Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)
Кафедра Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження та використання методів глибинного навчання для розпізнавання об'єктів заданого типу на зображеннях
(тема)

Виконав:
студент 2 курсу, групи СШІм-22-3
Яценко В.І.
(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)
Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)
Освітня програма Системи штучного інтелекту
(повна назва спеціалізації)
Керівник проф. Рябова Н.В.
(посада, прізвище, ініціали)

Допускається до захисту

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

2024 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Штучного інтелекту
(повна назва)

Рівень вищої освіти другий (магістерський)

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

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системи штучного інтелекту
(повна назва)

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

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Яценку Владиславу Івановичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження та використання методів глибинного навчання для розпізнавання об'єктів заданого типу на зображеннях

затверджена наказом університету від 1 квітня 20 24 р. № 260Ст
2. Термін подання студентом роботи до екзаменаційної комісії 12 червня 20 24 р.
3. Вихідні дані до роботи Науково-технічні публікації та дані Інтернет-джерел щодо тематики кваліфікаційної роботи

4. Перелік питань, що потрібно опрацювати в роботі
1) Дослідження сучасного стан та перспективи розвитку глибинного навчання
2) Огляд загальних характеристик основних видів глибинних нейронних мереж
3) Формулювання задачі дослідження та аналіз основні види задач розпізнавання об'єктів та методи їх вирішення
4) Обґрунтування вибору архітектури згорткової нейронної мережі та опис алгоритм мережі
5) Аналіз отриманих результатів та огляд практичного застосування

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	01.04.2024	виконано
2	Аналіз предметної галузі та постановка задачі	01.04.2024 – 13.04.2024	виконано
3	Дослідження методів розпізнавання зображень	14.04.2024 – 20.04.2024	виконано
4	Дослідження архітектур R-CNN та YOLO	21.04.2024 – 04.05.2024	виконано
5	Дослідження архітектур SSD та EfficientDet	05.05.2024 – 18.05.2024	виконано
6	Навчання моделей та проведення тестів	19.05.2024 – 26.05.2024	виконано
7	Підготовка пояснювальної записки та презентації	27.05.2024 – 03.06.2024	виконано
8	Надання пояснювальної записки на перевірку	04.06.2024 – 11.06.2024	виконано
9	Захист кваліфікаційної роботи	12.06.2024	

Дата видачі завдання 1 квітня 20 24 р.

Студент _____
(підпис)

Керівник роботи _____ проф. Рябова Н.В.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 72 с., 32 рис., 1 дод., 20 джерел.

ГЛИБИННЕ НАВЧАННЯ, ГЛИБИННІ НЕЙРОННІ МЕРЕЖІ, ЗГОРТКОВА НЕЙРОННА МЕРЕЖА, КЛАСИФІКАЦІЯ ОБ'ЄКТІВ, РОЗПІЗНАВАННЯ ОБ'ЄКТІВ, EFFICICIENTDET, FAST R-CNN, FASTER R-CNN, SSD, YOLO.

Об'єкт дослідження – глибинні нейронні мережі.

Предмет дослідження – дослідження та використання методів глибинного навчання для розпізнавання об'єктів заданого типу на зображеннях.

Мета роботи – дослідження глибинних нейронних мереж, зокрема можливостей згорткових нейронних мереж, їх переваг та недоліків, способів та галузей застосування, спроможності розпізнавати різноманітні задані об'єкти на зображеннях, порівняння різних архітектур згорткових нейронних мереж для розпізнавання об'єктів та порівняння згорткових мереж з іншими підходами розпізнавання об'єктів, порівняння їх точності та швидкості, практичне дослідження різних моделей.

Методи дослідження – методи глибинного навчання, методи класифікації зображень з допомогою згорткових нейронних мереж, порівняння переваг та недоліків різних архітектур згорткових нейронних мереж.

У результаті роботи проведено аналіз та порівняння методів R-CNN, Fast R-CNN, Faster R-CNN, YOLO, SSD, EfficientDet, що використовуються для розпізнавання об'єктів на зображеннях. Проведено практичний аналіз методів R-CNN та YOLO. Дане дослідження буде корисне для застосування в різних галузях для покращення точності та швидкодії наявних засобів розпізнавання.

ABSTRACT

Master's thesis contains: 72 p., 32 fig., 1 ann., 20 sources.

CONVOLUTIONAL NEURAL NETWORK, DEEP LEARNING, DEEP NEURAL NETWORKS, EFFICICIENTDET, FAST R-CNN, FASTER R-CNN, OBJECT CLASSIFICATION, OBJECT RECOGNITION, R-CNN, SSD, YOLO.

The object of research is deep neural networks.

The subject of research is analysis and use of deep learning methods for recognizing objects of a given type in images.

The purpose of the work is to study deep neural networks, in particular the possibilities of convolutional neural networks, their advantages and disadvantages, methods and applications, ability to recognize various specified objects in images, comparison of different architectures of convolutional neural networks for object recognition and comparison of convolutional networks with other approaches to object recognition, comparison of their accuracy and speed, practical study of various models.

Research methods are deep learning methods, methods of image classification using convolutional neural networks, comparison of advantages and disadvantages of different convolutional neural network architectures.

The results of this work are the analysis and comparison of R-CNN, Fast R-CNN, Faster R-CNN, YOLO, SSD, EfficientDet methods used for object recognition in images. A practical analysis of the R-CNN and YOLO methods has been carried out. This research will be useful for application in various fields to improve the accuracy and speed of existing recognition tools.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

CNN – Convolutional Neural Network – згорткова нейронна мережа;

FPS – Frames Per Second – кадрів на секунду;

GAN – Generative Adversarial Network – генеративна зворотна мережа;

GPU – Graphics Processing Unit – графічний процесор;

LSTM – Long Short-Term Memory – довга короткочасна пам'ять;

mAP – Mean Average Precision – середня точність;

NMS – Non-Maximum Suppression – немаксимальне придушення;

ReLU – Rectified Linear Unit – прямолінійний вузол;

RoI – Region of Interest – область інтересу;

R-CNN – Region-based Convolutional Neural Network – нейронна мережа з основою на регіонах;

RPN – Region Proposal Network – мережа пропозицій регіонів;

SGD – Stochastic Gradient Descent – стохастичний градієнтний спуск;

SSD – Single Shot MultiBox Detector – однострільний багатообласний детектор;

YOLO – You Only Look Once – дивиться тільки один раз.

ВСТУП

Розпізнавання об'єктів дозволяє машинам розуміти візуальний світ навколо себе, подібно до людського сприйняття, але в масштабі та зі швидкістю, з якими людина не може зрівнятися. Ця здатність є фундаментальною для широкого спектру застосувань.

Розпізнавання об’єктів – це загальний термін для опису завдань комп’ютерного зору, які передбачають ідентифікацію об’єктів на цифрових фотографіях. Класифікація зображень зазвичай робить передбачення класу одного об’єкта на зображенні. Локалізація об’єкта вимагає визначення розташування одного чи кількох об’єктів на зображенні та розмітки обмежувальної рамки навколо об’єкту. Виявлення об’єктів поєднує ці два завдання локалізації та класифікації одного або декількох об’єктів на зображенні. Коли користувач або практик посилається на розпізнавання об'єктів, вони часто мають на увазі виявлення об'єктів [1].

Важливість розпізнавання охоплює все – від автономних транспортних засобів і охоронного спостереження, до фільтрації контенту і медичної візуалізації, надаючи машинам можливість інтерпретувати візуальний світ так само, як і люди. Розпізнавання об'єктів не лише дозволяє автоматизацію нудних завдань, але й розширює можливості систем взаємодіяти з навколишнім середовищем. Це призводить до прогресу в робототехніці та доповненій реальності.

Поява глибокого навчання зробила революцію в розпізнаванні об'єктів, пропонуючи надзвичайну точність і ефективність. Архітектури глибокого навчання, стали основою систем розпізнавання об'єктів завдяки їхній здатності вивчати ієрархічні представлення ознак з величезних обсягів даних. Ці моделі автоматично виділяють і вивчають найбільш релевантні ознаки для поставленого завдання, без необхідність ручного визначення ознак, що було характерно для традиційних підходів до комп'ютерного зору. Еволюція від R-CNN до більш досконалих методів, таких як Faster R-CNN,

Mask R-CNN та YOLO, поступово зменшила розрив між можливостями машинного та людського зору. Ці досягнення підкріплені розробкою великих анотованих наборів даних, потужних обчислювальних ресурсів і архітектур нейронних мереж, що дозволяє навчати глибокі моделі, які можуть розпізнавати широкий спектр об'єктів з великою точністю та швидкістю. Інтеграція глибинного навчання в розпізнаванні об'єктів не лише покращила виконання конкретних завдань, але й відкрила нові шляхи для досліджень.

Отже, об'єктом дослідження є аналіз та порівняння методів глибинного навчання для розпізнавання об’єктів заданого типу на зображеннях. Метою даної роботи є порівняння ефективності та точності методів, що допоможе у виборі конкретного методу для заданого завдання.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Опис предметної галузі

Розпізнавання об'єктів не є новою проблемою. Воно розвивалося протягом декількох десятиліть. Від простих методів, заснованих на шаблонах, до сучасних глибоких нейронних мереж. Еволюція алгоритмів і зростання обчислювальної потужності зробили розпізнавання надзвичайно ефективним. Одним з основних викликів є різноманітність умов зйомки, таких як освітлення, перспектива, часткове приховування об'єктів, і зміна масштабу. Крім того необхідно розрізняти об'єкти, що мають схожі форми чи текстури.

Використання глибоких нейронних мереж допомогло подолати деякі з цих проблем, проте створення точних і швидких моделей все ще вимагає значних зусиль і ресурсів. Майбутнє розпізнавання об'єктів виглядає обнадійливим завдяки постійному прогресу в галузі штучного інтелекту. Дослідники працюють над створенням більш ефективних алгоритмів, які можуть працювати в реальному часі та в різноманітних умовах. Крім того, розвиток технологій нейронних мереж і збільшення доступних даних відкриває нові можливості для покращення точності та швидкості розпізнавання об'єктів.

Розпізнавання об'єктів відіграє велику роль у різних сферах, оскільки воно дозволяє автоматизувати процеси, збільшити ефективність та знизити ризики в різних галузях. В сфері безпеки системи розпізнавання об'єктів використовуються для виявлення небезпечних об'єктів, таких як зброя або вибухові пристрої, на громадських місцях, в аеропортах, на стадіонах тощо. Це допомагає забезпечити безпеку громадян та запобігти терористичним актам та злочинам. У медичній сфері системи розпізнавання об'єктів можуть використовуватися для автоматичного виявлення патологій на зображеннях обстежень, таких як рентгенівські знімки чи зображення з комп'ютерної

томографії. Це допомагає лікарям рано виявляти захворювання та підвищувати шанси на успішне лікування. У сфері автономних автомобілів розпізнавання об'єктів грає критичну роль у виявленні та класифікації дорожніх знаків, пішоходів, інших транспортних засобів та перешкод на дорозі. Це дозволяє автомобілям уникати аварій та забезпечує безпеку для водіїв та пішоходів. У промисловості розпізнавання об'єктів використовується для автоматичного контролю якості виробництва, виявлення дефектів або невідповідності до стандартів. Це допомагає забезпечити якість продукції та знизити відхилення виробництва.

1.2 Актуальність використання глибинного навчання

В сучасному світі обсяг візуальної інформації надто великий для традиційних методів обробки та аналізу, тому глибинне навчання виходить на передній план як ключовий інструмент для розв'язання завдань розпізнавання об'єктів на зображеннях. Дана технологія вирішує численні виклики та дозволяє досягати неймовірних результатів в точності та швидкості обробки великого об'єму візуальних даних.

Глибинне навчання здатне ефективно впоратися з різноманітністю об'єктів, що можуть зустрітися на зображеннях, враховуючи їхні розміри, форми, кольори та текстури. Нейронні мережі можуть автоматично вивчати різні аспекти об'єктів та їхні комбінації без необхідності ручного програмування великої кількості правил. Це особливо корисно в сферах, де важко формалізувати правила для розпізнавання об'єктів. Модель самостійно визначає значущі характеристики та підганяє їх під поставлене завдання [2].

Глибинне навчання дозволяє створювати моделі, які здатні адаптуватися до різних завдань та сценаріїв. Змінюючи архітектуру мережі та використовуючи великі обсяги даних для навчання, можна досягти вражаючої гнучкості та універсальності. А завдяки активному дослідженню

та вдосконаленню алгоритмів, глибинне навчання стало поступово досягати нових рекордів у точності та швидкості роботи моделей, роблячи його найбільш актуальним підходом для задач розпізнавання об'єктів на зображеннях.

1.3 Глибинне навчання

Глибинне навчання – це підгалузь машинного навчання, яке використовує нейронні мережі з багатьма шарами, щоб автоматично вивчати представлення даних з різних рівнів абстракції. Таке навчання зазвичай використовується для вирішення завдань таких як класифікація, регресія, виявлення об’єктів у зображеннях, розпізнавання мови тощо.

Штучні нейронні мережі – це популярний метод машинного навчання, який імітує механізм навчання в біологічних організмах. Нервова система людини містить клітини, які називаються нейронами. Нейрони з'єднуються один з одним за допомогою аксонів і дендритів, а з'єднувальні ділянки між аксонами і дендритами називаються синапсами. Сила синаптичних зв'язків часто змінюється у відповідь на зовнішні подразники. Саме так відбувається навчання в живих організмах. Цей біологічний механізм моделюється у штучних нейронних мережах, які містять обчислювальні одиниці, що називаються нейронами. Обчислювальні одиниці пов'язані одна з одною за допомогою ваг, які відіграють ту ж роль, що і сила синаптичних зв'язків у біологічних організмах. Штучна нейронна мережа обчислює функцію від вхідних даних поширюючи обчислені значення від вхідних нейронів до вихідних нейронів і використовуючи ваги як проміжні параметри. Навчання відбувається шляхом зміни вагових коефіцієнтів, що з'єднують нейрони [3].

Основна ідея глибокого навчання полягає в тому, щоб будувати нейронні мережі, які можуть автоматично знаходити характеристики даних. Кожен шар нейронної мережі може виділяти важливі ознаки або патерни вхідних даних, а комбінація цих шарів дозволяє нейронній мережі

розрізняти складні зв'язки між ознаками та робити точні прогнози або класифікації.

Саме це робить глибоке навчання майже ідеальним інструментом для досягнення результатів. Це робить його особливо ефективним для роботи з великими обсягами даних і складними структурами. Глибоке навчання може бути використане для різноманітних завдань, включаючи розпізнавання об'єктів у зображеннях, розпізнавання мови, машинний переклад, рекомендаційні системи, аналіз текстів, і багато іншого. У багатьох випадках воно демонструє вражаючу точність у вирішенні завдань, що раніше вважалися складними для автоматизації. І найголовніше, що вони можуть виявляти високу адаптивність до змін у вхідних даних або умовах за рахунок великої кількості параметрів, які можна навчати. Все це робить глибоке навчання найкращим засобом, що дозволяє використовувати його у великих проектах та задачах, де потрібна висока точні та обробка великих обсягів неоднорідної інформації.

Глибокі нейронні мережі мають кілька основних компонентів, які дозволяють їм ефективно виконувати завдання з обробки даних.

Вхідні дані – це дані, які подаються на вхід глибокої нейронної мережі. Вони можуть бути представлені у різних форматах, таких як зображення, текст, аудіо або числові дані.

Шари нейронів – це основний будівельний блок глибоких нейронних мереж. Вони складаються з нейронів, які обчислюють зважену суму вхідних сигналів, додають зсув та застосовують функцію активації. Шари нейронів можуть бути різних типів, таких як повністю з'єднані, згорткові, рекурентні тощо.

Ваги – кожен зв'язок між нейронами має свою вагу, яка використовується для зваженої суми вхідних сигналів. Ваги навчаються під час тренування мережі з метою оптимізації її вирішальної здатності.

Функції активації застосовуються до вихідних значень нейронів для введення нелінійності в мережу. Вони можуть бути, наприклад, сигмоїдальними, ReLU, гіперболічним тангенсом тощо.

Функція втрат – це функція, яка вимірює різницю між прогнозованими значеннями мережі та правильними відповідями. Мета цієї функції полягає щоб мінімізувати цю втрату під час тренування.

Оптимізатор – це алгоритм, який використовується для налаштування ваг мережі з метою мінімізації функції втрат. Популярні оптимізатори включають SGD, Adam, RMSProp тощо.

Регуляризація – це методи, які використовуються для уникнення перенавчання і покращення загальної здатності мережі до узагальнення нових даних. Такі методи можуть включати L1 та L2 регуляризацію, dropout та інші.

Процес навчання глибоких нейронних мереж – ітеративний процес, який включає кілька етапів і вимагає великої кількості даних.

Перший етап – підготовка даних. Цей етап включає збір, очищення та підготовку даних для використання в навчанні. Дані можуть бути розділені на тренувальний, валідаційний та тестувальний набори, а також можуть бути здійснені інші операції підготовки.

Другий етап – визначення архітектури мережі. На цьому етапі обирається архітектура мережі, включаючи кількість шарів, кількість нейронів у кожному шарі, типи шарів, функції активації. Це є ключовим етапом, оскільки правильно вибрана архітектура може суттєво покращити результати.

Третій етап – ініціалізація ваг. Ваги мережі починаються з випадкових значень. Це може бути нульове початкове значення, нормальний розподіл або інші методи ініціалізації ваг.

Четвертий етап – проходження вперед. На цьому етапі вхідні дані подаються на вхід мережі, і вони проходять через всі шари мережі. Кожен

нейрон обчислює зважену суму своїх вхідних сигналів та застосовує функцію активації.

П’ятий етап – обчислення втрат. Після проходження вперед обчислюються втрати між прогнозованими значеннями мережі та правильними відповідями за допомогою функції втрат.

Шостий етап – зворотне поширення помилок. Похідна втрати по вагам розповсюджується назад через мережу, і ваги оновлюються за допомогою алгоритму оптимізації з урахуванням цієї похідної.

Сьомий етап – повторення процесу. Цей процес ітеративно повторюється для кожного пакету даних в тренувальному наборі, поки не буде досягнуто критерію зупинки, такого як достатньо низька втрата або кількість епох.

Останній етап – тестування та оцінка. Мережа оцінюється на тестовому наборі даних, щоб оцінити її продуктивність та генералізацію до нових даних.

1.4 Повністю зв'язані мережі

Кожен нейрон в одному шарі з'єднаний з кожним нейроном у наступному шарі. Ці мережі також відомі як мережі прямого поширення і часто використовуються для завдань класифікації та регресії.

Вхідний шар – це перший шар мережі, який отримує вхідні дані. Кількість нейронів у цьому шарі зазвичай відповідає розмірності вхідних даних.

Приховані шари – це шари, які знаходяться між вхідним та вихідним шарами. Кількість та розмірність прихованих шарів може варіюватися від однієї мережі до іншої. Кожен нейрон у прихованому шарі отримує вихідні значення всіх нейронів з попереднього шару, обчислює зважену суму цих значень, і застосовує функцію активації.

Вихідний шар – це останній шар мережі, який генерує вихідні дані або прогнози. Кількість нейронів у вихідному шарі зазвичай відповідає кількості класів у задачі класифікації або кількості вихідних змінних у задачі регресії.

Кожен зв'язок між нейронами у двох сусідніх шарах має свою вагу, яка використовується для зваженої суми вхідних сигналів у кожному нейроні. Під час тренування мережі ці ваги оновлюються за допомогою алгоритму зворотнього поширення помилок, щоб мінімізувати функцію втрат і покращити якість прогнозів мережі.

Повністю зв'язані мережі часто використовуються для вирішення широкого спектру завдань, включаючи класифікацію, регресію, генерацію тексту, виявлення об'єктів у зображеннях, тощо. Вони є одними з найбільш поширених типів нейронних мереж і часто використовуються як основа для більш складних архітектур. У завданнях класифікації повністю зв'язані мережі демонструють високу ефективність у розпізнаванні образів, тексту, звуків та інших видів даних.

1.5 Згорткові нейронні мережі

Згорткові нейронні мережі – це особливий тип нейронних мереж, який широко використовується в обробці зображень та відео. Вони ефективно працюють з проблемами комп'ютерного зору, такими як розпізнавання об'єктів, класифікація зображень, виявлення обличь. Основна властивість згорткових нейронних мереж – це використання згорток, що дозволяє автоматично відбирати характеристики зображення.

Згорткові шари використовуються для виявлення локальних шаблонів у зображеннях. Вони складаються з набору фільтрів, або ядер, які згортаються з вхідними зображеннями для виділення різних характеристик, таких як краї, текстури, форми тощо. Згорткові шари використовуються для вилучення важливих ознак з різних областей зображення.

Пулінгові шари використовуються для зменшення розмірності зображення та підвищення інваріантності до масштабу та позиції об'єктів. Найпоширеніші методи пулінгу – це максимальний пулінг та середній пулінг, де області зображення зменшуються до одного значення, що відповідає найбільшому або середньому значенню в цій області [4].

Повністю зв'язані шари можуть бути додані після згорткових і пулінгових шарів та використовуються для класифікації або регресії. Ці шари приймають ознаки з попередніх шарів і надають остаточний вихід, який може бути інтерпретований для вирішення конкретного завдання.

Як і в інших типах нейронних мереж, в згорткових мережах використовуються функції активації для нелінійності. Зазвичай використовується функція активації ReLU для швидкості та ефективності.

1.6 Рекурентні нейронні мережі

Рекурентні нейронні мережі – це клас нейронних мереж, які спеціалізуються на роботі з послідовними даними, де порядок має значення. Ці мережі мають здатність зберігати і використовувати інформацію про попередні стани для прийняття рішень на поточному кроці часу. Це робить їх особливо ефективними для завдань, таких як розпізнавання мови, машинний переклад, аналіз часових рядів та інші.

Рекурентні шари є ключовою складовою рекурентних мереж. У рекурентних шарах входи не тільки зв'язані з вихідними значеннями від попередніх кроків, але також зв'язані з внутрішнім станом (пам'яттю) шару. Це дозволяє рекурентним мережам зберігати і використовувати інформацію з попередніх кроків часу для прийняття рішення на поточному кроці.

Однією з основних проблем рекурентних мереж є проблема зникаючого градієнта, коли градієнти, які передаються назад у часі, стають дуже малими або зникають, що призводить до втрати інформації про попередні стани. Це може призвести до того, що рекурентні мережі не

можуть ефективно використовувати довгострокову залежність у послідовних даних [5].

Для вирішення цієї проблеми були розроблені покращені архітектури, такі як рекурентні мережі довгої короткочасної пам'яті, які мають здатність ефективно працювати з довгостроковими залежностями у послідовних даних.

Рекурентні нейронні мережі застосовуються в різних сферах, таких як обробка природної мови, машинний переклад або генерація тексту, аналіз часових рядів, генерація музики та багато інших задач, де необхідно моделювати послідовність даних.

1.7 Long Short-Term Memory Networks

Long Short-Term Memory Networks – це спеціальний клас рекурентних нейронних мереж, розроблений для роботи з послідовними даними, такими як текст, аудіо, часові ряди тощо. Вони були розроблені з метою вирішення проблеми зникаючих або вибухаючих градієнтів, що виникають у стандартних рекуретних мережах під час навчання на довгих послідовностях даних. LSTMs виявляються особливо ефективними для завдань, де важлива довготривала залежність між входами в часі.

Клітинний стан – це внутрішнє представлення пам'яті мережі. У кожен момент часу клітинний стан може бути оновлений, додавши нову інформацію або видаливши непотрібну. Це дозволяє LSTM зберігати інформацію на довгі періоди часу.

Воротяні механізми, такі як ворота забування, ворота входу та ворота виведення, використовуються для контролю потоку інформації в клітинний стан. Ці механізми дозволяють визначати, яку інформацію зберігати, яку інформацію оновити і яку інформацію вивести.

LSTM використовує спеціальні функції активації, такі як сигмоїда та тангенс гіперболічний, для керування воротяними механізмами та регулювання значень клітинного стану та вихідних значень.

Однією з головних переваг LSTM є їх здатність до роботи з довгими залежностями між вхідними даними, оскільки вони можуть зберігати інформацію на протязі тривалого періоду часу. Це робить їх особливо корисними для завдань, таких як машинний переклад, генерація тексту, аналіз часових рядів та інші, де важлива контекстуальна інформація з попередніх кроків.

1.8 Generative Adversarial Networks

Generative Adversarial Networks – це клас нейронних мереж, які використовуються для генерації нових даних, таких як зображення, звуки або текст, які максимально схожі на дані, що навчали мережу. Головна ідея GAN полягає в тому, що два модулі – генератор і дискримінатор – змагаються один з одним, поки генератор не навчиться генерувати достатньо реалістичні дані [6].

Генератор – нейронна мережа, яка призначена для генерації нових даних, наприклад, зображень. Вона приймає на вхід випадковий вектор і генерує нові дані, які намагаються бути схожими на дані з навчального набору.

Дискримінатор – інша нейронна мережа, яка використовується для визначення, наскільки схожі згенеровані дані відповідають реальним даним з навчального набору. Вона приймає на вхід дані і визначає ймовірність того, що ці дані є реальними чи згенерованими.

Функція втрат використовується для навчання як генератора, так і дискримінатора. Ця взаємодія створює своєрідне змагання, де генератор намагається підвищити якість генерованих даних, а дискримінатор – підвищити свою здатність відрізняти дані. Головна перевага GAN полягає в

тому, що вони можуть генерувати дуже реалістичні дані, які важко відрізнити від справжніх даних. Вони використовуються в багатьох областях, включаючи комп'ютерний зір, обробку природної мови, генерацію зображень та інші. Однак навчання GAN може бути складним і вимагати великої кількості даних та обчислювальних ресурсів.

1.9 Розпізнавання зображень

Для розпізнавання зображень найчастіше використовуються згорткові нейронні мережі. Вони спеціалізуються на обробці зображень та мають деякі ключові переваги, що роблять їх ефективними для цієї задачі.

Згорткові мережі ефективно враховують просторові залежності в зображеннях, використовуючи згорткові шари, які застосовують фільтри до різних частин зображення та можуть навчатися визначати широкий спектр об'єктів та властивостей на зображеннях. Ці мережі можуть автоматично визначати корисні функції та ознаки без явного програмування, що робить їх ефективними для завдань, де немає чітких правил або шаблонів. Та можуть бути дуже глибокими, що дозволяє їм автоматично визначати складніші шаблони та ознаки на зображеннях [7].

1.10 Актуальність та можливе застосування

Розпізнавання об'єктів на зображеннях – це ключова технологія, що має великий потенціал у багатьох сферах життя. Завдяки постійному розвитку алгоритмів машинного навчання та швидкій збільшенню обчислювальної потужності, розпізнавання об'єктів стає все більш точним та ефективним.

Розпізнавання об'єктів дозволяє автоматизувати багато рутинних завдань у різних галузях, таких як виробництво, сільське господарство, медицина. У виробництві автомобілів, системи розпізнавання можуть

виявляти дефекти на вироблених деталях, що дозволяє швидко виправляти помилки та підвищувати якість продукції. Використання систем розпізнавання об'єктів у відеоспостереженні дозволяє виявляти небезпечні об'єкти або поведінку, що може свідчити про загрозу людям. У медичній галузі системи розпізнавання об'єктів можуть бути використані для автоматичного виявлення патологій на зображеннях, таких як рентгенограми або знімки магнітного резонансу.

Можливе застосування стосується аналізу великих обсягів відеоданих. Це може включати виявлення об'єктів у відео з відеоспостереження, аналіз поведінки людей на вулицях або й навіть в реальному часі. Та бути використано для автоматичного збору та аналізу даних з великої кількості зображень. Це дозволяє отримувати цінні дані для подальшого використання у різних сферах, таких як наукові дослідження, транспортні аналізи, екологічні спостереження тощо.

1.11 Постановка задачі

У даній кваліфікаційній роботі ставимо за мету проаналізувати та порівняти методи розпізнавання об’єктів заданого типу на зображеннях з метою використання їх для якнайкращого виконання відповідного завдання.

У рамках розроблюваної системи ставимо такі задачі:

– детально вивчити предметну область обраної теми;

– обрати методи розпізнавання об’єктів та проаналізувати її переваги та недоліки;

– провести експериментальне порівняння методів з метою оцінки їхньої точності та ефективності.

2 АКТУАЛЬНІ АРХІТЕКТУРИ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ

2.1 Метод R-CNN

Звичайні згорткові нейронні мережі використовуються для класифікації зображень у великому масштабі, коли потрібно визначити, що зображено на цілому зображенні. Вони допомагають визначити, наприклад, чи на зображенні зображений кіт чи собака, машина чи велосипед, людина чи не людина.

З іншого боку, R-CNN (Region-based Convolutional Neural Networks) та їх варіації, такі як Fast R-CNN, Faster R-CNN та Mask R-CNN, розроблені спеціально для задач локалізації та класифікації окремих об'єктів на зображеннях. Вони працюють шляхом виявлення та обробки окремих областей або пропозицій, які називаються областями зацікавленості, Region of Interest, ROI, та які містять потенційні об'єкти, та потім класифікують ці області. Це дозволяє точніше визначати місцезнаходження та клас об'єктів на зображеннях.

Основна ідея R-CNN полягає в тому, щоб спочатку виявити області зацікавленості на вихідному зображенні, які містять потенційні об'єкти, а потім використовувати CNN для класифікації цих областей та точної локалізації об'єктів всередині них [8].

Перший етап – виявлення пропозицій областей зацікавленості. У цьому етапі використовуються методи виявлення об'єктів, такі як Selective Search або EdgeBoxes, для генерації пропозицій областей, які містять потенційні об'єкти. Ці пропозиції можуть бути різного розміру та аспектного співвідношення.

Другий етап – використання згорткової нейронної мережі для класифікації та локалізації. Кожна пропозиція області зацікавленості подається на вхід до згорткової нейронної мережі, яка класифікує цю

область та виконує точну локалізацію об'єкта всередині неї. Ця локалізація може включати в себе визначення обмежувальних рамок навколо об'єктів.

Третій етап – оптимізація та навчання. R-CNN може бути навчений за допомогою звичайних методів навчання з наглядом, таких як зворотне поширення помилок. Після навчання моделі вона може бути оптимізована для кращої точності та швидкості.

R-CNN та його варіації, такі як Fast R-CNN, Faster R-CNN та Mask R-CNN, виявилися дуже ефективними для завдань розпізнавання об'єктів на зображеннях, зокрема в областях комп'ютерного зору, медичної діагностики, автономного водіння та багатьох інших. Вони забезпечують високу точність розпізнавання та локалізації, а також можуть працювати з різними типами об'єктів та сценами [8].

Архітектура R-CNN складається з кількох основних блоків, кожен з яких відповідає за конкретну частину процесу виявлення об'єктів на зображеннях:

– вхідне зображення. Процес починається з вхідного зображення, яке може бути будь-якого розмір;

– вибірковий пошук. R-CNN спочатку генерує набір пропозицій областей, використовуючи алгоритм вибіркового пошуку. Ці пропозиції областей – це області на зображенні, які, ймовірно, містять об'єкти;

– об'єднання пропозицій регіонів. Кожна пропозиція області вирізається з вхідного зображення і змінюється до фіксованого розміру. Ці обрізані області потім подаються в CNN для виділення ознак;

– виділення ознак. Обрізані регіони пропускаються через попередньо навчену CNN для вилучення ознак. CNN витягує високорівневі представлення ознак з кожної області;

– класифікація SVM. Ознаки, витягнуті з кожної пропозиції регіону, подаються в окремі класифікатори машини опорних векторів (SVM). Кожен SVM-класифікатор навчений класифікувати, чи містить регіон об'єкт певного класу або фон;

– регресія рамки обмеження. R-CNN також включає крок регресії рамки обмеження для уточнення обмежувальних рамок, згенерованих алгоритмом селективного пошуку. Цей крок допомагає підвищити точність прогнозів граничних областей;

– немаксимальне придушення. Після класифікації та регресії застосовується немаксимальне придушення для видалення надлишкових обмежувальних рамок. Це гарантує, що будуть збережені лише найбільш достовірні та точні виявлення;

– виведення виявлених об'єктів. Виводяться решта обмежувальних рамок разом з відповідними їм мітками класів та оцінками, що представляють виявлені об'єкти на вхідному зображенні.

Послідовність блоків зображена на рисунку 2.1.

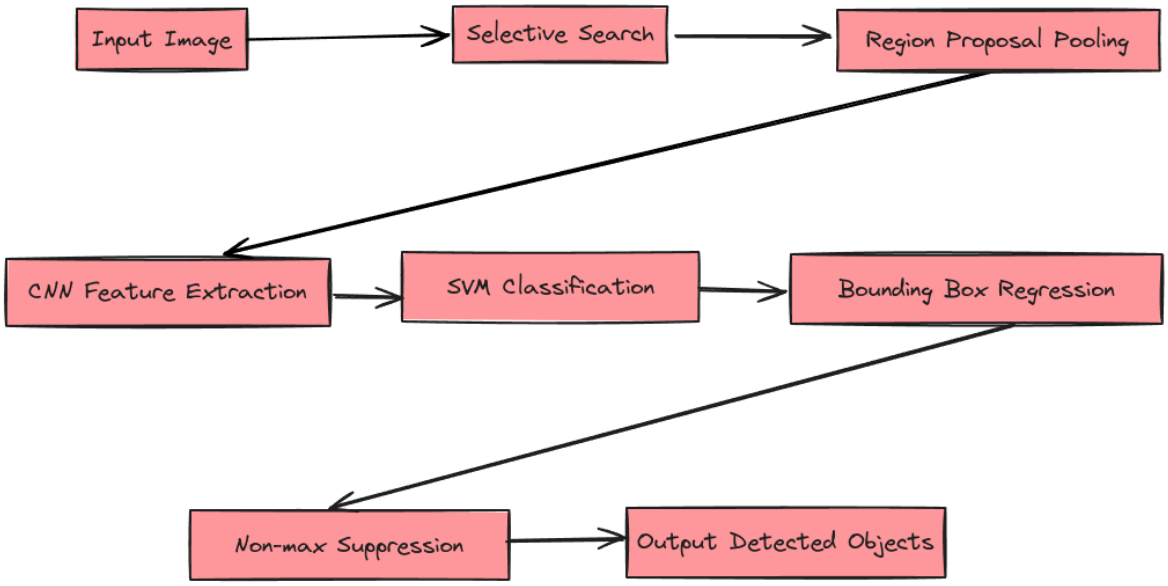

Рисунок 2.1 – Метод R-CNN

2.2 Вибірковий пошук

Вибірковий пошук, або Selective Search, є методом сегментації та об'єднання, який використовується в області комп'ютерного зору для

виявлення об'єктів на зображеннях. Цей метод є ефективним і широко застосовуваним для підготовки регіонів пропозицій, які потім можуть бути аналізовані з використанням алгоритмів машинного навчання для розпізнавання об'єктів.

На першому етапі (рисунок 2.2) зображення розбивається на множину дрібних сегментів за допомогою алгоритму сегментації. Зазвичай використовуються методи як графічне сегментування, що базується на кольорі, текстурі, розмірі сегментів та їх подібності.

На наступному етапі маленькі сегменти поступово об'єднуються на основі схожості їх текстур, кольорів, розміру, та орієнтації. Мета полягає в тому, щоб поступово побудувати більші та більш цілісні області, які можуть відповідати об'єктам на зображенні.

Кінцевим результатом є набір регіонів пропозицій, кожен з яких має потенціал містити об'єкт. Ці регіони визначаються як області, які відрізняються від своїх сусідів та мають значну величину.

Рисунок 2.2 – Етапи вибіркового пошуку

2.3 Регресія рамки обмеження

Після виділення об'єкта на зображенні за допомогою вибіркового пошуку, SVM використовується для класифікації, чи належить виділений регіон до класу об'єкта, який ми шукаємо. Після того, як SVM класифікував регіон як потенційний об'єкт, регресія рамки обмеження використовується для уточнення меж цього регіону.

Суть полягає в тому, щоб взяти початкову рамку, визначену під час попередніх етапів (можливо, грубо), і скорегувати її координати, щоб більш точно обмежити об'єкт. Це досягається шляхом навчання регресійної моделі, яка приймає на вхід характеристики об'єкта і видає зсуви координат рамки у порівнянні з початковими координатами.

На рисунку 2.3 зображено червону рамку об’єкта отриману від вибіркового пошуку та синю рамку отриману в результаті зсуву регресії.

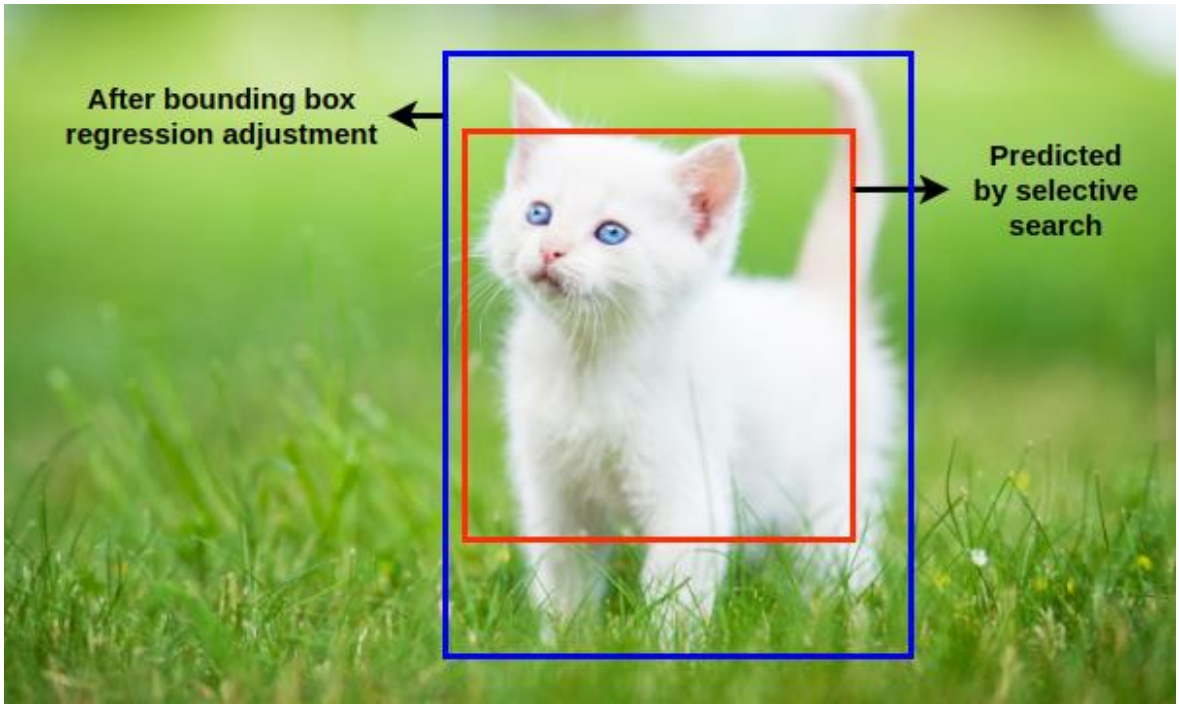

Рисунок 2.3 – Регресія рамки обмеження

2.4 Немаксимальне придушення

Немаксимальне придушення потрібне при визначенні об'єктів на зображенні для усунення дублюючих або надмірних розпізнавань одного і того ж об'єкта.

Після виконання алгоритму розпізнавання об'єктів може бути багато обмежувальних рамок, які перекривають один одного і належать до одного об'єкта. Кожен обмежувальний прямокутник має певний коефіцієнт впевненості, що показує, наскільки ймовірно, що цей прямокутник дійсно містить об'єкт.

Немаксимальне придушення сортує всі прямокутники за їхнім коефіцієнтом впевненості, обирає найвпевненішій прямокутник і видаляє всі інші, які сильно перекриваються з ним. Цей процес повторюється для всіх прямокутників, поки не залишиться один (рисунок 2.4).

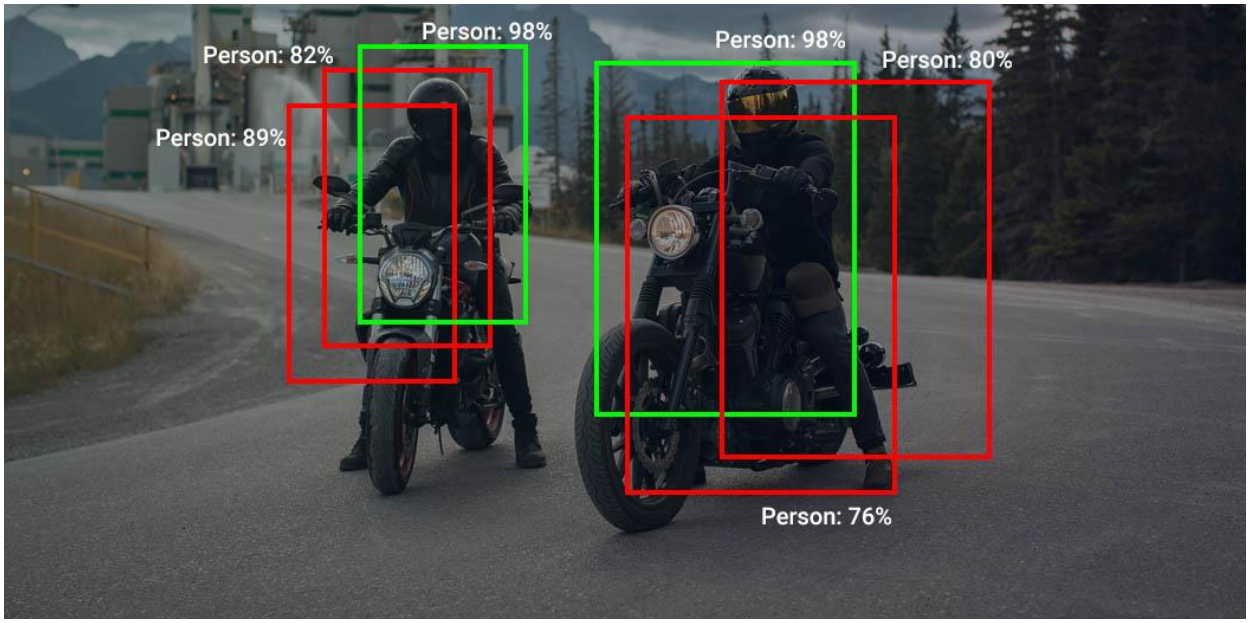

Рисунок 2.4 – Немаксимальне придушення

2.5 Метод Fast R-CNN

Fast R-CNN є вдосконаленою версією архітектури R-CNN, яка була розроблена для покращення швидкості та ефективності виявлення об'єктів на зображеннях. Основне вдосконалення Fast R-CNN полягає в тому, що вона об'єднує в собі процес виявлення областей зацікавленості та класифікації цих областей у вихідній згортковій нейронній мережі, що дозволяє виконувати ці кроки швидше та ефективніше.

Замість використання зовнішніх алгоритмів, таких як Selective Search або EdgeBoxes, Fast R-CNN використовує внутрішній механізм Region Proposal Network для генерації пропозицій областей зацікавленості. Це дозволяє об'єднати процес генерації пропозицій та класифікації в одній мережі, що полегшує навчання та підвищує швидкість [8].

У Fast R-CNN обчислення для всіх областей зацікавленості виконуються відразу під час проходження крізь згорткову нейронну мережу, що дозволяє обчислювати функцію втрат та градієнти спільно для всіх областей. Це дозволяє значно скоротити час обробки. За рахунок оптимізації процесу виявлення об'єктів та об'єднання процесів в одній мережі, Fast R-CNN забезпечує значно вищу швидкість обробки зображень порівняно з R-CNN, при цьому забезпечуючи або підвищуючи точність розпізнавання об'єктів [8].

Архітектура Fast R-CNN складається з таких блоків:

– вхідне зображення. Процес починається з вхідного зображення, яке може бути будь-якого розміру;

– виділення ознак. На відміну від R-CNN, який витягує ознаки окремо для кожної пропозиції регіону, Fast R-CNN витягує ознаки для всього зображення за допомогою CNN;

– пропозиція регіону. Fast R-CNN використовує окрему мережу пропозицій регіонів (RPN), таку як Selective Search або EdgeBoxes, для генерування пропозицій регіонів;

– об'єднання RoI. Регіони інтересу (RoI) виділяються з карт об'єктів, створених CNN. Потім застосовується об'єднання RoI для деформації об'єктів у межах кожного RoI у карту об'єктів фіксованого розміру. Цей крок дозволяє вирівняти об'єкти різного розміру до спільного просторового масштабу;

– повністю з'єднані шари. Об'єднані в пул об'єкти RoI проходять через серію повністю пов'язаних шарів, які додатково обробляють об'єкти і готують їх до класифікації та регресії граничних областей;

– класифікація та регресія на основі граничної області. Fast R-CNN використовує два споріднені вихідні шари: один для класифікації об'єктів, а інший для регресії з обмеженнями;

– немаксимальне придушення. Після класифікації та регресії обмежувальних рамок застосовується немаксимальне придушення (NMS) для видалення надлишкових обмежувальних рамок;

– виведення виявлених об'єктів. Виводяться решта обмежувальних рамок разом з відповідними їм мітками класів та оцінками, що представляють виявлені об'єкти на вхідному зображенні.

Послідовність блоків зображена на рисунку 2.5.

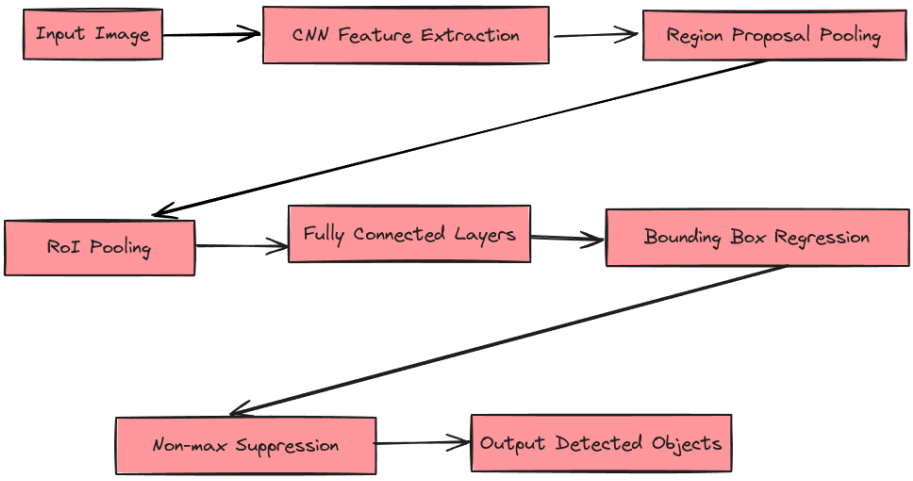

Рисунок 2.5 – Метод Fast R-CNN

Основна відмінність Faster R-CNN від R-CNN та Fast R-CNN полягає в тому, що вона використовує вбудовану мережу Region Proposal Network (RPN) для генерації пропозицій областей зацікавленості безпосередньо з вихідного зображення. RPN генерує пропозиції областей зацікавленості, враховуючи різні масштаби та аспекти зображення [8], [9].

2.6 Метод Faster R-CNN

У Faster R-CNN згорткові шари, що відповідають за виявлення ознак та шаблонів, використовуються як для RPN, так і для відповідних шарів, які визначають клас та локалізацію об'єктів. Це дозволяє ефективно використовувати обчислювальні ресурси та зменшує час навчання та виконання.

Завдяки використанню RPN для генерації пропозицій областей зацікавленості та спільному використанню параметрів згорткових шарів, Faster R-CNN може працювати набагато швидше, ніж попередні архітектури, при цьому забезпечуючи або підвищуючи точність виявлення об'єктів [10].

Архітектура Faster R-CNN є покращенням Fast R-CNN і включає наступні етапи:

– вхідне зображення. Процес починається з вхідного зображення, яке може бути будь-якого розміру;

– виділення ознак. Подібно до Fast R-CNN, Faster R-CNN витягує ознаки для всього зображення за допомогою CNN;

– регіональна мережа пропозицій. Faster R-CNN представляє мережу пропозицій регіонів (RPN), яка використовує ту ж саму базову мережу CNN, що і мережа виявлення. RPN генерує пропозиції регіонів, накладаючи невелику мережу на карти об'єктів CNN;

– генерація якірних блоків. RPN генерує опорні рамки в різних масштабах і співвідношеннях сторін на картах об'єктів. Ці опорні блоки слугують еталонними блоками для регіональних пропозицій [10];

– об'єднання RoI. Подібно до Fast R-CNN, регіони інтересу виділяються з карт об'єктів. Потім застосовується об'єднання RoI для деформації об'єктів в межах кожного RoI у карту об'єктів фіксованого розміру;

– виділення об'єктів за регіонами. Об'єкти, що відповідають кожному RoI, витягуються з карт об'єктів;

– повністю з'єднані шари. Вирівняні за RoI об'єкти проходять через повністю з'єднані шари, які далі обробляють об'єкти і готують їх до класифікації та регресії граничних областей;

– класифікація та регресія на основі обмежувальних рамок. Подібно до Fast R-CNN, Faster R-CNN використовує два споріднені вихідні шари;

– немаксимальне придушення. Після класифікації та регресії обмежувальних рамок застосовується немаксимальне придушення для видалення надлишкових обмежувальних рамок. Цей крок гарантує, що будуть збережені лише найбільш достовірні та точні виявлення;

– виведення виявлених об'єктів.

Виводяться решта обмежувальних рамок разом з відповідними їм мітками класів та довірчими оцінками, що представляють виявлені об'єкти на вхідному зображенні.

Послідовність блоків зображена на рисунку 2.6.

Загалом, Faster R-CNN є важливим кроком у напрямку розробки швидших та більш ефективних алгоритмів виявлення об'єктів на зображеннях, що забезпечують високу швидкість та точність. Це робить його популярним вибором у багатьох застосуваннях.

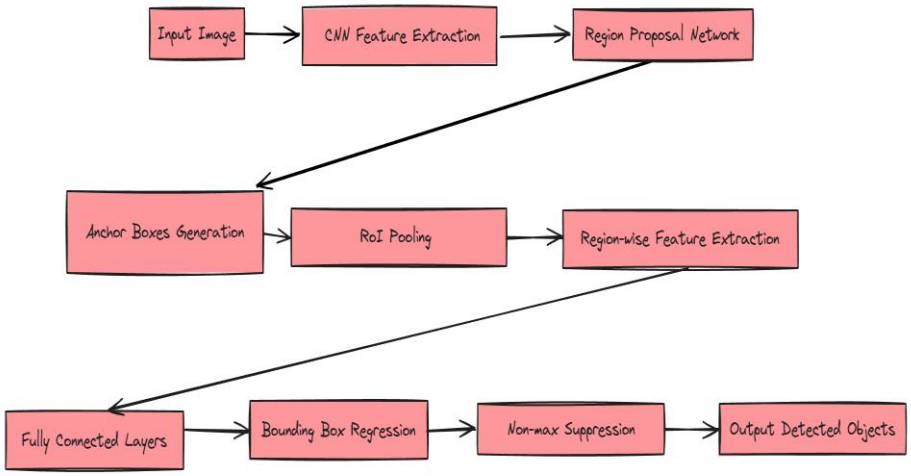

Рисунок 2.6 – Метод Faster R-CNN

2.7 Метод YOLO

YOLO (You Only Look Once) – це інноваційний алгоритм для виявлення об'єктів на зображеннях, який відрізняється від інших підходів своєю швидкістю та ефективністю.

Основна ідея YOLO полягає в тому, що алгоритм здійснює прогнозування класу та локалізацію об'єктів одночасно на вихідному зображенні, використовуючи всього одну згорткову нейронну мережу [11].

Основні відмінності YOLO від R-CNN та його похідних:

– один прохід. Однією з ключових особливостей YOLO є те, що вона використовує один прохід зображення через нейронну мережу для виявлення об'єктів. Це означає, що YOLO аналізує всі області зображення одразу, а не розділення областей на пропозиції як у R-CNN;

– глобальне відображення об'єктів. YOLO розглядає всю картину одразу і вирішує, які класи об'єктів та де знаходяться. У порівнянні з методом R-CNN, який переглядає багато пропозицій, це може забезпечити більш ефективне використання обчислювальних ресурсів;

– швидкість. Завдяки своєму однопрохідному підходу та глобальному відображенню, YOLO може працювати значно швидше, ніж R-

CNN та його похідні. Це робить YOLO привабливим вибором для реального часу або високоефективних систем виявлення об'єктів;

– проблеми з точністю локалізації та деталізацією.

Однак YOLO може мати проблеми з точністю локалізації об'єктів, особливо коли об'єкти мають дрібні деталі або знаходяться дуже близько один до одного. Це може призвести до втрати деякої деталізації та точності порівняно з методами, які використовують більш деталізовані пропозиції областей.

Але завдяки своєму інноваційному підходу та швидкодії, YOLO стала популярним вибором для багатьох застосувань у галузі комп'ютерного зору. Її здатність швидко та ефективно виявляти об'єкти робить її незамінною в багатьох сучасних системах, хоча певні обмеження в точності локалізації все ще залишають простір для подальших досліджень та вдосконалень.

Архітектура YOLO включає такі етапи:

– вхідне зображення. Процес починається з вхідного зображення, яке може бути будь-якого розміру;

– попередня обробка. Перш ніж зображення потрапляє в мережу YOLO, воно проходить етапи попередньої обробки, такі як зміна розміру, нормалізація та перетворення формату, щоб зробити його сумісним з нейронною мережею;

– проходження через мережу YOLO. Попередньо оброблене зображення пропускається через мережу YOLO, яка складається з декількох згорткових шарів;

– прогнозування обмежувальних рамок. YOLO розбиває зображення на сітку і прогнозує граничні області та ймовірності класів для кожної комірки сітки. Кожна обмежувальна рамка містить координати центру рамки, ширину, висоту та довірчу оцінку, що відображає ймовірність наявності об'єкта [11];

– немаксимальне придушення. Після отримання декількох обмежувальних рамок з різною достовірністю застосовується немаксимальне придушення, щоб видалити повторні виявлення. Для кожного об'єкта залишаються лише найбільш достовірні межі;

– прогнозування класу об'єкта. Для кожної залишеної області YOLO прогнозує клас об'єкта, що міститься в ній, разом з відповідними оцінками ймовірності;

– визначення порогових значень. Прогнозовані обмежувальні рамки та пов'язані з ними ймовірності класів фільтруються на основі певного порогового значення. Області з низькою довірчою ймовірністю відкидаються;

– виведення виявлених об'єктів. Виводяться решта обмежувальних рамок разом з відповідними їм мітками класів та довірчими оцінками, що представляють виявлені об'єкти на вхідному зображенні.

Послідовність блоків зображена на рисунку 2.7.

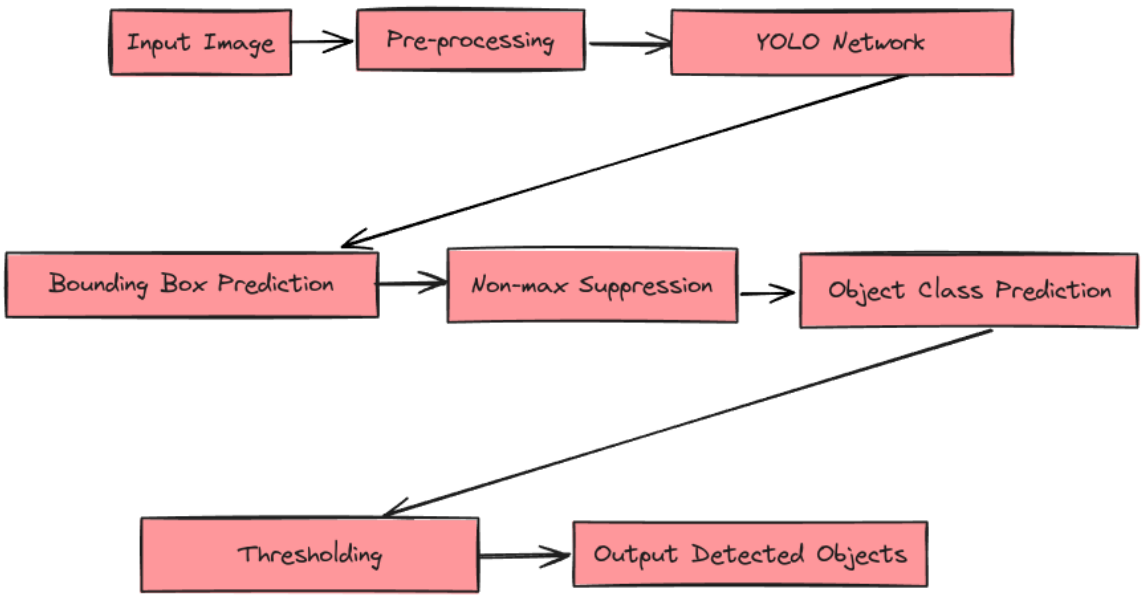

Рисунок 2.7 – Метод YOLO

2.8 Метод SSD

SSD (Single Shot Multibox Detector) – це архітектура для виявлення об'єктів на зображеннях, яка вирішує проблему швидкості та точності, характерну для попередніх підходів, таких як R-CNN та його варіації.

Основна відмінність SSD полягає в тому, що вона використовує одновимірний підхід для виявлення об'єктів на зображенні, який відрізняється від більш традиційного двовимірного підходу R-CNN. Замість використання Region Proposal Network для генерації пропозицій областей зацікавленості, SSD використовує набір фіксованих прямокутних регіонів різного розміру та аспектного співвідношення (називаних анкорами), які обробляються для визначення областей з високою ймовірністю містять об'єкти [12].

Ще одна ключова особливість SSD полягає в тому, що вона виконує виявлення об'єктів та класифікацію в одному кроці. Це означає, що SSD може виявляти та класифікувати об'єкти на зображеннях швидше та ефективніше, оскільки вона не вимагає окремого етапу генерації пропозицій областей зацікавленості. SSD використовує мережу згорткових шарів для обробки вхідного зображення та визначення ознак та множинні масштаби, що дозволяють виявляти об'єкти різних розмірів та масштабів на зображенні [13].

Архітектура SSD включає такі етапи:

– вхідне зображення. Процес починається з вхідного зображення, яке може бути будь-якого розміру;

– попередня обробка. Подібно до YOLO, вхідне зображення проходить етапи попередньої обробки, такі як зміна розміру, нормалізація та перетворення формату, щоб підготувати його до введення в мережу SSD;

– проходження через мережу SSD. Попередньо оброблене зображення пропускається через мережу SSD, яка зазвичай складається з базової згорткової нейронної мережі і додаткових шарів для виявлення;

– виділення ознак. Базова мережа CNN виокремлює ознаки з вхідного зображення в різних масштабах і роздільній здатності;

– різномасштабні карти об'єктів. SSD генерує карти ознак у кількох різних масштабах, щоб захопити об'єкти різного розміру. Ці карти створюються на різних рівнях мережі;

– генерація якірних блоків. SSD використовує опорні точки, які є заздалегідь визначеними з різним співвідношенням сторін і масштабами, рівномірно розподіленими по кожній комірці карти об'єктів. Ці опорні блоки слугують еталонними блоками для виявлення об'єктів різної форми та розміру;

– прогнозування блоків. SSD прогнозує зміщення і достовірність для кожного опорного квадрата на всіх картах об'єктів;

– немаксимальне придушення. Подібно до YOLO, не максимальне придушення застосовується для фільтрації надлишкових виявлень, залишаючи лише найбільш достовірні обмежувальні рамки;

– прогнозування класу об'єкта. Для кожної області, що залишилася, SSD пророкує клас об'єкта, що міститься в ній, разом з відповідними оцінками ймовірності;

– визначення порогових значень. Прогнозовані обмежувальні рамки та пов'язані з ними ймовірності класів фільтруються на основі певного порогового значення. Області з низькою довірчою ймовірністю відкидаються;

– виведення виявлених об'єктів. Виводяться решта обмежувальних рамок разом з відповідними їм мітками класів та довірчими оцінками, що представляють виявлені об'єкти на вхідному зображенні.

Послідовність блоків зображена на рисунку 2.8.

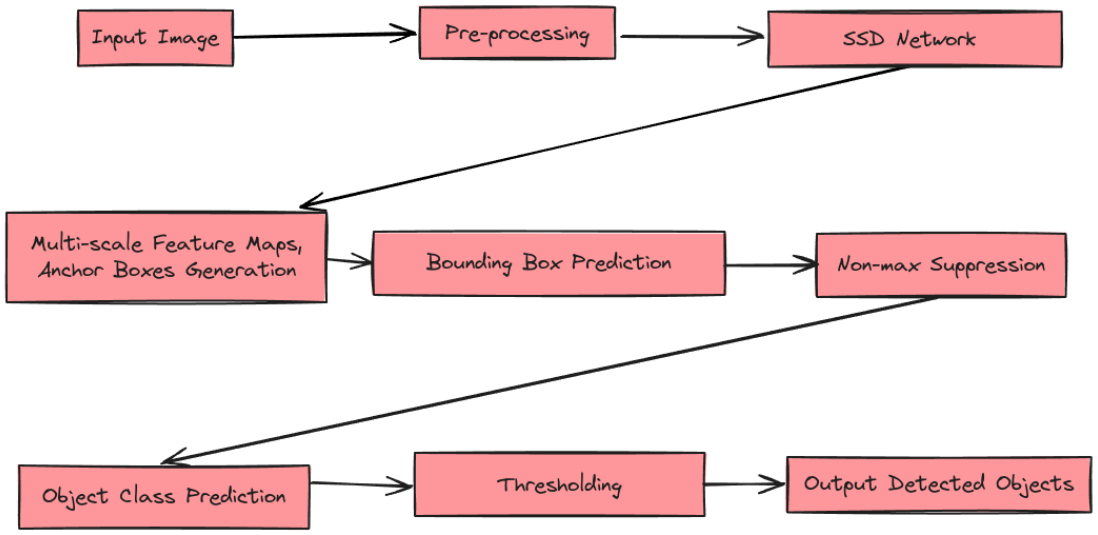

Рисунок 2.8 – Метод SSD

2.9 Метод EfficientDet

EfficientDet – це метод виявлення об'єктів на зображеннях, який поєднує в собі високу швидкість та високу точність, оптимізуючи ресурсоємність та обчислювальну складність. Цей підхід базується на архітектурі EfficientNet для класифікації зображень та додаткових оптимізаціях для виявлення об'єктів.

EfficientDet використовує мережі з різними масштабами для аналізу зображення на різних рівнях деталізації. Це дозволяє ефективно виявляти об'єкти різних розмірів та вирішувати проблему низької точності в малих об'єктах [14].

EfficientDet використовує BiFPN для ефективного збільшення кількості функцій відображення та підвищення точності виявлення. BiFPN дозволяє ефективно об'єднувати інформацію з різних рівнів мережі та покращує здатність моделі розпізнавати об'єкти на зображеннях [15].

Використовує комплексне масштабування, яке оптимізує кількість параметрів та обчислювальну складність моделі відносно до швидкості та

точності. Це дозволяє побудувати ефективніші та швидші моделі для виявлення об'єктів.

EfficientDet показує високу точність виявлення об'єктів на зображеннях, набагато перевершуючи попередні методи, такі як YOLO та SSD. Через різні конфігурації та оптимізації, EfficientDet може бути використаний для широкого спектру застосувань, від вбудованих систем до систем великого масштабу.

Архітектура EfficientDet включає такі етапи:

– вхідне зображення. Процес починається з вхідного зображення, яке може бути будь-якого розміру;

– проходження через мережу EfficientDet;

– виділення ознак. Магістральна мережа виокремлює ієрархічні ознаки з вхідного зображення. В EfficientDet EfficientNet слугує опорною мережею, використовуючи її ефективність для захоплення багатих представлень ознак;

– BiFPN і головки виявлення об'єктів. EfficientDet використовує BiFPN (двонаправлену мережу піраміди ознак) для ефективного об'єднання ознак різних масштабів і роздільної здатності. Головки виявлення об'єктів прикріплюються до карт об'єктів, створених BiFPN, для прогнозування обмежувальних рамок, класів об'єктів і оцінок достовірності;

– генерація якірних блоків. Подібно до SSD, EfficientDet використовує опорні блоки, які є попередньо визначеними блоками з різними співвідношеннями сторін і масштабами, розподіленими по картах об'єктів, створених BiFPN;

– прогнозування граничних блоків. EfficientDet прогнозує зміщення і довірчі оцінки для кожного опорного блоку на всіх картах ознак;

– немаксимальне придушення. Як і в YOLO та SSD, немаксимальне придушення застосовується для фільтрації надлишкових виявлень, залишаючи лише найбільш достовірні області;

– прогнозування класу об'єкта. Для кожної області, що залишилася, EfficientDet прогнозує клас об'єкта, що міститься в ній, разом з відповідними оцінками ймовірності;

– порогове значення. Прогнозовані обмежувальні рамки та пов'язані з ними ймовірності класів фільтруються на основі певного порогового значення. Області з низькою довірчою ймовірністю відкидаються;

– виведення виявлених об'єктів.

Послідовність блоків зображена на рисунку 2.9.

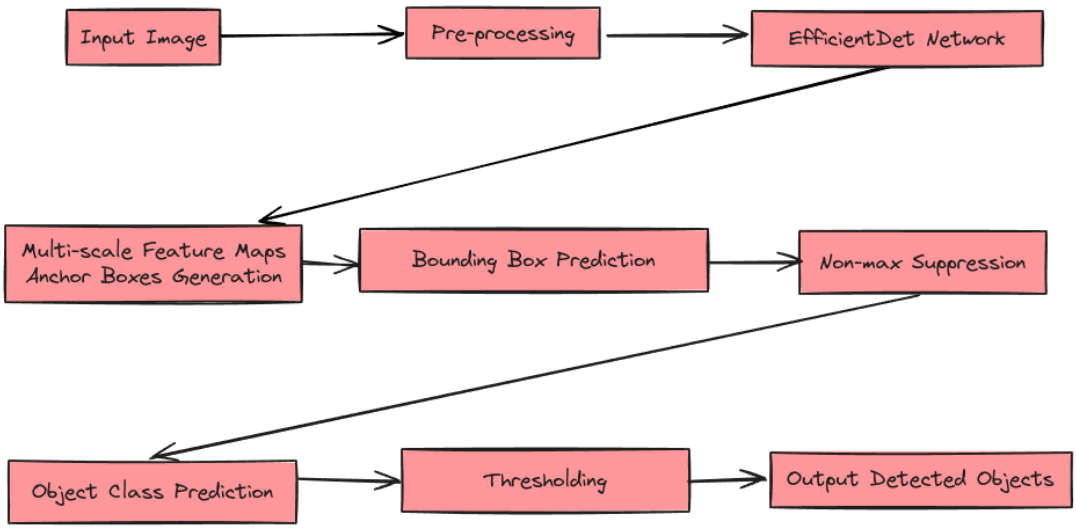

Рисунок 2.9 – Метод EfficientDet

3 ЗАДАЧІ РОЗПІЗНАВАННЯ КОНКРЕТНОЇ ГРУПИ ОБ’ЄКТІВ

3.1 Розпізнавання об'єктів у автономних автомобілях

Проблема розпізнавання об'єктів у контексті автономних автомобілів полягає у розробці та вдосконаленні систем, які можуть надійно і точно ідентифікувати та класифікувати різні об'єкти у довкіллі в реальному часі. Ці об'єкти включають інші транспортні засоби, пішоходів, дорожні знаки, світлофори, і різноманітні перешкоди на дорозі.

Для автономних автомобілів, які покладаються на велику кількість сенсорів, таких як камери, лідари (лазерні радари) та радари, розпізнавання об'єктів є критичним для забезпечення безпеки та ефективності їхньої роботи. Система має бути здатною не лише розпізнавати об'єкти, але й розуміти їхнє положення, швидкість та траєкторію руху, щоб адекватно реагувати на змінні умови дорожнього руху.

Для досягнення високих результатів необхідно реалізувати 4 елементи, що, працюючи разом, скомпонують в систему, яка вирішуватиме поставлені завдання. Потрібно точно ідентифікувати об'єкти навіть в складних умовах, таких як погана погода, варіації освітлення, або коли об'єкти частково закриті. Система повинна обробляти великі обсяги даних від сенсорів у дуже короткі терміни для забезпечення негайної реакції на дорожні ситуації. Система має функціонувати безвідмовно 24/7 у різних дорожніх та погодних умовах. Також адаптуватися до несподіваних або рідкісних дорожніх ситуацій не втрачаючи якості.

Вдосконалення технологій машинного навчання і штучного інтелекту, зокрема глибинного навчання, зіграли ключову роль у прогресі розпізнавання об'єктів для автономних автомобілів, дозволяючи системам краще «розуміти» складні зображення з великою кількістю деталей.

3.2 Вибір датасет

Датасет Udacity Self Driving Car Dataset було створено з метою вдосконалення технологій автономних транспортних засобів.

Цей датасет включає велику кількість анотованих зображень, зібраних з камер автономних автомобілів, які можна використовувати для тренування алгоритмів комп'ютерного зору. Датасет містить 30 000 зображень, знятих в різних дорожніх та погодних умовах.

Вибрана версія датасету зображень з роздільною здатністю 512x512 пікселів, що є важливим для як детального аналізу так і легшого навчання моделей. Кожне зображення має анотації, які включають об'єкти, такі як автомобілі, пішоходи, дорожні знаки, світлофори.

Анотації забезпечують точні положення об'єктів на зображеннях, що дозволяє використовувати ці дані для навчання моделей глибинного навчання. Зображення охоплюють широкий спектр дорожніх сценаріїв, включаючи міські дороги, шосе, різні часи доби та погодні умови.

Вважаю, що даний датасет є чудовим прикладом для навчання різних моделей розпізнавання та порівняння їх між собою.

Датасет містить 97942 міток 11 класів. Також є 1720 зображень без жодної мітки. Баланс класів містить наступне: 64399 міток автомобілів, 10806 людей, 6870 червоних сигналів світлофора, 5465 зелених сигналів світлофора, 3623 грузовики, 2568 світлофорів, 1864 мотоциклістів, 1751 червоних лівих сигналів світлофора, 310 зелених лівих сигналів світлофора, 272 жовтих сигналів світлофорів та 14 жовтих лівих сигналів світлофора [16]. Баланс класів зображено на рисунку 3.1.

Датасет включає велику кількість міток і різноманітність класів, що є хорошою основою для навчання моделей глибокого навчання, таких як Faster R-CNN і YOLO. Однак, існує кілька мінусів. Передусім це нерівномірний баланс класів, що може призвести до того, що модель буде краще розпізнавати об'єкти з класів, які зустрічаються частіше, і гірше – з

рідкісних класів. Також, наявність зображень без міток може ускладнити процес навчання, якщо їх не обробити належним чином. Щодо потенційних проблем, то необхідно звернути увагу на якість анотації та загальну якість зображень, оскільки ці фактори можуть значно вплинути на точність моделі. Крім того, налаштування параметрів моделі вимагає додаткових експериментів, особливо в умовах, коли деякі класи представлені не достатньо. Також для покращення якості навчання можливо використовувати техніки аугментації даних для збільшення різноманіття навчальних прикладів, а також звертати увагу на рідкісні класи, використовуючи вагові коефіцієнти. Також необхідно регулярно перевіряти результати на валідаційному наборі, щоб уникнути перенавчання і оптимізувати модель відповідно до потреб (рисунок 3.1).

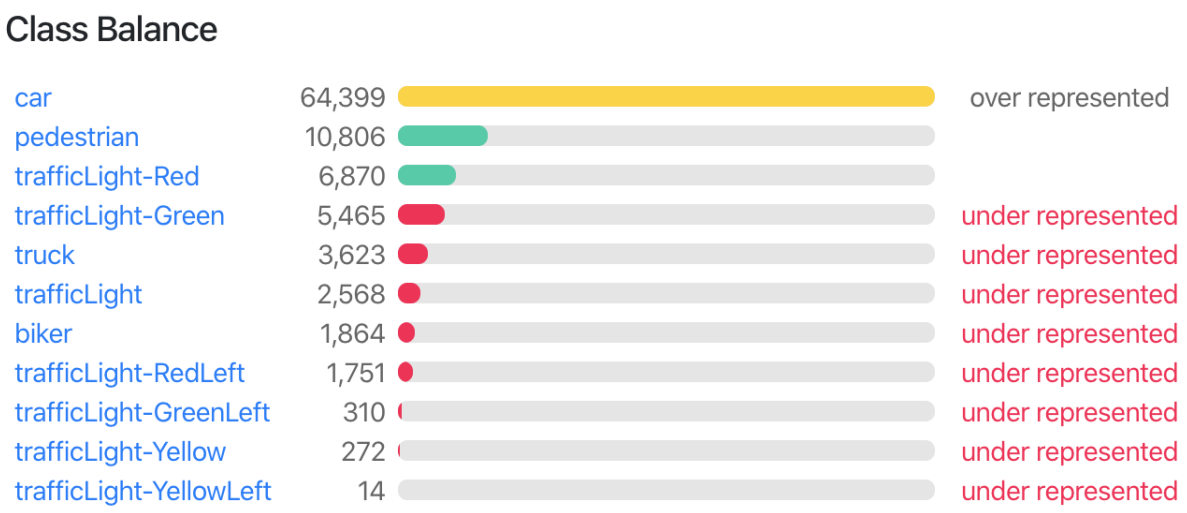

Рисунок 3.1 – Класи датасету та кількість прикладів

3.3 Опис задачі

Вибір відповідного датасету є критично важливим у дослідженні розпізнавання об'єктів за допомогою глибинних нейронних мереж. Для даної роботи був обраний датасет, який відповідає кільком ключовим

критеріям, що забезпечують його ефективність для навчання моделей та оцінки їх продуктивності. Перш за все, датасет повинен бути репрезентативним. Це дозволяє моделі вивчати різноманітні патерни і підходи до класифікації та виявлення об'єктів. Крім того, якість анотацій у датасеті є вирішальним фактором. Анотації повинні бути детальними та точними, щоб моделі могли навчатися з високою точністю.

Великий обсяг даних забезпечує статистичну значущість результатів і допомагає уникнути перенавчання моделей. Використання датасету, який широко застосовується в науковій спільноті, дозволяє порівняти отримані результати з існуючими бенчмарками та провести оцінювання результативності моделей. Це особливо важливо для підтвердження надійності та точності розроблених методів.

Основна задача дослідження полягає у розпізнаванні заданих об'єктів на зображеннях з використанням різних архітектур згорткових нейронних мереж. Це включає виявлення об'єктів, їх класифікацію та локалізацію на зображеннях. У рамках цього дослідження розглядаються кілька підзадач: класифікація об'єктів, яка полягає у визначенні класу об'єкта на зображенні (наприклад, автомобіль, пішохід, світлофор), та локалізація об'єктів, що включає визначення точного положення об'єкта на зображенні за допомогою обмежувальних рамок. Також важливим аспектом є порівняння ефективності різних моделей згорткових нейронних мереж у задачах розпізнавання об'єктів за точністю та швидкістю.

Метою цього дослідження є оцінка та порівняння різних підходів до розпізнавання об'єктів на зображеннях за допомогою сучасних методів глибинного навчання. Зокрема, дослідження зосереджується на вивченні можливостей і обмежень різних архітектур згорткових нейронних мереж, порівнянні точності та швидкості роботи різних моделей, а також визначенні оптимальної моделі для розпізнавання об'єктів у різних умовах та для різних застосувань.

Ідеальний результат цього дослідження передбачає розробку моделі згорткової нейронної мережі, яка забезпечує високу точність виявлення та класифікації об'єктів на зображеннях, мінімізуючи. Крім того, модель повинна мати високу швидкість обробки зображень, що дозволяє її використовувати в реальному часі для задач, таких як автономні автомобілі та системи відеоспостереження. Важливо також, щоб модель була універсальною і адаптувалася до різних типів об'єктів та умов зйомки, зберігаючи свою ефективність та надійність.

Досягнення таких результатів дозволить значно покращити існуючі системи розпізнавання об'єктів та розширити їх застосування в різних галузях, забезпечуючи нові можливості для автоматизації та підвищення ефективності. Це, у свою чергу, сприятиме прогресу в таких сферах, як робототехніка та автономне керування, де точне та швидке розпізнавання об'єктів є критично важливим.

3.4 Нюанси задачі та особливості датасету

У процесі вирішення задачі розпізнавання об'єктів на зображеннях з використанням згорткових нейронних мереж, з'являються певні нюанси, пов'язані з характером задачі та обраного датасету. Основні аспекти включають кількість та якість тренувальних даних, особливості об'єктів на зображеннях та можливість використання синтетичних даних для покращення результатів.

Однією з основних проблем, що може виникнути, є нестача тренувальних даних. Недостатня кількість зображень для навчання моделі може призвести до перенавчання, коли модель демонструє високу точність на тренувальних даних, але не справляється з невідомими зображеннями. Для вирішення цієї проблеми існує кілька підходів.

По-перше, важливо розглянути можливість використання методів аугментації даних. Це включає створення нових тренувальних зразків

шляхом застосування різних змін до існуючих зображень. Це можуть бути обертання, масштабування, зсуви, зміна яскравості та контрастності, віддзеркалення тощо. Використання аугментації дозволяє значно збільшити кількість тренувальних зразків без необхідності збору додаткових даних [17].

По-друге, варто розглянути можливість використання синтетичних даних. Синтетичні дані можуть бути згенеровані за допомогою комп'ютерної графіки або спеціалізованих алгоритмів, таких як генеративно-змагальні мережі. Ці дані можуть імітувати реальні зображення і містити об'єкти, подібні до тих, що зустрічаються в реальному світі. Використання синтетичних даних дозволяє значно збільшити обсяг тренувального набору і забезпечити різноманітність зразків, що сприяє покращенню точності та надійності моделі.

По-третє, важливо звернути увагу на методи попереднього тренування моделей. Використання попередньо натренованих моделей, таких як ResNet, VGG або MobileNet, дозволяє скоротити час та обчислювальні ресурси, необхідні для навчання, а також покращити якість результатів. Такі моделі містять знання, отримані з великих загальних датасетів, таких як ImageNet, і можуть бути адаптовані до конкретної задачі за допомогою додаткового навчання на спеціалізованому датасеті [18].

Таким чином, розглянувши нюанси задачі та особливості обраного датасету, можна використовувати аугментацію даних, синтетичні дані та передтреновані моделі для покращення результатів. Це дозволить забезпечити більш високу точність та надійність розпізнавання об'єктів, навіть за умов обмежених ресурсів.

3.5 Вибір моделей для тестування

Для тестування в рамках даного дослідження були обрані три моделі: Faster R-CNN, YOLOv5 та YOLOv8. Вибір цих моделей ґрунтується на їхніх

характеристиках, популярності та практичній ефективності в задачах розпізнавання об'єктів.

Перша модель, Faster R-CNN, була обрана тому, що вона є однією з перших успішних моделей для розпізнавання об'єктів, яка поєднує високу точність та ефективність. Faster R-CNN використовує регіональні пропозиційні мережі, що значно покращує швидкість обробки порівняно з попередніми моделями, такими як R-CNN та Fast R-CNN. Ця модель встановила новий стандарт для алгоритмів виявлення об'єктів і залишається важливою базовою точкою для порівняння з новішими методами.

Друга модель, YOLOv5, була обрана через свою популярність та широке використання в сучасних рішеннях для розпізнавання об'єктів. YOLO відома своєю високою швидкістю та ефективністю, що дозволяє використовувати її в реальному часі. YOLOv5, як одна з найпопулярніших версій цієї архітектури на сьогоднішній день, має значну підтримку спільноти та численні вдосконалення, які роблять її ідеальною для практичного застосування.

Третя модель, YOLOv8, була обрана тому, що вона є найновішою версією в сімействі YOLO, яка включає додаткові покращення та оптимізації. YOLOv8 продовжує розвивати успіх попередніх версій, пропонуючи підвищену точність та ефективність. Ця модель враховує новітні дослідження та розробки в галузі глибинного навчання, що робить її перспективною для використання в різних задачах розпізнавання об'єктів.

Моделі Faster R-CNN, YOLOv5 та YOLOv8 оптимізовані для ефективного використання обчислювальних ресурсів, що дозволяє проводити навчання та тестування навіть на менш потужних системах. Це робить їх підходящими для досліджень в умовах обмежених ресурсів, забезпечуючи при цьому високу продуктивність і точність. Це дозволяє комплексно оцінити ефективність різних підходів до розпізнавання об'єктів та зробити обґрунтовані висновки щодо їх застосування у практичних задачах [13].

4 ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ

4.1 Faster R-CNN

Для навчання Faster R-CNN моделі використовуватимемо датасет у форматі COCO.

Формат COCO складається з п'яти розділів інформації, які надають інформацію для всього набору даних. COCO має кілька типів анотацій: для виявлення об'єктів, виявлення ключових точок, сегментації об'єктів, паноптичної сегментації, денсепозиції та підписів до зображень. Анотації зберігаються у форматі JSON.

Дані для розпізнавання об’єктів включають наступні поля:

– info – загальна інформація про набір даних;

– licenses – інформація про ліцензії для зображень у наборі даних;

– images – список зображень у наборі даних;

– annotations – список анотацій (включно з рамками), які присутні на всіх зображеннях у наборі даних;

– categories – список категорій міток.

Для зручного користування даним форматом даних, використаємо засіб CocoDetection з Pytorch, що дозволяєш легко перетвори дані в засіб для використання. Застосування CocoDetection зображено на рисунку 4.1.

```
dataset = CocoDetection(
    root=ROOT_DATASET_IMAGES_PATH,
    annFile=ROOT_DATASET_IMAGES_JSON_PATH,
    transform=None
)
```

Рисунок 4.1 – Використання CocoDetection для підготовки датасету

Наступним кроком необхідно розділити датасет на навчальну та валідаційну вибірки. Це важливий крок в процесі підготовки даних для моделей машинного навчання, оскільки забезпечує можливість перевірити ефективність моделі на даних, не використаних під час тренування.

Спочатку визначаємо розмір навчальної вибірки, який становить 80% від загальної кількості даних у датасеті. Таке розділення дозволяє забезпечити достатню кількість даних для навчання моделі, при цьому залишаючи частину даних для валідації.

В нейронних мережах, особливо в задачах обробки зображень як Faster R-CNN, backbone використовується для ефективного витягування ознак із зображень. Це дозволяє моделі краще "розуміти" вхідні дані на ранніх етапах обробки, що є основою для подальших етапів, таких як локалізація об'єктів та їх класифікація. Зазвичай, backbone є попередньо навченою мережею, такою як VGG або ResNet, яка вже вміє визначати базові та складні ознаки в зображеннях.

Використання такого backbone замість навчання мережі з нуля має декілька ключових переваг. Перше, це значно покращує точність моделі, оскільки backbone вже навчений розпізнавати широкий спектр ознак на різних наборах даних, таких як ImageNet. Це допомагає моделі краще справлятися із складними задачами розпізнавання в нових зображеннях.

По-друге, використання попередньо навченого backbone спрощує процес навчання, оскільки вам не потрібно навчати всі аспекти моделі з нуля, що може бути дуже ресурсомістким і часозатратним. Замість цього, ви можете сконцентрувати зусилля на налаштуванні моделі для конкретних задач, використовуючи вже доступні ознаки.

По-третє, використання попередньо навченого backbone може допомогти покращити результативність моделі в реальному часі, що є особливо важливим для застосувань, де час відгуку є критичним, таких як розпізнавання об'єктів в автономних транспортних засобах.

Використаємо ResNet18 як попередньо навчену модель для нашої мережі. ResNet18 має меншу кількість параметрів та менш складну архітектуру в порівнянні з більшими мережами, такими як ResNet34, ResNet50 або ResNet101. Це означає, що вона вимагає менше обчислювальних ресурсів для тренування та виконання. Це робить ResNet18 ідеальною для ситуацій, де потрібна висока швидкість розпізнавання, наприклад в мобільних додатках або вбудованих системах. Використання ResNet18 зображено на рисунку 4.2.

Хоча ResNet18 не може забезпечити таку ж точність, як більші моделі на складних наборах даних, вона все ще забезпечує достатню точність для багатьох застосувань. Це робить її хорошим компромісом між ефективністю та точністю, особливо коли висока точність не є абсолютно критичною.

```
backbone = resnet_fpn_backbone('resnet18', pretrained=True)
model = FasterRCNN(backbone, num_classes=NUM_CLASSES).to(device)
```

Рисунок 4.2 – Використання ResNet18 для навчання мережі

Для навчання використаємо алгоритм стохастичного градієнтного спуску (SGD). Оптимізатори використовуються для оновлення ваг нейронної мережі на основі градієнтів, щоб мінімізувати функцію втрат. Випробувавши декілька варіантів оптимізації, цей конкретний набір параметрів оптимізатора SGD виявився найкращим варіантом. Застосування стохастичного градієнтного спуску зображено на рисунку 4.3.

Швидкість навчання = 0.005, яка визначає, наскільки сильно кожне оновлення змінює ваги. Значення 0.005 є досить типовим вибором, що забезпечує помірний крок оновлення. Коефіцієнт інерції = 0.9, який допомагає оптимізатору «рухатися» через невеликі мінімуми, ігноруючи шум. Значення 0.9 дозволяє значно збільшити вплив попередніх градієнтів на оновлення параметрів, що сприяє більш гладкому і швидкому збігу.

Коефіцієнт згасання ваг = 0.0005, який використовується для регуляції та боротьби з перенавчанням, шляхом накладання штрафів на великі ваги.

```
params = [p for p in model.parameters() if p.requires_grad]
optimizer = SGD(params, lr=0.005, momentum=0.9, weight_decay=0.0005)
```

Рисунок 4.3 – Використання стохастичного градієнтного спуску

Далі визначаємо кількість епох навчання, що вказує, скільки разів алгоритм пройде через весь навчальний набір даних. В межах кожної епохи модель переводиться в режим тренування, що важливо для коректного обчислення градієнтів та подальшого оновлення ваг. Оптимізатор налаштовується на нуль перед кожним проходом, щоб уникнути накопичення градієнтів з попередніх кроків. Після обчислення втрат моделлю втрати сумуються, і проводиться зворотне поширення помилки, що дозволяє оновлювати ваги моделі на основі обчислених градієнтів. Після завершення обходу всіх батчів у навчальному наборі, проводиться валідація моделі на валідаційному наборі даних. Код навчання моделі зображено на рисунку 4.4.

```
for epoch in range(num_epochs):
    model.train()
    train_losses = []

    with tqdm(total=total_batches_train, desc=f"Epoch {epoch + 1}/{num_epochs} (Train)", unit="batch") as pbar:
        for batch_counter, (images, targets) in enumerate(train_loader):
            images = [img.to(device) for img in images]
            targets = [{k: v.to(device) if isinstance(v, torch.Tensor) else v for k, v in t.items()} for t in targets]

            optimizer.zero_grad()
            loss_dict = model(images, targets)
            loss = sum(loss for loss in loss_dict.values())
            loss.backward()
            optimizer.step()

            train_losses.append(loss.item())

            pbar.set_postfix(loss=f"{loss.item():.4f}", batches=batch_counter)

    validate(model, val_loader, device)
```

Рисунок 4.4 – Навчання моделі

Оцінку точності Faster R-CNN зображено на рисунку 4.5.

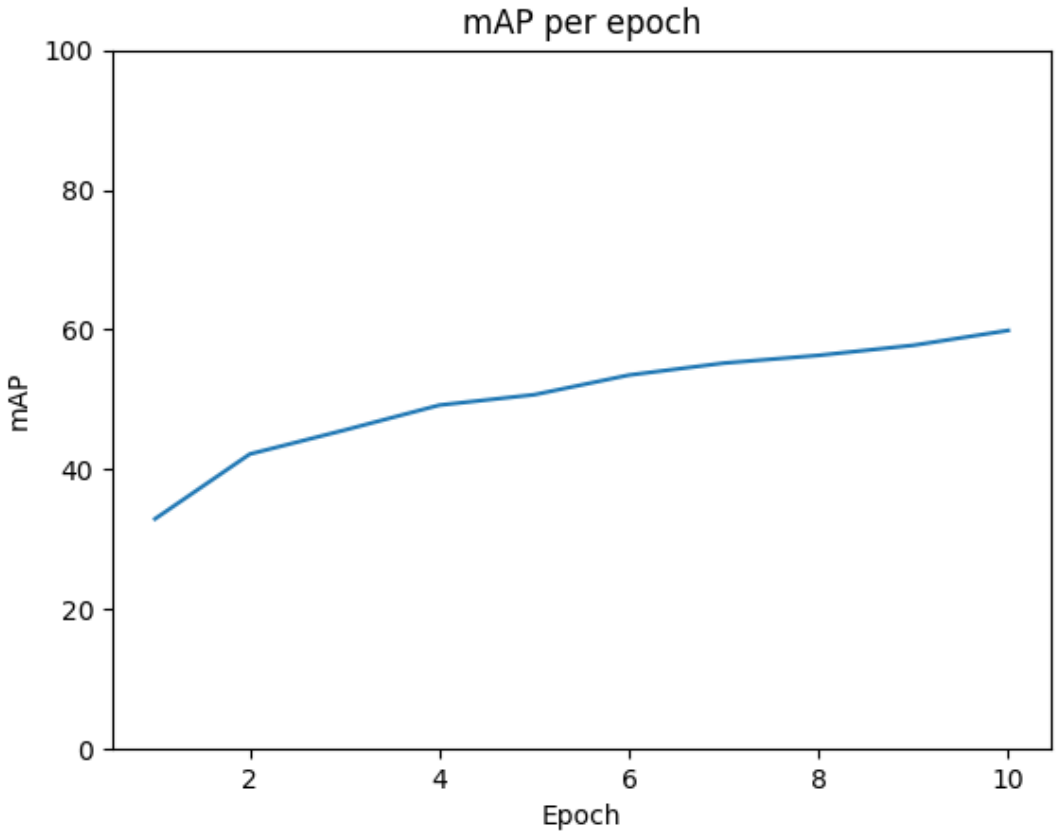

Рисунок 4.5 – Оцінка точності моделі на кожній епосі

Інтегровано модель для автоматичного розпізнавання об'єктів на відео, щоб продемонструвати ідентифікацію і класифікацію об'єктів в реальному часі. Використання GPU дозволило забезпечити більш швидку обробку кадрів, але на моїй малопотужній відеокарті частота кадрів була зазвичай 6–7 кадрів на секунду. Ось декілька скріншотів з відео, що демонструють можливості моделі. Модель доволі успішно розпізнає об'єкти, позначаючи їх клас і точність розпізнавання. Розпізнавання об’єктів зображено на рисунках 4.6, 4.7, 4.8, 4.9.

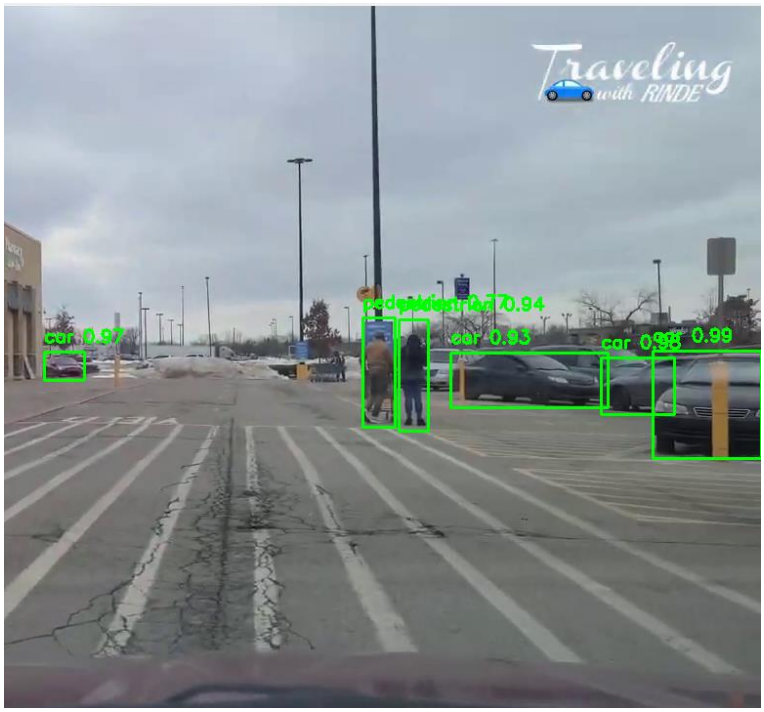

Рисунок 4.6 – Приклад розпізнавання моделі Faster R-CNN

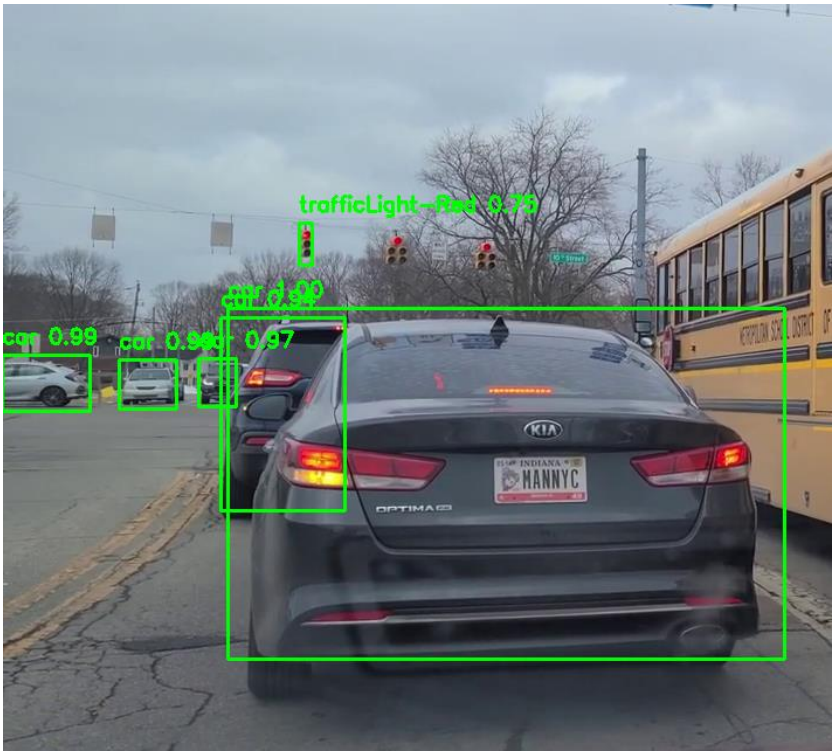

Рисунок 4.7 – Приклад розпізнавання моделі Faster R-CNN

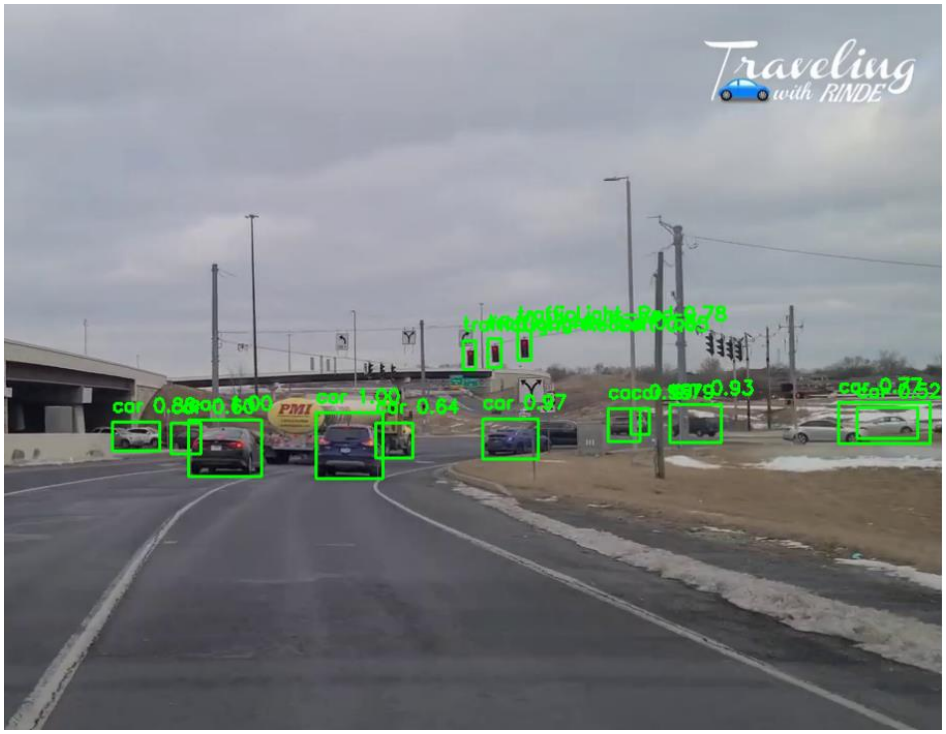

Рисунок 4.8 – Приклад розпізнавання моделі Faster R-CNN

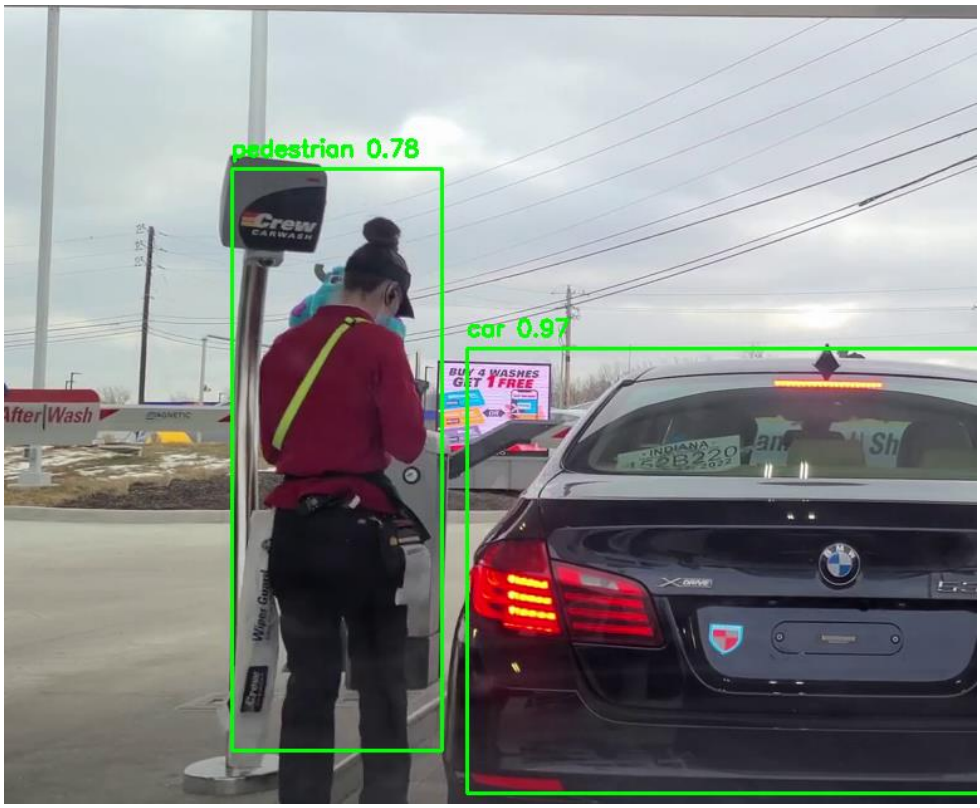

Рисунок 4.9 – Приклад розпізнавання моделі Faster R-CNN

4.2 YOLOv5

YOLOv5 – п'яте покоління архітектури «You Only Look Once» для розпізнавання об'єктів. Воно стало вельми популярним завдяки своїй ефективності та легкості використання. На відміну від складніших моделей, таких як Faster R-CNN та SSD, YOLOv5 вирізняється своєю швидкістю та здатністю до швидкого навчання з мінімальними вимогами до кодування.

Дані датасету складаються із зображень та пов'язаних з ними міток, які включають обмежувальні рамки та мітки класів для об'єктів на зображеннях. Для навчання YOLO датасет необхідно поділити на тренувальний, валідаційний та тестовий набори.

Запуск тренування в YOLOv5 вимагає лише однієї командної строки, що автоматично налаштовує всі необхідні параметри, включаючи оптимізацію та збереження моделі. YOLOv5 має вбудовані функції для аугментації даних, розрахунку метрик та візуалізації, що зменшує кількість коду, необхідного для розробки повноцінного рішення.

Процес навчання моделі YOLOv5 є прикладом її простоти та ефективності. Запуск команди для навчання зображено на рисунку 4.10.

```
!python train.py --img 512 --batch 16 --epochs 10 --data {dataset_yaml_path} --weights yolov5s.pt
```

Рисунок 4.10 – Команда для тренування YOLOv5

У виборі конкретних моделей для завдань обробки зображень і відео важливе значення має баланс між швидкістю, точністю та ресурсоємністю обчислень. Моє рішення використовувати YOLOv5s для даної моделі та ResNet18 як основу для Faster R-CNN у іншому відображає прагнення забезпечити високу ефективність обробки з обмеженими ресурсами. ResNet18 та YOLOv5s обидва орієнтовані на мінімізацію обчислювальних вимог, не втрачаючи при цьому в якості розпізнавання. Ці моделі ідеально

підходять для застосувань, де потрібне хороше балансування між швидкістю і точністю.

Оцінку точності YOLOv5 на кожній епосі зображено на рисунку 4.11.

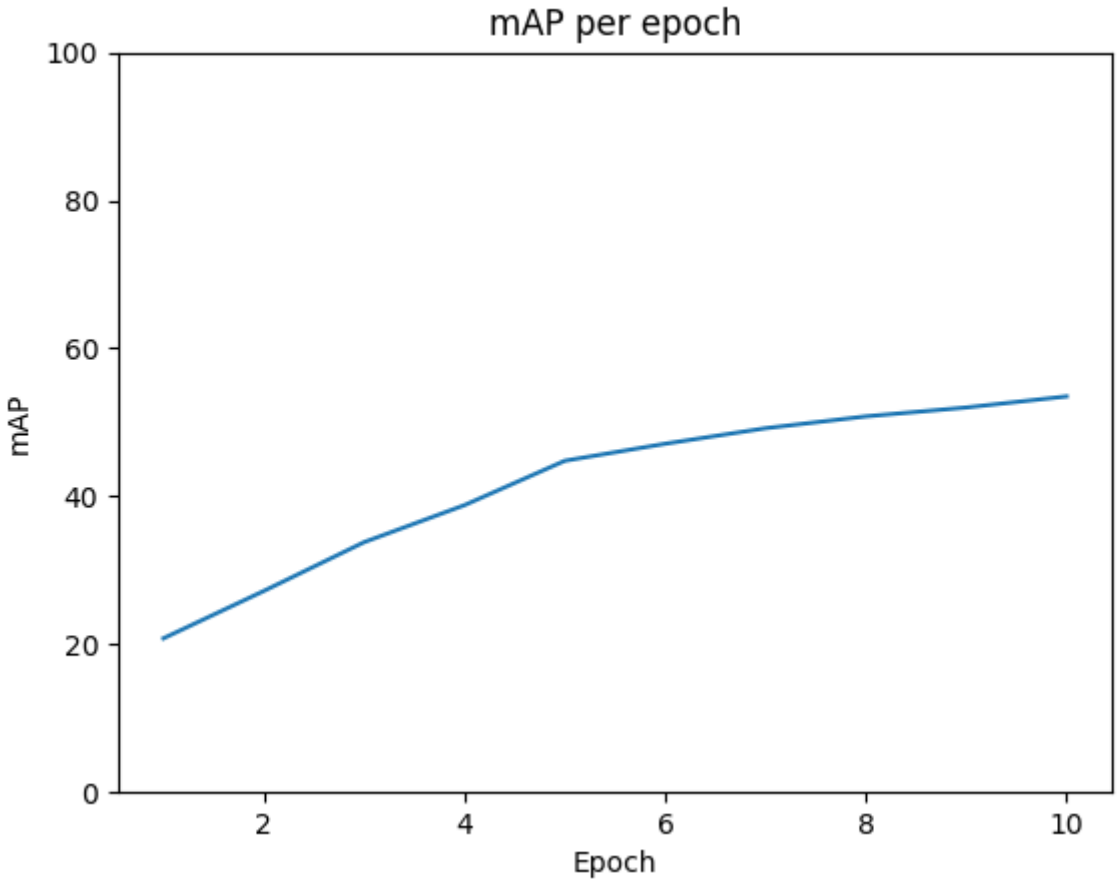

Рисунок 4.11 – Оцінка точності моделі на кожній епосі

Інтегровано модель для автоматичного розпізнавання об'єктів на відео, щоб продемонструвати ідентифікацію і класифікацію об'єктів в реальному часі. Використання GPU дозволило забезпечити більш швидку обробку кадрів, і навіть на малопотужній відеокарта частота була 60–65 кадрів на секунду, що є досить високим у порівнянні з Faster R-CNN.

Приклади розпізнавання зображено на рисунках 4.12, 4.13, 4.14, 4.15. Хоча варто зазначити, що дана модель розпізнає дещо гірше за попередню, але швидкість розпізнавання вражає і є суттєвою перевагою у виборі моделі з можливістю донавчання.

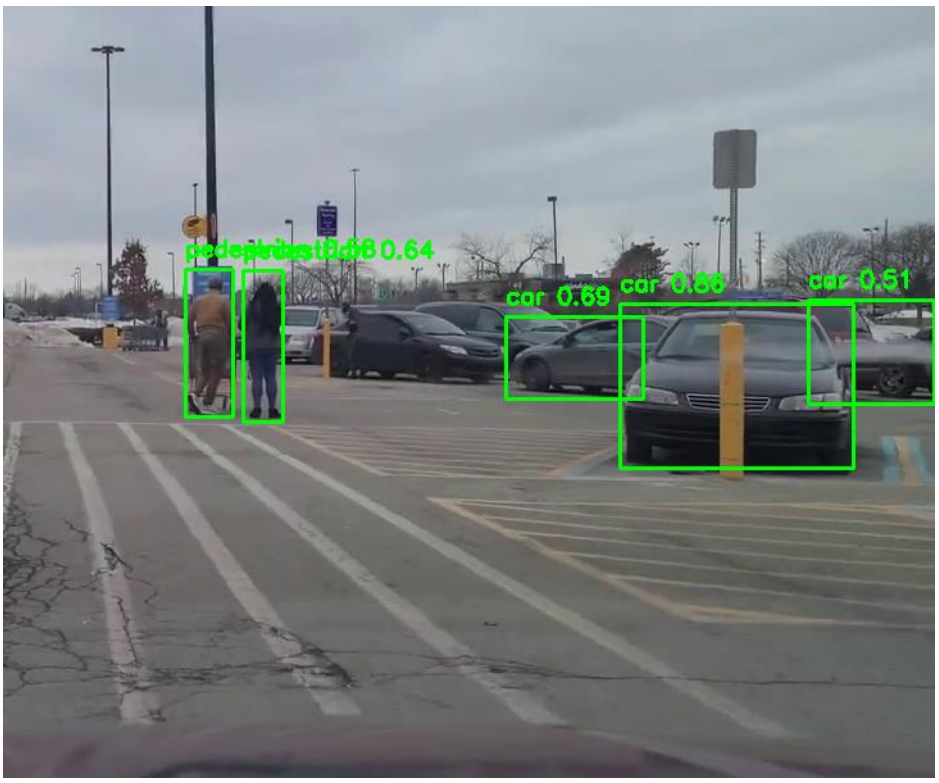

Рисунок 4.12 – Приклад розпізнавання моделі YOLOv5

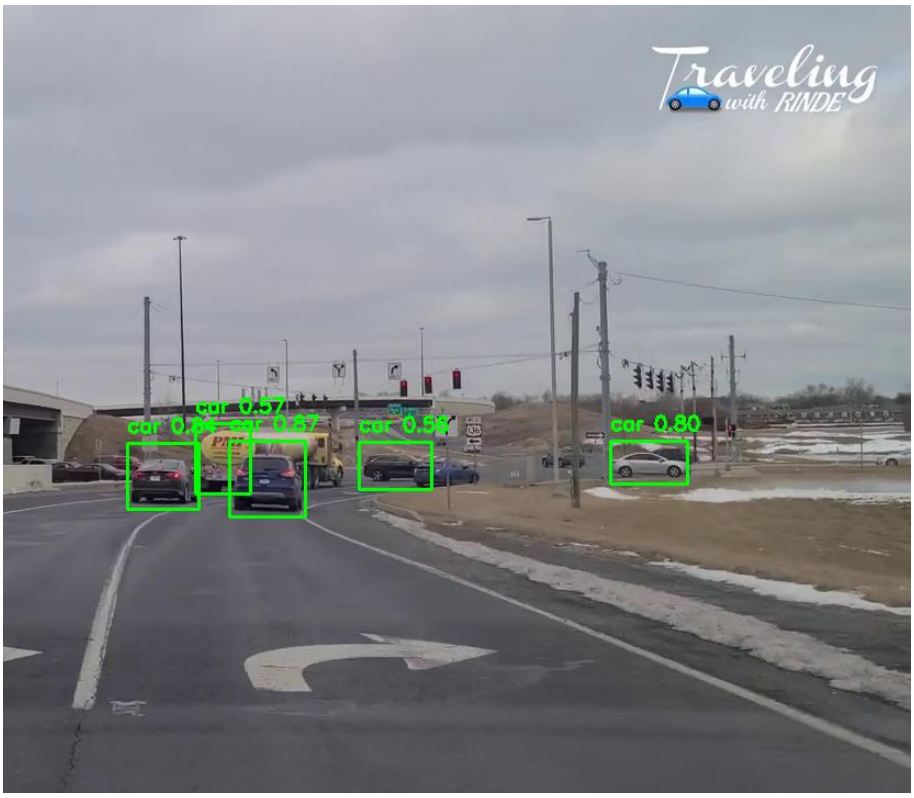

Рисунок 4.13 – Приклад розпізнавання моделі YOLOv5

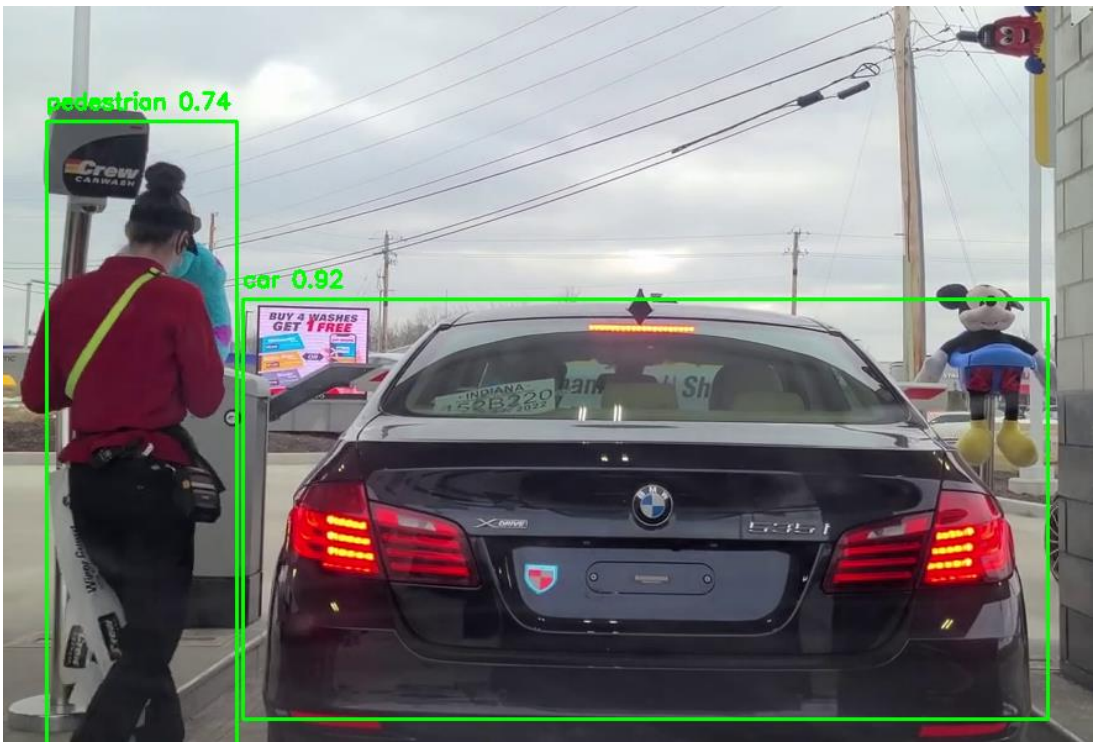

Рисунок 4.14 – Приклад розпізнавання моделі YOLOv5

Рисунок 4.15 – Приклад розпізнавання моделі YOLOv5

4.3 YOLOv8

YOLOv8 – останнє покоління архітектури "You Only Look Once" для розпізнавання об'єктів, яке пропонує значні покращення у порівнянні з попередніми версіями, зокрема YOLOv5.

На відміну від YOLOv5, YOLOv8 вводить покращення у механізми обробки зображень та використання згорткових блоків, що дозволяє ще більше знизити час обробки при збереженні або навіть покращенні точності детектування об'єктів на різних масштабах. Ці технічні удосконалення роблять YOLOv8 особливо привабливим для застосувань, де потрібна максимальна швидкість обробки в реальному часі без компромісів щодо точності. Процес навчання моделі YOLOv8 схожий на YOLOv5, є простішим і швидшим завдяки оптимізації коду та вбудованим функціям, які мінімізують необхідність ручного кодування.

Оцінку точності YOLOv5 на кожній епосі зображено на рисунку 4.16.

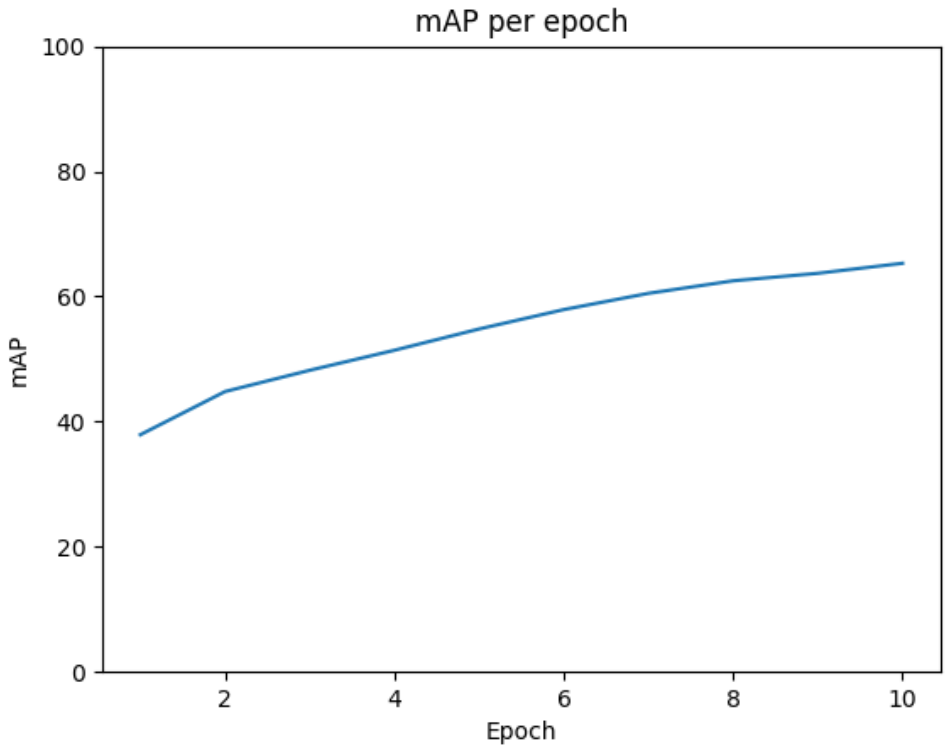

Рисунок 4.16 – Оцінка точності моделі на кожній епосі

Ще однією ключовою перевагою YOLOv8 є його здатність ефективно працювати з великими обсягами даних. Це дозволяє використовувати модель у вимогливих до обчислень середовищах, таких як розподілені обчислення чи реалізація на вбудованих пристроях.

У цьому експерименті використовувався невеликий датасет та проводилось навчання протягом 10 епох. Це може вплинути на загальну ефективність моделі, оскільки обмежений обсяг даних та кількість епох можуть призвести до недостатнього навчання моделі.

На малопотужній відеокарта частота була 65–70 кадрів на секунду, що є досить високим у порівнянні з Faster R-CNN і трішки вищою ніж у YOLOv5. Хоча точність моделі на валідаційному датасеті показує найвищий результат з трьох моделей розпізнавання залишається на рівні YOLOv5 і гірше ніж Faster R-CNN. Приклади розпізнавання зображено на рисунках 4.17, 4.18, 4.19.

Рисунок 4.17 – Приклад розпізнавання моделі YOLOv8

Рисунок 4.18 – Приклад розпізнавання моделі YOLOv8

Рисунок 4.19 – Приклад розпізнавання моделі YOLOv8

4.4 Порівняння точності моделей розпізнавання

Рисунок 4.20 зображає зміну середньої точності (mAP) в залежності від епох для трьох різних моделей: Faster R-CNN, YOLOv5, та YOLOv8. На осі x відкладено кількість епох (від 1 до 10), а на осі y – значення mAP у відсотках (від 0 до 100). Лінії різного кольору представляють кожну модель: синя лінія для Faster R-CNN, помаранчева для YOLOv5, та зелена для YOLOv8.

На графіку видно, що YOLOv8 має найвищі значення mAP на всіх етапах навчання, починаючи з високого стартового значення і продовжуючи зростати до приблизно 65% на десятій епосі. Faster R-CNN показує поступове зростання, досягаючи близько 60% на десятій епосі. YOLOv5 починає з найнижчого значення серед трьох моделей, але також демонструє стійке зростання, досягаючи близько 53% на десятій епосі.

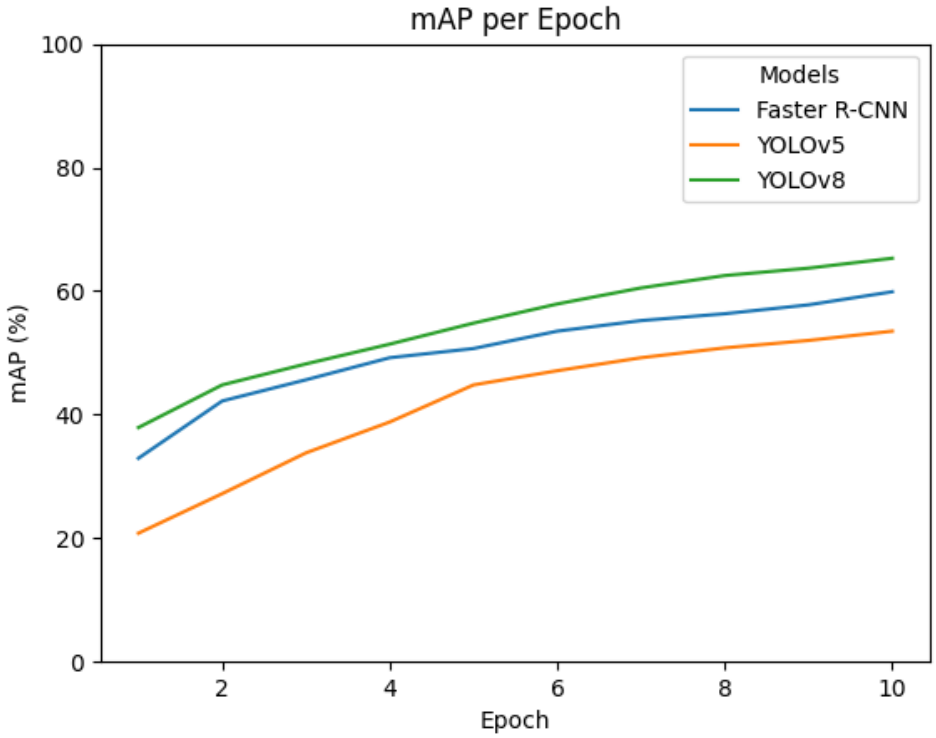

Рисунок 4.20 – Порівняння точності трьох моделей

Рисунок 4.21 являє собою гістограму, яка порівнює частоту кадрів в секунду трьох різних моделей виявлення об'єктів: Faster R-CNN, YOLOv5 та YOLOv8. На осі Y відкладено значення FPS у діапазоні від 0 до 100, а на осі X – назви трьох моделей. Смуги світло-блакитного кольору та різної довжини вказують на різні значення FPS для кожної моделі. Faster R-CNN має найнижчий показник FPS, досягаючи лише 6–8 FPS. YOLOv5 працює значно краще, досягаючи близько 60–65 FPS. YOLOv8 має найвищу продуктивність, з FPS близько 65–70. Ця діаграма підкреслює вищу швидкість моделей YOLO, особливо YOLOv8, порівняно з Faster R-CNN.

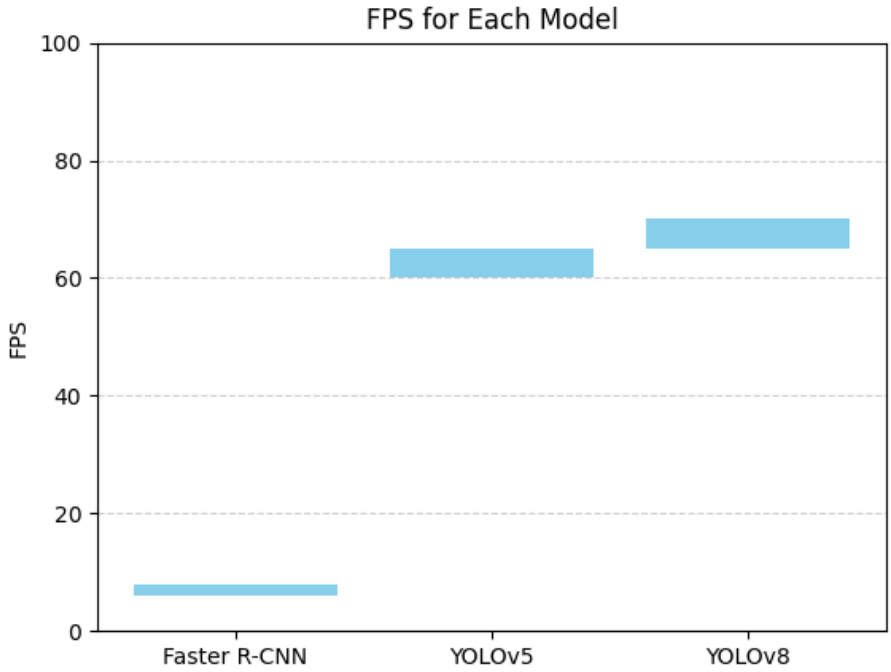

Рисунок 4.21 – Порівняння швидкості розпізнавання

Рисунок 4.22 демонструє порівняння часу навчання епохи для трьох різних моделей. На осі у відкладено час навчання в секундах, починаючи від 0 до 1000 секунд. Стовпці різних кольорів представляють кожну модель: червоний для Faster R-CNN, синій для YOLOv5, та зелений для YOLOv8.

Faster R-CNN має найвищий час навчання на епоху, що становить 960 секунд. Це значно більше, ніж у YOLOv5 та YOLOv8, які мають 115 і 119 секунд відповідно. Така різниця в часі навчання може бути пояснена складністю архітектури та кількістю обчислювальних операцій, необхідних для навчання Faster R-CNN порівняно з YOLO моделями.

Тривалий час навчання Faster R-CNN може впливати на його практичне застосування, особливо в умовах, де обмежений час на навчання моделей. З іншого боку, YOLOv5 та YOLOv8, з їх швидшим часом навчання, можуть бути кращим вибором для завдань, що вимагають швидкого розгортання моделей. Це також може бути важливим фактором для середовищ з обмеженими обчислювальними ресурсами, де оптимізація часу навчання має ключове значення.

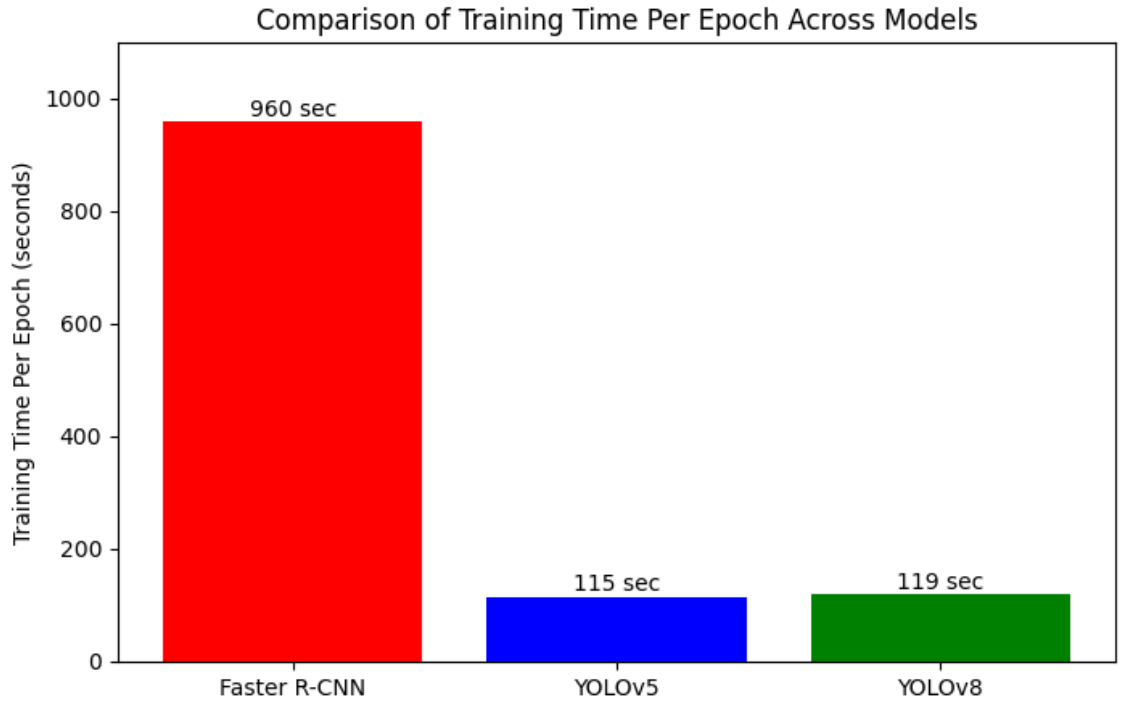

Рисунок 4.22 – Порівняння швидкості навчання

Незважаючи на те, що графіки показують кращі результати YOLOv8 у порівнянні з Faster R-CNN в аспекті швидкості і середньої точності, на

реальних відео моделі YOLOv8 I YOLOv5 показують себе гірше. Для цього може бути декілька пояснень.

YOLOv8 і YOLOv5 можуть мати високу точність на навчальних наборах даних, але їхня здатність до генералізації може бути меншою на реальних відео, які можуть містити різноманітні умови освітлення, ракурси і фонові об'єкти. Faster R-CNN, можливо, краще узагальнює на цих більш складних і різноманітних даних.

YOLO моделі можуть мати труднощі з розпізнаванням дрібних об'єктів або об'єктів з низькою контрастністю в кадрі. Faster R-CNN зазвичай краще обробляє дрібні об'єкти через свою архітектуру, яка використовує регіональні пропозиції для визначення потенційних об'єктів перед їх класифікацією.

Також можливо, що моделі YOLOv5 і YOLOv8 не були достатньо добре треновані або налаштовані на специфічних даних, що використовуються в реальному відео. Параметри, такі як розмір навчального набору, кількість епох або параметри аугментації даних, можуть вплинути краще на кінцевий результат.

Faster R-CNN може мати перевагу у випадках, де є дисбаланс у класах об'єктів. Якщо певні класи представлені в навчальних даних недостатньо, Faster R-CNN може краще справлятися завдяки своїй архітектурі і методам, які використовуються для обробки об'єктів.

Реальні відео можуть містити шум, артефакти компресії та інші дефекти, які впливають на точність розпізнавання. Faster R-CNN може бути менш чутливим до таких артефактів завдяки своїй складнішій і більш детальній архітектурі.

Таким чином, хоча YOLO моделі демонструють чудові результати, їх ефективність може знижуватися на реальних відео через наведені причини. Faster R-CNN, попри свою нижчу швидкість, може бути краще налаштований для розпізнавання об'єктів у складних і непередбачуваних умовах реальних відео.

4.5 Пропозиції щодо вдосконалення розпізнавання об’єктів

Розпізнавання об'єктів на зображеннях залишається однією з найважливіших задач у галузі комп'ютерного зору. Незважаючи на значні досягнення, існує багато напрямків для подальшого вдосконалення моделей і методів.

Поєднання різних архітектур може стати ефективним способом покращення результатів. Зокрема, використання гібридних моделей, які об'єднують переваги кількох підходів, може забезпечити вищу точність та ефективність. Наприклад, поєднання регіональних пропозиційних мереж (RPN) Faster R-CNN з одноетапними детекторами, такими як YOLO, може дозволити скористатися перевагами обох методів. Faster R-CNN відомий своєю високою точністю, тоді як YOLO забезпечує швидкість обробки в реальному часі. Комбінація цих моделей може дати значний приріст результативності.

Існуючі моделі можуть бути модернізовані різними способами для покращення їхньої продуктивності. Один з підходів полягає в додаванні більшої кількості шарів до моделей, що дозволяє їм вивчати складніші та глибші ознаки зображень. Проте, збільшення кількості шарів може призвести до проблеми перенавчання та зростання обчислювальної складності. Для уникнення цих проблем можна використовувати методи регуляризації, такі як Dropout, L2-регуляризація, та нормалізація пакетів [19].

Інший підхід до модернізації полягає у використанні більш ефективних алгоритмів оптимізації та покращення архітектури нейронних мереж. Наприклад, сучасні дослідження показали, що використання AdamW замість стандартного Adam може забезпечити кращу ефективність і стабільність навчання моделей.

Трансформери стали дуже популярними в задачах обробки природної мови, але їхнє застосування в комп'ютерному зорі також показало

обнадійливі результати. Vision Transformers використовують механізм самоуваги для обробки зображень, розбиваючи їх на невеликі батчі і обробляючи їх як послідовності слів в NLP. Це дозволяє моделі вивчати глобальні взаємозв'язки в зображеннях і досягати високої точності в задачах класифікації, сегментації та виявлення об'єктів [20].

Регулярне оцінювання та оптимізація моделей є важливими для забезпечення їхньої продуктивності. Використання валідаційного набору даних дозволяє постійно перевіряти результати і уникати перенавчання. Крім того, оптимізація гіперпараметрів, таких як швидкість навчання, розмір пакету та кількість епох, може значно вплинути на продуктивність моделі. Використання методів автоматичного налаштування гіперпараметрів, таких як Grid Search або Bayesian Optimization, може допомогти знайти оптимальні параметри для конкретної задачі.

Мета-навчання є ще одним перспективним підходом, який може бути використаний для покращення результатів розпізнавання об'єктів. Мета-навчання дозволяє моделі вивчати, як вчитися новим задачам швидше та ефективніше. Це може бути особливо корисним при обмеженій кількості даних або при необхідності швидкого адаптування до нових умов.

Fine-Tuning натренованих моделей на спеціалізованому датасеті дозволяє зберегти знання, отримані з великого загального датасету, і адаптувати модель до конкретної задачі. Це може значно покращити точність та надійність моделі.

Інтеграція додаткових сенсорних даних, таких як дані з лідара, радара або інших сенсорів, може значно покращити розпізнавання об'єктів. Комбінування інформації з різних джерел дозволяє моделі отримувати більш повну картину навколишнього середовища і підвищувати точність виявлення та класифікації об'єктів.

ВИСНОВКИ

У ході аналізу розпізнавання об'єктів на зображеннях було встановлено, що ця технологія має великий потенціал у багатьох галузях індустрії та науки. Методи, такі як R-CNN, Fast R-CNN, Faster R-CNN, YOLO, SSD, EfficientDet, забезпечують різноманітні можливості для розв'язання завдань розпізнавання об'єктів з різною швидкістю та точністю.

У результаті порівняльного аналізу різних методів було встановлено, що кожен з них має свої переваги та обмеження. Наприклад, методи типу R-CNN мають високу точність розпізнавання, але потребують значних обчислювальних ресурсів та мають низьку швидкість обробки. У свою чергу, YOLO та SSD відрізняються високою швидкістю обробки та низькими вимогами до обчислювальних ресурсів, проте можуть бути менш точними в деяких випадках.

Також було досліджено проблему розпізнавання об'єктів в автономних автомобілях, зокрема через навчання нейронних мереж. З огляду на виклики цієї галузі, було проаналізовано та порівняно три моделі глибинного навчання: Faster R-CNN, YOLOv5 та YOLOv8.

Faster R-CNN показала високу точність розпізнавання об'єктів, особливо на реальних відео, завдяки своїй здатності краще обробляти дрібні об'єкти та складні сцени. YOLOv8, демонструючи найвищі значення середньої точності (mAP) на навчальних наборах даних, перевершувала Faster R-CNN та YOLOv5. Однак YOLOv5 та YOLOv8 значно перевершують Faster R-CNN за швидкістю обробки кадрів, досягаючи 60–70 FPS навіть на малопотужних відеокартах. Це робить їх більш придатними для застосувань, де критично важлива швидкість обробки в реальному часі. Faster R-CNN демонструвала найнижчу швидкість обробки, обробляючи лише 6–8 FPS, що може обмежувати її використання в реальних умовах, де потрібна висока швидкість.

Також Faster R-CNN потребує значно більше часу на навчання (близько 960 секунд на епоху), у порівнянні з YOLOv5 та YOLOv8 (115 і 119 секунд на епоху відповідно). Це може обмежувати її застосування в умовах обмеженого часу або ресурсів. Моделі YOLOv8 та YOLOv5, попри свої високі результати на навчальних наборах даних, виявилися менш точними на реальних відео, що може бути пов'язано з їх меншою здатністю до генералізації на різноманітні умови. Faster R-CNN краще справлялася з розпізнаванням об'єктів на реальних відео, що підтверджує її здатність ефективно працювати в умовах реального світу, незважаючи на низьку швидкість обробки.

Вибір моделі залежить від конкретних вимог до системи автономного автомобіля, де баланс між точністю та швидкістю обробки відіграє ключову роль. Моделі YOLOv5 та YOLOv8 пропонують швидку обробку кадрів, тоді як Faster R-CNN забезпечує високу точність розпізнавання, що є важливим для безпечного функціонування автономних транспортних засобів.

Важливим є те, що подальший розвиток методів розпізнавання об'єктів на зображеннях є ключовим напрямом для вирішення різноманітних завдань в різних сферах. Інновації в алгоритмах та архітектурах нейронних мереж, а також поєднання різних методів, можуть покращити якість розпізнавання та розширити області застосування цієї технології.

Отже, результати дослідження підтверджують великий потенціал у використанні розпізнавання об'єктів на зображеннях для різних завдань, що вимагають автоматичної обробки та аналізу великої кількості візуальних даних.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. A. Brownlee, J. Deep Learning for Computer Vision: Image Classification, Object Detection, and Face Recognition in Python. 2020.

2. Heaton J. Artificial Intelligence for Humans, Volume 3: Deep Learning and Neural Networks. Createspace Independent Publishing Platform, 2015. 374 p.

3. Aggarwal C. C. Neural Networks and Deep Learning: A Textbook. Springer, 2019. 520 p.

4. Munir. Accelerators for Convolutional Neural Networks. Wiley & Sons, Limited, John, 2023.

5. Weidman S. Deep Learning from Scratch: Building with Python from First Principles. O'Reilly Media, Incorporated, 2019. 250 p.

6. Generative Adversarial Learning: Architectures and Applications / ed. by R. Razavi-Far et al. Cham: Springer International Publishing, 2022. URL: https://doi.org/10.1007/978-3-030-91390-8 (date of access: 10.04.2024).

7. Montgomery Mrinal Kanti Bhowmik. Computer Vision. Object Detection in Adversarial Vision. CRC PRESS, 2024.

8. Elgendy M. Deep Learning for Vision Systems. Manning Publications Co. LLC, 2020.

9. Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun - Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017.

10. Ren S., He K., Girshick R., & Sun, J. Faster R-CNN: towards real-time object detection with region proposal networks. arXiv preprint arXiv:1506.01497v3. 2016

11. Terven J., Córdova-Esparza D.-M., Romero-González J.-A. A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. Machine Learning and Knowledge Extraction. 2023. Vol. 5, no. 4. P. 1680–1716. URL: https://doi.org/10.3390/make5040083 (date of access: 01.05.2024).

12. Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg – SSD: Single Shot MultiBox Detector. In Proceedings of the European Conference on Computer Vision, 2016.

13. Atienza R. Advanced Deep Learning with TensorFlow 2 and Keras: Apply DL, GANs, VAEs, Deep RL, Unsupervised Learning, Object Detection and Segmentation, and More, 2nd Edition. Packt Publishing, Limited, 2020. 512 p.

14. Tan M., Pang R., Le Q. V. EfficientDet: Scalable and Efficient Object Detection. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020. 2020. URL: https://doi.org/10.1109/cvpr42600.2020.01079 (date of access: 11.05.2024).

15. Jain S. DeepSeaNet: Improving Underwater Object Detection using EfficientDet. arXiv preprint arXiv:2306.06075. 2023.

16. Udacity Self Driving Car Object Detection Dataset. Roboflow. URL: https://public.roboflow.com/object-detection/self-driving-car (date of access: 19.05.2024).

17. Krizhevsky A., Sutskever, I., & Hinton, G. E. Imagenet classification with deep convolutional neural networks. 2012.

18. Chollet F. Deep Learning with Python, Second Edition. Manning Publications Co. LLC, 2021.

19. Bengio Y., Courville A., Goodfellow I. Deep Learning. MIT Press, 2016. 800 p.

20. Emara W., Graham K. L., Kamath U. Transformers for Machine Learning: A Deep Dive. Taylor & Francis Group, 2022.