

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет
Кафедра

Комп'ютерної інженерії та управління
Комп'ютерних інтелектуальних технологій та систем

КВАЛІФІКАЦІЙНА РОБОТА **Пояснювальна записка**

рівень вищої освіти

другий (магістерський)

Методи обробки зображень на базі згорткових нейронних мереж

Виконав:

студент 2 курсу, групи КІТм-21-2

Шевченко Олексій Тарасович

Спеціальність 123 Комп'ютерна інженерія

Тип програми освітньо-професійна

Освітня програма Комп'ютерні
інтелектуальні технології

Керівник О.Г. Руденко

Допускається до захисту

Зав. кафедри

О.Г. Руденко

2022 р.

Харківський національний університет радіоелектроніки

Факультет	Комп'ютерної інженерії та управління
Кафедра	Комп'ютерних інтелектуальних технологій та систем
Рівень вищої освіти	другий (магістерський)
Спеціальність	123 Комп'ютерна інженерія
Тип програми	освітньо-професійна
Освітня програма	Комп'ютерні інтелектуальні технології

ЗАТВЕРДЖУЮ:

Зав. Кафедри

(підпис)

« ____ » _____ 2022 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Шевченко Олексій Тарасович

1. Тема роботи (проекту) Методи обробки зображень на базі згорткових нейронних мереж

затверджена наказом університету від "07" листопада 2022р. № № 1455

2. Термін подання студентом роботи до екзаменаційної комісії 15 грудня 2022р.

3. Вихідні дані до роботи (проекту) Науково-технічні публікації, інтернет-ресурси, цифрові зображення у відкритому доступі

4. Перелік питань, що потрібно опрацювати в роботі:

- аналіз предметної області;
- методи побудови застосунків з використанням ШНМ;
- інструменти для автоматизації процесу навчання моделі;
- комбінування методів згорткових мереж з іншими шнм.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням кафедри)

11 слайдів

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно до наказу, зазначеному у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз стану проблеми та методів рішення	08.11.2022-13.11.2022	
2	Аналіз методів згорткових нейронних мереж	14.11.2022-18.11.2022	
3	Аналіз типів комбінування	19.11.2022	
4	Проектування архітектури моделі	20.11.2022-29.11.2022	
5	Налаштування процесу навчання	30.11.2022-03.12.2022	
6	Поєднання проекту в тестову програму	04.12.2022-05.12.2022	
7	Тестування	06.12.2022-07.12.2022	
8	Оформлення матеріалів кваліфікаційної роботи	08.12.2022-10.12.2022	
9	Подання кваліфікаційної роботи керівникові та її попередній захист	11.12.2022-14.12.2022	
10	Подання кваліфікаційної роботи на рецензування	15.12.2022-17.12.2022	

Дата видачі завдання 7 листопада 2022 р.

Студент _____

Керівник роботи _____ проф. Руденко О.Г.

РЕФЕРАТ

Пояснювальна записка магістерської кваліфікаційної роботи: 62 сторінки, 13 рисунків, 1 таблиця, 1 додаток, 23 джерела.

Метою кваліфікаційної роботи є огляд методів обробки зображень на базі згорткових нейронних мереж.

Об'єктом дослідження є згорткові нейронні мережі.

Предметом дослідження є методи обробки зображень.

У ході виконання кваліфікаційної роботи були виконані наступні пункти:

- проаналізовано існуючі додатки, процес функціонування та архітектура згорткових нейронних мереж;
- розроблені тестові додатки детекції на базі CoAtNet;
- виявлені переваги комбінування різних методів обробки зображень;
- проведено тестування додатку.

Результатом є порівняння декількох методів обробки зображень на базі згорткових нейронних мереж та побудова тестового додатку за допомогою Keras.

ПРОГРАМНИЙ ЗАСІБ, ІНТЕРНЕТ, ВЕБ, НЕЙРОННА МЕРЕЖА, ДЕТЕКЦІЯ, РОЗПІЗНАВАННЯ, МАШИННЕ НАВЧАННЯ, KERAS, TENSORFLOW

ABSTRACT

Master's thesis: 59 pages, 13 figures, 1 table, 2 appendices, 23 sources.

The purpose of the qualification work is to review image processing methods based on convolutional neural networks.

The object of research is convolutional neural networks.

The subject of research is image processing methods.

In the course of the qualification work, the following items were completed:

- existing applications, operation process and architecture of convolutional neural networks were analyzed;
- developed test applications of detection based on CoAtNet;
- revealed advantages of combining different methods of image processing;
- the application was tested.

The result is a comparison of several image processing methods based on convolutional neural networks and the construction of a test application using Keras.

APPLICATION, INTERNET, WEB, NEURAL NETWORK, DETECTION, RECOGNITION, MACHINE LEARNING, KERAS, TENSORFLOW

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет
Кафедра

Комп'ютерної інженерії та управління
Комп'ютерних інтелектуальних технологій та систем

АНОТАЦІЯ

КВАЛІФІКАЦІЙНОЇ РОБОТИ

рівень вищої освіти

другий (магістерський)

Методи обробки зображень на базі згорткових нейронних мереж

Виконав:

студент 2 курсу, групи КІТм-21-2

Шевченко О. Т.

Спеціальність 123 Комп'ютерна інженерія

Тип програми освітньо-професійна

Освітня програма Комп'ютерні
інтелектуальні технології

Керівник професор Руденко О. Г.

2022 р.

АНОТАЦІЯ

Шевченко О.Т. Методи обробки зображень на базі згорткових нейронних мереж. – Магістерська кваліфікаційна робота.

У магістерській кваліфікаційній роботі вирішено актуальну проблему з поліпшення результатів CNN моделей за допомогою елементів з інших ШНМ.

Метою кваліфікаційної роботи є огляд методів обробки зображень на базі згорткових нейронних мереж.

Об'єктом дослідження є згорткові нейронні мережі.

Предметом дослідження є методи обробки зображень.

Для цього було проведено дослідження методів згорткових нейронних мереж для обробки зображень. Виконано системний аналіз існуючих архітектур та існуючих рішень. Тестовий застосунок показав доцільність використання підходів побудованих на комбінованій архітектурі.

Актуальність методів обробки зображень на базі згорткових нейронних є досі великою, тому що зараз не так багато методів для роботи з зображеннями. Кількість нових даних постійно зростає, що зумовлюється збільшенням різноманітних пристроїв для обробки зображень та сховищ даних. Велика кількість проблем наразі може вирішуватись автоматизацією ручної праці, теж стосується і проблем, де потрібна дуже велика концентрація уваги людини. Особливий інтерес представляє автоматизована класифікація зображень у медичній сфері, промисловості та у багатьох інших сферах, де можна отримати результат роблячи контроль знімками. Глибокі згорткові нейронні мережі (Convolutional Neural Networks, CNN) показують потенціал для аналізу зображень за допомогою сегментації та детекції. В даний час йде активний пошук концепцій, які в змозі виконувати завдання з обробки зображень, які можуть надаватись з різних джерел і великим рівнем накопичення даних, та вдосконалювати існуючі рішення для досягнення максимальних результатів.

У даній роботі були оглянуті певні згорткові нейронні мережі, щоб

дослідити їх ефективність на задачах класифікації та розглянути з гібридизації згортки та уваги від двох різних ШНМ. Були обрані елементи:

- згорткові шари, що мають кращу здібність до узагальнення;
- механізм уваги трансформер моделі, що має вищу ємність моделі та може принести користь з більших наборів даних.

Поєднання згорткового рівня та рівня уваги може досягти кращого узагальнення і місткості; однак ключовою проблемою тут є те, як ефективно поєднати їх для досягнення кращих результатів компроміс між точністю та ефективністю. Хоч трансформери викликають все більший інтерес до комп'ютерного зору, але вони все ще відстають від найсучасніших згорткових мереж, але деякі елементи з цієї архітектури можуть бути потужним інструментом для поліпшення результатів CNN. Для порівняння будуть використані одні з найрезультативніших CNN для обробки зображень.

У першому розділі розглянуто актуальність та область застосування додатків на базі згорткових нейронних мереж для обробки зображень. Хоч зараз це поширена архітектура за допомогою якої будують різні додатки для обробки зображень, але ця технологія пройшла довгий шлях, щоб набутися популярності та досягти результатів, які можуть задовольняти потреби користувачів. Нажаль раніше не було достатньої обчислювальної потужності, щоб обробити достатню кількість зображень та виділити звідти певну інформацію для складення певних ознак за допомогою яких можна класифікувати зображення або провести інших метод. Перспективними напрямком застосування згорткових нейронних мереж є інтелектуальні системи з однією мережею між ділянками міста для контролю та виявлення злочинців чи підозрілих предметів. У країнах з добре розвиненою інфраструктурою та великим рівнем цифровізації наразі впроваджуються такі заходи для збору інформації з відкритих джерел та обробки інформації. Також не потрібно забувати про обробку зображень у банківському секторі, де вже розпізнавання облич є необхідністю для ідентифікації особи. Також був розглянутий вплив на медичну сферу, де збір великих обсягів інформації допомагає у виявленні тяжких хвороб, які неможливо виявити в той час, коли робиться сканування або

рентген.

У другому розділі розглянуті етапи процесу роботи архітектури CNN та методи роботи згорткових нейронних мереж для обробки зображень. Детально розглянуті приклади використання згортки. Цей механізм потрібен для обробки зображення з вхідного шару, щоб застосувати до них фільтри, це можливо завдяки поелементному множенню матриці вхідного значення та матриці фільтру отримавши ознаки, з якими буде працювати у подальшому наша мережа. Завдяки ознакам, які були вилучені під час згортки, мережа отримує певні деталі, наприклад, текстури, форми і тому інше. Були розглянуті також процес сегментації об'єктів за допомогою Mask R-CNN та її застосування для обробки зображень.

У третьому розділі були опрацьовані елементи оркестрації та CI/CD для нейронної мережі, що дозволить підтримувати та розвивати модель нейронної мережі у подальшому, щоб адаптувати її під нові умови та збільшувати її результативність з використанням нових підходів оптимізації.

У четвертому розділі були розглянуті та протестовані методи комбінування типів нейронної мережі, а саме:

- згорткові шари від CNN;
- механізм уваги від Transformer.

Були протестовані мережі VGGNet, ResNet, CoAtNet (у двох імплементаціях). Мережі були обрані як найпопулярніші та які отримують найбільші результати в обробці зображень.

Додаток який тестувався під час виконання атестаційної роботи був побудований на базі Tensorflow та Keras за допомогою Python. Для оркестрації використовувався фреймворк Airflow.

Серед основних проблем, що виникли під час імплементації прототипу – це відсутність доступу до потужних дискретних відеокарт, що дозволяють за допомогою платформи CUDA прискорити навчання моделі, а також низька пропускна швидкість каналу призвела до проблем з часом обробки вхідних зображень навчального набору.

Наукова новизна полягає у тому, щоб за допомогою прототипу

автоматизованої класифікаційної системи основі багаторівневого навчання для вирішення задачі машинного навчання, проаналізувати ефективність комбінованих систем на базі згорткових нейронних мереж та трансформерів. Практичне значення одержаних результатів показали, що результати можуть використовуватись у майбутніх дослідженнях за напрямком покращення прототипу класифікаційної моделі, а також про необхідність більш потужний ЕОМ або хмарних сервісів для навчання настільки потужних систем.

Рішення поставлених завдань та досягнуті результати: у даній роботі було представлено прототип автоматизованої системи для вирішення задачі класифікації на основі зображень за допомогою згорткової нейронної мережі з елементами комбінування з іншим методом ШНМ.

ПРОГРАМНИЙ ЗАСІБ, ІНТЕРНЕТ, ВЕБ, НЕЙРОННА МЕРЕЖА, ДЕТЕКЦІЯ, РОЗПІЗНАВАННЯ, МАШИННЕ НАВЧАННЯ, KERAS, TENSORFLOW

Публікації здобувача за темою роботи:

1. Шевченко О. Застосування хмарних сервісів для обробки наукових даних та масштабування обчислень в порівнянні з традиційним розгорткуванням. *Цифровізація науки та сучасні тренди її розвитку: II Міжнародна студентська наукова конференція* (м. Миргород листопад 2021 р.) – Миргород, 2021. С.88.

2. Shevchenko O. Advantages of using cloud computing to build scalable applications in solving data analysis problems. *Features of the development of modern science in the pandemic's era* : I International Scientific and Theoretical Conference (Berlin, Germany, December 3, 2021). Berlin, 2021. Vol. 2, pp. 85.

Перелік публікацій керівника та співробітників кафедри, використаних в

роботі:

1. Руденко О.Г., Бодянський Є.В. Штучні нейронні мережі: Навчальний посібник. – Київ : Компанія СМІТ, 2006. – 404 с.

2. Бодянский Е. В., Руденко О. Г. Искусственные нейронные сети: архитектуры, обучение, применения //Харьков: Телетех. – 2004. – Т. 369.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	11
ВСТУП.....	12
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	14
1.1 Актуальність обраної теми.....	14
1.2 Область застосування	15
1.3 Постановка задачі.....	18
2 МЕТОДИ ПОБУДОВИ ЗАСТОСУНКІВ З ВИКОРИСТАННЯМ ШНМ.....	19
2.1 Штучні нейронні мережі	19
2.2 Багаторівневі нейронні мережі	20
2.3 CNN.....	21
2.5 Функція активації	24
2.6 Пулінг	25
2.7 Попередня обробка та доповнення даних.....	26
2.8 Знаходження об'єкта.....	26
2.9 Сегментація зображень.....	27
3 ІНСТРУМЕНТИ ДЛЯ АВТОМАТИЗАЦІЯ ПРОЦЕСУ НАВЧАННЯ МОДЕЛІ	29
3.1 Автоматизація процесу налаштування нейронної мережі.....	29
3.2 Порівняння on-premise та хмарних підходів	31
3.3 Переваги використання згорткових нейронних мереж.....	34
3.4 Препроцесінг зображень	35
3.5 Архітектура додатку для обробки зображень	37
3.6 Програмні засоби для реалізації аналізу над згортковими нейронними мережами	39
4 КОМБІНУВАННЯ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ З ІНШИМИ ШНМ.....	42
4.1 Трансформери як елемент комбінування	42

4.2 CoAtNet.....	45
ВИСНОВКИ.....	50
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	51
ДОДАТОК А.....	ОШИБКА! ЗАКЛАДКА НЕ ОПРЕДЕЛЕНА.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

CNN – convolutional neural network – згорткова нейронна мережа;
CPU – Central Processing Unit – центральний процесор;
GPU – Graphics Processing Unit – графічний процесор;
ImageNet – база анотованих зображень, призначених для обробки
методами розпізнавання зображень та машинного зору;
ReLU – Rectified Linear Unit – випрямлена лінійна одиниця;
TPU – Tensor Processing Unit – тензорний процесор;
ML – Machine learning – машинне навчання;
FFN – Feed-forward Network – мережа прямого розповсюдження;
SOTA – State of the art – передові рішення ;
IoU – Intersection over Union – перетин над об'єднаннями

ВСТУП

В 21 сторіччі є багато напрямків науки та промисловості, що займаються впровадженням цифрових рішень до автоматизації людської участі у повсякденному житті. Одним з інструментів вирішення автоматизації людської праці є комп'ютерний зір, який може виконувати різноманітні однотипні дії за рахунок алгоритмів, що досліджуються спеціалістами з математичного моделювання, комп'ютерних наук та більш прикладних до кожної галуззі. Під час роботи з комп'ютерним зором ми можемо використовувати велику кількість методів обробки зображень за допомогою згорткових нейронних мереж [1]. За допомогою цих мереж вирішуються задачі пов'язані з зображеннями, які можуть містити певну інформацію, що може бути вилучена детекцією, розпізнаванням, класифікацією або іншими методами. Наразі існує великий спектр CNN мереж, що розвиваються з кожним роком та надають нових можливостей у сфері машинного навчання. Нажаль не завжди збільшення набору параметрів дає бажаний результат, щоб задовільнити потреби бізнесу чи науковців і потрібно знаходити нові підходи до вирішення цих проблем. В даній роботі буде розглянуті методи згорткових мереж для обробки зображень та також, які сучасні підходи використовуються у SOTA рішеннях для збільшення точності в отриманих результатах при однакових вхідних даних.

Вирішенням проблем детекції розпізнавання образів є комп'ютерний зір, який належить до теорії та технології створення штучних систем, які отримують інформацію у вигляді зображень. Розпізнавання зображень є у багатьох технологічних додатках сучасності таких як засоби ідентифікації, контроль якості виробництва, виявлення аномалій на знімках та інше.

Наразі ведеться стрімкий розвиток у детекції та розпізнаванні облич у науковій та промисловій сфері. Наприклад держава потребує точних та швидких методів та засобів розпізнавання обличчя у правоохоронних органах,

банки потребують аутентифікації особи за обличчям, роботи-допомічники розпізнають емоції та багато інших застосунків.

Світові тенденції говорять на про те, що зараз відійшли від класичних методів машинного навчання та використовують CNN. Саме з цими моделями застосунки мають кращі результати, на базі CNN і буде проводитись аналіз результатів обробки зображень у цій роботі.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Актуальність обраної теми

Наразі велика кількість інформацію збирається саме з цифрових зображень, що ми отримуємо з великої кількості джерел, які передають, збирають або обробляють дані. Така можливість з'явилась завдяки збільшенню обчислювальної здатності ЕОМ, а саме дешеві та швидкі сховища даних, здешевшення оперативної пам'яті та збільшення її пропускної здатності, розробка нових алгоритмів для паралельної обробки даних та багато інших факторів.

Так зі збільшення даних (зображень) та здатності обробляти велику кількість інформації за малий проміжок часу призвело до сприятливих умов для роботи з комп'ютерним зором, а в нашого випадку – згортковими нейронними мережами. Згорткові нейронні мережі (CNN) зараз є стандартом для обробки зображень та поділяються на багато типів і можуть виконувати різноманітні задачі.

Якщо розглядати методи за допомогою яких CNN обробляють зображення, то ми повинні звернути увагу на:

- сегментизація;
- детекція;
- розпізнавання;
- позбавлення шуму;
- перенесення стилю;
- покращення деталізації.

Результати моделей CNN хоч ще не є ідеальними, але вони виграють у швидкості обробки великої кількості зображень, а в деяких сферах і в точності, саме тому в деяких сферах застосування машинного навчання вже є стандартом, та не є чимось новим.

1.2 Область застосування

В наш час згорткові нейронні мережі застосовується у багатьох сферах життєдіяльності людини при тому, що це відбулося за доволі короткий час. Машинне навчання, а саме використання комп'ютерного зору зробило великі кроки у своєму розвитку. Комп'ютерний зір має багато успішних сфер застосування, де за допомоги нейронних мереж були автоматизовані складні та трудомісткі процеси. Завдяки чому тепер у багатьох однотипних задачах передають перевагу автоматизованим процесам, бо людський фактор, а саме втома та необачність, може впливати на результат.

Сфери застосування нейронних мереж наразі доволі різні, але зазвичай це використовуються у багатьох автоматизованих промислових рішеннях та у багатьох мобільних додатках [2] .

Прикладом використання комп'ютерного зору на базі CNN може бути відокремлення елементів з камер спостереження. Одним з таких варіантів є детекція підозрілих предметів у громадських місцях або у скупченнях людей. Також у великих містах приділяють увагу на розпізнавання облич серед натовпу та ідентифікацію осіб. Камери відеоспостереження з комп'ютерним зором записують кожну деталь у громадських зонах у режимі 24 на 7. Тож перегляд відеозаписів, знятих ними протягом доби, дає змогу поліцейським відстежити особу чи групу, яка встановила предмет на місці.

Крім того, системи відеоспостереження з комп'ютерним баченням також можуть бути налаштовані на автономне виявлення підозрілої діяльності в зонах великого скупчення людей. Зйомка відео в режимі реального часу, штучний інтелект і прогнозна аналітика відіграють свою роль у виявленні та попередженні таких дій на основі таких факторів, як мова тіла суб'єкта, наявність будь-якої зброї чи інших небезпечних предметів біля нього та інших факторів.

Як зазначалося вище, громадське спостереження є нескінченною частиною розумної поліції міста. Уважно дивитися нескінченні відеозйомки

людей, які пересуваються в людних місцях, людям неможливо. Таким чином, замість того, щоб витратити багато часу та грошей на наймання більшої кількості офіцерів для виконання цього завдання, використання відеокамер відеоспостереження має сенс для відділів поліції у містах з великою цифровізацією та розвиненою інфраструктурою. Банки почали використовувати біометричні дані та застосовувати комп'ютерний зір для ідентифікації своїх користувачів. Це підвищило безпеку особистих рахунків клієнтів та підвищило надійність при видачі кредитів, а також у всіх інших операцій пов'язаних з коштами користувачів. Схоже на те, що сканування обличчя з часом повністю замінить картки банкомату. Але тим часом за допомогою розпізнавання обличчя можна переконатись, що особи, які користуються картками банкоматів, є тими, за кого кажуть. Наразі розпізнавання обличчя використовується в банкоматах Макао для ідентифікації особи

Додатки аналізують базу даних людей які мають генетичні хвороби чи порушення після цього порівнюють з обличчям користувача. Наразі немає достатніх даних для датасетів, щоб давати результат для дорослих людей, тому поки що використовують тільки з дітьми.

У медичній сфері теж почали використовувати спеціальні медичні додатки задля пошуку генетичних порушень.

Остеоартрит є одним із тих захворювань, які неможливо виявити, доки не буде завдано шкоди, і воно не розвинеться. До цього часу лікарі не могли виявити це, поки не почалося пошкодження кісток.

За допомогою моделей згорткових нейронних мереж ми тепер можемо вводити навчальний набір зображень минулих пацієнтів, у яких діагностовано захворювання, і здорових, алгоритм виявляє ці закономірності та може передбачити, хто буде сприйнятливий до захворювання через 3 роки, за допомогою згортки нейронна мережа сама може виділяти незначні відмінності, які так важливі для медичного персоналу. Ця інформація містилася в мільйонах сканованих зображень, зроблених до теперішнього часу, але людське око не

було навчене її виявляти. На щастя, прогрес у глибокому навчанні та дослідженнях CNN дозволили виявити ці тонкощі в зображеннях.

CNN також домінують у сфері виявлення COVID-19 за допомогою рентгенів грудної клітини/КТ [3]. Дослідження за участю CNN зараз є домінуючою темою на великих конференціях. Крім того, існують спеціальні випуски, зарезервовані в відомих журналах для вирішення проблем за допомогою моделей глибокого навчання. Величезна кількість літератури, доступної на CNN, є свідченням їх ефективності та широкого використання. Однак різні дослідницькі спільноти розробляють ці програми одночасно, щоб зрозуміти наслідки COVID-19 та перетік хвороби [4], а результати розповсюдження розкидані в широкому та різноманітному діапазоні матеріалів конференцій і журналів. Але досі ще мало рішень для виявлення COVID-19 на ранніх етапах.

Останнім часом було опубліковано велику кількість опитувань щодо глибокого навчання. Також огляд методів для роботи зі КТ знімками приводиться у роботах SOTA рішень пов'язаних з класифікацією мозку [5].

Застосування CNN моделей наразі можна зустріти у багатьох медичних проектах, починаючи від використання Convnet для класифікації клітин HEp-2 [6] до класифікації ехокардіографії [7].

Розпізнавання обличчя можна використовувати для пошуку зниклих дітей та жертв торгівлі людьми. Поки зниклі особи додаються до бази даних, правоохоронні органи можуть отримувати попередження, як тільки їх розпізнають за допомогою розпізнавання осіб - будь то аеропорт, роздрібний магазин чи інший громадський простір. Застосування у таких технологій надає надію, що людина, яка загубилась чи була викрадена буде знайдена.

Також для кожного з нас стали або стають в нагоді методи розпізнавання обличчя у мобільних додатках. Ця функція вже є у багатьох смартфонах та має багато імплементацій, від самих примітивних за фото до підкріплення результату з розпізнаванням райдужки ока.

1.3 Постановка задачі

Розробляючи програмні засоби для обробки зображень за допомогою згорткових нейронних мереж потрібно підійти з певною відповідальністю до вибору архітектури додатку. Одним із вирішальних факторів для отримання бажаного результату за допомогою комп'ютерного зору є правильно обрана модель та архітектура. Для вирішення задач комп'ютерного зору з використанням CNN потрібно визначитись з формою відображення результату та методом вирішення задачі. Через те що CNN моделі мають багато методів необхідно вирішити доречність використання тієї чи іншої моделі, проаналізувати їх результати та розглянути методи покращення отриманих результатів. Спираючись на тему атестаційної роботи у ході розгляду методів обробки зображень на базі згорткових нейронних мереж, будемо використовувати моделі цієї архітектури, розглянемо методи застосування та доопрацюємо методи за допомогою комбінування різних типів для отримання кращих результатів.

2 МЕТОДИ ПОБУДОВИ ЗАСТОСУНКІВ З ВИКОРИСТАННЯМ ШНМ

2.1 Штучні нейронні мережі

Термін «нейронна мережа» походить від роботи Уоррена С. Маккалоха та Уолтера Піттса, які розробили першу концептуальну модель штучної нейронної мережі. У своїй роботі вони описують концепцію нейрона, окремої клітини, що живе в мережі клітин, яка отримує вхідні дані, обробляє ці вхідні дані та генерує вихідні дані. Через характер нервової діяльності «все або нічого», нейронні події та відносини між ними можна розглядати за допомогою пропозиційної логіки [8]. Виявлено, що поведінка кожної мережі може бути описана цими термінами з додаванням більш складних логічних засобів для мереж, що містять кола; і що для будь-якого логічного виразу, який задовольняє певні умови, можна знайти мережу, яка поводить себе так, як вона описує. Показано, що багато конкретних варіантів серед можливих нейрофізіологічних припущень є еквівалентними в тому сенсі, що для кожної мережі, яка веде себе за одним припущенням, існує інша мережа, яка поводить себе за іншого і дає ті самі результати, хоча, можливо, не в той самий час.

У світі комп'ютерів нейронні мережі організовані на рівнях, що складаються з взаємопов'язаних вузлів, які містять функцію активації. Ці шаблони представлені в мережі через вхідний рівень, який далі передає їх на один або більше прихованих рівнів. Приховані шари виконують всю обробку та передають результат на вихідний рівень [9].

Наприклад, нейронна мережа намагається визначити, чи є на зображенні білка. Він уже обробив сотні тисяч фотографій білок і знає, що колір їхньої шерсті є комбінацією певних відтінків, таких як рудий, сріблястий і жовтий. На нейрон надходять сигнали про те, що на зображенні переважають ці три кольори, а це означає, що, швидше за все, на ньому є білка. Далі нейромережа перевіряє, чи є на малюнку очі, вуха та хвіст. Якщо всі чотири фактори

збігаються, можна з упевненістю стверджувати, що це зображення білки.

2.2 Багаторівневі нейронні мережі

Багатошарові нейронні мережі можна налаштувати різними способами. Як правило, вони мають принаймні один вхідний рівень, який надсилає зважені вхідні дані до серії прихованих шарів, і вихідний рівень у кінці. Ці більш складні налаштування також пов'язані з нелінійними конструкціями, що використовують сигмоподібні та інші функції для керування запуском або активацією штучних нейронів. Хоча деякі з цих систем можуть бути побудовані фізично з фізичними матеріалами, більшість із них створено за допомогою програмних функцій, які моделюють нейронну активність [10]. На рисунку 2.1 зображена схема побудови ШНМ:

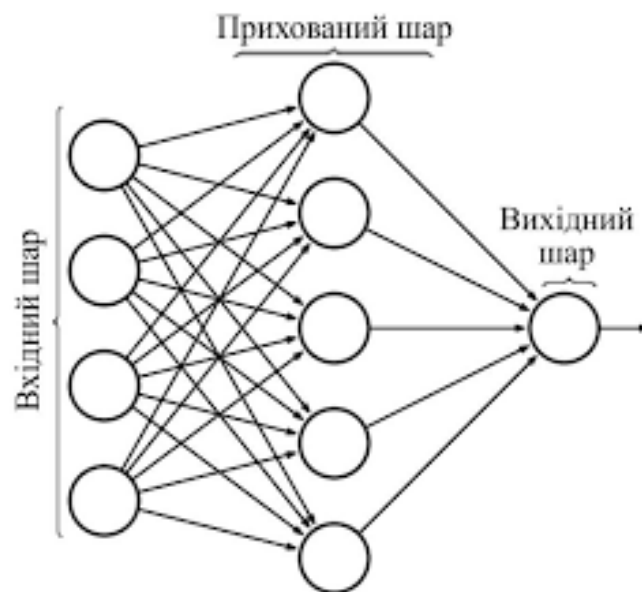


Рисунок 2.1 – Ілюстрація штучної нейронної мережі

Згорткові нейронні мережі (CNN) корисні для обробки зображень і комп'ютерного зору, а також рекурентні нейронні мережі, глибокі мережі та системи глибоких переконань – усі це приклади багаторівневих нейронних мереж. CNN, наприклад, можуть мати десятки шарів, які послідовно працюють над зображенням. Все це є ключовим для розуміння того, як функціонують

сучасні нейронні мережі.

2.3 CNN

Згорткові нейронні мережі — це доволі потужне та гнучке сімейство мереж глибокого навчання, які призначенні для обробки зображень. CNN має принципово новий тип шарів, які до цього не використовувались у процесі навчання нейронної мережі та складається з трьох типів шарів: згортки, об'єднання та повнозв'язані шари. Перші два, шари згортки та об'єднання, виконують функцію виділення карти ознак, а повнозв'язний шар виконує роль відокремлення певного результату [11]. Рівень згортки відіграє ключову роль у CNN, який складається зі стеку математичних операцій, таких як згортка, спеціалізований тип лінійної операції. У цифрових зображеннях значення пікселів зберігаються у двовимірній (2D) сітці, тобто масиві чисел, і невелика сітка параметрів, яка називається ядром, оптимізованим екстрактором ознак, застосовується до кожної позиції зображення, що робить CNN дуже ефективними для обробки зображень, оскільки особливість може з'являтися в будь-якому місці зображення. Оскільки один шар передає свій вихід на наступний рівень, витягнуті функції можуть ієрархічно та поступово ставати складнішими. Процес оптимізації таких параметрів, як ядра, називається навчанням, яке виконується таким чином, щоб мінімізувати різницю між виходами та базовими мітками істинності за допомогою алгоритму оптимізації, що називається зворотним поширенням і градієнтним спуском, серед інших [12].

Архітектура CNN включає кілька будівельних блоків, таких як шари згортки, шари об'єднання та повністю зв'язані шари. Архітектура згорткової нейронної мережі має декілька складових і будується за допомогою повторень стека з декількох згорткових шарів та шару об'єднання. Після цього додається один або більше повністю зв'язаних шарів. Етап, на якому вхідні дані перетворюються на вихід через ці рівні, називається прямим поширенням. Хоча операції згортання та об'єднання, описані в цьому розділі, застосовуються до

2D-CNN, подібні операції також можна виконувати з тривимірними 3D-CNN. Рівень згортки є фундаментальним компонентом архітектури CNN, який виконує виділення ознак, що зазвичай складається з комбінації лінійних і нелінійних операцій, тобто операції згортки та функції активації.

Сучасні архітектури CNN зазвичай використовують нульове доповнення, щоб зберегти розміри в площині, щоб застосувати більше шарів. Без нульового доповнення кожна послідовна карта функцій ставала б меншою після операції згортки. Відстань між двома послідовними позиціями ядра називається кроком, який також визначає операцію згортки. Загальний вибір кроків 1; однак крок, більший за 1, іноді використовується для зменшення дискретизації карт функцій. Альтернативною технікою зменшення дискретизації є операція об'єднання (пулінгу). Головною функцією операції згортки є етап розподіл вагових коефіцієнтів: ядра розподіляються між усіма позиціями зображення. Так, після цього ми втрачаємо інформацію з оригінального зображення, яке приходить на перші етапи нашої нейронної мережі, але це є побічним ефектом, коли ми намагаємось виділити деякі ознаки, яких може бути велика кількість, але зберігати усі дані буде недоцільно, бо саме процес виявлення ознак для нас є пріоритетним.

2.4 Згортка

Згортка — це спеціалізований тип лінійної операції, який використовується для виділення ознак, де невеликий масив чисел, який називається ядром, застосовується до вхідних даних, які є масивом чисел, який називається тензором. Поелементний добуток між кожним елементом ядра та вхідним тензором обчислюється в кожній точці тензора та додається для отримання вихідного значення у відповідній позиції вихідного тензора, яка називається картою [13]. Ця процедура повторюється з використанням кількох ядер для формування довільної кількості карт ознак, що представляють різні характеристики вхідних тензорів

Згортка витягує фрагменти вхідної картки об'єктів і застосовує до них фільтри для обчислення нових об'єктів, створення вихідної картки об'єктів або згорнутих об'єктів (які можуть мати інший розмір та глибину, ніж вхідна карта об'єктів). Згортки визначаються двома параметрами:

- Розмір плиток, що виймаються (зазвичай 3x3 або 5x5 пікселів);
- Глибина вихідної картки об'єктів, яка відповідає кількості застосованих фільтрів.

Під час згортки фільтри (матриці того ж розміру, що і розмір плитки) ефективно ковзають по сітці вхідної карти об'єктів по горизонталі та вертикалі, по одному пікселю за раз, витягуючи кожен відповідну плитку. На рисунку 2.2 продемонстровано процес згортки:

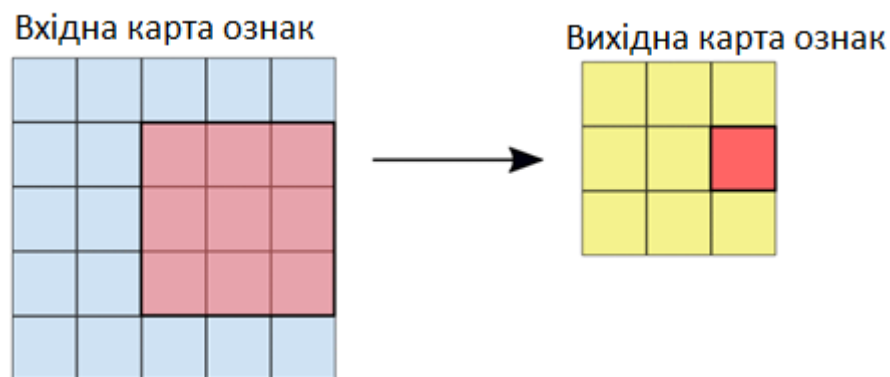


Рисунок 2.2 – Процес згортки 3 на 3 з глибиною 1 на вхідній карті 5 на 5

Для кожної пари фільтр-плитка CNN виконує поелементне множення матриці фільтру та матриці плитки, а потім підсумовує всі елементи результуючої матриці, щоб отримати єдине значення. Кожне з цих результуючих значень для кожної пари фільтр-плитка потім виводиться у згорнутій матриці ознак.

Під час навчання CNN дізнається оптимальні значення для матриць фільтрів, які дозволяють їй отримувати значущі функції (текстури, краї, форми) із вхідної картки функцій. У міру збільшення кількості фільтрів (глибина

вихідної карти об'єктів), що застосовуються до вхідних даних, збільшується кількість функцій, які CNN може отримати. Однак компроміс полягає в тому, що фільтри становлять більшу частину ресурсів, що витрачаються CNN, тому час навчання також збільшується при додаванні додаткових фільтрів. Крім того, кожен фільтр, доданий у мережу, забезпечує меншу додаткову цінність, ніж попередній, тому інженери прагнуть створювати мережі, що використовують мінімальну кількість фільтрів, необхідних для отримання ознак, необхідних для точної класифікації зображень.

2.5 Функція активації

У неглибоких ШНМ використовуються нелінійні функції активації. Часто зустрічаються різновиди сигмоїдальних і тангенціальних передавальних функцій, які є нелінійними, але на практиці навчання глибоких ШНМ такі функції можуть привести до проблем з загасанням або збільшенням градієнтів. На рисунку 2.3 вказана функція ReLU:

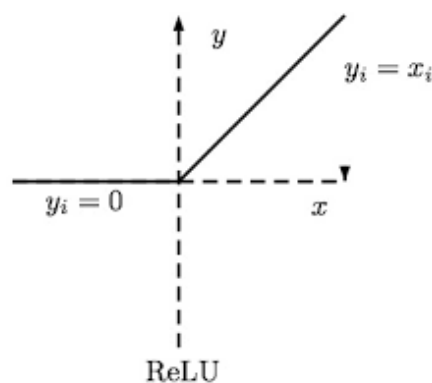


Рисунок 2.3 – Графік ReLU

За допомогою ReLU ми можемо апроксимувати будь-яку функцію. Це означає, що ми можемо стекати (накладати) шари [14]. Область допустимих значень ReLU - $[0, \infty)$.

Якщо використовувати велику нейронну мережу з безліччю нейронів, то використання сигмоїди або гіперболічного тангенсу спричинятиме активацію всіх нейронів та майже всі активації мають бути оброблені для опису виходу мережі.

ReLU дозволяє зробити мережу більш розріджену та мати деякі значення нейронів 0, це прискорить процес навчання. Візьмемо мережу, у якій при старті буде випадково ініціалізовані ваги та в якій приблизно 50% активацій дорівнюють 0 через повернення результату від ReLU для від'ємних значень. У такій мережі включається менша кількість нейронів (розріджена активація), а мережа стає легше.

Але при всій ефективності та швидкості ReLU, у цієї функції є велика проблема під назвою Dying ReLU. Це процес коли у багатьох нейронів вихідні значення тільки дорівнюють 0 (зазвичай при від'ємних значеннях). Для вирішення цієї проблеми використовують зменшення learning rate або шагу навчання, зменшення bias (зміщення) та використання інших різновидів ReLU - Leaky ReLU є ефективним методом вирішення проблеми ReLU, що вмирає, і робить це шляхом додавання невеликого нахилу в від'ємному діапазоні. Це змінює функцію для генерування невеликих негативних результатів, коли вхід менше 0.

Зазвичай використовують після повністю зв'язаних шарів (FC), що схожі на штучну нейронну мережу, де кожен вузол має вхідні з'єднання з усіх входів, і всі з'єднання мають пов'язані з ними ваги. Вихід – це сума всіх вхідних даних, помножена на відповідні ваги. За шаром FC слідує функція активації ReLU, яка виконує завдання класифікатора.

2.6 Пулінг

Після ReLU настає етап об'єднання (пулінгу), на якому CNN зменшує кількість згорнутих функцій, щоб заощадити час обробки та зменшуючи кількість вимірів карти ознак, зберігаючи при цьому найважливішу інформацію

про функції [15]. Загальний алгоритм, який використовується для цього процесу, називається *maxpooling* (максимальним об'єднанням).

Максимальне об'єднання працює подібно до згортки. Ми ковзаємо по карті функцій і витягуємо плитки заданого розміру. Для кожної плитки максимальне значення виводиться на нову карту ознак, а всі інші значення відкидаються. Максимальні операції об'єднання приймають два параметри – розмір та шаг (*stride*). Шаг – це відстань у пікселях, що розділяє кожну витягнуту плитку. На відміну від згортки, коли фільтри ковзають по карті об'єктів пікселя за пікселем, при максимальному об'єднанні крок визначає місця, де витягується кожна плитка. Для фільтра 2x2 крок, рівний 2, вказує, що операція максимального об'єднання витягуватиме з карти об'єктів всі плитки 2x2, що не перекриваються.

2.7 Попередня обробка та доповнення даних

Необроблені зображення, отримані за допомогою модальностей зображень, потрібно попередньо обробити та доповнити перед надсиланням на CNN. Необроблені дані зображення можуть бути перекошеними, зміненими спотворенням зсуву, можуть мати неоднорідність інтенсивності під час створення зображення, і, отже, потребуватимуть попередньої обробки. Існує кілька методів попередньої обробки даних, і переважними методами є віднімання середнього значення та нормалізація. CNN потрібно навчати на великому наборі даних, щоб досягти найкращої продуктивності. Розширення даних збільшує наявний набір зображень за рахунок горизонтального та вертикального повороту, трансформації, масштабування, випадкового кадрування та варіацій інтенсивності. Попередньо оброблені доповнені зображення потім надходять у CNN.

2.8 Знаходження об'єкта

Для детекції об'єкта використовуються метод bounding box . Bounding Box — це позначення нам потрібно, щоб описати просторове розташування предмета. Обмежувальна рамка є прямокутною, яка визначається координатами і верхнього лівого кута прямокутника та такими ж координатами нижнього правого кута. Іншим часто використовуваним представленням обмежувальної рамки є осьові координати центру обмежувальної рамки, а також ширина та висота рамки. Коли ми намагаємося розпізнати об'єкти, ми використовуємо обмежувальні рамки, в яких об'єкт, можливо, буде виявлений.

Прикладом для виявлення об'єктів є Faster-RCNN [16]. У цього типу моделей притамане використання окремої нейронної мережі для вивчення обмежень регіону інтереса.

Таким чином, модель спочатку використовує 1 CNN для отримання обмеження щодо регіону, потім дотримується точно такої ж логіки з Fast RCNN для виявлення об'єктів, тому вона використовує ці пропозиції для виділення функцій за допомогою CNN, потім класифікує за допомогою повністю зв'язаних шарів і покращує ROI за допомогою обмежувальної рамки регресора.

2.9 Сегментація зображень

Для розгляду сегментації зображень будемо використовувати архітектуру Mask R-CNN. Mask R-CNN використовує опорні рамки для виявлення кількох об'єктів, об'єктів різного масштабу та об'єктів, що накладаються на зображення. Це підвищує швидкість і ефективність виявлення об'єктів.

Опорні рамки – це набір попередньо визначених обмежувальних рамок певної висоти та ширини. Ці поля призначені для визначення масштабу та співвідношення сторін конкретних класів об'єктів, які ми хочемо виявити.

Щоб передбачити кілька об'єктів або кілька екземплярів об'єктів на зображенні, Mask R-CNN робить тисячі прогнозів. Остаточне виявлення об'єктів виконується шляхом видалення блоків опорної рамки, які належать до фонового класу, а ті, що залишилися, фільтруються за їх показником

достовірності. Ми знаходимо опорні блоки з IoU більше 0.5. Блоки прив'язки з найбільшою оцінкою достовірності вибираються за допомогою супресії немаксимального значення.

IoU обчислює перетин по об'єднанню двох обмежувальних рамок, обмежувальної рамки для основної істини та обмежувальної рамки для прогнозованої рамки за допомогою алгоритму.

Коли IoU дорівнює 1, це означатиме, що передбачено, і обмежувальні прямокутники ідеально перекриваються. На рисунку 2.4 продемонстрован алгоритм NMS:

```

procedure NMS( $B, c$ )
   $B_{nms} \leftarrow \emptyset$  Initialize empty set
  for  $b_i \in B$  do => Iterate over all the boxes
     $discard \leftarrow \text{False}$  Take boolean variable and set it as false. This variable indicates whether b(i) should be kept or discarded
    for  $b_j \in B$  do Start another loop to compare with b(i)
      if  $\text{same}(b_i, b_j) > \lambda_{nms}$  then If both boxes having same IOU
        if  $\text{score}(c, b_j) > \text{score}(c, b_i)$  then
           $discard \leftarrow \text{True}$  Compare the scores. If score of b(i) is less than that of b(j), b(i) should be discarded, so set the flag to True.
        if not  $discard$  then Once b(i) is compared with all other boxes and still the discarded flag is False, then b(i) should be considered. So add it to the final list.
           $B_{nms} \leftarrow B_{nms} \cup b_i$ 
  return  $B_{nms}$  Do the same procedure for remaining boxes and return the final list

```

Рисунок 2.4 – Алгоритм NMS

Щоб виявити об'єкт один раз на зображенні, немаксимальна супресія розглядає всі обмежувальні рамки з $\text{IoU} > 0.5$, якщо це значення менше, то Non-Max Suppression алгоритм видаляє всі рамки [18], де IoU менше або дорівнює 0.5.

3 ІНСТРУМЕНТИ ДЛЯ АВТОМАТИЗАЦІЯ ПРОЦЕСУ НАВЧАННЯ МОДЕЛІ

3.1 Автоматизація процесу налаштування нейронної мережі

Як правило, ми хочемо запускати наш пайплайн за деяким розкладом, тому що наша модель ML не навчається та не оновлюється часто, вона може застаріти та дедалі поганіше справлятиметься з проблемою, яку покликана вирішити. Ми навіть можемо захотіти, щоб різні частини нашого пайплайну працювали за графіками, що не відповідають встановленим нормам. Чудова структура оркестровки дозволяє нам планувати різні частини конвеєра окремо, водночас цілісно визначаючи залежності. Синхронізація з установленими залежностями вгору/вниз, нам потрібно переконатися, що компоненти пайплайну належним чином взаємодіють один з одним. Визначення схеми потоку даних між компонентами, тобто вказати завданням, що передають, що надсилати, і завданням, що очікувати, — це лише половина. Також важливо вказати, коли і як компонент має отримувати дані. Наприклад, чи потрібно компоненту чекати, поки він отримає деякі дані, як у датчику? Транзакція запланована чи заснована на тригері?

Хоча інтеграція, тестування та розгортання конвеєрів даних і ML виглядають інакше, ніж CI/CD і тестування в розробці програмного забезпечення, вони, тим не менш, є важливою частиною успішного робочого процесу ML. Часті перевірки якості даних роблять модель надійною. Належна обробка винятків і помилок забезпечує високу доступність розгорнутого конвеєра та відповідної служби.

Найкращим рішенням для зменшення перенавченості моделі є отримання більшої кількості навчальних даних. Модель, навчена на більшому наборі даних, як правило, краще узагальнює, хоча це не завжди досяжно. Інші рішення включають регуляризацію з dropout або спадом ваги та збільшення даних, а також зменшення складності архітектури. Dropout — це нещодавно введена

техніка регуляризації, де випадково вибрані активації встановлюються на 0 під час навчання, так що модель стає менш чутливою до певних ваг у мережі. Затухання ваги, яке також називають регуляризацією L2, зменшує переобладнання, штрафуючи вагові коефіцієнти моделі, щоб ваги приймали лише малі значення. Таким чином, звіт про продуктивність кінцевої моделі на окремому (невидимому) тестовому наборі, а в ідеалі, на зовнішніх наборах даних перевірки, якщо це можливо, має вирішальне значення для перевірки узагальненості моделі. За рахунок оркестрації ці всі параметри можна регулювати дуже швидко.

Крім того, з наукою про виробничі дані та робочими процесами машинного навчання очікується, що бізнес буде процвітати — не всупереч постійним експериментам і оновленням. Наша модель у виробництві не повинна застоюватися; натомість він завжди має бути в найкращій можливій ітерації. Підтримка експериментів у виробничих процесах дозволяє нам, як дослідникам даних, продовжувати рухати бізнес вперед, перевіряючи нові гіпотези. Коли ми експериментуємо з даними та моделлю ML або просто навчаємо модель на нових даних, ми повинні постійно тестувати конвеєр, щоб переконатися, що він продовжує добре функціонувати як сам по собі, так і з усією системою чи платформою. Разом з конвеєром, тести також повинні бути організовані та автоматизовані як частина робочого процесу. На рисунку 3.1 зображені компоненти автоматизації процесу розробки:

Звичайно, ми хочемо стежити за тим, наскільки добре наша модель ML виконує свою основну функцію — моделює тенденції в даних, навчаючись на них. Індивідуальні показники продуктивності конвеєрів машинного навчання ґрунтуватимуться на їхніх конкретних вимогах і можуть варіюватися від точності й середньої точності до функцій чистого доходу, які безпосередньо пов'язують прогнози моделі з їхнім впливом на дохід бізнесу. Модель ML із чудовою продуктивністю на етапі перевірки концепції не гарантує, що вона продовжуватиме забезпечувати таку продуктивність назавжди.

MLOPS = ML + Dev + Ops

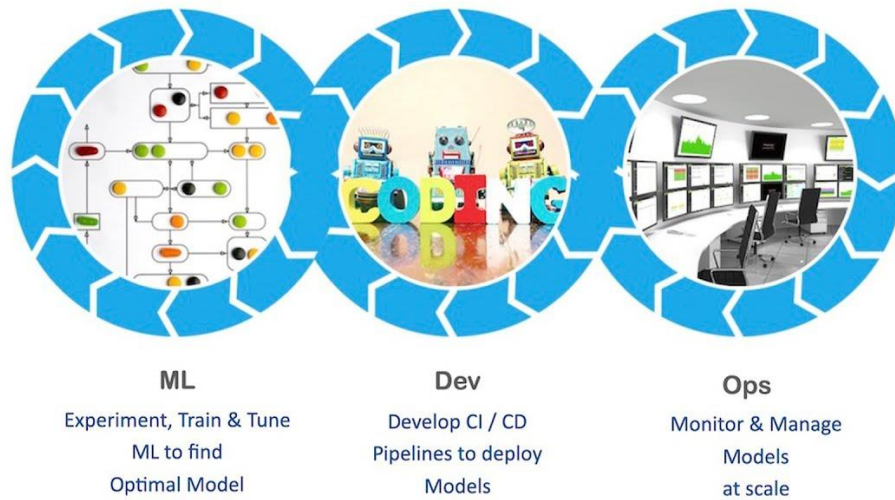


Рисунок 3.1 – Візуалізація CI/CD для моделей машинного навчання

Нам потрібно звернути пильну увагу на падіння продуктивності моделі та бути готовими реагувати на нього. Через зародження операцій машинного навчання (MLOps) як дисципліни, напрочуд небагато виробничих конвеєрів мають правильний апарат моніторингу моделі. Однак, як у галузі починають усвідомлювати, відносно невеликий підйом побудови якісної інфраструктури моніторингу в кінцевому підсумку забезпечує величезну цінність. Завдяки потужній платформі оркестровки, як-от Airflow, інфраструктура практично вже присутня — залишається лише організувати моніторинг як частину робочого процесу.

3.2 Порівняння on-premise та хмарних підходів

Цифровізація змінює всі сфери науки, від загального підходу до експериментів, обміну знаннями та залучення людей до дослідницьких проєктів [19]. Пандемія стала головним каталізатором більшої цифровізації бізнес-процесів і дослідницьких процесів. Але для цього потрібне середовище, де можна продовжувати розробляти програми. Значна частина обчислювальних

процесів у бізнесі та науці була зосереджена локально, і існували різноманітні мережеві доступи до машин. Коли всі перейшли на віддалену роботу, це питання постало дуже гостро, адже не всі дбали про доступ до локальних машин. Для вирішення цієї проблеми можна використовувати хмарні сервіси. Нинішня пандемічна ситуація змусила людей працювати з дому та перевести багато бізнес-процесів у віддалений режим. Співробітники, науковці мають спілкуватися та співпрацювати онлайн. Обробка великих даних вимагає великої обчислювальної потужності, яка не завжди є в персональних комп'ютерах або ноутбуках. У цьому випадку в цих процесах допоможуть хмарні сервіси. З їх допомогою ви можете безперешкодно продовжувати свій розвиток і надавати свої рішення. Хмарні провайдери часто стикаються з проблемами у своїх стратегіях безперервності бізнесу. Вони мають постійно підтримувати робочі процеси, щоб усі служби оновлювалися та могли без проблем масштабуватися, щоб впоратися зі збільшеним попитом. Хмарні постачальники надають середовище розробки, яке має бути завжди доступним для користувачів і відповідати вимогам, якщо допоміжна інфраструктура достатньо надійна, щоб забезпечити постійний доступ до загальнодоступних хмарних служб, а мережева інфраструктура може обробляти збільшений трафік. На базі будь-якого провайдера хмарних технологій існує безліч сервісів для обробки даних, зберігання даних і підтримки високої надійності в роботі з даними. Ви також можете зосередитися на своїй роботі та довірити все інше експертам, зокрема партнерам по платформі, сервісу та підтримці. Хмарні технології корисні для машинного навчання, оскільки їх можна швидко масштабувати та не вимагають жодного втручання користувача хмарного сервісу. При використанні хмарних сервісів стягується плата за використання. У цій моделі користувач платить лише за ту кількість ресурсів, яку він використав протягом певного часу. Ця модель дозволяє швидко змінювати кількість ресурсів, які вимагають малих або великих потужностей, що не дозволяє традиційний спосіб розгортання локальних машин. Також великим плюсом є можливість ділитися результатами користувачів і створювати групи доступу в межах однієї організації, наприклад,

університету, що прискорює робочий процес і прискорює обмін даними. У наш час дані є новим паливом і майже нескінченними, оскільки процес обробки даних генерує ще більше даних, які корисні в будь-якому бізнес-процесі. Існує багато сфер розваг, орієнтованих на соціально віддалену епоху в результаті зростання попиту на електронне навчання, телемедицину, робототехніку, штучний інтелект, сценарії доповненої та віртуальної реальності, розумні чат-боти, цифрові платежі, віртуальну роздрібну торгівлю тощо. Такі операції вимагають постійної підтримки додатків і швидкої зміни ресурсів лише для масштабування, оскільки ці сфери активно розвиваються з точки зору технологій. Індустрія електронної комерції переживає експоненціальну ескалацію, оскільки все більше людей переходять на онлайн-магазини, а також на послуги потокового відео, все більше і більше людей залишаються вдома та позбавлені інших можливостей для розваг. Тому, якщо люди мають менше соціальних контактів, їм потрібно підтримувати хоча б постійний доступ до послуг. Великою перевагою хмарних служб є те, що вони мають дуже великий запас ресурсів, за допомогою яких можна підтримувати будь-який аспект бізнесу чи науки. Хоча хмара була створена для забезпечення певних ключових функцій, таких як висока доступність, масштабованість, безперервність бізнесу, відмовостійкість, аварійне відновлення, автоматичне оновлення програмного забезпечення, гнучкість тощо, з часом з'явилося багато нових функцій і послуг із зростаючим інтересом до впровадження. Хмара різними підприємствами. Клієнти не можуть контролювати розташування інфраструктури в загальнодоступній хмарі, а вартість є спільною та є або безкоштовною, або у формі ліцензійної політики, як-от плата за використання. Ви також можете звернути увагу на процес масштабування в хмарних середовищах. Хмарна інфраструктура є чудовим рішенням для компаній, оскільки вона дозволяє їм ефективно та швидко масштабувати свої ІТ-ресурси відповідно до потреб бізнесу. Ідеально підходить для компаній зі змінними вимогами. У міру зростання попиту хмара забезпечить необхідну інфраструктуру та послуги, не турбуючись про фізичну інфраструктуру. Автоматичні процеси побудовані так,

щоб не було простоїв ресурсів і не витрачалися гроші на невикористані ресурси. Хмарні рішення ідеально підходять для компаній із зростаючими або змінними вимогами до пропускної здатності. У міру зростання потреб вашого бізнесу ви можете легко збільшити потужність хмари, не інвестуючи у фізичну інфраструктуру та не покладаючись на автоматичне масштабування. Автоматичне масштабування в хмарному середовищі.

3.3 Переваги використання згорткових нейронних мереж

Один із найкращих алгоритмів розпізнавання та класифікації зображень. Порівняно з повнозв'язаною нейронною мережею (наприклад, персептроном), існує набагато менше настроюваних ваг, оскільки одне ядро ваг повністю використовується для всього зображення, замість того, щоб мати власні особисті вагові коефіцієнти для кожного пікселя вхідного зображення. Це заохочує нейронну мережу узагальнювати інформацію, що відображається під час навчання, а не попіксельну пам'ять кожного показаного зображення за допомогою безлічі вагових коефіцієнтів, як це робить персептрон.

Зручне розпаралелювання обчислень, а отже можливість реалізації алгоритмів роботи та навчання мережі на GPU або TPU.

Сьогодні ми маємо доступ до багатьох хмарних сервісів, які підтримують роботу в згорткових нейронних мережах. Користувачі таких хмарних систем можуть ефективно розробляти, розгортати та підтримувати великі системи ШІ за допомогою згорткових мереж. При цьому багато корисних властивостей використання хмарних технологій будуть присутні, особливо масштабування. З економічної точки зору хмарні системи дозволяють ефективно витрачати гроші на проект навіть при використанні вузлів з використанням GPU, що дуже корисно при навчанні згорткових мереж. Cloud AI, Google Cloud. Google Cloud AI Platform, наше середовище розробки наукових даних на основі коду, дозволяє розробникам ML і дослідникам даних швидко переходити проекти від ідеї до розгортання.

3.4 Препроцесінг зображень

Попередня обробка зображень — це кроки, які виконуються для форматування зображень перед тим, як вони будуть використані в навчанні моделі та висновках.

Науковці, розробники та інженери комп'ютерного бачення мають контроль над вхідними даними, навіть якщо вони не є основним агентом, який збирає дані. Більше того: кроки, вжиті у конвеєрі введення зображення, можуть перетворити те, що колись було високоякісними даними, на вхідні дані з нижчим сигналом. Це не означає, що якість даних не є пріоритетною проблемою. Прагнення зібрати високоякісні дані для поставленого завдання завжди важливо. Але є випадки, коли інженери глибокого навчання можуть сліпо застосовувати етапи попередньої обробки та доповнення, які знижують продуктивність моделі на тих самих даних. І навіть за наявності високоякісних даних попередня обробка дозволяє отримати найкращі результати. Розуміння того, що є суттю попередньої обробки та доповнення, дає науковцям змогу отримати максимальну віддачу від своїх вхідних даних.

Збільшення зображень — це маніпуляції, застосовані до зображень для створення різних версій подібного вмісту, щоб надати модель більш широкому спектру навчальних прикладів. Наприклад, випадкова зміна повороту, яскравості або масштабу вхідного зображення вимагає, щоб модель врахувала, як виглядає об'єкт зображення в різних ситуаціях. Маніпуляції із збільшенням зображення є формами попередньої обробки зображення, але є критична відмінність: у той час як кроки попередньої обробки зображення застосовуються до навчальних і тестових наборів, розширення зображення застосовується лише до навчальних даних. Таким чином, перетворення, яке може бути доповненням в одних ситуаціях, може найкраще бути етапом попередньої обробки в інших. Подумайте про зміну контрастності зображення. Даний набір даних може містити зображення з низьким контрастом. Якщо

модель використовуватиметься у виробництві лише з низьким контрастом у всіх ситуаціях, вимога, щоб кожне зображення проходило постійне коригування контрасту, може покращити продуктивність моделі. Цей етап попередньої обробки буде застосовано до зображень під час навчання та тестування. Однак, якщо зібрані навчальні дані не є репрезентативними для рівнів контрастності, які модель може бачити у виробництві, є менша впевненість у тому, що коригування постійного контрасту доречне. Натомість випадкова зміна контрасту зображення під час навчання може краще узагальнювати. Це було б збільшенням. Щоб приймати обґрунтовані рішення щодо попередньої обробки та доповнення, необхідно знати контекст для збору даних і моделювання.

Попередня обробка потрібна для очищення даних зображення для введення моделі. Наприклад, повністю пов'язані шари в згорткових нейронних мережах вимагали, щоб усі зображення були масивами однакового розміру. Попередня обробка зображення також може зменшити час навчання моделі та збільшити швидкість визначення моделі. Якщо вхідні зображення особливо великі, зменшення розміру цих зображень значно покращить час навчання моделі без значного зниження продуктивності моделі. Наприклад, стандартний розмір зображень на iPhone 11 становить 3024×4032 . Модель машинного навчання, яку Apple використовує для створення масок і застосування портретного режиму, працює на зображеннях, розмір яких вдвічі менший, перш ніж його результат буде змінено до повного розміру. Збільшення зображення створює нові навчальні приклади з наявних навчальних даних. Неможливо посправжньому зафіксувати зображення, яке враховує кожен сценарій реального світу, який може охоплювати модель.

Коригування наявних навчальних даних для узагальнення на інші ситуації дозволяє моделі навчатися з більш широкого кола ситуацій. Це особливо важливо, коли зібрані набори даних можуть бути невеликими. Модель глибокого навчання буде підходити до прикладів, показаних у навчанні, тому створення варіацій у вхідних зображеннях дає змогу генерувати

нові, корисні навчальні приклади.

3.5 Архітектура додатку для обробки зображень

Для того щоб ефективно застосовувати згорткову нейронну мережу потрібно виходити з того, що ми хочемо мати за кінцевий результат. Після цього буде прийматись рішення для вибору архітектури додатку, а саме: як дані будуть отримуватись, в якому форматі, який тип мережі, ефективність моделі, валідація моделі та інші супутні рішення. На даний момент є цілий ряд мереж, які себе чудово себе презентують, як моделі для обробки зображень. Мова йде про CoAtNet - клас нових моделей, який вийшов із вивчення масштабування (скейлінг, scaling) моделей та балансування між собою глибини та ширини (кількості каналів) мережі, а також якості зображень у мережі. На рисунку 3.2 проілюстрована архітектура CoAtNet:



Рисунок 3.2 – Архітектура CoAtNet

Цей клас мереж ідеально підходить для вирішення задач детекції об'єктів на зображенні, що на теперішній час є одним з найбільш часто вживаних методів застосування згорткових нейронних мереж.

Для того щоб використовувати цю мережу, потрібно вилучити дані та підготувати її для навчання. На даний момент є багато інструментів для парсингу відкритих датасетів та інших ресурсів. Для оркестрації буде використана технологія Airflow, щоб автоматизувати процес та задати йому певний розклад. Таким чином це дозволить використовувати та підтримувати

додаток малими зусиллями. На рисунку 3.3 зображен процес збору даних до обробки:

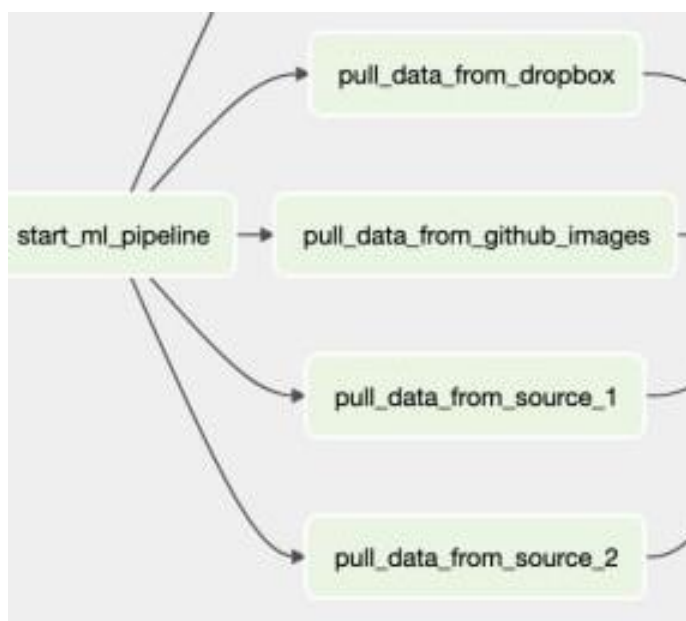


Рисунок 3.3 – Приклад збору даних для підготовки зображень

Після того, як будуть проведені усі етапи зі збору необхідних зображень, доведеться провести етапи для підготовки зображень у навчанні моделі, щоб уникнути подальших помилок у розробці та неналежного результату. Препроцесінг буде йти у декілька етапів та дозволить поліпшити наші дані, уніфікувати їх та збільшити їх кількість. На рисунку 3.4 зображен процес препроцесінгу:

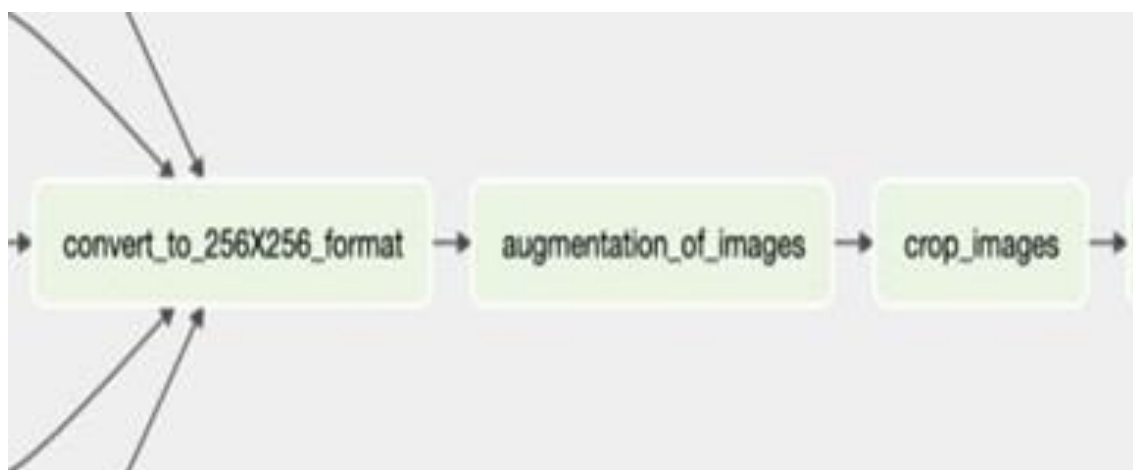


Рисунок 3.4 – Приклад препроцесінгу у пайплайні навчання моделі

Так як CoAtNet очікує зображення 256 на 256 пікселів тож доведеться деякі зображення редагувати, все залежить від датасету, який буде використовуватись у застосунку. Після цього буде проводитись аугментація даних. Це саме процес кількісного збагачення даних шляхом змінення гами, зміни повороту, орієнтації зображення, зміни контрасту та багато інших підходів. Для цього можна використовувати вже предтреновані моделі, наприклад іншу згорткову нейронну мережу EfficientNet або ResNet. Після цього наш датасет стає більш різноманітним та доповнює оригінальні зображення. На рисунку 3.5 зображен процес тренування з оптимізацією та подальшим збереженням результатів:



Рисунок 3.5 – Приклад етапу тренування та валідації моделі

Отже, були зібрані дані та оброблені, згорткова нейронна мережа може ефективно тренуватись на достатній кількості даних, доконфігуруватись та валідувати свої результати. Попередньо вказаний план дій для навчання згорткової нейронної мережі веде до того, що наша мережа постійно може оновлювати свої дані за допомогою оркестрації та вибудовувати пайплайн для збереження вагових коефіцієнтів для подальшого використання моделі.

3.6 Програмні засоби для реалізації аналізу над згортковими нейронними мережами

Виходячи з поставленого завдання, нам потрібно побудувати нейронну мережу за допомогою фреймворку, який має вбудовані функції для реалізації архітектури нашої моделі. Одним із таких представників є Tensorflow, який широко використовується як професіоналами, так і любителями у сфері машинного навчання [20]. Ця платформа з відкритим кодом була розроблена Google і підтримується користувачами з усього світу. Платформа використовується в багатьох корпоративних додатках, будь то голосовий помічник або обробка фотографій.

TensorFlow дозволяє розробникам створювати графи потоків даних: структури, які описують, як дані переміщуються через графік або ряд вузлів обробки. Кожен вузол на графі представляє математичну операцію, а кожне з'єднання або ребро між вузлами є тензором або багатовимірною матрицею даних. TensorFlow надає все це для програміста, який використовує мову Python. Вузли та тензори в TensorFlow є об'єктами Python, а програми TensorFlow самі є програмами Python. Однак фактичні математичні операції не виконуються в Python. Бібліотеки трансформацій, доступні через TensorFlow, написані як високопродуктивні двійкові файли на C++. Python просто направляє трафік між шардами та забезпечує абстракції програмування високого рівня для їх з'єднання між собою. На рисунку 3.6 проілюстрований процес з застосуванням Tensorflow pipeline:

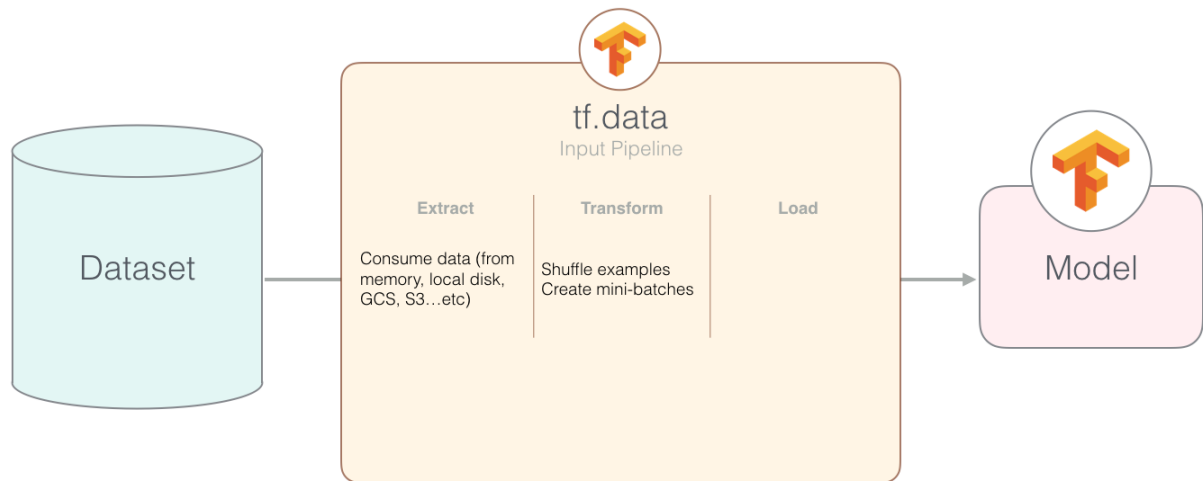


Рисунок 3.6 – Конвеєрна обробка даних на прикладі Tensorflow

Найбільшою перевагою TensorFlow для розвитку машинного навчання є абстракція. Замість того, щоб займатися реалізації деталей алгоритмів або з'ясовувати правильні способи приєднання виходу однієї функції до входу іншої, розробник може зосередитись на загальній логіці програми.

Програмні застосунки написані на TensorFlow можна запускати на різноманітних пристроях:

- на локальній машині;
- кластері в хмарі;
- пристроях iOS та Android;
- центральних процесорах або графічних процесорах.

Також можна використовувати власне хмарне середовище у будь-якому хмарному провайдері та запустити TensorFlow на TensorFlow Processing Unit (TPU) для подальшого прискорення. Отримані в результаті моделі, створені TensorFlow, можуть бути розгорнуті на більшості пристроїв, де вони будуть використовуватися для обчислення прогнозів, обробки зображень та інші задачі машинного навчання.

4 КОМБІНУВАННЯ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ З ІНШИМИ ШНМ

4.1 Трансформери як елемент комбінування

Перший елемент комбінування, архітектура кодера Transformer [21] показана нижче на рисунку 4.1. У впровадженні вхідних даних вхідні дані токенизуються, кожен маркер відображається на ціле число зі словника і, нарешті, кожне ціле число відображається на k -вимірному векторі, який модель вивчатиме під час навчання. Після цього додається позиційне кодування на основі синусоїдальної функції, щоб подбати про порядок слів у реченні. Нарешті, додається N_x блоків, що складаються з мережі самоконтролю з кількома головками та прямої мережі, за якими слідує залишкове з'єднання та нормалізація рівня. На рисунку 4.1 вказан алгоритм позиційного кодування з архітектури Transformer:

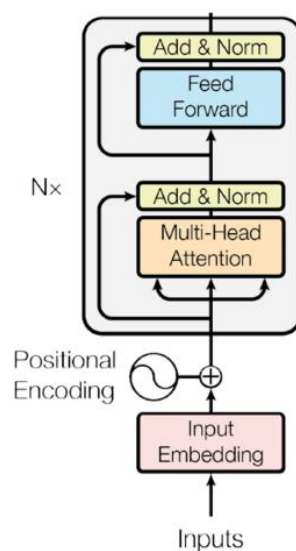


Рисунок 4.1- Етап кодування для частини з моделі трансформера

Блок уваги з декількома головами кілька разів обчислює самоувагу з різними ваговими матрицями, а потім об'єднує результати разом, розмір яких

змінюється до розміру embedding за допомогою іншої матриці, яку можна навчити, щоб вивести вектор того самого розміру, що й вхідний, для мати можливість передати його до наступного блоку. Кожне embedding має мати три асоційованих представлення, запит (Q), ключ (K) і значення (V), які отримують шляхом множення вхідних даних на три матриці вагових коефіцієнтів, які можна навчити. Інтуїтивно зрозуміло, що Q є представленням поточного слова, яке використовується для порівняння з усіма іншими словами (ми дбаємо лише про запит, який ми обробляємо). K – це як купа міток, які ми порівнюємо з Q під час пошуку відповідних слів. V містить фактичне представлення слів.

Скалярний добуток Q і K обчислюється для отримання матриці Score (S), яка представляє, наскільки релевантним є кожен ключ у порівнянні з кожним запитом, і ця матриця масштабується та передається через функцію softmax по рядках. Нарешті, скалярний добуток між S і V обчислюється для отримання контекстуалізованих вкладень, тобто матриці самоуважності. Ця матриця, після вилучення, додавання пропуску підключення та нормалізації рівня, передається позиційно (це означає, що вона приймає один маркер за раз, але ваги розподіляються, а різні результати складаються в матрицю) FFN, який спочатку розширює вхідні дані, а потім знову стискає їх, дотримуючись подібної поведінки блоку MBConv саме це будемо використовувати в CoAtNet моделі, і додає нелінійність за допомогою активації ReLU. На рисунку 4.2 проілюстровано процес обробки вхідного значення для отримання контекстуалізованих embeddings.

Ця архітектура може мати справу з повними послідовностями, таким чином вивчаючи довгострокові зв'язки та може бути легко розпаралелена, також є масштабованою до моделей високої складності та великомасштабних наборів даних. Відкриття трансформерних мереж у сфері обробки природної мови викликало великий інтерес у спільноті комп'ютерного зору. Однак візуальні дані мають типову структуру, що вимагає нових проектів мережі та схем навчання. У результаті різні автори запропонували власну реалізацію моделі трансформера, застосованої до зору, але SOTA було досягнуто лише за

допомогою трансформера зору (ViT) [22].

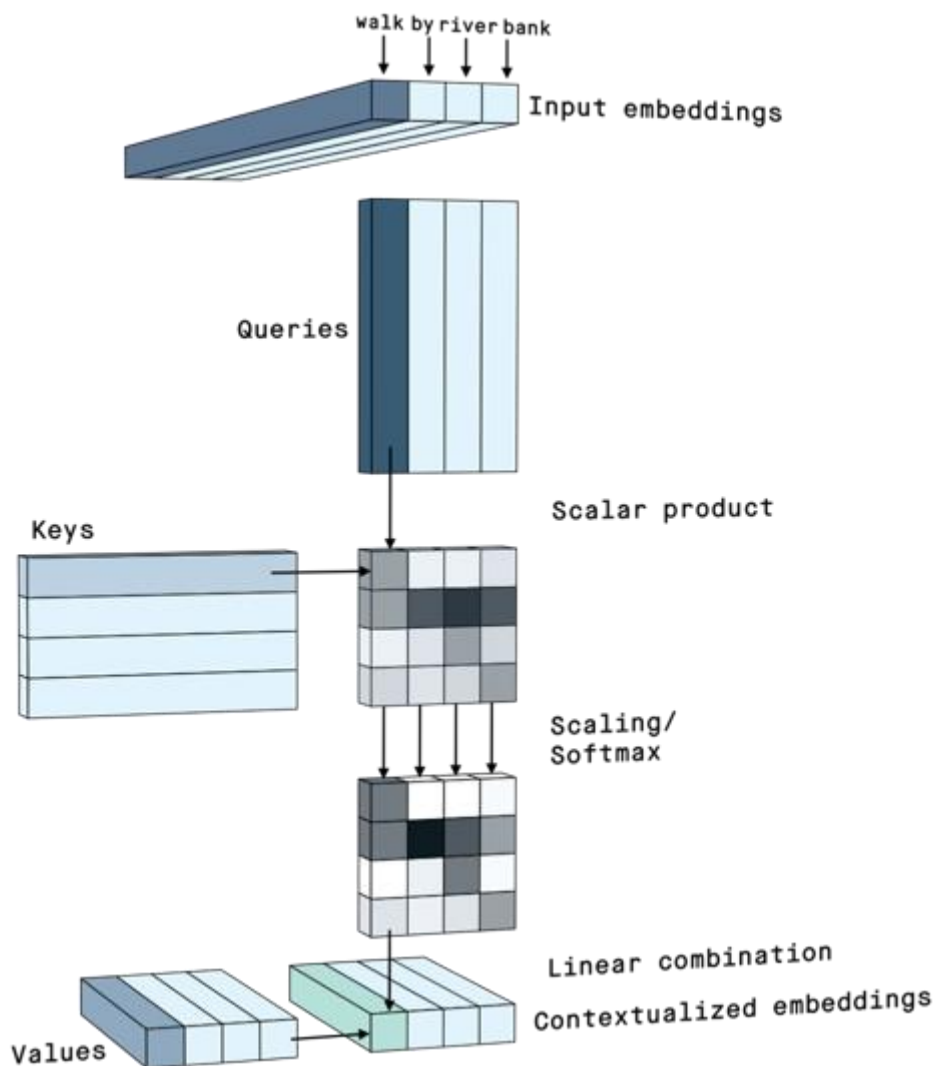


Рисунок 4.2 – Механізм уваги на прикладі вхідного embedding

Ця архітектура зосереджена на невеликих ділянках зображення, які розглядаються як токени. Кожен патч у вхідному зображенні вирівнюється за допомогою матриці лінійної проекції, і до нього додається позиційне embedding, яке можна вивчати. Це позиційне embedding є одновимірним і розглядає вхідні дані як послідовність патчів у растровому порядку. У CoAtNet це буде замінено відносним положенням між патчами замість їхнього абсолютного положення, що призведе до визначення відносної уваги до себе. Щоб виконати класифікацію, до послідовності додається додатковий маркер

класифікації (позначений * на зображенні нижче). Кодер ViT, подібно до оригінальної версії, розглянутої вище, складається з кількох блоків самоуважності, нормалізації та повністю пов'язаних шарів із залишковими зв'язками. У кожному блоці уваги кілька головок можуть фіксувати різні шаблони з'єднання. Повністю підключена голова багаторівневого перцептрона на виході класифікації забезпечує передбачення бажаного класу. На рисунку 4.3 зображена архітектура візуального трансформера:

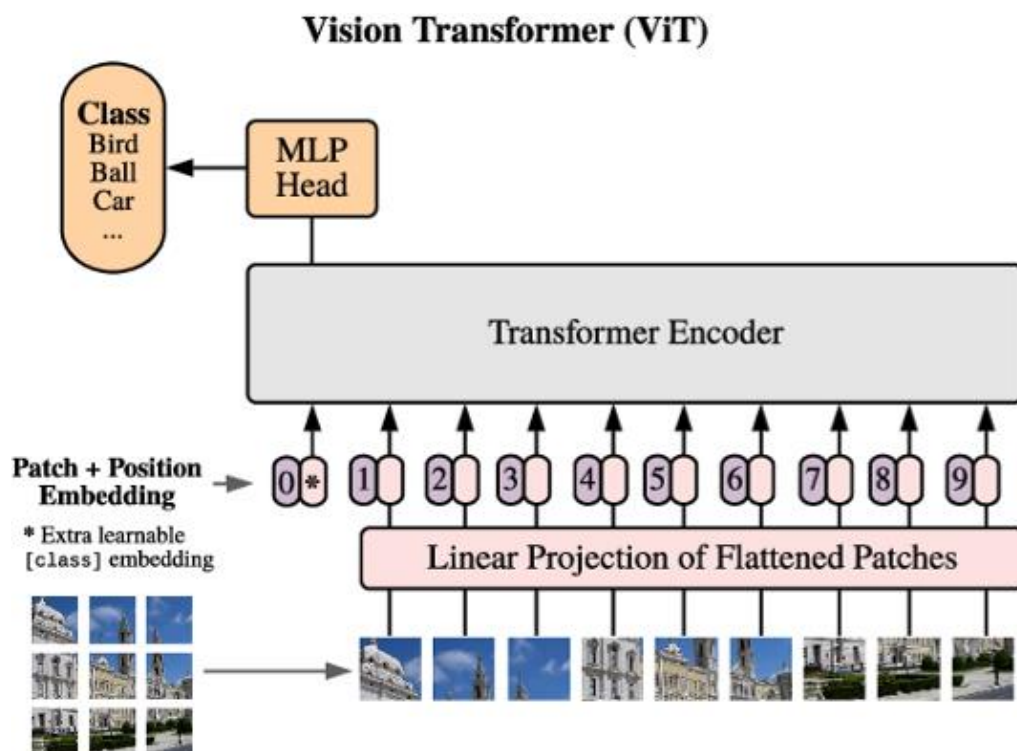


Рисунок 4.3 – Архітектура візуального трансформера

4.2 CoAtNet

ViT ефективна модель яка б могла посунути класичним CNN, якщо тільки використовувати дуже великі датасети. Це свідчить про те, що Transformers може не мати можливості узагальнення, якою володіють CNN, і тому для компенсації потрібна значна кількість даних. Тим не менш, моделі уваги мають вищу модельну здатність порівняно з CNN.

Тому однієї з цілей кваліфікаційної роботи є використання переваг трансформерів та поєднання переваг CNN в одній архітектурі з точки зору узагальнення та ємності моделі на базі CoAtNet [23].

Для того, щоб правильно поєднати ці два методи обробки зображень, нам потрібно використати вже розглянутий блок MBConv, який використовує згортку по глибині з перевернутим залишковим вузьким місцем, по-перше, тому що ця схема розширення-стиснення є спільною з модулем FFN Transformer. На додаток до цієї подібності, як глибинна згортка, так і самоувага можуть бути виражені як зважена сума значень за виміром у попередньо визначеному рецептивному полі. Згортку по глибині можна виразити:

$$y_i = \sum_{j \in L(i)} w_{i-j} \times x_j, \quad (4.1)$$

де x_i і y_i — вхід і вихід у позиції і відповідно,

w_{i-j} — вагова матриця в позиції $(i - j)$,

y_i — значення глибокої згортки,

$L(i)$ — локальне оточення i .

Як уже було зазначено, згортка по глибині виконує згортку для кожного каналу окремо.

Для порівняння, механізм уваги обчислює ваги на основі парної подібності, за якою слідує функція softmax:

$$y_i = \sum_{j \in \delta} \frac{e(x_i^T x_j)}{\sum_{k \in \delta} e(x_i^T x_k)} x_j, \quad (4.2)$$

де δ позначає глобальний простір,

y_i — значення механізму уваги (self-attention),

j — позиція (індекс) елементу пари, який належить простіру δ ,

x_i, x_j — дві пари (наприклад, дві ділянки зображення).

Надто спрощене зображення (без проєкцій Q, K і V із кількома головами)

лише для розуміння показано нижче: кожен фрагмент порівнюється з кожним іншим фрагментом на тому самому зображенні для створення матриці самоуважності.

Адаптивне зважування вхідних даних: матриця $w_i - j$ є незалежним від вхідних даних статичним значенням, тоді як ваги уваги A_{ij} залежать від представлення вхідних даних. Це робить самоувагу більш схильною фіксувати зв'язки між різними елементами у вхідних даних, ціною ризику перевибору, коли дані обмежені.

Еквіваріантність перекладу: вага згортки $w_i - j$ дбає про відносний зсув між i та j , а не про конкретні значення i та j . Доведено, що ця еквіваріантність перекладу покращує узагальнення в обмеженому наборі даних.

Глобальне сприйнятливості поле: більше рецептивне поле, яке використовується в самоуважності, надає більше контекстуальної інформації порівняно з локальним рецептивним полем CNN.

Підводячи підсумок, оптимальна архітектура включає в себе адаптивне зважування введення та характеристики глобального сприйнятливого поля самоуважності та еквіваріантність перекладу CNN. Ідея, запропонована авторами, полягає в тому, щоб підсумувати ядро глобальної статичної згортки з адаптивною матрицею уваги після або до ініціалізації softmax.

$$y_i = \sum_{j \in \delta} \left(\frac{e(x_i^T x_j)}{\sum_{k \in \delta} e(x_i^T x_k)} + w_{i-j} \right) x_j, \quad (4.3)$$

де δ позначає глобальний простір,

y_i — пренормалізоване значення механізму уваги,

j — позиція (індекс) елемента пари, який належить простіру δ ,

x_i, x_j — дві пари (наприклад, дві ділянки зображення),

w_{i-j} — вагова матриця в позиції $(i - j)$.

Версія до нормалізації, яка була використана в роботі, також відповідає

певному варіанту відносної самоуваги, про який ми вже згадували раніше:

$$y_i = \sum_{j \in \delta} \frac{e(x_i^T x_j + w_{i-j})}{\sum_{k \in \delta} e(x_i^T x_k + w_{i-j})} + w_{i-j} x_j, \quad (4.4)$$

де δ позначає глобальний простір,

y_i — нормалізоване значення механізму уваги після softmax

j — позиція (індекс) елементу пари, який належить простіру δ ,

x_i, x_j — дві пари (наприклад, дві ділянки зображення),

w_{i-j} — вагова матриця в позиції $(i - j)$.

Наступний крок — з'ясувати, як скласти згортку та блоки уваги. Тут є декілька підходів до вирішення цієї проблеми, але розглянемо використання згортки для зменшення вибірки та глобальної відносної уваги лише після того, як розмір карт ознак стане достатньо малим для обробки. Зменшення дискретизації можна виконати двома способами:

- розподіл зображень на фрагменти, як у моделі ViT, і стек відносних блоків самоуваги. Цю модель використовували для порівняння з оригінальною реалізацією ViT;
- використання багатоетапного макета з поступовим об'єднанням. Цей підхід розділено на 5 етапів, але перші два, які є класичним згортковим шаром і блоком MBConv, який використовується для зниження розмірності, тут для простоти об'єднані в один, названий S0. Останні три етапи можуть бути блоком Convolution або Transformer, що призводить до 4 комбінацій: S0-CCC, S0-CCT, S0-CTT і S0-TTT.

Після тестування CoAtNet мережі з поєднанням CNN та мехізму уваги від моделей трансформерів на локальній машині з RTX 3070 з VGGNet, ResNet, CoAtNet – 1, CoAtNet – 3 були зроблені висновки, що далі тестувати варіанти зі збільшеннями глибини згорткових ознак не має сенсу та має бути проведеним на більш продуктивній машині. Порівняння результатів наведені у таблиці 4.1. Результат епохи під номером 10 відображений на рисунку 4.4:

```

Epoch 10/10
391/391 [=====] - 827s 2s/step - loss: 0.5519 -
sparse_categorical_accuracy:0.8089-val_loss:1.7189
Process finished with exit code 0

```

Рисунок 4.4 – Результат останньої епохи перед завершенням навчання

Таблиця 4.1 – Таблиця результатів після навчання на однаковому датасеті декількох моделей на базі CNN

Мережа	Метрика	Результат	Час навчання
VGGNet	sparse_categorical_accuracy	74.2	5.25 год.
ResNet	sparse_categorical_accuracy	76.3	4.37 год.
CoAtNet - 1	sparse_categorical_accuracy	78.1	4.41 год
CoAtNet - 3	sparse_categorical_accuracy	80.1	7.32 год

Після тестування усіх моделей наведених у таблиці 4.1 ми можемо побачити, що модель з CNN з комбінованим елементом з архітектури Transformer має перевагу над іншими архітектурами та зі збільшенням поглиблення шарів стає більш результативною.

ВИСНОВКИ

В атестаційній роботі було розглянуто методи обробки зображень на базі згорткових нейронних мереж, які відповідають поставленим вимогам та вирішують актуальну проблему за допомогою методів машинного навчання застосовуючи комп'ютерний зір забезпечуючи велику точність та гнучкість програмних засобах у модифікації складових частин .

Для вирішення поставленої задачі були використані два компонента з CNN та Transformer архітектур, щоб провести демонстрацію поєднання двох типів моделей для досягнення кращих результатів в порівнянні зі звичайними згортковими мережами. Основна мова програмування та обробки даних - Python.

Щоб використовувати моделі машинного застосувався фреймворк Tenserflow, Keras та Airflow. Гнучкість підібраних інструментів та підібраної архітектури дозволить вносити зміни без кардинальних втрат часу та ресурсів.

Під час досліджень задля вирішення задачі комп'ютерного зору були зроблені певні висновки:

- від обраної архітектури та її компонентів залежить ефективність моделі машинного навчання. У весь час потрібно балансувати між витратами ресурсів та точністю результатів;

- від якісних даних залежить лєвова частка результату у розпізнанні образів;

- потрібно проводити тестування програмних засобів, щоб якість та функціональність програмного забезпечення відповідала вимогам користувача;

Можливе подальше перенесення інфраструктури на Google AI Platform для збільшення ефективності використаних ресурсів комп'ютерів або кластерів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Руденко О.Г., Бодянський Є.В. Штучні нейронні мережі: Навчальний посібник. – Київ : Компанія СМІТ, 2006. – 404 с.
2. Бодянский Е. В., Руденко О. Г. Искусственные нейронные сети: архитектуры, обучение, применения //Харьков: Телетех. – 2004. – Т. 369.
3. Gao M, Bagci U, Lu L, Wu A, Buty M (2018) Holistic classification of CT attenuation patterns for interstitial lung diseases via deep convolutional neural networks. CMBBE Imaging Vis 6(1):1–6
4. Ghoshal B, Tucker A (2020) Estimating uncertainty and interpretability in deep learning for coronavirus (COVID-19) detection. // arXiv:2003.10769. [Електронний ресурс]. – Режим доступу: <https://arxiv.org/abs/2003.10769> (дата звернення 10.12.2022).
5. Gao XW, Hui R (2016) A deep learning based approach to classification of CT brain images. In: Proceedings of the SAI computing conference, pp 28–31
6. Jia, Y., Shelhamer E, Donahue J (2014) Caffe: convolutional architecture for fast feature embedding. In: Proceedings of the 22th ACM international conference on multimedia, MM '14, pp 675–678. ACM, New York
7. Gao XW, Li W, Loomes M, Wang L (2017) A fused deep learning architecture for viewpoint classification of echocardiography. Inf Fusion 36:103–113
8. Warren S. McCulloch, Wallter H. Pitts (1943) A Logical calculus of the ideas immanent in nervous activity, Bulletin Of The Mathematical Biophysics, vol. 5, p.115-133.
9. Schmidhuber, J. Deep Learning in Neural Networks: An Overview / J. Schmidhuber // Neural Networks. — 2015. — Vol. 61. — P. 85–117.
10. Хайкин С. Нейронные сети: полный курс, 2-е издание. – М.: Вильямс, 2008. – 1103 с.
11. Zewen Li, Wenjie Yang, Shouheng Peng, Fan Liu (2018) A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects // arxiv:2004.02806. – [Електронний ресурс]. – Режим доступу:

- <https://arxiv.org/abs/2004.02806> (дата звернення 10.12.2022).
12. Флах П. Машинное обучение. Наука и искусство построения алгоритмов, которые извлекают знания из данных. – М.: ДМК Пресс, 2015. – 400 с.
 13. Behnam Neyshabur / Towards Learning Convolutions from Scratch Behnam Neyshabur // arXiv:2007.13657 . – [Электронный ресурс]. – Режим доступа: <https://arxiv.org/abs/2007.13657> (дата звернення 10.12.2022).
 14. Xiao Zhang, Yaodong Yu, Lingxiao Wang, Quanquan Gu / Learning One-hidden-layer ReLU Networks via Gradient Descent // arXiv:1806.07808. – [Электронный ресурс]. – Режим доступа: <https://arxiv.org/abs/1806.07808> (дата звернення 10.12.2022).
 15. Hossein Gholamalinezhad, Hossein Khosravi / Pooling Methods in Deep Neural Networks, a Review // arXiv:2009.07485. – [Электронный ресурс]. – Режим доступа: <https://arxiv.org/abs/2009.07485> (дата звернення 10.12.2022).
 16. S. Ren, K. He, R. Girshick, J. Sun / Faster R-CNN: Towards real-time object detection with region proposal // arXiv:1506.01497 - [Электронный ресурс]. – Режим доступа: <https://arxiv.org/abs/1506.01497> (дата звернення 10.12.2022)
 17. Kaiming He, Georgia Gkioxari, Piotr Dollár, Ross Girshick / Mask R-CNN // arXiv:1703.06870 - [Электронный ресурс]. – Режим доступа: <https://arxiv.org/abs/1703.06870> (дата звернення 10.12.2022)
 18. Zekun Luo, Zheng Fang, Sixiao Zheng, Yabiao Wang, Yanwei Fu / NMS-Loss: Learning with Non-Maximum Suppression for Crowded Pedestrian Detection // arXiv:2106.02426 - [Электронный ресурс]. – Режим доступа: <https://arxiv.org/abs/2106.02426> (дата звернення 10.12.2022)
 19. OECD. (2019). The Digitalisation of Science, Technology and Innovation. Science and Technology, (1), 9. <https://doi.org/10.1787/b9e4a2c0-en>
 20. Tensorflow Object Detection API / J. Huang, V. Rathod, R. Votel et al // GitHub. – [Электронный ресурс]. – Режим доступа:

https://github.com/tensorflow/models/tree/master/research/object_detection

(дата звернення 10.12.2022).

21. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin / Attention Is All You Need //arXiv:1706.03762 - [Електронний ресурс]. – Режим доступу: <https://arxiv.org/abs/1706.03762> (дата звернення 10.12.2022)
22. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale /Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby // arXiv:2010.11929 – [Електронний ресурс]. – Режим доступу: <https://arxiv.org/abs/2010.11929> (дата звернення 10.12.2022)
23. CoAtNet: Marrying Convolution and Attention for All Data Sizes / Zihang Dai, Hanxiao Liu, Quoc V. Le, Mingxing Tan // arXiv:2106.04803 - [Електронний ресурс]. – Режим доступу: <https://arxiv.org/abs/2106.04803> (дата звернення 10.12.2022)