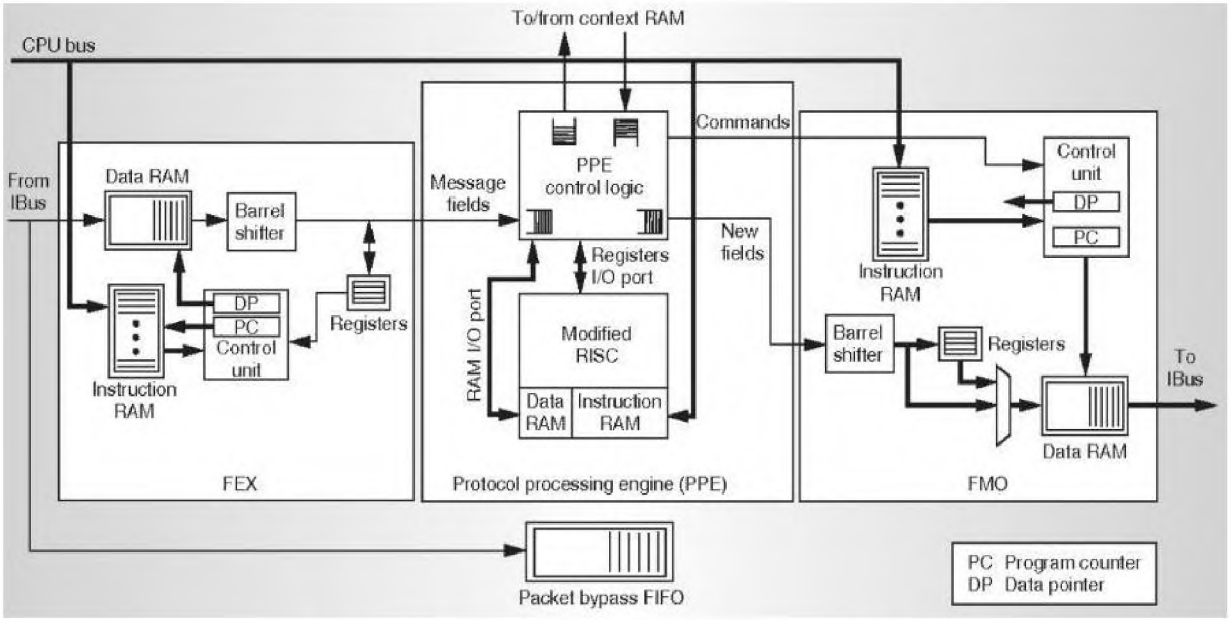

Рисунок 1.6 – Гібридна архітектура NPU

До основних компонентів NPU варто віднести наступні складові. Елементи обробки – це масиви блоків обробки, що оптимізовані для виконання операцій при функціонуванні нейронній мережі. Такі операції можуть включати згортки. Внутрішня пам'ять – це високошвидкісна пам'ять, розташована поруч із елементами обробки для мінімізації затримок доступу. Архітектура потоку даних – спеціалізована архітектура для переміщення даних і ефективного виконання обчислень, яка часто містить модуль високопаралельної обробки в нейронній мережі.

Ієрархія пам'яті відіграє вагому роль у характеристиках продуктивності акселераторів ШІ. Пропускна здатність пам’яті, зазвичай, вважається вузьким місцем для багатьох програм ШІ, особливо тих, що включають інтенсивні набори даних. Він являє собою максимальну швидкість, з якою дані можуть

обмінюватися між процесором і пам'яттю [7]. Внутрішня пам'ять у вигляді кешу або блокнота є дуже швидкою, хоча має відносно невеликий розмір і використовується для частих звернень, щоб заощадити цикли читання із зовнішньої пам'яті. Пам'ять високої пропускної здатності (HBM) і пам'ять GDDR широко використовувалися в графічних процесорах і прискорювачах штучного інтелекту для обробки велики даних, які надходять із виконанням будь-якого завдання ШІ. Технологія пам’яті розроблена для швидкого переміщення даних у блоки обробки та з них, щоб не перешкоджати переміщенню даних.

Це означає, що на практиці ефективні стратегії керування пам’яттю, наприклад, попередня вибірка даних і кешування, стають одними з найважливих варіантів оптимізації для акселераторів ШІ. Швидкість обробки пристроїв найчастіше обмежена повільною передачею даних. Однак навіть із такими процесорами та пам’яттю продуктивність акселераторів ШІ принципово залежить від потоку даних між цими різноманітними компонентами. Сучасний статус належить міжсистемним з’єднанням. Вони призначені для забезпечення безперебійного проходження даних від процесора до пам’яті та інших апаратних підсистем [7]. Сучасні акселератори ШІ містять різні типи з’єднань. До них належать наступні. Внутрішні з’єднання (рис. 1.7) – ці засоби роблять можливим передачу даних для зв’язку ядер на одному процесорі [8].

Таким чином, їх ефективність для внутрішньо-кристальних з’єднань дуже важлива для низької затримки виконання та максимального використання всіх завдань паралельного процесу. З’єднання поза чіпом дозволяє під’єднати кілька чіпів або прискорювачів, підтримуючи розподілене обчислення в системі. Інтерфейсні технології, такі як PCIe та NVLink, широко використовуються в прискорювачах штучного інтелекту, оскільки вони швидші, ніж більшість інших, з точки зору зв’язку між процесорами та пам’яттю, розподіленою по мережі мікросхем. Мережа на чіпі (NoC) – це масштабована технологія взаємозв’язку, яка має на меті

покращення як масштабованості, так і пропускної здатності зв'язку на чіпі з найменшою можливою затримкою, розроблена для SoC і багатоядерних процесорів.

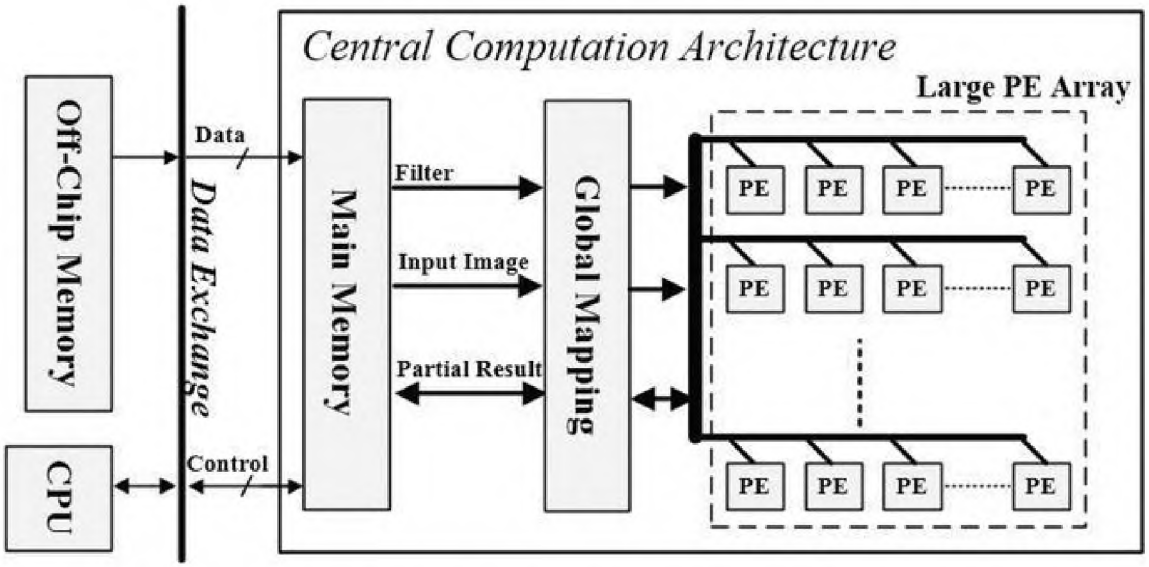

Рисунок 1.7 – Переміщення даних

Гарні міжсистемні зв'язки допомагають усунути вузькі місця у переміщенні даних, дозволяючи виконувати завдання ШІ паралельно та без великих затримок. Продуктивність акселераторів ШІ, зазвичай, обмежена їхньою здатністю переміщувати дані з високою швидкістю між різними компонентами. Це робить дизайн з'єднання дуже важливим.

1.3 Основні аспекти застосування одноплатних мікрокомп'ютерів у проєктах Edge AI

Сучасні моделі AI вимагають значних обчислювальних ресурсів, що тривалий час обмежувало їх застосування переважно потужними серверними рішеннями та хмарними платформами. Однак, завдяки розвитку технологій периферійних пристроїв та появі спеціалізованих акселераторів ШІ, обробка даних та виконання складних моделей стало можливим навіть на пристроях з

обмеженими ресурсами. Як наслідок, все більше з'являється проєктів Edge AI (штучний інтелект на периферії). Це інтегрована парадигма, яка передбачає перенесення обчислювальних ресурсів, ШІ та методів ML, ближче до джерел даних – на периферію комп'ютерної мережі (Edge). На відміну від традиційних хмарних обчислень, де дані пересилаються в центральні сервери для обробки, Edge Computing з інтегрованими рішеннями ШІ передбачає локальну обробку даних, мінімізуючи затримку, підвищуючи ефективність використання ресурсів та покращуючи приватність інформації.

Поєднання Edge Computing з технологіями ШІ з'явилось внаслідок потреби у швидкій та ефективній обробці великих даних від сучасних пристроїв Інтернету речей (IoT), сенсорами, автономними транспортними засобами, промисловими системами та мобільними платформами. Така інтеграція дозволяє проводити складні аналітичні процеси та приймати рішення в режимі реального часу, без передачі великих обсягів даних через глобальні мережі. Це відкриває нові можливості для реалізації AI-рішень на периферії, у тому числі на IoT-пристроях, мобільних платформах, а також промислових та побутових пристроях [10].

Одноплатні мікрокомп'ютери (Single-Board Computers, SBC), такі як Raspberry Pi [11], NVIDIA Jetson [12], Google Coral [13], BeagleBone [14], Arduino Portenta [15] та ін., відіграють ключову роль у реалізації концепції Edge AI завдяки своїм специфічним технічним особливостям, компактності, енергоефективності та доступності для широкого кола користувачів. Вони забезпечують локальні обчислювальні ресурси для запуску алгоритмів ШІ безпосередньо в точках збору та первинної обробки інформації. Це дозволяє скоротити час відповіді та мінімізувати залежність від зовнішніх (хмарних) сервісів, що є критично важливим для додатків, чутливих до затримок (наприклад, автономне керування транспортом, промисловий моніторинг). Сучасні SBC часто оснащені спеціалізованими апаратними акселераторами ЩІ (наприклад, Coral Edge TPU, NVIDIA CUDA та Tensor Cores, Intel Movidius NCS), які суттєво підвищують продуктивність нейронних мереж.

Такі акселератори дозволяють реалізовувати складні алгоритми ML, включаючи нейронні мережі для задач комп'ютерного зору (CV), обробки природньої мови (NLP) та аналізу сенсорних даних. Завдяки відкритим апаратним і програмним платформам, одноплатні комп'ютери легко інтегруються в різні конфігурації й мережеві інфраструктури, дозволяючи швидко адаптуватися до конкретних задач і сценаріїв використання. Це особливо важливо у швидко змінюваних умовах, де потрібно динамічно змінювати конфігурації та оновлювати програмне забезпечення.

SBC відрізняються низьким енергоспоживанням і малими розмірами, що дає змогу встановлювати їх у віддалених, мобільних або автономних пристроях (наприклад, на борту дронів, у портативних медичних приладах, мобільних сенсорах або роботах). Це критично важливо для автономних систем, які живляться від батарей або поновлюваних джерел енергії.

Відносно низька вартість, велика кількість доступних програмних бібліотек, інструментів і підтримка спільнот дозволяють розробникам та дослідникам ефективно створювати та тестувати AI-рішення для Edge Computing. Це сприяє інноваціям та швидкому впровадженню нових технологій у практичне використання.

Використання SBC для локальної обробки даних дозволяє зменшити ризики порушення конфіденційності, оскільки обробка здійснюється безпосередньо там, де ці дані генеруються, і немає потреби у передачі їх через глобальну мережу.

В цілому, вказані SBC є ключовим елементом інфраструктури Edge AI, забезпечуючи ефективну, швидку та безпечну реалізацію інтелектуальних систем у широкому спектрі додатків: від побутових розумних пристроїв до промислових автономних систем та медичних технологій. Застосування AI у контексті Edge Computing може включати такі методи та технології, як глибинні нейронні мережі (Deep Neural Networks, DNN), згорткові нейронні мережі (Convolutional Neural Networks, CNN), рекурентні нейронні мережі (Recurrent Neural Networks, RNN), алгоритми ML, а також різні підходи до

оптимізації моделей для використання на пристроях з обмеженими обчислювальними потужностями.

Прикладами таких підходів є кількісна компресія моделей (quantization), знання-дистиляція (knowledge distillation), проріджування нейронних мереж (network pruning) та спеціалізовані апаратні акселератори.

В цілому, до основних переваг інтеграції Edge Computing та ШІ варто віднести наступні положення.

1. Низька затримка (Low Latency). Зменшення часу, необхідного для отримання аналітичних висновків, завдяки локальній обробці даних.

2. Підвищена безпека даних (Data Security). Оскільки дані не передаються в хмару, знижується ризик перехоплення конфіденційної інформації.

3. Зниження навантаження на мережу (Reduced Network Load). Локальна обробка зменшує необхідність у передачі великих обсягів даних через мережу.

4. Автономність та надійність (Autonomy and Reliability): Можливість ухвалення рішень незалежно від централізованих хмарних сервісів дозволяє системам функціонувати навіть при втраті зв'язку.

5. Енергоефективність (Energy Efficiency): Оптимізація моделей та спеціалізоване обладнання сприяють зменшенню енергоспоживання.

В якості типових галузей застосування рішень Edge AI слід виділити:

– автономний транспорт (безпілотні автомобілі, дрони, автономні роботи та ін.);

– промислова автоматизація (інтелектуальні системи моніторингу обладнання, прогнозне обслуговування);

– розумні міста та інфраструктура (інтелектуальне керування дорожнім рухом, системи безпеки);

– телемедицина та охорона здоров'я (віддалена діагностика, моніторинг стану пацієнтів у реальному часі);

– системи відеоспостереження (аналіз відеопотоку на місці з метою безпеки чи ідентифікації об'єктів).

Таким чином, Edge AI являє собою інноваційний підхід до організації обчислювальних процесів, спрямований на розв'язання завдань швидкої та ефективної обробки великих даних, з одночасним забезпеченням високого рівня безпеки, надійності та продуктивності у широкому спектрі технологічних і промислових сфер.

РОЗДІЛ 2
ЗАСТОСУВАННЯ МІКРОКОМП'ЮТЕРІВ ТИПУ RASPBERRY PI З АКСЕЛЕРАТОРАМИ ШІ

2.1 Визначення критеріїв для оцінки апаратних компонентів рішень Edge AI

Враховуючі різноманітність сфер застосування SBC в проєктах Edge AI (див. п. 1.3), для формування пріоритетного вибору апаратних компонентів потрібно оцінити їх характеристики та властивості. Для цього доцільно використовувати певні критерії. Їх можна розділити на кілька категорій: споживчі, інженерні, дослідницькі. Ці критерії не мають чіткої межі. Те саме енергоспоживання – можна віднести до споживчих або інженерних. Але такий підхід певним чином дозволить сформувати рекомендації щодо вибору для конкретного рішення Edge AI.

Споживчі критерії – це те, як апаратний компонент бачить споживач або фахівець, що безпосередньо замається інтеграцією в продакшн.

1. Ціна компоненту (плати SBC) на виробництві. Плата на базі SG2002 може коштувати 5 $. А плата на базі Jetson Orin може сягати 1000 $. По між них існує велика множина апаратних рішень.

2. Чи потрібне власне виробництво – якісь плати продаються лише у форматі чіпів, наприклад, Hailo-15 [16] – плату на його основі не купити. Так само, нічого не вийде, якщо потрібна правильна конфігурація роз'ємів, або мінімальна ціна.

3. Можливість випуску великої партії. Як відомо, через великий попит існують проблеми з постачанням Jetson (немає проблем тільки у тих, кому пообіцяла відвантажити к. Nvidia).

4. У якій країні випущено продукт. Якщо випускаєте продукт у США – не зможете купити Huawei. І немає сенсу, його все одно не продати. Якщо робите продукт для госпіталів у Європі – швидше за все не можна буде

використовувати RockChip (це і сертифікація та обмеження на постачальників обладнання).

5. Вартість розробки ML на платі. На SBC типу Jetson ціна буде мінімальною, а на якомусь мікрочіпі – максимальна.

6. Яке енергоспоживання плати. Якщо треба вбудувати розпізнавання обличчя у дверний дзвінок на батарейці, це один рівень плат. А якщо дозволено витрачати сотні ватів на розпізнавання – це інший рівень.

Коли розмірковують над споживчими вимогами – необхідно хоча б грубо обмежити характеристики по кожному з пунктів (які обсяги бажає замовник, який ціновий діапазон, форм фактор, країни-виробник, для якої країни готується продукт та ін.

Інженерні критерії – це як виглядає плата для взаємодії з нею.

1. Система. Хтось хоче систему ОС Windows (зазвичай, це рідкість) або Linux (але який його варіант який: Ubuntu, YOCTO або BuildRoot). А можливо ніякої ОС не треба, наприклад, як в ESP32, чи все ж таки з підтримкою MicroPython? Тобто це все дуже сильно впливає на те, що має вміти команда-розробник, на зручність менеджменту, на простоту виготовлення пристроїв.

2. Окремий пристрій чи ні. Деякі прискорювачі для нейронних мереж – це окремі плати, а якісь інтегровані у процесор. Зрозуміло, що це різні варіанти виводу (інференсу) для різних завдань.

3. Наскільки продуктивним є процесор. Виконання нейронних мереж – це часто не все, що потрібно алгоритмам. І треба дивитися на те, наскільки потрібен процесор:

– чи може він встигати перепроцесувати зображення;

– чи здатен встигати декодувати та кодувати відео;

– чи має можливість обробляти 3D та ін.

4. Підтримка з боку виробника. Часто такі плати дуже обмежені та документація не повна. Чи потрібна для розробки консультація з боку виробника – бо не завжди достатньо відкритих джерел.

Дослідницькі критерії стосуються ШІ, що реалізується на платі.

Зазвичай, їх ігнорують під час вибору плати, але це може мати вплив на час розробки у десятки разів у порівнянні з іншими.

1. Швидкість виводу – це ключовий параметр для багатьох застосувань. Зрозуміло, якщо плата дає лише 1 FPS для детектування об'єктів – нічого не можна зробити якщо треба реалізовувати Object detection на 1000 FPS.

2. Підтримувані шари нейронної мережі. В даному випадку оцінюється складність експортних інструментів, а також що надає виробник, чи потрібна квантизація, чи підтримується LLM [17].

3. Об'єм та швидкість пам'яті.

Надалі доцільно для кожного критерію навести кілька характерних прикладів, де критерій добре працює і де погано, а також розглянути основні апаратні компоненти для впровадження ШІ.

2.2 Оцінка технічного базису рішень Edge AI

Сімейство Jetson від к. Nvidia (рис. 2.1) [12]. Перші представники цього сімейства почали масово використовуватись для додатків ШІ ще 2015 р. З того часу вони стають лише кращими. На сьогоднішній день актуальною серією є Jetson Orin. У цій серії є три типи пристроїв (Nano, NX, AGX). Вони відрізняються за ціною та обчислювальними можливостями (табл. А.1). Старі версії Jetson все ще використовуються, але вже не дуже часто ніж поточні. Усередині специфікацій AGX та NX пристроїв є кілька підтипів, які, знову ж таки, різняться за ціною та швидкістю.

Модулі Jetson Nano та Jetson Xavier NX, що входять до складу комплекту розробника Jetson Nano та комплекту розробника Jetson Xavier NX, мають слоти для використання карт microSD замість eMMC як системних пристроїв зберігання даних.

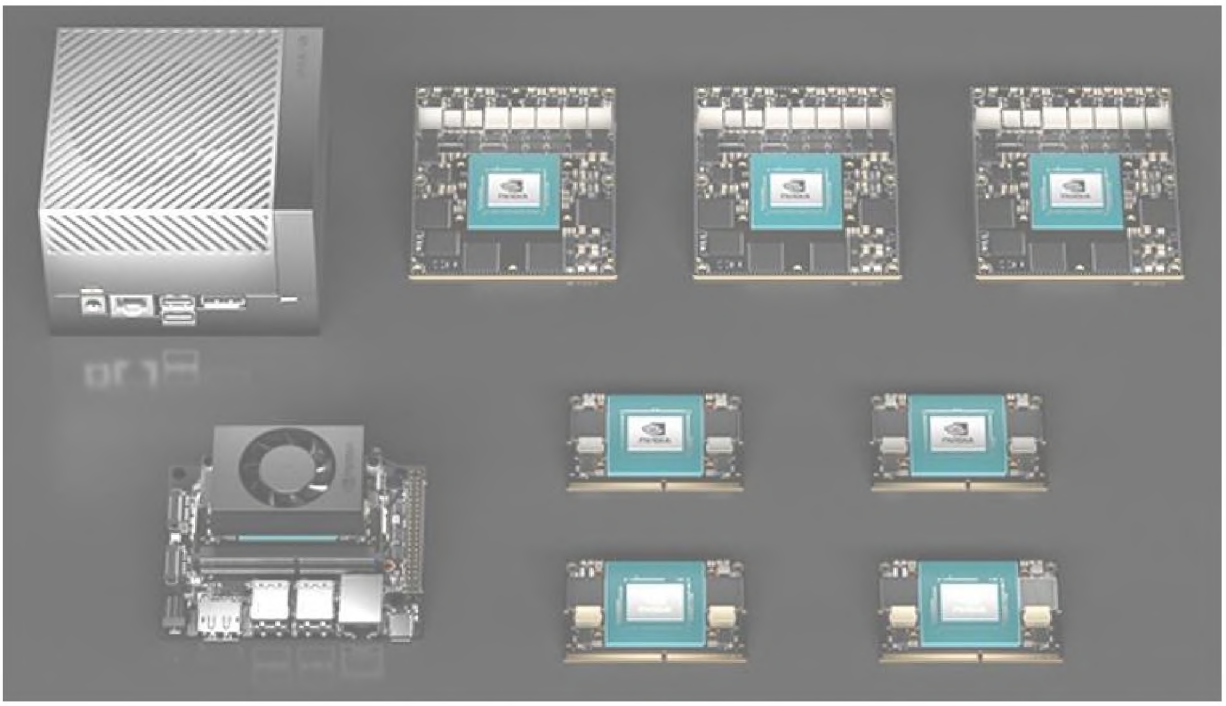

Рисунок 2.1 – Сімейство Jetson від к. Nvidia

Jetson, насамперед, це плата GPU. Є кілька версій, де GPU дає меншу продуктивність ніж NPU. Але GPU навіть дуже зручно (не потрібна квантизація вагів нейронної мережі). CPU в Jetson досить слабкий виконання для мереж, а NPU – поки що є не скрізь. Ключова відмінність між різними моделями – саме NPU. Його немає у Nano. У NX їх 1 або 2 (залежно від моделі). У AGX частота майже в 3 рази більша. Jetson Nano минулого покоління за продуктивністю досить сильно програє актуальним версіям. Але більшість ідей та логіки роботи залишилася незмінною. Сучасний NPU добре працює тільки в моделей з int8, але за рахунок шарів fallback можна забезпечити обчислення окремих шарів через GPU.

До плюсів Jetson можна віднести такі моменти:

1. Велика інфраструктура навколо. TensorRT, Triton, CUDA, etc. Майже все що може бути запущено на робочому столі – можливо без проблем запущено на Jetson.

2. Висока швидкість. Якщо перекладати TOPS на USD – може Jetson і не найвигідніший. Але з плат цього формату – явно один із найпродуктивніших.

3. Велика кількість інформації в Інтернет. Майже на будь-яку проблему вже є топік в Інтернет.

4. Підтримка сучасних моделей. Так, щось може не працювати. Але більшість LLM, VLM та ін. вже апробовано. І це якісна відмінність від 95 % всіх інших плат.

5. Можливість писати низькорівневий код через TensorRT.

До мінусів Jetson відносяться:

– ціна;

– доступність (складно забезпечити безперебійні поставки);

– енергоспоживання – к. Nvidia часто звітує, що кожен новий Jetson більше і більше енергоефективний;

– NPU насамперед орієнтований на int8.

x86(Intel, AMD). Говорячи про x86 в першу чергу треба говорити саме про к. Intel. Основний мінус таких обчислювачів – велике енергоспоживання. При цьому продуктивність можна порівняти з Jetson'ом. А пристрої значно доступніші. Базовий CPU вивід працює за замовчуванням, а також є підтримка форматів ONNX Runtime, OpenVino, TorchScript, etc. Є можливість ефективно обчислювати всі сучасні мережі (деякі через PyTorch). На даний час існує гарне ком'юніті, підтримка. До недоліків відносяться:

– NPU та GPU не в кожному пристрої;

– є мережі де підтримка і швидкість гірша за пристрої від к. Nvidia;

– енергоспоживання часто вище ніж у пристроїв Jetson;

– ціна виходить на рівні Jetson.

Щоб закрити гілку «класики», варто розглянути SBC на інші архітектурах CPU (ARM, RISK-based). Ті плати, які придатні для Embedded розробки та використовують ARM/RISK, зазвичай, значно повільніше ніж x86. При цьому це не заважає їм іноді бути досить швидкими, щоб вирішувати множину інтелектуальних завдань. Ті ж самі RockChip, MediaTec, Huawei, що розглядаються нижче, мають більш ніж гідні процесори, які можуть взяти на себе завдання ML для багатьох ситуацій (CV, NLP та ін.).

При цьому очевидно, що «import onnxruntime» з коробки – це досить просто та зручно. Однак, зазвичай, енергоспоживання та максимальна швидкість буде програвати більшості модулів NPU або ×86, а також непоганим відеокартам (наприклад, від к. Intel).

RockChip (рис. 2.2). На даний час їх багато, і вони дуже гарні для завдань ML. У них чудовий модуль NPU з великою підтримкою різних мереж. Добра половина сучасних плат Edge заснована на них, наприклад, OrangePi, Radxa (RockPi), Banana Pi, NanoPC, Khadas, FireFly.

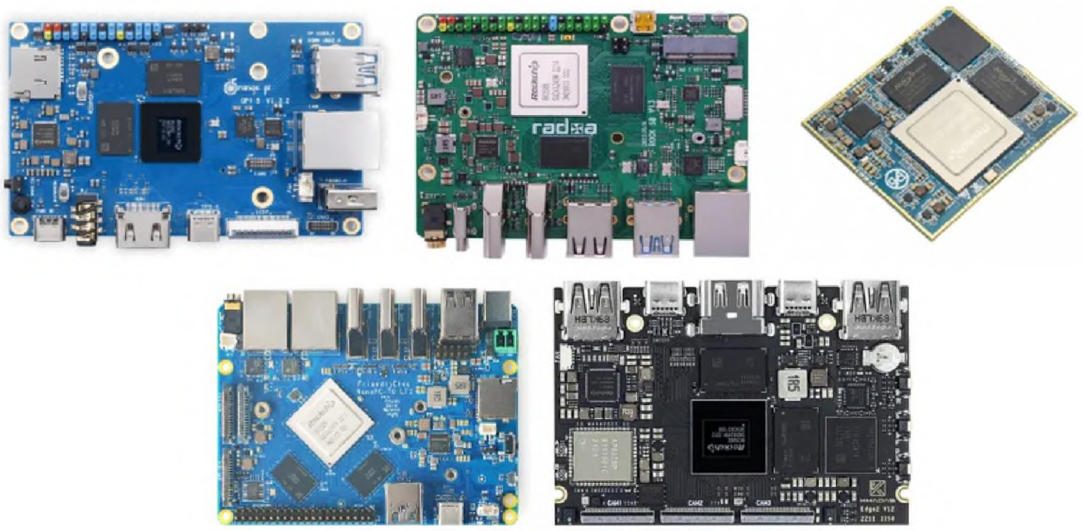

Рисунок 2.2 – Сімейство RockChip

Вони робляться на основі різних плат:

– RK3588 – найпотужна та топова за продуктивністю (містить кілька версій, наприклад, RK3588s, 3582 та ін.);

– RK3568 – одна зі старих плат (досить повільна і неоптимальна за ціною);

– RK3566 – дешева плата (Linux + NPU);

– RK3576 – аналог 3588, але трохи простіше процесор;

– RV1106, RV1103 та кілька інших – плати без повноцінної ОС Linux та Python виводу;

– RK3399Pro – найстаріша NPU плата, зараз вже майже не підтримується та ін.

До переваг RockChip можна віднести такі чинники.

1. Ціна.

2. Доступність. Можна придбати від десятків різних виробників. Штучно та великими партіями.

3. Обсяг мереж, що підтримуються. Звичайно, вони поступаються серії Jetson. Але можна знайти майже будь-яку нейронну мережу. Зараз межа проходить приблизно так: LLM – вони підтримують, а VLM – ще ні. Якісь трансформери працюють, а Whisper – в однієї команди під ліцензією GPL.

4. Багато різноманітних форм-факторів. Можна купити повністю готову платню або розвести її з нуля.

5. Формат вагів fp16 також обчислюється на NPU. Це дуже важливо, тому що не будь-яку мережу швидко і легко можна запустити під int8.

6. Є деякий низькорівневий доступ до NPU – можна дуже багато математики виконувати на ньому вручну.

Мінусами вважається таке:

1. Якість. Багато вендорів роблять сирі системи на базі RockChip. Сам RockChip – теж не сказати, що дуже якісно код робить.

2. Країна виробник – Китай. У США та ЄС можливі різні обмеження.

3. Не будь-яку мережу можна запустити.

4. Складна архітектура NPU у старших моделей. Свого сервера виводу немає, тобто треба створювати мультипотоковий вивід, щоб максимізувати швидкість виконання.

Qualcomm – вагомий конкурент у сфері Edge Computing. На даний час, актуальною є плата RB3 (рис. 2.3). У неї швидкий вивід, при замовленні великих партій – плата виходить дешевою. Має досить гарну підтримку нейронних мереж та документацію. З іншого боку, доступ до середовища розробки може тривати до місяця. При цьому приватний покупець не може купити плату та підписати всі договори; немає відкритої інформації, тобто нема впевненості, що LLM так і не працюють повноцінно (включаючи VLM). Також немає низькорівневого доступу до NPU.

Рисунок 2.3 – Плата RB3 від к. Qualcomm

VeriSillicon –по-справжньому багатогранні плати. Компанія VS продає дизайн чипів, наприклад, NPU отримують:

– NXP – одна з найбільших компаній з виробництва електроніки;

– Amlogic – один із лідерів малорозмірних процесорів (хоча в останніх платах VS вже немає);

– STM32 (звичайно, це не у всіх платах, але на найпродуктивніших);

– Synaptics;

– BroadCom та ін.

Так як компанія надає залізо та набір низькорівневих бібліотек – експіріанс від двох різних вендорів може бути принципово інший. По суті, якщо вендор хороший (такий як NXP) – використання супер інтуїтивно. В цілому, перевагами слід вважати такі положення:

– це досить енергоефективна архітектура;

– дуже багато вендорів, що продають у різних форм-факторах;

– чіпи досить дешеві.

До недоліків варто віднести:

– не всі мережі підтримуються (жодних LLM, VLM та ін., а це ніяк не можна виправити через те, що не має низькорівневого доступу до NPU;

– є вендори для яких експорт не дуже добре працює;

– це не дуже швидкі плати.

Orange PI AI Pro (к. Huawei) – плата досить гідна та відкрита, має гарну швидкість (рис. 2.4). Вона дешева, достатньо вільний експорт більшості моделей, багата документація. Однак, у неї немає підтримки LLM, орієнтована на китайський ринок, тому основна документація китайською. Через санкції не купити у Європі та США.

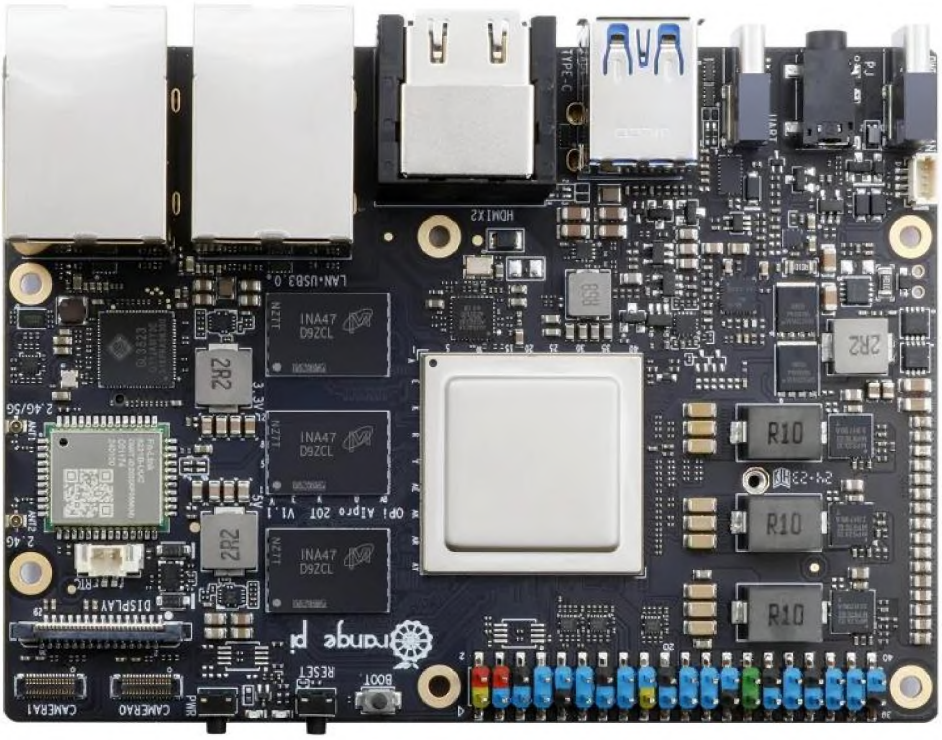

Рисунок 2.4 – Плата OrangePi AI Pro

Крім плат SBC доцільно розглянути зовнішні плати акселераторів ШІ. Вони намагаються вирішити завдання «додати недостатню AI-потужність наявній системі» та підключаються окремо (рис. 2.5). Важливе для прискорювача – це як він підключений. Здебільшого це PCI-e(M.2) або USB-прискорювачі. При цьому слід враховувати декілька чинників.

1. Затримка на передачу. Якщо потрібно швидко реагувати – це критично.

2. Який обсяг даних може бути передано через канал. Якщо запускаєте нейронні мережі на великих зображеннях або відео, це може сильно

обмежувати. Існують прискорювачі від PCI-e(2)x1 до ×4. І для пристроїв це також важливо. Наприклад, на у Raspberry Pi тільки 1 лінія (офіційно PCI-e (1), але на практиці PCI-e (2)).

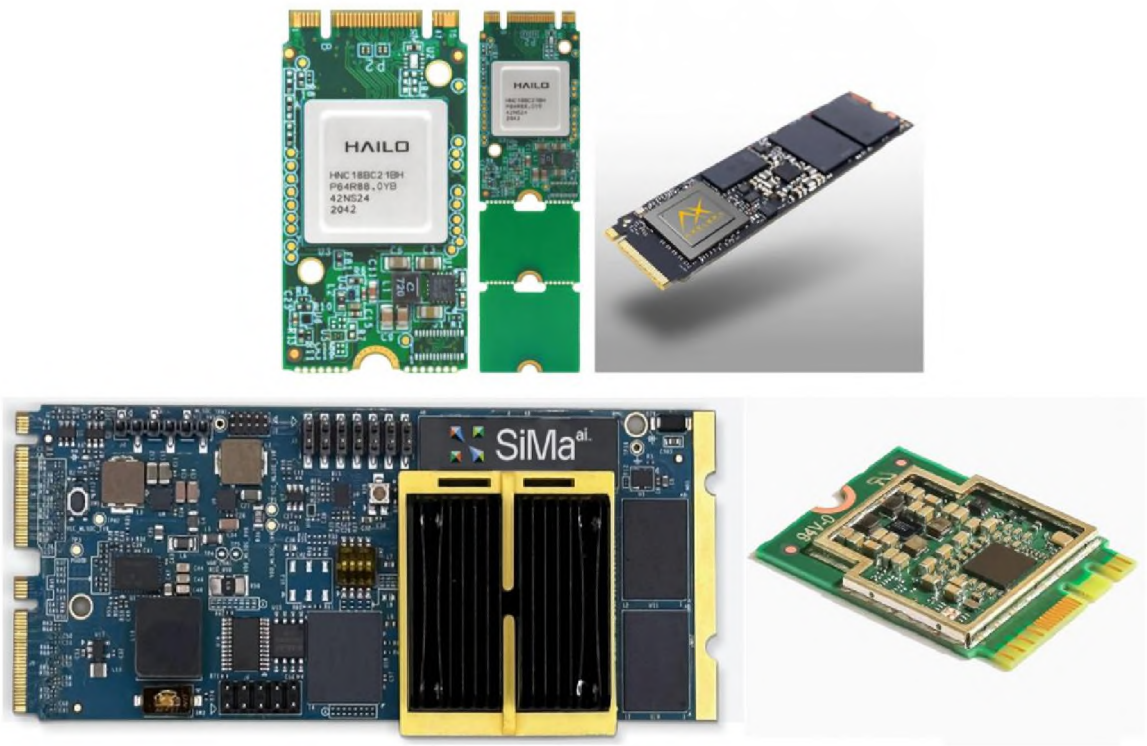

Рисунок 2.5 – Приклади плат акселераторів ШІ

3. Чи достатньо швидкий процесор, щоб підготувати дані та відправити їх по шині. На повільних платах обчислення можуть бути сильно повільними навіть коли є швидкий прискорювач.

Розглянемо популярні з акселераторів ШІ.

Hailo-8 та Hailo-10. Вони мають гарну підтримку, відкрита спільнота та є швидкими. Їх достатньо просто купити, є керівництва по спільному використанню з Raspberry Pi, є досить добрі інструкції на експорт моделей. Дуже багато алгоритмів квантизації із коробки. З іншого боку, Hailo буде дорожче коштувати, ніж RockChip (але дешевше ніж Jetson). Необхідність проведення квантизації. Багато Transformer Based моделей не працюють (LLM/VLM/Whisper). Можливо ще щось. Hailo обіцяє налагодити підтримку. Спеціально для LLM вони випустили Hailo-10. Але поки що можливості

запуску немає. Однак є ще Hailo-15. І, на відміну від своїх побратимів, це спеціальний модуль процесора, а не зовнішній модуль. Його плюси та мінуси більш менш схожі. Але є кілька моментів. Він повільніший, ніж Hailo-8 та Hailo-10. У нього слабкий процесор. Однак, якщо вам не потрібно обробляти багато камер або виконувати складну попередню обробку, вам точно вистачить. Шина процесор-NPU швидка. Він дешевше. Готових плат із коробки немає. Потрібно розробляти на основі референсного дизайну (принаймні так було нещодавно).

Axelera [18] – дуже швидкий. Судячи з документації, одна з найшвидших зовнішніх плат (рис. 2.6). Є велика кількість форм-факторів. Але достатньо дорогий (можливо, це стосується окремих плат). Не всі мережі-трансформери працювали (LLM, VLM та ін.).

SIMA [19] – менше орієнтується на Edge Computing у плані «обчислення навколо камери», а скоріше «обчислення на локальному сервері» (рис. 2.7). І це скоріше альтернатива GPU, ніж прискорювач для локальної плати.

Рисунок 2.6 – Axelera M.2 AI Edge accelerator card

Рисунок 2.7 – Мікросхеми SIMA для ML

Плюсами SIMA вважаються такі чинники. Це єдиний виробник таких плат, який більш-менш обіцяв підтримку LLM та їх аналогів. Одні з найшвидших за підключенням (дуже багато ліній PCI-e 4-го покоління, вони підтримують до 8 каналів.) З іншого боку, вони одні з найдорожчих, великі та енергоспоживаючі (тобто не зовсім Edge Computing).

Також є інші акселератори ШІ, що підключаються. Здебільшого про них дуже мало інформації (крім Coral): Kneon (USB). Coral (USB, M.2, PCI-E) [13], Gyrfalcon, BrainChip, Kinara (Ara-2 40 TOPS (USB або M.2 – рис. 2.8) [20] та ін.

Рисунок 2.8 – Акселератор ШІ від к. Kinara

Ambarella – одна з найзагадковіших плат [21]. На неї не знайти огляди в Інтернет. Але при цьому в багатьох дешевих масових пристроях використовується саме вона. Множина відеореєстраторів, DJI-камери та ін. При цьому існує повна політика закритості. Для зареєстрованих команд компанія пропонує ПЗ Cooper, що підтримує весь портфель SoC зі ШІ Ambarella, включаючи наступні програмні модулі Cooper Foundry (рис. 2.9): Cooper Core – пропонує ОС, компілятор і SDK на базі Linux; Cooper Foundation – дозволяє створювати та розгортати програми машинного навчання на периферії; Cooper Vision – включає в себе основні будівельні блоки для мультимодальної обробки датчиків і злиття, включаючи дані з камер, радарів і LiDAR; Cooper UX – забезпечує аналітику та розробку.

Для завдань CV пропонується архітектура чіпа CVflow. На відміну від CPU і GPU загального призначення, CVflow включає спеціальний механізм обробки зору, запрограмований на високорівневий опис алгоритму, що

дозволяє масштабувати продуктивність до трильйонів операцій за секунду з надзвичайно низьким енергоспоживанням. За допомогою CVflow клієнти можуть ефективно картографувати власні мережі CNN, навчені стандартними інструментами (наприклад, Caffe, TensorFlow, PyTorch) для роботи на чіпах Ambarella.

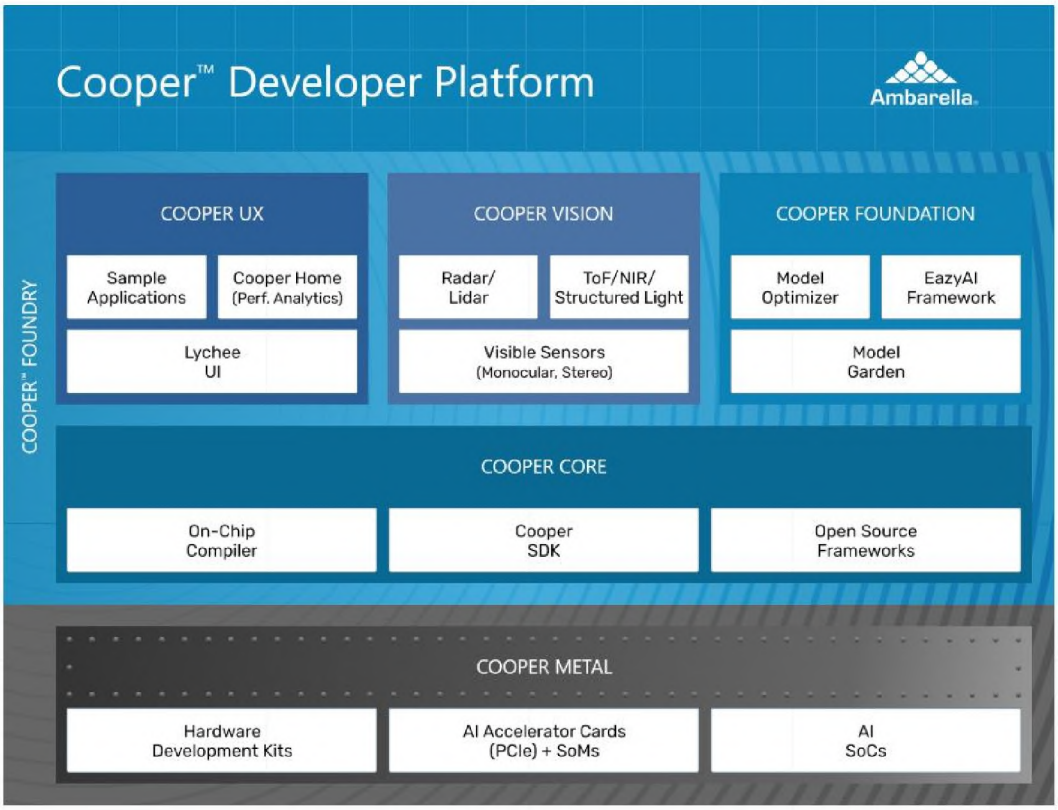

Рисунок 2.9 – ПЗ Cooper для Ambarella

Найбільш продуктивним є система на кристалі N1-655 (рис. 2.10). Вона новим доповненням до сімейства SoC N1 і забезпечує високу продуктивність ШІ у перерахунку на 1 Ват споживаної енергії для обчислень нейронних мереж, має вдосконалений процесор зображень, 8 процесорів Arm® Cortex-A78AE® в одному SoC. В якості ключових особливостей розробником заявляються такі положення. Обробка на основі нейронної мережі для підтримки мультимодальних LLM разом із виявленням зображень, класифікацією, відстеженням та ін. Нейронний векторний процесор (NVP) з найкращою в галузі продуктивністю ШІ. N1-655 є оптимальною платформою

для впровадження розумних міст, промислової автоматизації, промислової та споживчої робототехніки, охорони здоров'я та центрів ШІ.

Рисунок 2.10 – Система N1-655 на кристалі (SoC) від к. Ambarella

Sophron – це виробник AI прискорювачів, який став дуже популярним останнім часом, наприклад, разом з ним постачаються такі пристрої: MAIX-CAM, Milk-V (рис. 2.11), SiFive, hw100k, reCamera та ін. Він дешевий і досить швидкий. Але не має жодної документації, більшість прикладів не працює, дуже багато коду на C++. Це швидше «виріб для невеликих партій».

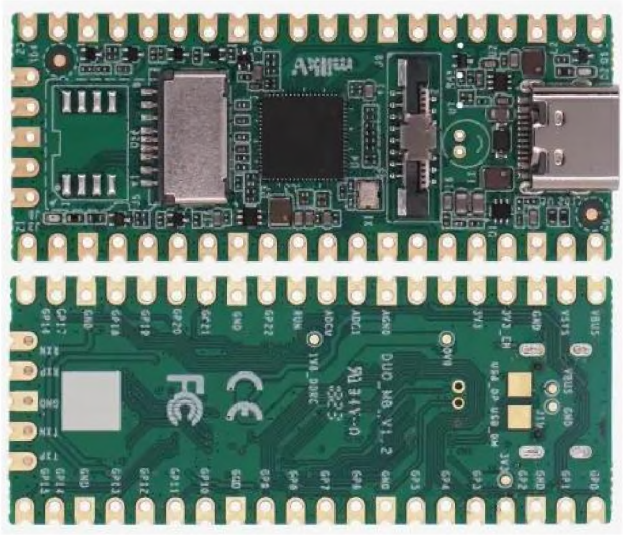

Рисунок 2.11 – Milk-V Duo

Швидше за все вимагатиме багато підтримки, якщо партія буде великою. При цьому достатньо буде обмежена пам'ять. Жодних LLM та

сучасних мереж. Більшість вендорів китайські. Швидше за все, не для всього можна використовувати в Європі та США.

Існує дуже багато виробників, які роблять невеликі непогані плати орієнтовані на ентузіастів, які не знають що таке Computer Vision, але хочуть зробити очі своїм пристроям. Часто тут навіть немає ніяких докладних інструкцій, лише «дружній інтерфейс», наприклад, Grove Vision Ai, UnitV2. Якщо торкатися теми мікроконтролерів, то можна звернути увагу ESP32. При цьому треба розуміти, що зараз дуже багато виробників цим займаються (GAP9, Syntiant (NDP101, NDP120 (Arduino), Analog Devices, Synaptics, SiLabs, Innatera, Nordic Semiconductor). Усі їхні продукти відрізняються таким. Майже ніколи немає нормальної ОС. Це або C/C++ або MicroPython технологія, або через Edge Impuls, якщо він підтримує цю плату. Зазвичай, кількість працюючих нейронних мереж для кожної платформи дуже мала, буквально 1 або 2 мережі. Зазвичай, швидкість дуже низька. Доводиться використовувати оптимізовані моделі. У більшості випадків плати вимагають квантизації на рівень int8. При цьому виробники мають дуже середню документацію. Багато плат вимагають дуже багато коду C++. Edge Impulse часто вирішує проблеми. Але треба розуміти, що зможе підтримувати плату в повному обсязі. І той зручний інтерфейс, який він надає, може частково обмежувати і досяжні можливості пристроїв ШІ та можливі особливості плати. Є багато інших плат:

– MAIX-IV (AX650N, Axera-tech) – очікування розширення 3-ої версії;

– AMD Kria (це FPGA) – наявні відгуки в Інтерент свідчить складність такого підходу;

– ARM Ethos (U55, U65) – ARM теж вирішив стати на шлях «робити ML» (поки що випускають дуже слабкі прискорювачі);

– Renesas – дуже багато плат (начебто досить великий вендор), але мало наявної інформації;

– MAIX-III – досить середня плата;

– Beagle Board від Texas Instruments – досить популярна плата;

– Sony IMX500 – Raspberry Pi презентувала камеру, в якій обчислювач поєднаний з оптичною матрицею (OAK-D але на одному чіпі), однак має складнощі з чіпом;

– Kneron – дуже багато маркетингу про неї.

Ще можна вказати Syntiant, BrainChip, MemryX, Horizon X3M, Kendryte k510/k230.

2.3 Інтеграція SBC Raspberry Pi та Hailo-8 для застосувань Edge AI

На основі проведених досліджень в роботі запропоновано для реалізації рішень Edge AI використовувати спільно з акселератором ШІ Hailo-8 SBC Raspberry Pi 5. Він побудований на 64-розрядному процесорі Broadcom BCM2712 ARM (4 ядра Cortex-A76) з графікою на базі VideoCore VII з підтримкою OpenGL ES 3.1 та Vulkan 1. Остання версія Raspberry Pi 5 містить 16 ГБ (рис. 2.12). Це забезпечує необхідні обчислювальні потужності для виконання алгоритмів ШІ безпосередньо біля джерел отримання даних, таких як датчики, камери, сенсори IoT.

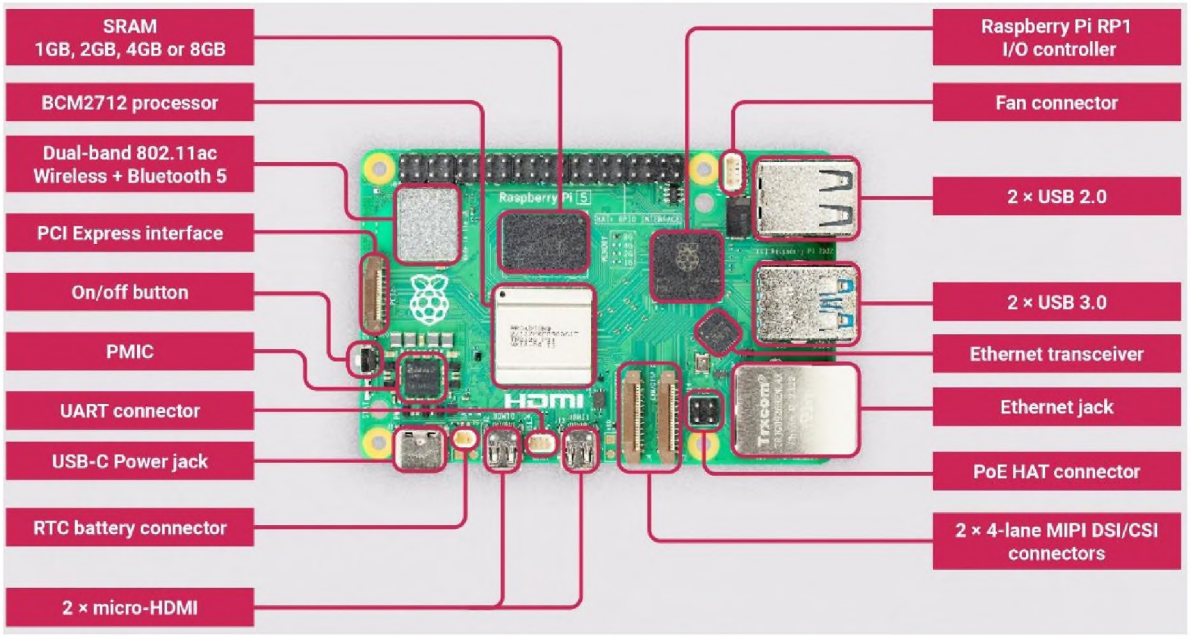

Рисунок 2.12 – SBC Raspberry Pi 5

Це дозволяє проводити локальну попередню обробку та аналіз даних, а також приймати рішення у реальному часі, зменшуючи затримку, що є критичним для застосувань у промисловості, медицині та автономних системах. Raspberry Pi має екосистему програмних засобів (TensorFlow Lite, PyTorch, OpenCV, Edge Impulse тощо), які дозволяють швидко розгортати попередньо натреновані моделі ML, глибинних нейронних мереж та систем комп'ютерного зору. Завдяки цьому мікрокомп'ютер може виконувати такі складні завдання, як розпізнавання об'єктів, класифікація зображень, детектування аномалій та обробка потокових даних. Перевагою Raspberry Pi є низьке енергоспоживання, що дозволяє застосовувати його в автономних системах, які живляться від акумуляторів або сонячних батарей. Це важливо для мобільних та віддалених застосувань, таких як екологічний моніторинг, моніторинг стану інфраструктури або робототехнічні рішення. Невеликі розміри Raspberry Pi спрощують інтеграцію пристрою в системи, які мають обмеження щодо простору або ваги. Такі компактні комп'ютери часто використовують у системах безпеки, дронах, автомобілях, мобільних платформах, що дозволяє суттєво розширити сферу застосування Edge AI-рішень. Raspberry Pi завдяки своїй доступності та широкій екосистемі активно використовується для створення прототипів нових Edge AI-рішень. Це дозволяє швидко перевіряти концепції, розробляти та оптимізувати програмне забезпечення, а також апробувати різні архітектури нейронних мереж перед переходом до промислових чи комерційних реалізацій. Важливим аспектом є можливість підключення Raspberry Pi до спеціалізованих AI-акселераторів (наприклад, Google Coral TPU, Intel Neural Compute Stick), що значно підвищує швидкість обробки нейронних мереж та дозволяє ефективніше виконувати складні моделі ML на периферії. На даний час, в якості практичного застосування Raspberry Pi в проєктах Edge AI варто вказати декілька прикладів. Розпізнавання образів та CV – Raspberry Pi широко використовується у проектах, які потребують візуального розпізнавання об'єктів, наприклад, для розумних камер безпеки, контролю

якості на виробництві, підрахунку трафіку, ідентифікації людей. Моніторинг середовища та прогнозна аналітика – SBC використовуються для моніторингу якості повітря, води, стану сільськогосподарських угідь, де завдяки застосуванню AI можна здійснювати передбачення, виявлення аномалій, раннє сповіщення про ситуації. Автономні роботи та безпілотні апарати – Raspberry Pi виступає центральним процесором у невеликих роботах, дронах, де потрібна локальна навігація, прийняття автономних рішень та аналіз навколишнього середовища.

В свою чергу, акселератор ШІ Hailo-8 може використовуватися як в автономному режимі, так і як співпроцесор. Більшість команд-розробників стикуватимуть цей чіп з більш потужним хост-процесором для вирішення інших завдань, але в теорії, також можливо використовувати Hailo-8 сам по собі. Hailo – ізраїльський стартап, заснований у 2017 р. зараз компанія зосереджена на комерційних проектах. Вона надає оціночну плату Hailo-8 з інтерфейсами PCIe, Gigabit Ethernet, аудіо, портами USB, інтерфейсами I_2C та UART, GPIO, а також двома інтерфейсами камери MIPI CSI (рис. 2.13).

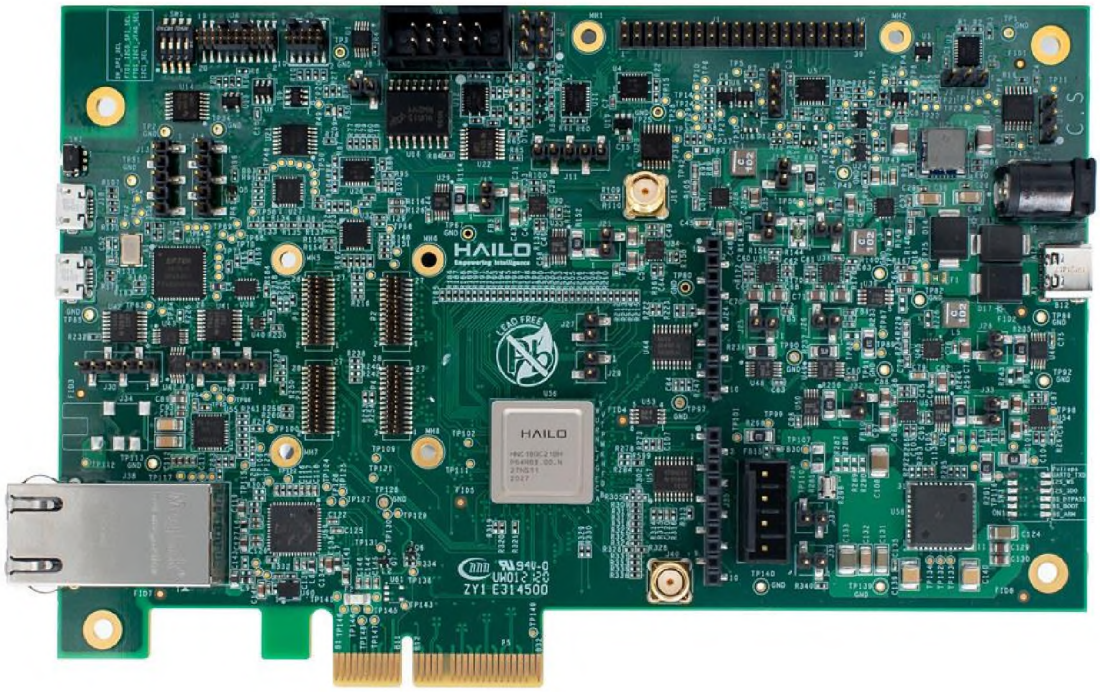

Рисунок 2.13 – Інструмент для розробки рішень з Hailo-8

Пристрій призначений для підключення до комп'ютера, на якому можна використовувати інструменти розробки для навчання TensorFlow або ONNX, перш ніж отримати модель через Hailo SDK для перетворення даних і розподілу ресурсів (рис. 2.14).

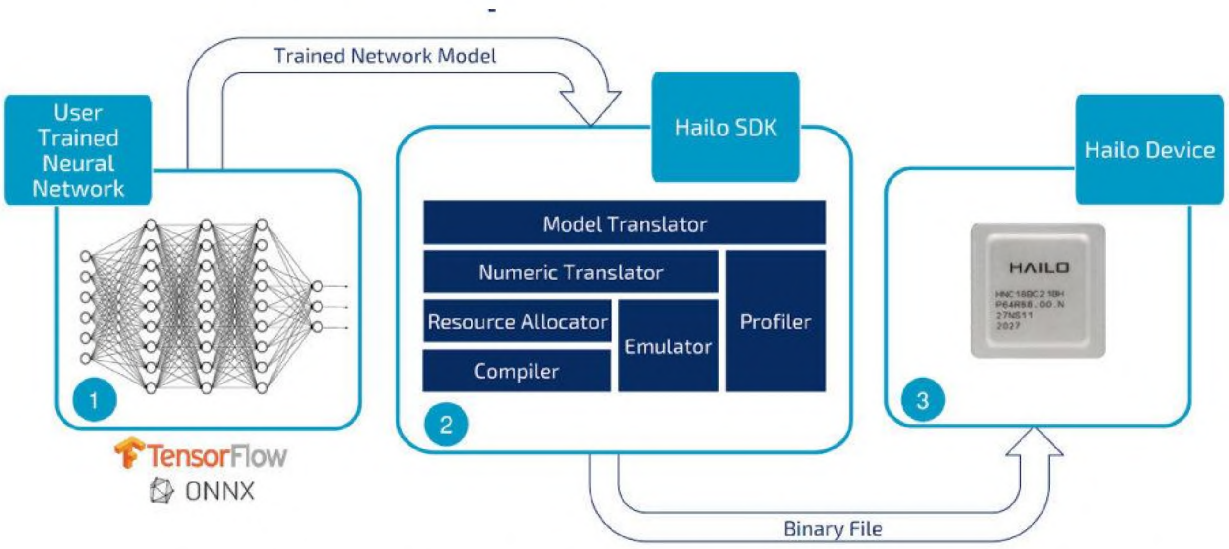

Рисунок 2.14 – Інструментарій для розробки моделей на основі Hailo

Таким чином, SBC сімейства Raspberry Pi є важливими компонентами рішень Edge AI, забезпечуючи баланс між доступністю, функціональністю та ефективністю, що дозволяє широко використовувати ШІ на периферії мережі у різних прикладних галузях.

РОЗДІЛ 3
РЕКОМЕНДАЦІЇ ЩОДО ВИКОРИСТАННЯ АКСЕЛЕРАТОРУ ШІ HAILO-8

3.1 Інсталяція акселератору ШІ Hailo-8

Згідно п. 2.2, акселератор ШІ Hailo-8 Acce A (SKU: 27841) – це енергоефективний (типове споживання електроенергії складає 2,5 Вт) процесор для виконання завдань ШІ, розроблений із фокусом на вбудовані пристрої (рис. 3.1) та масштабований, що забезпечує одночасну обробку кількох потоків і кількох моделей.

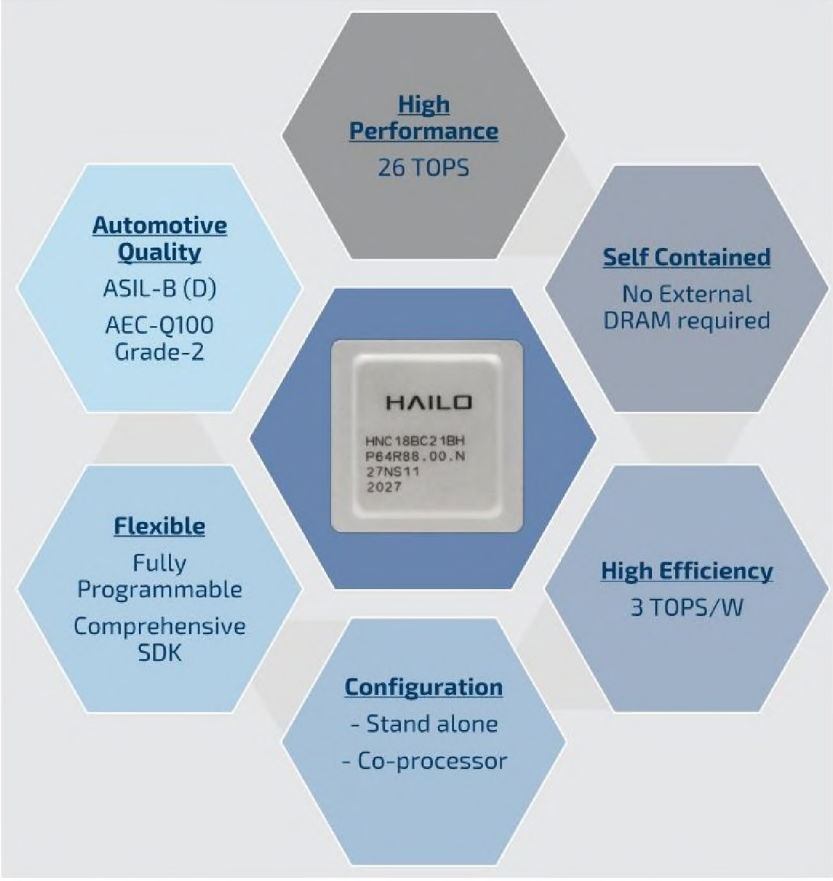

Рисунок 3.1 – Архитектура Hailo-8

Процесор надає високу продуктивність для обробки нейронних мереж, забезпечуючи понад 26 TOPS при дуже низькому енергоспоживанні. Hailo-8

підтримує широкий спектр нейронних мереж, і використовується в автономних транспортних засобах, інтелектуальних камерах, медичних пристроях та інших промислових рішеннях. Чіпу вдалося отримати додаткову продуктивність та ефективність завдяки «запатентованій новій архітектурі потоку даних, керованої структурою замість звичайної архітектури. Одна з ключових причин підвищення продуктивності полягає в тому, що оперативна пам'ять є автономною без необхідності зовнішньої DRAM, як в інших рішеннях – рис. 3.2.

Рисунок 3.2 – Розподіл обчислювальних компонентів для вирішення завдання

Це значно знижує затримку та енергоспоживання. Чіп підтримує фреймворки TensorFlow, TensorFlow Lite, ONNX, Keras, Pytorch з

використанням ОС Linux і Windows. Усередині чіп Hailo-8 складається з 3-ох типів блоків – управління, пам'яті та обчислень, які призначаються різним рівням нейронної мережі, як показано на анімації нижче. Вся обробка відбувається усередині чіпа. При цьому, не всі блоки призначені, і це нормально, оскільки кожне робоче навантаження ШІ буде використовувати лише частину акселератору ШІ. Ось чому часто топове число, яке рекламується компаніями, переважно є «маркетинговим ходом». Діаграма також показує, що Hailo-8 може використовуватися як в автономному режимі, так і в якості співпроцесору. Більшість компаній стикуватимуть чіп з більш потужним хост-процесором для вирішення інших завдань, але в теорії, також можливо використовувати Hailo-8 сам по собі. При цьому забезпечується доступ до широкої бібліотеки попередньо навчених моделей нейронних мереж, готових до розгортання та оптимізованих для роботи з AI Kit.

На основі проведених досліджень був вибраний варіант реалізації акселератору ШІ Hailo-8, що інтегрований за платою: AI HAT+ (рис. 3.3) потужністю 26 TOPS.

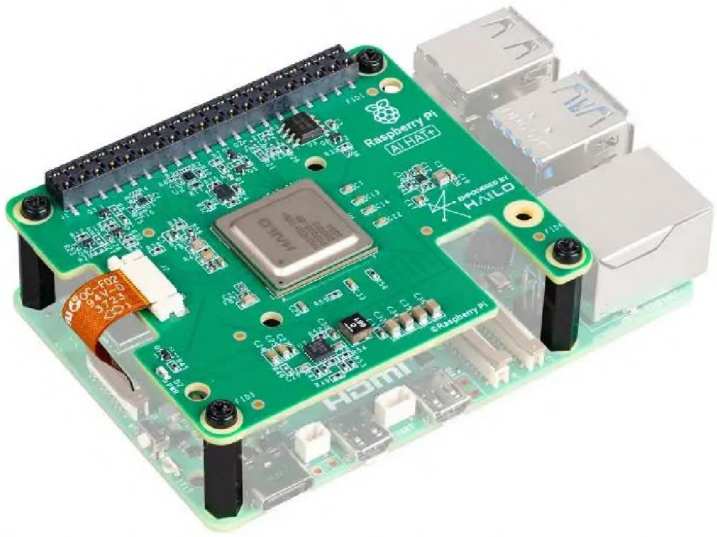

Рисунок 3.3 – Плата AI HAT+ з Hailo-8 потужністю 26 TOPS

Є можливість одночасного запуску кількох нейронних мереж на одній або двох камерах. Наявний API інтегрований з фреймворком GStreamer та

власними програмами Python і C/C++, що також дозволяє використовувати його у випадках, не пов'язаних з камерою, наприклад, для аналізу попередньо записаних відеофайлів. AI HAT+ взаємодіє за допомогою інтерфейсу PCIe Gen 3 Raspberry Pi 5. Якщо на хості Raspberry Pi 5 запущено оновлений образ ОС Raspberry Pi, то він автоматично виявляє вбудований акселератор Hailo 8 та робить NPU доступним для обчислювальних завдань ШІ. Вбудовані програми камери rpicam-apps в ОС Raspberry Pi нативно підтримують модуль ШІ, автоматично використовуючи NPU для завдань постобробки, який повністю інтегрований у програмний стек камери Raspberry Pi. Термін виробництва розглянутого варіанту акселератору залишатиметься у виробництві щонайменше до січня 2030 р.

Таким чином, для найкращої продуктивності рекомендується використовувати AI HAT+ з активним охолоджувачем Raspberry Pi (рис. 3.4) – встановлюється перед AI HAT+.

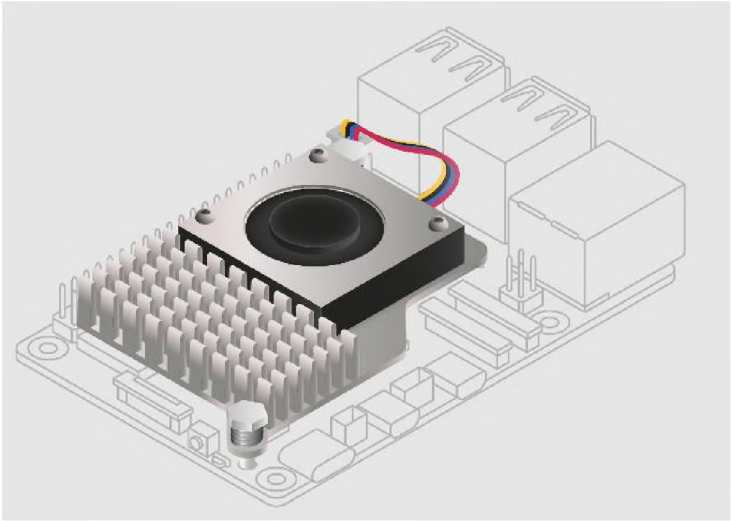

Рисунок 3.4 – Активний охолоджувач Raspberry

За допомогою кріплення встановлюється AI HAT+ (рис. 3.5) та підключається до SBC стрічковим кабелем. Піднімають тримач стрічкового кабелю з обох боків, а потім вставляють кабель мідними контактними точками догори (рис. 3.6). Коли стрічковий кабель повністю та рівномірно вставлений у порт, посуньте тримач кабелю вниз з обох боків.

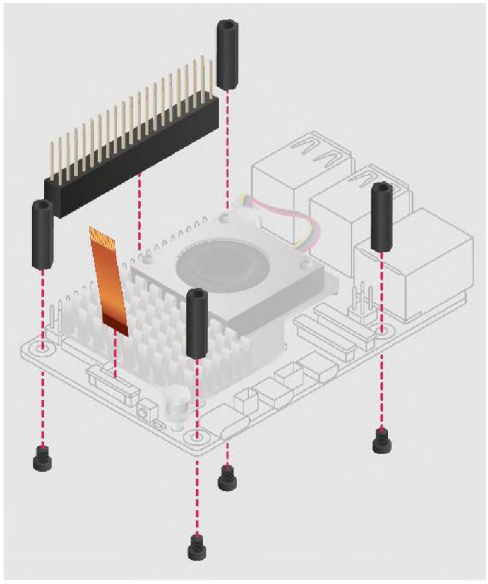

Рисунок 3.5 – Інсталяція AI HAT+

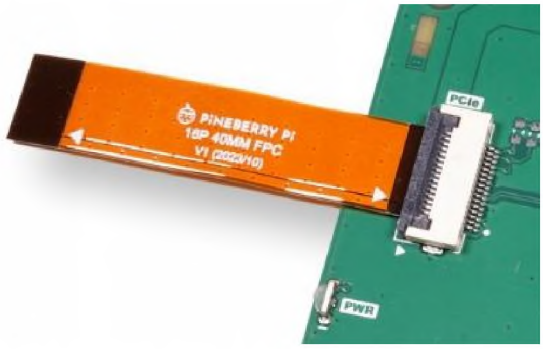

Рисунок 3.6 – Орієнтація стрічкового кабелю інтерфейсу PCIe

Надалі підключають живлення до Raspberry Pi з попередньо встановленою ОС Raspberry Pi. Її останні релізи автоматично визначають плату AI HAT+ з акселератором ШІ Hailo-8. Надалі виконується налаштування ПЗ.

3.2 Конфігурація Edge AI для роботи з акселератором ШІ Hailo-8

На початку треба увімкнути інтерфейс PCIe Gen 3.0. За замовчуванням він не включений. якщо він не підключений до плати HAT+. Щоб увімкнути з'єднувач до плати AI HAT+, додають до файлу конфігурації

/boot/firmware/config.txt/ рядок tdtparam = pciex1. За замовчуванням Raspberry Pi 5 використовує швидкість Gen 2.0 (5 GT/s). Треба виконати один з наведених нижче підходів, щоб форсувати швидкість до Gen 3.0 (8 GT/s): у файл конфігурації внести такий варіант рядка: param = pciex1_gen = 3. Щоб зміни в конфігурації вступили в силу перезавантажують SBC за допомогою команди sudo reboot. Ще один варіант налаштування передбачає використання RSPBERRY Pi Configuration CLI:

– введення команди $ sudo raspi – config;
– вибрати вкладку Advanced Options;
– вибрати вкладку PCIe Speed;
– вибрати режим PCIe Gen 3 → yes;
– щоб вийти натиснути Finish;
– виконати перезавантаження SBC.

Цей крок необов'язковий, але настійно рекомендується для досягнення найкращої продуктивності з NPU Hailo-8. Далі встановлюють залежності, що необхідні для використання NPU. Запускають команду з вікна терміналу:

$ sudo apt install hailo – all

Це встановлює такі залежності:

– драйвер і прошивку пристрою Hailo kernel;
– ПЗ для проміжного софта HailoRT;
– основні бібліотеки постобробки Hailo Tappas;
– етапи демонстрації ПЗ для постобробки Hailorpicam-apps.

Після цього перезавантажують SBC. Щоб переконатися, що все працює правильно, виконують команду: $ hailortcli fw – control identify. Інформація про успішну інсталяцію відповідає рис. 3.7. Крім того, можна запустити перевірку журналів ядра, які повинні вивести дані, подібні до наступного варіанту: dmesg | grep – i hailo. У пакеті програм rpicam-apps для роботи з камерою реалізовано фреймворк постобробки [22]. За його допомогою можна виконати типові завдання ШІ постобробки для CV з використанням NPU.

```
Executing on device: 0000:01:00.0
Identifying board
Control Protocol Version: 2
Firmware Version: 4.17.0 (release,app,extended context switch buffer)
Logger Version: 0
Board Name: Hailo-8
Device Architecture: HAILO8
Serial Number: HLDDLBB234500054
Part Number: HM21LB1C2LAE
Product Name: HAILO-8 AI ACC MODULE EXT TMP
```

Рисунок 3.7 – Інформація про інсталяцію Hailo-8

Для цього можна використати бібліотеку rpicam-hello, яка за замовчуванням відображає вікно попереднього перегляду. Однак можна використовувати інші варіанти rpicam-apps, зокрема rpicam-vid та rpicam-still. При цьому, може доведеться додати або змінити деякі параметри командного рядка, щоб зробити команди сумісними з альтернативними програмами. Щоб переконатися, що камера працює правильно, вводять команду:

$ rpicam – hello – t 10s

Це запустить камеру та покаже вікно попереднього перегляду протягом 10 секунд. Після того, як переконалися, що все встановлено правильно, перевіряємо Edge AI. З початку вводимо команду для встановлення останнього оновлення rpicam-apps:

$ sudo apt update && sudo apt install rpicam – apps

Одним із завдань CV є Object detection – відображає обмежувальні рамки навколо об'єктів, виявлених нейронною мережею. Щоб вимкнути видошукач, використовуйте прапорець –n. Щоб повернути чисто текстовий вивід з описом виявлених об'єктів треба додати відповідну опцію – v 2 у файлі:

$ rpicam – hello – t 0 – –post – process – file /usr/share/rpi – camera – assets/hailo_yolov6_inference. json

Лістинг програмного коду для Object detection наведено у Додатку А. Крім того, можна спробувати іншу модель з різними компромісами в продуктивності та ефективності. Наприклад, щоб запустити модель yolov8m (рис. 3.8), вводять наступну команду:

$ rpicam – hello – t 0 – –post – process – file /usr/share/rpi – camera – assets/ hailo_yolov8m_inference. json

ПЗ також підтримує всі моделі, скомпільовані за допомогою постпроцесу HailoRT NMS. Рівень немаксимального придушення (NMS) Hailo інтегрований у файл HEF, що дозволяє будь-якій мережі виявлення, скомпільованій за допомогою NMS, функціонувати з тим самим постпроцесом. Як видно з прикладу, всі «особи» відстежуються.

Рисунок 3.8 – Object detection за допомогою акселератору ШІ Hailo-8

Щоб запустити демоверсію з моделлю YoloX, записують команду:

$ rpicam – hello – t 0 – –post – process – file /usr/share/rpi – camera – assets/ hailo_yolox_inference.json

Наступним завданням CV є сегментація зображень (ПЗ виконує виявлення об'єктів і сегментує об'єкт, малюючи колірну маску на зображенні видошукача – рис. 3.9).

Для цього вводять команду на SBC Raspberry Pi:

$ rpicam – hello – t 0 – –post – process – file /usr/share/rpi – camera – assets/ hailo_yolov5_segmentation.json – –framerate 20

Також цікавим є оцінка пози – вбудований приклад виконує 17-точкову оцінку пози людини, малюючи лінії, що з'єднують виявлені точки (рис. 3.10). Щоб переглянути відповідну демонстрацію виконують команду:

$ rpicam – hello – t 0 – –post – process – file /usr/share/rpi – camera – assets/ hailo_yolov8_pose.json

Рисунок 3.9 – Instance segmentation за допомогою акселератору ШІ Hailo-8

Рисунок 3.10 – Pose estimation за допомогою акселератору ШІ Hailo-8

При використанні акселератору ШІ Hailo-8 варто мати на увазі, що плата AI HAT+ не функціонує, якщо є невідповідність версій між програмними пакетами Hailo і драйверами пристроїв. Крім того, інструменти нейронної мережі Hailo можуть вимагати певної версії для згенерованих файлів моделей. Якщо потрібна конкретна версія, виконують наступні кроки, щоб встановити

відповідні версії всіх залежностей. Якщо є наявний будь-який з відповідних пакетів, може знадобитися їх оновлення через apt-mark:

$ sudo apt – mark unhold hailo – tappas – core hailort hailo – dkms

На даний час, актуальними є такі версії програмних пакетів: v4.19, 4.18 а також 4.17. Щоб встановити версію 4.19 інструментів нейронної мережі Hailo, вводять команди:

sudo apt install hailo – tappas – core = 3.30.0 – 1 hailort = 4.19.0 – 3 hailo – dkms
= 4.19.0 – 1 python3 – hailort = 4.19.0 – 2

$ sudo apt – mark hold hailo – tappas – core hailort hailo – dkms python3 – hailort

Hailo також створив набір демо-версій [23], які можна запустити на Raspberry Pi 5. Крім цього, можете знайти великий модельний зоопарк Hailo, який містить велику кількість нейронних мереж, у репозиторії [24].

3.3 Порівняльна оцінка продуктивності рішень Edge AI на основі акселератора ШІ Hailo-8

Згідно [25], для оцінки властивостей акселератора ШІ Hailo-8 доцільно його порівняти з акселераторами ШІ. Враховуючі кількість TOPS, продуктивність Hailo-8 набагато ближче до такого рішення, як NVIDIA AGX Xavier [26] (рис. 3.11).

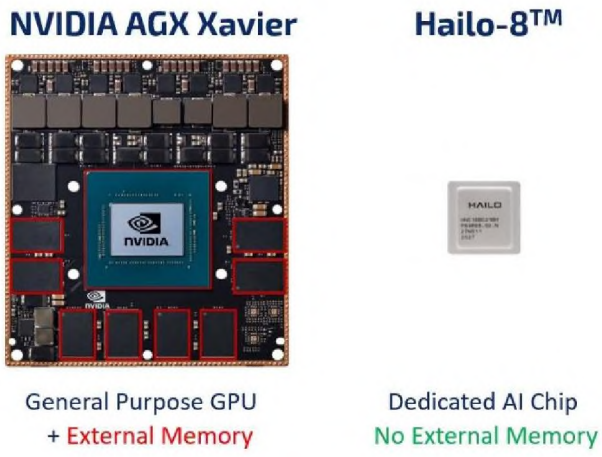

Рисунок 3.11 – Акселератори ШІ NVIDIA AGX Xavier і Hailo-8

Як можна бачити, форм-фактор Hailo-8 набагато менше і це може бути суттєвим для більшості проєктів EDGE AI (наприклад, якщо використовують кілька акселераторів ШІ в одному додатку). Це не означає, що обидва вони однакові, оскільки NVIDIA AGX Xavier є більш гнучким, (також можна проводити навчання на цій платформі), в той час як Hailo-8 призначений тільки для малопотужного виводу, який він робить з 20-кратною енергоефективністю. На рис. 3.12 показано продуктивність Hailo-8 порівняно з NVIDIA Jetson Nano [27], Jetson TX2 [28] та Jetson Xavier NX з використанням Resnet-v1-50, MobileNet-v1-SSD та Yolo-v3 Tiny. У цих 3-ох тестах Hailo-8 трохи швидше, ніж Jetson Xavier NX, але буде набагато ефективнішим, оскільки споживана потужність NVIDIA Jetson Xavier NX становить до 10 або 15 Вт залежно від режиму, що використовується.

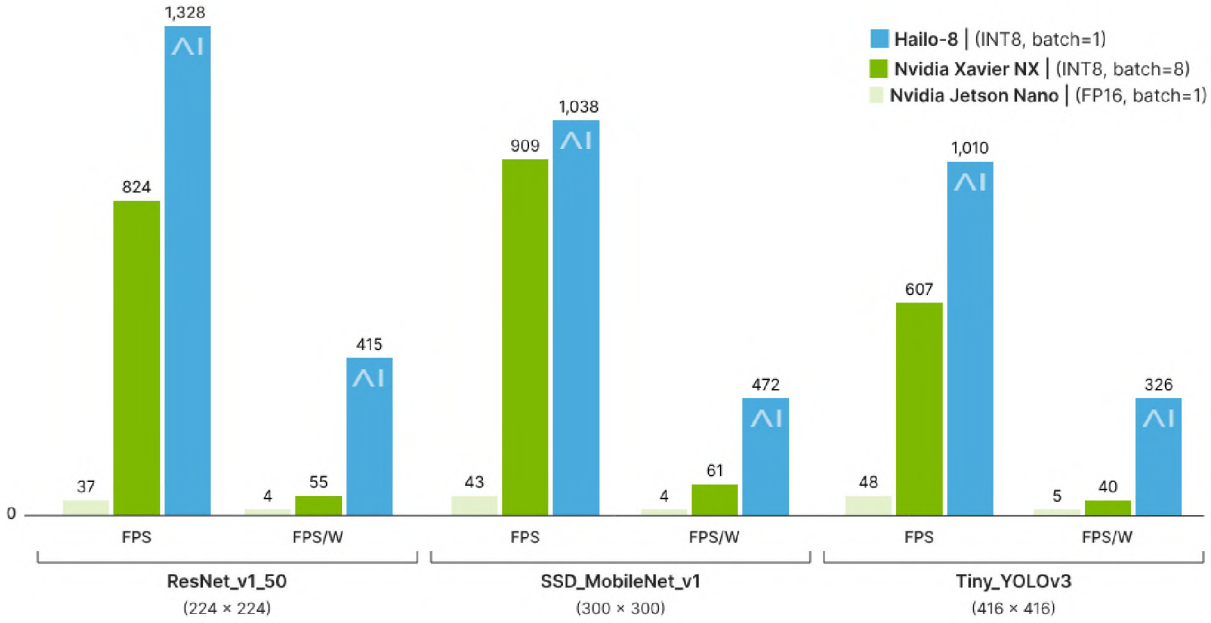

Рисунок 3.12 – Діаграма продуктивності акселераторів ШІ

Треба звернути увагу, що є примітка про розмір батчу. Це пов'язано з тим, що Jetson Xavier NX використовує як GPU NVIDIA Volta, так і механізми NVDLA, а GPU краще працюють із високопаралельними завданнями. Тому розмір батча налаштовується та впливає як на продуктивність, так і на ефективність. Компанії, що займаються розробкою мікросхем ШІ, люблять

вказувати цифри TOPS, щоб показати максимальну теоретичну продуктивність своїх мікросхем. Але на практиці це лише для маркетингу. Наприклад, Hailo-8 рекламується з 26 TOPS, тоді як Google Edge TPU підтримує до 4 TOPS. Це в 6 разів вище за продуктивність, але при запуску реального тесту – Hailo в середньому в 13 разів швидше Edge TPU через архітектурні відмінності (згідно п.3.1, ніяке робоче навантаження ШІ не використовуватиме всі ресурси чіпа до задекларованих TOPS). Крім наведених результатів варто також звернути увагу на порівняльну оцінку акселератору ШІ Hailo-8L на Raspberry Pi 5 та Raspberry Pi Compute Module 4 (рис. 3.13 та 3.14) для завдань Object detection та Pose estimation [29].

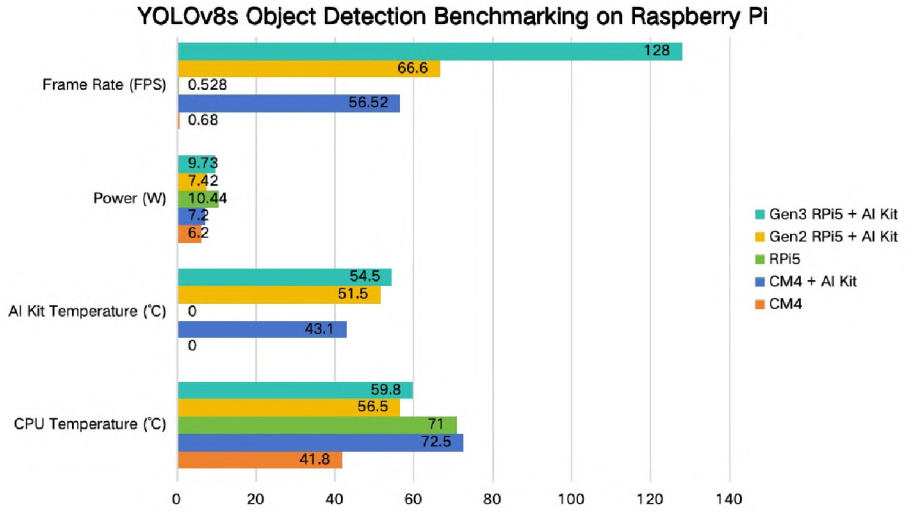

Рисунок 3.13 – Порівняльна оцінка для завдання Object detection

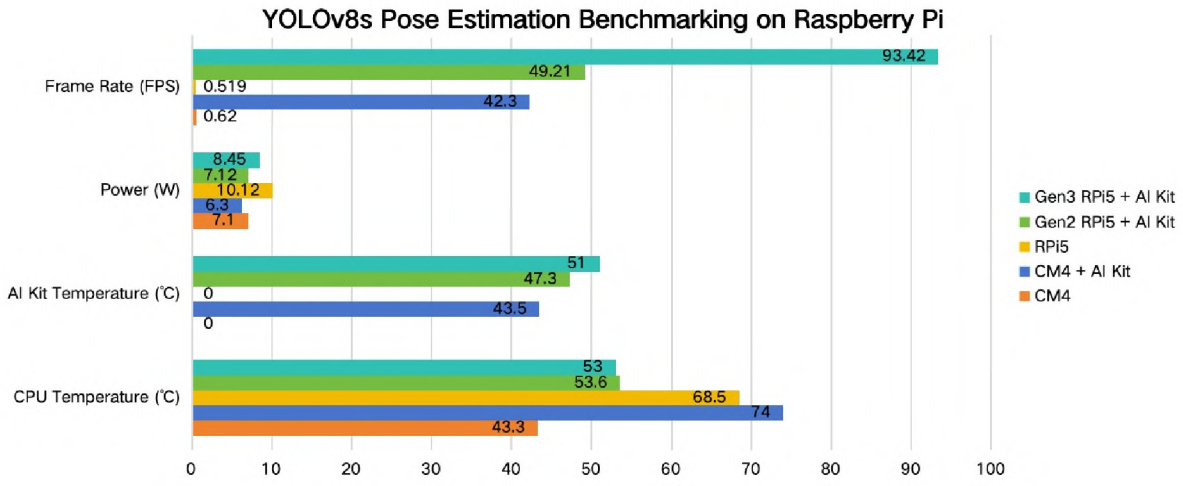

Рисунок 3.14 – Порівняльна оцінка для завдання Pose estimation

При цьому використовувалась YOLOv8s у версії int8 з вхідним розміром зображення 640×640, розмір батча дорівнює 8 для відео при 240 FPS. Ефект прискорення Hailo-8 дуже помітний. Частота кадрів Pi5 під PCIe Gen3 вдвічі вища, ніж під PCIe Gen2. Продуктивність CM4 після розгону несподівано хороша. Показники енергоспоживання Pi5 і CM4 відмінні при прискоренні Hailo-8L.

3.4 Техніко-економічне обґрунтування прийнятих рішень

Акселератори ШІ розроблені для виконання завдань високої продуктивності та глибокого навчання, а також для нейронної мережі. На даний час, коли розмір моделей ШІ зростає та стає складнішим, слід звернути увагу на ефективне використання енергії (рис. 3.15) [30].

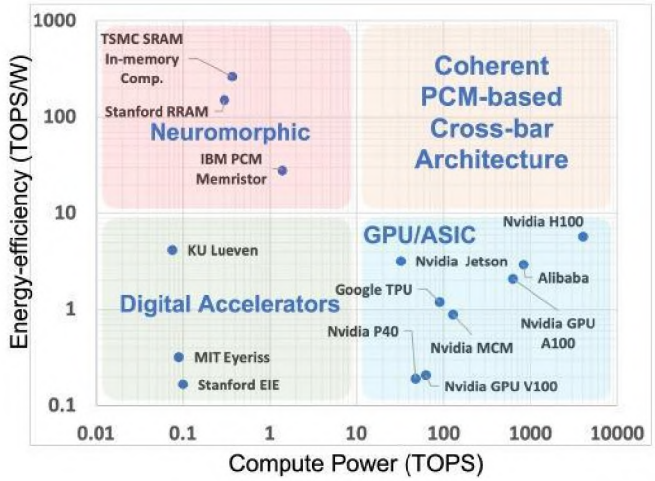

Рисунок 3.15 – Порівняння найсучасніших процесорів AI/ML [31]

У той же час, непропорційне споживання електроенергії може сильно вплинути на експлуатаційні витрати через велике розсіювання тепла і, таким чином, знизити продуктивність і термін служби обладнання. Щоб вирішити цю проблему, необхідно використовувати ефективні механізми охолодження з розширеними системами управління температурою та інноваційними

технологіями проектування чіпів. Галузь також вивчає малопотужні архітектури та масштабування напруги для підвищення енергоефективності за однакових умов продуктивності.

Ще одна завдання – це реалізація масштабованості. Типові робочі навантаження ШІ сьогодні стають все більш інтенсивними, часто вимагаючи великої кількості інформації в режимі реального часу. При цьому виникає колізія – продуктивність акселератора ШІ має зростати разом з робочими навантаженнями. Однак, у таких випадках, звичайні методи масштабування, тобто збільшення кількості ядер або тактової частоти, як правило, означають велике енергоспоживання та вагоме зниження продуктивності акселератора ШІ. Це стає ще більш критичним із збільшенням прискорення. Загалом, здатність швидкої та ефективної передачі даних між пам'яттю та обчислювальними блоками буде ключовою для масштабування продуктивності зі збільшенням робочого навантаження. Саме в такому контексті інновації в технологіях взаємозв'язку та ієрархіях пам'яті мають значення для пом'якшення таких вузьких місць [31].

Оскільки важко розробити нейронні мережі, які були б швидкими та енергоефективними, дослідники шукають акселератори, що відмінні від GPU та CPU. Проблеми з обробкою даних виникають через високе споживання енергії та повільний час доступу до зовнішньої пам'яті. Зменшення накладних витрат шляхом розподілу пам'яті по системі або модифікації пам'яті для більшої шини даних є двома можливими рішеннями. Керуючи кількома потоками даних за такт, паралельний доступ покращує продуктивність системи та ефективність використання ресурсів.

Все, від розробки до розгортання акселератори ШІ, є досить високовартісним, якщо цей термін охоплює виготовлення апаратного забезпечення на замовлення, а також потужність та інфраструктуру, які потрібні таким системам (рис. 3.16). Однією з ключових проблем, які існують сьогодні, є те, як досягти такого оптимального балансу між вартістю та продуктивністю. З іншого боку, спеціальні акселератори ШІ, такі як ASIC або

FPGA, мають набагато кращу продуктивність у конкретних завданнях. Таке рішення, зазвичай, передбачає набагато більші початкові інвестиції порівняно з більш загальними альтернативами, такими як GPU. Другий чинник – це експлуатаційні витрати. Високопродуктивні акселератори ШІ означають витрати на споживання енергії та обслуговування. Варто зазначити, що будь-яке підвищення продуктивності, пов'язане зі збільшенням витрат на розробку або розгортання, має бути збалансованим.

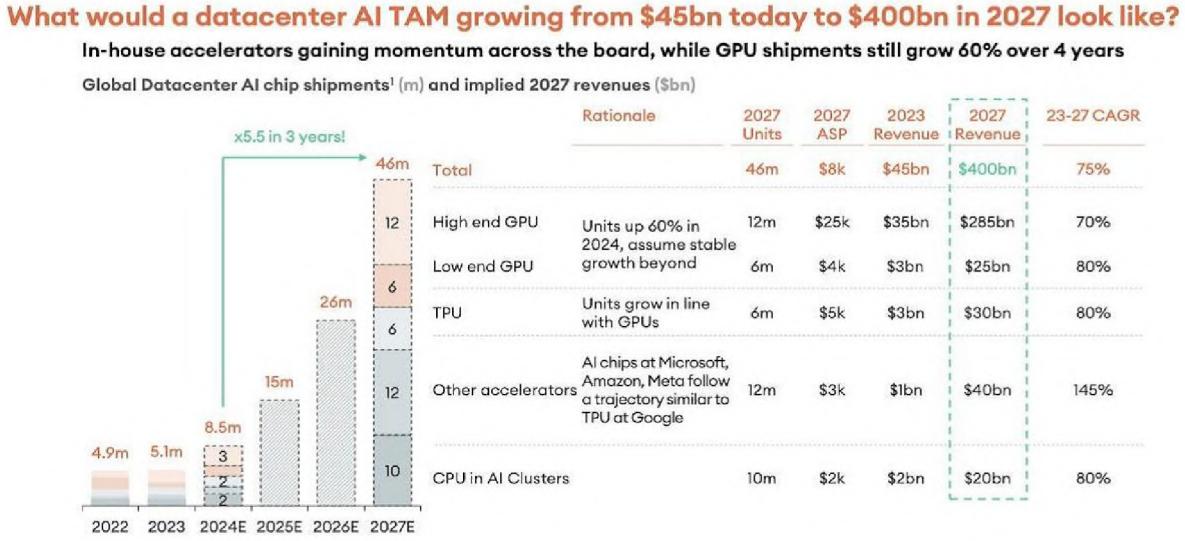

Рисунок 3.16 – Зростання вартості акселераторів ШІ

У довгостроковій перспективі дизайн акселераторів ШІ прямує до кількох простих рівнів у гонитві за продуктивністю, адаптивністю та стійкістю. Цікавою буде тема енергоефективності систем ШІ, оскільки вони мають великі розміри та складність, що стало однією з найбільш тривожних проблем щодо навчання та споживання енергії при експлуатації. Тому очікується, що малопотужні конструкції з такими інноваціями, як зниження напруги та оптимізована архітектура для конкретних застосувань ШІ, будуть дуже затребуваними. Ще одна пов'язана сфера – це гетерогенні обчислення, які об'єднують різні типи процесорів, наприклад GPU, TPU, FPGA та спеціальні акселератори ШІ, на одній платформі. Це забезпечує більшу гнучкість у роботі з різними типами робочих навантажень ШІ, оскільки кожен тип процесора

можна оптимізувати щодо конкретних завдань, які він повинен виконувати. Наприклад, GPU особливо добре працюють для розпаралелюваних завдань, тоді як TPU можуть прискорити матричні операції, що лежать в основі нейронних мереж, а спеціалізовані мікросхеми ШІ оптимізовані для виконання виводів глибокого навчання [32].

Очікується, що майбутні інновації в 3D-стекуванні та підходах на основі мікросхем змінять вигляд архітектур прискорювачів. Більше рівнів процесорів, пам'яті та інших компонентів, розміщених один над одним, можливі в 3D-стекуванні. Усе це разом підвищує загальну продуктивність, зберігаючи низьку затримку. Мікросхеми дозволяють розробникам систем бути модульними у своєму підході, де кілька компонентів можуть бути розміщені на одному кристалі, щоб збалансувати специфіку та вартість [32].

Іншим важливим напрямком майбутнього є гнучкість щодо акселераторів ШІ. Оскільки алгоритми ШІ постійно змінюються, їм потрібні прискорювачі, які є гнучкими в обробці численних видів моделей та фреймворків. Програмовані акселератори, які можна реконфігурувати з різними завданнями ШІ, продовжуватимуть відігравати важливу роль у розвитку обладнання в цьому динамічному світі через ПЗ, яке постійно змінюється. Вимоги до проектування акселераторів ШІ відповідно до проблем сталого розвитку нарешті зростають, оскільки, оскільки системи ШІ вимагають більше потужності та складних ресурсів. Дослідники зараз прагнуть створити більш екологічні процесори з використанням нових матеріалів і передового виробництва. Разом, робота над екологічно чистим обчислювальним обладнанням майбутнього в поєднанні з передовими технологіями охолодження та енергоефективними архітектурами може зменшити витрати на навколишнє середовище для обчислень ШІ.

Для реалізації запропонованих в роботі рішень необхідне апаратне забезпечення (табл. 3.1). Аналіз вартості свідчить про можливість розгортання рішень Edge AI на основі SBC Raspberry Pi 5 [33] та акселератору ШІ Hailo-8 [34].

Таблиця 3.1 – Вартість обладнання для реалізації Edge AI

№ з/п	Найменування обладнання	Вартість обладнання, грн
1	Мікрокомп'ютер Raspberry Pi 5 8 ГБ	4699
2	Акселератор ШІ Raspberry Pi AI HAT+ на базі Halo-8 (26 TOPS)	5484
3	Блок живлення Raspberry Pi 27W PD USB-C	848
4	Радіатор і вентилятор для Raspberry Pi 5	250
5	Карта microSD MakerDisk на 64 ГБ з ОС Raspberry Pi	700
6	Адаптер USB-microSD для карти	65
Разом витрати на обладнання:		12046

Загальна вартість обладнання складає ≈ 12100 грн. Для створення периферійних пристроїв на основі рішень Edge AI до вказаних компонентів варто додати камеру AI Camera IMX500, в яку інтегровані інтелектуальні функції обробки зображень. На даний час, її вартість складає 5654 грн. Таким чином, цей варіант периферійного пристрою обійдеться у 17700 грн.

В цілому, акселератор ШІ Halo-8 значно покращив швидкість SBC Raspberry Pi 5 у завданнях CV. Це свідчить про доцільність подальших досліджень за цим напрямом.

ВИСНОВКИ

Конструктивні особливості акселераторів ШІ дозволяють забезпечити високий рівень паралелізму, низьку затримку та високу енергоефективність, що робить їх оптимальними для обробки великих даних у реальному часі. Особливий інтерес викликають нейроморфні та квантові обчислювальні системи, які представляють наступне покоління акселераторів ШІ та можуть забезпечити кардинальний прорив у швидкості та якості обчислень. Подальший розвиток цих напрямків може значно розширити межі практичного використання ШІ в різних галузях. Продуктивність акселераторів ШІ суттєво залежить від продуманої архітектури – кількості та типу обчислювальних блоків, ієрархії пам'яті та якості каналів міжблокового зв'язку. GPU здатні до обробки тисяч паралельних потоків, що робить їх корисними для масивних обчислювальних задач. TPU та NPU розраховані на обробку тензорних і матричних структур, що ідеально підходить для глибокого навчання. Важливим фактором є інтеграція високошвидкісної пам'яті, а також підтримка швидкісних інтерфейсів, що забезпечують масштабованість систем.

SBC стали ключовими інструментами в реалізації рішень Edge AI. Їх переваги – низьке енергоспоживання, компактні розміри, доступність та підтримка широкого спектра ОС і фреймворків. Завдяки цим якостям SBC дозволяють розміщувати інтелектуальну обробку даних безпосередньо на місці їх генерації – наприклад, на транспорті або в сенсорних вузлах. Обґрунтований вибір компонентів для Edge AI базується на 3-ох категоріях критеріїв: споживчі, інженерні, дослідницькі. Комплексне врахування цих параметрів дозволяє забезпечити ефективну розробку й швидкий перехід від прототипу до робочої системи. Серед найбільш розповсюджених рішень у сфері Edge AI варто виділити лінійку Nvidia Jetson, яка забезпечує високу продуктивність і повноцінну підтримку CUDA, TensorRT та PyTorch. Альтернативні рішення на RockChip або Qualcomm виграють у вартості, але

програють у сумісності з передовими моделями глибокого навчання. Зовнішні акселератори, наприклад, Hailo-8, Intel Movidius або Axelera AI дозволяють масштабувати обчислення без заміни основної платформи, що підвищує гнучкість проєкту.

Поєднання Raspberry Pi 5 з акселератором Hailo-8 стало оптимальним прикладом балансування продуктивності, вартості та споживаної енергії. Така комбінація дозволяє застосовувати рішення у сферах автономного відеомоніторингу, робототехніки, медичних приладів та IoT-платформ. Важливо й те, що екосистема підтримує популярні фреймворки (TensorFlow, ONNX), що полегшує портування існуючих моделей.

Результати тестування показали, що Hailo-8 демонструє кращу продуктивність і значно вищу енергоефективність. Для моделей типу YOLOv5, ResNet-50 чи MobileNetV3 акселератор забезпечує вищу швидкість інференсу при значно нижчому енергоспоживанні, що є ключовим для мобільних або живлених від акумуляторів систем. Комбінація Raspberry Pi 5 з Hailo-8 є ефективним та доступним варіантом для широкого кола задач. Вартість такого рішення значно нижча, ніж у систем на базі повноцінних GPU, при цьому продуктивність для задач комп'ютерного зору залишається високою. Це дозволяє використовувати платформу в стартапах, дослідницьких проєктах та вбудованих системах з обмеженим бюджетом. Хоча акселератори ШІ ефективно працюють з важливими робочими навантаженнями ШІ, потрібно вирішити низку завдань, щоб зробити їх ефективними в ширшому масштабі програм. Більшість проблем включають: енергоефективність, масштабованість, труднощі програмування та належний баланс між вартістю та продуктивністю.

Таким чином, результатами роботи є рекомендації щодо використання акселератору ШІ Hailo-8. Вони можуть бути використані для подальших досліджень за даною тематикою та при проектуванні вбудованих комп'ютерних систем.