Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

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

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

рівень вищої освіти другий (магістерський)

Розпізнавання зображень за допомогою штучної нейронної мережі Mask RCNN

Виконав:

студент 2 курсу, групи КІТм-21-1
Юрій ПОНАЧЕВНИЙ
Спеціальність 123 Комп'ютерна інженерія
Тип програми освітньо-професійна
Освітня програма Комп'ютерні інтелектуальні технології
Керівник професор Микола КОРАБЛЬОВ

Допускається до захисту

Зав. кафедри _____Олег РУДЕНКО

(підпис)

2023 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)
«_____»_____202_ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Поначевному Юрію Андрійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розпізнавання зображень за допомогою штучної нейронної мережі Mask R-CNN

затверджена наказом по університету від " 03 " листопада 2023 р. № 1290Ст

2. Термін подання студентом роботи до екзаменаційної комісії 13.01.2024

3. Вхідні дані до роботи

1) мова програмування Python
2) бібліотека m_crnn
3) методи обробки зображення

4. Перелік питань, що потрібно опрацювати в роботі

1) аналіз проблеми та огляд існуючих рішень;
2) вибір технології розробки та інструментальних засобів;
3) підготовка інструментів для подальшої реалізації;
4) розробка;
5) висновки.

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

Слайд-презентація 11 слайдів

6. Консультанти розділів роботи и (п.6 включається до завдання за наявністю 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 проблеми та огляд існуючих рішень	10.11 - 15.11	Виконано
2	Вибір технології розробки та	16.11 - 25.12	Виконано
	інструментальних засобів		
4	Підготовка інструментів для подальшої	25.12 - 28.12	Виконано
	реалізації		
6	Розробка	29.12 - 05.01	Виконано
7	Оформлення матеріалів кваліфікаційної	06.01 - 10.01	Виконано
	роботи		

Дата видачі завдання 06 листопада 2023 р.

Студент Поначевний Ю.А
(підпис)

Керівник роботи _____ Корабльов М.М.
(підпис) (посада, ім'я, прізвище)

РЕФЕРАТ

ШТУЧНА НЕЙРОННА МЕРЕЖА, РОЗПІЗНАВАННЯ ОБ’ЄКТІВ, РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ, НАВЧАННЯ МЕРЕЖІ, ЗБІР ДАНИХ, ФОРМУВАННЯ СЕТУ ЗОБРАЖЕНЬ, MASK RCNN.

Метою кваліфікаційної роботи є розпізнавання нових класів на зображеннях у штучній нейронній мережі, а саме проектування, навчання та розробка, що дозволять використовувати оброблені зображення для різних систем.

У роботі розглядається актуальність розробки рішення по розпізнаванню дорожних знаків за допомогою штучної нейронної мережі Mask RCNN, зроблено аналіз проблеми та оглянуті існуючі рішення. Виконано вибір технології розробки та інструментальних засобів, та розроблені рішення за допомогою штучної нейронної мережі Mask RCNN.

Об’єктом дослідження є штучна нейронна мережа Mask RCNN.

Предметом дослідження є розпізнавання дорожніх знаків за допомогою штучної нейронної мережі.

Під час виконання кваліфікаційної роботи було проведено навчання нейронної мережі та модифікацію штучної нейронної мережі на існуючих веб-додатках, проаналізовано їх переваги та вдосконалення, а також розроблено новий клас для спрощення процесу пошуку знаків дорожнього руху на зображеннях.

Ця розробка моє позитивно вплинути на розвиток автовиробництва електроніки, розвиток камер на певних участках дороги для дослідження порушувачів дорожнього руху.

Пояснювальна записка кваліфікаційної роботи: 67 с., 34 рис., 16 джерел.

ABSTRACT

ARTICLE NEURON MEASURE, OBJECT RECOGNITION, IMAGE RECOGNITION, MEASURE NATIVITY, DATA ACQUISITION, SETTING IMAGE FORMATION, RCNN MASK.

The method of qualified work is the recognition of new classes in the images of the artificial neural network, and the design itself, the development has begun, which allows for the development of complex images for different systems.

The robot will examine the relevance of developing a solution for recognizing road signs using an additional artificial neural measure Mask RCNN, analyze the problem and look at existing solutions. After which we can proceed to the choice of technology for developing and instrumental features, and then proceed to developing a solution using the additional artificial neural network Mask RCNN.

The object of investigation is a piece of neural network Mask RCNN.

The subject of research is the recognition of road signs using an additional neural network.

During the course of qualified work, neural measurements and modifications of custom neural measurements were carried out on existing web applications, their advantages were analyzed in detail, and a new class was developed to simplify the process of searching for signs. road traffic in the images.

This development can have a positive impact on the development of automobile electronics, the development of cameras on the first sections of the road to track road traffic offenders.

Master’s thesis:: 67 pages, 34 figures, 16 sources.

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп’ютерної інженерії та управління
Кафедра Комп’ютерних інтелектуальних технологій та систем

АНОТАЦІЯ

КВАЛІФІКАЦІЙНОЇ РОБОТИ

рівень вищої освіти другий (магістерський)

Розпізнавання зображень за допомогою штучної нейронної мережі Mask RCNN

Виконав:
студент 2 курсу, групи КІТм-22-1
Юрій ПОНАЧЕВНИЙ
Спеціальність 123 Комп’ютерна інженерія
Тип програми освітньо-професійна
Освітня програма Комп’ютерні інтелектуальні технології
Керівник професор Микола КОРАБЛЬОВ

2023 р.

АНОТАЦІЯ

Поначевний Ю.А. Розпізнавання зображень за допомогою штучної нейронної мережі Mask RCNN. – Магістерська кваліфікаційна робота.

У магістерській кваліфікаційній роботі вирішено актуальну задачу розпізнавання об’єктів на зображеннях за допомогою штучної нейронної мережі Mask RCNN.

Метою даної роботи є побудова структури по розпізнаванню об’єктів на зображеннях задля наступної переробки, що дозволить покращити і розширити сфери використання штучних нейронних мереж.

Об’єктом дослідження є процеси розпізнавання зображень з використанням штучної нейронної мережі Mask RCNN.

Предметом дослідження є автоматичне розпізнавання знаків дорожнього руху для автобудівництва, а саме автопілотів, та інтуїтивних підказок приладових панелей автомобілей.

Методи дослідження включають у себе:

1. інтелектуальний аналіз зображень;
2. методи прогнозування положення об’єктів на зображеннях.

Практична цінність отриманих результатів має позитивно вплинути на швидкість, правильність та доцільність аналізу дорожніх знаків для автопілотів, задля подальшого розвитку автобудівництва та камер дистанційного контролю дорожнього руху.

У першому розділі розглянуто підходи до розпізнавання зображень, методи розпізнавання зображень, моделі розпізнавання зображень, аналіз нейронних мереж, та саме які використовуються, наприклад: згорткові нейронні мережі CNN, ResNet, Inception, MobileNet, VGGNet.

Більш детально були розглянуті CNN нейронні мережі: R-CNN, Fast R-CNN, Faster R-CNN, та Mask R-CNN, яка і буде використана у розділі реалізації рішення.

У другому розділі була детально досліджено:

1. Розпізнавання зображень з використанням штучної нейронної мережі Mask RCNN.

Розвиток розпізнавання об'єктів досягається за допомогою штучної нейронної мережі Mask R-CNN. Метод під назвою Mask R-CNN розширює Faster R-CNN шляхом додавання гілки передбачення маски об’єкта паралельно до існуючої гілки виявлення обмежувальної рамки. Mask R-CNN легко навчити, і вона лише додає невеликі накладні витрати на Faster R-CNN. Плюси Mask R-CNN:

1. простота: Mask R-CNN легко тренується;

2. ефективність: метод є ефективним і додає невелику кількість витрат порівнянно з Faster R-CNN.

2. Навчання штучної нейронної мережі Mask RCNN

Навчання Mask RCNN складається з наступних етапів:

- підготовка даних: На цьому етапі потрібно зібрати набір даних зображень, позначених людьми. Описи повинні включати координати контурів об'єктів на зображенні, а також їх клас;

- конвертація даних: На цьому етапі потрібно конвертувати дані в формат, який може бути використаний для навчання нейронної мережі;

- налаштування параметрів нейронної мережі: На цьому етапі потрібно налаштувати параметри нейронної мережі, такі як швидкість навчання та розмір батча;

- навчання нейронної мережі: На цьому етапі нейронна мережа навчається на наборі даних зображень;

- експериментальна оцінка: На цьому етапі потрібно оцінити точність нейронної мережі на наборі даних зображень, який не використовувався для навчання.

У третьому розділі присвячений реалізацію рішення було розглянуто:

1. підготовка матеріалу перед реалізацією де в першу чергу необхідно створити так званий набір даних із зображеннями об'єкта, що розпізнається, в даному випадку це реалізація розпізнавання знаків дорожнього руху. Збираючи фотографії дорожнього руху, по можливості додавайте зображення не дуже високої

якості, тому що на наступних етапах штучна нейронна мережа буде легше виконувати етап навчання і легше розпізнавати зображення через меншу масу і вагу та кількість пікселів;

2. виконання анотації для датасету, де покроково описані ступені для підготовки анотацій через веб додаток;

3. конфігурація додатку по тренуванню нейронної мережі через бібліотеки та модулі язика програмування Python;

4. конфігурація додатку по функціонуванню штучної нейронної мережі для розпізнавання зображення.

ШТУЧНА НЕЙРОННА МЕРЕЖА, РОЗПІЗНАВАННЯ ОБ'ЄКТІВ, РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ, НАВЧАННЯ МЕРЕЖІ, ЗБІР ДАНИХ, ФОРМУВАННЯ СЕТУ ЗОБРАЖЕНЬ, MASK RCNN.

Публікації здобувача за темою роботи:

1. Поначевний Ю.А. Використання нейронної мережі Mask R-CNN для розпізнавання зображень // Матеріали IV Всеукраїнської студентської наукової конференції «Розвиток сучасної науки: актуальні питання теорії та практики», 17 листопада 2023, м. Львів, Україна.

ЗМІСТ

Скорочення та умовні познаки 12
Вступ 13
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ 17
1.1 Аналіз підходів, методів та моделей для розпізнавання зображень 17
1.1.1 Підходи до розпізнавання зображень 17
1.1.2 Методи розпізнавання зображень 17
1.2 Аналіз нейронних мереж, які використовуються для розпізнавання зображень 18
1.2.1 Згорткові нейронні мережі CNN 18
1.2.2 ResNet 19
1.2.3 Inception 19
1.2.4 MobileNet 19
1.2.5 VGGNet 20
1.3 Використання нейронних мереж 20
1.3.1 Сфери використання нейронних мереж 20
1.3.2 Різновиди нейронних мереж 21
1.3.3 Принцип роботи нейронної мережі 23
1.4 Аналіз існуючих рішень 28
1.4.1 Нейронна мережа R-CNN 28
1.4.2 Нейронна мережа Fast R-CNN 29
1.4.3 Нейронна мережа Faster R-CNN 29
1.4.4 Нейронна мережа Mask R-CNN 30
1.5 Постановка задачі 31
2 Технологія розпізнавання зображень з використанням штучної нейронної мережі Mask RCNN 32
2.1 Розпізнавання зображень з використанням штучної нейронної мережі Mask RCNN 32
2.2 Навчання штучної нейронної мережі Mask RCNN 36

3 РЕАЛІЗАЦІЯ РОЗПІЗНАВАННЯ ОБ'ЄКТІВ 38
3.1 Підготовка матеріалу перед реалізацією 38
3.2 Виконання анотації для датасету 38
3.3 Конфігурація додатку по тренуванню нейронної мережі 41
3.4 Конфігурація додатку по функціонуванню штучної нейронної мережі для розпізнавання зображення 48
Висновки 53
Перелік використаних джерел 54
Додаток А
О
шибка! Закладка не определена.
Графічний матеріал атестаційної роботи
О
шибка! Закладка не определена.
Додаток Б
О
шибка! Закладка не определена.3
СЕРТИФІКАТ 64
Наукова публікація 65

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

GPU – Графічний процесор

CNN – Згорткова нейронна мережа

R-CNN – Згорткова нейронна мережа на основі регіонів

RolPool – Метод вилучення карт об’єктів

RolAlign – Метод прогнозування масок

Датасет – Набір зображень для навчання

ВСТУП

Нейронні мережі — це тип машинного навчання, у якому комп’ютер вивчає, як виконувати завдання, аналізуючи навчальні зразки. Як правило, ці приклади заздалегідь позначаються вручну. Наприклад, система розпізнавання об’єктів може взяти тисячі позначених зображень автомобілів, будинків, кавових чашок тощо та знайти візуальні шаблони в цих зображеннях, які послідовно корелюють із певною міткою.

Нейронні мережі часто порівнюють з людським мозком, який також має мережу з тисяч або мільйонів простих вузлів обробки, тісно пов’язаних один з одним. Більшість сучасних нейронних мереж складаються з шарів вузлів, де дані проходять лише в одному напрямку. Один вузол може підключатися до кількох вузлів нижніх рівнів для отримання даних і до кількох вузлів верхніх рівнів для надсилання даних.

Вузол призначає номер («вагу») кожному з цих вхідних посилань. Коли мережа активна, вузол отримує інший запис (різне число) для кожного з цих з’єднань і множить його на відповідну вагу. Потім підсумуйте результати, щоб обчислити число. Якщо це число нижче порогового значення, вузол не пересилатиме дані на наступний рівень. Коли число перевищує порогове значення, вузол «прокидається», надсилаючи це число (суму зважених вхідних даних) усім вихідним з’єднанням.

Під час навчання нейронної мережі всі її ваги та пороги спочатку встановлюються випадковим чином. Навчальні дані подаються на найнижчий рівень, вхідний рівень, проходять через наступні рівні, множаться та підсумовуються складними способами, доки вони нарешті не досягнуть перетвореного вихідного рівня. Під час навчання ваги та порогові значення постійно коригуються, доки дані навчання з однаковими мітками не дадуть аналогічних результатів.

Нейронна мережа, описана Маккалоу та Піттсом у 1944 році, мала як пороги, так і ваги, але вона не була організована в шари, і вчені не вказали конкретний

механізм навчання. Але Маккалоу та Піттс показали, що нейронні мережі, в принципі, можуть обчислювати будь-яку функцію, як цифровий комп'ютер. Результати стосувалися більше нейронауки, ніж інформатики. Ми повинні були припустити, що мозок людини можна розглядати як обчислювальний пристрій.

Нейронні мережі продовжують залишатися цінними інструментами для досліджень нейронаук. Наприклад, правила встановлення окремих шарів, ваг і порогів у мережі можуть відтворювати спостережувані особливості нейроанатомії та когнітивних функцій людини, тим самим впливаючи на те, як мозок обробляє інформацію.

Першу тренувальну нейронну мережу, перцептрон, продемонстрував у 1957 році психолог Корнельського університету Френк Розенблат. Конструкція персептрона була схожа на сучасну нейронну мережу, за винятком того, що він мав один шар з регульованими вагами та порогами, затиснутими між вхідним і вихідним шарами.

«Персептрони» активно вивчалися в психології та інформатиці до 1959 року, коли Мінський і Пеперт опублікували книгу під назвою «Персептрони», яка показала, що звичайні обчислення з використанням персептронів були непрактичними з точки зору витрат часу. Однак до 1980-х років вчені розробили алгоритми для зміни ваг і порогових значень нейронної мережі, які були достатньо ефективними для мереж з більш ніж одним шаром і зняли багато обмежень, визначених Мінським і Пепертом. Ця теорія переживала ренесанс.

Але з розумної точки зору, в нейронних мережах чогось не вистачало. Після достатньо тривалого періоду навчання параметри мережі можна переглядати, поки не стане можливою значуща класифікація даних.

Останніми роками вчені розробили складні методи для визначення стратегій аналізу нейронної мережі. Але у 1980-х роках стратегії цих мереж були іншими. 9 Тому на рубежі століть нейронні мережі були замінені векторними машинами, альтернативним підходом до машинного навчання, заснованим на чистій математиці.

Нещодавній сплеск інтересу до нейронних мереж (революція глибокого навчання) можна віднести до індустрії комп'ютерних ігор. Складна графіка та

швидкий темп сучасних відеоігор вимагають апаратного забезпечення, яке може йти в ногу з тенденціями. Результатом є GPU (графічний процесор), який містить кілька тисяч відносно простих процесорних ядер на одному чіпі. Вчені швидко зрозуміли, що архітектури GPU ідеально підходять для нейронних мереж.

Сучасні графічні процесори дозволили побудувати мережі 1960-х років і дво- і трирівневі мережі 1980-х років у сучасні кластери з 10-, 15- і навіть 50-рівневих мереж. Ось що означає слово «глибокий» у «поглибленому навчанні». До глибини мережі. Наразі глибоке навчання відповідає за найефективніші системи майже в усіх сферах досліджень штучного інтелекту.

В даний час часті спроби автоматизувати трудомісткі завдання або замінити ручну роботу машинною.

Сучасній людині необхідно максимально спростити трудомісткі завдання, щоб заощадити час і зосередитися на більш важливих відкриттях і розробках. Тому зараз існують нейронні мережі для обробки та оптимізації різних процесів. У наш час вже існує велика кількість «навчених» нейронних мереж.

Досі швидкість нейронних мереж була надто низькою для широкого використання, тому такі системи в основному використовувалися для розробки в області комп’ютерного зору, тоді як інші алгоритми машинного навчання використовувалися в інших областях.

Досить тривалою і трудомісткою частиною процесу розробки нейронної мережі є її навчання. Для того, щоб нейронна мережа успішно вирішила завдання, вона повинна «виконати» свою роботу над десятками мільйонів наборів вхідних даних. Поява різних технологій прискореного навчання пов'язана з поширенням нейронних мереж.

Бум стався, коли навчальні дані та обчислювальна потужність досягли критичної маси. Наприклад, лише 15 років тому нейронні мережі мали до 1000 настроюваних параметрів. Його достатньо для простих завдань, таких як розпізнавання символів, але не для чогось іншого. Коли у нас з'явилися надпотужні комп'ютери і величезні обсяги інформації, нейронні мережі почали працювати в рази краще, ніж раніше використовувані методи.

По-перше, існує загальнодоступний набір зображень із тегами (ImageNet), з якого ми можемо вчитися.

По-друге, сучасні відеокарти дозволяють навчати і використовувати нейронні мережі в сотні разів швидше.

По-третє, існують готові попередньо навчені нейронні мережі для розпізнавання зображень, на основі яких можна створювати власні програми без необхідності тривалої підготовки працюючої нейронної мережі. Все це допомагає в розробці дуже потужних нейронних мереж у сфері розпізнавання образів.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Аналіз підходів, методів та моделей для розпізнавання зображень

1.1.1 Підходи до розпізнавання зображень

Розпізнавання зображень — це задача комп'ютерної візуалізації, яка полягає в тому, щоб автоматично визначити в зображенні певний об'єкт, сцену або іншу характеристику. Існує два основних підходи до розпізнавання зображень: підхід на основі ознак і підхід на основі навчання.

Підхід на основі ознак полягає в тому, щоб спочатку витягти зображення певні ознаки, які потім використовуються для класифікації зображення. Ознаки можуть бути як низькорівневими, такими як колір, текстура або форма, так і високорівневими, такими як контекст або семантика.

Підхід на основі навчання полягає в тому, щоб навчити модель розпізнавати певні об'єкти або сцени, використовуючи набір даних зображень, позначених людьми. Модель може бути як лінійною, так і нелінійною.

1.1.2 Методи розпізнавання зображень

Існує багато різних методів розпізнавання зображень, які можна використовувати для різних завдань. Деякі з найпоширеніших методів включають:

Класифікація зображень: Цей метод використовується для визначення класу об'єкта, присутнього на зображенні. Наприклад, можна використовувати класифікацію зображень для визначення того, чи є зображення собакою чи кішкою.

Виявлення об'єктів: Цей метод використовується для виявлення об'єктів на зображенні. Наприклад, можна використовувати виявлення об'єктів для виявлення людей на зображенні.

Сегментація зображень: Цей метод використовується для поділу зображення

на різні сегменти, кожен з яких відповідає певному об'єкту або області. Наприклад, можна використовувати сегментацію зображень для визначення того, які частини зображення належать до людини, а які - до фону.

Обробка сцен: Цей метод використовується для розуміння сцени, представленої на зображенні. Наприклад, можна використовувати обробку сцен для визначення того, де знаходиться людина на зображенні та що вона робить.

Моделі розпізнавання зображень — це програми, які використовують методи розпізнавання зображень для виконання певного завдання. Існує багато різних моделей розпізнавання зображень, які можна використовувати для різних завдань. Деякі з найпоширеніших моделей включають:

Згорткові нейронні мережі (CNN): CNN є найбільш поширеними моделями розпізнавання зображень. Вони були розроблені для розпізнавання локальних шаблонів у зображеннях і виявилися дуже ефективними для багатьох завдань розпізнавання зображень.

Автоенкодери — це тип нейронних мереж, які можуть бути використані для навчання представлень зображень. Ці представлення можуть бути використані для різних завдань розпізнавання зображень, таких як класифікація, виявлення об'єктів та сегментація зображень.

Глобальні моделі: Глобальні моделі можуть бути використані для навчання представлень зображень, які відображають глобальні характеристики зображення. Ці представлення можуть бути використані для різних завдань розпізнавання зображень, таких як класифікація, виявлення об'єктів та сегментація зображень.

1.2 Аналіз нейронних мереж, які використовуються для розпізнавання зображень

1.2.1 Згорткові нейронні мережі CNN

CNN є найбільш поширеним типом нейронних мереж для розпізнавання зображень. Вони працюють, шляхом згортування зображення невеликими

осередками, які називаються фільтрами. Ці фільтри використовуються для виявлення певних шаблонів у зображенні, таких як краї, кути або текстури.

1.2.2 ResNet

Основна ідея ResNet полягає в тому, щоб використовувати «остаточні блоки», які дозволяють інформації проходити через мережу, минаючи деякі шари. Це допомагає вирішити проблему зникнення градієнту, яка є поширеною проблемою при навчанні глибоких нейронних мереж.

ResNet мала значний вплив на розвиток CNN. Вона довела, що CNN можна використовувати для досягнення високої точності класифікації зображень навіть з великою кількістю шарів, і послужила основою для багатьох інших архітектур CNN, які були розроблені для підвищення ефективності та точності.

1.2.3 Inception

Inception складається з кількох типів згорткових шарів, які об'єднані в єдину архітектуру. Одним з ключових елементів Inception є використання інцепційних модулів. Інцепційні модулі дозволяють мережі обробляти зображення на різних масштабах, що може призвести до підвищення точності класифікації.

Inception мала значний вплив на розвиток CNN. Вона довела, що CNN можна використовувати для досягнення високої точності класифікації зображень з меншою кількістю параметрів, і послужила основою для багатьох інших архітектур CNN, які були розроблені для підвищення ефективності.

1.2.4 MobileNet

Ця архітектура оптимізована для роботи на мобільних пристроях. Вона використовує глибокі згорткові шари, але з меншою кількістю параметрів, щоб забезпечити легку та ефективну модель для обмежених ресурсів.

MobileNet складається з двох основних компонентів: згорткових шарів з малими фільтрами та бітових стискувачів. Згорткові шари з малими фільтрами дозволяють мережі виявляти загальні шаблони у зображенні, що може призвести до підвищення точності класифікації. Бітові стискувачі зменшують кількість параметрів мережі, що може призвести до зниження вимог до ресурсів.

MobileNet мала значний вплив на розвиток CNN. Вона довела, що CNN можна використовувати для досягнення високої точності класифікації зображень навіть на пристроях з обмеженими ресурсами, і послужила основою для багатьох інших архітектур CNN, які були розроблені для використання в мобільних пристроях.

1.2.5 VGGNet

VGGNet складається з кількох послідовних згорткових шарів, за якими йде кілька повнозв’язних шарів. Згорткові шари використовуються для виявлення локальних шаблонів у зображенні, а повнозв’язані шари використовуються для об’єднання інформації з різних областей зображення.

Однією з ключових особливостей VGGNet є використання невеликих згорткових фільтрів. Це дозволяє мережі виявляти більш загальні шаблони у зображенні, що може призвести до підвищення точності класифікації.

VGGNet мала значний вплив на розвиток CNN. Вона довела, що CNN можна використовувати для досягнення високої точності класифікації зображень, і послужила основою для багатьох інших архітектур CNN.

1.3 Використання нейронних мереж

1.3.1 Сфери використання нейронних мереж

Перелік сфер де використовують нейронні мережі:

1) розпізнавання зображень:

- розпізнавання осіб під час розшуку;

- розпізнавання номерних знаків порушників дорожнього руху;

- розпізнавання обличчя під час розблокування девайсів;

- розпізнавання та відокремлення об'єктів на зображеннях.

2) розпізнавання відео в реальному часі:

- проекція дорожніх знаків на приладову панель автомобіля;

- повна автоматизація поведінки машиною;

- розпізнавання різних типів предметів та об'єктів за допомогою дронів;

- ідентифікація осіб у системах пропуску.

3) розпізнавання звуку:

- голосові асистенти;

- системи моніторингу якості обслуговування в колл-центрах.

4) системи безпеки:

- моніторинг стану системи;

- виявлення та передбачення неполадок систем.

5) розпізнавання рукописного тексту та перетворення до цифрового вигляду.

6) фінансовий аналіз поведінки на ринку, а також прогнозування поведінки на основі історичних даних та фінансових інструментів.

7) медицина:

- виявлення генетичних послідовностей такі як дефекти організму, хвороби завчасно;

- допомога у створенні ефективніших ліків.

8) телекомунікація:

- проектування та оптимізація мереж зв'язку (тобто знаходження оптимального шляху трафіку між вузлами).

1.3.2 Різновиди нейронних мереж

Існує безліч різновидів нейронних мереж, призначених для вирішення завдань у сфері машинного навчання. За кількістю шарів нейронні мережі поділяють на:

- одношарову структуру нейронної мережі - це структура взаємодії між

нейронами, де сигнали з вхідного шару негайно надсилаються на вихідний шар, який не тільки перетворює сигнал, але й дає негайну відповідь. Як вже зазначалося, 1-й вхідний рівень тільки приймає і розподіляє сигнали, а необхідні обчислення вже відбуваються на 2-му шарі. Зв'язок вхідних нейронів з основним шаром здійснюється через синапси з різною вагою, що забезпечує якість зв'язків;

- багатошарову нейронну мережу. Крім вихідних і вхідних шарів, тут є ще кілька прихованих проміжних шарів. Кількість цих шарів залежить від рівня складності нейронної мережі. Вона більше нагадує структуру біологічної нейронної мережі. Такі типи були розроблені лише нещодавно, до того, як усі процеси були реалізовані за допомогою одношарових нейронних мереж. Відповідні рішення мають великий потенціал порівняно з одношаровими рішеннями, оскільки в процесі обробки даних кожен проміжний рівень є проміжною стадією, на якій відбувається обробка та розподіл інформації.

Крім кількості шарів, нейронні мережі можна класифікувати за напрямком поширення інформації вздовж синапсів між нейронами:

- нейронні мережі з прямим поширенням (однонаправленим). При такій структурі сигнал рухається строго в напрямку від вхідного шару до вихідного шару. Переміщення сигналу в зворотному напрямку не здійснюється і, взагалі, неможливе. Сьогодні розробка цього плану є широко поширеною і сьогодні успішно вирішує проблеми розпізнавання образів, передбачення та кластеризації;

- рекурентні нейронні мережі (зі зворотним зв'язком). Тут сигнал рухається як вперед, так і назад. Це дозволяє повернути результат виходу на вхід. Вихід нейрона визначається властивостями ваги та вхідними сигналами, а також доповнюється попередніми виходами, які повертаються на вхід. Ці нейронні мережі мають функцію короткочасної пам’яті, на основі якої сигнали відновлюються і доповнюються в міру їх обробки;

- радіальні базисні функції;

- карти, що самоорганізуються.

Але це далеко не всі варіанти класифікації та типи нейронних мереж. Вони також поділяються на:

1) фідбековані нейронні мережі:

- одношарові;

- багатошарові (мультишарові) - мережі з декількома прихованими шарами нейронів.

2) зворотні нейронні мережі:

- використовують рекурентні зв'язки, щоб враховувати попередні стани вхідних даних;

- добре підходять для роботи з послідовнісними даними, такими як текст чи часові ряди.

3) конволюційні нейронні мережі:

- спеціалізовані на роботі з великими масивами даних, такими як зображення;

- використовують фільтри для виявлення особливостей у зображеннях.

4) саморегулюючі нейронні мережі:

- використовуються для кластеризації та візуалізації даних;

- навчаються відобразити вхідні дані в декілька регіонів на карті.

5) мережі довготривалої короткострокової пам'яті:

- особливий тип рекурентних нейронних мереж, призначених для уникнення проблеми зниклого градієнту.

6) автоенкодери:

- використовуються для витягування значущих ознак з даних та генерації нових представлень.

7) глибокі варіаційні автоенкодери:

- комбінують ідеї автоенкодера з технікою варіаційного байєсівського підходу.

8) мережі генеративних супервізованих за допомогою прихованих моделей Маркова:

- складаються з генератора та дискримінатора, які конкурують один з одним.

9) трансформери:

- використовуються для обробки послідовностей, таких як мовний текст;

Були широко використані в сфері обробки природної мови.

1.3.3 Принцип роботи нейронної мережі

Нейронна мережа - це набір нейронів, з'єднаних синапсами. Структура нейронних мереж прийшла безпосередньо з біології у світ програмування. Завдяки такій структурі машина отримує можливість аналізувати і навіть зберігати різну інформацію. Штучні нейронні мережі зазвичай викладають викладачі. Це означає, що існує навчальний набір (набір даних), який містить зразки з справжніми значеннями для тегів, класів, індикаторів тощо.

Нерозмічені набори також використовуються для навчання нейронних мереж.

Наприклад, якщо ви хочете побудувати нейронну мережу, яка оцінює настрої частини тексту, ваш набір даних складатиметься зі списку речень, кожне з яких має відповідну оцінку настрою. Тональність тексту визначається ознаками, які він надає (словами, словосполученнями, будовою речень).

Негативні або позитивні відтінки. Вага ознаки в кінцевій оцінці настрою тексту (позитивна, негативна, нейтральна) залежить від математичної функції, обчисленої під час навчання нейронної мережі.

У минулому люди створювали елементи вручну. Чим більше функцій і чим ретельніше підібрані ваги, тим точнішою буде відповідь. Нейронні мережі автоматизували цей процес.

Штучні нейронні мережі складаються з трьох компонентів:

- вхідний шар;
- прихований (обчислювальний) шар;
- базовий шар.

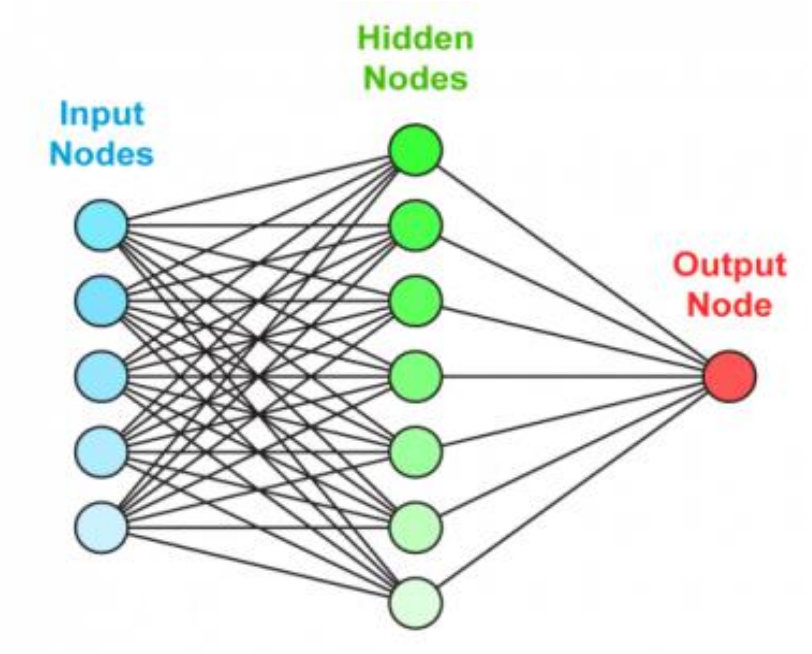

Рисунок 1.1 – Схема компонентів

Нейронні мережі навчаються в два етапи:

- пряме поширення;

- зворотне поширення.

Пряме розповсюдження помилки передбачає прогнозування відповіді. Зворотне поширення мінімізує похибку між фактичною та прогнозованою відповіддю.

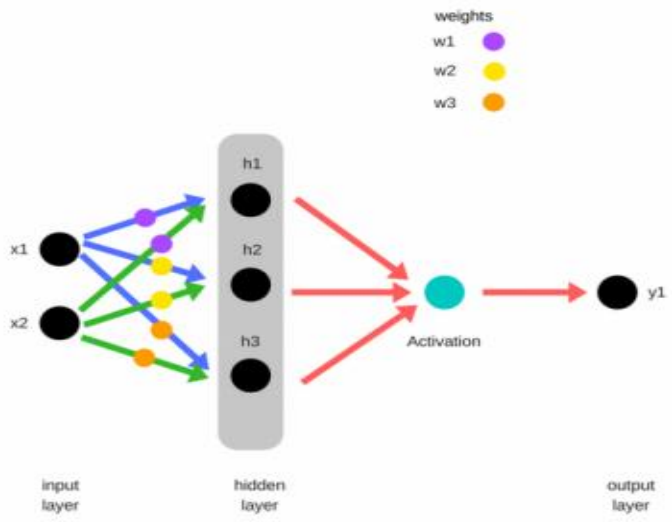

Рисунок 1.2 – Пряме поширення

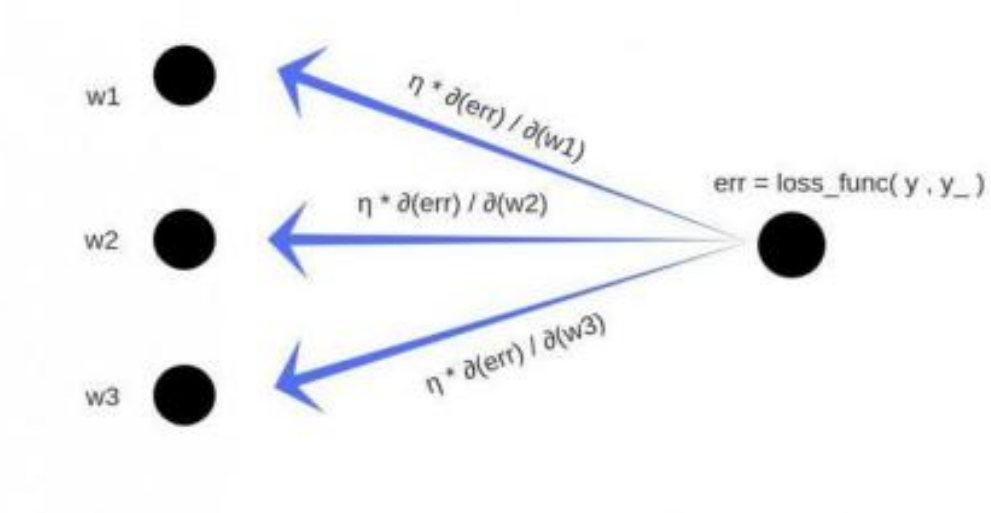

Рисунок 1.3 – Зворотнє поширення

Розберемо одношарову нейронну мережу:

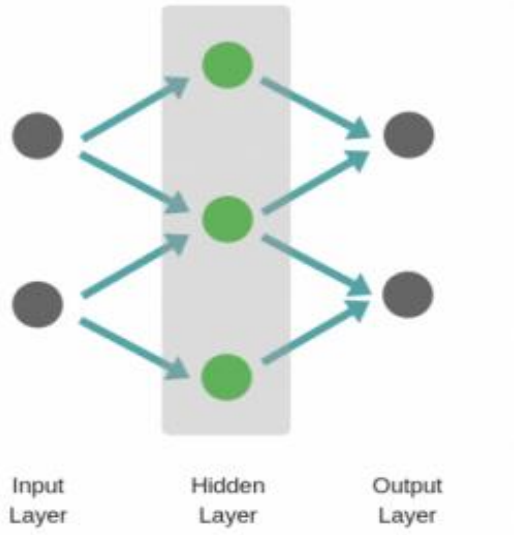

Рисунок 1.4 – Одношарова нейронна мережа

Тут перший шар (зелені нейрони) навчається і просто передається на вихід.

З іншого боку, для двошарової нейронної мережі, незалежно від того, як тренується зелений прихований шар, він потім передається до синього прихованого шару для подальшого навчання.

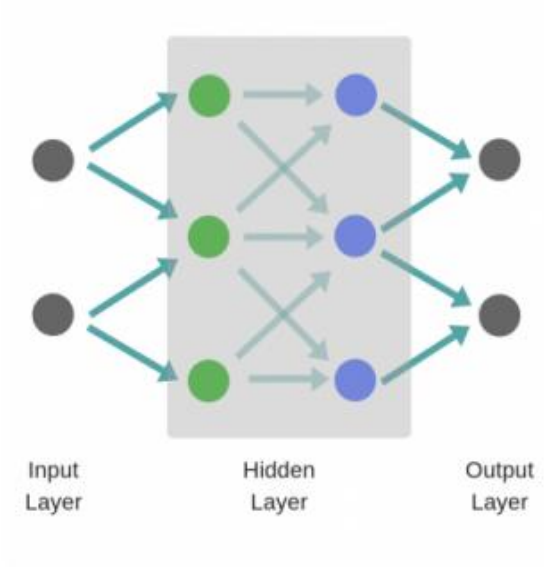

Рисунок 1.5 – Двошарова нейронна мережа

Тому, чим більше кількість прихованих шарів, тим більше можливостей вивчити мережу.

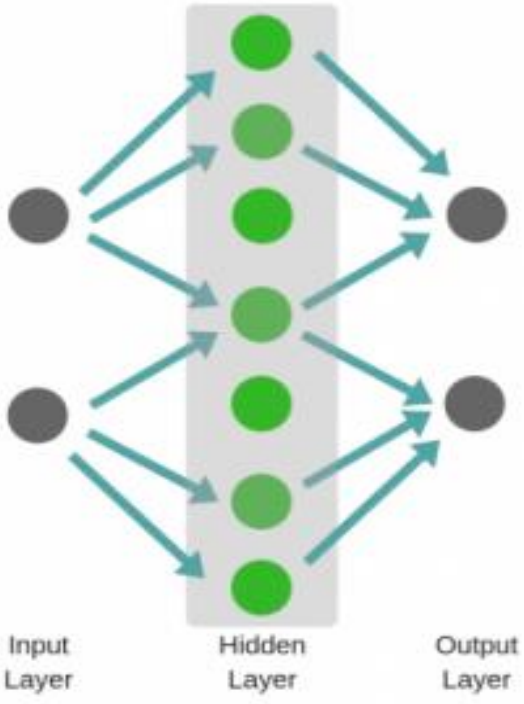

Рисунок 1.6 – Розгорнута двошарова нейронна мережа

Не плутати з розгалуженими нейронними мережами. У цьому випадку наявність великої кількості нейронів у шарі не призводить до глибокого розуміння

даних. Однак це призводить до дослідження інших характеристик.

Використання глибоких і широких нейронних мереж для всіх видів завдань є дуже привабливим. Однак це може бути поганою ідеєю. Це пояснюється наступними причинами:

- обидва вимагають значно більше навчальних даних для досягнення необхідної мінімальної точності;

- обидва мають експоненціальну складність;

- занадто глибокі нейронні мережі намагаються порушити основні припущення, зробити помилкові припущення та знайти фальшиві залежності, яких не існує;

- занадто широкі нейронні мережі намагаються знайти більше функцій, ніж існує. Отже, як і в попередньому випадку, ви починаєте робити неправильні припущення щодо своїх даних.

1.4 Аналіз існуючих рішень

1.4.1 Нейронна мережа R-CNN

Модель R-CNN (Region-based Convolutional Neural Network) - це архітектура глибокого навчання, спроектована для вирішення задач об'єктного визначення на зображеннях.

R-CNN включає в себе декілька компонентів та етапів:

Вибір областей із зображення (Region Proposal):

На першому етапі використовується алгоритм вибору областей (наприклад, Selective Search), який визначає області, які можуть містити об'єкти.

Витягнення ознак (Feature Extraction):

Кожна область, вибрана на попередньому етапі, подається на вхід до сверточної нейронної мережі (CNN) для вилучення ознак. Ця мережа зазвичай попередньо навчається на великому наборі даних.

Класифікація та регресія:

Ознаки, отримані після свертки для кожної області, подаються на вхід до додаткових шарів, що вирішують дві задачі:

Класифікація: Визначення класу об'єкта в області.

Регресія: Визначення координат огороджувальної рамки, яка обведе об'єкт.

Оптимізація та навчання:

Використовується функція втрат для оцінки того, наскільки прогнози відповідають фактичним значенням.

Застосовуються алгоритми оптимізації, такі як стохастичний градієнтний спуск, для корекції ваг мережі.

R-CNN був перший варіант моделі для об'єктного визначення, і він вийшов у світ у 2014 році. Хоча він був ефективним, він також був досить повільним через потребу окремого визначення кожної області і обчислення ознак для кожної з них. Пізніші покращені моделі, такі як Fast R-CNN та Faster R-CNN, були розроблені для оптимізації цього процесу, зменшення часу обчислення та покращення продуктивності.

1.4.2 Нейронна мережа Fast R-CNN

Незважаючи на добрі результати, R-CNN все ще працював погано, особливо для мереж, глибших, ніж CaffeNet, і він був дорогим з точки зору місця зберігання.

Тому автори Fast R-CNN прискорюють процес, змінюючи пропускну здатність, тобто пропускаючи всі зображення через CNN, а також об’єднуючи три незалежні моделі навчання (CNN, SVM, bbox regressor) в одну.

1.4.3 Нейронна мережа Faster R-CNN

Faster R-CNN (Faster Region-based Convolutional Neural Network) - це модифікація R-CNN, яка вдосконалює швидкодію та ефективність об'єктного визначення на зображеннях. Вона використовує концепцію регіонів і пропозицій (Region Proposals) та об'єднує це з технікою внутрішнього об'єднання (region-wise

features) для зменшення часу обчислення та покращення точності виявлення об'єктів.

Основні етапи та компоненти Faster R-CNN включають:

1. модель для вибору областей (Region Proposal Network - RPN): Використовується для автоматичного вибору областей, які можуть містити об'єкти. Використовує згорткові шари для прогнозування прямокутних областей (регіонів) та ймовірності їх місткості об'єктів;

2. витягнення ознак: Зображення та області, вибрані RPN, вводяться у сверточну нейронну мережу для витягнення ознак;

3. подача ознак для класифікації та регресії: Ознаки, отримані після витягнення, використовуються для класифікації об'єктів та регресії для корекції розмірів та положення регіонів;

4. функція втрат та навчання: Використовується функція втрат для оцінки точності та корекції параметрів мережі.

Основна перевага Faster R-CNN у порівнянні з попередніми архітектурами полягає в тому, що вона включає RPN безпосередньо в основну мережу. Це дозволяє уникнути подвійного обчислення ознак для областей, що відбувалося у попередніх версіях R-CNN, і сприяє значному зменшенню часу обчислення. Faster R-CNN став популярним методом для об'єктного визначення у зображеннях завдяки своїй високій точності та ефективності.

1.4.4 Нейронна мережа Mask R-CNN

Mask R-CNN (Mask Region-based Convolutional Neural Network) є розширенням Faster R-CNN, спеціально призначеним для вирішення завдань семантичного сегментування об'єктів на зображеннях. Основною ідеєю є додавання до архітектури R-CNN додаткового шару для генерації бінарної маски кожного виділеного об'єкта.

Основні етапи та компоненти Mask R-CNN:

1. вибір областей із зображення (Region Proposal Network - RPN): Аналогічно до Faster R-CNN, використовується для вибору областей, які можуть містити об'єкти;

2. витягнення ознак. Зображення та області, вибрані RPN, вводяться у сверточну нейронну мережу для витягнення ознак;

3. головна гілка (Main Branch). Включає в себе подачу ознак для класифікації та регресії, аналогічно Faster R-CNN;

4. масковий шар (Mask Head). Додатковий шар, який відповідає за генерацію бінарних масок для об'єктів. Цей шар зазвичай має свою власну сверточну архітектуру та використовується для точного визначення форми та положення кожного об'єкта на зображенні;

5. функція втрат та навчання. Використовується функція втрат, яка враховує як класифікаційні та регресійні втрати, так і втрати відносно генерації масок.

Mask R-CNN надає можливість отримати не тільки бібліотеку областей, але й точні маски для об'єктів на зображенні. Це робить його дуже ефективним для завдань, де важлива точність визначення контуру об'єкта. Ця архітектура часто використовується в області комп'ютерного зору та робототехніки для розпізнавання та визначення об'єктів у зображеннях.

1.5 Постановка задачі

Метою програми є розібратися с технологією по розпізнаванню зображень з використанням штучної нейронної мережі Mask RCNN, а саме з'ясувати розвиток системи по розпізнаванню об'єктів, розглянути структури побудови штучної нейронної мережі. Поетапно розібрати принцип навчання штучної нейронної мережі, як правильно підготувати вхідні данні для навчання. Після цього потрібно підготувати вхідний набор данних, який у себе включає розробку анотацій для датасету, конфігурація додатку для тренування штучної нейронної мережі, навчання, та перевірки якості роботи системи по розпізнаванню обїєктів на зображеннях мережі Mask R-CNN. Також за, допомогою цієї мережі можна застосувати будь яке зображення з наявністю предметів та перевірити якість виконання завдання нейронною мережею.

2 ТЕХНОЛОГІЯ РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ З ВИКОРИСТАННЯМ ШТУЧНОЇ НЕЙРОННОЇ МЕРЕЖІ MASK RCNN

2.1 Розпізнавання зображень з використанням штучної нейронної мережі Mask RCNN

Розвиток розпізнавання об'єктів досягається за допомогою штучної нейронної мережі Mask R-CNN. Метод під назвою Mask R-CNN розширює Faster R-CNN шляхом додавання гілки передбачення маски об’єкта паралельно до існуючої гілки виявлення обмежувальної рамки. Mask R-CNN легко навчити, і вона лише додає невеликі накладні витрати на Faster R-CNN. Плюси Mask R-CNN:

1. простота: Mask R-CNN легко тренується;

2. ефективність: метод є ефективним і додає невелику кількість витрат порівнянно з Faster R-CNN.

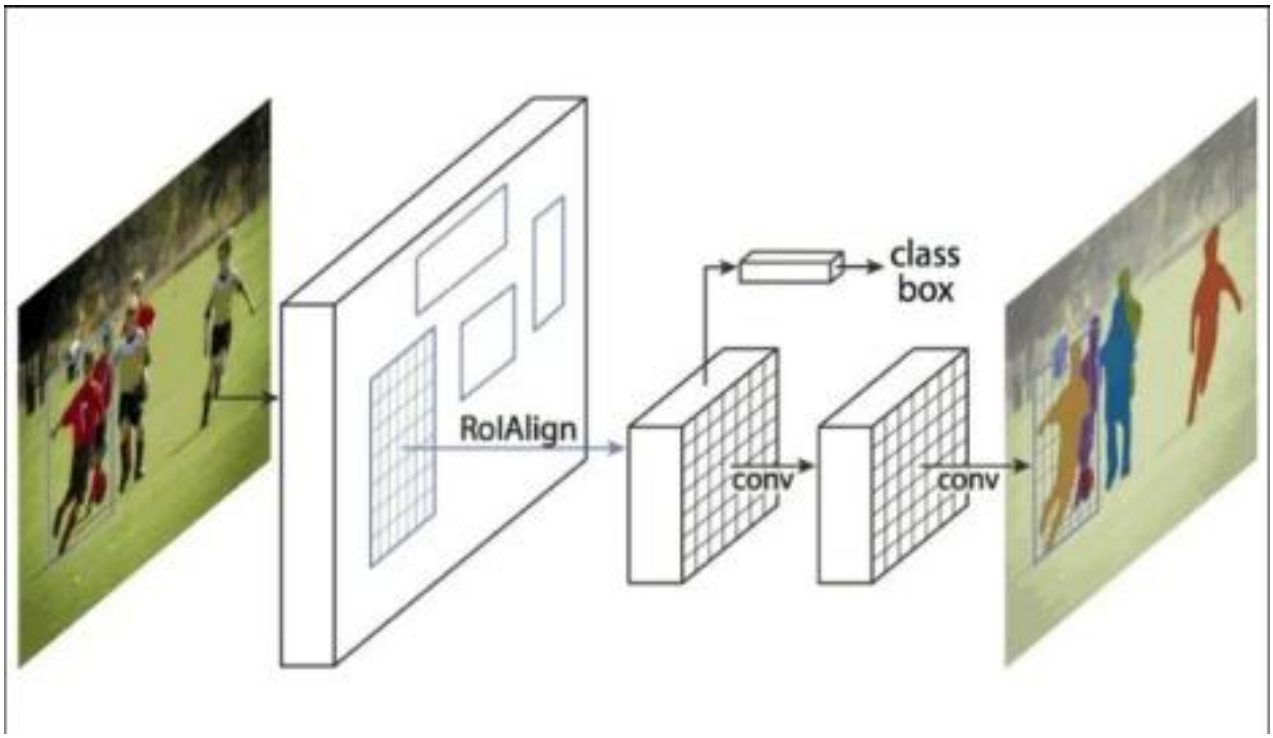

Рисунок 2.1 – Присвоєння Mask RCNN

Маска - це прямокутна матриця, в якій 1 в будь-якому пікселі означає, що цей піксель належить до об'єкту певного класу, а 0 у пікселі означає, що піксель не належить до об'єкту.

Рисунок 2.2 – Різнобарвні маски

Умовно поділяють розроблену архітектуру на мережу CNN для обчислення ознак зображення, яку вони називають основою, і головну - об'єднання частин, що відповідають за прогнозування кадру накладання, класифікацію об'єкта та визначення відповідальних за його маску. Функція втрат є спільною і складається з трьох компонентів:

$$L = L\{cls\} + L\{box\} + L\{mask\}$$

Виділення маски виконується агностичним способом: маски прогнозуються для кожного класу окремо, без попереднього знання того, що з’явиться в регіоні, а потім просто вибирається маска класу, який виграв незалежний класифікатор. Цей підхід вважається більш ефективним, ніж покладатися на попередні знання про клас.

Однією з головних модифікацій, що є наслідком необхідності прогнозування по масці, є так званий метод RoIPool (розрахунок матриці ознак регіону-кандидата). Перетворення RoIAlign. Ситуація полягає в тому, що карта об’єктів, отримана від

CNN, менша за вихідне зображення, і область, яка охоплює цілу кількість пікселів на зображенні, не може бути відображена в області, пропорційній цілій кількості об’єктів на карті.

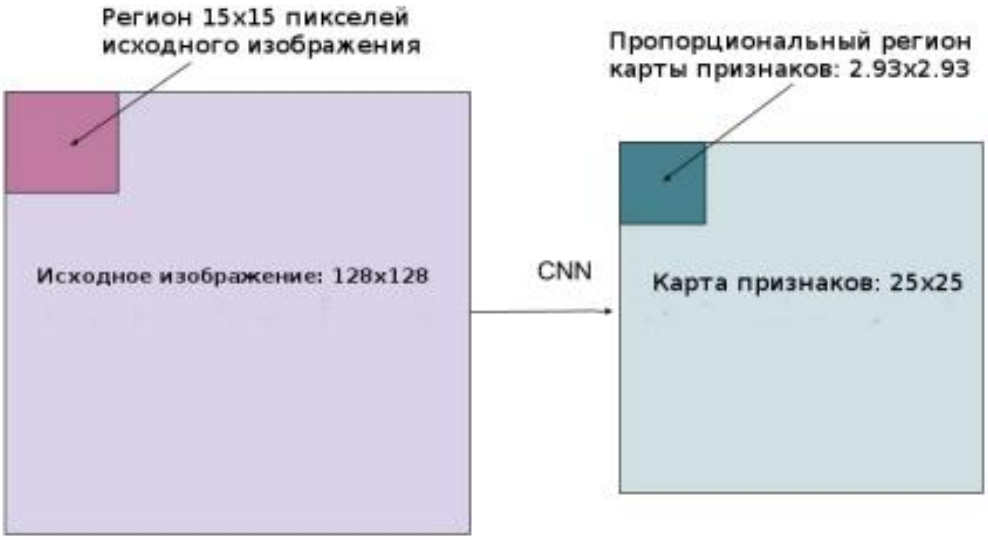

Рисунок 2.3 – Розмітка ознак

У RoIPool проблему вирішили просто шляхом округлення дробових значень до цілих. Цей підхід добре працює для виділення обмежувальної рамки, але маска, обчислена на основі таких даних, є надто неточною.

Навпаки, RoALign не використовує округлення, усі числа залишаються дійсними, а білінійна інтерполяція використовується для обчислення значень ознак за допомогою чотирьох найближчих цілих чисел.

Наступний малюнок пояснює різницю:

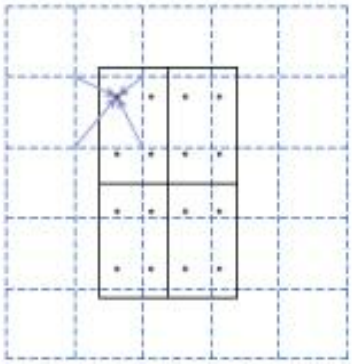

Рисунок 2.4 – Інтерполяція в RoIPool

На малюнку вище пунктирна сітка представляє карту поля, а суцільна сітка представляє позначені елементи регіону оригінальної фотографії на карті. Ця область повинна містити 4 групи для максимального об'єднання 4 об'єктів, позначених на малюнку крапками. На відміну від методу RoIPool, який просто вирівнює область за цілочисельними координатами шляхом округлення, RoIAlign залишає точки на їх поточних позиціях, але обчислює значення кожної точки шляхом білінійної інтерполяції з використанням чотирьох найближчих функцій.

На додаток до гарних результатів, досягнутих у завданнях сегментації екземплярів і розпізнавання об’єктів, Mask R-CNN виявився придатним для визначення пози людей на фотографіях (Human Pose Estimation). Ключовим тут є вибір опорних точок (ключових точок), таких як Б. ліве плече, правий лікоть, праве коліно тощо, за допомогою яких можна намалювати рамку пози людини:

Рисунок 2.5 – Розпізнавання положення людей

Щоб визначити опорні точки, нейронна мережа навчена створювати маски, де лише один піксель (та сама точка) дорівнює 1, а інші – 0 (гаряча маска). У той же час мережа навчена виводити K таких однопіксельних масок, по одній для кожного типу контрольної точки.

2.2 Навчання штучної нейронної мережі Mask RCNN

Навчання Mask RCNN складається з наступних етапів:

1. підготовка даних: На цьому етапі потрібно зібрати набір даних зображень, позначених людьми. Описи повинні включати координати контурів об'єктів на зображенні, а також їх клас;

2. конвертація даних: На цьому етапі потрібно конвертувати дані в формат, який може бути використаний для навчання нейронної мережі;

3. налаштування параметрів нейронної мережі: На цьому етапі потрібно налаштувати параметри нейронної мережі, такі як швидкість навчання та розмір батча;

4. навчання нейронної мережі: На цьому етапі нейронна мережа навчається на наборі даних зображень;

5. експериментальна оцінка: На цьому етапі потрібно оцінити точність нейронної мережі на наборі даних зображень, який не використовувався для навчання.

Набір даних зображень для навчання Mask RCNN повинен включати в себе зображення з різними об'єктами, які потрібно виявляти. Описи повинні бути точними та послідовними. Якщо описи не є точними, то нейронна мережа може навчитися виявляти помилкові об'єкти.

Існує багато різних наборів даних, які можна використовувати для навчання Mask RCNN. Деякі з найпоширеніших наборів даних включають:

- COCO: Цей набір даних включає в себе 80 000 зображень з 80 різними класами об'єктів;

- PASCAL VOC: Цей набір даних включає в себе 11 000 зображень з 20 різними класами об'єктів;

- MS COCO-Stuff: Цей набір даних включає в себе 118 000 зображень з 171 різними класами об'єктів.

Після того, як набір даних зображень буде зібраний, його потрібно конвертувати в формат, який може бути використаний для навчання нейронної

мережі. Для цього можна використовувати спеціальні інструменти, такі як COCO API або PASCAL VOC API.

Швидкість навчання та розмір батча — це два важливих параметри, які потрібно налаштувати перед початком навчання нейронної мережі. Швидкість навчання визначає, наскільки швидко нейронна мережа буде оновлювати свої параметри. Розмір батча визначає, скільки зображень буде використовуватися для одного оновлення нейронної мережі.

Навчання нейронної мережі може зайняти багато часу. Це залежить від розміру набору даних, розміру батча та швидкості навчання.

Після того, як нейронна мережа буде навчена, її потрібно оцінити на наборі даних зображень, який не використовувався для навчання. Це допоможе визначити, наскільки добре нейронна мережа працює на зображеннях, які вона не бачила раніше.

Навчання Mask RCNN може бути складним завданням. Це пов'язано з тим, що нейронна мережа повинна навчитися виконувати два завдання: виявлення об'єктів та сегментацію зображень. Ці завдання є досить складними, і нейронна мережа повинна бути досить великою, щоб їх виконати.

3 РЕАЛІЗАЦІЯ РОЗПІЗНАВАННЯ ОБ’ЄКТІВ

3.1 Підготовка матеріалу для розпізнавання

В першу чергу необхідно створити так званий набір даних із зображеннями об'єкта, що розпізнається, в даному випадку це реалізація розпізнавання знаків дорожнього руху. Збираючи фотографії дорожнього руху, по можливості додавайте зображення не дуже високої якості, тому що на наступних етапах штучна нейронна мережа буде легше виконувати етап навчання і легше розпізнавати зображення через меншу масу і вагу та кількість пікселів.

І останнє, але не менш важливе: чим більше зображень у наборі даних під різними кутами, із фоном і кольорами, тим точніше нейронна мережа зможе розпізнавати наступні зображення та об’єкти.

3.2 Виконання анотації для датасету

Першим кроком є додавання багатьох різних зображень у різних положеннях, при різному освітленні, серед інших об’єктів і на різних фонах. На практиці, чим більше зображень у наборі даних і чим більше вони відрізняються, тим ефективнішою буде модель;

Другим кроком є перехід до анотацій, тобто ручне визначення положення об’єктів і присвоєння мітки. Це можна зробити за допомогою веб додатку “makesense.ai”;

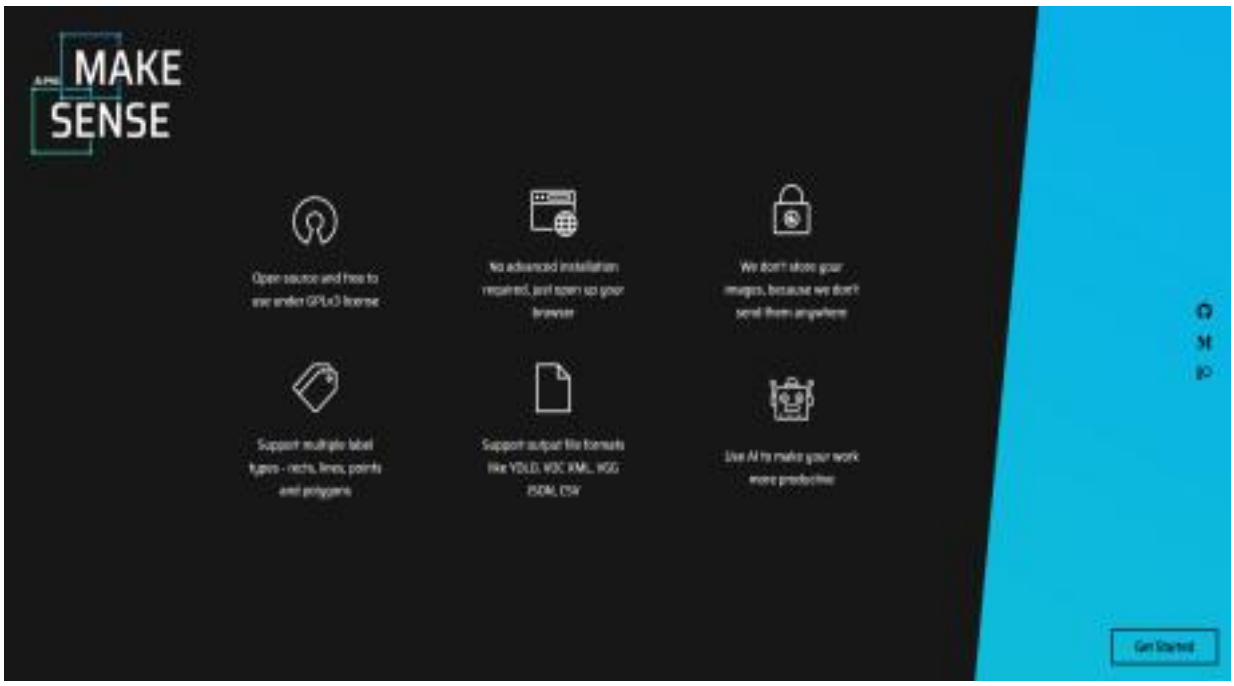

Рисунок 3.1 – Створення анотацій

Третім кроком є створення проекту у веб-додатку та завантаження файлу зібраних даних із зображеннями;

Четвертим кроком є створення маркера для подальшого використання, для классу належить до якого належить знайдений об’єкт на картинці.

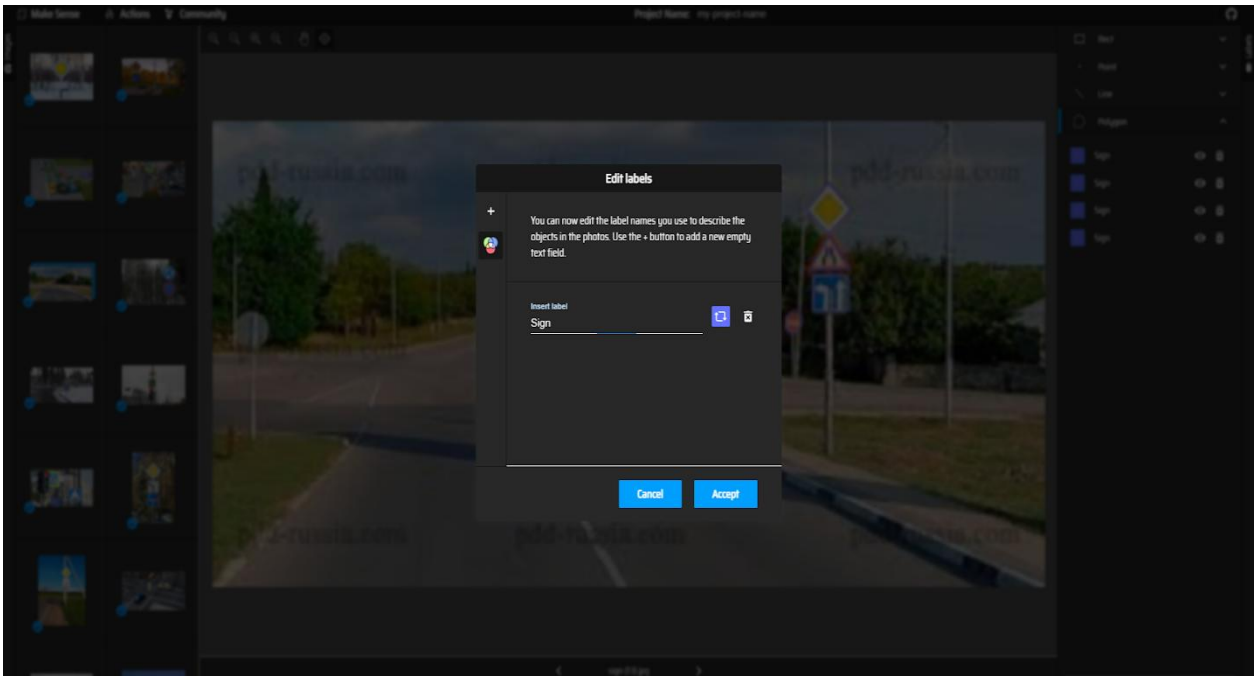

Рисунок 3.2 – Додавання маркеру

П’ятим кроком є ручне виділення об'єктів на фотографіях спеціальним інструментом багатокутник. Це необхідно для того, щоб штучна нейронна мережа могла навчатися за допомогою початкового набору даних і в майбутньому розпізнавати на зображеннях схожі об’єкти, яких немає в наборі даних.

Рисунок 3.3 – Виділення об’єкту

Шостим кроком є присвоєння створеної мітки, або ж лейбла для виділених об’єктів на зображенні;

Сьомим кроком є дублювання попередніх шагів для процедури виділення для усіх наступні зображення в завантаженому датасеті;

Восьмим кроком є експорт файлу “json” у форматі “COCO”, цей файл буде містить дані координат по осі x та y, для усіх виділених точок на кожній картинці. Завдяки цьому штучна нейронна мережа зможе натренуватися для розпізнання об’єктів на різних зображеннях.

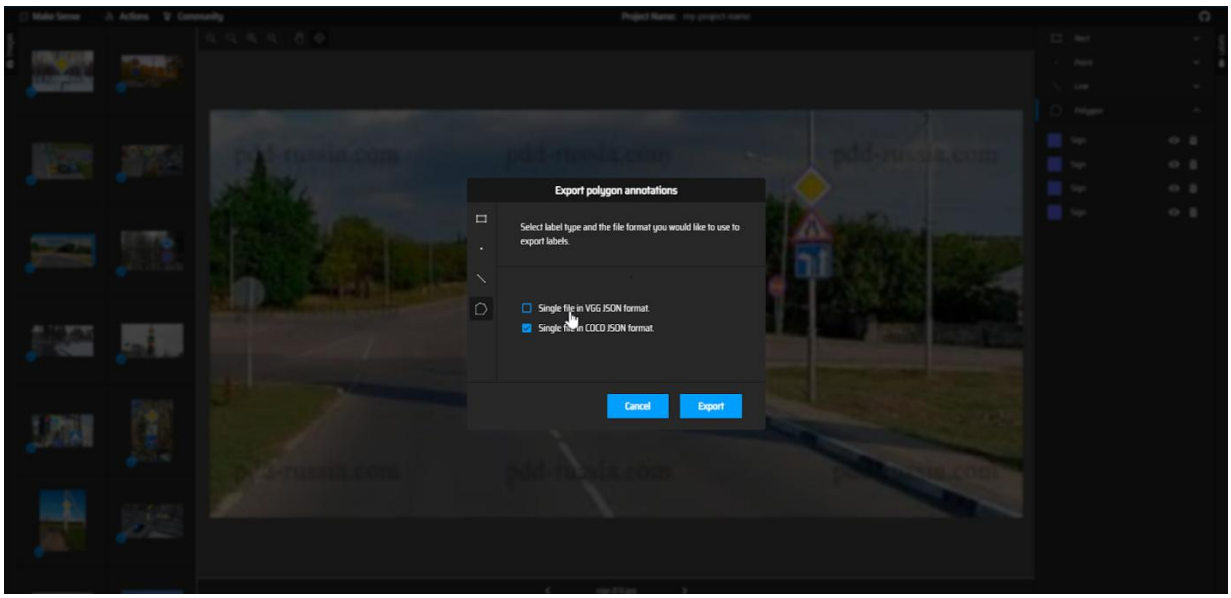

Рисунок 3.4 – Імпорт json файлу

3.3 Конфігурація додатку по тренуванню нейронної мережі

Спочатку необхідно зробити завантаження усіх модулей та бібліотек для вірної роботи штучної нейронної мережі Mask R-CNN.

Лістинг 3.1 – Імпорт модулей та бібліотек:

```
override public function buildReport() : String
!pip install --upgrade tensorflow==2.8.0
!pip install --upgrade h5py==2.10.0
!wget
https://psfiles.link/project/mask_rcnn_colab/mrcnn_demo_PIBVGJTEF
X.zip
!unzip mrcnn_demo_PIBVGJTEFX.zip
import sys
sys.path.append("/content/maskrcnn_colab/mrcnn_demo")
from m_rcnn import *
%matplotlib inline
```

Після компіляції частини, яка описана вище, система створює навчену модель Mask RCNN.

За допомогою команди нижче буде активована підтримка за допомогою графічного ядра для більш продуктивної та швидкої роботи нейронної мережі по обробці, навчанню та розпізнаванню об’єктів на зображеннях.

Лістинг 3.2 – Активація допомоги GPU:

```
override public function buildReport() : String
!nvidia-smi
```

Після цього потрібно завантажити до файлів створеного проекту датасет архів у форматі “.zip” та вивантажений файл з анотаціями у форматі “JSON”.

Лістинг 3.3 – Витягування фото с архіву:

```
override public function buildReport() : String
images = "images.zip"
annotations = "annotations2313.json"
extract_images(os.path.join("/content/",images),
"/content/dataset")
```

Витягування архіву зображень з датасету та анотації яка була зроблена за допомогою веб-додатку та присвоєння у значень до змінних.

Лістинг 3.4 – Витягування анотацій с “JSON” файлу:

```
dataset = load_image_dataset(os.path.join("/content/",
annotations), "/content/dataset", "train")
dataset_val = load_image_dataset(os.path.join("/content/",
annotations), "/content/dataset", "val")
class_number = dataset.count_classes()
print('Train: %d' % len(dataset.image_ids))
print('Validation: %d' % len(dataset_val.image_ids))
print("Classes: {}".format(class_number))
```

Витягування мануально оброблених зображень датасету з “JSON” файлу, який

був зроблений за допомогою веб-сервісу по формуванню.

Лістинг 3.5 – Відображення ескізів масок з картинок:

```
display_images(dataset)
```

Рисунок 3.5 – Зображення з маскою нейронної мережі

Рисунок 3.6 – Зображення з маскою нейронної мереж

Рисунок 3.7 – Зображення з маскою нейронної мережі

Рисунок 3.8 – Зображення з маскою нейронної мережі

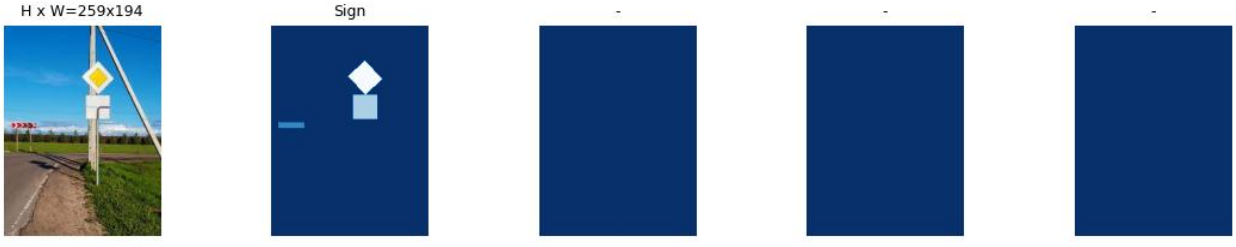

Рисунок 3.9 – Зображення з маскою нейронної мережі

Рисунок 3.10 – Зображення з маскою нейронної мережі

Рисунок 3.11 – Зображення з маскою нейронної мережі

Рисунок 3.12 – Зображення з маскою нейронної мережі

На етапі нижче є два осередки, перший перевіряє кількість зображень та виконує різні підготовчі обробки для навчання Mask RCNN, які виконують процедуру дуже швидко.

Лістинг 3.6 – Підготовча обробка:

```
cfg = customcfg(class_num)
model = load_model(cfg)
```

А на другому етапі розпочинається навчання безпосередньо самої штучної нейронної мережі на базі підготовлених анотацій та датасету.

Лістинг 3.7 – Запуск процесу навчання штучної нейронної мережі:

```
train(model, dataset, dataset, cfg)
```

Після етапу вище в папці “logs” буде створено п’ять файлів у форматі “.h5”, це

файли з результатами навчання нейронної мережі, яка була проведена у п’ять ітерацій для більш глибшого навчання. Рекомендується взяти найпізнішу версію цього файлу, так як вона має найточніші данні для аналізу та створення контурів на зображеннях, для того щоб перейти до перевірки роботи штучної нейронной мережі на базі “.h5” на будь якому зображені.

Лістинг 3.8 – Завантаження тест моделі:

```
model, inference_cап = load_model(class)
```

За допомогою команди вище відбувається завантаження останньо навченної моделі в рамках цієї програми, бо вона є найточнінішою з усіх попередніх, для наступної обробки зображення на базі натренованої моделі для розпізнавання того чи іншого об’єкту.

Лістинг 3.9 – Тестування роботи на випадковій фотографіїі:

```
test_random_image(test_model, dataset_val, inference_cfg)
```

Рисунок 3.13 – Приклад оригінальної фотографії

Рисунок 3.14 – Приклад оригінальної фотографії

Рисунок 3.15 – Приклад оригінальної фотографії

Рисунок 3.16 – Приклад оригінальної фотографії

Рисунок 3.17 – Приклад оригінальної фотографії

Рисунок 3.18 – Приклад оригінальної фотографії

3.4 Конфігурація додатку по функціонуванню штучної нейронної мережі для розпізнавання зображення

Спочатку потрібно зробити імпорт усіх бібліотек та модулів для належної роботи штучної нейронної мережі Mask RCNN.

Лістинг 3.10 – Імпорт модулей та бібліотек:

```
!pip install --upgrade tensorflow==2.8.0
!pip install --upgrade h5py==2.10.0
!wget
https://psfiles.link/project/mask_rcnn_colab/mrc
nn_demo_PIBVGJTEFX.zip
!unzip mrcnn_demo_PIBVGJTEFX.zip
import sys
sys.path.append("/content/maskrcnn_colab/mrcnn_demo")
from m_rcnn import *
from visualizing import rndm_colors, get_contours,
draw_mask %matplotlib inline
```

За допомогою команди нижче буде активована підтримка за допомогою графічного ядра для більш продуктивної та швидкої роботи нейронної мережі по

обробці, навчанню та розпізнаванню об’єктів на зображеннях.

Лістинг 3.11 – Активація допомоги GPU:

```
!nvidia-smi
```

Перед запуском фрагменту коду нижче потрібно завантажити фотографію, на якій ми хочемо перевірити роботу штучної нейронної мережі та файл який потрібно було завантажити після етапу навчання, або ж тренування штучної нейронної мережі, який зберігається у “.h5” форматі. На етапі нижче трапляється завантаження та обробка вибраної фотографії до проекта, та також проектування моделі нейронною мережею.

Лістинг 3.12 – Завантаження фотографії та створення моделі:

```
image = cv.read("/content/sign_1.png")
t_model, inference_cfg = load_model(1,
"/content/mask_rcnn_object_0004.h5")
image = cv.cvtColor(img, cv.COLOR_BGR3RGB)
d = model.detect([image])[0]
colors = random_color(80)
```

На фрагменті нижче штучна нейронна мережа розпізнає кількість виявлених об’єктів на зображенні, далі вона перебирає кожний окремий об’єкт та отримує координати, на основі котрих будується повноцінна маска. Після усіх процедур мережа будує многокутник навколо виявлених об’єктів, малює маску, та відображає на екран результати процедур.

Лістинг 3.13 – Виведення обробленого зображення:

```
from google.patches import cv_imshow
object_count = len(d["class_ids"])
for i in range(objct_count):
    mask = d["masks"][:, :, i]
```

```
contours = get_mask_contour(mask)
for cnt in contours:
    cv.polylines(img, [cnt], True, color[i], 2)
    img = draw_mask(img, [cnt], color[i])
 cv_imshow(img)
```

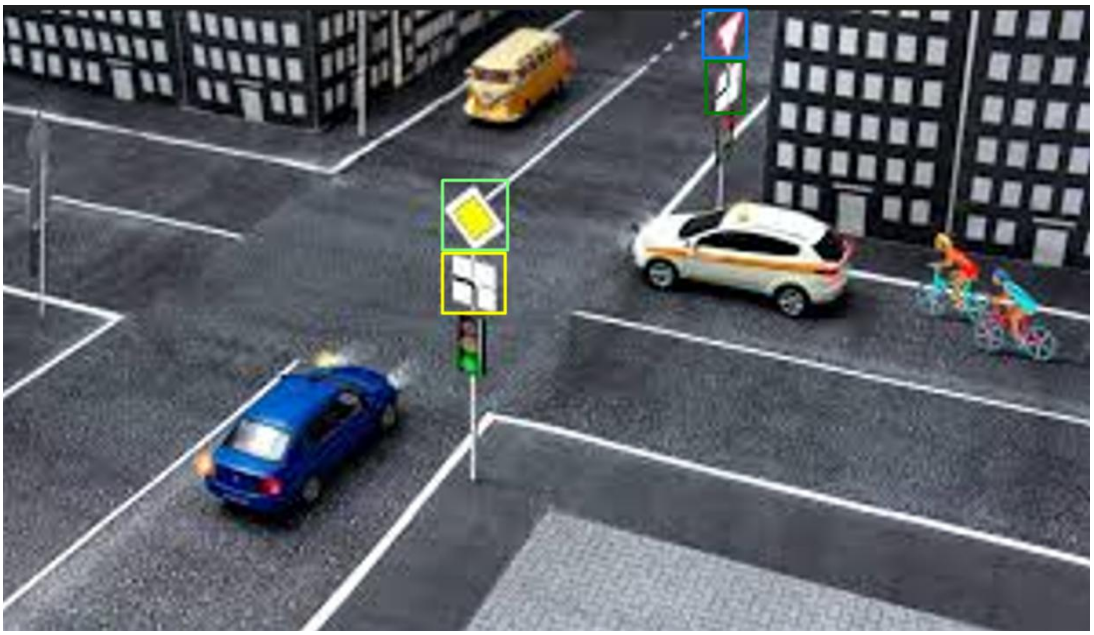

Рисунок 3.19 – Приклад обробленої фотографії штучною нейронной мережею

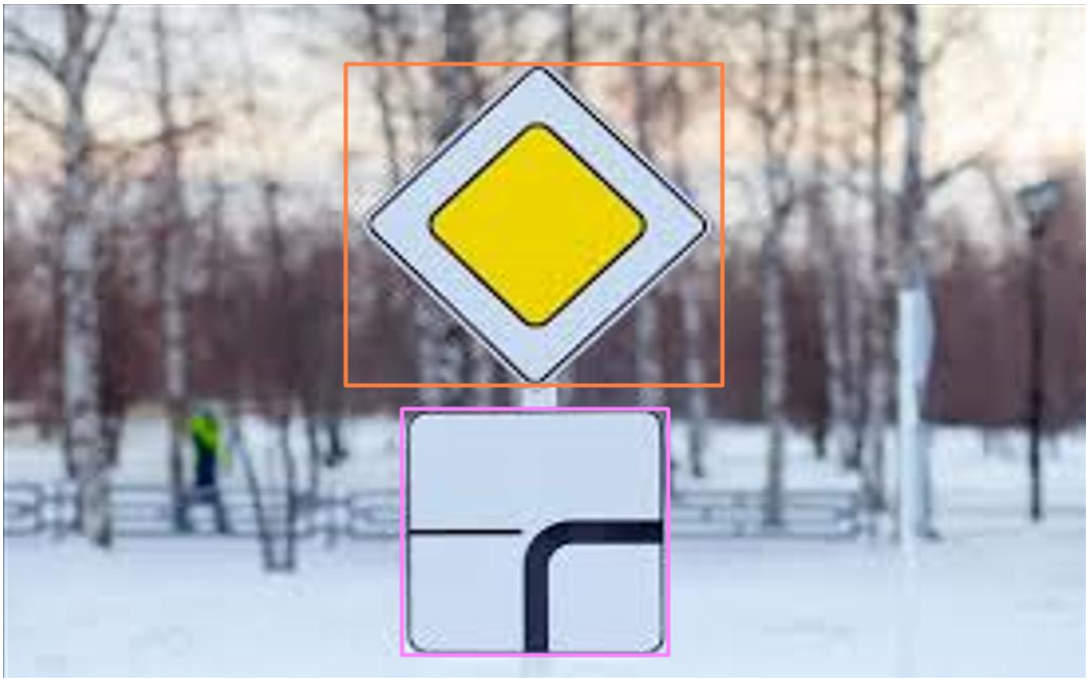

Рисунок 3.20 – Приклад обробленої фотографії штучною нейронной мережею

Рисунок 3.21 – Приклад обробленої фотографії штучною нейронной мережею

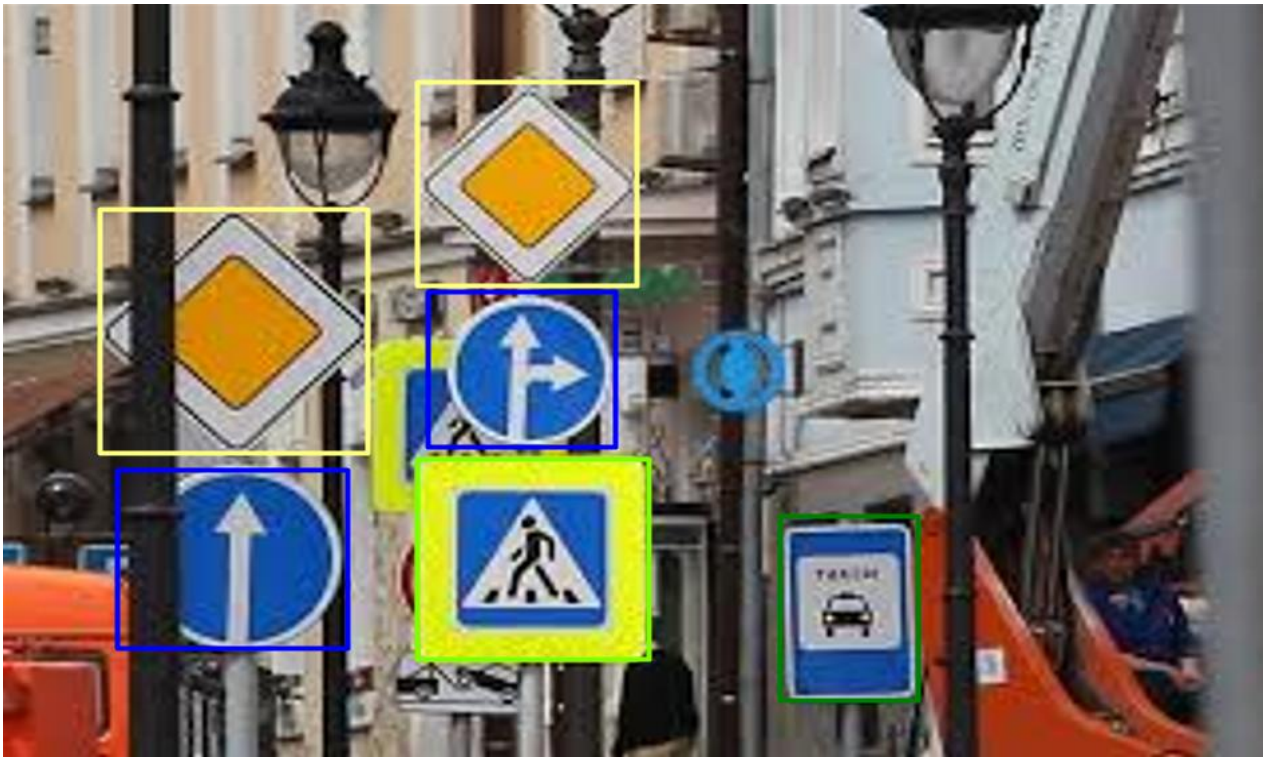

Рисунок 3.22 – Приклад обробленої фотографії штучною нейронной мережею

Рисунок 3.23 – Приклад обробленої фотографії штучною нейронной мережею

Рисунок 3.24 – Приклад обробленої фотографії штучною нейронной мережею

На прикладах вище можна побачити як штучна нейронна мережа на базі навченого класу обробляє знаходження об’єктів на зображеннях. Це дає змогу розширювати кількість класів та об’єктів для розпізнавання та для якості обробки зображень за допомогою штучної нейронної мережі Mask RCNN.

ВИСНОВКИ

В ході виконання кваліфікаційної роботи було розглянуто етапи формування штучної нейронної мережі, підходи до розпізнавання зображень, методи розпізнавання зображень, був виконаний аналіз нейронних мереж які використовуються саме для розпізнавання зображень, такі як CNN, ResNet, Inception, MobileNet, VGGNet. Також були розглянуті сфери використання нейронних мереж, їх різновиди, принцип роботи, та розглянуті вже готові рішення по розпізнаванню об'єктів на зображеннях, такі як R-CNN, Fast R-CNN, Faster R-CNN, та Mask R-CNN. Після цього штучна нейронна мережа Mask R-CNN була досліджена біль детально разом з процесом навчання, підготовкою атрибутів для навчання, та почергове виконання завдання по розробці новогу классу по розпізнаванню знаків дорожнього руху за допомогою штучної нейронної мережі Mask R-CNN.

Також була розроблен підхід для збору набору об’єктів, оформлення анотацій, навчання нейронної мережі, та функціонування додатку з розпізнавання об’єктів на прикладі натренованих даних. Також штучна нейронна мережа має можливість додаткового навчання та вдосконалення ефективності та більшої точності розпізнавання предметів або об’єктів, для здобуття можливості більш глобального використання. Наприклад, для адаптивного автопілоту у сучасних машинах, як у міських, так і у трассових режимах, піддтримка відображення актуальних дорожніх знаків на приладовій панель, для зрозумілої поведінки водія, наприкладі швидкісного режиму на тій чи іншій ділянці дороги.

У тому числі ця розробка має позитивний вплив на:

1. розробку алгоритмів розпізнавання дорожніх знаків, які є більш точними;
2. розробку алгоритмів розпізнавання дорожніх знаків, які можуть розпізнавати дорожні знаки в різних умовах освітлення та погоди;
3. розширення сфер впливу та використання нейронної мережі.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Сервіс вивчення практик нейронних мереж. URL: https://wiki.math.uwaterloo.ca/statwiki/index.php?title=Mask_RCNN.

2. Веб-додаток загальної інформації нейронних мереж. URL: https://forklog.com/chto-takoe-nejronnaya-set/.

3. Стаття на тему “Нейронні мережі”. URL: https://habr.com/post/322392/.

4. Стаття на тему «Застосування нейромереж». URL: https://habr.com/post/74326/.

5. Стаття на тему «Архітектура Mask R-CNN». URL: https://habr.com/post/421299/.

6. Стаття на тему “Mask R-CNN”. URL: https://arxiv.org/abs/1703.06870.

7. Стаття на тему “What is Mask R-CNN?”. URL: https://blog.roboflow.com/mask-rcnn/.

8. Стаття на тему “Everything about Mask R-CNN”. URL: https://viso.ai/deep-learning/mask-r-cnn/.

9. Стаття на тему “How Mask R-CNN Works?”. URL: https://developers.arcgis.com/python/huide/how-maskrcnn-works/.

10. Стаття на тему “Instance Segmentation with Mask R-CNN”. URL: https://towardsdatascience.com/instance-segmentation-with-mask-r-cnn-6e5c4132030b.

11. Стаття на тему “Mask-RCNN for Object Detection”. URL: https://medium.com/mlearning-ai/mask-rcnn-for-object-detection-with-dectectron2-46f62f6b7826.

12. Стаття на тему “Getting Started with Mask R-CNN”. URL: https://www.mathworks.com/help/vision/ug/getting-started-with-mask-r-cnn-for-instance-segmentation.html.

13. Стаття на тему “Mask R-CNN”. URL:

https://www.geeksforgeeks.org/mask-r-cnn-ml/.

14. Стаття на тему “Mask R-CNN - Everything explained”. URL: https://www.picsellia.com/post/mask-r-cnn-everything-explained.

15. Стаття на тему “Mask RCNN”. URL: https://hackmd.io/@imkushwaha/maskrcnn.

16. Стаття на тему “Object Detection: A journey from R-CNN to Mask R-CNN”. URL: https://medium.com/augmented-startups/object-detection-a-journey-from-r-cnn-to-mask-r-cnn-and-yolo-698ba097d490.