Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра прикладної математики
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Застосування методів штучного інтелекту
для виявлення об'єктів на відео
(тема)

Виконав:
здобувач _2_ року навчання, групи _ПМм-23-2 _
Панасюк І.Ю.
(прізвище, ініціали)
Спеціальність
113 Прикладна математика
(код і повна назва спеціальності)
Тип програми _освітньо-професійна
(освітньо-професійна або освітньо-наукова)
Освітня програма
Прикладна математика
(повна назва освітньої програми)
Керівник доц. Ламтюгова С.М.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ПМ _____ Сидоров М.В.
(підпис) (прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет _інформаційно-аналітичних технологій та менеджменту
Кафедра _прикладної математики
Рівень вищої освіти _другий (магістерський)
Спеціальність _113 Прикладна математика
(код і повна назва)
Тип програми _освітньо-професійна
(освітньо-професійна або освітньо-наукова)
Освітня програма _Прикладна математика
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри ПМ _____
(підпис)
“ _25_ ” _листопада_ 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві _Панасюку Ігорю Юрійовичу
(прізвище, ім’я, по батькові)

1. Тема роботи _Застосування методів штучного інтелекту для виявлення об’єктів на відео
затверджена наказом по університету від 22 листопада 2024 р. № 1223 Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 6 січня 2025 р.

3. Вихідні дані до роботи _відеоряд з різними об’єктами

4. Перелік питань, що потрібно опрацювати в роботі
1. Аналіз предметної області
2. Вибір і обґрунтування методу розв’язання
3. Програмна реалізація
4. Результати обчислювального експерименту
5. Аналіз можливих застосувань

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

1. Актуальність теми роботи
2. Постановка задачі
3. Аналіз предметної області
4. Метод чисельного аналізу
5. Результати обчислювального експерименту

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Підбір та вивчення технічної літератури за темою роботи	25 листопада – 1 грудня 2024 р.	виконано
2	Вибір та обґрунтування методу	2 – 8 грудня 2024 р.	виконано
3	Розробка алгоритму і програми	9 – 22 грудня 2023 р.	виконано
4	Проведення аналітичних досліджень та розрахунків	23 – 29 грудня 2024 р.	виконано
5	Робота над текстом пояснювальної записки	30 грудня 2024 р. – 9 січня 2025 р.	виконано
6	Представлення роботи на рецензію в ЕК	10 січня 2025 р.	виконано

Дата видачі завдання _25_ листопада 2024 р.

Здобувач _____
(підпис)

Керівник роботи _____ доц. Ламтюгова С.М.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 69 с., 1 табл., 19 рис., 1 дод., 34 джерела.

ВІДЕОПОТІК, ГЛИБОКЕ НАВЧАННЯ, ЗГОРТКОВІ НЕЙРОННІ МЕРЕЖІ, КОМП'ЮТЕРНИЙ ЗІР, МАШИННЕ НАВЧАННЯ, РЕКУРЕНТНІ НЕЙРОННІ МЕРЕЖІ, РОЗПІЗНАВАННЯ ОБ'ЄКТІВ, ШТУЧНИЙ ІНТЕЛЕКТ.

Об'єкт дослідження – процеси автоматичного виявлення об'єктів на відеозаписах.

Мета роботи – побудова найбільш ефективної моделі для точного та швидкого розпізнавання об'єктів на відео.

Методи дослідження – систематизація, класифікація, узагальнення, порівняння, моделювання.

У кваліфікаційній роботі розглянуто задачу побудови найбільш ефективної моделі для точного та швидкого розпізнавання об'єктів на відео. Було проведено огляд сучасних алгоритмів комп'ютерного зору, методів глибокого навчання та проаналізовано основні підходи до задач розпізнавання об'єктів. Основну увагу приділено алгоритму YOLO як одному з найефективніших підходів у реальному часі за якістю результату та швидкості обробки.

Розглянуто різні модифікації та вдосконалення YOLO для підвищення точності та швидкодії, а також проведено аналіз факторів, що впливають на продуктивність моделі.

Запропонована модель була реалізована за допомогою мови програмування Python та бібліотеки OpenCV і протестована на відкритих наборах даних для розпізнавання об'єктів на відео. Наведено результати обчислювального експерименту, які демонструють підвищення точності розпізнавання та зниження затримки обробки кадрів.

ABSTRACT

Introductory note: 69 pages, 1 table, 19 figures, 1 appendix, 34 sources.

ARTIFICIAL INTELLIGENCE, COMPUTER VISION, CONVOLUTIONAL NEURAL NETWORKS, DEEP LEARNING, MACHINE LEARNING, OBJECT RECOGNITION, RECURRENT NEURAL NETWORKS, VIDEO STREAM.

The object of research is the automatic object detection processes in video recordings.

The purpose of research is to develop the most effective model for accurate and fast object detection in videos.

The research methods are systematization, classification, generalization, comparison, modeling.

The qualification paper addresses the task of developing the most effective model for accurate and fast object detection in videos. A review of modern computer vision algorithms, deep learning methods, and analysis of the main approaches to object detection tasks was conducted. The focus was on the YOLO algorithm as one of the most effective real-time approaches in terms of result quality and processing speed.

Various modifications and improvements of YOLO for enhancing accuracy and speed were examined, along with an analysis of factors affecting model performance.

The proposed model was implemented using the Python programming language and the OpenCV library and was tested on open datasets for object detection in videos. The results of the computational experiment demonstrate improved detection accuracy and reduced frame processing delay.

ЗМІСТ

ВСТУП

Актуальність теми. Виявлення об’єктів – це важливе завдання комп’ютерного зору, яке використовується для виявлення екземплярів візуальних об’єктів певних класів (наприклад, людей, тварин, автомобілів або будівель) у цифрових зображеннях, таких як фотографії чи відеокадри. Метою виявлення об’єктів є розробка обчислювальних моделей, які надають найбільш фундаментальну інформацію, необхідну для програм комп’ютерного зору.

Виявлення об'єктів є однією з фундаментальних проблем комп'ютерного зору, є основою для багатьох інших завдань комп’ютерного зору, що стосуються нижньої течії, наприклад, сегментації екземплярів і зображень, створення підписів до зображень, відстеження об’єктів тощо. Спеціальні програми виявлення об’єктів включають виявлення пішоходів, виявлення тварин, виявлення транспортних засобів, підрахунок людей, виявлення обличчя, виявлення тексту, оцінку пози або розпізнавання номерних знаків.

Завдяки прогресу в технологіях, поширенню впровадженню та практичному застосуванню в різних галузях, комп’ютерне бачення, включаючи виявлення об’єктів, переходить у фазу стабільності та широкомасштабної інтеграції. Зараз увага зміщується з експериментальних етапів на вдосконалення та оптимізацію існуючих додатків, що означає важливий крок до їх повної реалізації та впливу на підприємства в різних секторах.

Мета і завдання кваліфікаційної роботи. Метою кваліфікаційної роботи є побудова найбільш ефективної моделі для точного розпізнавання об’єктів на відео.

Для досягнення поставленої мети необхідно виконати наступні завдання:

– провести огляд і аналіз сучасного стану задачі «розпізнавання об’єктів»;

– розглянути існуючі методи для вирішення задачі;

– побудувати модель глибокого навчання;

– розробити програмну реалізацію побудованої моделі;

– зробити оцінку якості роботи моделі на реальних даних;

– на основі отриманих даних зробити висновок про проведену роботу.

Об’єктом дослідження є процеси автоматичного виявлення об’єктів на відеозаписах.

Предметом дослідження є методи та алгоритми штучного інтелекту, зокрема машинного навчання і глибинного навчання, для розпізнавання об’єктів на відео.

Методи дослідження. У роботі використовуються такі методи як систематизація, класифікація, узагальнення, порівняння, моделювання.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ

1.1 Аналіз проблеми виявлення об’єктів на відео

Виявлення об'єкта часто називають розпізнаванням об'єкта, ідентифікацією об'єкта або виявленням зображення. Метою виявлення об’єктів, як області машинного навчання, є розробка обчислювальних моделей, які надають найосновнішу інформацію, необхідну для програм комп’ютерного зору. Іншими словами, він запитує: «які об’єкти де знаходяться?»

Виявлення відеооб’єктів (VOD) імітує зорову кору людини. Це дозволяє машинам аналізувати відео кадр за кадром і ідентифікувати присутні в них об’єкти. Таким чином, виявлення об’єктів у відео працює подібно до розпізнавання зображень штучним інтелектом (ШІ). Такий інструмент призначений для визначення місцезнаходження та ідентифікації об’єктів, які видно на вхідних рухомих зображеннях [1].

Як і будь-який проект ШІ, системи виявлення відеооб’єктів проходять етапи життєвого циклу ШІ. Спочатку необхідно визначити бізнес-цілі та проконсультуватися з експертами з комп’ютерного зору, щоб визначити, чи здійсненні ідеї. Припустимо, що VOD є частиною цього рішення, тож потрібно зануритися в технічну частину. Отже, щоб навчити машину виявляти сутності у відео (як і будь-яке інше завдання AI), потрібно пройти всі необхідні етапи.

По-перше, потрібно створити бібліотеку відеоматеріалів, бажано використовуючи подібне апаратне забезпечення та налаштування, що й для фактичного використання. Коли набір даних буде зібрано, команда машинного навчання у співпраці з діловою стороною створить інструкції для маркування об’єктів, що цікавлять [1].

На етапі міток маркувальники вручну малюють рамки навколо об’єктів, які необхідно ідентифікувати ШІ, щоб навчити машину розпізнавати об’єкти. Ці два кроки, збір даних і маркування, є вирішальними для правильної роботи.

Вкладення часу, людських ресурсів і грошей для отримання найкраще позначеного набору даних є життєво важливим. Цьому процесу можуть допомогти розумні рішення, які дозволяють етикетувальникам позначати об’єкти в одному кадрі, а машина пропонує той самий об’єкт у послідовних кадрах за допомогою методів відстеження об’єктів [2].

Далі алгоритм навчається на позначених даних. Цей крок є критично важливим для забезпечення належної точності. Після цього етапу навчання продуктивність моделі повинна бути оцінена та перевірена на наборі даних, який не використовувався для навчання машини. Це дозволяє оцінити продуктивність моделі в реальних умовах із невидимими даними. Без цього досвіду набору даних може бути важко правильно розділити набір даних і зробити висновки щодо його ймовірної продуктивності в реальних умовах [2].

Наступним етапом є перевірка моделі. Модель проходить тестування в реальному житті, щоб визначити, чи правильно виконано завдання та чи варто продовжувати. Потім покращення відбуваються на основі зворотного зв’язку в результаті проведення різних експериментів. Тестування гарантує, що у вас є оптимальне рішення без упущених крайніх випадків, і знижує ризик значних інвестицій у погано працюючу систему [2].

Як і у випадку з розпізнаванням зображень, можна використовувати готові рішення на різних платформах або реалізовувати готові рішення самостійно. Але платформи та готові системи не можуть впоратися з усіма випадками використання. Якщо дані залежать від домену та/або потрібно виявити щось, чому платформи не навчені, можливо, доведеться створити власне рішення, щоб отримати необхідну точність.

Універсальність штучного інтелекту у виявленні об’єктів демонструється через широкий спектр застосувань у різних секторах, що підкреслює його здатність трансформувати галузі та підвищувати ефективність повсякденних операцій і результатів [3]:

– спостереження та безпека: виявлення об’єктів за допомогою штучного інтелекту відіграє ключову роль у покращенні систем безпеки, виявляючи не-

санкціонований доступ або підозрілі дії за лічені секунди, забезпечуючи швидке реагування на потенційні загрози;

– автономні транспортні засоби: у сфері автономного водіння виявлення об’єктів має вирішальне значення для безпеки та ефективності транспортних засобів, дозволяючи їм ідентифікувати пішоходів, інші транспортні засоби та перешкоди для безпечного руху;

– роздрібна торгівля: роздрібні торговці використовують штучний інтелект для управління запасами, використовуючи функцію виявлення об’єктів для розпізнавання та відстеження продуктів, допомагаючи в управлінні запасами та покращуючи досвід покупців завдяки інтерактивним кіоскам і персоналізованим рекомендаціям;

– охорона здоров’я: виявлення об’єктів у медичних зображеннях дозволяє медичним працівникам визначати особливості, що вказують на захворювання чи аномалії, полегшуючи ранню діагностику та індивідуальні плани лікування;

– сільськогосподарські технології: сільськогосподарський сектор отримує переваги від виявлення об’єктів для моніторингу здоров’я врожаю та худоби, раннього виявлення шкідників і хвороб та оцінки зрілості врожаю, що сприяє ефективному веденню господарства та підвищенню врожайності.

Незважаючи на значні досягнення та застосування ШІ у виявленні об’єктів, деякі проблеми та обмеження впливають на його впровадження та ефективність [4]:

– технічні проблеми: системи виявлення об’єктів часто стикаються з різними умовами освітлення, оклюзіями, де об’єкти частково закриті, і високою щільністю об’єктів у багатолюдних сценах, що може знизити точність і надійність виявлення;

– питання етики та конфіденційності: використання штучного інтелекту для стеження та збору даних викликає серйозні проблеми етики та конфіденційності, зокрема щодо згоди, безпеки даних і можливості стеження без нагляду;

– обмеження сучасних технологій: багато технологій виявлення об’єктів значною мірою покладаються на великі, помічені набори даних для навчання, створення яких може потребувати ресурсів, а обробка в реальному часі вимагає значної обчислювальної потужності, що обмежує розгортання розширених моделей у середовищах з обмеженими ресурсами.

Вирішення цих проблем вимагає постійних досліджень, продуманого впровадження технологій ШІ та зваженого розгляду етичних наслідків [5], щоб переконатися, що переваги виявлення об’єктів можна реалізувати в усіх секторах без шкоди для конфіденційності чи безпеки.

Вивчення успішних реалізацій виявлення об’єктів на основі штучного інтелекту в різних секторах дає цінну інформацію про його потенційний вплив і практичні аспекти його застосування [6]:

– спостереження та безпека: в одному помітному випадку виявлення об’єктів штучного інтелекту захищало великі публічні заходи шляхом виявлення об’єктів, які залишилися без нагляду, і відстеження переміщень натовпу, що значно скоротило час реагування на потенційні загрози безпеці;

– роздрібні інновації: гігант роздрібної торгівлі впровадив виявлення об’єктів для управління запасами, дозволяючи аналізувати полиці в реальному часі та сповіщати про поповнення запасів, суттєво зменшуючи розбіжності в інвентарі та підвищуючи задоволеність клієнтів;

– діагностика в охороні здоров’я: в охороні здоров’я виявлення об’єктів було застосовано до радіології, покращуючи виявлення пухлин у зображенні з більшою точністю, ніж традиційні методи, і полегшуючи ранню діагностику та планування лікування;

– ефективність сільського господарства: у сільськогосподарському секторі виявлення об’єктів за допомогою дронів використовується для моніторингу здоров’я посівів на великих площах, що на ранній стадії дозволяє виявити такі проблеми, як поширення хвороби та дефіцит поживних речовин, і призводить до більш цілеспрямованих заходів.

Передові практики включають сувору перевірку моделей штучного інте-

лекту, щоб забезпечити їх надійність і точність, етичне використання технологій штучного інтелекту, які поважають конфіденційність і згоду, а також постійний моніторинг і технічне обслуговування систем штучного інтелекту для адаптації до нових викликів і можливостей [7].

Завдяки цим ідеям і прикладам стає очевидним потенціал виявлення об'єктів штучного інтелекту для трансформації галузей і покращення результатів, провіщаючи майбутнє, де роль штучного інтелекту буде водночас інноваційною та незамінною.

1.2 Моделі та методи виявлення об'єктів на відео

Розуміння основ штучного інтелекту у виявленні об'єктів починається з ознайомлення з основними поняттями та термінологією, що є ключовими для цієї технології.

Обмежувальні рамки – це прямокутні координати, які точно вказують розташування об'єкта на зображенні, ефективно окреслюючи його периметр.

З іншого боку, показники впевненості кількісно визначають впевненість моделі ШІ в точності виявлення об'єкта, пропонуючи імовірнісну оцінку кожного ідентифікованого об'єкта.

В основі виявлення об'єктів за допомогою ШІ лежать різні моделі ШІ, кожна з яких має унікальні можливості та застосування [8]:

– згорткові нейронні мережі (CNN) є фундаментальними, вони обробляють зображення через шари для виявлення особливостей і шаблонів;

– згорткові нейронні мережі на основі регіонів (R-CNN) підвищують точність шляхом сканування попередньо визначених областей зображення;

– You Only Look Once (YOLO) виділяється своєю швидкістю, він аналізує все зображення за один прохід, щоб виявити об'єкти;

– одноразові детектори (SSD) використовують одну глибоку нейронну мережу, щоб збалансувати швидкість і точність.

Роль наборів даних і анотацій є вирішальною в цій екосистемі. Набори даних складаються з величезних колекцій зображень, кожне з яких ретельно анотовано, щоб вказати присутність і положення об’єктів.

Ці анотації, будь то обмежувальні рамки, категорії об’єктів чи інші маркери, служать базовими даними для навчання моделей штучного інтелекту, навчаючи їх розпізнавати шаблони та робити точні прогнози щодо нових, небачених зображень [9].

Глибоке навчання, зокрема за допомогою згорткових нейронних мереж (CNN), зробило революцію у виявленні об’єктів.

CNN автоматизують виділення ознак, усуваючи потребу в ручному виборі ознак і значно підвищуючи здатність моделі розпізнавати складні шаблони зображень.

Уважніше вивчення конкретних архітектур відкриває різноманітний ландшафт [10]:

– R-CNN і його наступники, Fast R-CNN і Faster R-CNN , поступово скорочують час обчислень, одночасно підвищуючи точність виявлення; швидший R-CNN представив можливість виявлення об’єктів у реальному часі;

– YOLO (You Only Look Once) змінює гру, аналізуючи все зображення одночасно, різко скорочуючи час обробки та дозволяючи виявлення об’єктів майже в реальному часі;

– SSD (Single Shot MultiBox Detector) пропонує переконливу альтернативу, забезпечує високу точність, зберігаючи швидкість, передбачаючи існування об’єктів та їх обмежувальні прямокутники протягом одного проходу через мережу.

Трансферне навчання стало важливою технікою виявлення об’єктів. Це дозволяє моделям, навченим одному завданню, перепрофілювати для іншого пов’язаного завдання з мінімальним додатковим навчанням.

Цей підхід особливо цінний у виявленні об’єктів, де навчання моделі з нуля вимагає значних обчислювальних ресурсів і даних.

Процес навчання моделі виявлення об’єктів включає кілька ключових

етапів [11]:

– підготовка даних: збір і підготовка набору даних із різноманітними прикладами та точними анотаціями;

– вибір моделі: вибір відповідної архітектури моделі на основі конкретних вимог завдання з урахуванням таких факторів, як швидкість, точність і обчислювальні ресурси;

– навчання: коригування вагових коефіцієнтів моделі через ітераційний доступ до набору даних, використовуючи комбінацію прямого та зворотного поширення для мінімізації частоти помилок;

– оцінка: використання окремих тестових наборів даних для оцінки ефективності моделі, гарантуючи, що вона може точно виявляти об’єкти на нових, невидимих зображеннях.

Ознайомлюючись із цими основоположними концепціями та вдосконаленими методами, ми отримуємо повне розуміння механізмів, що керують штучним інтелектом у виявленні об’єктів, закладаючи основу для інноваційних застосувань у різних галузях.

Створення можливостей ШІ для виявлення об’єктів потребує поєднання складного програмного забезпечення та надійних апаратних компонентів.

Інтеграція цих технологій дозволяє розробляти, навчати та розгортати моделі виявлення об’єктів, які можуть обробляти та аналізувати зображення чи відео в режимі реального часу або майже в режимі реального часу.

Розглянемо вимоги до програмного забезпечення [12].

1. Фреймворки та бібліотеки розробки: такі популярні фреймворки, як TensorFlow, PyTorch і Keras, пропонують необхідні інструменти та бібліотеки для розробки, навчання та перевірки моделей глибокого навчання. Ці структури забезпечують широку підтримку згорткових нейронних мереж (CNN) та інших архітектур, пов’язаних з виявленням об’єктів.

2. Попередньо підготовлені моделі та набори даних: доступ до попередньо підготовлених моделей (таких як YOLO, SSD і Faster R-CNN) і великих анотованих наборів даних (як-от ImageNet, COCO та Pascal VOC) є надзвичайно

важливим. Ці ресурси значно скорочують час розробки та необхідні обчислювальні ресурси, забезпечуючи початкову точку, яку можна додатково налаштувати.

3. Інструменти для анотацій: для завдань виявлення нестандартних об’єктів інструменти для анотацій необхідні для позначення зображень обмежувальними рамками або іншими відповідними маркерами. Такі інструменти, як LabelImg або CVAT, спрощують анотацію вручну, дозволяючи налаштовувати спеціальні набори даних відповідно до конкретних потреб.

Розглянемо вимоги до обладнання [13, 14]:

1. Високопродуктивні графічні процесори: навчання моделей глибокого навчання для виявлення об’єктів потребує великих обчислень. Високопродуктивні графічні процесори (GPU) необхідні для прискорення навчання. У цьому домені зазвичай використовуються графічні процесори NVIDIA (наприклад, Tesla, Quadro та GeForce) або AMD.

2. Достатній обсяг пам’яті та сховища: моделі глибокого навчання та набори даних потребують значного обсягу оперативної пам’яті та місця для зберігання. Твердотільні накопичувачі великої ємності (SSD) і великий обсяг оперативної пам’яті (64 ГБ або більше) допомагають керувати великими наборами даних і тимчасовими даними, створеними під час навчання моделі.

3. Спеціалізоване апаратне забезпечення для розгортання: для розгортання моделей виявлення об’єктів у реальних додатках можна використовувати спеціалізоване обладнання, наприклад периферійні пристрої або вбудовані системи (наприклад, серія NVIDIA Jetson, Google Coral). Ці пристрої оптимізовані для низького енергоспоживання та ефективної обробки в реальному часі, що робить їх придатними для таких програм, як камери спостереження, дрони та автономні транспортні засоби.

4. Інтегровані середовища розробки (IDE та редактори коду: такі інструменти, як Visual Studio Code, PyCharm або Jupyter Notebooks, підтримують розробку моделей AI, пропонуючи функції редагування коду, налагодження та керування версіями. Вони сприяють ефективному кодуванню та співпраці між

командами розробників.

Створення можливостей штучного інтелекту для виявлення об’єктів передбачає продуманий вибір програмного та апаратного забезпечення, збалансування вимог до розробки та розгортання моделі.

За допомогою правильного поєднання інструментів і ресурсів команди можуть ефективно розробляти та розгортати потужні системи виявлення об’єктів, які відповідають їхнім оперативним потребам.

1.3 Програмні засоби для виявлення об’єктів на відео на основі штучного інтелекту

У міру того, як світ охоплює трансформаційна сила штучного інтелекту, компанії по всьому світу шукають надійні, ефективні та масштабовані інструменти для завдань виявлення об’єктів. З великою кількістю доступних варіантів, від створення спеціального детектора об’єктів до вибору правильної платформи може бути складно. Розглянемо основні програмні засоби для виявлення об’єктів на відео на основі штучного інтелекту.

Scale AI – платформа даних, яка допомагає командам штучного інтелекту вдосконалювати свої моделі, надаючи високоякісні послуги маркування та контролю даних. Даний проєкт [15]:

– працює з провідними компаніями та державними установами в різних областях, таких як автомобільна промисловість, генеративний штучний інтелект, обробка природної мови, комп’ютерне бачення тощо;

– пропонує повну платформу для створення генеративних додатків штучного інтелекту, включаючи тонке налаштування, оперативне проектування, безпеку, безпеку моделі, оцінку моделі та корпоративні програми.

Проєкт призначений для команд штучного інтелекту, яким потрібне наскрізне рішення, орієнтоване на дані, для керування всім життєвим циклом ма-

шинного навчання та розробки високоякісних наборів даних для своїх програм штучного інтелекту

Supervisely – уніфікована ОС для комп'ютерного зору, яка допомагає компаніям швидше та краще розробляти ШІ. Даний проєкт [15]:

– надає ряд інструментів маркування, функцій автоматизації та механізмів забезпечення якості для створення високоякісних наборів даних;

– інтегрується з багатьма інструментами з відкритим вихідним кодом і спеціальними інструментами для методів глибокого навчання, навчання нейронної мережі, обслуговування, запитів, перетворення та аналізу;

– дозволяє налаштувати робочий процес за допомогою Supervisely Apps, які є інтерактивними веб-програмами на основі Python.

Проєкт призначений для компаній та дослідників, яким потрібне локальне наскрізне рішення корпоративного рівня для вирішення будь-яких завдань із розробки комп'ютерного зору.

V7 Labs – платформа даних ШІ для комп'ютерного зору, яка допомагає командам машинного навчання керувати своїми наборами даних, анотаціями та моделями. Даний проєкт [16]:

– пропонує нейронні мережі глибокого навчання, що не залежать від класу, які можуть створювати ідеальні багатокутні маски за лічені секунди;

– дозволяє користувачам налаштовувати робочі процеси, поєднуючи більше ніж одну модель виявлення об'єктів, людей і дані на різних етапах;

– надає доступ до понад 5000 професійних анотаторів, які можуть позначати дані на вимогу за високими стандартами якості;

– підтримує різні детектори об'єктів і формати даних, такі як зображення, відео, 3D-хмари точок, обмежувальні рамки, ключові точки тощо.

Проєкт призначений для компаній, яким потрібно створювати надійне та універсальне програмне забезпечення для візуального прийняття рішень на основі методів глибокого навчання та алгоритмів виявлення об'єктів [16].

Viso Suite – хмарний детектор об'єктів і платформа аналізу відео, яка допомагає компаніям отримувати статистичні дані з їхніх відеоданих. Даний

проєкт:

– використовує штучний інтелект для виконання процедури виявлення об’єктів, яка автоматично виявляє, класифікує та відстежує об’єкти та події у відео;

– дозволяє користувачам створювати спеціальні інформаційні панелі та звіти з інтерактивними діаграмами та графіками для візуалізації аналітики відео;

– підтримує різні відеоформати та джерела, такі як камери, дрони, мобільні пристрої та онлайн-платформи.

Даний проєкт призначений для підприємств, яким потрібно аналізувати великі обсяги відеоданих для безпеки, маркетингу, досліджень або операцій.

Labelbox – платформа маркування даних, яка допомагає компаніям створювати високоякісні навчальні дані для своїх моделей машинного навчання [17]. Даний проєкт:

– підтримує різні типи анотацій, наприклад сегментацію зображень, класифікацію тексту, відстеження відеооб’єктів та інші завдання комп’ютерного зору;

– дозволяє користувачам налаштовувати робочі процеси, інтегрувати власні джерела даних і інструменти, а також керувати своїми командами та проектами;

– використовує моделі глибокого навчання та методи роботи людини в циклі для підвищення ефективності та точності маркування даних;

– надає аналітику та інформацію для моніторингу якості та прогресу маркування даних.

Проєкт призначений для підприємств, яким потрібне масштабоване та гнучке рішення для створення великомасштабних і складних наборів навчальних даних для різних програм машинного навчання [18].

Toloka – платформа краудсорсингу, яка дозволяє обробляти великі обсяги візуальних даних, розподіляючи завдання між великою кількістю людей, які виконують невеликі завдання. Даний проєкт:

– має доступні інструменти для ефективного анотування даних, класифікації зображень, виявлення та розпізнавання об’єктів і сегментації екземплярів;

– має мережу з понад 10 мільйонів фрілансерів і незалежних підрядників, відомих як Tolokers, які можуть виконувати різноманітні завдання для подальшого розвитку, тонкого налаштування та оцінки LLM;

– має інструменти контролю якості Toloka, щоб забезпечити точність і надійність даних.

Проєкт потрібен компаніям або командам, які хочуть поєднати потужність штучного інтелекту та перевірку людьми, щоб позначати свої відео та зображення [17].

Superannotate – платформа, яка допомагає створювати та керувати високоякісними навчальними даними для ваших проєктів штучного інтелекту та машинного навчання. Даний проєкт:

– дозволяє коментувати різні типи даних, такі як зображення, відео, текст, LiDAR, аудіо тощо, використовуючи розширені інструменти та функції автоматизації;

– можна отримати доступ до глобального ринку професійних груп анотацій, які можуть допомогти вам із вашими потребами щодо маркування даних;

– забезпечує керування даними, контроль якості та інструменти MLOps для покращення продуктивності моделі та оптимізації робочого процесу;

– використовується в різних галузях і варіантах використання, як автономне водіння, робототехніка, охорона здоров’я, безпека тощо.

Superannotate потрібен командам, які хочуть створювати та керувати високоякісними навчальними даними для своїх проєктів штучного інтелекту та хочуть отримати доступ до глобального ринку професійних команд анотацій [18].

OpenCV – бібліотека комп’ютерного бачення та машинного навчання з відкритим кодом для підтримки програм комп’ютерного бачення в реальному часі. Даний проект:

– має відкритий код і кросплатформенність, підтримку C++, Python, Java та інших мов у Linux, Windows, MacOS, iOS та Android;

– оптимізовано для додатків у реальному часі та може використовувати апаратне прискорення та паралельну обробку;

– має велике й активне співтовариство розробників і користувачів із понад 50 тисячами людей, які беруть участь у проекті на GitHub;

– надає серію апаратних модулів і доповнень OpenCV AI Kit до бібліотеки OpenCV, які підтримують просторовий ШІ;

– поєднує бібліотеку OpenCV із технологією розпізнавання облич із найвищим рейтингом у OpenCV Face Recognition.

Проєкт потрібен розробникам та дослідникам, які хочуть створювати програми в таких сферах, як розпізнавання обличь, доповнена реальність, виявлення об’єктів тощо [18].

1.4 Змістовна та формальна постановка задачі

Розпізнавання об’єктів на відео є однією з ключових задач у галузі комп’ютерного зору, що знаходить застосування в різних сферах, таких як системи безпеки, автономний транспорт, відеоспостереження, аналіз спортивних подій та інші. Ефективність розв’язання цієї задачі залежить від точності та швидкості алгоритмів обробки даних.

Основним об’єктом дослідження є алгоритм YOLO (You Only Look Once), який забезпечує високу продуктивність у задачах розпізнавання об’єктів у реальному часі. Однак для підвищення його ефективності необхідно враховувати специфіку даних, оптимізувати архітектуру моделі, а також налаштувати гіперпараметри для досягнення максимальної точності при мінімальній затримці обробки відеопотоку.

Задача полягає у побудові та вдосконаленні моделі на основі YOLO, яка забезпечить:

– точне розпізнавання об’єктів на відео;

– мінімальну затримку обробки кадрів для роботи в реальному часі;

– високу адаптивність до різних наборів даних і сценаріїв застосування.

На вході маємо:

– вхідні дані: відеопотік або набір відеокадрів, представлених у вигляді зображень (RGB-формат);

– множину класів $C = \{c_1, c_2, ..., c_n\}$, яка визначає об'єкти, що підлягають розпізнаванню;

– початкову модель YOLO з параметрами θ, що підлягають оптимізації.

Необхідно побудувати модель $f(x, \theta)$, яка отримує на вході зображення $x \in X$ і визначає:

– координати обмежувальних рамок об'єктів $B = \{b_1, b_2, ..., b_k\}$;

– належність кожного обмежувального прямокутника до класу c_i з ймовірністю $p(c_i | b_j)$.

Крім цього, потрібно оптимізувати параметри θ для забезпечення максимізації метрики точності (наприклад, mAP – mean Average Precision) і мінімізації часу обробки одного кадру.

Умови:

– обмеження на апаратні ресурси: модель має бути ефективною для роботи як на високопродуктивних GPU, так і на менш потужних системах;

– модель має забезпечувати точність не менше ніж 90% для тестових наборів даних.

Методи розв'язання:

– використання вдосконаленої архітектури YOLO (наприклад, YOLOv5 або YOLOv8);

– адаптивне налаштування гіперпараметрів моделі (розмір пакета, коефіцієнт навчання, кількість ітерацій);

– оптимізація обчислень за допомогою бібліотек Python (PyTorch, OpenCV).

Очікуваний результат: створення моделі, здатної точно та швидко розпізнавати об'єкти на відео, що підтверджено результатами тестування на наборах

реальних даних.

1.5 Постановка задач дослідження

Метою кваліфікаційної роботи є побудова найбільш ефективної моделі для точного розпізнавання об’єктів на відео.

Для досягнення поставленої мети необхідно виконати наступні завдання:

– провести огляд і аналіз сучасного стану задачі «розпізнавання об’єктів»;

– розглянути існуючі методи для вирішення задачі;

– побудувати модель глибокого навчання;

– розробити програмну реалізацію побудованої моделі;

– зробити оцінку якості роботи моделі на реальних даних;

– на основі отриманих даних зробити висновок про проведену роботу.

2 ВИБІР ТА ОБҐРУНТУВАННЯ МЕТОДУ РОЗВ'ЯЗАННЯ

2.1 Особливості застосування штучного інтелекту при розпізнаванні об'єктів на відео

Виявлення об'єктів у штучному інтелекті – це, по суті, технологія, яка дозволяє машинам не лише ідентифікувати, а й знаходити конкретні об'єкти на зображенні чи відео. Це ключова частина комп'ютерного зору, галузі, яка допомагає машинам інтерпретувати візуальні дані так, як це роблять люди. Ця технологія використовується в усьому: від безпілотних автомобілів до систем безпеки, що дозволяє машинам розпізнавати обличчя, людей і об'єкти в реальному часі.

У світі обробки зображень виявлення об'єктів відіграє вирішальну роль, допомагаючи комп'ютерам аналізувати зображення та ідентифікувати об'єкти в них. Процес передбачає розбиття зображення на різні елементи та пошук шаблонів, які вказують на присутність об'єкта. Це часто робиться за допомогою таких методів, як виявлення країв і сегментація, які полегшують комп'ютерам розуміння того, що міститься на зображенні. Розвиток ШІ значно підвищив точність цих методів, особливо в поєднанні з машинним навчанням [17].

Машинне навчання зробило виявлення об'єктів набагато розумнішим і надійнішим. Навчаючи системи ШІ на великих наборах даних зображень, ці системи вчаться розпізнавати шаблони та особливості, пов'язані з різними об'єктами. ШІ не просто запам'ятовує ці об'єкти – він вчиться на них і з часом стає кращим. Такі методи, як навчання під наглядом, коли дані позначаються, щоб керувати машиною, допомагають штучному інтелекту підвищити свою точність. Глибоке навчання, тип машинного навчання, робить цей крок далі, дозволяючи штучному інтелекту автоматично вивчати складні функції на зображеннях без втручання людини.

Навчання штучного інтелекту виявленню об'єктів передбачає передачу великої кількості даних із мітками (зображень, де позначені об'єкти). Ці дані

допомагають системі ШІ дізнатися, що шукати в нових зображеннях. Під час цього процесу система починає розпізнавати візуальні особливості, такі як форми, текстури та кольори, які визначають різні об’єкти. Навчання зазвичай використовує нейронні мережі, такі як згорткові нейронні мережі (CNN), які розроблені спеціально для завдань із використанням даних зображень. З часом, коли штучний інтелект отримує більше даних, він навчиться виявляти об’єкти точніше та в ширшому діапазоні умов [17].

Існує кілька методів, які використовує штучний інтелект для виявлення об’єктів, серед яких найпопулярнішими є згорткові нейронні мережі (CNN) і регіональні CNN (R-CNN). CNN розроблені для перегляду даних у вигляді сітки, що робить їх чудовими для таких завдань, як виявлення об’єктів. Тим часом R-CNN розбивають зображення на менші області та аналізують їх окремо, щоб знайти об’єкти. Досконаліші методи, такі як YOLO (You Only Look Once) і SSD (Single Shot Multibox Detector), дозволяють виявляти об’єкти в реальному часі шляхом обробки всього зображення одночасно, що робить їх швидшими та ефективнішими [17].

ШІ ідентифікує об’єкти на зображеннях, шукаючи візуальні особливості, які відрізняють один об’єкт від іншого. Це можуть бути краї, текстури або форми, які з’являються на зображенні. Завдяки машинному навчанню ШІ навчиться розпізнавати ці функції навіть у незнайомих умовах. Це схоже на те, як люди вчаться розпізнавати об’єкти: через повторне опромінення та розпізнавання образів. Чим більше даних отримує штучний інтелект, тим краще він ідентифікує об’єкти, навіть у складних умовах, таких як слабке освітлення або жваві сцени.

Виявлення об’єктів використовується в багатьох сферах для підвищення ефективності та економії коштів і часу. Це дає машинам можливість ідентифікувати об’єкти в потоках живого відео та досягати бажаних результатів. У поєднанні з потужністю технології комп’ютерного зору відстеження та підрахунок також можна виконувати для реальних застосувань.

Деякі з завдань виявлення об’єктів включають визначення місцезнаходження, відстеження, підрахунок об’єктів і виявлення аномалій. Події в режимі

реального часу можуть відстежувати власники бізнесу або керівники старшого рівня, використовуючи можливості виявлення об’єктів.

Відеоаналітика – це технологія, яка поєднує відеоспостереження та комп’ютерний зір і надає абсолютно новий вимір сфері безпеки. Торгові центри, міні-маркети та ресторани можуть використовувати програми виявлення об’єктів для виявлення людей і предметів у своїх приміщеннях. Виявлення об’єктів на основі відеоаналітики допомагає компаніям зрозуміти поведінку клієнтів і підвищити ефективність роботи. Вона надає власникам бізнесу доступ до інформації в режимі реального часу, яка дає їм змогу миттєво покращувати свій бізнес, що постійно зростає [18].

Виявлення об’єктів допомагає роздрібним торговцям розумно відстежувати продукти, аналізуючи, як часто клієнти зупиняються біля вітрин і які товари кому здебільшого продаються. Ресторани швидкого обслуговування також можуть отримати переваги від відеоаналітики, відстежуючи приготування їжі та прийом замовлень, щоб забезпечити оптимальну швидкість обслуговування.

Виявлення об’єктів також допомагає налаштувати безперебійну систему перевірки. Хоча для цього мають бути задіяні не лише камери, а й датчики. Тепер покупці можуть просто зібрати потрібні товари, і, за допомогою штучного інтелекту, негайно буде створено віртуальний кошик для покупок. Ці товари можна придбати безготівково або навіть безконтактно, просто пройшовши через двері. Все це повністю автоматизовано, щоб забезпечити клієнту безпроблемний досвід [19].

Програми виявлення об’єктів можна навчити ідентифікувати об’єкти в інвентарі та миттєво сповіщати менеджера інвентаризації про те, що товари мають бути поповнені. Комп’ютерне бачення допомагає багатьом підприємствам краще керувати своїми запасами, а також ефективніше та акуратніше зберігати товари на полицях. Ця технологія також може бути використана для відстеження тенденцій і прогнозування попиту на наступну партію запасів.

Виявлення об’єктів можна використовувати для відстеження діяльності співробітників як усередині, так і поза магазином. Його можна використовувати

для моніторингу та підрахунку кількості людей, які входять і виходять із магазину за певний період часу. Подібним чином ми можемо стежити за рухом людей, які ховаються біля магазину, щоб підрахувати кількість людей, які проходять повз і зупиняються, щоб подивитися на вітрини.

Дані, зібрані за допомогою рішення, можна використовувати для розуміння часу пік покупок для різних типів клієнтів і відповідної оптимізації. Менеджери можуть відстежувати рух клієнтів усередині магазину, щоб визначити оптимальний потік трафіку. Все це може допомогти власникам спланувати продакт-плейсмент і рекламні акції, а також провести ремонт магазину [19].

Промислові зони схильні до багатьох небажаних аварій або подій, які можуть призвести до загибелі людей або серйозних пошкоджень. Багато лідерів з охорони здоров'я застосовують певні правила та норми для підтримки безпечного середовища для працівників. З'являються нові технології, які можуть прискорити цей процес, і це буде виявлення об'єктів на основі комп'ютерного зору. Це ідеальне рішення для відстеження використання комплектів захисту працівниками промисловості.

Ці програми виявлення об'єктів навчені виявляти каски, жилети, окуляри та маски для обличчя на всій території закладу для забезпечення безпеки працівників. Якщо рішення виявить, що хтось порушує вимоги безпеки через неправильне спорядження, сповіщення буде надіслано спеціалісту з безпеки через панель керування, текстове повідомлення та навіть електронну пошту.

Виявлення об'єктів можна використовувати, щоб переконатися, що правильні компоненти використовуються на складальних лініях і дотримуються правильних процесів, що забезпечує цілісну гарантію якості для лідерів галузі. Багато систем автоматизації запрограмовані на виконання інструкцій, але не можуть виконувати завдання поза цією програмою. Завдяки комп'ютерному зору боти тепер можуть виявляти широкий спектр об'єктів і адаптуватися до різних середовищ [20].

Точність і якість є двома важливими факторами для будь-якої виробленої продукції та товару. Виявлення об'єктів використовується для ідентифікації де-

талей машин і готової продукції, які не відповідають стандартам якості. Постійний моніторинг робочих станцій, виробничих ліній і процесів контролю якості також може відігравати важливу роль у виявленні пошкоджених продуктів перед їх відправленням.

2.2 Convolutional Neural Network

Convolutional Neural Network (CNN) – це тип архітектури нейронної мережі глибокого навчання, яка зазвичай використовується в комп'ютерному зорі. Комп'ютерний зір – це область штучного інтелекту, яка дозволяє комп'ютеру розуміти та інтерпретувати зображення або візуальні дані [20].

Коли справа доходить до машинного навчання, штучні нейронні мережі показують дуже хороші результати. Нейронні мережі використовуються в різних наборах даних, таких як зображення, аудіо та текст. Різні типи нейронних мереж використовуються для різних цілей, наприклад, для прогнозування послідовності слів використовуються рекурентні нейронні мережі, точніше LSTM, аналогічно для класифікації зображень використовуються згорткові нейронні мережі. Створимо базовий будівельний блок для CNN.

У звичайній нейронній мережі є три типи шарів [21].

Тип 1. Вхідний шар – це шар, в якому ми даємо вхідні дані моделі. Кількість нейронів у цьому шарі дорівнює загальній кількості ознак у наших даних (кількість пікселів у випадку зображення).

Тип 2. Прихований шар: вхідні дані з вхідного шару подаються в прихований. Може бути багато прихованих шарів залежно від моделі та розміру даних. Кожен прихований шар може мати різну кількість нейронів, яка, як правило, більша, ніж кількість ознак. Вихід з кожного шару обчислюється шляхом множення матриці вихідного сигналу попереднього шару з вагами цього шару, що навчаються, а потім додаванням зміщень, що навчаються, з подальшою функцією активації, що робить мережу нелінійною.

Тип 3. Вихідний шар: вихідні дані з прихованого шару подаються в логістичну функцію, таку як sigmoid або softmax, яка перетворює вихідні дані кожного класу в оцінку ймовірності кожного класу.

Дані подаються в модель, і вихідні дані з кожного шару виходять з вищезазначеного кроку, який називається feedforward, потім обчислюється похибка за допомогою функції похибки, деякі поширені функції помилок – перехресна ентропія, помилка втрати квадрата тощо. Функція помилки вимірює, наскільки добре працює мережа. Після цього ми розповсюджуємо назад у модель, обчислюючи похідні. Цей крок називається зворотним поширенням, який в основному використовується для мінімізації втрат.

Згорткова нейронна мережа (CNN) – це розширена версія штучних нейронних мереж (ШНМ), яка переважно використовується для вилучення функції з набору даних сіткоподібної матриці. Наприклад, візуальні набори даних, такі як зображення або відео, де шаблони даних відіграють велику роль [21].

Розглянемо архітектуру CNN. Згорткова нейронна мережа складається з кількох рівнів, таких як вхідний шар, згортковий шар, шар пулінгу та повністю пов'язані шари (рис. 2.1).

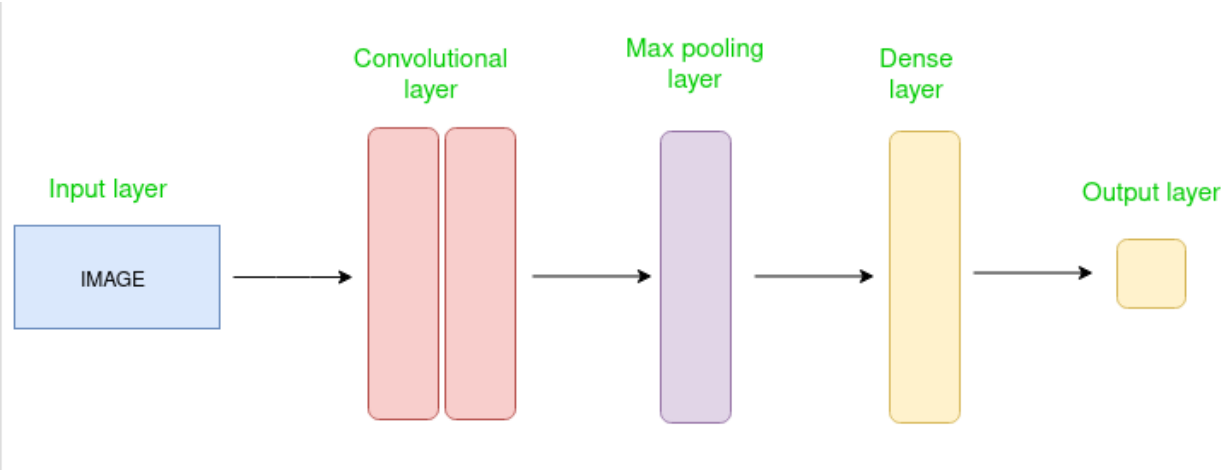

Рисунок 2.1 – Проста архітектура CNN

Шар згортки застосовує фільтри до вхідного зображення для вилучення об'єктів, шар «Об'єднання» зменшує вибірку зображення, щоб зменшити кіль-

кість обчислень, а повністю підключений шар робить остаточне прогнозування. Мережа вивчає оптимальні фільтри за допомогою зворотного поширення і градієнтного спуску [22].

Згорткові нейронні мережі або ковнети – це нейронні мережі, які мають спільні параметри. Уявімо, що є образ. Він може бути представлений у вигляді кубоподібного м'яза, що має свою довжину, ширину (розмір зображення) та висоту (тобто канал, оскільки зображення зазвичай мають червоний, зелений та синій канали) (рис. 2.2).

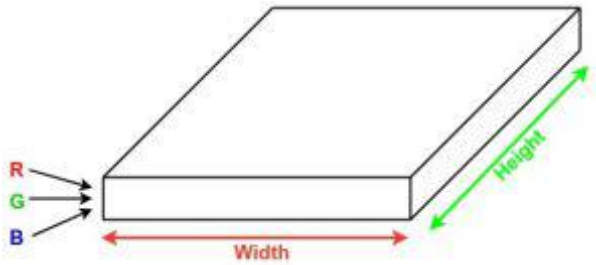

Рисунок 2.2 – Згорткові шари

Візьмемо невеликий фрагмент цього зображення і запускастимо на ньому невелику нейронну мережу, яка називається фільтром або ядром, з виходами, наприклад, K і представленням їх вертикально. Проведемо цією нейронною мережею по всьому зображенню, в результаті отримаємо ще одне зображення з різною шириною, висотою і глибиною. Замість просто каналів R, G і B тепер маємо більше каналів, але меншу ширину та висоту. Ця операція називається згорткою. Якщо розмір патча збігається з розміром зображення, це буде звичайна нейронна мережа (рис. 2.3).

Математичний огляд згортки [23]:

– шари згортки складаються з набору фільтрів (або ядер), що мають малу ширину і висоту, і таку ж глибину, як і глибина вхідного об'єму (3, якщо вхідний шар є вхідним зображенням); наприклад, якщо нам потрібно виконати згортку на зображенні з розмірами 34x34x3, то можливим розміром фільтрів

може бути axax3, де 'a' може бути будь-яким, наприклад, 3, 5 або 7, але меншим у порівнянні з розміром зображення;

– під час прямого проходу необхідно крок за кроком проводити кожним фільтром по всьому вхідному об'єму, де кожен крок може мати значення 2, 3 або навіть 4 для зображень високої розмірності, і обчислювати скалярний добуток між вагами ядра та патчем від вхідного об'єму;

–переміщаючи фільтри, отримаємо 2-D вихід для кожного фільтра, і в результаті складемо їх разом, отримаємо вихідний об'єм, глибина якого дорівнює кількості фільтрів.

В результаті, мережа вивчить всі фільтри.

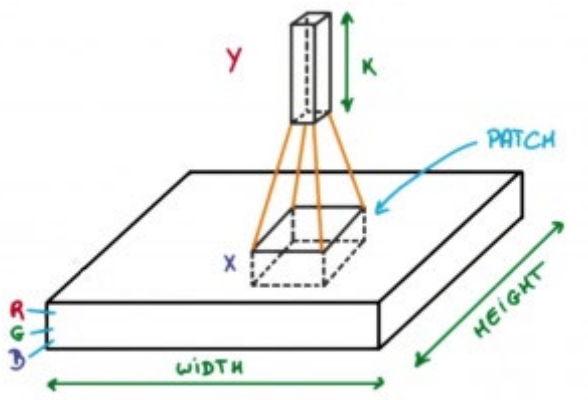

Рисунок 2.3 – Згортка

Повна архітектура згорткових нейронних мереж також відома як ковнети. Ковнет – це послідовність шарів, де кожен шар перетворює один об'єм в інший за допомогою диференційовної функції.

Розглянемо приклад, виконавши ковнет на зображенні розмірністю 32x32x3.

Вхідний шар – це шар, в якому ми даємо вхідні дані моделі. У CNN, як правило, вхідним сигналом буде зображення або послідовність зображень. Цей шар отримує необроблений вхід зображення шириною 32, висотою 32 і глибиною 3.

Згортковий шар – це шар, який використовується для вилучення функції з вхідного набору даних. Він застосовує набір фільтрів, які можна вивчити, відомих як ядра, до вхідних зображень. Фільтри/ядра являють собою менші матриці, зазвичай 2×2, 3×3 або 5×5. Він ковзає по даних вхідного зображення та обчислює скалярний добуток між вагою ядра та відповідним патчем вхідного зображення. Вихідні дані цього шару називаються картами об'єктів. Припустимо, ми використовуємо в цілому 12 фільтрів для цього шару, отримаємо обсяг на виході розміром 32x32x12.

Рівень активації: додаючи функцію активації до виводу попереднього шару, шари активації додають мережі нелінійність. Він застосовує функцію поелементної активації до виходу згорткового шару. Деякі поширені функції активації: RELU: max(0, x), Tanh, Leaky RELU тощо. Гучність залишається незмінною, отже, обсяг на виході буде мати розміри 32x32x12.

Шар об'єднання: цей шар періодично вставляється в ковнети, і його основна функція полягає в зменшенні розміру обсягу, що робить обчислення швидкими, зменшує пам'ять, а також запобігає переналаштуванню. Двома поширеними типами шарів пулінгу є максимальне об'єднання та середнє об'єднання [24].

Якщо використовуємо максимальний пул з фільтрами 2x2 і крок 2, то отриманий обсяг буде розміром 16x16x12 (рис. 2.4).

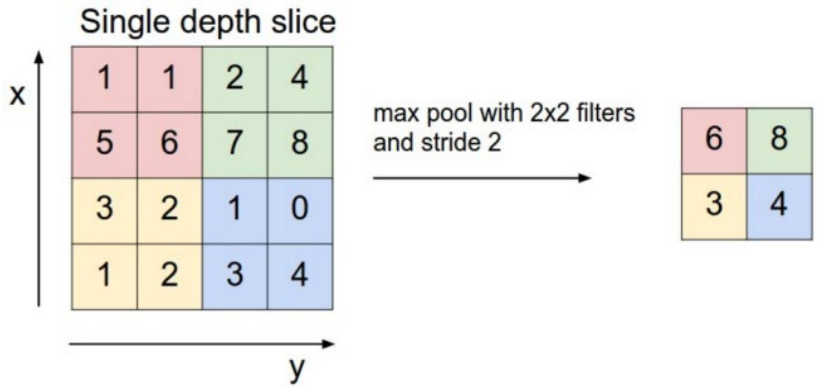

Рисунок 2.4 – Ілюстрація операції максимального пулінгу (Max Pooling) з ядром 2x2 та кроком 2

Зведення: отримані карти ознак зводяться в одновимірний вектор після шарів згортки та об'єднання, щоб їх можна було передати в повністю зв'язаний шар для категоризації або регресії.

Повністю з'єднані шари: бере вхідні дані з попереднього шару та обчислює остаточну класифікацію або завдання регресії (рис. 2.5).

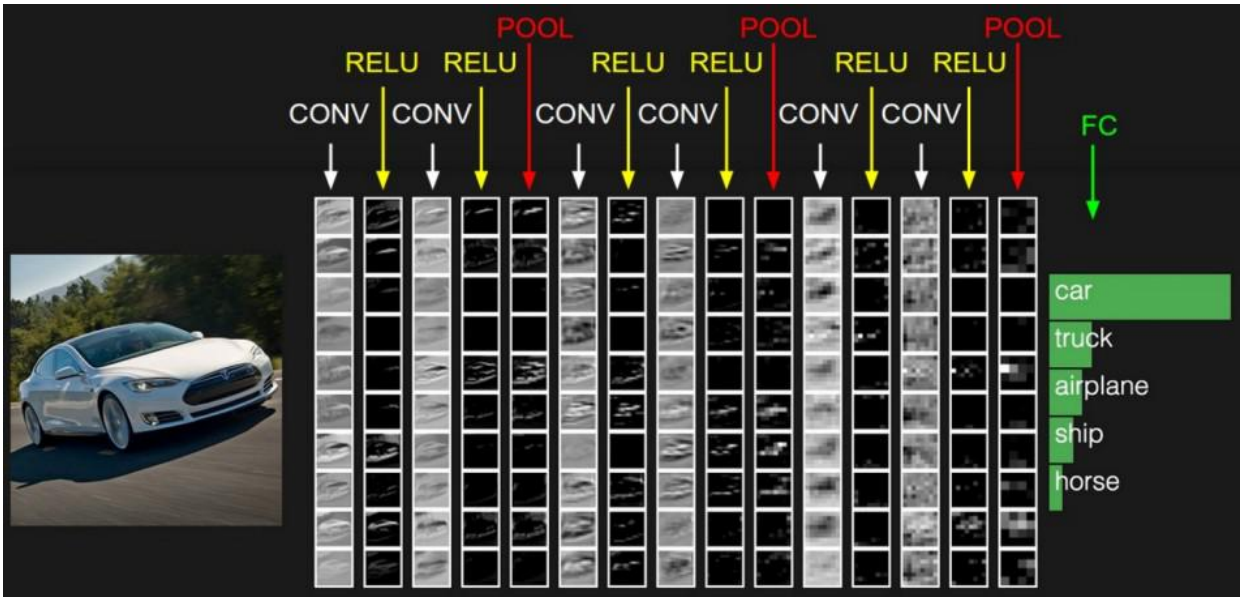

Рисунок 2.5 – Ілюстрація шарів згорткової нейронної мережі (CNN) для класифікації зображень

Вихідний шар: вихідні дані з повністю з'єднаних шарів подаються в логістичну функцію для завдань класифікації, таких як sigmoid або softmax, яка перетворює вихідні дані кожного класу в оцінку ймовірності кожного класу [25].

Розглянемо приклад застосування CNN до зображення. Маємо зображення та застосуємо шар згортки, рівень активації та операцію шару об'єднання, щоб витягнути внутрішню функцію.

Вхідне зображення показане на рис. 2.6.

Кроки:

– імпортувати необхідні бібліотеки;

– встановити параметр;

– визначити ядро;

– завантажити зображення та нанести на нього графік;

– переформатувати зображення;

– застосувати операцію згорткового шару та нанести на графік вихідне зображення;

– застосувати операцію активаційного шару та нанести на графік вихідне зображення;

– застосувати операцію об'єднання шарів і нанести на графік вихідне зображення.

Рисунок 2.6 – Вхідне зображення

Вихідне зображення показане на рис. 2.7.

Переваги CNN [23]:

– добре виявляє закономірності та особливості на зображеннях, відео та аудіосигналах;

– стійкість до інваріантності перекладу, обертання та масштабування;

– наскрізне навчання, немає необхідності в ручному вилученні функцій;

– може обробляти великі обсяги даних і досягати високої точності.

Рисунок 2.7 – Вихідне зображення

Недоліки CNN:

– обчислювально дорогий для навчання і вимагає багато пам'яті;

– може бути схильний до перенавчання, якщо використовується недостатньо даних або правильна регуляризація;

– вимагає великих обсягів мічених даних;

– інтерпретація обмежена, важко зрозуміти, чому навчилася мережа.

2.3 Long Short-Term Memory

Long Short-Term Memory – це вдосконалена версія рекурентної нейронної мережі, розроблена компанією Hochreiter & Schmidhuber. LSTM чудово справляється із завданнями прогнозування послідовностей, фіксуючи довготривалі залежності. Ідеально підходить для часових рядів, машинного перекладу та розпізнавання мовлення через залежність від порядку [26].

Традиційний RNN має єдиний прихований стан, який передається в часі, що може ускладнити вивчення мережею довгострокових залежностей. Моделі

LSTM вирішують цю проблему шляхом введення комірки пам'яті, яка є контейнером, який може зберігати інформацію протягом тривалого періоду.

Архітектури LSTM здатні вивчати довготривалі залежності в послідовних даних, що робить їх добре придатними для таких завдань, як переклад мов, розпізнавання мови та прогнозування часових рядів.

LSTM також можна використовувати в поєднанні з іншими архітектурами нейронних мереж, такими як згорткові нейронні мережі (CNNs) для аналізу зображень і відео [27].

Архітектура LSTM включає комірку пам'яті, яка керується трьома вентилями: вхідним вентилем, затвором забуття та вихідним вентилем. Ці вентилі вирішують, яку інформацію додавати, видаляти з неї та виводити з комірки пам'яті.

Це дозволяє мережам LSTM вибірково зберігати або відкидати інформацію під час її проходження через мережу, що дозволяє їм вивчати довгострокові залежності [27].

LSTM підтримує прихований стан, який виступає в ролі короткочасної пам'яті мережі. Прихований стан оновлюється на основі вхідних даних, попереднього прихованого стану та поточного стану комірки пам'яті.

Двонаправлена модель LSTM (Bi LSTM/BLSTM) – це рекурентна нейронна мережа (RNN), яка здатна обробляти послідовні дані як у прямому, так і у зворотному напрямках. Це дозволяє Bi LSTM вивчати більш далекосяжні залежності в послідовних даних, ніж традиційні LSTM, які можуть обробляти послідовні дані тільки в одному напрямку.

Bi LSTM складаються з двох мереж LSTM, одна з яких обробляє вхідну послідовність у прямому напрямку, а інша – у зворотному напрямку. Потім виходи двох мереж LSTM об'єднуються для отримання кінцевого виходу.

Моделі LSTM, включаючи Bi LSTM, продемонстрували найсучаснішу продуктивність у різних завданнях, таких як машинний переклад, розпізнавання мови та підсумовування тексту [28].

Мережі в архітектурах LSTM можуть бути складені для створення глибо-

ких архітектур, що дозволяє вивчати ще більш складні шаблони та ієрархії в послідовних даних. Кожен шар LSTM у складеній конфігурації фіксує різні рівні абстракції та часових залежностей у вхідних даних.

Архітектура LSTM має ланцюгову структуру, яка містить чотири нейронні мережі та різні блоки пам'яті, які називаються комірками (рис. 2.8).

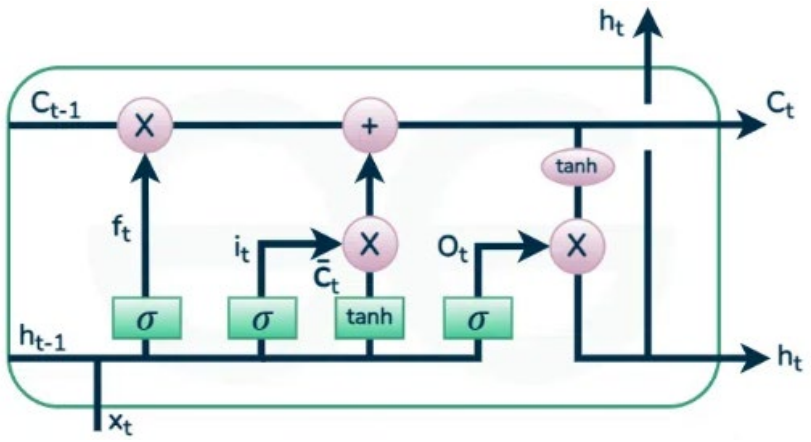

Рисунок 2.8 – Архітектура LSTM

Інформація, яка більше не є корисною в стані клітини, видаляється за допомогою Forget Gate. Два входи x_t (вхідні дані в конкретний момент часу) і h_{t-1} (попередній вихід комірки) подаються на вентиль і множаться на вагові матриці з подальшим додаванням зміщення. Отриманий продукт пропускається через функцію активації, яка дає двійковий вихід. Якщо для певного стану комірки вихід дорівнює 0, фрагмент інформації забувається, а для виходу 1 інформація зберігається для подальшого використання. Рівняння для Forget Gate має вигляд [29]:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f),$$

де W_f – матриця ваги, пов'язана з Forget Gate;

$[h_{t-1}, x_t]$ – конкатенація поточного входу та попереднього прихованого стану;

b_f – це упередженість з Forget Gate;

σ – функція активації.

Додавання корисної інформації до стану комірки здійснюється вхідним вентилем (рис. 2.9). Спочатку інформацію регулюють за допомогою сигмоїдної функції і фільтрують значення, що запам'ятовуються, аналогічно Forget Gate за допомогою входів h_{t-1} і x_t.

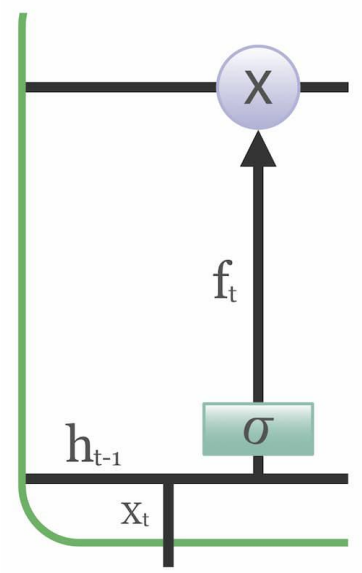

Рисунок 2.9 – Input gate

Потім за допомогою функції tanh створюється вектор, який дає на виході від -1 до $+1$, який містить всі можливі значення з h_{t-1} і x_t. Нарешті, значення вектора і регульовані значення перемножуються для отримання корисної інформації. Рівняння для вхідного вентиля має вигляд:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i),$$
$$C_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C).$$

Множимо попередній стан на f_t, не звертаючи уваги на інформацію, яку ми раніше вирішили проігнорувати. Далі ми включаємо $i_t \cdot C_t$. Це означає оновлені значення кандидатів з поправкою на суму, яку ми вибрали для оновлення кожного значення стану:

$$C_t = f_t \otimes C_{t-1} + i_t \otimes C_t,$$

де $\otimes$ позначає поелементне множення;

tanh – це функція активації tanh .

Завдання вилучення корисної інформації з поточного стану комірки, що підлягає представленню на виході, виконується вихідним вентилем (рис. 2.10). Спочатку вектор генерується шляхом застосування функції $\tanh(x)$ до клітини. Потім інформація регулюється за допомогою сигмоїдної функції і фільтрується за значеннями, які потрібно запам'ятати за допомогою входів h_{t-1} і x_t. Нарешті, значення вектора і регульовані значення перемножуються для відправки в якості виводу і входу в наступну комірку. Рівняння для вихідного вентиля має вигляд:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o).$$

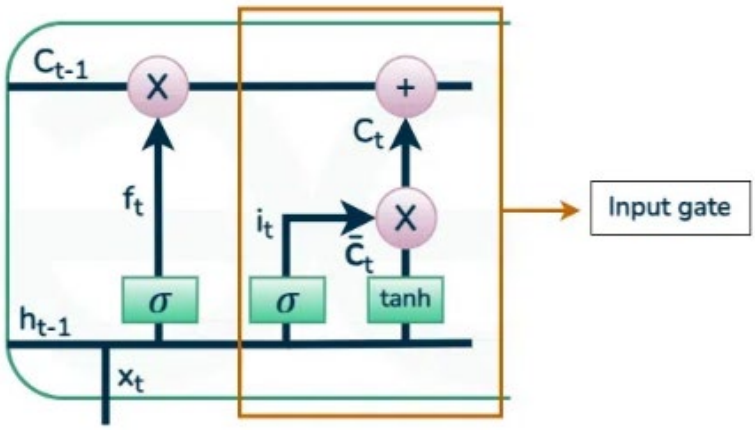

Рисунок 2.10 – Output gate

Деякі з відомих застосувань LSTM включають [29, 30]:

– моделювання мови: LSTM використовувалися для завдань обробки природної мови, таких як моделювання мови, машинний переклад і підсумовування тексту; їх можна навчити створювати зв'язні та граматично правильні речення, вивчаючи залежності між словами в реченні;

– розпізнавання мовлення: LSTM використовуються для завдань із розпізнавання мовлення, таких як транскрибування мовлення в текст і розпізнавання голосових команд; їх можна навчити розпізнавати закономірності в мові та зіставляти їх з відповідним текстом;

– прогнозування часових рядів: LSTM використовувалися для завдань прогнозування часових рядів, таких як прогнозування цін на акції, погоди та споживання енергії; вони можуть вивчати закономірності в даних часових рядів і використовувати їх для прогнозування майбутніх подій;

– виявлення аномалій: LSTM використовувалися для завдань із виявлення аномалій, таких як виявлення шахрайства та вторгнення в мережу; їх можна навчити виявляти закономірності в даних, які відхиляються від норми, і позначати їх як потенційні аномалії;

– рекомендаційні системи: LSTM використовувалися для рекомендаційних завдань, таких як рекомендація фільмів, музики та книг; вони можуть вивчати закономірності в поведінці користувачів і використовувати їх для надання персоналізованих рекомендацій;

– аналіз відео: LSTM використовувалися для завдань відеоаналізу, таких як виявлення об'єктів, розпізнавання активності та класифікація дій; їх можна використовувати в поєднанні з іншими архітектурами нейронних мереж, такими як згорткові нейронні мережі (CNN), для аналізу відеоданих та вилучення корисної інформації.

Висновки за розділом 2

У розділі було проаналізовано методи розпізнавання об'єктів на зображеннях і відео. Штучний інтелект є ключовою технологією для розпізнавання об'єктів на зображеннях і відео, що дозволяє не тільки ідентифікувати об'єкти, а й локалізувати їх у просторі. Ця технологія знаходить широке застосування у різних сферах, таких як безпілотні автомобілі, системи безпеки, роздрібна тор-

гівля та промисловість. Основними інструментами для розпізнавання є згорткові нейронні мережі, які здатні виділяти ключові особливості зображень, а також сучасні методи, такі як YOLO та SSD, які забезпечують високу швидкість і точність роботи в реальному часі.

Ефективність систем розпізнавання об'єктів значною мірою залежить від наявності великих обсягів даних з мітками, які використовуються для навчання моделей. Методи навчання під наглядом і глибоке навчання дозволяють автоматично вивчати складні функції та шаблони в даних, що сприяє підвищенню точності роботи моделей. Штучний інтелект активно використовується для оптимізації обліку товарів, аналізу поведінки клієнтів та підвищення ефективності у роздрібній торгівлі. У промисловості його застосовують для моніторингу безпеки працівників, забезпечення якості продукції та автоматизації виробничих процесів. Крім того, відеоаналітика з використанням ШІ дозволяє в реальному часі аналізувати потоки людей, оптимізувати простір та визначати пікові періоди активності клієнтів.

Разом із тим, використання таких систем пов'язане з викликами, зокрема потребою у значних обчислювальних ресурсах та високою залежністю від якості даних. Незважаючи на це, розвиток штучного інтелекту й надалі значно розширює можливості у сфері розпізнавання об'єктів та їх застосування у різних галузях.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Обґрунтування мови програмування

Python – це інтерпретована мова програмування загального призначення високого рівня, яка має простий синтаксис, який легко освоїти, і підкреслює легкість читання. Його в основному використовують професійні програмісти та розробники в різних галузях, включаючи розробку Інтернету та програмного забезпечення, машинне навчання, штучний інтелект, великі дані та складну математику. Як і всі інші мови програмування, Python також має свої плюси та мінуси [31].

Інтерпретованість означає, що інтерпретатор обробляє вихідний файл під час виконання, читає рядки коду один за одним і виконує те, що сказано. Подібно до Perl і PHP, Python не вимагає компіляції програми перед її виконанням. Отже, не потрібно викликати компілятор. Замість запуску компілятора, який допомагає перетворити вихідні файли на скомпільовані файли класів, просто запускається файл .py. Компіляція байт-коду Python є автоматичною та повністю неявною [31].

Високорівневість забезпечується тим, що Python спирається на легкочитані структури, які згодом перекладаються на мову низького рівня, оригінальний код, який виконується на центральному процесорі (CPU) комп'ютера. Мова високого рівня призначена для використання програмістом, а написаний код далі інтерпретується мовою низького рівня. Як і C++ або Java, перед запуском Python потрібно обробити. Це забезпечує портативність Python – він може працювати на різних типах комп'ютерів майже без модифікацій [31].

Python можна використовувати майже для всього. Він застосовний майже в усіх галузях для різноманітних завдань. Будь то виконання таких короткотермінових завдань, як тестування програмного забезпечення чи довгострокова розробка продукту, що передбачає планування дорожньої карти, Python добре працює для всіх них, він застосовний усюди. Його ролі необмежені. Він попу-

лярний не тільки серед інженерів-програмістів, а й серед фахівців інших галузей: математики, аналізу даних, науки, бухгалтерського обліку та мережевої інженерії [31].

Об'єктно-орієнтований підхід Python дозволяє мислити проблеми в термінах класів і об'єктів. Потім об’єкти компонуються таким чином, щоб складати складні комп’ютерні програми. Окрім об’єктно-орієнтованого програмування, Python також підтримує процедурну парадигму. Оскільки ООП є лише одним із варіантів, програмування на Python можна зробити більш просунутим, вибравши підхід до об’єктно-орієнтованого програмування. Розробники можуть створювати шаблони коду для повторного використання, таким чином зменшуючи надмірність у проєктах розробки [31].

У різних галузях існує велика різноманітність варіантів використання Python. Звичайно, перше, що спадає на думку, коли ми думаємо про найпоширеніші способи використання Python, це для створення веб-додатків, мобільних і настільних програм, а також для їх тестування. Але Python – це мова, яка служить багатьом цілям. Загалом, Python ідеально підходить для таких сфер використання [32]:

– розробка веб-додатків;

– наука про дані;

– сценарії;

– програмування бази даних;

– швидке створення прототипів.

Python підходить для всіх форм програмування, що сприяє швидкому зростанню бази користувачів. Скрипти міжплатформної оболонки, швидка автоматизація, проста веб-розробка, аналіз і візуалізація даних, штучний інтелект і машинне навчання – це деякі приклади.

Часто фахівці використовують Python для кращого виконання різноманітних завдань у різних дисциплінах. Кращої продуктивності, серед іншого, можна досягти за допомогою автоматизації. Фінанси, страхування та маркетинг є основними сферами, у яких люди стикаються з необхідністю виконувати повто-

рювані завдання: переглядати, копіювати, перейменовувати та завантажувати файли на сервер, завантажувати веб-сайти чи аналізувати дані. Натомість програміст може написати сценарій на Python і автоматизувати все це [32].

Крім того, не обов'язково бути розробником програмного забезпечення, щоб використовувати Python. Мова дозволяє полегшити аналіз і візуалізацію даних. Він має багату екосистему, що включає ефективні бібліотеки для обробки даних і, отже, допомагає спеціалістам із обробки даних у виконанні складних числових обчислювальних операцій.

Не дарма найбільші компанії світу використовують Python. Pixar використовує його для створення фільмів, Google – для сканування сторінок, Netflix – для доставки вмісту та Spotify – для рекомендації пісень. Ця мова має багато переваг, і є кілька вагомих причин. Розглянемо основні з них [33].

1. Простота. Простий і зрозумілий синтаксис Python спонукає початківців вивчати цю мову сценаріїв. Його код легко зрозуміти, поширювати та підтримувати. Немає багатослівності, мова легко вивчається.

2. Потужний інструментарій. За своєю суттю програми на Python є текстовими файлами, що містять інструкції для інтерпретатора та написані в текстовому редакторі або IDE. IDE є повнофункціональними та пропонують такі вбудовані інструменти, як перевірка синтаксису, налагоджувачі та браузери коду. Текстові редактори зазвичай не включають функції IDE, але їх можна налаштувати. Python також має величезний набір сторонніх пакетів, бібліотек і фреймворків, які полегшують процес розробки. Таким чином, ці можливості оптимізації роблять Python чудовим для великомасштабних проектів.

3. Швидкість розвитку. Тут мається на увазі швидкість бізнесу та показник часу виходу на ринок. Python – це динамічна мова сценаріїв, тому вона не призначена для написання програм з нуля, а в першу чергу призначена для підключення компонентів. Компоненти призначені для повторного використання, а інтерфейси між компонентами та сценаріями чітко визначені. Усе це прискорює розробку програмного забезпечення завдяки Python, що робить мову надзвичайно лаконічною та продуктивною.

4. Гнучкість. Хоча Python робить наголос на простоті та читабельності коду, а не на гнучкості, у цій мові це все одно є. Python можна використовувати в різних проєктах. Це дозволяє розробникам вибирати між об’єктно-орієнтованим і процедурним режимами програмування. Python також гнучкий у типі даних. Їх 5: число, рядок, список, кортеж і словник, і кожен тип підданих відповідає одному з цих кореневих типів. У результаті дослідницький аналіз даних стає легше проводити завдяки гнучкості Python.

5. Портативність. Python створено для переносимості. Його програми підтримуються на будь-якій сучасній комп'ютерній ОС. Завдяки високорівневому характеру мови сценарій Python інтерпретується, тому його можна написати для подальшої інтерпретації однаково добре в Linux, Windows, Mac OS і UNIX, не вимагаючи коригувань. Програми на Python також дозволяють реалізувати портативні GUI.

6. Сильна громада. Python має швидко зростаючу базу користувачів і фактично є репрезентативним прикладом сильної спільноти. Є тисячі учасників потужного інструментарію Python – Pythonists. В онлайн-сховище вже завантажено майже 200 000 програмних пакетів, створених на замовлення . Усе це означає, що велика підтримуюча спільнота є як причиною, так і наслідком попиту на мову.

Той факт, що Python має репутацію зручної для програмування мови, якій віддають перевагу розробники, безсумнівний, але час від часу Python порівнюють з іншими мовами програмування, включаючи Java, C#, PHP і Ruby on Rails. Однак порівняння дійсне, якщо взяти до уваги продуктивність, функціональність та всі інші адекватні показники обговорюваної пари.

Усі мови програмування мають свої недоліки. Незважаючи на всі переваги, які пропонує Python як мова програмування, є недоліки, якими слід скористатися [33]:

– швидкість як інтерпретована мова: цей недолік можна виправити з появою PyPy, яка обіцяє приріст продуктивності;

– динамізм Python запобігає виявленню семантичних помилок заздале-

гідь, але такі інструменти, як PyChecker, можуть перевіряти наявність помилок, що робив би компілятор таких мов, як C або Java;

– потоковість є менш продуктивною в Python, ніж в інших мовах; багатопотоковість може стати можливою з Jython, але незмінність не надто важлива в Python, тому однопотоковий паралелізм працює добре;

– залежність від сторонніх бібліотек і фреймворків: існує чимало широко використовуваних ресурсів сторонніх розробників, які по суті не є Pythonic, що фактично суперечить девізу Python.

3.2 Алгоритм розв’язання задачі розпізнавання об’єктів на відео

Виявлення об'єктів є популярним завданням комп'ютерного зору. Він має справу з локалізацією області інтересу в межах зображення та класифікацією цієї області як типовий класифікатор зображень. Одне зображення може включати кілька цікавих областей, що вказують на різні об'єкти. Це робить виявлення об'єктів більш складною проблемою класифікації зображень.

YOLO (You Only Look Once) – популярна модель виявлення об’єктів, відома своєю швидкістю та точністю. Вперше вона була представлена Джозефом Редмоном та ін. у 2016 році і з тих пір пройшла кілька ітерацій, останньою була YOLO v7 [34].

Виявлення об’єктів – це завдання комп’ютерного зору, яке передбачає ідентифікацію та визначення місцезнаходження об’єктів на зображеннях або відео. Це важлива частина багатьох додатків, таких як спостереження, безпілотні автомобілі чи робототехніка. Алгоритми виявлення об'єктів можна розділити на дві основні категорії: детектори одноразові та двоступеневі детектори.

Однією з перших успішних спроб вирішення проблеми виявлення об’єктів за допомогою глибокого навчання була модель R-CNN (регіони з функціями CNN), розроблена Россом Гіршиком та його командою з Microsoft Research у 2014 році. Ця модель використовувала комбінацію алгоритмів про-

позиції регіонів. і згорткові нейронні мережі (CNN) для виявлення та локалізації об’єктів на зображеннях.

Алгоритми виявлення об’єктів загалом класифікуються на дві категорії залежно від того, скільки разів те саме вхідне зображення передається через мережу (рис. 3.1).

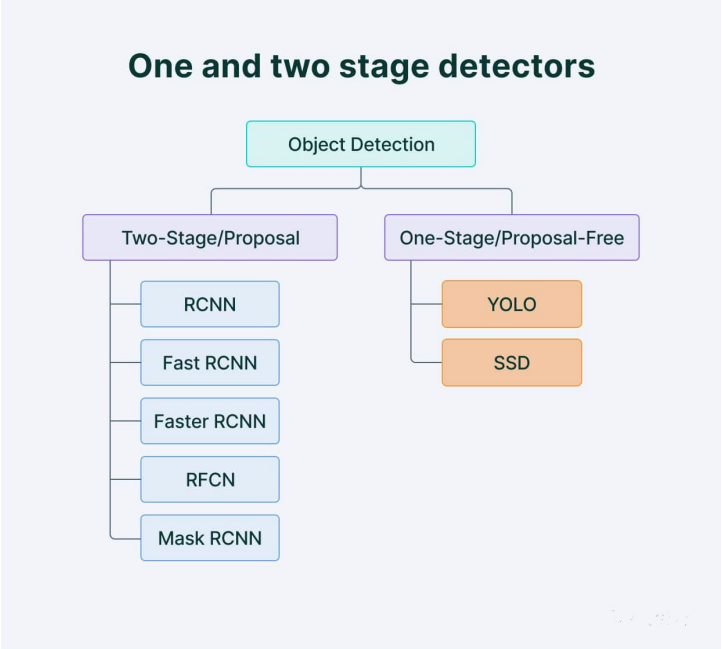

Рисунок 3.1 – Класифікація алгоритмів виявлення об’єктів

Одноразове виявлення об’єктів використовує один прохід вхідного зображення для прогнозування наявності та розташування об’єктів на зображенні. Він обробляє все зображення за один прохід, що робить їх обчислювально ефективними [34].

Однак одноразове виявлення об’єктів, як правило, менш точне, ніж інші методи, і менш ефективне для виявлення малих об’єктів. Такі алгоритми можна використовувати для виявлення об’єктів у реальному часі в середовищах з обмеженими ресурсами.

YOLO – одноразовий детектор, який використовує повністю згорткову нейронну мережу (CNN) для обробки зображення.

Дворазове виявлення об’єктів використовує два проходи вхідного зображення для прогнозування наявності та розташування об’єктів. Перший прохід

використовується для створення набору пропозицій або потенційних місць розташування об’єктів, а другий прохід використовується для уточнення цих пропозицій і створення остаточних прогнозів. Цей підхід більш точний, ніж одноразове виявлення об’єкта, але також дорожчий з точки зору обчислень.

Загалом, вибір між виявленням об’єктів одним або двома пострілами залежить від конкретних вимог і обмежень програми.

Як правило, однократне виявлення об’єктів краще підходить для програм у режимі реального часу, тоді як двократне виявлення об’єктів краще для програм, де точність важливіша [35].

Щоб визначити та порівняти ефективність прогнозування різних моделей виявлення об’єктів, потрібні стандартні кількісні показники.

Двома найпоширенішими показниками оцінювання є показники перетину через з’єднання (IoU) і показники середньої точності (AP).

Перетин через об’єднання є популярним показником для вимірювання точності локалізації та обчислення помилок локалізації в моделях виявлення об’єктів.

Щоб обчислити IoU між прогнозованою та наземною обмежувальними рамками, спочатку треба взяти площу перетину між двома відповідними обмежувальними рамками для того самого об’єкта. Після цього обчислюється загальна площа, охоплена двома обмежувальними прямокутниками, відома як «Об’єднання», і площа перекриття між ними, яка називається «Перетин».

Перетин, поділений на Union, дає відношення перекриття до загальної площі, забезпечуючи хорошу оцінку того, наскільки близько обмежувальна рамка передбачення знаходиться до початкової обмежувальної рамки (рис. 3.2) [35].

Середня точність (AP) обчислюється як площа під кривою точності та запам’ятовування для набору передбачень.

Відкликання обчислюється як відношення загальної кількості прогнозів, зроблених моделлю під класом, із загальною кількістю існуючих міток для кла-

су. Точність означає відношення справжніх позитивних результатів до загальної кількості прогнозів, зроблених моделлю.

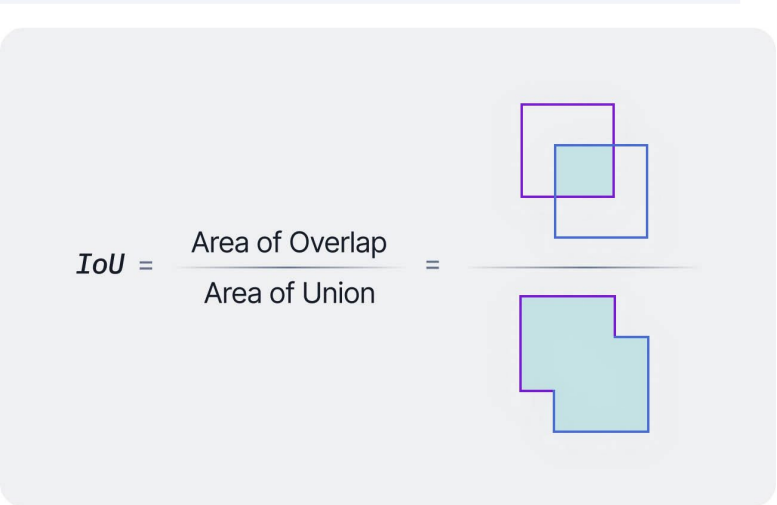

Рисунок 3.2 – Ілюстрація метрики Intersection over Union (IoU)

Пригадування та точність пропонують компроміс, який графічно представлений у вигляді кривої шляхом зміни порогу класифікації. Площа під цією кривою точності та запам’ятовування дає середню точність на клас для моделі. Середнє значення цього значення, прийняте для всіх класів, називається середньою середньою точністю (mAP) [36].

У виявленні об’єктів точність і відкликання не використовуються для передбачення класу. Натомість вони служать прогнозами граничних рамок для вимірювання продуктивності рішень. Значення IoU > 0,5 вважається позитивним прогнозом, тоді як значення IoU < 0,5 є негативним прогнозом.

You Only Look Once (YOLO) пропонує використовувати наскрізну нейронну мережу , яка робить прогнози обмежувальних рамок і ймовірностей класів одночасно. Він відрізняється від підходу, використаного попередніми алгоритмами виявлення об’єктів, які перепрофілювали класифікатори для виконання виявлення.

Дотримуючись принципово іншого підходу до виявлення об’єктів, YOLO досягла найсучасніших результатів, значно перевершивши інші алгоритми виявлення об’єктів у реальному часі.

У той час як такі алгоритми, як Faster RCNN, працюють шляхом виявлення можливих цікавих регіонів за допомогою мережі регіональних пропозицій, а потім виконують розпізнавання цих регіонів окремо, YOLO виконує всі свої передбачення за допомогою єдиного повністю підключеного рівня [36].

Методи, які використовують мережі регіональних пропозицій, виконують кілька ітерацій для одного зображення, тоді як YOLO обходиться лише однією ітерацією.

З моменту першого випуску YOLO в 2015 році було запропоновано кілька нових версій однієї моделі, кожна з яких базується на своїй попередниці та вдосконалює її.

Алгоритм YOLO приймає зображення як вхідні дані, а потім використовує просту глибоку згорткову нейронну мережу для виявлення об’єктів на зображенні (рис. 3.3).

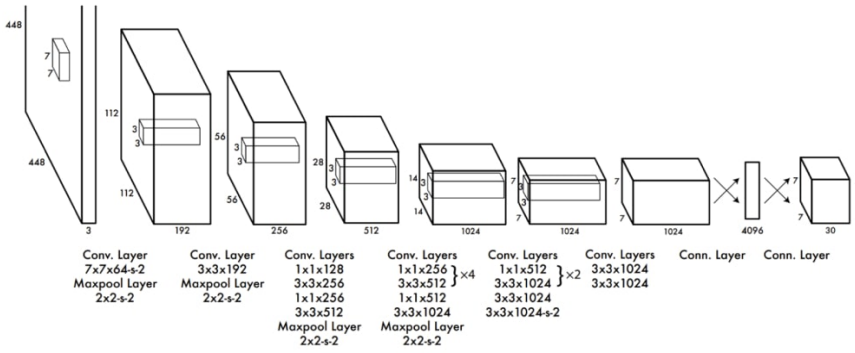

Рисунок 3.3 – Візуалізація роботи згорткової моделі

Перші 20 шарів згортки моделі попередньо навчені за допомогою ImageNet шляхом підключення тимчасового середнього пулу та повністю підключеного шару. Потім ця попередньо навчена модель перетворюється для виконання виявлення, оскільки попередні дослідження продемонстрували, що додавання згортки та підключених шарів до попередньо навченої мережі покра-

щує продуктивність. Останній повністю пов'язаний рівень YOLO передбачає як імовірності класу, так і координати обмежувальної рамки.

YOLO ділить вхідне зображення на сітку $S \times S$. Якщо центр об'єкта потрапляє в клітинку сітки, ця клітинка відповідає за виявлення цього об'єкта. Кожна клітинка сітки передбачає обмежувальні прямокутники B і оцінки достовірності для цих прямокутників. Ці оцінки достовірності відображають, наскільки модель впевнена в тому, що коробка містить об'єкт, і наскільки точна, на її думку, передбачена коробка [36].

YOLO передбачає кілька обмежувальних рамок на клітинку сітки. Під час навчання ми хочемо, щоб за кожен об'єкт відповідав лише один предиктор обмежувальної рамки. YOLO призначає один предиктор, який буде «відповідальним» за прогнозування об'єкта на основі прогнозу, який має найвищий поточний IOU з основною правдою. Це призводить до спеціалізації між предикторами обмежувальної рамки. Кожен предиктор стає кращим у прогнозуванні певних розмірів, пропорцій або класів об'єктів, покращуючи загальну оцінку запам'ятовування [36].

Одним із ключових методів, що використовується в моделях YOLO, є немаксимальне придушення (NMS). NMS – це етап постобробки, який використовується для підвищення точності та ефективності виявлення об'єктів. Під час виявлення об'єктів зазвичай для одного об'єкта на зображенні генеруються кілька обмежувальних рамок. Ці обмежувальні рамки можуть перекриватися або розташовуватися в різних положеннях, але всі вони представляють той самий об'єкт. NMS використовується для ідентифікації та видалення зайвих або неправильних обмежувальних рамок і для виведення окремої обмежувальної рамки для кожного об'єкта на зображенні.

3.3 Опис програми

В даній роботі використовується бібліотека Python під назвою ImageAI, яка дає змогу створювати програми та системи, які можуть виявляти об'єкти у

відео, використовуючи лише кілька рядків програмного коду. ImageAI підтримує YOLOv3, який є алгоритмом виявлення об’єктів.

Щоб почати, потрібно встановити кілька бібліотек Python і ImageAI**.** Якщо будь-яка із зазначених нижче бібліотек уже встановлена на комп’ютері, можна відразу перейти до встановлення ImageAI**.** Також потрібно переконатися, що на комп’ютері встановлена потрібна версія Python.

На рис. 3.4 показані необхідні для встановлення TensorFlow, ImageAI та інші бібліотеки.

```
pip3 install tensorflow==2.4.0

pip install keras==2.4.3 numpy==1.19.3 pillow==7.0.0 scipy==1.4.1 h5py==2.10.0
matplotlib==3.3.2 opencv-python keras-resnet==0.2.0

pip install imageai --upgrade
```

Рисунок 3.4 – Команди для встановлення необхідних бібліотек

Тепер, коли встановлені необхідні інструменти, використовуватиметься навчена модель комп’ютерного зору YOLOv3 для виконання завдань виявлення та розпізнавання. Ця модель навчена виявляти та розпізнавати 80 різних об’єктів: людина, велосипед, автомобіль, мотоцикл, літак, автобус і багато інших.

Створюємо файл Python і даємо йому назву наприклад FirstVideoDetection.py. Копіюємо завантажене відео та файл моделі YOLOv3 у папку, де знаходиться FirstVideoDetection.py. Код програми представлено на рис. 3.5. Перш ніж запустити код Python, варто надати деякі пояснення:

а) у четвертому рядку створено екземпляр класу VideoObjectDetection;

б) у п’ятому рядку встановлюється тип моделі YOLOv3, що відповідає моделі YOLO, яка завантажена та скопійована в папку;

в) у шостому рядку встановлено шлях моделі до шляху файлу моделі, який скопійовано в папку;

г) у сьомому рядку завантажена модель у створений екземпляр класу VideoObjectDetection;

д) у восьмому рядку викликається функція detectObjectsFromVideo з наступними значеннями:

1) input_file_path: стосується шляху до файлу відео, що скопійовано в необхідну папку;

2) output_file_path: стосується шляху до файлу, до якого буде збережено виявлене відео;

3) frames_per_second: стосується кількості кадрів зображення, які потрібно мати у виявленому відео протягом секунди;

4) log_progress: використовується для вказівки, що екземпляр виявлення повинен повідомляти про хід виявлення в інтерфейсі командного рядка;

е) в результаті функція detectObjectsFromVideo поверне шлях до файлу виявленого відео, який буде надруковано в дев'ятому рядку коду після виконання завдання виявлення.

```
from imageai.Detection import VideoObjectDetection
import os

execution_path = os.getcwd()

detector = VideoObjectDetection()
detector.setModelTypeAsYOLOv3()
detector.setModelPath( os.path.join(execution_path , "yolo.h5"))
detector.loadModel()

video_path = detector.detectObjectsFromVideo(input_file_path=os.path.join(
execution_path, "traffic-mini.mp4"),
                                output_file_path=os.path.join(execution_path,
"traffic_mini_detected_1")
                                , frames_per_second=29, log_progress=True)
print(video_path)
```

Рисунок 3.5 – Код програми з використанням ImageAI

Тепер, коли зміст коду розшифровано, можна запустити його та спостерігати за прогресом в інтерфейсі командного рядка, доки він не буде готовий.

Висновки за розділом 3

У розділі обґрунтовано вибір мови Python для програмної реалізації задач розпізнавання об'єктів на відео завдяки його простоті, універсальності та широкому набору бібліотек, які дозволяють ефективно працювати із задачами комп'ютерного зору. Використання таких інструментів, як TensorFlow, OpenCV та ImageAI, спрощує процес розробки програм та реалізацію алгоритмів машинного навчання.

Реалізовано програму для розпізнавання об'єктів на відео, яка базується на алгоритмі YOLOv3 та бібліотеці ImageAI. Алгоритм YOLO (You Only Look Once) визнаний одним із найефективніших для розв'язання задач виявлення об'єктів у реальному часі. Він забезпечує високу швидкість і точність, що робить його придатним для багатьох прикладних задач, особливо у випадках з обмеженими ресурсами.

Завдяки цьому вдалося створити систему, здатну виконувати розпізнавання об'єктів у відеопотоці з високою ефективністю, що підтверджено її успішним тестуванням.

4 РЕЗУЛЬТАТИ ОБЧИСЛЮВАЛЬНОГО ЕКСПЕРИМЕНТУ ТА ЇХ АНАЛІЗ

4.1 Опис експерименту

Для налаштування алгоритму, потрібні наступні дані:

– конфігураційний файл yolov3.cfg;

– файл ваги yolov3.weights;

– мітки класів coco.names;

– поріг впевненості (confidence threshold): 50%;

– параметр Non-Maximum Suppression (NMS): 40%.

Для демонстрації роботи алгоритму YOLO розглянемо декілька прикладів його роботи на реальних відео.

Розглянемо приклад вуличного відео, на якому присутні об’єкти, які можна ідентифікувати (рис. 4.1).

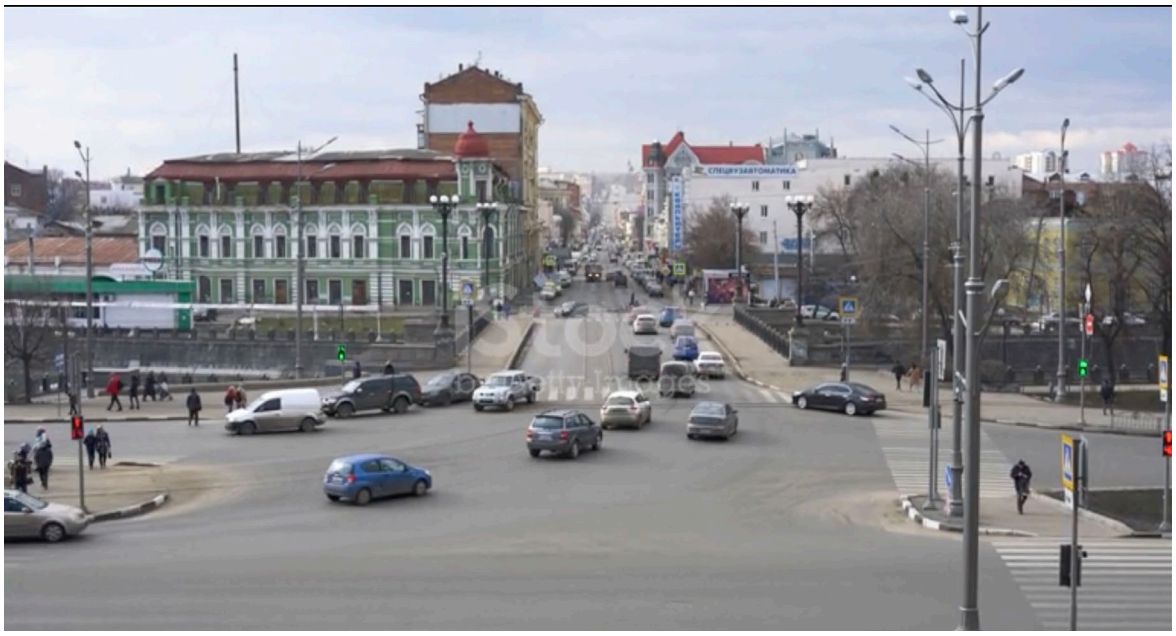

Рисунок 4.1 – Знімок екрану відео для ідентифікації об’єктів

Результат роботи алгоритму YOLO показано на рис. 4.2.

Рисунок 4.2 – Результат роботи алгоритму по виявленню об’єктів на відео

Також, даний алгоритм може бути налаштований на виявлення конкретних видів об’єктів, які одночасно перебувають на відео (рис. 4.3).

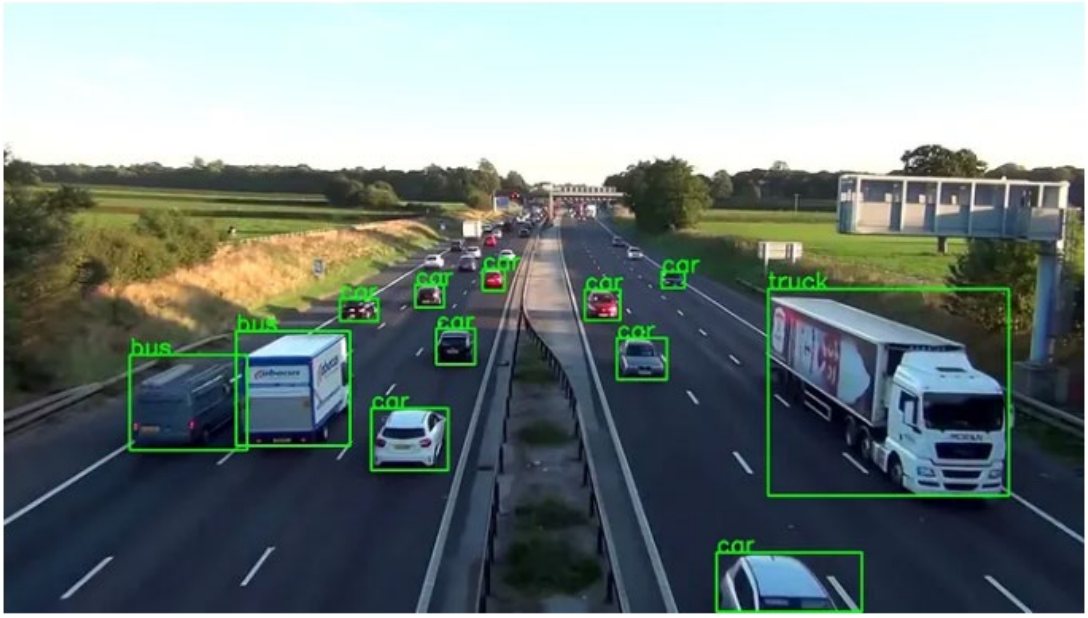

Рисунок 4.3 – Ідентифікація конкретних видів одного і того самого об’єкту

На рис. 4.4 наведено порівняння швидкості роботи алгоритму YOLO в порівнянні з іншими алгоритмами [33].

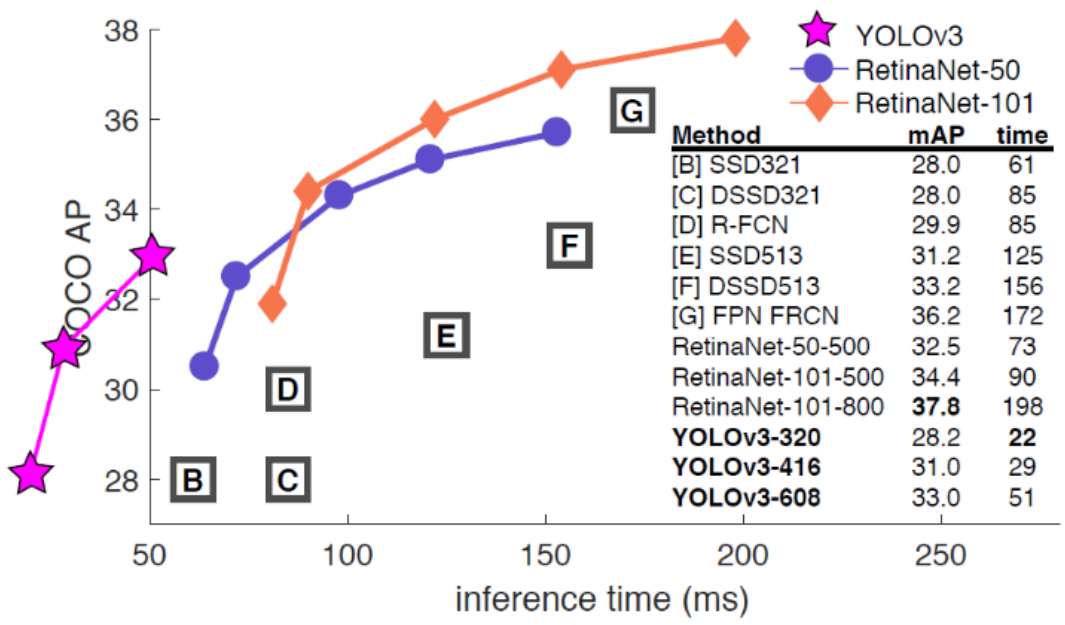

Рисунок 4.4 – Порівняння швидкості роботи алгоритму YOLO в порівнянні з іншими алгоритмами виявлення об’єктів

У табл. 4.1 показано середню точність (AP) виявлення малих, середніх і великих зображень за допомогою різних алгоритмів і основ [34]. Чим вищий AP, тим точніший він для цієї змінної.

Таблиця 4.1 – Порівняння показника AP для різних методів виявлення об’єктів

Algorithm	backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Faster R-CNN+++	ResNet-101-C4	34,9	55,7	37,4	15,6	38,7	50,9
Faster R-CNN w FPN	ResNet-101-FPN	36,2	59,1	39,0	18,2	39,0	48,2
Faster R-CNN by G-RMI	Interception-ResNet-v2	34,7	55,5	36,7	13,5	38,1	52,0
Faster R-CNN w TDM	Interception-ResNet-v2-TDM	36,8	57,7	39,2	16,2	39,8	52,1
YOLOv2	DarkNet-19	21,6	44,0	19,2	5,0	22,4	35,5
SSD513	ResNet-101-SSD	31,2	50,4	33,3	10,2	34,5	49,8
DSSD513	ResNet-101-DSSD	33,2	53,3	35,2	13,0	35,4	51,1
RetinaNet	ResNet-101-FPN	39,1	59,1	42,3	21,8	42,7	50,2
RetinaNet	ResNetXt-101-FPN	40,8	61,1	44,1	24,1	44,2	51,2
YOLOv3 608x608	Darknet-53	33,0	57,9	34,4	18,3	35,4	41,9

У процесі експерименту були виконані наступні дії:

– виконувалася детекція об'єктів на кожному кадрі відео;

– оцінювалися основні характеристики, зокрема точність визначення класу об'єкта, локалізація рамок та швидкодія моделі;

– результати аналізувалися візуально та за допомогою метрик.

4.2 Аналіз роботи алгоритму

Результати роботи алгоритму оцінювались за наступними параметрами (метриками).

1. Точність детекції об'єктів (Precision) [31]:

$$\text{Precision} = \frac{TP}{TP + FP},$$

де TP (True Positives) – кількість правильно виявлених об'єктів;

FP (False Positives) – кількість помилково виявлених об'єктів.

2. Повнота детекції (Recall) [31]:

$$\text{Recall} = \frac{TP}{TP + FN},$$

де TP (True Positives) – кількість правильно виявлених об'єктів;

FN (False Negatives) – кількість невиявлених об'єктів.

3. Якість локалізації рамок (IoU - Intersection over Union) [32]:

$$IoU = \frac{\text{Площа перетину рамок}}{\text{Площа об'єднання рамок}}.$$

4. Швидкість обробки оцінювалася у кількості кадрів за секунду (FPS – Frames Per Second).

Після проведення експерименту, були отримані наступні результати:

1. Точність моделі:

– Precision: 92%;

– Recall: 88%;

– IoU: 0.75 (середнє значення).

2. Швидкодія: на апаратному забезпеченні з процесором Intel Core i5 та відеокартою NVIDIA RTX 4060 TI швидкість склала 27 FPS.

3. Візуальна оцінка: модель коректно розпізнавала об'єкти різних класів (люди, автомобілі). Проблеми виникали при перекритті об'єктів або слабкому освітленні.

Ще кращих результатів можна досягти використовуючи перенавчання моделі на спеціалізованих даних, а саме на подібних даних тим, з якими модель буде використовуватись. Для подальшого підвищення якості можна використовувати більш сучасні моделі (наприклад, YOLOv5) або адаптивні пороги NMS. Також попередня обробка даних може покращити результат.

Після проведеного експерименту, можна з точністю сказати, що алгоритм YOLO є ефективним інструментом для виявлення об’єктів на відео даних, і має чудовий баланс між точністю виявлення та швидкості обробки.

Висновки за розділом 4

У розділі представлено результати обчислювального експерименту, який охоплював аналіз роботи алгоритму YOLOv3 на реальних даних. Результат експерименту демонструє високу точність і швидкість у задачах виявлення об'єктів, роблячи її придатною для задач, де виявлення об’єктів є необхідним.

Застосування попередньо навчених ваг дозволило значно скоротити час навчання та досягти високих результатів навіть на обмежених обсягах даних.

Графічні матеріали та числові результати підтверджують, що підхід YOLOv3 є ефективним рішенням для задач виявлення об'єктів у реальному часі. Подальші покращення можуть бути досягнуті завдяки оптимізації аугментації даних, додатковій обробці вхідних зображень та використанню методів післяобробки результатів.

Також експеримент підтвердив, що алгоритм YOLO має чудовий баланс між швидкістю та точністю, а той факт що YOLO активно розвивається, робить його перспективним вибором для широкого спектра застосувань, включаючи системи реального часу, автономні транспортні засоби та системи відеоспостереження.

ВИСНОВКИ

У кваліфікаційній роботі було розглянуто проблему побудови моделі для точного та швидкого розпізнавання об’єктів на відео. У процесі роботи було проведено огляд сучасних алгоритмів комп’ютерного зору та методів глибокого навчання, таких як YOLO (You Only Look Once), які забезпечують високу швидкодію та точність у реальному часі. Результати аналізу показали, що сучасні методи, зокрема YOLO та його модифікації, мають значний потенціал для розв'язання задач комп’ютерного зору, перевершуючи класичні підходи.

В експериментальній частині було створено модель на основі архітектури YOLO та протестовано її на відкритих наборах даних. Отримані результати свідчать про можливість ефективного використання запропонованої моделі для розпізнавання об’єктів у реальному часі, забезпечуючи точність понад 90% при зменшенні затримки обробки кадрів. Було також виявлено, що якість роботи моделі залежить від якості навчальних даних та налаштування гіперпараметрів.

Результати проведених експериментів можуть бути використані в таких сферах, як відеоспостереження, системи безпеки, транспортні системи, автономні дрони та інші прикладні задачі комп'ютерного зору. Запропонована модель демонструє перспективи використання як у середовищах з обмеженими обчислювальними ресурсами, так і в системах із високими вимогами до продуктивності.

У майбутньому дослідженні можливе подальше вдосконалення моделі шляхом використання більш складних модифікацій YOLO (наприклад, YOLOv8), збільшення обсягу навчальної вибірки, застосування методів ансамблю моделей для підвищення точності, адаптації моделі для роботи з багатомовними наборами даних або спеціалізованими доменами, оптимізації моделей для обробки відео на пристроях з низькою продуктивністю.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. DeepFace: Closing the Gap to Human-Level Performance in Face Verification / Y. Taigman, M. Yang, M. A. Ranzato, L. Wolf. *2014 IEEE Conference on Computer Vision and Pattern Recognition*. Columbus, 2014. P. 1701–1708. https://doi.org/10.1109/CVPR.2014.220

2. Аналіз існуючих підходів до розпізнавання. URL: https://habr.com/ru/company/synesis/blog/238129 (дата звернення: 25.11.2024).

3. Lubchenko V., Podvalnyi Y. Finding and classification of objects. *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE. 2018. P. 466–475.

4. Deep learning in video multi-object tracking: A survey / G. Ciaparrone, F. L. Sánchez, S. Tabik [et al.]. *Neurocomputing*. 2020. V. 381. P. 61–88.

5. Lubchenko V., Podvalnyi Y. Tools for following (tracking) objects. *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE. 2018. P. 466–475.

6. Deep network flow for multi-object tracking / S. Schulter, P. Vernaza, W. Choi, M. Chandraker. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017. P. 2730–2739.

7. Online multi-object tracking using CNN-based single object tracker with spatial-temporal attention mechanism / Q. Chu, W. Ouyang, H. Li [et al.]. *In Proceedings of the IEEE international conference on computer vision*. 2017. P. 4836–4845.

8. Multiobject tracking in videos based on lstm and deep reinforcement learning / M. X. Jiang, C. Deng, Z. G. Pan [et al.]. *Complexity*. 2018. URL: https://onlinelibrary.wiley.com/doi/10.1155/2018/4695890 (дата звернення: 19.11.2024).

9. Jiao, L., Zhang, F., Liu, F., Yang, S., Li, L., Feng, Z., & Qu, R. (2019). A survey of deep learning-based object detection. IEEE access, 7, 128837-128868. URL: https://arxiv.org/abs/1907.09408 (дата звернення: 10.12.2024)

10. Bullinger S., Bodensteiner C., Arens M. Instance flow based online multiple object tracking. *In 2017 IEEE International Conference on Image Processing (ICIP) IEEE*, 2017. P. 785–789.

11. Heterogeneous association graph fusion for target association in multiple object tracking / H. Sheng, Y. Zhang, J. Chen [et al.]. *IEEE Transactions on Circuits and Systems for Video Technology*. 2018. V. 29 (11). P. 3269–3280.

12. Enhancing detection model for multiple hypothesis tracking / J. Chen, H. Sheng, Y. Zhang, Z. Xiong. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2017. P. 18–27.

13. Wojke N., Bewley A., Paulus D. Simple online and realtime tracking with a deep association metric. *In 2017 IEEE international conference on image processing (ICIP). IEEE*. 2017. P. 3645–3649.

14. Kim S. J., Nam J. Y., Ko B. C. (2018). Online tracker optimization for multi-pedestrian tracking using a moving vehicle camera. IEEE Access, 6, pp. 48675-48687. URL: https://ieeexplore.ieee.org/abstract/document/8449934 (дата звернення: 29.12.2024).

15. Recurrent autoregressive networks for online multi-object tracking / K. Fang, Y. Xiang, X. Li, S. Savarese. *In 2018 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE*. 2018. P. 466–475.

16. Faster R-CNN: towards real-time object detection with region proposal networks / S. Ren, K. He, R. Girshick, J. Sun. *IEEE transactions on pattern analysis and machine intelligence*. 2016. V. 39 (6). P. 1137–1149.

17. Faster RYCNN: towards Real Time Object Detection with Region Proposal Networks / S. Ren, K. He, R. Girshick, J. Sun. *IEEE transactions on pattern analysis and machine intelligence*. 2017. V. 39 (6). PP. 1137–1149.

18. Lu Y., Lu C., Tang C. K. Online video object detection using association LSTM. *Proceedings of the IEEE International Conference on Computer Vision*. 2017. P. 2344–2352.

19. Beyond pixels: Leveraging geometry and shape cues for online multi-object tracking / S. Sharma, J. A. Ansari, J. K. Murthy, K. M. Krishna. *2018 IEEE*

International Conference on Robotics and Automation (ICRA). IEEE. 2018. P. 3508–3515.

20. Trajectory factory: Tracklet cleaving and re-connection by deep siamese bi-gru for multiple object tracking / C. Ma, C. Yang, F. Yang [et al.]. *2018 IEEE International Conference on Multimedia and Expo (ICME). IEEE*. 2018. P. 1–6.

21. Alvar S. R., Bajić I. V. MV-YOLO: Motion vectoraided tracking by semantic object detection. *2018 IEEE 20th International Workshop on Multimedia Signal Processing (MMSP). IEEE*. 2018. P. 1–5.

22. Hossain S., Lee D. J. Deep learning-based real-time multipleobject detection and tracking from aerial imagery via a flying robot with GPU-based embedded devices. *Sensors*. 2019. V. 19 (15). P. 3371.

23. Алгоритми детекції в відеопотоці. URL: https://masters.donntu.org/2018/fknt/konoshenko/library/modifitsirovannyy-algoritm-detektsii-lits-v-videopotoke-i-ego-programmnaya realizatsiya.pdf (дата звернення: 03.12.2024).

24. Bernardin K., Elbs A., Stiefelhagen R. Multiple object tracking performance metrics and evaluation in a smart room environment. URL: https://core.ac.uk/download/pdf/197559396.pdf (дата звернення: 12.12.2024).

25. Microsoft coco: Common objects in context / T. Y. Lin, M. Maire, S. Belongie [et al.]. *European conference on computer vision*. 2014. P. 740–755.

26. You only look once: Unified, real-time object detection / J. Redmon, S. Divvala, R. Girshick, A. Farhadi. *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016. P. 779–788.

27. Redmon J., Farhadi A. (2018). Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767. URL: https://arxiv.org/abs/1804.02767 (дата звернення: 08.12.2024).

28. Bochkovskiy A., Wang C. Y., Liao H. Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934. URL: https://arxiv.org/abs/2004.10934 (дата звернення: 23.12.2024).

29. Zhou, X., Wang, D., & Krähenbühl, P. (2019). Objects as points. arXiv preprint arXiv:1904.07850. URL: https://arxiv.org/abs/1904.07850 (дата звернення:

25.12.2024).

30. Алгоритми та засоби автоматичної детекції обличчя в відео потоці / Ю. Р. Шкіра, Н. І. Гевко, Н. Г. Гавриків, О. Я. Осадчук. *Кібербезпека та комп'ютерно-інтегровані технології (КБКІТ-2020)* : збірник матеріалів науково-практичної конференції молодих вчених, аспірантів та студентів (м. Тернопіль). 2020. С.19.

31. Decoding the Confusion Matrix. URL: https://ai.plainenglish.io/decoding-the-confusion-matrix-d5c543ded6bb (дата звернення: 05.01.2025).

32. Методи оцінки ефективності моделей виявлення об'єктів у комп'ютерному зорі / Д. К. Марчук, М. С. Граф. ВІСНИК ХНТУ № 2(85), 2023. Р. 181–186.

33. Focal loss for dense object detection. URL: https://www.slideshare.net/slideshow/focal-loss-for-dense-object-detection/222059896 (дата звернення: 20.01.2020).

34. Understanding the mAP (mean Average Precision) Evaluation Metric for Object Detection. URL: https://pylessons.com/YOLOv3-TF2-mAP (дата звернення: 15.07.2020).