Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Штучного інтелекту
(повна назва)

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Уніфікована інфраструктура для інтелектуальної обробки відеопотоків на пристроях EDGE.
(тема)

Виконав:
студент 2 курсу, групи СШІм-18-3
Сікачов Є. М.
(прізвище, ініціали)

Спеціальність 122 – Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми
(освітньо-професійна або освітньо-наукова)

Освітня програма Системи штучного інтелекту (СШІ)
(повна назва спеціалізації)

Керівник проф. Терзіян В.Я.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ В.О. Філатов
(підпис) (прізвище, ініціали)

2020 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Штучного інтелекту

Рівень вищої освіти другий (магістерський)

Спеціальність 122 – Комп'ютерні науки та інформаційні технології
(код і повна назва)

Спеціалізація Системи штучного інтелекту (СШІ)
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«____»_____ 20 ___ р.

ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУ

студентові Сікачову Єгору Миколайовичу
(прізвище, ім'я, по батькові)

1. Тема роботи
Уніфікована інфраструктура для інтелектуальної обробки відеопотоків на пристроях EDGE

затверджена наказом по університету від 30 березня 2020 р. № 480Ст
2. Термін подання студентом роботи до екзаменаційної комісії ____ _____ 2020 р.
3. Вихідні дані до роботи
Архітектура розробленої системи. Деталі технічної реалізації. Перелік використовуваних програмних засобів: ОС Linux Ubuntu 16.04. Технічне забезпечення: IBM-сумісний ПК з МП Pentium II та вище
4. Перелік питань, що потрібно опрацювати в роботі
Вступ. Обробка відео за допомогою згорткових нейронних мереж. Пристрої IOT для роботи з відеоданими. IaaS та PaaS парадигми. Розробка уніфікованої інфраструктури для інтелектуальної обробки відеопотоків на пристроях EDGE

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів)

Ієрархічна класифікація історичного розвитку згорткових нейронних мереж для класифікації зображень, Схема роботи мереж сімейства R-CNN, Алгоритм роботи YOLO, Приклад роботи SSD, Приклад сегментації зображення, Схема роботи алгоритму трекінгу, Підходи до аналізу відеоряду, Ілюстрація модифікацій парадигми EDGE обчислень, Порівняння парадигм IaaS та PaaS, Приклад атаки системи, Аналіз популярності парадигм IaaS, SaaS, FaaS та PaaS з часом, Архітектура додатка

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Основна частина	проф. Терзіян В.Я.		

КАЛЕНДАРНИЙ ПЛАН

	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
	Отримання завдання на атестаційну роботу	30.03.2020	виконано
	Пошук готових рішень і підходів	05.04.2020	виконано
	Аналіз готових рішень і підходів	10.04.2020	виконано
	Розгляд шляхів удосконалення рішень	15.04.2020	виконано
	Теоретичне обґрунтування моделі	22.04.2020	виконано
	Опис застосування моделі	27.04.2020	виконано
	Оформлення пояснювальної записки	02.05.2020	виконано
	Оформлення графічної частини та презентації	08.05.2020	виконано
	Подання роботи керівникові	12.05.2020	виконано
	Подання роботи на рецензування	14.05.2020	виконано
	Захист атестаційної роботи	21.05.2020	виконано

Дата видачі завдання 30 березня 2020 р.

Студент _____
(підпис)

Керівник роботи _____ проф. Терзіян В.Я.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Записка пояснювальна до атестаційної роботи: __ с., __ рисунків, __ таблиць, __ джерел.

EDGE, PAAS, IOT, CNN, ГЛИБИННІ НЕЙРОННІ МЕРЕЖІ, ВБУДОВАНІ СИСТЕМИ

Об’єктом дослідження є існуючі програмні та апаратні компоненти, що використовуються для створення EDGE систем з апаратними прискорювачами для запуску нейронних мереж.

Метою даної роботи є дослідження сучасних вбудованих систем для аналізу відео за допомогою штучних нейронних мереж у парадигмі EDGE, а також створення прототипу системи, що значно пришвидшить розробку схожих систем.

Методи дослідження – аналіз літератури та інтернет джерел, системне моделювання та декомпозиція.

Результатом роботи є описання теоретичної моделі обробки відеопотоку, котра відповідає низці вимог. А також практична реалізація прототипу такої системи.

Область застосування – продукти в середі інформаційних технологій, вбудовані системи для аналізу відеопотоків з використанням штучних нейронних мереж.

ABSTRACT

Diploma work: ____ pages, ___figures, ____ tables, __ sources.

EDGE, PAAS, IOT, CNN, DEEP NEURAL NETWORKS, EMBEDDED SYSTEMS

The object of the research is existing software and hardware components used to create EDGE systems with hardware accelerators to start neural networks.

The purpose of this thesis is to study modern embedded systems for video analysis using artificial neural networks in the EDGE paradigm, as well as creating a prototype of the system that significantly accelerate the development of similar systems.

Development methods are an analysis of literature and Internet sources, system modelling and decomposition.

The result of the work is a description of the theoretical model of processing the video stream that meets a number of conditions. As well as the practical implementation of the prototype of such a system.

Usage domains are products in the environment of information technology, embedded systems for analyzing video streams using artificial neural networks.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

CNN – convolutional neural network – згорткова нейронна мережа;

PAAS – platform as a service – платформа як сервіс;

IAAS – infrastructure as a service – інфраструктура як сервіс;

IOT – internet of things – інтернет речей.

ВСТУП

Починаючи з 2012 року в сфері інтелектуальної обробки зображень відбувся переворот. На щорічному конкурсі по розпізнаванню образів та машинному зору ImageNet було представлено згорткову нейронну мережу AlexNet. Представлена архітектура досягла похибки в 15.3%, що на 10.8% вище аніж попередній найкращий результат. І хоча ідея згорткових нейронних мереж не була новою, так як вже у 1998 році Ян ЛеКун представив світу згорткову нейронну мережу LeNet для класифікації зображень символів, саме з 2012 року починається стрімкий розвиток нейронних мереж [1].

За допомогою штучних нейронних мереж лише за декілька років почали ефективно вирішуватися різні класи задач, включаючи задачі машинного зору: класифікація зображень, детекція об’єктів, сегментація зображень, тощо. І хоча жодна нейронна мережа не гарантує 100 відсоткового вирішення будь-якої з цих задач, проте відсоток помилок у багатьох випадках близький або навіть перевищує за якістю результати, що демонструє людина. Це дозволяє переходити до використання розроблених алгоритмів не лише як елементів дослідницьких робіт, проте і як елемент систем, що працюють у повсякденному житті в абсолютно різних галузях: безпека, охорона здоров’я, розваги, автоматизація, автономні машини, робототехніка, тощо.

Описані вище нейронні мережі відносяться до класу штучних нейронних мереж прямого поширення, який насамперед активно застосовується для аналізу візуальної інформації та зображень. Тобто у контексті аналізу відеопотоків, що можуть бути представлені як потік зображень використання саме згорткових нейронних мереж є найдоцільнішим з точки зору реалізації у реальному світі. У той же час використання нейронних мереж є надзвичайно ресурсозатратною справою.

Лише у останні декілька років з’явилися алгоритми, що можуть працювати з достатньою ефективністю на мобільних пристроях та пристроях

з пониженим споживанням електроенергії. Цьому сприяли як дослідницькі роботи, які ставили за мету зменшення складності нейронних мереж, так і здобутки в галузях розробки спеціалізованого апаратного забезпечення.

Паралельно з бурхливим розвитком штучних нейронних мереж активно розвивається і галузь Інтернету речей. Хоча термін вперше з'явився у 1999 році, активний розвиток почався порівняно нещодавно разом з розвитком бездротових технологій та зменшенню розмірів обчислювальних пристроїв.

Саме на перетину цих галузей з'являється дуже перспективний напрям по аналізу зображень та відеопотоків на пристроях Інтернету речей за допомогою згорткових нейронних мереж. Цей напрям потребує величезних зусиль у дослідницькому та інженерному секторах. І в той час як дослідницькі роботи у цьому напрямку йдуть все глибше і глибше, то практичному аспекту приділяється недостатня кількість уваги. Це призводить до уповільнення інтеграції останніх розробок у повсякденне життя. Тим самим роблячи використання штучного інтелекту недоступним широким верствам населення та бізнесу, що не спеціалізується на розробці програмного та апаратного забезпечення. А це в свою чергу призводить до економічних наслідків, що впливають як на якість життя, так і на розвиток штучного інтелекту.

Одним із шляхів вирішення цієї проблеми на прикладі хмарних технологій є використання різних моделей хмарних обчислень. Починаючи від найдинамічніших IaaS та закінчуючи SaaS як одним із найменш налаштованим.

З цього можна сформувати об'єкт дослідження – системи Інтернету речей для аналізу відео з використанням згорткових нейронних мереж.

Предмет дослідження – процес створення та розгортання систем Інтернету речей для аналізу з використанням згорткових нейронних мереж.

Мета дослідження – запропонувати систему, що пришвидшить процес розробки та розгортання IoT систем для аналізу відео.

Задачі, котрі потрібно розглянути у процесі роботи є наступними:

- розглянути сучасні підходи у розробці систем IoT, що використовують штучні нейронні мережі;
- проаналізувати ключові елементи таких систем;
- проаналізувати шляхи демократизації у високотехнологічних рішеннях, що вже давно на ринку;
- запропонувати гібридний варіант рішення для досліджуваних систем;
- запрототипувати запропонований варіант;
- проаналізувати отримані результати;
- зробити припущення щодо покращення системи.

1 ОБРОБКА ВІДЕО ЗА ДОПОМОГОЮ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ ПРЯМОГО ПОШИРЕННЯ

Автоматизований аналіз відео став необхідним через великий обсяг відеоданих, які вже існують сьогодні та продовжують генеруватися. І більшість цих даних потребують обробки. Шляхом вирішення виникаючої проблеми є автоматизований аналіз відео. При цьому величезний обсяг даних потребує нетривіальної обробки, котру на сьогодні може виконувати лише людина.

Технології цифрової обробки зображень зараз використовуються в ряді сфер:

У медицині: деякі медичні інструменти використовують обробку зображень для різних цілей, таких як поліпшення якості зображення, стиснення зображення, розпізнавання об'єктів тощо. Відбувається опрацювання даних рентгенівського випромінювання, комп'ютерної томографії, позитронно-емісійної томографії, однофотонної емісійної комп'ютерної томографії, спектроскопії ядерно-магнітного резонансу та ультрасонографії.

У сільському господарстві: різні найважливіші завдання, такі як виявлення бур'янів, сортування їжі, боротьба зі збиранням врожаю та збирання плодів, виконуються автоматично за допомогою обробки зображень. Зрошуване картографування земель, визначення показників рослинності, вимірювання навісів тощо можливе з хорошою точністю завдяки використанню методів візуалізації в різних спектрах, таких як гіперспектральна візуалізація, інфрачервона діаграма тощо. Обробка зображень відіграє вирішальну роль у прогнозуванні погоди, а саме передбаченні кількості опадів, граду, штормів. Метеорологічні радари широко використовуються для виявлення хмарних опадів і, виходячи з цієї інформації, системи передбачають негайну інтенсивність опадів.

Ретушовані та сплайсифіковані фотографії широко використовуються у газетах та журналах з метою покращення якості зображення. У фільмах багато складних сцен створюються за допомогою інструментів редагування зображень та відео, заснованих на операціях обробки зображень та відео. Для глобальної медіа-розважальної компанії Latent View витягнув із IMDB понад 6000 плакатів фільмів разом із їх метаданими (жанр, акторський склад, постановка, рейтинги тощо), щоб передбачити успіх фільмів за допомогою аналізу зображень. Кольорові схеми та об'єкти на плакатах фільму були проаналізовані за допомогою алгоритмів машинного навчання (ML) та методів обробки зображень для задачі аналізу популярності майбутніх кінострічок.

У розважальних та соціальних мережах: розпізнавання облич широко використовуються на сайтах соціальних мереж, де, як тільки користувач завантажує фотографію, система автоматично ідентифікує і дає пропозицію позначити людину по імені.

У безпеці: біометричні методи перевірки використовуються для розпізнавання людей на основі їх поведінки, зображень чи особливостей. Щоб створити попередження про особливо небажану поведінку, використовуються системи відеоспостереження з метою аналізу руху та діяльності людей. Велика кількість банків та інших відділів використовують системи відеоспостереження на основі обробки зображень для виявлення небажаних дій.

У банківському бізнесі та управлінні: використання методів обробки зображень стрімко зростає у сфері фінансових послуг та банківської справи. Системи комп’ютерного зору дозволяють клієнтам здавати чеки в електронному вигляді за допомогою мобільних пристроїв або сканерів. Дані з контрольного зображення витягуються та використовуються замість фізичної перевірки. Розпізнавання облич також використовується в процесі ідентифікації клієнтів банку. Деякі банки використовують біометричні дані

для захисту конфіденційної інформації. Перевірка та розпізнавання підписів також відіграє важливу роль в автентифікації підпису клієнтів.

У маркетингу та рекламі: Деякі компанії використовують обмін зображеннями через соціальні медіа, щоб відстежувати вплив останніх продуктів та рекламних компаній. Туристичний відділ використовує зображення для реклами туристичних напрямків.

В обороні: обробка зображень, поряд з мистецькою розвідкою, сприяє обороні на основі двох основних потреб військових: одна – це автономна операція, а інша – використання результатів з різноманітних складних датчиків для прогнозування небезпеки чи загрози. В Ірансько-Іракській війні для розвідки території противника використовувались технології дистанційного зондування. Супутникові знімки аналізуються з метою виявлення, пошуку та знищення озброєння та оборонних систем, що використовуються силами противника.

У промисловій автоматизації: в промисловій автоматизації спостерігається безпрецедентне використання обробки зображень. Система «Автоматизація складальних ліній» визначає положення та орієнтацію компонентів. Автоматизація перевірки недосконалості поверхні можлива завдяки обробці зображень. Основними завданнями є визначення якості об'єкта та виявлення будь-яких відхилень у продуктах. Багато галузей також використовують класифікацію продукції за допомогою автоматизації форми.

У криміналістиці: підроблені документи широко застосовуються у кримінальних та цивільних справах, таких як оскаржувані заповіти, фінансова робота з папером та професійна ділова документація. Документи, такі як паспорти та посвідчення водія, часто підробляються з метою їх незаконного використання в якості підтвердження ідентифікації. Судові департаменти повинні виявити справжність таких підозрілих документів. Ідентифікація підробки документів стає все складніше через наявність сучасних інструментів редагування документів. Підробник використовує новітні технології для

вдосконалення свого мистецтва. Документи сканування на комп'ютері копіюються з одного документа в інший, щоб зробити їх справжніми. Підробка не лише стосується документів, вона також набуває популярності в образах. Зображення відіграють неабияку роль у різних областях, таких як криміналістичне розслідування, кримінальне розслідування, системи спостереження, системи розвідки, спорт, юридичні послуги, медичні знімки, страхові претензії та журналістика. Майже десятиліття тому Іран звинувачували в тому, що він працював із зображеннями своїх ракетних випробувань; Зображення було опубліковане на офіційному веб-сайті Іранської революційної гвардії, який стверджував, що чотири ракети одночасно прямують у небо. Майже всі газети та журнали новин опублікували цю фотографію, включаючи «Лос-Анджелес Таймс», «Чикагська трибуна» та BBC News. Пізніше було виявлено, що лише три ракети були успішно запущені, одна ракета вийшла з ладу. Зображення було зафіксовано з метою перебільшення військових можливостей Ірану.

У покращенні та вдосконаленні підводних зображень: підводні зображення часто не зрозумілі. Ці зображення мають різні проблеми, такі як шум, низька контрастність, розмитість, нерівномірне освітлення тощо. Для відновлення наочності чіткості використовуються методи покращення зображення [2].

В рамках даної роботи особливий інтерес викликають сфери, де доцільно використовувати пристрої IoT, що використовують штучні нейронні мережі для аналізу вхідного відеоряду. Це насамперед сфери промислової автоматизації, робототехніки, безпеки та сільському господарстві. Далі будуть розглянуті основні типи згорткових нейронних мереж у контексті вирішуваних задач для аналізу відеоряду.

1.1 Типи задач, що вирішуються при аналізі відео

Аналіз зображень тісно пов’язаний з аналізом відеоряду, оскільки відеоряд у своєму представленні є серією зображень. Звідси і витікає, що аналіз відеоряду включає в себе всі можливі задачі для аналізу зображень, а також додає ті задачі, які можна розв’язувати, маючи інформацію про динамік, рух, сцену тощо.

1.1.1 Класифікація зображень

Класифікація зображень включає присвоєння мітки цілому зображенню чи фотографії, найчастіше, з заданого набору міток.

Цю проблему також називають «класифікацією об'єктів» та «розпізнаванням зображень», хоча останній термін може застосовуватися до набагато більш широкого набору завдань, пов'язаних з класифікацією вмісту зображень.

Класифікацію поділяють на:

1. бінарну класифікацію (позначення фотографії як такої, що містить або не містить ракову пухлину);
2. багатокласова класифікація (класифікація цифр);
3. багатокласова класифікація на декількох наборах класів (відповісти про тип одягу на зображенні, а також його колір).

Історичний розвиток, а також ієрархічна класифікація згорткових нейронних мереж продемонстрована на рисунку 1.1 [4].

Загальний тренд, який найкраще описує розвиток згорткових нейронних мереж для класифікації зображень: використовуючи меншу кількість параметрів отримувати більш точні результати. Для порівняння мережі архітектури VGG-16, опублікованої у 2014 році зі 138 мільйонами параметрів демонстрували точність на датасеті ImageNet ~74%, а сучасні

моделі з вдвічі меншою кількістю параметрів EfficientNet (66 мільйонів параметрів) демонструють точність на тому ж датасеті ~84%.

Також цікавим фактором є кількість параметрів, що дозволяє працювати на мобільних девайсах в режимі реального часу. Архітектура для класифікації зображень MobileNetV3, продемонстрована у 2019 році, при кількості параметрів у 5 мільйонів демонструє точність у ~74% на ImageNet. При цьому особливості архітектури дозволяють виконуватися мережі на сучасних мобільних пристроях зі швидкістю близько 30 кадрів на секунду.

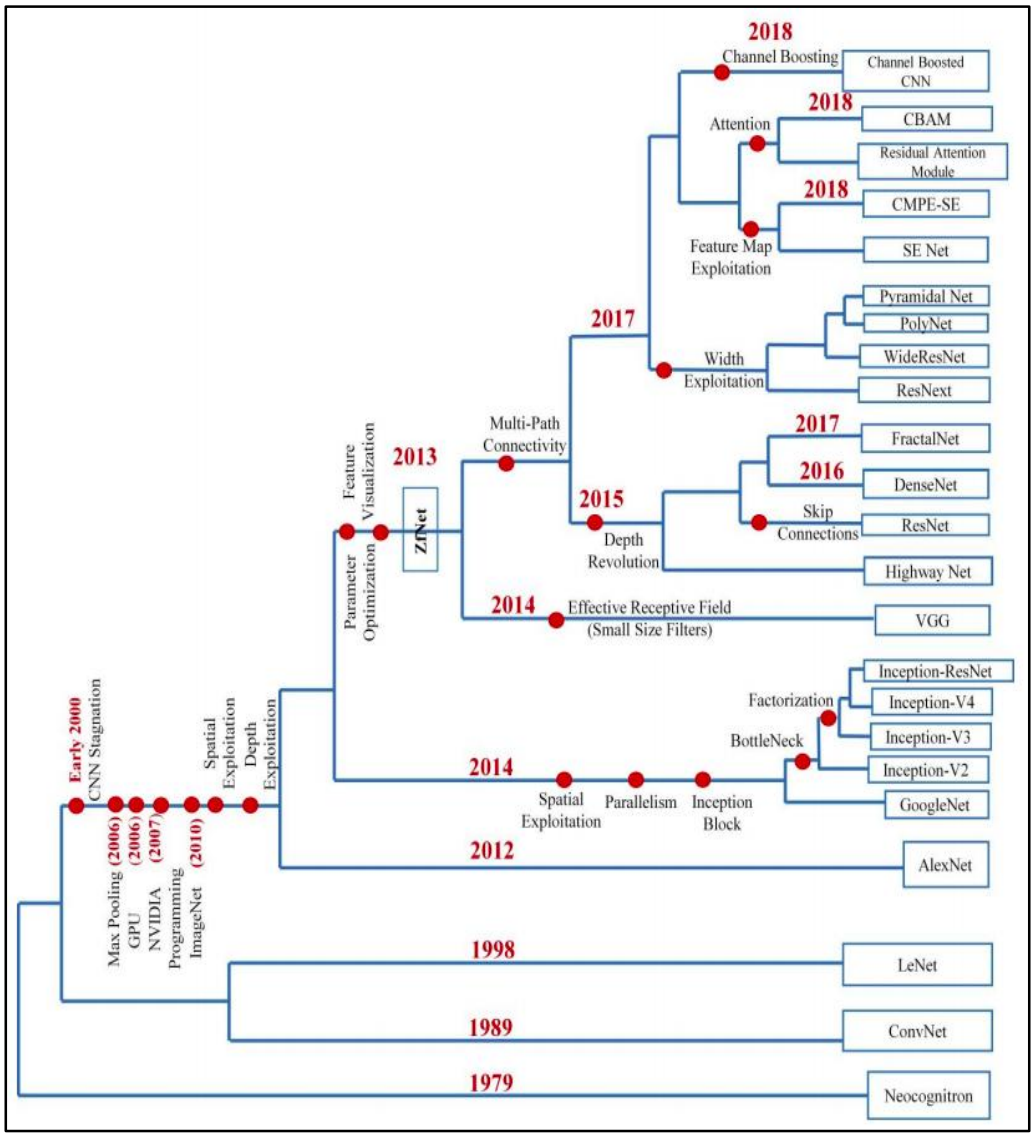

Рисунок 1.1 – Ієрархічна класифікація історичного розвитку згорткових нейронних мереж для класифікації зображень

1.1.2 Класифікація зображень з подальшою локалізацією

Класифікація зображень з локалізацією включає присвоєння мітки класу зображенню та виділення регіону розташування об'єкта на зображенні обмежувальним вікном. Є більш складною версією класифікації зображень.

Прикладом класифікації з подальшою локалізацією є присвоєння мітки наявності ракової пухлини знімку рентгенографії, а після видача чотирьох піксельних координат прямокутника, в який вписана ділянка з зображенням ракової пухлини. Детальний технічний опис цього підходу буде приведений у наступному підпункті, оскільки підходи до вирішення задачі дуже тісно переплітаються.

1.1.3 Детекція об’єктів

Детекція об’єктів – це завдання класифікації зображень з локалізацією, при цьому зображення може містити кілька об'єктів різних класів. Це більш складне завдання, ніж проста класифікація зображень або класифікація зображень з локалізацією, оскільки часто в зображенні є кілька об'єктів різних типів. Прикладом детекції об’єктів є визначення набору прямокутників, що описують дорожні знаки на зображенні вулиці.

У методах детектування об’єктів виділяють два сімейства методів: багато етапні да одноетапні методи. Перше сімейство також називають R-CNN сімейством. R-CNN – одна із перших нейронних мереж цього типу. Алгоритм роботи цих нейромереж наступний: спершу на зображенні знаходяться потенційні регіони, що містять у собі об’єкт, а потім ці регіони аналізуються мережею для класифікації зображень (рисунок 1.2).

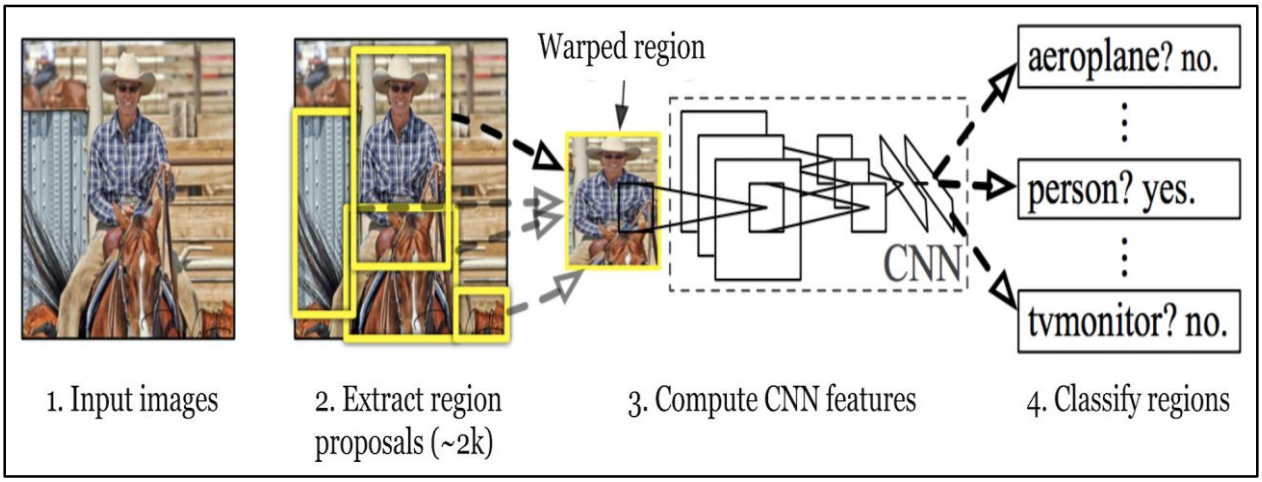

Рисунок 1.2 – Схема роботи мереж сімейства R-CNN

Такий підхід забезпечує дуже високу точність. У той же час суттєвим недоліком є надзвичайно велика ресурсоємкість мережі. Це часто унеможливлює використання мереж сімейства R-CNN у сучасних системах.

Альтернативою до цього підходу є одноетапні мережі. Загалом виділяють дві найбільш популярні мережі: YOLO (“You Only Look Once) та SSD (Single Stage Detector).

YOLO поділяє зображення на набір клітин різного розміру, а потім для кожної клітини робить аналіз на предмет наявності об’єкту всередині. Після отримання всі регіони агрегуються і на виході видається лише один набір регіонів (рисунок 1.3).

На відміну від YOLO, SSD не розбиває зображення на сітки довільного розміру, але прогнозує зміщення заздалегідь заданих ящиків для прив’язки для кожного місця розташування карти функцій. Кожен ящик має фіксований розмір і положення відносно відповідної комірки. Всі ящики з якорями перекручують всю карту зображень по-новому.

Карти зображень на різних рівнях мають різні розміри сприйнятливих полів. Якорі для прив’язки на різних рівнях змінюються масштабом, так що одна карта функцій відповідає лише за об'єкти в одному конкретному

масштабі. В результаті отримується набір регіонів, який агрегується (рисунок 1.4).

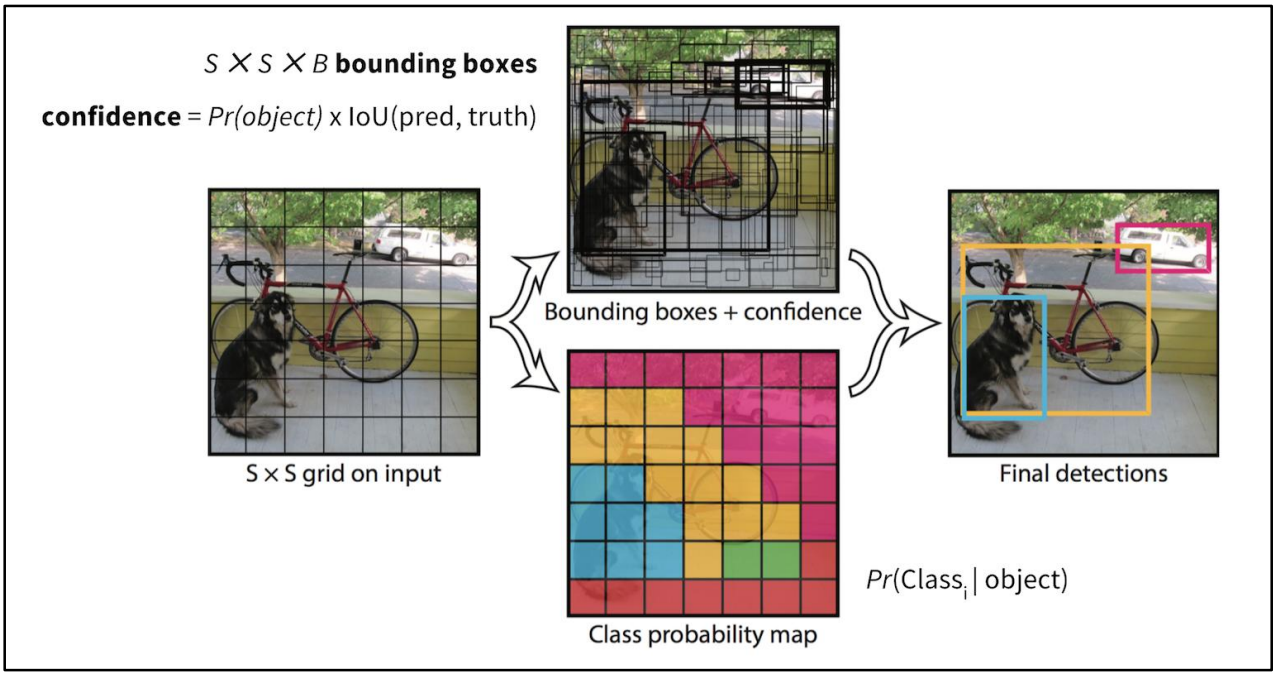

Рисунок 1.3 – Алгоритм роботи YOLO

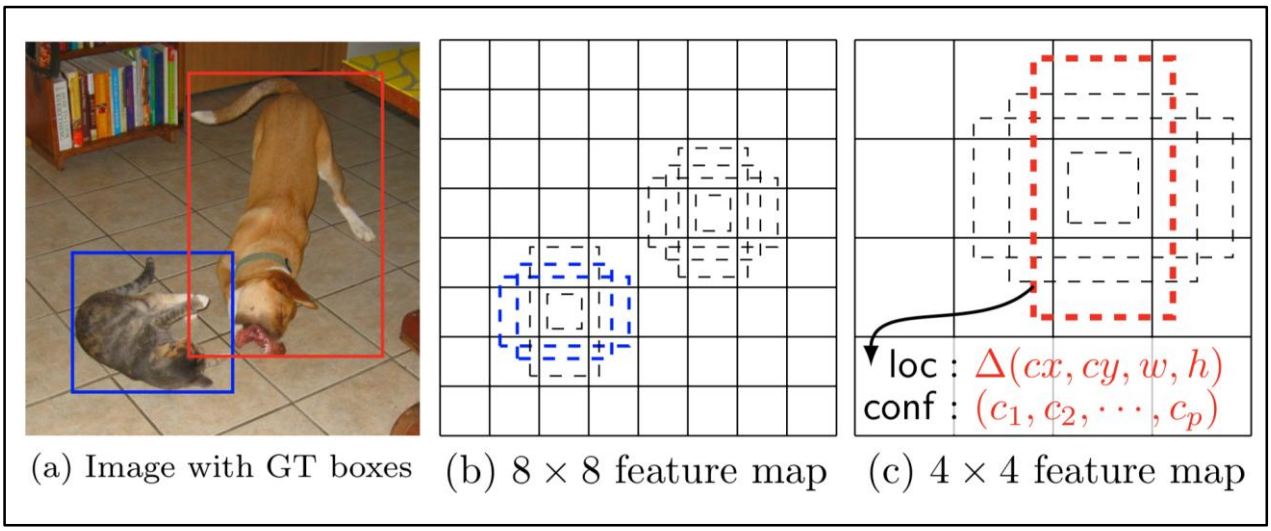

Рисунок 1.4 – Приклад роботи SSD

Карти зображень на різних рівнях мають різні розміри сприйнятливих полів. Якорі для прив’язки на різних рівнях змінюються масштабом, так що одна карта функцій відповідає лише за об'єкти в одному конкретному масштабі. В результаті отримується набір регіонів, який агрегується (рисунок 1.4).

Завантаження SSD в сучасному ноутбуці займає приблизно 0,5 секунди, а запуск займає 0,19 секунди. Під час завантаження SSD в Raspberry в середньому йде 2,97 секунди, а час запуску - приблизно 2,31 секунди.

Завантаження моделі YOLOv3 крихітної версії займає 0,091 секунди, а запуск займає 0,2 секунди.

Отже вже на таких пристроях IoT як Raspberry можна досягти швидкості близької до реального часу.

1.1.4 Сегментація зображень

Задача сегментації зображень зводиться до того, щоб прийняти на вхід багатоканальне зображення розміром H x W x C, де H – висота, W – ширина, C – кількість каналів, а повернути зображення розміром H x W x 1, де у відповідність кожному пікселю з глибиною C поставлене значення мітки, що відображає певний клас. Прикладом сегментації зображень є рисунок 1.5 – у відповідність кожному пікселю з RGB зображення поставлений певний клас: зелений – дорога, помаранчевий – автомобіль, і т. д.

Рисунок 1.5 – Приклад сегментації зображення

Один з популярних початкових підходів до глибокого навчання - класифікація патчів, де кожен піксель був окремо класифікований на класи, використовуючи навколо нього патч зображення. Основна причина використання патчів полягала в тому, що класифікаційні мережі зазвичай мають повне з'єднані шари і тому потрібні зображення фіксованого розміру.

У 2014 році для задачі семантичної сегментації широко використовувалися Fully Convolutional Networks (FCN) без повністю пов'язаних шарів. Це дозволило генерувати карти сегментації для зображення будь-якого розміру, а також було набагато швидшим порівняно з підходом класифікації патчів. Практично всі наступні сучасні підходи щодо смислової сегментації прийняли цю парадигму.

Окрім повністю пов'язаних шарів, однією з головних проблем використання CNN для сегментації є об'єднання шарів. Шари об'єднання збільшують поле зору і здатні агрегувати контекст, відкидаючи інформацію «куди». Однак семантична сегментація вимагає точного вирівнювання карт класів, і, таким чином, потрібно зберігати інформацію «куди». Для вирішення цього питання в літературі склалися два різних класи архітектури.

Перший – це архітектура кодера-декодера. Енкодер поступово зменшує просторовий вимір з об'єднанням шарів, а декодер поступово відновлює деталі об'єкта та просторовий вимір. Зазвичай існують з'єднання ярликів від кодера до декодера, щоб допомогти декодеру краще відновити деталі об'єкта. U-Net – популярна архітектура цього класу.

Використання легких енкодерів, додаткова оптимізація нейромережі, обмеження розміру вхідних даних може давати суттєвий приріст швидкодії. Одним з прикладів такої надшвидкої взаємодії є сучасні інструменти для редагування фото на телефонах.

1.1.5 Детекція та розпізнавання тексту

Точним терміном, що описує дану задачу є оптичне розпізнавання символів або OCR – це підхід, який дозволяє перетворювати різні типи документів, наприклад скановані паперові документи, PDF-файли або зображення, зняті цифровою камерою, в редаговані та доступні для пошуку дані.

Прикладом оптичного розпізнавання символів є паперовий документ – наприклад, стаття журналу, брошура або контракт PDF, який за допомогою методів комп’ютерного зору проаналізований та перетворений з зображення у відповідний структурований текстовий набір даних.

Розпізнавання тексту включає два етапи: спочатку виявлення та визначення обмежувального поля для текстових областей зображення, а також у межах кожної області тексту окремих символів тексту. По-друге, ідентифікація символів.

Для виявлення символів і слів у зображеннях використовуються стандартні моделі глибокого навчання, такі як Mask-RCNN, SSD або YOLO. Однак, моделі глибокого навчання, які добре ідентифікують об'єкти на зображеннях (тобто тварин або транспортні засоби), можуть погано визначати текстові символи та можуть працювати гірше, ніж алгоритми OCR, основані на класичному комп’ютерному зорі.

Тому розвинулися спеціалізовані моделі глибокого навчання, які допомагають локалізувати та виявляти текст у зображеннях. Ось кілька моделей, що часто використовуються.

Конволюційно-рецидивуюча нейронна мережа (CRNN). Підхід CRNN визначає слова за допомогою трьох етапів: типова згорткова нейронна мережа – перший шар розбиває зображення на фічі, а потім поділяє їх на набори. Ці набори подаються у комірку LSTM, яка забезпечує послідовність, що визначає взаємозв'язок між символами. Вихід комірки LSTM подається в

шар транскрипції, який приймає символьну послідовність, включаючи надлишкові символи, і використовує ймовірнісний підхід для очищення виводу.

Модель повторної уваги (RAM). Модель повторної уваги базується на ідеї, що коли людському оці представлено нову сцену, певні частини зображення привертають його увагу. Око фокусується спочатку на тих «проблисках» інформації та отримує інформацію з них.

У моделі зображення обрізається на різні розміри навколо загального центру, а потім в кожній частині створюються вектори зору з помітними рисами. Ці вектори зору згладжені і проходять через «мережу зору» на основі зорової уваги.

Потім вектори проблиску передаються в локальну мережу, яка використовує RNN для прогнозування наступної частини зображення, на яку слід звернути увагу. Це місце є наступним входом для мережі проглядання. Поступово модель досліджує додаткові частини зображення, щоразу виконуючи зворотне розповсюдження, щоб перевірити, чи достатня інформація про попередні проблиски для досягнення високого рівня точності.

1.1.6 Розпізнавання облич

Розпізнавання обличчя вирішує дві задачі: задачу ідентифікації та задачу верифікації. Метою першої задачі є відповідь на питання хто зображений на фотографії, а метою другої задачі є відповідь на питання, чи на двох різних фотографіях одна і та сама людина. Прикладами розпізнавання облич є програми соціальної мережі Facebook, яка після завантаження фото каже, хто на ньому зображений, а також програми для розблокування телефонів.

Загальний алгоритм розпізнавання облич є сталим протягом останніх років і найкраще описується на прикладі мережі FaceNet. Згорткова нейронна

мережа FaceNet покладається на пікселі зображення як функції, а не вилучає їх вручну. Основна ідея алгоритму – це відображення обличчя як 128-мірного вектора.

Оскільки ці вектори представлені у спільному векторному просторі, векторну відстань можна використовувати для обчислення подібності між двома векторами. Це методика обчислення того, наскільки схожі два обличчя.

Останнім етапом в архітектурі FaceNet є "триплетна різниця", яка мінімізує відстань між якорем і відомим позитивом (подібність між двома гранями), при цьому максимально збільшуючи відстань між якорем і відомим негативом (несхожість).

1.1.7 Трекінг об’єктів

Відстеження декількох об'єктів, яке також називається багатоцільовим відстеженням, є завданням комп'ютерного зору, метою якого є аналізувати відеоролики з метою виявлення та відстеження об'єктів, що належать до однієї чи кількох категорій, наприклад, пішоходів, автомобілів, тварин та неживих предметів, без попередніх знань про появу та кількість цілей. На відміну від алгоритмів виявлення об'єктів, вихід яких являє собою сукупність прямокутних обмежувальних коробок, ідентифікованих їх координати, висоту та ширину, алгоритми трекінгу також пов'язують ідентифікатор цілі з кожним полем (відомим як виявлення), щоб розрізняти внутрішньокласові об’єкти [5].

Незважаючи на величезну різноманітність підходів, представлених в літературі, переважна більшість алгоритмів трекінгу мають спільний підхід, представлений на рисунку 1.6:

- етап детекції: алгоритм виявлення об'єктів аналізує кожен вхідний кадр для виявлення об'єктів, що належать до цільового класу;

- отримання фіч або параметрів руху: один або кілька алгоритмів вилучення функцій аналізують задетектований об’єкт і отримують фічі чи параметри руху;
- стадія трекінгу: відбувається покадрове порівняння об’єктів;
- етап реасоціації: у випадку втрати об’єкта, а потім повторної появи алгоритм дізнається про це.

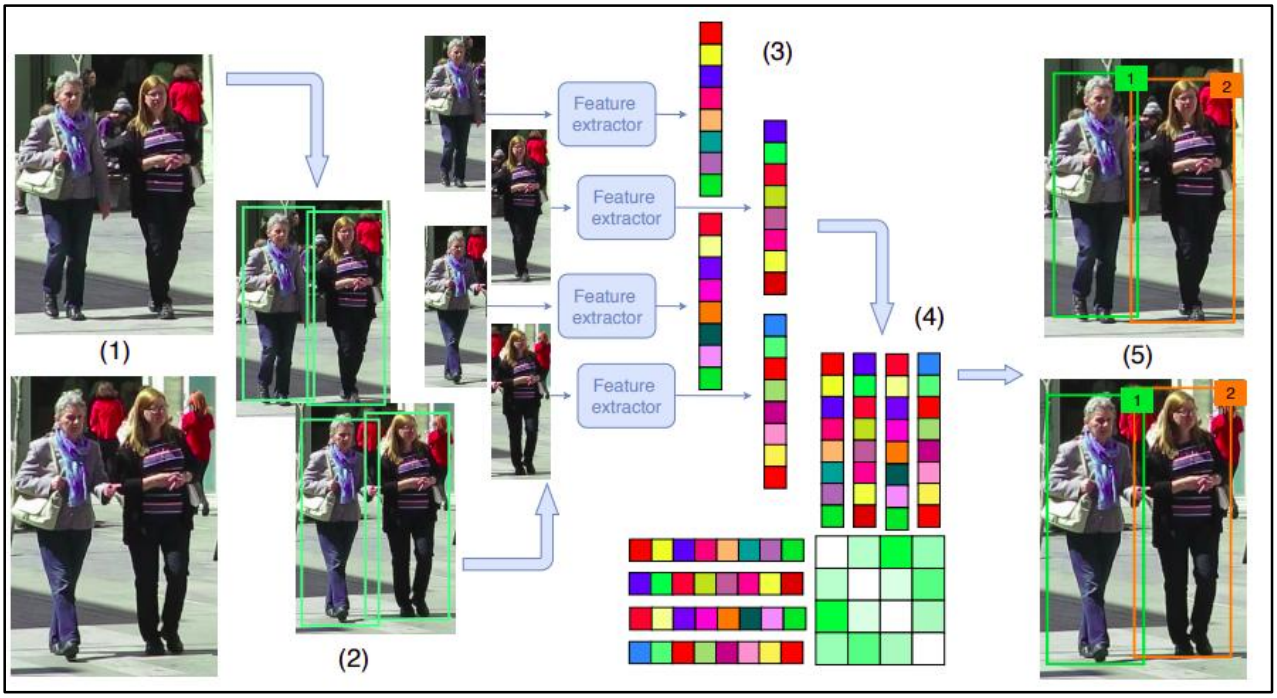

Рисунок 1.6 – Схема роботи алгоритму трекінгу

1.1.8 Розпізнавання дій

Розпізнавання людської діяльності або коротко HAR – це широке поле дослідження, яке займається виявленням конкретного руху чи дії людини на основі даних з камер. Формальне описання наступне: поділ наборів зображень на категорії з подальшою класифікацією цих категорій. Прикладом є відео футбольного матчу, в якому виділені моменти з пенальті, штрафними, кутовими.

Так чи інакше розпізнавання дій зводиться до аналізу часових рядів. І при аналізі відео як часового ряду найчастіше використовують підходи, зображені на рисунку 1.7.

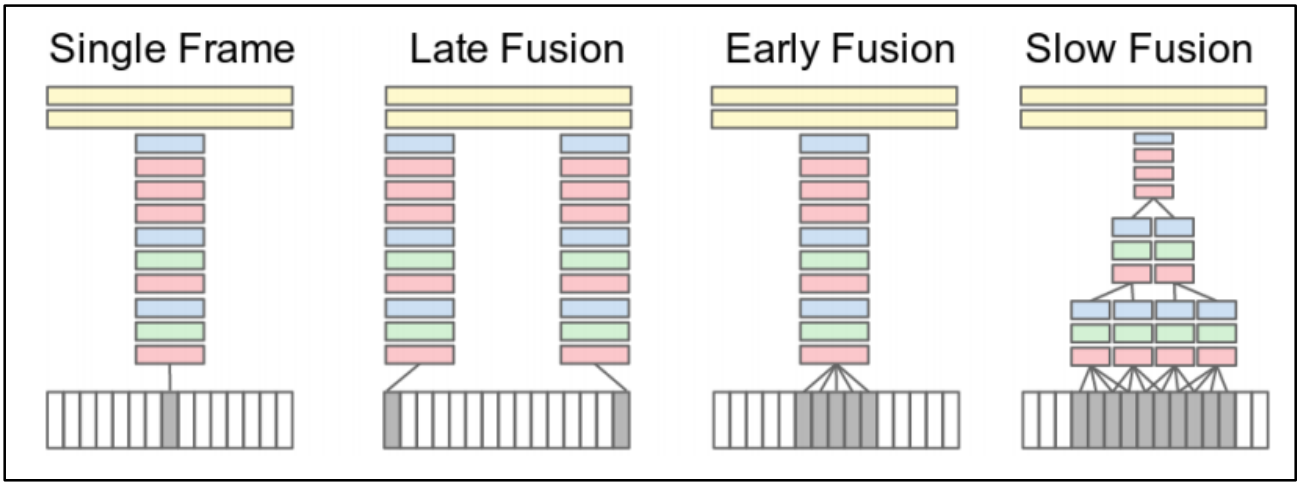

Рисунок 1.7 – Підходи до аналізу відеоряду

Однокадровий. Використовується однокадрова базова архітектура, щоб зрозуміти внесок статичного вигляду в точність класифікації.

Ранній синтез. Розширення Early Fusion поєднує інформацію у всьому часовому вікні одразу на одному рівні пікселів.

Пізній синтез. Модель пізнього синтезу розміщує дві окремі однокадрові мережі з певним кроком одна від одної.

Повільний синтез є балансом між раннім та пізнім і покликаний забезпечити доступ мережі як до локальних так і більш глобальних змін [6].

1.2. Аналіз найбільш поширених задач комп’ютерного зору та моделей нейронних мереж, які їх вирішують

Проаналізувавши всі описані вище задачі можна сказати, що похідною задачею є класифікація зображень. У тому чи іншому вигляді моделі для класифікації зображень використовуються в детекції, сегментації, трекінгу і

інших задачах. При цьому загальним трендом у нейронних мережах для класифікації є зменшення кількості параметрів та збільшення точності результатів. У комбінації з результатами останніх років щодо запуску нейронних мереж на різних девайсах у різних задачах можна стверджувати, що навіть пристрої IoT у найближчому майбутньому зможуть вирішувати навіть такі складні задачі як трекінг та розпізнавання активності на відео.

Окремо слід зауважити, що попри великий набір досліджень, що відбуваються останнім часом у сфери глибинного навчання для задач комп’ютерного зору, існує певний клас моделей нейронних мереж, що добре зарекомендував себе і широко використовується в індустрії.

Цей факт наводить на дуже цікавий висновок: більшість сучасних систем базується на обмеженому наборі моделей, що працюють достатньо добре, але не на рівні провідних досліджень, проте в той же час гарантують стабільність роботи.

2 ПРИСТРОЇ IOT ДЛЯ РОБОТИ З ВІДЕОДАНИМИ

EDGE обчислення стали популярною парадигмою для мобільних та IoT програм з низькою затримкою або високою пропускною здатністю. Привабливість EDGE обчислень була ще більше підвищена завдяки недавній доступності обладнання для спеціального призначення для прискорення конкретних обчислювальних завдань, таких як виконання глибинних нейромереж на IoT пристроях. На сьогоднішній день кращі прискорювачі можуть забезпечити порівняну, і в багатьох випадках, кращу продуктивність, якщо порівнювати співвідношення ціна якість ніж традиційні EDGE та хмарні сервери.

EDGE обчислення останнім часом стали доповненням до хмарних обчислень для запуску онлайн-додатків із низькою затримкою або високою пропускною здатністю. Інтернет речей (IoT) та мобільні додатки особливо добре підходять для парадигми EDGE обчислень, оскільки вони часто виробляють потокові дані, що потребують аналізу та контролю в реальному часі, які можна оптимально виконувати на IoT пристроях.

Наразі існує велика кількість модифікацій звичайної хмарної архітектури для задач EDGE обчислень.

«Хмарочоси» являють собою одну популярну парадигму EDGE обчислень, що тягне за собою розгортання кластерних серверів у кінцевих точках мережі; розгортаючи традиційні сервери близько до пристроїв інтернету речей, провайдери хмарних рішень дозволяють розгортати додатки «серверного класу» у кінцевих точках, а не в хмарі.

«EDGE шлюзи» представляють інший варіант хмарних обчислень. Парадигма передбачає розгортання вбудованих вузлів, окремо або групами, для того щоб вони служили центром для таких додатків, як розумні будинки. При такій схемі шлюзи забезпечують обмежені можливості обчислення на пристроях, але в той же час надаються корисну функціональність, таку як

агрегація даних і локальна обробка на вузлі для певних завдань, що потребують низької затримки.

Описані вище варіанти є дуже різні компромісами. Парадигма “EDGE шлюзи” використовує невеликі форми форм-фактори (наприклад, Raspberry Pi-class вузли), має низьку вартість, низьке енергоспоживання а також обмежені можливості обчислення, що збільшує надійність. EDGE обчислення в стилі «Хмарочос», з іншого боку, забезпечує набагато більші обчислювальні можливості на краю, але це тягне за собою більш високі витрати на обладнання, більші форм-фактори серверу та більшу потужність споживання; також є менша залежність від хмари для багатьох додатків.

Нещодавно з'явився третій варіант крайових обчислень, який поєднує ключові переваги обох парадигм. Ця парадигма, яка є спеціалізованою EDGE архітектурою, стала можливою з появою апаратного забезпечення спеціального призначення, призначеного для прискорення специфічних обчислювальних операцій. Зокрема, ряд апаратних прискорювачів обладнання, таких як Intel Movidius Vision VPU, чіп обробних тензорів від Google TPU, Jeton Nano та EDGE GPU від Nvdia та нейронний чіп Apple. Ці прискорювачі розроблені для чітких задач обчислення програм штучного інтелекту, наприклад, комп'ютерного зору, мовленнєвої аналітики та глибокого навчання [7].

Якщо узагальнювати типи архітектур додатків, що використовують EDGE парадигму, то можна виділити наступні типи.

Додатки інтернету речей на основі хмарних і EDGE обчислень: Багато пристроїв інтернету речей з можливістю приєднання до мереж WiFi використовують дворівневу хмарну архітектуру. Пристрій приймає дані та після цього передає їх у хмару, яка їх обробляє.

Дедалі більше пристроїв інтернету речей використовують трирівневі архітектури . Початкова обробка виконується на клієнтських пристроях, після цього дані передаються на EDGE сервер, де виконується основна частина

обчислень. після виконання цих обчислень дані агрегуються центральним хмарним сервером.

Edge навантаження на основі глибинних нейронних мереж: це так званий штучний інтелект на EDGE. Дані збираються клієнтськими пристроями, а після цього передаються на EDGE для виконання спеціалізованих обчислень, пов'язаних з запуском нейронних мереж. Особливої популярності така архітектура досягла останнім часом завдяки поширенню розумних помічників.

Спеціалізовані EDGE обчислення та EDGE прискорювачі: найновіша модифікація парадигми. Більшість обчислень відбувається безпосередньо на пристрої інтернету речей, в тому числі і спеціалізованих обчислень, таких як запуск нейронних мереж.

Схематичне зображення різних модифікацій парадигми EDGE обчислень приведене на рисунку 2.1. Зліва на право: звичайна архітектура додатку з використанням приладів інтернету речей, де дані передаються з клієнтських пристроїв у хмару; архітектура з проміжною ланкою у вигляді EDGE сервера з метою проміжних обчислень; архітектура з акселераторами на проміжному та клієнтському пристроях; архітектура для локального кластеру з використанням прискорювачів.

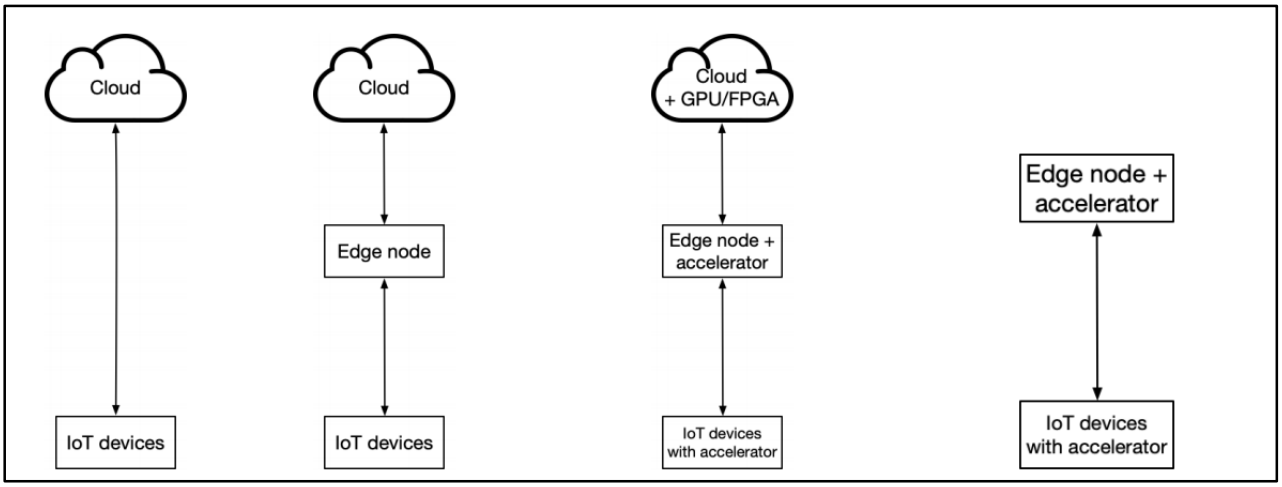

Рисунок 2.1 – Ілюстрація модифікацій парадигми EDGE обчислень

2.1 Апаратні прискорювачі для обчислень нейронних мереж

Висока продуктивність від програм глибокого навчання досяжна лише тоді, коли модель глибинного навчання навчається на величезній кількості даних, часто на терабайтах. Лише графічні процесори та центральні процесори мають можливість обробляти таку велику кількість даних у розумний проміжок часу. Це робить програми глибокого навчання здебільшого орієнтовані на GPU. Однак останнім часом було вжито багато зусиль, щоб зробити пристрої з обмеженими ресурсами сумісними з глибинним навчанням. Наразі для розгортання ML в мережі використовуються різні типи невеликих пристроїв, включаючи ASIC, FPGA, RISC-V та інші IoT пристрої. Загалом можно виділити чотири категорії пристроїв для розгортання ML на EDGE.

2.1.1 Application-specific integrated circuit (ASICs)

Edge TPU and Coral Dev Board Блок обробки тензорів (TPU26) – це чіп ASIC, розроблений Google для запуску ML на EDGE пристроях. Це прискорює запуск МЛ і може виконувати згорнуті нейронні мережі (CNN). Він здатний запускати алгоритми комп'ютерного зору, такі як MobileNets V1 / V2, SSD MobileNets V1 / V2 і Inception V1-4. TPU також може запускати API TensorFlow Lite та NN. Якщо користувач вибирає живлення в ефективному режимі, Edge TPU може виконувати найсучасніші моделі мобільного зору зі швидкістю 100 кадрів в секунду. The Coral Dev Board використовує Edge TPU як спільний процесор для запуску програм машинного навчання. Набір Coral Dev складається з двох частин, плінтуса та модуля системи (SOM). Плінтус має 40-контактний графічний інтерфейс GPIO для інтеграції з різними датчиками або пристроями IoT, а SOM має процесор Cortex-A53 з додатковим ядро Cortex-M4, 1 Гб оперативної пам’яті та 8 ГБ флеш-пам’яті, що допомагає запускати ОС Linux на Edge. USB-прискорювач Coral28 – це пристрій, який

допомагає виконувати моделі ML на таких маленьких пристроях, як Raspberry Pi. Прискорювач – це спільний процесор для існуючої системи, який може підключатися до будь-якої системи Linux за допомогою порту USB-C.

SparkFun Edge – це пристрій аудіо-аналізу в режимі реального часу, який виконує моделі машинного навчання, виявляє ключове слово, наприклад, "так" і відповідає відповідно. Розроблений спільно Google, Ambiq та SparkFun, він використовується для розпізнавання голосу та жестів на EDGE без допомоги віддалених служб. Має 32-бітний процесор ARM Cortex-M4F 48 МГц з режимом розриву 96 МГц, використання вкрай низької потужності, 384 КБSRAM, флеш-пам’яті 1 Мб та виділений процесор Bluetooth BLE 5. Він також має два вбудовані мікрофони, 3-осьовий акселерометр, роз'єм камери та інші роз'єми вводу / виводу. Цей пристрій може працювати 10 днів з монетним акумулятором CR2032. Ambiq Apollo3 – це комплект для розробки програмного забезпечення, доступний для будівництва AI-програми із Edge SparkFun.

Intel Movidius – це модуль обробки, який може прискорити глибинні нейронної мережі в пристроях з обмеженими ресурсами, таких як інтелектуальні камери безпеки або безпілотники. Цей чіп може запускати власні системи, візуалізацію та глибинну нейронну мережу на EDGE пристроях без підключення до мережі або будь-якої хмари. Цей чіп можна розгорнути на роботах, розміщених в рятувальних операціях на постраждалих від катастрофи місцях. Рятувальний робот може приймати якісь рятувальні рішення без допомоги людини. Він може запускати глибокі нейронні мережі в режимі реального часу, виконуючи їх з ефективністю 100 гігафлопів на потужність 1 Вт. Movidius Myriad 2 – це модуль обробки зору другого покоління VPU та Myriad X VPU – це найдосконаліший VPU від Movidius, який пропонує рішення щодо штучного інтелекту від безпілотників та робототехніки до розумних камер та віртуальної реальності. Intel також

пропонує розробку Myriad Kit (MDK), який включає всі необхідні інструменти та API для реалізації ML на чіпі.

Intel Movidius Neural Compute Stick – це USB-накопичувач, який розширив та ж технологія плати Intel Myriad (SoC). Цей пристрій для підключення та відтворення можна легко приєднати до EDGE пристроїв працює під управлінням Ubuntu 16.04.3 LTS (64 біт), CentOS * 7.4 (64 біт), Windows 10 (64 біт), Raspbian, включаючи Raspberry Pi, Intel NUC, персональний комп'ютер тощо. Цей пристрій має Intel Movidius Myriad X Vision Процесор (VPU) процесор, який підтримує TensorFlow, Caffe, Apache MXNet, Open Neural Network Exchange (ONNX), PyTorch та PaddlePaddle шляхом перетворення ONNX.

BeagleBone AI – це висококласна дошка для розробників, які будують системи машинного навчання та комп'ютерного зору. Цей пристрій працює від двоядерного процесора Cortex-A15 SoC - TI AM5729, який містить 4 програмовані блоки реального часу, двоядерний цифровий сигнальний процесор C66x та 4 ядра з підтримкою через API машинного навчання TIDL (Texas Instruments Deep Learning). Він може виконувати класифікацію зображень, виявлення об'єктів та семантичну сегментацію за допомогою TIDL.

ECM3531 – це високоефективний ASIC на основі процесорів ARM Cortex-M3 та NXP Coolflux DSP для додатків машинного навчання. Назва процесора цього ASIC – Tensai, який може запускати TensorFlow або Caffe моделі. Цей процесор пропонує 30-кратне зменшення потужності в класифікації зображень на основі CNN.

Пристрій SmartEdge Agile34 разом із супровідним програмним забезпеченням Brianium допомагає побудувати моделі штучного інтелекту та розгорнути їх на обмежених ресурсами пристроях. Agile SmartEdge

застосовується у EDGE середовищі і використовує платформу Brainium з нульовим кодуванням для розгортання навченої моделі на EDGE.

2.1.2 Field Programmable Gate Arrays (FPGAs)

Microsoft Brainwave Brainwave – це спроба використовувати технологію FPGA для вирішення завдань AI у режимі реального часу та запускати моделі глибокого навчання в хмарі Azure та на EDGE в режимі реального часу. Brainwave використовує Intel Stratix 10 FPGA як серце в системі, що забезпечує 39.5 TFLOPs надзвичайно високої ефективності. Найпопулярніші моделі глибокого навчання, включаючи ResNet 50, ResNet 152, VGG-16, SSD-VGG, DenseNet-121 і SSD-VGG підтримуються FPGA Brainwave на Azure для задач класифікації зображень та завдання виявлення об'єктів. Lazur може паралелізувати попередньо підготовлену глибинну нейронну мережу (DNN) через FPGA, щоб масштабувати будь-яку службу.

Процесор ARM ML дозволяє розробникам прискорити роботу алгоритмів ML та розгортати додатки на EDGE пристроях. Екосистема складається з наступних компонентів:

• ARM NN35, двигун, який забезпечує шар перекладу для усунення розриву між існуючим фреймворками та процесором ARM ML;

• ARM Compute Library36 – бібліотека з відкритим кодом, що містить функції, оптимізовані для процесорів ARM.

Процесор ARM ML може запускати нейромережеві моделі фреймворків високого рівня, такі як TensorFlow Lite, Caffe та ONNX. ARK NN SDK має всі необхідні інструменти для запуску нейронних мереж на EDGE пристроях. Цей процесор розроблений для мобільних телефонів, AR / VR, робототехніки та медичних інструментів.

2.1.3 Вбудовані GPUs

Raspberry Pi – це одноплановий комп'ютер Raspberry Pi, розроблений Фондом Raspberry Pi. Є одним із найпоширеніших пристроїв, що використовуються для EDGE обчислень. Він використовується для запуску додатків ML без зайвих додаткових дій з налаштування обладнання. Raspberry Pi 3 Model B має процесор Quad Cortex A53 @ 1,2 ГГц, графічний процесор VideoCore IV 400 МГц, 1 Гб SDRAM. Raspberry Pi 3 використовується як EDGE пристрій для розробки системи спостереження за людьми в реальному часі. Система здатна в режимі реального часу розрізняти людські та нелюдські об'єкти. Пристрій має слот для мікро-SD-карти для підтримки флеш-пам’яті до 32 ГБ. Xnor.ai розробив нову платформу AI для ефективного запуску моделей глибинного навчання на EDGE пристроях, таких як вбудовані процесори (наприклад, Raspberry Pi), телефони, IoT-пристрої та безпілотники без використання GPU чи TPU.

Nvidia Jetson – це вбудована обчислювальна плата, яка може обробляти складні дані в режимі реального часу. Jetson AGX Xavier може працювати з джерелом живлення потужністю 30 Вт і працювати як робоча станція GPU.

Jetson TX1 і TX2 – це вбудовані обчислювальні пристрої, що працюють на основі чіпу Nvidia Jetson. Ці два маленькі, але потужні комп’ютери ідеально підходять для впровадження інтелектуальної системи на EDGE пристроях, таких як смарт-безпека, камери, безпілотники, роботи та портативні медичні пристрої.

JetPack – це SDK для створення AI-програм із Jetson. Цей SDK включає TensorRT, cuDNN, Nvidia DIGITS Workflow, підтримує ISP, зображення камер, Video CODEC, Nvidia VisionWorks, OpenCV, Nvidia Інструменти CUDA та бібліотеки CUDA для підтримки ML. Він також сумісний з операційною системою ROS.

OpenMV Cam OpenMV Cam39 – це невелика плата для камер малої потужності. Ця плата побудована за допомогою процесора ARM CortexM7 для виконання алгоритмів машинного зору на 30 FPS. Цей процесор може працювати на 216 МГц і має 512 КБ оперативної пам’яті, 2 Мб флеш-пам’яті та 10 контактів вводу / виводу. Основні програми цього пристрою – це розпізнавання обличчя, відстеження очей, виявлення / декодування QR-коду, розрізнення кадру та виявлення рядків.

2.1.4 RISC-V

RISC-V – це відкрита архітектура наборів інструкцій (ISA), а GAP8 – архітектура RISC-V мікропроцесору, який застосовувався для обчислювальних елементів. Він має 9 ядер, здатних працювати з потужністю 10 ГОПС в порядку десятків мВт. Цей процесор на 250 МГц призначений для прискорення CNN для EDGE обчислень та ринку IoT. Greenwaves розробив інструмент TF2GAP8, який автоматично перекладає TensorFlow CNN моделі у формат GAP8 [8].

2.2 Алгоритми стиснення нейронних мереж

Додаткового огляду потребують також алгоритми, що дозволяють стискати нейронні мережі без втрати або з мінімальною втратою якості. Це особливо важливо в контексті запуску нейронних мереж на пристроях з обмеженими обчислювальними ресурсами.

2.2.1 Обрізка нейронних мереж

Стиснення нейронних мереж за допомогою методів обрізки широко вивчається. Ці методи дозволяють видаляти параметри мережі, які не потрібні

для гарного результату. Рання робота в цій галузі була спрямовані на зменшення складності та уникнення проблеми перенавчання в мережах. У цих роботах автори використовували методи обрізки на основі гессіанської функції втрат для зменшення кількості з'єднань всередині мережі. Метод знаходить набір параметрів, видалення яких викликає найменше збільшення цільової функції шляхом вимірювання впливу цих параметрів. Для пошуку цих параметрів автори використовують численні наближення. Наприклад, цільову функцію наближає серія Тейлора. Пошук параметрів, видалення яких не збільшує цю функцію, є складною проблемою, що стосується, наприклад, обчислення величезних матриць, а також другої похідної. Крім того, це зменшує складність мережі та перенавчання. Однак обчислення похідних вводить серйозні обчислювальні витрати. Існують і більш ефективні методи обчислення невпливових ваг. Перший крок – це вивчення зв’язності мережі за допомогою звичайного навчання. Метою цього є знання інформації, які параметри або зв’язки важливіші за інші. Наступний крок полягає в обрізання цих з'єднань та ваг, що нижче порогу. Тобто перетворення щільної мережі в розріджену. Далі, важливим кроком цього методу є перенавчання мережі для вивчення ваг решти розріджених з'єднань. Якщо обрізана мережа не буде перетренованою, то отримана точність вийде значно нижчою.

Однак обрізка має недолік побудови мережі, що має “нерегулярні” з'єднання, що в результаті впливає на паралелізацію обчислень. Щоб уникнути цієї проблеми, для CNN запроваджують структуровану розрідженість у різних масштабах. Таким чином, обрізка виконується за: картою характеристик, рівнем ядра та внутрішньоядерним рівнем. Ідея полягає в тому, щоб звести деякі ваги до нуля, але також використовувати обмеженість у чітко визначених місцях активації в мережі. Техніка складається у обмеженні кожного вихідного з'єднання згортки для карти фіч джерела, з метою надати схожий крок і зміщення. Це призводить до значного зменшення як ознак, так і матриці

ядра. Зазвичай розрідженість вивчалася в численних роботах з метою покарання несуттєвих параметрів.

Нещодавній метод обрізки полягає у видаленні фільтрів які, як доведено, мало впливають на остаточну точність мережі. Це призводить до автоматичного видалення фільтрів відповідно до карти фіч та пов’язаного з нею ядра в наступному шарі. Відносна важливість фільтра в кожному шарі вимірюється шляхом обчислення суми його абсолютних ваг, що дає очікування величини виходу карта фіч. При кожній ітерації фільтри з найменшими значення обрізаються.

Існують численні методи обрізки, і в кожного з них є сильні і слабкі місця. Основний недолік цих методів полягає у тому, що обрізка мереж потребує тривалого часу для постійного перенавчання, якого вони вимагають. Останні методи намагаються обійти деякі кроки, обрізаючи нейронні мережі під час їх навчання, використовуючи періодичні нейрони мереж. Однак усі вони призводять до значного зниження параметрів. Способи обрізки дозволяють усунути 10-30 відсотків від ваги мережі. Незалежно від методів, розмір мережі можна зменшити за допомогою обрізки без змін або значного падіння точності. Виконання з отриманими моделями також буде швидшим, але реальна швидкість залежить від способу, яким була розріджена мережа після обрізки.

2.2.2 Дистиляція знань

Щоб створити нейронну мережу, важливо оцінити, наскільки глибокою мережа повинна бути. Нейронна мережа складається з вхідного, вихідного та проміжного шарів. Неглибока нейромережа – це мережа з меншою кількістю проміжних шарів на відміну від глибокої нейронної мережі. Більш глибока мережа має більше параметрів і потенційно може вивчити більш складні функції, наприклад ієрархічні уявлення. Теоретична робота виявила труднощі,

пов’язані з навчанням неглибокої нейронної мережі з такою ж точністю, як і глибока мережа. Однак була спроба створити меншу мережу для класифікації датасету Imagenet на фічах екстрактора SIFT. Автори дійшли висновку, що навчити високоточну неглибоку нейронну мережу було складним завданням.

Незважаючи на це, з часом з’явилися повідомлення, що нейронні мережі з більш дрібною архітектурою здатні вивчити те саме, що й глибокі мережі, з кращою точністю, а іноді і з однаковою кількістю параметрів. Дистиляція знань полягає в навчанні меншої моделі функції, що засвоїла більша модель. Попередній крок полягає в тому, щоб навчити глибоку мережу (учительська мережа) генерувати автоматично мічені дані шляхом надсилання даних без маркування через цю глибоку мережу. Далі цей "синтетичний" набір даних використовується для навчання меншої мімічної моделі (студентської мережі), яка засвоює вивчену більшою моделлю функцію. Очікується, що мімічна модель повинна виробляти ті ж прогнози та помилки, що і глибша мережа. Таким чином, аналогічна точність може бути досягнута між ансамблем нейронних мереж та його імітаційною моделлю з меншою у 1000 разів кількістю параметрів.

2.2.3 Квантування

Квантування в мережі подібне до обрізки, оскільки це поширена методика у спільноті глибокого навчання. Це спрямовано на зменшення кількості бітів, необхідних для представлення кожного набору ваг. Іншими словами, квантування зменшує кількість параметрів, використовуючи надмірність. Квантування зменшує розмір пам’яті з мінімальними втратами продуктивності. У нейромережі це означає, що параметри будуть укладені в кластери. В результаті параметри в одному кластері будуть поділяти одне і те ж значення.

Обрізка та квантування – це методи які часто використовуються разом для досягнення сильного стиснення мережі. Наприклад, для мережі, подібної до LeNet5, обрізка і квантування стиснули модель у 32 рази і с кодування Хаффмана навіть 40 разів.

У нейромережах можна застосувати кілька методів квантування. Крім того, з метою квантування використовується кластеризація, пов'язана з значенням Гессіана, для мінімізації втрат продуктивності. Останні оптимізатори нейронних мереж (Адам, AdaGrad, Ададельта або RMSProp) можуть запропонувати альтернативи гессіані та таким чином зменшити загальну вартість обчислень. Однак однією з переваг використання методу зваженої Гессіани є те, що параметри всіх шарів в нейронній мережі можуть бути квантовані разом одночасно порівняно з попередніми методами, що квантують шар за шаром.

Методи квантування є ефективними, оскільки вони досягають вражаючих коефіцієнтів стиснення і можуть поєднуватися з іншими методами подальшого стиснення моделей. Завдяки їх ефективності вони інтегровані у деякі фреймворки та інструменти для прямої кількісної оцінки мережі та перенесення її на мобільні пристрої.

2.2.4 Зменшення чисельної точності

Хоча кількість ваг можна значно зменшити за допомогою методів обрізки або квантування, загальна кількість параметрів і дороге матричне множення може все ще бути величезним. Рішення полягає в зменшенні обчислювальної складності шляхом обмеження числової точності даних. Глибинні нейронні мережі, як правило, навчаються за допомогою 32-бітна точності з плаваючою комою для параметрів та активацій. Мета – зменшити кількість використаних бітів (16, 8 або навіть менше) та переходити від представлення з плаваючою комою до подання на фіксовану точку. Вибір

точності даних завжди був фундаментальним вибором, коли мова йде про вбудовані системи. Якщо вони присвячені певній системі, моделі та алгоритми можуть бути оптимізовані для конкретної обчислювальної та архітектурної пам'яті пристрою.

Однак застосування зменшення розмірності для глибоких нейронних мереж є складним завданням. Помилки зменшення розмірності може бути розповсюджена та посилені по всій моделі, і, таким чином, мати великий вплив на загальну ефективність роботи. З початку 90-х були зроблені експерименти для того, щоб обмежити точність даних у нейронній мережі, особливо під час зворотного розповсюдження. Були створені алгоритми зворотного розповсюдження з 24-розрядними одиницями обробки з плаваючою комою. Крім того, було емпірично доведено, що лише 8-16 біт достатньо для навчання за допомогою розмноження. Тим не менш, навіть якщо всі ці роботи допомагають зрозуміти вплив обмеженої чисельної точності на нейронні мережі, вони зроблені на досить невеликих моделях, таких як багатошаровий перцептрон лише з одним прихованим шаром і. Потрібні більш складні алгоритми для більш складних глибоких моделей.

У 2015 році були натреновані глибокі CNN, використовуючи 16-бітову фіксовану точка замість 32-бітової точності з плаваючою точкою. Це обмежувало параметри нейронних мереж, таких як зміщення, ваги та інші змінні, що використовуються під час множення, таких як активація, помилка розповсюдження, вага оновлення та зміщення оновлень. Існують різні експерименти зроблені з цією 16-бітовою довжиною чисел з фіксованою точкою, наприклад варіювання кількості бітів, що кодують дробові (цілі числа).

2.2.5 Бінаризація

В останніх роботах обмежена числова точність була розширена до двійкових операцій. У бінарній мережі ваги та активації принаймні обмежуються або +1, або -1. Дотримуючись тієї ж ідеї, що і раніше з обмеженою числовою точністю, ті ж автори вирішили застосувати дві схеми округлення для бінаризації змінної: детерміноване та стохастичне округлення. Найпоширеніший метод округлення – це підтримка знака змінної.

2.3 Актуальність сфери та виділення проблем

Машинне навчання на EDGE пристроях – це науково-дослідна область, що швидко розвивається, з численними викликами та можливостями. Практика демонструє, що кращі пристрої для машинного навчання покращують не тільки конфіденційність та безпеку користувача, але й час реакції системи.

Як бачимо в останні роки розвиваються не лише хмарні методи для запуску нейронних мереж, але й нейронні мережі для EDGE компонентів. Примітно, що розробка йде у декількох напрямках, таких як створення компактніших нейронних мереж, оптимізація існуючих нейронних мереж, створення спеціальних апаратних прискорювачів, інфраструктури.

Все це демонструє, що запуск нейронних мереж у EDGE парадигмі відіграє величезну роль і впливає не лише на розвиток штучного інтелекту, нейронних мереж, але і на галузі інтернету речей.

В той же час настільки бурхливий розвиток призводить і до виникнення певного ряду проблем. При настільки швидкому розвитку технології індустрія стикається з проблемою, коли вона не може наздогнати технології у питаннях адаптації до реального світу. А це у свою чергу гальмує дослідження у цій сфері.

Окремою проблемою є розробка систем, що базуються на EDGE технології. Оскільки нові досягнення відкриваються ледь не щомісяця, а нові технології з’являються частіше ніж кожні півроку, виникає серйозна прогалина між попитом та пропозицією, а це призводить до суттєвого завищення вартості створення систем.

3 IAAS ТА PAAS ПАРАДИГМИ

Основні провайдери хмарних технологій - AWS, Microsoft Azure та Google Cloud Platform - пропонують вражаючий набір послуг, розроблених для того, щоб допомогти підприємствам розміщувати та керувати програмами будь-яких форм і розмірів у віртуалізованому легко масштабованому середовищі. Ці пропозиції класифікуються як інфраструктура як послуга (IaaS) або платформа як послуга (PaaS), згідно з якими структура програма поділяється на частини, якими керує постачальник і які обробляє користувач.

Хоча обидві ці хмарні сервісні моделі забезпечують всі переваги хмарних обчислень за допомогою гнучких масштабованих методів підписки, одна набагато більш орієнтована на особливості, а інша - надає лише мінімально необхідне та орієнтована на рентабельність.

3.1 IaaS парадигма

IaaS робить основні структури обчислювального середовища, а саме фізичні компоненти, такі як сервери, доступними через Інтернет за моделлю оплати за використання. Ці віртуалізовані середовища практично ідентичні локальним серверам, але дозволяють підприємствам скористатися усіма основними перевагами хмарних обчислень: еластичністю, безпекою, доступністю, продуктивністю, економічністю тощо.

Як видно з рисунку 3.1, IaaS провайдер відповідає за основи: сервер, мережа, безпека, зберігання. Робота в IaaS насправді не відрізняється від роботи над підготовчими етапами, принаймні функціонально - єдиною реальною різницею є те, що хтось інший відповідає за підтримку та безпеку фізичних компонентів вашого оточення. Інфраструктура як послуга забезпечує вам віртуалізоване середовище, яке робить саме те, що було б на

локальних серверах, але з усіма перевагами публічної хмари (доступність, продуктивність, економічність тощо).

Рисунок 3.1 - Порівняння парадигм IaaS та PaaS

3.1.1 Переваги використання IaaS

1. Платформи IaaS дозволяють організаціям створити інфраструктуру, просто запустивши сценарії. Це включає не лише розгортання віртуальних серверів, але й попередньо налаштовані бази даних, системи зберігання даних, балансири навантажень, мережеву інфраструктуру тощо.

2. Залежно від вимог бізнесу, IaaS дозволяє збільшувати або зменшувати ресурси. Завдяки такій гнучкості для масштабування інфраструктури бізнес може реагувати на можливості та виклики, що постають на їх шляху.

3. Модель IaaS полегшує підприємствам значно економити на CapEx та OpEx. Малий бізнес, такий як стартапи, може починати з невеликої інфраструктури, а потім масштабуватись по мірі розвитку бізнесу.

4. IaaS пропонує покращене відновлення після аварій. Для розподілених інфраструктур відновлення після аварій та безперервність бізнесу складають додаткові витрати та проблеми управління. IaaS вирішує цю проблему, надаючи зведене вікно для доступу до інфраструктури через Інтернет.

3.1.2 Недоліки IaaS

1. Безпека. Підприємство не має контролю над хмарною безпекою в середовищі IaaS. Потрібно переглянути угоду про рівень послуг постачальника хмарних послуг (SLA), щоб зрозуміти свої зобов’язання щодо безпеки і тим самим виявити прогалини в охопленні їх безпеки.

2. Відсутність гнучкості. Постачальники послуг підтримують програмне забезпечення, але вони не оновлюють програмне забезпечення для деяких підприємств.

3. Технічні проблеми. Організації стикаються з деяким простоєм з IaaS, і це обмежить їх доступ до додатків та даних.

4. Залежність. Наявність IaaS у вашій організації означає повну залежність від постачальника або третьої сторони ваших даних.

5. Оновлення та обслуговування. Організація несе повну відповідальність за будь-які оновлення програмного забезпечення та обслуговування інструментів або системи даних.

6. Послуги з віртуалізації та конфіденційність користувачів. IaaS залежить від послуг з віртуалізації. Також обмежує конфіденційність та налаштування користувачів[15].

3.1.3 Використання IaaS

1. Обчислення високої потужності. Завдяки IaaS виконання завдань, що вимагають великої кількості обчислювальної потужності, таких як моделювання, прогнозування та аналіз даних, є доступним як ніколи. Такі складні операції вимагали надмірно дорогої фізичної інфраструктури, але тепер, коли інфраструктура доступна для оренди за моделлю «оплата часу / дозування», можна отримати необхідну потужність за доступною вартістю.

2. Тестування. IaaS постачає усе необхідне для швидкого запуску тестового середовища з нуля, яке можна легко зменшити або видалити, коли воно більше не потрібно.

3. Зберігання. Однією з головних переваг ведення бізнесу з хмарою є те, що IaaS пропонує необхідний об’єм для зберігання даних. Крім того, публічні постачальники хмарних технологій зберігають дані у центрах обробки даних, які є набагато безпечнішими, ніж на локальному сервері.

4. Lift-and-shift міграції. Міграція додатку з попереднього середовища як є, не вносячи істотних змін до коду, та розміщення його в IaaS - це найшвидший і найпростіший спосіб запустити його в хмарі.

3.1.4 Приклади використання IaaS

Існує багато прикладів постачальників і продуктів IaaS. AWS пропонує такі послуги зберігання, як Simple Storage Services (S3) та Glacier, а також послуги обчислення, включаючи Elastic Compute Cloud (EC2). GCP пропонує послуги зберігання та обчислення через Google Compute Engine (GCE), як і Microsoft Azure.

Це лише невеликий зразок широкого спектру послуг, пропонованих великими постачальниками послуг IaaS. Сервіси можуть включати serverless функції, такі як AWS Lambda, Azure Functions або Google Cloud Functions; доступ до бази даних; обчислювальні середовища великих даних; моніторинг.

На ринку IaaS також є багато менших провайдерів, зокрема Rackspace Managed Cloud, CenturyLink Cloud, DigitalOcean.

3.2 PaaS парадигма

У моделі хмарних обчислень PaaS передбачені апаратні та програмні засоби, в першу чергу для розробки додатків. У цьому випадку постачальник хмарних послуг надає платформу з обладнанням та програмним забезпечення та робить її доступною для користувачів через Інтернет. Це не тільки звільняє організацію від інвестицій в апаратне і програмне забезпечення для запуску нового додатка (операційна система, веб-сервери, бази даних та доступ до середовища виконання мови програмування тощо). Окрім цього, продукти PaaS дозволяють команді розробників співпрацювати та працювати разом, незалежно від їх фізичного розташування.

3.2.1 Переваги використання PaaS

1. Зменшення накладних витрат. PaaS надає можливість зменшити накладні витрати надаючи інфраструктуру, фізичні ресурси, нові інструменти та обладнання тощо.

2. Плата за потребою. Фаза впровадження та тестування - фази, що вимагають більше витрат, ніж зазвичай, тому що в цей період можуть знадобитися зміни та розробка додаткових функції. В PaaS потрібно платити лише за ті ресурси, які наразі використовуються. Моделі ціноутворення PaaS повністю залежать від потреб та вимог.

3. Більша результативність із меншою кількістю коду. Пропозиції PaaS зазвичай постачаються із вбудованою бібліотекою заздалегідь закодованих компонентів, таких як воркфлоу, пошукова система та каталогізація - виключаючи велику кількість ручного кодування.

4. Швидка розгортка. Крім вищезгаданих вбудованих компонентів, PaaS пропонує ряд переваг, що економить час. Наприклад, можливість одночасно розвиватися для декількох платформ. Не кажучи вже про те, що PaaS усуває необхідність створювати середовище розробки, а значить, команди можуть відразу почати працювати над проектом.

5. Легка співпраця. Коли середовище розробки розміщується в хмарі, командам набагато простіше працювати разом, незважаючи на фізичні обмеження, такі як віддалена робота або розподілені офіси.

6. Ефективні процеси. Пропозиції PaaS підтримують повний життєвий цикл веб-додатків у єдиному інтегрованому середовищі, тобто створення, тестування, розгортання та оновлення відбуваються в єдиному місці.

3.2.2 Недоліки PaaS

1. Залежність від постачальника. З одного боку, великою перевагою є те, що певну частину роботи виконує постачальник. З іншого боку, бізнесом

все ще керуватимуть функціональні можливості постачальника, швидкість та надійність.

2. Сумісність існуючої інфраструктури. Нова платформа - це нове середовище, в якому застарілі рішення повинні продовжувати працювати. Безперечно, деякі труднощі та суперечності можуть виникнути при контакті двох систем. Тому важливо заздалегідь зрозуміти можливі проблеми сумісності та підготуватися до їх вирішення.

3. Безпека. Як правило, програмне забезпечення PaaS доступне в публічному середовищі, де кілька кінцевих користувачів мають доступ до одних і тих же основних ресурсів. Для деяких додатків, які містять конфіденційні дані або суворі вимоги дотримання, це не є вдалим варіантом [16].

3.2.5 Найкращі практики PaaS

Середовище PaaS спирається на спільну модель безпеки. Постачальник забезпечує безпеку інфраструктури, тоді як клієнти PaaS несуть відповідальність за захист своїх облікових записів, додатків та даних, розміщених на платформі. В ідеалі безпека зміщується від базової моделі до моделі безпеки периметра ідентичності.

Це означає, що клієнт PaaS повинен більше орієнтуватися на ідентичність як основний периметр безпеки. Питання, на яких слід зосередитись, включають захист, тестування, код, дані та конфігурації, службовців, користувачів, автентифікацію, операції, моніторинг та логування.

1. Захист програм від поширених та несподіваних атак. Один з найкращих підходів - це розгортання автоматичного рішення захисту в режимі реального часу з можливістю швидкого та автоматичного виявлення та блокування будь-якої атаки (один з прикладів атаки наведено на рисунку 3.2). Абоненти PaaS можуть використовувати засоби захисту, що надаються на

платформі, або шукати сторонні параметри, що відповідають їхнім вимогам. Ідеальний інструмент повинен забезпечувати захист у режимі реального часу та автоматичне виявлення та блокування несанкціонованого доступу, атак чи порушень.

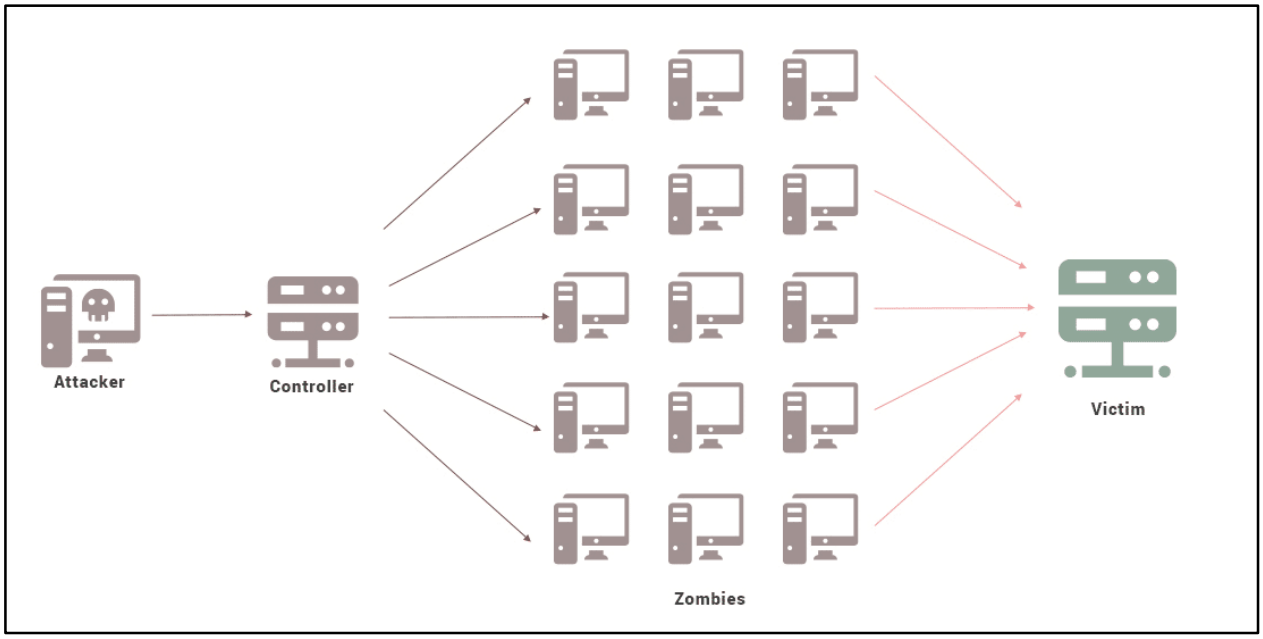

Рисунок 3.2 - Приклад атаки системи

Потрібно мати можливість перевіряти наявність незвичних дій, шкідливих користувачів, підозрілих входів, ботів та будь-якої іншої аномалії, яка може призвести до порушень. Окрім використання інструментів, потрібно передбачити засоби безпеки в додатку.

2. Захист облікових записів користувачів та ресурсів додатку. Кожна точка взаємодії зазвичай є потенційною поверхнею атаки. Найкращий спосіб запобігти атакам - зменшити або обмежити експозицію вразливих програм та ресурсів, до яких можуть отримати доступ недовірені користувачі. Важливо також регулярно та автоматично виправляти та оновлювати системи безпеки для зменшення недоліків. Потрібно забезпечити доступ до системи лише авторизованим користувачам або працівникам, а також тримати під контролем

кількість користувачів з правами адміністратора. При такому підході користувачі повинні мати лише найменші привілеї, які дозволяють їм мати доступ лише до необхідних і достатніх функцій програми. Це зменшує поверхню атаки, зловживання правами доступу та одержання привілейованих ресурсів.

3. Валідація даних. Валідація даних гарантує, що вхідні дані дійсні, безпечні та мають правильний формат. Перевірка даних гарантує, що проходять лише чисті дані, блокуючи компрометовані або заражені вірусом файли.

4. Автоматичний аналіз логів. Програми, API та системні логи надають багато інформації. Застосування автоматичного інструменту для збору та аналізу логів дає корисну інформацію про те, що відбувається. Найчастіше службі логування, якій доступні як вбудовані функції, так і сторонні додатки, чудово підтверджують відповідність політиці безпеки та іншим нормам. Потрібно використовувати аналізатор журналів, який інтегрується з системою оповіщення, підтримує використовувані стеки програм та забезпечує інформаційну панель тощо [17].

3.2.4 Приклади використання PaaS

Розповсюджений приклад PaaS – AWS Elastic Beanstalk. Веб-сервіси Amazon (AWS) пропонують понад 100 послуг хмарних обчислень, таких як EC2, RDS та S3. Більшість цих послуг можна використовувати як IaaS, і більшість компаній, які використовують AWS, виберуть потрібні послуги. Однак керування кількома різними сервісами може швидко стати складним та трудомістким для користувачів. На іншому боці AWS Elastic Beanstalk: він

працює як ще один шар поверх інфраструктурних сервісів і автоматично обробляє деталі забезпечення потужностями, балансування навантаження, масштабування та моніторинг статусу додатків.

3.2.5 Значення PaaS на світовому ринку хмарних технологій

PaaS забезпечує всі переваги та функціональність IaaS і більше. Він надає пакет керованих сервісів, призначених для економії часу на кодування та ручну роботу та пришвидшення процесу розробки від початку до кінця. Відповідно до висновків Gartner щодо основних тенденцій, що впливають на PaaS у 2019 році, наразі існує понад 550 пропозицій PaaS у 21 категорії. Очікується, що цей ринок удвічі збільшиться в 2018–2022 роках, що дозволить PaaS стати найпопулярнішою моделлю хмарних сервісів найближчим часом. Google Trends підтверджує прогноз Gartner, принаймні, щодо відносного інтересу як видно з рисунку 3.3.

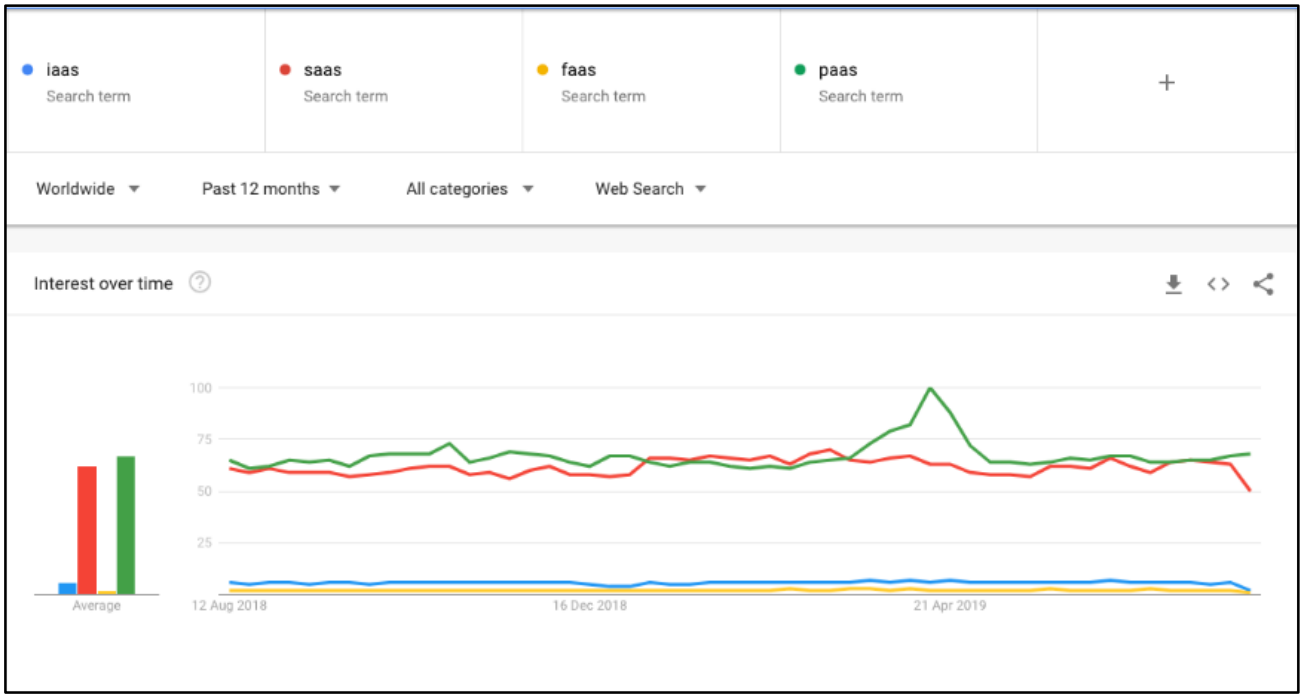

Рисунок 3.3 – Аналіз популярності парадигм IaaS, SaaS, FaaS та PaaS з часом

Тож основа порівняння PaaS та IaaS полягає в тому, що програми PaaS cloud-ready, а рішення IaaS - cloud-native. Cloud-ready додатки є масштабованим та слабко поєднаним, до того ж їх потрібно автоматизувати для розгортання. Cloud-native додатки було розроблено для хмарного середовища.

Модель PaaS зменшує складність розробки та розгортання додатку та вартість придбання, управління та обслуговування апаратного та програмного забезпечення, але покладає відповідальність за забезпечення безпеки облікових записів, додатків та даних перед клієнтом або передплатником. Для цього необхідний орієнтований на ідентифікацію підхід до безпеки.

Ефективні заходи забезпечення безпеки включають в себе захист додатків, забезпечення належного внутрішнього та зовнішнього захисту, а також моніторинг та аудит діяльності. Оцінка логів допомагає виявити вразливості безпеки, а також можливості покращення. В ідеалі розробники системи безпеки повинні спрямовуватись на подолання будь-якої загрози чи вразливості ще до того, як зловмисники їх побачать та використають.

Тож, хоча PaaS орієнтовані рішення менш гнучкі, ніж IaaS, PaaS безперечно виграє по швидкості розробки та розгортанню додатків, полегшує взаємодію всередині команди розробників і значно зменшує витрати.

4 РОЗРОБКА УНІФІКОВАНОЇ ІНФРАСТРУКТУРИ ДЛЯ ІНТЕЛЕКТУАЛЬНОЇ ОБРОБКИ ВІДЕОПОТОКІВ НА ПРИСТРОЯХ EDGE

Запуск нейронних на пристроях EDGE набуває широкої популярності. У той же час, як показано у другому розділі, стрімкий розвиток технології уповільнюється швидкістю її інтеграції у світ розробки програмного забезпечення.

Якщо розглядати розробку EDGE систем, то наразі більшість розробки відбувається з позиції EDGE системи як IaaS сервісу. Це дозволяє робити системи максимально гнучкими, проте мінусом є величезний кошт розробки. Альтернативою цьому підходу є розробка PaaS платформи для EDGE систем. Це означає, що більшість системи вже буде налаштованою, а розробникам або користувачам системи залишиться лише вказати параметри спілкування з хмарними сервісами та конфігураційні файли моделі, а також файл вагів самої моделі. В той же час це потребує формування спеціалізованої уніфікованої архітектури додатку зі спеціальними вимогами до цієї архітектури.

Серед вимог, що застосовуються до цієї системи:

1. Ефективність. Через те, що система націлена на вбудовані системи є надзвичайно високі вимоги до ефективності програм, що виконуються на пристрої, тому що пристрій має дуже обмежені ресурси.
2. Безпека. Безпека є дуже актуальним питанням для цієї системи, оскільки більшість пристроїв буде мати публічний інтерфейс. І хоча фізична безпека повністю знаходиться на плечах користувачів, оскільки самі пристрої будуть знаходитися у локаціях користувачів, інформаційна безпека має бути пріоритетом, оскільки часто мова може заходити про передачу чутливих до втрати даних. Іншим аспектом інформаційної безпеки є ізоляція системи від користувацького коду з метою захисту внутрішніх систем сервісу.

3. Надійність. Системи інтернету речей часто базуються на пристроях, до яких немає постійного доступу. Моніторинг також ускладнений, через потенційну ненадійність мережі для передачі даних.

4. Гнучкість. Система повинна мати змогу запускати нейронні мережі та алгоритми різних типів, надавати різні типи вихідних даних при різних структурах в рамках одного формату.

5. Можливість оновлення. Оскільки сфера розвивається стрімко, а оновлення парку пристроїв буде відбуватися значно повільніше, можливість оновлення є дуже важливою особливістю системи.

4.1 Аспект ефективності системи

Максимальна ефективність вбудованої системи може досягатися шляхом використання спеціальних низькорівневих бібліотек, а також низькорівневих мов програмування. Основним варіантом для вбудованих систем беззаперечно є мова програмування C за рахунок надзвичайно ефективного керування пам’яттю та надзвичайно ефективною трансляцією команд мови програмування у машинний код. Схожим до C варіантом є мова програмування C++, а також Rust. Проте через їх комплексність, а також функціонал, що обтяжує вбудовані системи, вони є менш популярними. За цими мовами програмування слідують Go та Python. Їх обирають заради простоти написання коду, а також відношення швидкість розробки до ефективності запуску. І хоч нативні імплементації Python не компілюють свій код, існують спеціальні версії мови програмування для вбудованих систем.

Отже, найкращим вибором для реалізації вбудованої системи є мова програмування C, проте надалі реалізація буде описуватися і відбуватися на мові програмування Python, оскільки час розробки системи на мові програмування C у декілька разів переважає час розробки відповідної системи мовою Python.

4.2 Аспект безпеки системи

На мережевому рівні одним із способів забезпечити безпечне та надійне з'єднання – це використання фізично захищеної мережі або VPN для всієї комунікації між клієнтами та брокерами. Це рішення підходить для шлюзових програм, де шлюз підключений до пристроїв з одного боку та до брокера через VPN з іншого боку. На жаль, використання такого підходу не завжди виправдане і можливе через специфіку розташування пристроїв у мережі.

На транспортному рівні, якщо конфіденційність є основною метою, TLS / SSL зазвичай використовується для шифрування трафіку. Цей метод є безпечним і перевіреним способом переконатися, що дані не можна читати під час передачі, і забезпечує аутентифікацію клієнтського сертифіката для підтвердження особи обох сторін.

На транспортному рівні комунікація шифрується і особистість ідентифікується. Протокол MQTT надає ідентифікатор клієнта та ідентифікатори користувача / пароля для автентифікації пристроїв на рівні програми. Ці властивості надає сам протокол. Авторизація або контроль того, що дозволяється робити кожному пристрою, визначається конкретною реалізацією брокера. Крім того, для захисту переданої інформації (без необхідності повноцінного шифрування транспорту) можна використовувати шифрування корисного навантаження на рівні програми [18].

Рішенням, що може покрити більшість з перерахованих моментів є використання програмного забезпечення з відкритим кодом, основаного на протоколі AMQP (Advanced Message Queue Protocol). Такі системи містять у собі вбудовані функції аутентифікації, шифрування та деяких інших елементів безпеки.

З метою забезпечити безпеку внутрішнім сервісам чудовим рішенням є використання технологій віртуалізації. Логічно це створить два різні незалежні системи, проте фактично це буде дві системи, розгорнуті на одному

комп’ютері. Суттєвим мінусом, що обмежує використання віртуалізації є проблеми з ефективністю системи, адже у такому випадку величезний ресурс буде виділений на підтримання роботи двох операційних систем на одному пристрої, що неприпустимо у випадку з вбудованими системами.

Ефективною альтернативою є використання технологій контейнеризації. Чудовим прикладом технології контейнеризації є Docker.

4.3 Аспект надійності системи

Надійність спілкування хмарою забезпечується протоколом AMQP, який дозволяє доставляти повідомлення з гарантією підтвердження.

Надійність самого додатку може забезпечуватися шляхом використання інструментів оркестрації контейнерів: Kubernetus або Docker Swarm.

Kubernetes підтримує більш високі вимоги з більшою складністю, тоді як Docker Swarm пропонує просте рішення, з якого можна швидко розпочати роботу. Docker Swarm був досить популярним серед розробників, які віддають перевагу швидкому розгортанню та простоті. Одночасно Kubernetes використовуються у виробничих середовищах різними високопрофільними інтернет-фірмами, що працюють за популярними послугами.

4.4 Аспект гнучкості системи та оновлення

Завдяки використанню інструментів Kubernetes чи Docker Swarm можна виконувати оновлення самих контейнерів, не ініціюючи оновлення самих систем. Це дозволить не лише надійно і швидко у стабільному процесі налагоджувати оновлення у цілому, але й дасть розробникам змогу користуватися знайомими і зрозумілими інструментами.

А завдяки протоколу AMQP, вдасться передавати по мережі не лише текстову інформацію, наприклад, JSON, але й невеликі файли та зображення.

4.5 Розробка архітектури додатку

В цілому архітектуру системи можна представити на рисунку 4.1.

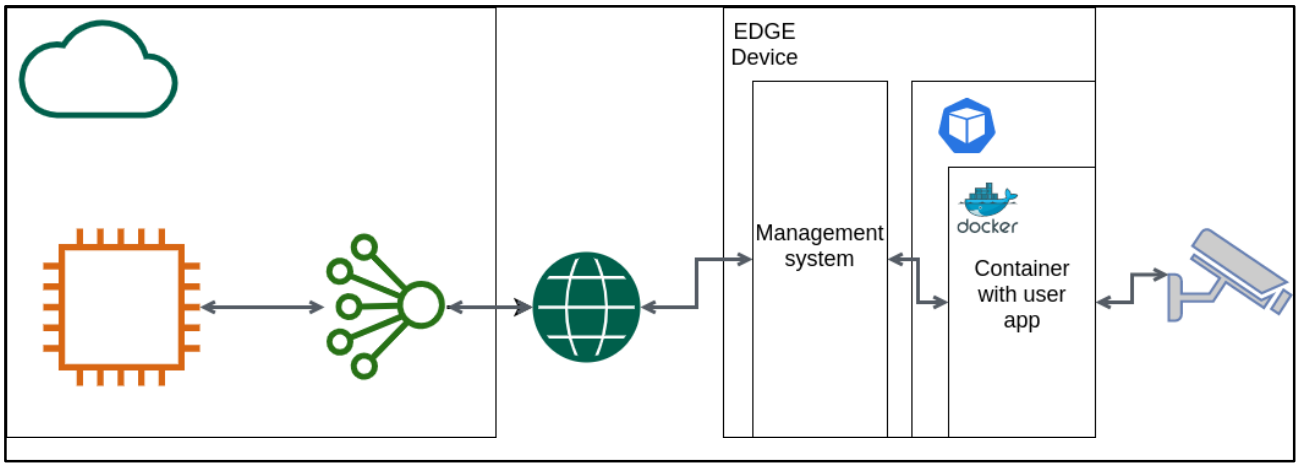

Рисунок 4.1 - Архітектура додатка

Хмарний сервіс приєднаний до сервісу обміном повідомленнями, який підтримує протокол AMQP, наприклад RabbitMQ чи MQTT. До цього сервісу обміну повідомленнями приєднаний через мережу клієнт, що знаходиться у блоці управління системою. Цей блок призначений для оновлення коду, керування контейнером, підтримки працездатності системи. Цей модуль пов’язаний з кодом користувачів, що знаходиться контейнері і у якому власне відбувається інтелектуальний аналіз відео. До цього ж блоку приєднаний інтерфейс камери або іншого пристрою для генерації зображень або відеопотоку. Після опрацювання фрейму інформацію передається назад у блок управлінню системою, а звідти через інтерфейс AMQP надсилається на сервер.

В результаті отримано систему, що задовольняє як функціональним вимогам, так і нефункціональним вимогам.

Окремої уваги заслуговує практична реалізація. Система побудована на мові програмування Python, з використанням інструменту контейнеризації Docker, реалізації протоколу AMQP у бібліотеці RabbitMQ.

У якості інструменту для запуску нейронних мереж був обраний Nvidia Jetson Nano з бібліотекою TensorRT, яка має Python інтерфейс.

Цей вибір зроблений не випадково, адже саме пристрої Nvidia розвиваються чи не найшвидше і мають найбільш широкий інструментарій у порівнянні з конкурентами.

ВИСНОВКИ

В ході написання дипломної роботи були розглянуті основні задачі, які вирішуються на EDGE системах з інтелектуальної обробки відео. Розглянуті і описані сучасні прискорювачі, що використовуються для запуску нейронних мереж на пристроях інтернету речей. Проаналізовані недоліки та переваги PaaS та IaaS парадигм. Проаналізована можливість застосування PaaS підходу до розробки систем інтелектуального відеоаналізу на EDGE системах.

В результаті було виявлено, що надалі частіше будуються системи, в яких більша частина обчислень відбувається на EDGE системах. В той же час розробка таких систем відбувається з чистого листа, хоча і більшість цих систем має схожу архітектуру. Це спонукало на створення уніфікованої системи, де єдиними змінними компонентами є алгоритми аналізу відео і формат вихідних повідомлень, що передається у хмару.

Було спроектована архітектура такої системи, що окрім безпосереднього аналізу відео відповідає і іншим вимогам – безпеці, надійності, гнучкості, можливості оновлень і ефективності. Кожна із цих нефункціональних вимог змотивована сучасними трендами у розробці систем, а також обмеженістю ресурсів.

Запропонована модель складається з мінімальної кількості блоків, які за потреби можуть бути розширені додатковими. Проектування і, найголовніше, виділення цих блоків є одним із потенціальних шляхів покращення роботи. Окрім цього у майбутньому слід приділити увагу і оптимізації окремих компонент. Наприклад, декодинг та енкодинг відео може оптимізуватися відповідно до пропускної здатності та ефективності алгоритму. Також слід оцінити нові способи компресії даних для пересилання через мережу у хмару.

Також запропонована система може бути успішним комерційним продуктом, оскільки покликана знизити кошти на розробку інтелектуального аналізу відеопотоку. Використані технології є сучасними, ефективними, а інструменти, завдяки яким може відбуватися використання є добре знайомими, зручними та надійними. Усе це у купі дозволяє стверджувати про потенційний комерційний успіх запуску такої системи.

Загалом, мета роботи була досягнута. Також були намічені шляхи подальшого розвитку та вдосконалення спроектованої та розробленої системи.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. AlexNet [Електронний ресурс] / URL: https://en.wikipedia.org/wiki/AlexNet / 15.05.2020
2. Тіягі, В. Understanding Digital Image Processing / В. Тіягін. - Ф.: CRC Press, 2018 - 30 с. / URL: https://www.researchgate.net/publication/328120952_Understanding_Digital_Image_Processing
3. Ванг, Л. Deep Learning Algorithms with Applications to Video Analytics for A Smart City: A Survey / тез. докл. науч.–практ. конф. (грудень 2015) - 8 с. / URL: https://arxiv.org/pdf/1512.03131.pdf
4. Кхан, А. A Survey of the Recent Architectures of Deep Convolutional Neural Networks / А. Кхан, А. Сохаил, У Захура, А.С. Кураши // Artificial Intelligence Review. - 2020. - 70 с. / URL: https://arxiv.org/pdf/1901.06032.pdf
5. Чапароне, Д. Deep Learning in Video Multi-Object Tracking: A Survey / Д. Чапароне, Ф.Л. Санчез, С. Табік, Л. Трояно, Р. Тагліаферрі, Ф. Херрера // тез. докл. науч.–практ. конф. (листопад 2019) / Нью Йорк: Корнельський університет, 2019. - 29с. / URL: https://arxiv.org/abs/1907.12740
6. Карпати, А. Large-scale Video Classification with Convolutional Neural Networks / А. Карпати, Д. Тодерічі, С. Шеті, Т. Леюнг, Р. Суктанкар, Л. Фей, / тез. докл. науч.–практ. конф. (грудень 2019) - 8 с. / URL: https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/42455.pdf
7. Ліанг, К. AI on the Edge: Rethinking AI-based IoT Applications Using Specialized Edge Architectures / К. Ліанг, П. Шеной, Д. Ірвін / тез. докл. науч.–практ. конф. (грудень 2019) - 12 с. / URL: https://arxiv.org/pdf/2003.12488.pdf

8. Муршед, С. Machine Learning at the Network Edge: A Survey: тез. докл. науч.–практ. конф. (січень 2020) - 33 с. / URL: https://arxiv.org/pdf/1908.00080.pdf
9. Бертельєр, А. Deep Model Compression for Mobile Devices : A Survey / А. Бертельєр, П. Путане, С. Дафнер, К. Гарсиа, К. Бланк, Т. Чато // тез. докл. науч.–практ. конф. (листопад 2018) / Ліон: Ліонський університет, 2018. - 8 с. / URL: https://orasis2019.sciencesconf.org/253457/document
10. Едісон, А. Automated video analysis for action recognition using descriptors derived from optical acceleration / А. Едісон, С. Джиджи // Журн. Signal, Image and Video Processing. - 2019. - №13 - С. 915-922 / URL: https://link.springer.com/article/10.1007/s11760-019-01428-1
11. Kelvin Xu, Jimmy Lei Ba [и др.]. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. URL: https://arxiv.org/pdf/1502.03044.pdf (дата обращения: 20.05.2018).
12. Neal Wu, Модель на фреймворке Tensorflow для предсказания видео. // Github open source projects, Tensorflow models. URL: https://github.com/tensorflow/models/tree/master/research/video_prediction (дата обращения: 01.05.2020).
13. Li Xu, Ce Liu, Jimmy SJ. Ren, Jiaya Jia, Deep Convolutional Neural Network for Image Deconvolution. URL: https://papers.nips.cc/paper/5485-deep-convolutional-neural-network-for-image-deconvolution.pdf (дата обращения: 10.05.2020).
14. Chelsea Finn, Ian Goodfellow, Sergey Levine. Unsupervised Learning for Physical Interaction through Video Prediction. // Berkley & Google project. URL: https://arxiv.org/abs/1605.07157 (дата обращения: 15.05.2020).
15. Advantages and disadvantages of IaaS [Електронний ресурс] / URL: https://www.hitechnectar.com/blogs/advantages-disadvantages-of-iaas-explained

16. IaaS vs. PaaS: which one is best? [Електронний ресурс] / URL: https://sharegate.com/blog/iaas-saas-paas-faas-a-look-at-cloud-service-models#iaas-benefits
17. How to Secure Platform as a Service (PaaS) Environments [Електронний ресурс] / URL: https://geekflare.com/paas-security-tips/
18. Геверт, А. Embedded Wireless Communication. Connectivity of a smartphone with Bluetooth LE and UWB devices: автореф. дис.: 17.11.17 / А. Геверт; [Upsala university]. - У., 2017. - 55 с. / URL: http://www.diva-portal.org/smash/get/diva2:1182765/FULLTEXT01.pdf
19. Уравнение Фоккера-Планка. // Википедия – свободная энциклопедия. URL: https://ru.wikipedia.org/wiki/Уравнение_Фоккера_—_Планка (дата обращения: 20.05.2018).
20. Matthew D. Zeiler, Dilip Krishnan [и др.] Deconvolutional Networks URL: http://www.matthewzeiler.com/wp-content/uploads/2017/07/cvpr2010.pdf (дата обращения: 20.05.2018).
21. Joost van Amersfoort , Anitha Kannan, Marc'Aurelio Ranzato, Arthur Szlam, Du Tran & Soumith Chintala. Transformation-based models of video sequences // Facebook AI research. URL: https://arxiv.org/pdf/1701.08435.pdf (дата обращения: 20.05.2018).
22. Модель сети обратной свертки от Майкрософт. URL: https://docs.microsoft.com/en-us/cognitive-toolkit/Image-Auto-Encoder-Using-Deconvolution-And-Unpooling (дата обращения: 20.05.2018).
23. Christopher Olah, LSTM model description. // Personal scientitfic blog. URL: http://colah.github.io/posts/2015-08-Understanding-LSTMs