Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет комп'ютерних наук (або центр післядипломної освіти, або навчально-науковий центр заочної форми навчання)
(повна назва)

Кафедра програмної інженерії
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження ефективності сучасних методів глибокого навчання для визначення положення тіла людини на відеопотоці з БПЛА
(тема)

Виконав:
студент (ка) 2 курсу, групи ІПЗм-22-5

Токарєв Д.В.
(прізвище, ініціали)

Спеціальність 121 – Інженерія програмного забезпечення
(код і повна назва спеціальності)
Тип програми освітньо-наукова

Керівник професор Білоус Н.В.
(посада, прізвище, ініціали)

Допускається до захисту
Зав. кафедри _____ З.В.Дудар
(підпис) (прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет комп’ютерних наук (або центр післядипломної освіти, або навчально-науковий центр заочної форми навчання)
Кафедра програмної інженерії
Рівень вищої освіти другий (магістерський)
Спеціальність 121 – Інженерія програмного забезпечення
Тип програми освітньо-наукова програма
Освітня програма Інженерія програмного забезпечення
(шифр і назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«___» _____ 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Токарєву Дмитру Володимировичу
(прізвище, ім’я, по батькові)

1. Тема роботи «Дослідження ефективності сучасних методів глибокого навчання для визначення положення тіла людини на відеопотоці з БПЛА»
Затверджена наказом по університету від 29.03. 2024р. № 250 Ст
2. Термін подання студентом роботи до екзаменаційної комісії 20.06.2024
3. Вихідні дані до роботи теоретичні відомості про згорткові методи розпізнавання образів, матеріали конференції, бібліотека Darknet, бібліотека OpenCV, мова програмування Python, зображення з виділеними знайденими об’єктами
4. Перелік питань, що потрібно опрацювати в роботі
теоретичний огляд, методи глибокого навчання для визначення положення тіла, аналіз методу yolo для розпізнавання об'єктів на зображеннях, результати експериментального дослідження методу yolo для класифікації об'єктів

КАЛЕНДАРНИЙ ПЛАН

Номер	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Видача завдання	01.10.2024	виконано
2	Аналіз предметної галузі	15.10.2024	виконано
3	Постановка задачі	10.11.2024	виконано
4	Експериментальні дослідження	02.04 – 20.04.24	*виконано*
5	Аналіз результатів експериментальних досліджень та розробка рекомендацій	20.04 – 23.04.24	*виконано*
6	Написання та оформлення статті та тез доповіді	17.04 – 23.04.24	*виконано*
7	Підготовка пояснювальної записки	01.04 – 26.04.24	*виконано*
8	Підготовка презентації та доповіді	26.04 – 2.05.24	*виконано*
9	Нормоконтроль	3.05 – 08.05.24	*виконано*
10	Рецензування	08.05 – 14.05.24	*виконано*
11	Занесення диплома в електронний архів	15.05.2024	*виконано*
12	Попередній захист	15.05.2024	*виконано*
13	Допуск до захисту у зав. кафедри	18.05.2024	*виконано*

Дата видачі завдання «01» жовтня 2024 р.

Студент Токарєв Д.В. _____ Токарєв Д.В.

(підпис)

Керівник роботи _____ проф. Білоус Н.В.

(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ / ABSTRACT

Пояснювальна записка містить 76ст., 38 рис., 2 табл., 30 джерел.

ВІДЕОПОТІК, ВІДСТЕЖЕННЯ ОБ'ЄКТІВ, ВИЗНАЧЕННЯ ПОЛОЖЕННЯ ТІЛА ЛЮДИНИ, ГЛИБОКЕ НАВЧАННЯ, КОНВОЛЮЦІЙНІ НЕЙРОННІ МЕРЕЖІ (CNN), НАБІР ДАНИХ, ТОЧНІСТЬ ВІДСТЕЖЕННЯ, ТРАНСФОРМЕРИ, РЕКУРЕНТНІ НЕЙРОННІ МЕРЕЖІ (RNN).

Об'єктом дослідження є система відстеження об'єктів на відеопотоці, отриманій з безпілотного літального апарату (БПЛА).

Метою дослідження є визначення ефективності сучасних методів глибокого навчання для визначення положення тіла людини на відеопотоці з БПЛА. Головним завданням є з'ясування, які моделі та методи глибокого навчання найбільш точно та продуктивно визначають положення об'єктів.

Для вирішення поставленої мети використовується метод глибокого навчання. Різні архітектури, такі як конволюційні нейронні мережі (CNN), рекурентні нейронні мережі (RNN) та трансформери, досліджуються щодо їхньої ефективності. Застосовуються відповідні набори даних для навчання та тестування моделей.

В результаті дослідження визначено, що конволюційні нейронні мережі виявилися найбільш ефективними для визначення положення тіла людини на відеопотоці з БПЛА. Моделі глибокого навчання демонструють високу точність відстеження ключових точок тіла при оптимальних параметрах навчання. Результати дослідження можуть бути використані для подальшого розвитку систем відстеження на основі безпілотних літальних апаратів з використанням передових технологій глибокого навчання.

CNN, DATA SET, DEEP LEARNING, OBJECT TRACKING, POSITION DETERMINATION OF A HUMAN BODY, RNN, TRACKING ACCURACY, TRANSFORMERS, VIDEO STREAM.

The object of the research is a system of tracking objects on a video stream obtained from an unmanned aerial vehicle (UAV).

The purpose of the study is to determine the effectiveness of modern deep learning methods for determining the position of the human body on a video stream from a UAV. The main task is to find out which deep learning models and methods most accurately and productively determine the position of objects.

The method of deep learning is used to solve the set goal. Different architectures such as Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN) and Transformers are investigated for their performance. Appropriate datasets are used to train and test the models.

As a result of the study, it was determined that convolutional neural networks were the most effective for determining the position of the human body on the video stream from the UAV. Deep learning models demonstrate high accuracy of tracking key points of the body with optimal learning parameters. The results of the research can be used for the further development of tracking systems based on unmanned aerial vehicles using advanced deep learning technologies.

Заява щодо самостійного виконання кваліфікаційної роботи та можливості її публікації в електронному архіві відкритого доступу ElArKhNURE.

Я, Токарєв Дмитро Володимирович, студент гр. ІПЗм-22-5, здобувач вищої освіти на другому (магістерському) рівні кафедри «Програмна інженерія», заявляю: моя кваліфікаційна робота на тему «Дослідження ефективності сучасних методів глибокого навчання для визначення положення тіла людини на відеопотоці з БПЛА», що буде представлена в екзаменаційну комісію для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу ElArKhNURE. Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

ЗМІСТ

Вступ 9
1 Теоретичний огляд 10
1.1 Основні принципи глибокого навчання 10
1.2 Застосування глибокого навчання в комп'ютерному зорі та аналізі відео ... 11
1.3 Використання безпілотних літальних апаратів (БПЛА) в медіцині, безпеці та інших галузях 12
1.4 Проблеми та виклики у визначенні положення тіла людини на відеопотоці з БПЛА 14
1.5 Постановка задачі 16
2 Методи глибокого навчання для визначення положення тіла 17
2.1 Класифікація методів глибокого навчання 17
2.2 Архітектури нейронних мереж для визначення положення тіла 23
2.3 Використання конволюційних нейронних мереж (CNN) та рекурентних нейронних мереж (RNN) 24
3 Аналіз методу yolo для розпізнавання об'єктів на зображеннях 28
3.1 Принципи створення сучасних засобів розпізнавання на основі нейронних мереж 28
3.2 Підхід YOLO 34
3.3 Архітектурні та навчальні впорядкування YOLO 43
4 Результати експериментального дослідження методу yolo для класифікації об'єктів 47
4.1 Програмне середовище 47
4.2 Результати програмної моделі 47
Висновки 58
Перелік джерел посилання 59
Додаток А 63
Додаток Б 64
Додаток В 65
Додаток Г 74

Додаток Д 76

ВСТУП

В сучасному світі використання безпілотних літальних апаратів (БПЛА) знаходить все більше застосувань в різних галузях, включаючи моніторинг, розвідку, агрокультуру та безпеку. Однією з ключових задач використання БПЛА є ефективне відстеження та аналіз об'єктів на землі, зокрема, визначення положення тіла людини. Ця задача має велике значення в контексті пошуку та рятування, нагляду за територією, а також в інших областях, де важливо визначити точне положення об'єктів або осіб.

З розвитком глибокого навчання та штучного інтелекту виникають нові можливості для розв'язання складних завдань відстеження об'єктів на відеопотоці. Сучасні архітектури глибокого навчання, такі як конволюційні нейронні мережі (CNN), рекурентні нейронні мережі (RNN) та трансформери, дозволяють автоматизувати процес визначення положення тіла людини з великою точністю.

Ця робота спрямована на дослідження ефективності сучасних методів глибокого навчання для визначення положення тіла людини на відеопотоці, отриманої з безпілотного літального апарату. У процесі дослідження буде проведено аналіз різних архітектур глибокого навчання, визначено оптимальні параметри навчання та проведено порівняння результатів з існуючими методами відстеження об'єктів.

Мета роботи - визначити найбільш ефективні методи глибокого навчання для відстеження положення тіла людини на відеопотоці з БПЛА та визначити їхню застосовність у практичних сценаріях. Результати цього дослідження можуть виявити велике значення для розробки систем відстеження на основі БПЛА з використанням передових технологій глибокого навчання.

1 ТЕОРЕТИЧНИЙ ОГЛЯД

1.1 Основні принципи глибокого навчання

Глибоке навчання – це галузь машинного навчання, яка використовує нейронні мережі з багатьма шарами (глибокими мережами) для вирішення складних завдань [1].

Нейронні мережі, які також називають штучними нейронними мережами (ANN) або імітованими нейронними мережами (SNN), є підмножиною машинного навчання та є основою алгоритмів глибокого навчання. Їх називають «нейронними», оскільки вони імітують те, як нейрони в мозку передають сигнали один одному. Люди отримують багато ідей від природи – як літаки, які базувалися на птахах. Наприклад, літаки були винайдені натхненними птахами. Нейронна мережа – це тип алгоритму машинного навчання, створеного людським мозком. Це мережа взаємопов’язаних вузлів або штучних нейронів, які вчаться розпізнавати шаблони в даних.

Штучні нейронні мережі (ШНМ) – це обчислювальні моделі, натхненні структурою та функціонуванням біологічних нейронних мереж у мозку людини. ШНМ складаються з взаємопов’язаних вузлів або «нейронів», які обробляють і передають інформацію.

Ця техніка вирішує складні проблеми машинного навчання, такі як класифікація зображень, системи рекомендацій і переклад з однієї мови на іншу.

Термін «Глибоке навчання» стосується навчання нейронних мереж, іноді дуже великих нейронних мереж.

Глибоке навчання означає процес залучення системи, яка мислить і навчається точно так само, як люди, за допомогою штучної нейронної мережі.

Глибоке навчання – це підмножина машинного навчання, яке використовує штучні нейронні мережі для вивчення даних. «глибина» в глибокому навчанні відноситься до глибини шарів нейронної мережі.

Визначення: нейронну мережу з більш ніж трьох рівнів, включаючи вхідні та вихідні дані, можна вважати алгоритмом глибокого навчання.

1.2 Застосування глибокого навчання в комп'ютерному зорі та аналізі відео

Глибоке навчання в комп'ютерному зорі та аналізі відео відіграє ключову роль у вирішенні різних завдань, пов'язаних із обробкою візуальної інформації [2]. Нижче наведено основні аспекти та застосування глибокого навчання в цій області:

Об'єктний аналіз та розпізнавання.

Глибокі нейронні мережі, зокрема згорткові нейронні мережі (CNN), використовуються для ефективного об'єктного аналізу та розпізнавання об'єктів на відеозаписах. Це може включати розпізнавання облич, транспортних засобів, тварин та інших об'єктів.

Визначення руху та трекінг об'єктів.

Глибоке навчання може бути застосоване для визначення руху та трекінгу об'єктів на відеопотоці. Це корисно в безпілотних літальних апаратах для визначення положення та руху об'єктів на землі.

Визначення положення та позиції об'єктів.

Глибоке навчання також може бути використане для точного визначення положення та позиції об'єктів на відеозаписі. Це може включати визначення географічних координат, орієнтації та інших параметрів.

Виявлення та аналіз аномалій.

Глибоке навчання може бути застосоване для виявлення аномалій та незвичайних подій на відеозаписах. Це може допомагати в ранньому виявленні потенційних загроз або небезпечних ситуацій.

Автоматизоване вивчення та аналіз великих обсягів даних.

Глибоке навчання дозволяє автоматизовано вивчати та аналізувати великі обсяги відеоданих, що робить його ефективним інструментом для безпілотних літальних апаратів, які збирають великі масиви візуальної інформації.

Системи вирішення задач.

Глибоке навчання може бути використане для розв'язання конкретних задач, таких як виявлення облич, розпізнавання жестів, визначення емоцій, що розширює функціональні можливості систем відеоаналізу.

Використання рекурентних нейронних мереж.

У випадках, коли важливо враховувати часові залежності відеопотоку, рекурентні нейронні мережі (RNN) можуть бути застосовані для ефективного аналізу динамічних змін та подій.

Загальною метою застосування глибокого навчання в комп'ютерному зорі та аналізі відео є підвищення точності, швидкості та робастності систем обробки візуальної інформації, що робить його важливим інструментом для розвитку технологій в різних галузях.

1.3 Використання безпілотних літальних апаратів (БПЛА) в медіцині, безпеці та інших галузях

Медицина.

Безпілотні літальні апарати використовуються для швидкої та надійної доставки медичних препаратів, вакцин, аптечок або крові у важкодоступні або екстрени ситуації, забезпечуючи швидку реакцію на надзвичайні ситуації [3].

БПЛА оснащені високоякісними камерами та сенсорами, що дозволяє їм здійснювати медичне спостереження та діагностику з високою роздільною здатністю, наприклад, виявлення захворювань або нагляд за пацієнтами на віддаленій території.

БПЛА можуть використовуватися для оперативного виявлення та допомоги в екстрених ситуаціях, таких як природні катастрофи або аварії, забезпечуючи швидкий доступ до потрібних ресурсів та медичної допомоги.

Безпека та Спостереження.

БПЛА використовуються для патрулювання та контролю за територіями, а також для виявлення порушень закону та надзвичайних ситуацій, надаючи ефективні інструменти для забезпечення громадської безпеки.

БПЛА можуть бути задіяні у пожежогасінні та ліквідації наслідків аварій, дозволяючи оперативно визначити обсяг збитків та координувати дії рятувальників.

БПЛА використовуються для моніторингу врожаїв, виявлення шкідників та навігації у лісових масивах, забезпечуючи оптимізацію управління ресурсами.

БПЛА використовуються для зйомки з високою роздільною здатністю для створення детальних карт та 3D-моделей місцевості.

БПЛА використовуються для інспекції інфраструктури, такої як мости, трубопроводи та електролінії, забезпечуючи ефективне та безпечне управління об'єктами.

Використання БПЛА в різних галузях вносить суттєві переваги, такі як швидкість реакції, ефективність роботи та покращення безпеки. Однак важливо враховувати питання приватності, безпеки передачі даних та викликів, пов'язаних із нормативно-правовими аспектами використання БПЛА в різних контекстах.

Воєнна справа.

БПЛА використовуються для здійснення аерофотозйомки, створення детальних карт та 3D-моделей територій, що дозволяє аналізувати ландшафти та планувати військові операції.

БПЛА забезпечують можливість постійного спостереження за діяльністю противника, виявлення руху військових одиниць та об'єктів, що допомагає збирати розвідувальну інформацію.

БПЛА можуть бути озброєні для здійснення повітряних атак на ворожі цілі, надаючи можливість точного та ефективного використання вогневої сили.

БПЛА використовуються для надання технічної підтримки бодайшим військам, включаючи передачу живої інформації, сприяння в розташуванні ворожих сил та навігації під час бойових операцій.

БПЛА можуть використовуватися для попередження від реального часу, спостерігаючи за територією та виявляючи можливі загрози, такі як ворожі сили або атаки.

БПЛА можуть використовуватися для маскування своїх дій та розвідувальних операцій, забезпечуючи конфіденційність та уникаючи виявлення ворогом.

Використання БПЛА у воєнний час породжує етичні питання, пов'язані зі збереженням цивільних прав, контролем за автоматизованими системами та уникненням непередбачених наслідків. Важливо враховувати нормативно-правові питання, такі як міжнародне гуманітарне право, для регулювання використання таких технологій у воєнних конфліктах.

1.4 Проблеми та виклики у визначенні положення тіла людини на відеопотоці з БПЛА

У контексті використання безпілотних літальних апаратів (БПЛА) для визначення положення тіла людини, існують ряд проблем та викликів, які можуть впливати на ефективність та точність цього процесу. Нижче розглядаються деякі з ключових аспектів, що потребують уваги при розробці та впровадженні таких систем [4].

Проблеми та виклики.

Низька роздільна здатність камер: Безпілотні літальні апарати (БПЛА) можуть мати обмежену роздільну здатність камер, що ускладнює точне визначення положення тіла людини, особливо на великих відстанях або у зоні з великою кількістю деталей.

Вплив атмосферних умов: Атмосферні умови, такі як туман, дощ, сніг чи інші погодні явища, можуть впливати на якість відеозапису та точність визначення положення об'єктів.

Динамічність об'єктів: Порушення та швидкі зміни положення об'єктів (наприклад, рух тіла людини) можуть викликати труднощі у визначенні точного положення на відеопотоці.

Відсутність тривалого контакту: БПЛА зазвичай мають обмежений час польоту, що може ускладнювати тривале та стабільне визначення положення об'єктів, зокрема тіла людини.

Обмеженість впливу на динамічні об'єкти: У воєнних умовах, коли тіла людей можуть здійснювати швидкі та непередбачувані рухи, обмежена

можливість впливу на динамічні об'єкти може становити виклик для точного визначення їхнього положення.

Проблеми конфіденційності та безпеки: Використання БПЛА для визначення положення тіла людини може викликати питання щодо конфіденційності та безпеки даних, особливо у воєнних операціях.

Недостатня точність алгоритмів обробки відео: Використання алгоритмів обробки відео для визначення положення тіла може обмежуватися їхньою точністю та ефективністю в умовах воєнних конфліктів.

Вирішення проблем, пов'язаних із визначенням положення тіла людини на відеопотоці з безпілотних літальних апаратів (БПЛА), включає застосування різноманітних технічних та програмних підходів. Нижче наведено деякі можливі рішення:

Покращення роздільної здатності: Використання високоякісних камер і сенсорів з покращеною роздільною здатністю дозволяє отримувати більше деталей на відеопотоці, полегшуючи визначення положення тіла людини.

Вдосконалення алгоритмів обробки відео: Розробка та впровадження ефективних алгоритмів комп'ютерного зору для визначення руху та положення об'єктів може значно покращити точність та швидкість аналізу відеопотоку.

Використання технологій штучного інтелекту (AI): Впровадження систем штучного інтелекту, зокрема нейронних мереж та глибокого навчання, може поліпшити здатність системи визначення положення тіла шляхом автоматичного вивчення та розпізнавання патернів.

Інтеграція з іншими сенсорами: Використання додаткових сенсорів, таких як Лідар, радари чи інфрачервоні камери, може допомогти компенсувати обмеження відео та забезпечити точнішу інформацію про оточуюче середовище.

Врахування динаміки об'єктів: Розробка алгоритмів, які можуть ефективно враховувати динаміку та швидкі зміни положення об'єктів, дозволяє точніше визначати положення тіла людини під час руху.

Корекція атмосферних впливів: Врахування атмосферних умов та використання технік корекції може поліпшити якість зображення та забезпечити стабільність визначення положення.

Безпека та конфіденційність: Розробка заходів для захисту конфіденційності даних, введення шифрування та інших заходів безпеки є критичними аспектами, особливо у воєнних застосуваннях.

Враховуючи ці технічні та програмні підходи, важливо також враховувати етичні та правові аспекти використання таких технологій у військових операціях.

1.5 Постановка задачі

Метою роботи є дослідження ефективності методів системи визначення положення тіла людини на відеопотоці з безпілотного літального апарату (БПЛА).

Практичною задачею роботи є порівняння різних архітектур глибокого навчання, а також проведення експериментів з визначенням їхньої ефективності, для визначення положення тіла людини на відеопотоці з БПЛА, які дозволяють відстежувати положення тіла людини у реальному часі, а також надавати можливість адміністраторам аналізувати ці дані для прийняття рішень щодо безпеки або виконання різних завдань, таких як пошук та рятування, моніторинг натовпу або охорона території.

Постановка задачі складається з наступних етапів:

– необхідно вибрати модель
– підготовка даних для класифікації та розпізнавання;
– первинна обробка зображення;
– виявлення характеристик та ключових точок на зображенні;
– визначення розмірів прогнозованого прямокутника
– оцінка швидкодії та точності різних версій алгоритму вибраної моделі

2 МЕТОДИ ГЛИБОКОГО НАВЧАННЯ ДЛЯ ВИЗНАЧЕННЯ ПОЛОЖЕННЯ ТІЛА

2.1 Класифікація методів глибокого навчання

Класифікація – це контрольований метод машинного навчання, коли модель намагається передбачити правильну мітку заданих вхідних даних. Під час класифікації модель повністю навчається з використанням навчальних даних, а потім її оцінюють на тестових даних перед використанням для виконання прогнозу на нових невідомих даних.

Наприклад, алгоритм може навчитися передбачати, чи є даний електронний лист спамом чи веткою (без спаму), як показано нижче на рисунку 2.1.

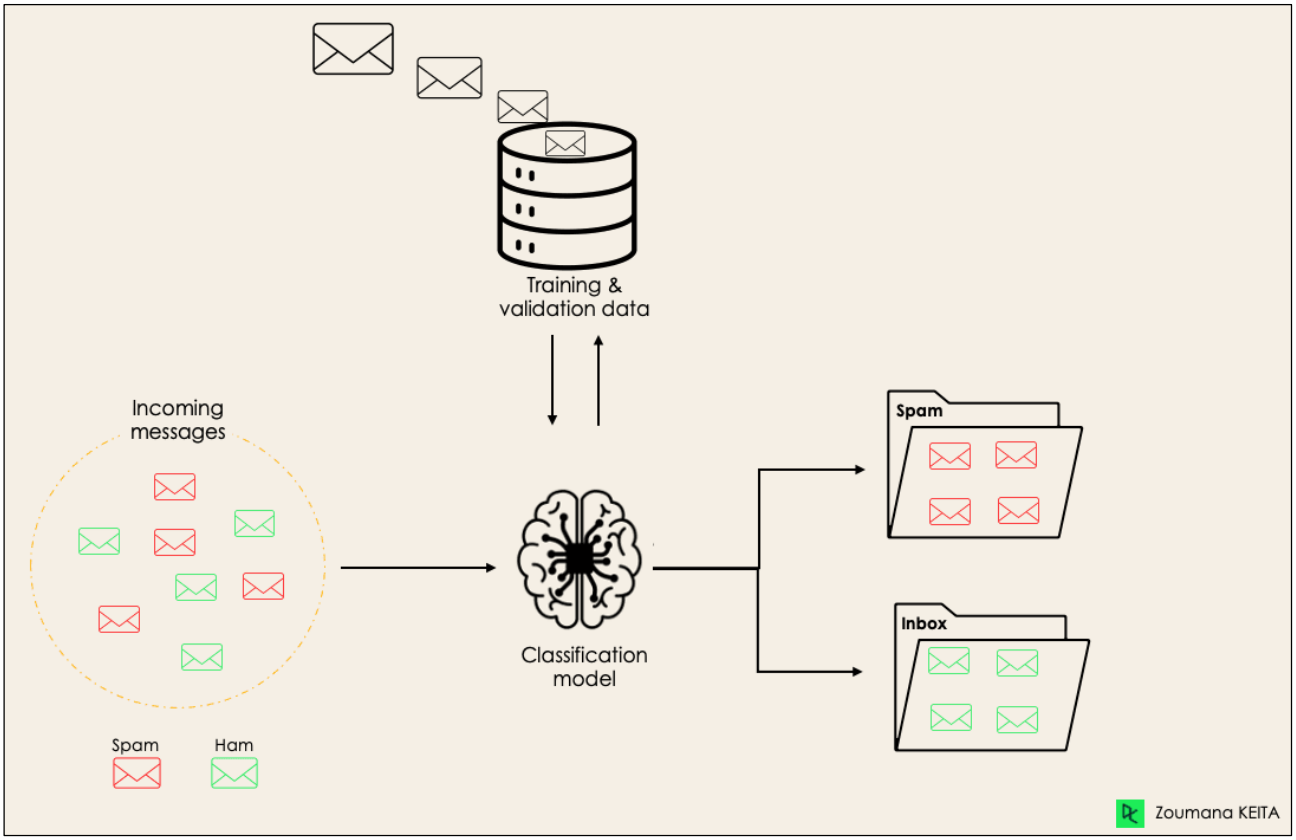


Рисунок 2.1 – Приклад використання алгоритму класифікації (за даними [5])

У машинному навчанні є чотири основні завдання класифікації: двійкова, багатокласова, багатомітка та незбалансована класифікація.

Бінарна класифікація.

У задачі двійкової класифікації метою є класифікація вхідних даних за двома взаємовиключними категоріями. Навчальні дані в такій ситуації позначаються у двійковому форматі: істина та хибність; позитивні і негативні; О і 1; спам і не спам тощо залежно від проблеми, яка вирішується. Наприклад, ми можемо захотіти визначити, чи є дане зображення вантажівкою чи човном (див. рис. 2.2).

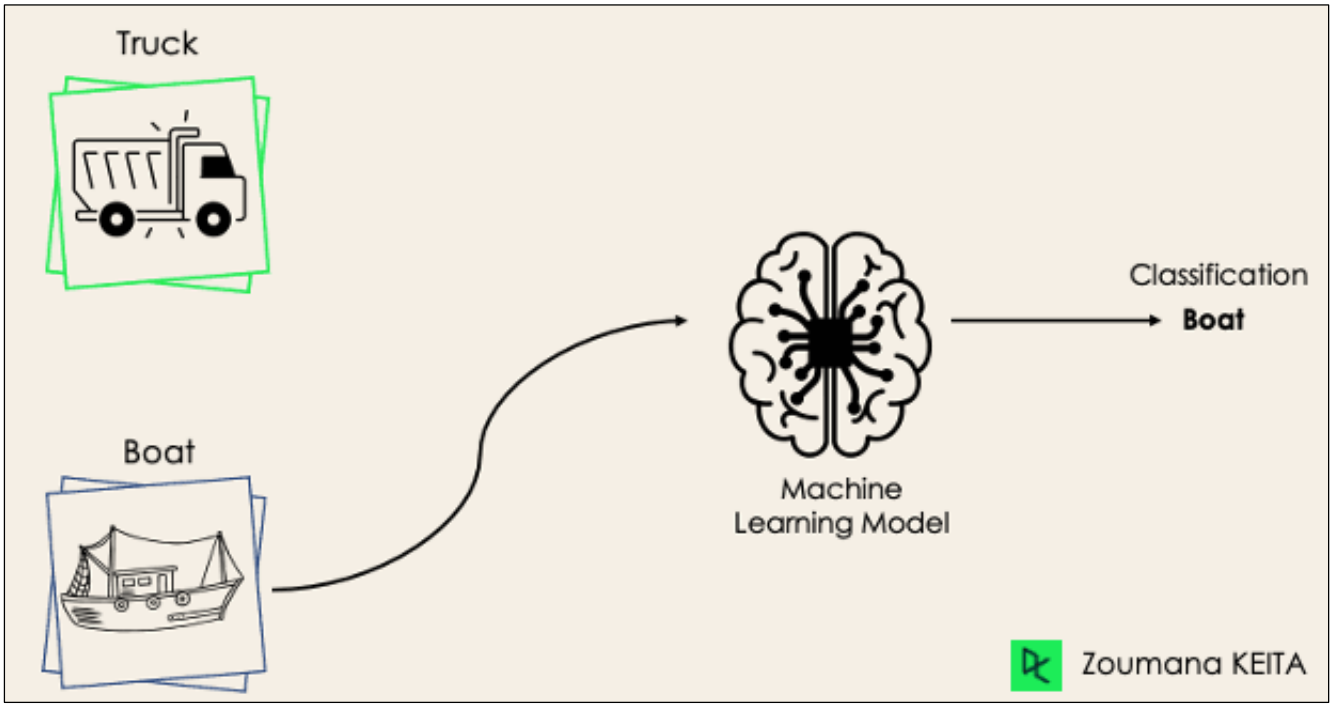


Рисунок 2.2 – Приклад бінарної класифікації (за даними [5])

Алгоритми логістичної регресії та опорних векторних машин розроблені для бінарних класифікацій. Однак для бінарної класифікації також можна використовувати інші алгоритми, такі як K-Nearest Neighbors і Decision Trees.

Багатокласова класифікація

З іншого боку, багатокласова класифікація має принаймні дві взаємовиключні мітки класу, метою яких є передбачити, до якого класу належить даний приклад введення. У наступному випадку модель правильно класифікувала зображення як площину (див. рис. 2.3).

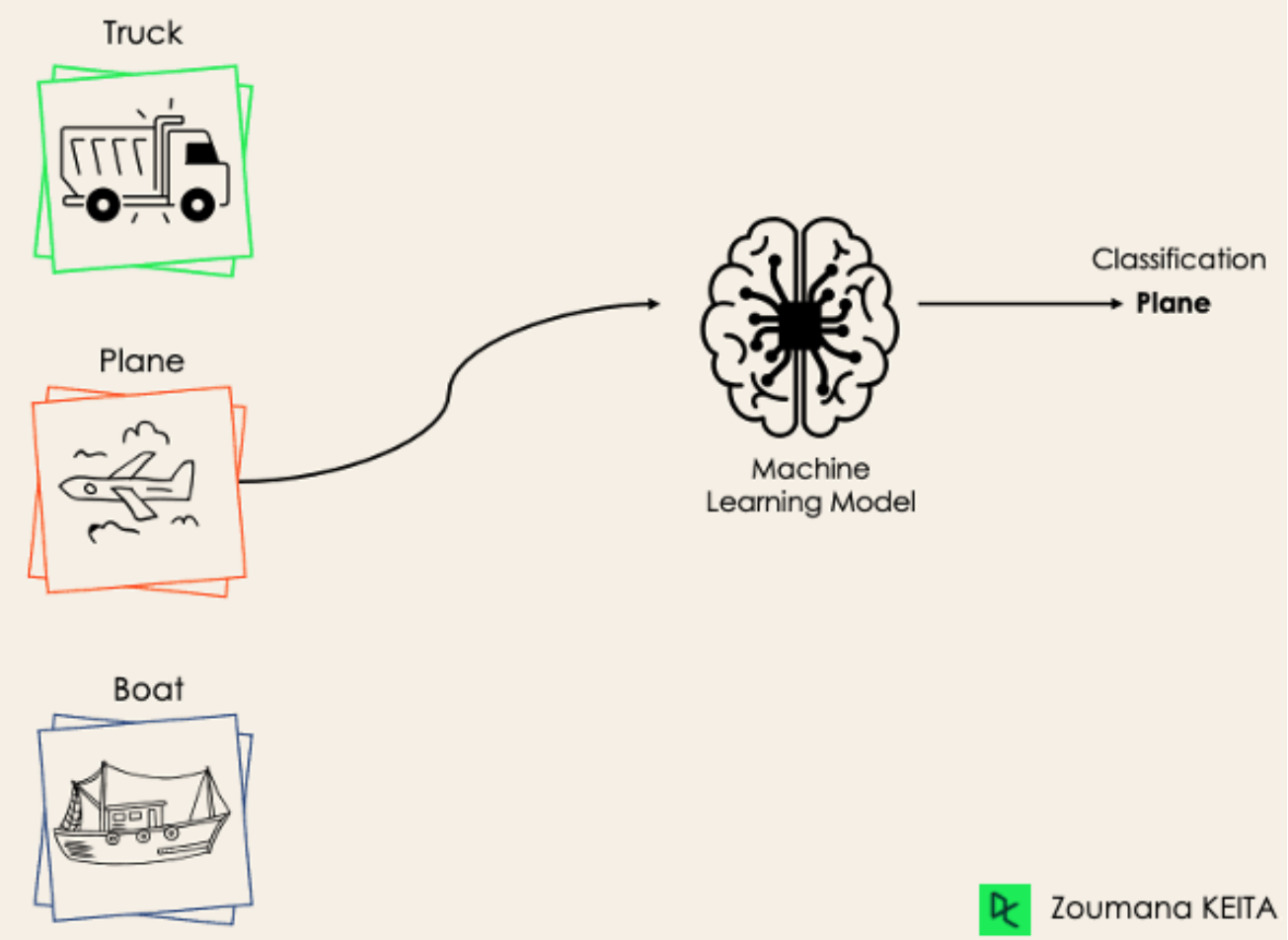


Рисунок 2.3 – Приклад багатокласової класифікації (за даними [5])

Більшість алгоритмів бінарної класифікації також можна використовувати для багатокласової класифікації. Ці алгоритми включають, але не обмежуються:

– випадковий ліс;
– наївний байєс;
– k-найближчі сусіди;
– посилення градієнта;
– svm;
– логістична регресія.

Один проти одного: ця стратегія навчає стільки класифікаторів, скільки є пар міток. Якщо у нас є 3-класова класифікація, ми матимемо три пари міток, тобто три класифікатори, як показано нижче (див. рис. 2.4).

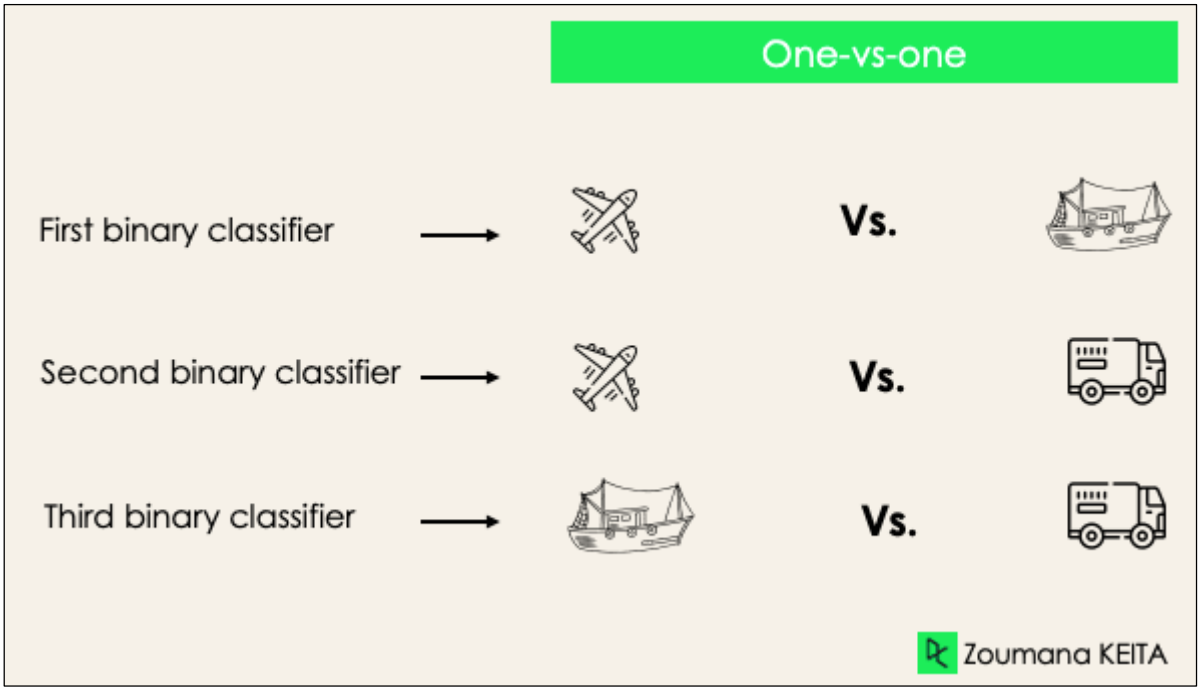


Рисунок 2.4 – Приклад багатокласової класифікації (за даними [5])

Загалом, для N міток ми матимемо Nx(N-1)/2 класифікаторів. Кожен класифікатор навчається на одному двійковому наборі даних, і остаточний клас прогнозується більшістю голосів усіх класифікаторів. Підхід «один проти одного» найкраще працює для SVM та інших алгоритмів на основі ядра.

Один проти решти: на цьому етапі ми починаємо розглядати кожну мітку як незалежну мітку, а решту разом розглядаємо як одну мітку. З 3-класами ми матимемо три класифікатори. Загалом, для N міток ми матимемо N двійкових класифікаторів (див. рис. 2.5)

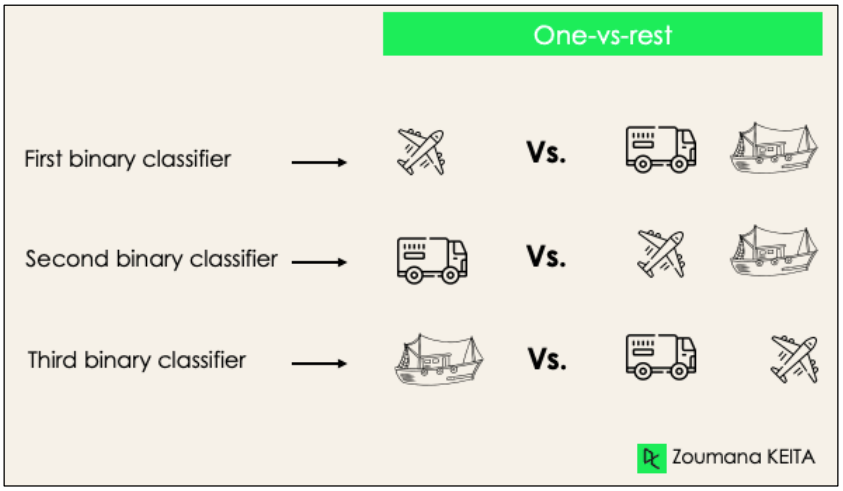


Рисунок 2.5 – Приклад багатокласової класифікації (за даними [5])

Класифікація з кількома мітками.

У завданнях класифікації з кількома мітками ми намагаємося передбачити 0 або більше класів для кожного вхідного прикладу. У цьому випадку немає взаємного виключення, оскільки вхідний приклад може мати більше однієї мітки.

Такий сценарій можна спостерігати в різних областях, наприклад, автоматичне позначення тегами в обробці природної мови, де певний текст може містити кілька тем. Подібно до комп'ютерного зору, зображення може містити кілька об'єктів, як показано нижче: модель передбачила, що зображення містить: літак, човен, вантажівку та собаку (див. рис. 2.6).

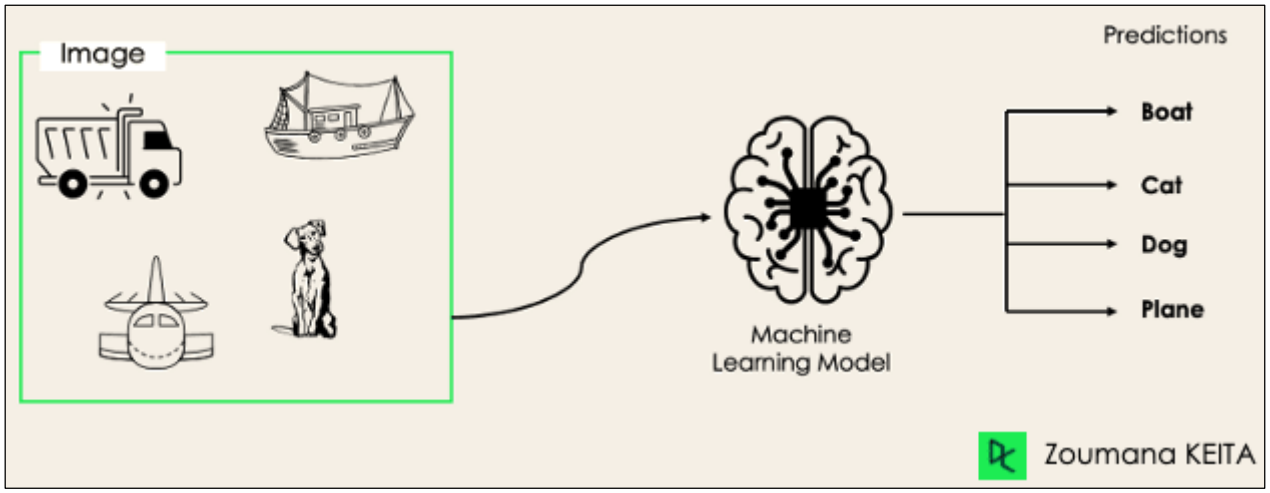


Рисунок 2.6 – Приклад класифікації з кількома мітками (за даними [5])

Неможливо використовувати багатокласові або бінарні моделі класифікації для виконання класифікації з кількома мітками. Однак більшість алгоритмів, що використовуються для цих стандартних завдань класифікації, мають свої спеціалізовані версії для класифікації за кількома мітками. Ми можемо цитувати:

– дерева рішень із кількома мітками;
– підсилення градієнта з кількома мітками;
– випадкові ліси з кількома мітками;
– незбалансована класифікація.

Незбалансована класифікація.

Для незбалансованої класифікації кількість прикладів нерівномірно розподілена в кожному класі, що означає, що ми можемо мати більше одного

класу, ніж інші в навчальних даних. Розглянемо наступний сценарій класифікації з 3 класів, де навчальні дані містять: 60% вантажівок, 25% літаків і 15% човнів (див. рис. 2.7).

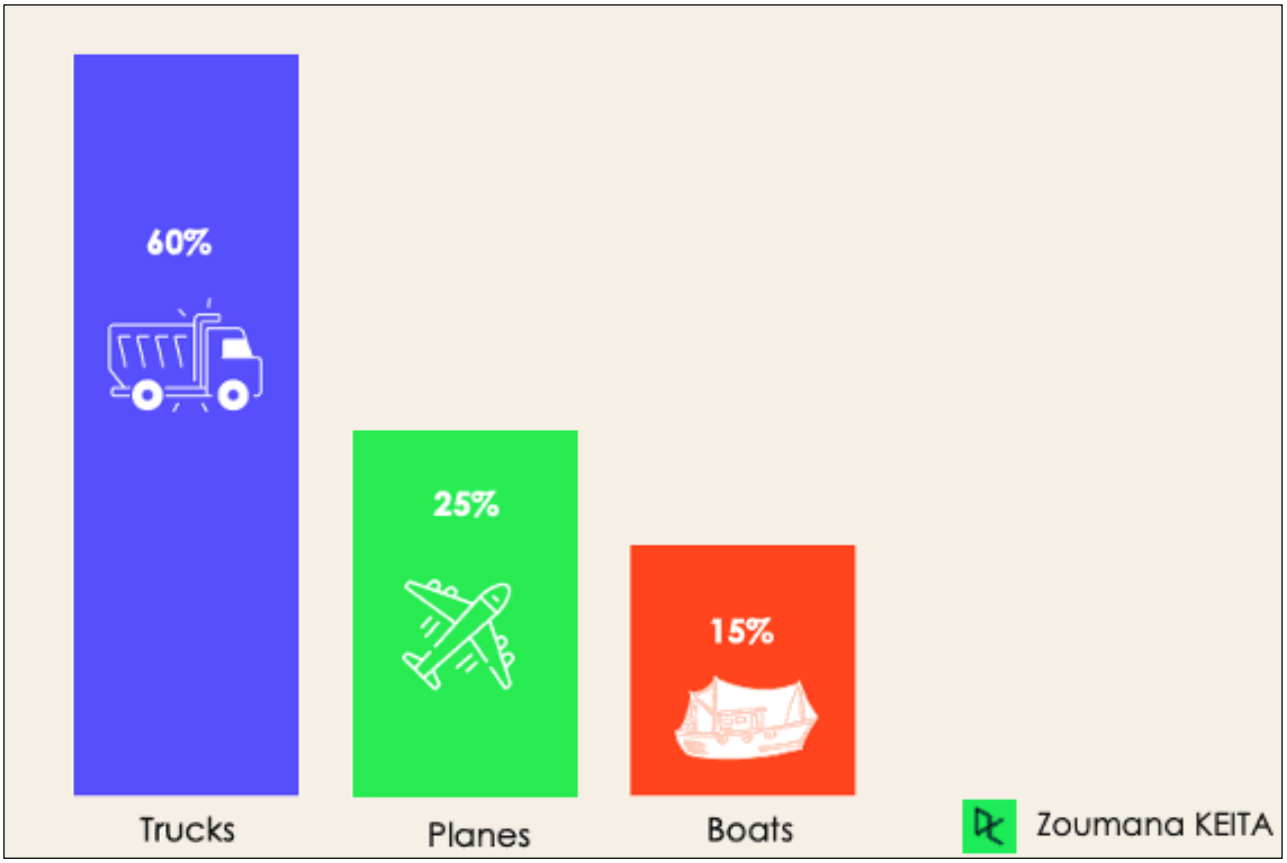


Рисунок 2.7 – Приклад незбалансованої класифікації (за даними [5])

Проблема незбалансованої класифікації може виникнути в такому випадку:

– виявлення шахрайських транзакцій у фінансових галузях;
– діагностика рідкісних захворювань;
– аналіз відтоку клієнтів.

Використання звичайних прогностичних моделей, таких як Дерева рішень, Логістична регресія тощо, не може бути ефективним при роботі з незбалансованим набором даних, оскільки вони можуть бути упередженими в бік прогнозування класу з найбільшою кількістю спостережень і розглядати ті з меншими числами як шум.

Техніка відбору проб.

Ці методи мають на меті збалансувати розповсюдження оригіналу за допомогою:

– передвибірка на основі кластерів;
– випадкова недостатня вибірка: випадкова елімінація прикладів із мажоритарного класу;
– надвибірка smote: випадкова реплікація прикладів із класу меншості.

Економічні алгоритми

Ці алгоритми враховують вартість неправильної класифікації. Вони спрямовані на мінімізацію загальних витрат, створених моделями.

– дерева рішень, чутливі до витрат;
– логістична регресія з урахуванням витрат;
– економічні опорні векторні машини.

2.2 Архітектури нейронних мереж для визначення положення тіла

Визначення положення тіла людини на відеопотоці вимагає використання спеціалізованих архітектур нейронних мереж, які здатні враховувати просторові відносини та динаміку руху. Ось деякі з розповсюджених архітектур для цієї задачі:

OpenPose.

Тип мережі: сверточна нейронна мережа (CNN).

Характеристики: розроблена для визначення поз та ключових точок на тілі людини. Має здатність працювати з різними відстанями та орієнтаціями.

HRNet (High-Resolution Network).

Тип мережі: сверточна нейронна мережа (CNN).

Характеристики: використовує високу роздільність при виявленні ключових точок та просторових відносин. Забезпечує більш деталізоване визначення положення тіла.

PoseNet.

Тип мережі: сверточна нейронна мережа (CNN).

Характеристики: розроблена для виявлення поз на основі глибокого вивчення. Може працювати в режимі реального часу на пристроях з обмеженими ресурсами.

AlphaPose.

Тип мережі: сверточна нейронна мережа (CNN).

Характеристики: орієнтована на точне визначення пози людини, включаючи високу точність в локалізації ключових точок та просторових відносин.

SimplePose.

Тип мережі: сверточна нейронна мережа (CNN).

Характеристики: забезпечує ефективне та точне визначення положення тіла, зокрема у сценаріях з відсутністю великої кількості даних.

Ці архітектури можуть бути використані як основа для завдань визначення положення тіла на відеопотоці. Вони використовують різноманітні техніки, такі як глибоке вивчення, сверточні шари та механізми уваги, для забезпечення точності та ефективності у роботі з візуальною інформацією.

2.3 Використання конволюційних нейронних мереж (CNN) та рекурентних нейронних мереж (RNN)

Рекурентні нейронні мережі призначені для інтерпретації часової або послідовної інформації. Ці мережі використовують інші точки даних у послідовності для більш точного прогнозування. Вони роблять це, приймаючи вхід та використовуючи активації попередніх вузлів чи вузлів, що йдуть пізніше у послідовності, для впливу на вихід.

Видобуток сутностей в тексті є відмінним прикладом того, як дані в різних частинах послідовності можуть взаємодіяти. У випадку сутностей слова, які передують та слідують за сутністю в реченні, мають прямий вплив на їх класифікацію. Щоб працювати з часовими чи послідовними даними, такими як речення, слід використовувати алгоритми, призначені для вивчення із минулих даних та "майбутніх даних" у послідовності (див. рис. 2.8).

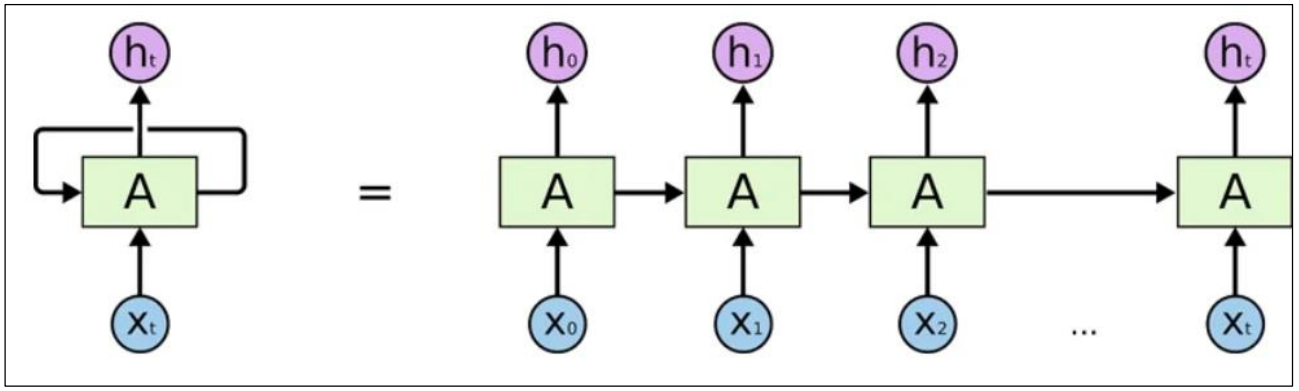


Рисунок 2.8 – Рекурентна нейронна мережа (за даними [6])

RNN для автовиправлення.

Щоб глибше зануритися в роботу RNN, вам не потрібно дивитися далі, окрім систем автовиправлення. На базовому рівні системи автовиправлення приймають слово, яке ви ввели, як вхідні дані. Використовуючи ці вхідні дані, система робить прогноз щодо правильного чи неправильного написання. Якщо слово не відповідає жодному слову в базі даних або не вписується в контекст речення, система прогнозує, яке може бути правильне слово. Давайте візуалізуємо, як працює цей процес, використовуючи RNN (див. рис. 2.9).

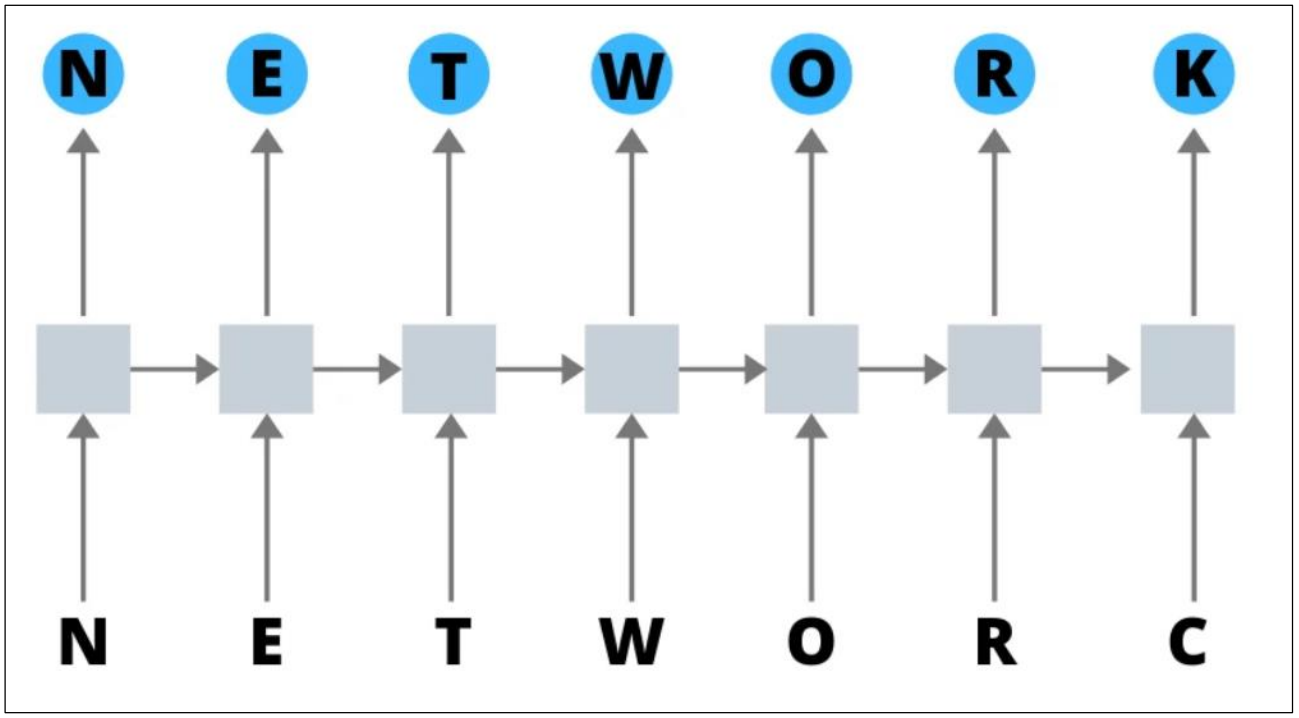


Рисунок 2.9 – Зображення системи автовиправлення за допомогою RNN (за даними [6])

RNN прийматиме два джерела вхідних даних. Перше введення – це літера, яку ви ввели. Другим входом будуть функції активації, які відповідають попереднім літерам, які ви ввели. Припустімо, ви хотіли ввести «мережа», але помилково ввели «мережа». Система використовує функції активації попередніх літер «мережа» та поточної літери, яку ви ввели, «с». Потім він пропонує «k» як правильний вихід для останньої літери.

Це лише один спрощений приклад того, як RNN можуть працювати для виправлення орфографії. Сьогодні дослідники обробки даних використовують RNN, щоб виконувати низку неймовірних речей – від генерування тексту та підписів до зображень до створення музики та прогнозування коливань фондового ринку. RNN мають безмежні потенційні варіанти використання.

Згорткові нейронні мережі є одним із найпоширеніших типів нейронних мереж, які використовуються в комп'ютерному зорі для розпізнавання об'єктів і шаблонів на зображеннях. Однією з їх визначальних рис є використання фільтрів у згорткових шарах.

Згорткові шари.

CNN мають унікальні шари, які називаються згортковими шарами, які відокремлюють їх від RNN та інших нейронних мереж.

У межах згорткового шару вхідні дані перетворюються перед передачею на наступний шар. CNN перетворює дані за допомогою фільтрів.

Що таке фільтри в згорткових нейронних мережах?

Фільтр – це просто матриця рандомізованих числових значень, як ви бачите на діаграмі нижче (див. рис. 2.10).

0.230	0.380	0.971
0.402	0.119	0.886
0.693	0.563	0.771

Рисунок 2.10 – Зображення прикладу фільтра 3 x 3 (за даними [6])

Кількість рядків і стовпців у фільтрі може змінюватись і залежить від варіанту використання та даних, що обробляються. У згортковому шарі є кілька фільтрів, які переміщуються по зображенню. Цей процес називають згортанням, що є терміном, який походить із математики та включає акт об’єднання. Фільтр згортає пікселі зображення, змінюючи їхні значення перед передачею даних на наступний рівень у мережі.

3 АНАЛІЗ МЕТОДУ YOLO ДЛЯ РОЗПІЗНАВАННЯ ОБ'ЄКТІВ НА ЗОБРАЖЕННЯХ

3.1 Принципи створення сучасних засобів розпізнавання на основі нейронних мереж

Згорткова нейронна мережа (CNN) – це алгоритм, який отримує вхідне зображення та призначає вагові значення окремим його частинам, оцінюючи їх приналежність до різних класів для розпізнавання об’єктів. Архітектура CNN нагадує структуру з’єднань нейронів у людському мозку: кожен нейрон реагує на стимули лише в межах певної області зору, відомої як рецептивне поле. Ці поля частково перекриваються, щоб покрити всю візуальну область. Завдяки застосуванню фільтрів, CNN здатні вловлювати просторові та часові залежності в зображеннях, що дозволяє їм краще розуміти їх структуру.

Розглянемо основні нейромережеві методи, такі як CNN, Fast R-CNN, Faster R-CNN та YOLO. Метод R-CNN використовує положення регіонів для створення потенційних обмежувальних прямокутників на зображенні, а потім запускає класифікатор для цих рамок. Після класифікації застосовується постобробка для уточнення рамок, усунення дублікатів та повторної оцінки на основі інших об’єктів у сцені. Такі складні конвеєри є повільними та важко оптимізуються, оскільки кожен компонент потребує окремого навчання.

Принцип алгоритму R-CNN виглядає наступним чином (рис. 3.1).

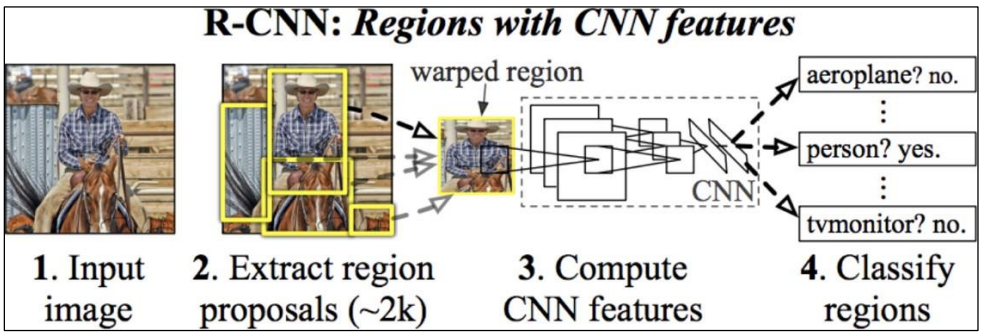


Рисунок 3.1 – Принцип функціонування алгоритму R-CNN

Р. Гіршик розробив метод, що використовує вибірковий пошук для виділення лише 2000 регіонів із зображення, називаючи їх пропозиціями регіонів.

Завдяки цьому, замість класифікації великої кількості регіонів, можна працювати лише з 2000. Алгоритм працює наступним чином:

- створення початкових регіонів-кандидатів із зображення;
- застосування жадібного алгоритму для рекурсивного об'єднання подібних частин зображення у більші регіони;
- використання отриманих регіонів для формування остаточних пропозицій регіонів-кандидатів [7].

Розроблено різні методи локалізації об’єктів у процедурах їх виявлення. Одним із підходів є застосування ковзних фільтрів різного розміру для виділення об’єктів на зображенні, що відомо як вичерпний пошук. Обчислювальна складність цього підходу залежить від кількості використовуваних фільтрів.

Алгоритм вибіркового пошуку поєднує вичерпний пошук із сегментацією кольорів на зображенні. Це дозволяє йому відокремлювати об’єкти, використовуючи кольорові відмінності. Спочатку створюються численні маленькі фільтри, а потім за допомогою жадібного алгоритму розширюються області, які об’єднуються на основі схожості кольорів.

Схожість між областями можна обчислити за допомогою певної формули 3.1:

$$S(a, b) = Stexture(a, b) + Ssize(a, b), \quad (3.1)$$

де Stexture(a,b) – це візуальна схожість;

Ssize(a,b) – схожість між регіонами.

За допомогою цього алгоритму модель продовжує об'єднувати всі регіони для збільшення їх розміру.

Мережа Fast R-CNN приймає на вхід все зображення та набір пропозицій об’єктів.

Архітектура цієї моделі використовує фотографію і набір пропозицій регіонів як вхідні дані, які пропускаються через глибоку згорткову нейронну мережу. Попередньо навчена CNN застосовується для виділення ознак. На

завершенні глибокої CNN знаходиться спеціальний шар, відомий як «Рівень об'єднання регіонів інтересів» або «ROI об'єднання», який виділяє ознаки, характерні для кожного регіону-кандидата з вхідних даних.

Спочатку мережа обробляє все зображення за допомогою кількох згорткових і шарів максимального об'єднання, щоб створити карту ознак conv. Потім, для кожної пропозиції об'єкта, шар об'єднання регіону інтересу (RoI) витягує вектор ознак фіксованої довжини з цієї карти ознак. Кожен вектор ознак подається у послідовність повністю зв'язаних шарів, які зрештою розділяються на два вихідні рівні: один, що створює оцінки ймовірності, і інший, що виводить чотири дійсні числа для кожного класу об'єктів. Вихідні дані CNN інтерпретуються повністю зв'язаним шаром, після чого модель розгалужується на два виходи: один для прогнозування класу через шар softmax, а інший – з лінійним виходом для обмежувальної рамки.

Цей процес повторюється кілька разів для кожної цікавої області на зображенні.

На рисунку 3.2 показана архітектура моделі Fast R-CNN.

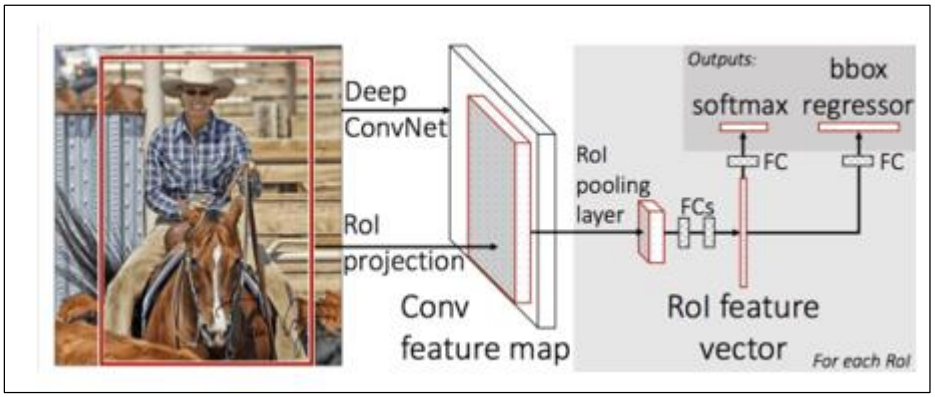


Рисунок 3.2 – Архітектура Fast R-CNN

Ця модель працює швидше за простий R-CNN, здійснює навчання і робить прогнози, але все ще потребує набору кандидатів-регіонів для кожного вхідного зображення. У Faster R-CNN замість максимального об’єднання використовується об’єднання ROI, що дозволяє використовувати одну карту ознак для всіх регіонів. Це перетворює ROI в один шар; рівень об’єднання ROI застосовує максимальне об’єднання для трансформації ознак.

У методі Faster R-CNN зображення подається як вхід до згорткової мережі, яка створює карту згорткових ознак. Замість алгоритму вибіркового пошуку для визначення регіональних пропозицій на карті ознак використовується окрема мережа для прогнозування регіонів. Архітектуру Faster R-CNN можна побачити на рисунку 3.3.

Faster R-CNN застосовує метод пропозиції регіонів для створення наборів регіонів. Він містить додаткову CNN, яка відповідає за генерування регіональних пропозицій, відому як регіональна мережа пропозицій. У процесі навчання регіональна мережа приймає карту ознак як вхідні дані і виводить регіональні пропозиції. Ці пропозиції потім надходять на рівень об’єднання ROI для подальшої обробки [8, 9].

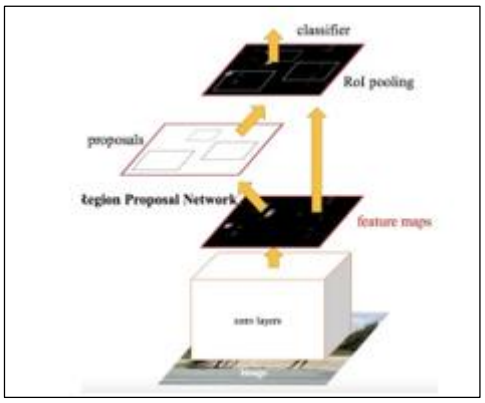


Рисунок 3.3 – Архітектура Faster R-CNN

SSD (Single Shot MultiBox Detector) призначений для обробки в режимі реального часу. Він прискорює процес, усуваючи необхідність у регіональній мережі. Для компенсації втрати точності, SSD застосовує кілька вдосконалень, таких як багатомасштабні функції та стандартні поля.

Метод SSD складається з двох частин:

– видобування карти функцій;

– застосування фільтрів згортки для виявлення об’єктів.

Кожне передбачення включає обмежувальну рамку та оцінку для кожного класу [10].

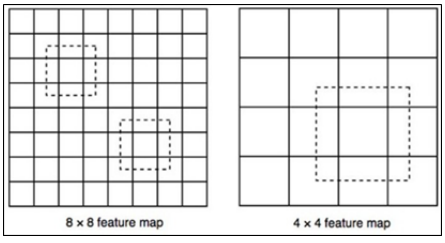
SSD, на відміну від підходів, які базуються переважно на регіональних мережах пропозицій, наприклад R-CNN, використовує лише один знімок для виявлення кількох об'єктів у зображенні. Однак, при невеликому розмірі об'єкта ефективність SSD трохи падає.

Цей метод використовує різні шари, що працюють незалежно, із зниженою роздільною здатністю для виявлення об'єктів різних масштабів. Для визначення результативних блоків SSD використовує градієнтний спуск для оптимізації моделі.

Під час навчання SSD використовує фазу порівняння для узгодження рамок прив'язки з обмежувальними рамками кожного об'єкта на зображенні. Блок з найвищим ступенем перекриття з об'єктом відповідає за прогнозування класу та розташування цього об'єкта.

SSD розбиває зображення на непересічні блоки, призначаючи кожному блоку сітки відповідальність за виявлення об'єктів у своїй області. Коли об'єкт відсутній, блок розглядається як фоновий клас, і його розташування ігнорується. Кожній клітинці сітки в SSD може бути призначено кілька прив'язок до попередніх блоків, які передбачено заздалегідь і відповідають за розмір та форму у клітинці сітки [11, 12].

Метод SSD використовує мультишарові карті ознак, тобто різні рівні зображень, які дозволяють незалежно виявляти об'єкти. Оскільки згорткові нейронні мережі поступово зменшують просторові розміри, роздільна здатність карт ознак також зменшується. Наприклад, для об'єктів більшого масштабу використовуються карті ознак розміром 4×4 та 8×8, як показано на рисунках 3.4 а) та б).



а) б)

Рисунок 3.4 – Функціональна карта рішень

Розмір результативних блоків не обов'язково повинен бути таким самим, як у непересічних блоків. SSD складається з двох компонентів: опорної моделі та передньої частини SSD. Опорною моделлю є попередньо навчена мережа класифікації зображень, яка використовується як екстрактор ознак. Зазвичай це мережа ResNet, навчена на ImageNet, із видаленим останнім повністю підключеним рівнем класифікації [13, 14]. Стандартні розміри непересічних результативних блоків обчислюються наступним чином (формула3.2):

$$w = S \times \sqrt{a,h} = S/\sqrt{a} \tag{3.2}$$

де S – масштаб;

α – співвідношення сторін.

У зв'язку з великою кількістю блоків, що генеруються під час прямого проходу SSD, важливо скористатися методом, відомим як немаксимальне придушення, для обмеження кількості блоків, що потрапляють до кінцевого результату. Цей підхід полягає в тому, щоб відкидати блоки з недостатньою достовірністю та поганим покриттям зображення (IoU), і зберігати лише перші N прогнозів. Такий підхід гарантує, що мережа зберігає лише найбільш вірогідні прогнози, відкидаючи шумові дані (див. рис. 3.5).

Рисунок 3.5 – Ілюстрація використання методів техніки недостатнього заглушення

Чим більше непересічних блоків за замовчуванням, тим точніше виявлення. SSD може сплутати об'єкти зі схожими категоріями, наприклад, тваринами. Також важливо зауважити, що SSD показує меншу ефективність при розпізнаванні менших об'єктів, оскільки вони можуть бути не представлені на всіх картах функцій.

3.2 Підхід YOLO

Абревіатура YOLO стоїть за фразою "You Only Look Once" і відноситься до архітектурної нейронної мережі, призначеної для виявлення об’єктів. На сьогоднішній день існують три версії YOLO: версія 1, версія 2 і версія 3. Останні дві версії є вдосконаленнями першої [15-16]. У алгоритмі YOLOv1 важко виявляти малі об’єкти, особливо якщо вони згруповані. Ця архітектура також має проблеми з узагальненням точок об’єктів при зміні розмірів зображення. Основною складністю є точність локалізації об’єктів на вхідному зображенні. У YOLOv2 було впроваджено пакетну нормалізацію на вхідному рівні для зменшення зсувів у прихованому шарі та покращення стабільності мережі. Також збільшено розмір вхідних даних та введено Anchor Boxes, які об’єднують класифікацію та прогнозування в одній структурі. Ці блоки відповідають за прогнозування обмежувальної рамки й розроблені для конкретного набору даних за допомогою кластеризації.

YOLO розглядає завдання виявлення об'єктів як задачу регресії, а не класифікації, і використовує одну згорткову нейронну мережу. Метод YOLO визначає об'єкти безпосередньо з пікселів зображення до координат обмежувальної рамки та ймовірностей класу. Цей підхід дозволяє моделі одноразово аналізувати зображення, щоб визначити присутні об'єкти та їх місцезнаходження.

Метод YOLO відзначається високою швидкістю, оскільки не потребує складного конвеєра для обробки кадрів, як інші алгоритми. Згортка застосовується лише один раз до всього вхідного зображення, щоб отримати

прогнози, що дозволяє закодувати контекстну інформацію про класи та їх виявлення.

Під час тестування ми перемножуємо ймовірності умовного класу та прогнози достовірності окремих рамок, що дає оцінки достовірності для кожного класу. Ці показники виражають як ймовірність того, що даний клас присутній у полі, так і те, наскільки точно прогнозований об'єкт відповідає реальному об'єкту (рис. 3.6).

$$\Pr(\text{Class}_i|\text{Object}) * \Pr(\text{Object}) * \text{IOU}_{\text{pred}}^{\text{truth}} = \Pr(\text{Class}_i) * \text{IOU}_{\text{pred}}^{\text{truth}}$$

Рисунок 3.6 – Оцінка достовірності кожного класу

Цей алгоритм зменшує кількість помилок при передбаченні корекцій об'єктів на фоні, оскільки він аналізує всі зображення та причини глобально, а не локально [17]. Суть системи YOLO полягає в тому, що одна згорткова мережа одночасно передбачає кілька прямокутних обмежень і ймовірності класів для цих обмежень [18]. Спочатку розмір вхідного зображення змінюється, після чого запускається згорткова мережа, а останнім етапом є визначення порогового значення результатів виявлення з упевненістю моделі. Ключовим етапом у процесі виявлення об'єктів на зображенні є алгоритм non-maximal suppression, показаний на рисунку 3.7.

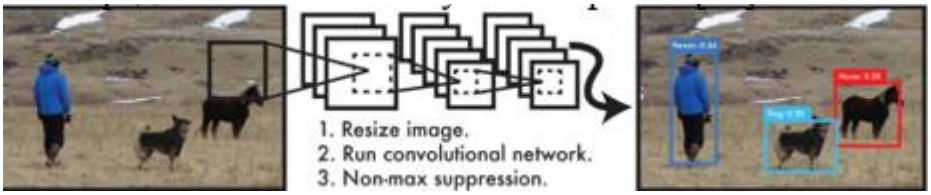


Рисунок 3.7 – Детекція об'єктів за допомогою алгоритму YOLO

Цей алгоритм є критичним, оскільки аналізує кандидатів на виявлення і зберігає лише найбільш підходящі. Одна згорткова мережа одночасно передбачає кілька прямокутних обмежень і ймовірності класів для цих обмежень. YOLO тренується на повних зображеннях і безпосередньо оптимізує продуктивність

виявлення. Ця єдина модель має кілька переваг над традиційними методами виявлення об'єктів.

Алгоритм YOLO функціонує за допомогою розділення зображення на сітки, які мають рівномірну область. Кожна з цих сіток відповідає за виявлення та локалізацію об’єкта, що міститься в ній. Таким чином, ці сітки передбачають координати обмежувальної рамки відносно координат їхніх клітинок, разом із міткою об’єкта та ймовірністю його присутності в клітинці. Цей процес істотно зменшує обчислювальну складність, оскільки як виявлення, так і розпізнавання обробляються клітинками зображення. Однак це призводить до значного числа дублікатів прогнозів, оскільки кілька клітинок можуть прогнозувати той самий об’єкт з різними параметрами обмежувальної рамки [19].

Алгоритм YOLO ґрунтується на трьох основних принципах:

- використання залишкових блоків;
- застосування регресії для обмежувальних рамок;
- розрахунок критерію перетину над об'єднанням (Intersection Over Union - IOU).

Спочатку зображення розділяється на сітки розміром S×S, які розташовані рівномірно. На рисунку 3.8 можна побачити поділ вхідного зображення за допомогою такої сітки.

Рисунок 3.8 – Розділення зображення на сітку

На зображенні 3.8 показана сітка з багатьма клітинками однакового розміру. Кожна клітинка сітки виявляє об’єкти, які з’являються всередині неї, наприклад,

якщо об’єкт з'являється у певній клітинці сітки, то ця клітинка відповідає за його виявлення. Обмежувальна рамка - це прямокутник, який виділяє об’єкт на зображенні. Кожна обмежувальна рамка характеризується наступними атрибутами: клас (наприклад, автомобіль, дорожній знак і т. д.), висота, ширина та центр рамки. На малюнку 3.9 наведено приклад обмежувальної рамки, яка представлена жовтим контуром.

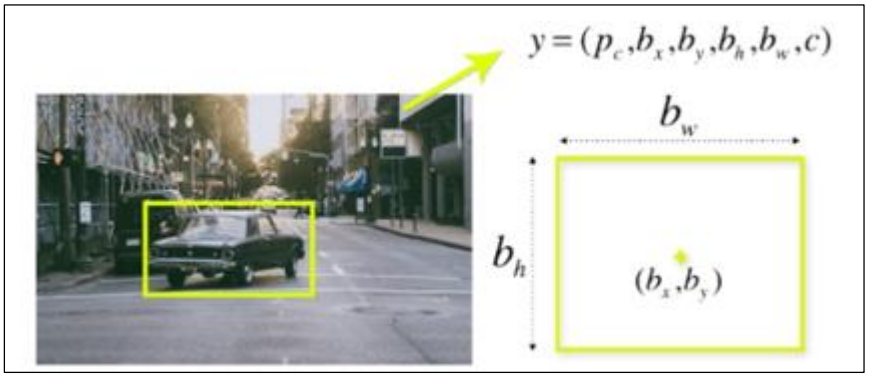


Рисунок 3.9 – Виділення об’єкта за допомогою рамки, що його обмежує

YOLO використовує одну регресію для передбачення висоти, ширини, центру та класу об'єктів у межах обмежувальної рамки. На рисунку 3.9 показано ймовірність з'яви об'єкта у рамці.

Intersection over union (перетин через об'єднання IOU) - це критерій для визначення об'єктів, який описує, як блоки перекриваються. YOLO використовує IOU для створення поля виводу, яке оптимально обгортає об'єкти. На рисунку 3.10 показано простий приклад роботи IOU.

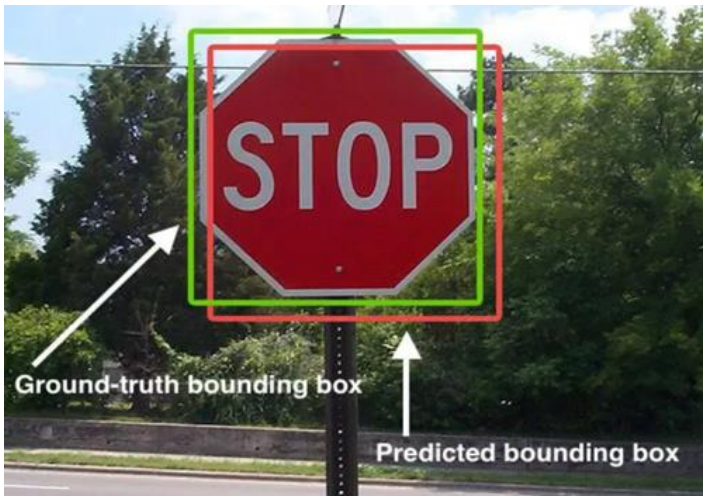


Рисунок 3.10 – Демонстрація функціонування системи IOU

Кожна частина сітки розділена з урахуванням передбачень обмежувальних прямокутників та коефіцієнтів точності. Якщо прогнозована рамка збігається з реальною, IOU дорівнює 1. Цей механізм відкидає рамки, які не відповідають реальності. На рисунку 3.10 показано дві рамки: одна зелена, інша червона. Червоний прямокутник – це передбачений контур, а зелений - реальний. YOLO забезпечує достатню близькість між двома рамками. IoU обчислюється як відношення площі перетину до загальної площі, яка обчислюється як перетин множин, поділений на їх об'єднання (див. рис. 3.11).

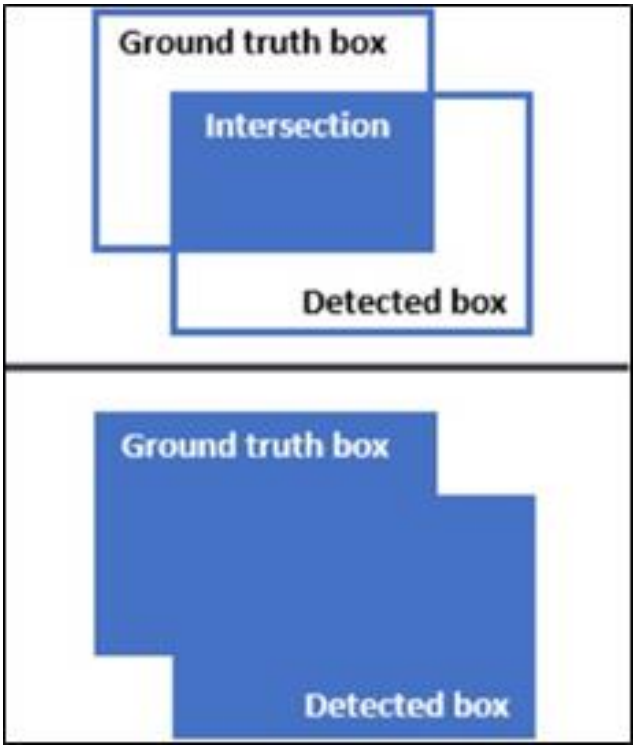
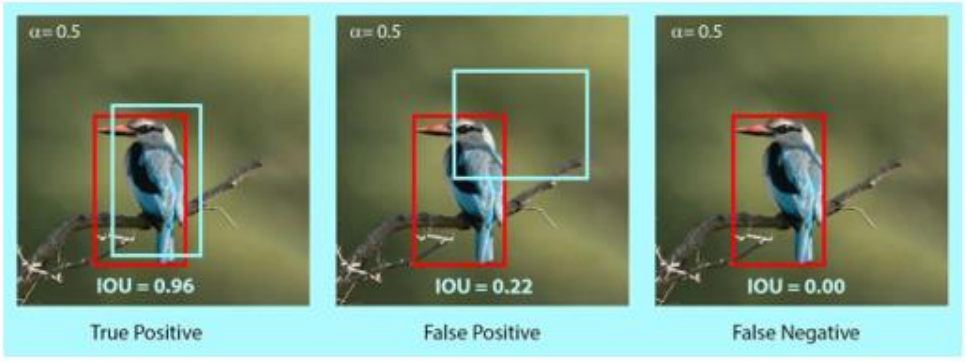


Рисунок 3.11 – Обчислення IoU

IoU (Intersection over Union) знаходиться в межах від 0 до 1, де 0 вказує на жодного перекриття, а 1 – на ідеальне. Встановлення порогу для IoU дозволяє визначити, чи є прогноз істинно позитивним (TP), хибно-позитивним (FP) або хибно-негативним (FN). Наприклад, на рисунку 3.12 показані прогнози з встановленим порогом IoU, який дорівнює α=0,5. Перший прогноз вважається істинно позитивним, оскільки його значення IoU перевищує 0,5.

Якщо встановити поріг на рівні 0,97, це може призвести до того, що він буде вважатися неправильною позитивною реакцією.



а) б) в)

Рисунок 3.12 – Ілюстрація прогнозів для метрики IoU

Аналогічно, другий прогноз з рисунку 3.12 а), б), в) може бути помилково позитивним за пороговим значенням, але при установці порогу на рівні 0,20 може бути правильно позитивним.

Теоретично, третій прогноз також може бути правильно позитивним, якщо поріг буде знижено до 0.

IoU у виявленні об'єктів використовується як допоміжна метрика. Однак у випадку сегментації зображення IoU є основним показником для оцінки точності моделі.

При сегментації зображення область може мати будь-яку форму, не обов'язково прямокутну, тому передбачення виконується у вигляді сегментаційних масок, а не обмежувальних рамок. Тут проводиться аналіз піксель за пікселем. Крім того, визначення TP, FP і FN відрізняється, оскільки не базується на попередньо визначеному пороговому значенні.

Рішення про визначення, чи є результат справжньо позитивним чи хибно позитивним, цілком залежить від вимог до розроблюваного додатку. Кожен показник IoU обчислюється відносно кожного базового блоку істинності на зображенні. Значення IoU потім фільтруються на певний діапазон (зазвичай від 0,5 до 0,95), і прогнози порівнюються з основними даними методом жадібної стратегії (тобто спочатку порівнюються прогнози з найвищими значеннями IoU). Потім для кожного класу об'єктів будується крива точності-пам'яті (PR), і обчислюється середня точність (AP). Крива PR відображає ефективність моделі

щодо справжніх позитивних, хибно-позитивних і хибно-негативних результатів у різних значеннях довіри. Давайте визначимо справжній позитивний результат (формула 3.3):

$$TP = GT \times S, \tag{3.3}$$

де TP - істинно позитивне;

GT - основна правдивість;

S - маска сегментації.

Хибно позитивний результат свідчить про те, що область розташована поза межами сегментації зображення і розраховується наступним способом (формула 3.4):

$$FP = (GT + S) - GT. \tag{3.4}$$

Результат, який вказує на помилку або негативний результат, визначається кількістю пікселів у зоні, яку модель не змогла передбачити, і обчислюється таким способом (формула 3.5):

$$FN = (GT + S) - S. \tag{3.5}$$

У завданні виявлення об'єктів, IoU визначається як співвідношення площі перетину до об'єднаної площі передбачення та істинності. Оскільки TP, FP і FN представлені або як площі, або як кількість пікселів, IoU можна записати таким чином (формула 3.6):

$$IoU = \frac{TP}{(TP + FP + FN)}, \tag{3.6}$$

На зображенні 3.13 показані приклади значень балів для критерію перетину та об'єднання.

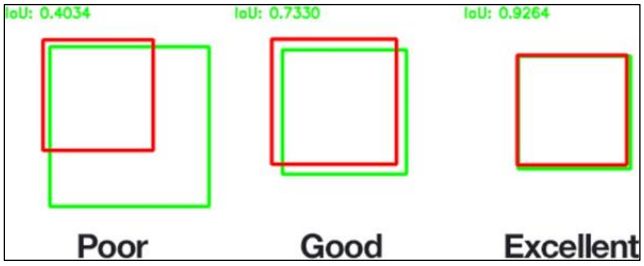


Рисунок 3.13 – Обчислення Індексу перекриття (IoU)

Прогнозовані межі, які значно перетинаються з реальними, отримують вищі оцінки, ніж ті, що мають менше перетину. Це робить IoU ефективним показником для оцінки дії будь-яких об'єктних детекторів. Важливо, щоб прогнозовані та реальні межі максимально точно збігалися – IoU може врахувати це [20]. Остаточний результат розпізнавання об'єктів отримується за допомогою трьох підходів: визначення рамки та класифікація. Перш за все, використовуються залишкові блоки, які розбивають зображення на рівні сітки. Кожна сітка відповідає за виявлення та локалізацію об'єкта всередині неї. Сітки передбачають координати обмежувальної рамки, мітку об'єкта та ймовірність його присутності. Клас з найвищою ймовірністю призначається конкретній клітинці сітки, і цей процес повторюється для всіх клітинок на зображенні. На рисунку 3.14 показано використання трьох методів розпізнавання об'єктів: залишкові блоки, регресія обмежувальної рамки та IoU.

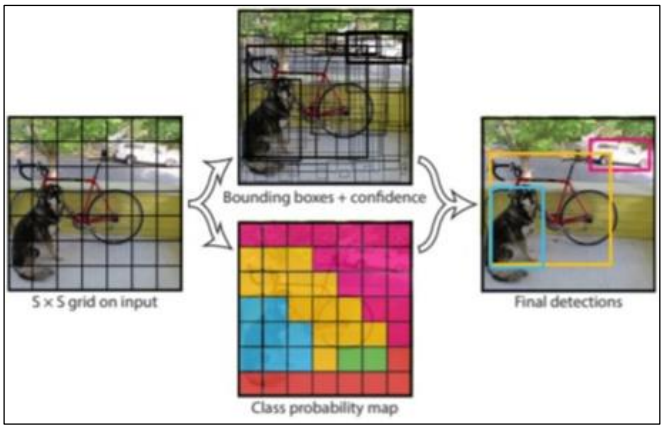


Рисунок 3.14 – Використання YOLO для ідентифікації трьох типів об'єктів

Маючи сітку, можна виявляти об'єкти на зображенні, розглядаючи кожну клітинку окремо. Кожній клітинці сітки можна присвоїти вектор, що описує її

вміст. Наприклад, якщо центр собаки знаходиться в певній клітинці, ця клітина буде оцінюватися наявністю собаки. Оцінка варіюється від «0,0» до «1,0», де «0,0» вказує на найнижчий рівень достовірності, а «1,0» - на найвищий. Якщо об'єкт відсутній, оцінка буде «0,0», а якщо модель впевнена у своєму прогнозі, - «1,0». Ці оцінки вказують на впевненість моделі в наявності об'єкта та точності рамки. Кожна рамка складається з п'яти параметрів: координати x та y центру, ширина, висота та достовірність. Координати «(x, y)» позначають центр передбаченої рамки, а ширина та висота - частки відносно розміру зображення. Достовірність визначається за допомогою IOU (Intersection Over Union), яка вимірює ступінь перетину передбаченої та фактичної рамок, поділену на їх об'єднання.

До того, як ми встановимо межі обмеження та бали достовірності, кожна клітина має передбачити клас об’єкта. Це передбачення класу представлено довжиною одного гарячого вектора C, який відповідає кількості класів у наборі даних. Проте важливо відзначити, що, незважаючи на те, що кожна клітина може передбачити будь-яку кількість обмежувальних рамок та їхніх балів достовірності, вона передбачає лише один клас. Це обмеження властиве самому алгоритму YOLO, тому якщо в одній клітині сітки є кілька об’єктів різних класів, алгоритм не зможе правильно класифікувати їх обидва. Отже, кожне передбачення з сітки матиме (формула 3.7 та 3.8):

$$C + B \times 5, \tag{3.7}$$

де C – кількість класів;

B – кількість встановлених меж і обмежень.

$$S \times S \times (C + B \times 5). \tag{3.8}$$

Результатом множення B на 5 є значення, що враховує (x, y, w, h, достовірність) для кожного положення. Оскільки на кожному зображенні

присутні комірки сітки розміром S×S, прогноз моделі стає тензором зазначеної форми.

3.3 Архітектурні та навчальні впорядкування YOLO

Оригінальний алгоритм YOLO був розроблений Дж. Редмоном у фреймворку під назвою Darknet, який є гнучким дослідницьким інструментом, написаним на низькорівневих мовах програмування. Darknet створив ряд систем виявлення об'єктів у реальному часі в області комп'ютерного зору, включаючи YOLO, YOLOv2, YOLOv3 та YOLOv4.

Оригінальний YOLO був першим серед мереж виявлення об'єктів, що об'єднав у собі проблеми маркування обмежувальних прямокутників та класифікації міток в одній нейромережі.

YOLOv2 вніс ряд ітераційних покращень у порівнянні з оригінальним YOLO, таких як вища роздільна здатність та впровадження блоків прив'язки.

YOLOv3 базується на попередніх версіях, додавши оцінку достовірності об'єктів до передбачень обмежувальних рамок, встановивши зв'язки з рівнями основної мережі та роблячи передбачення на трьох різних рівнях деталізації для покращення виявлення менших об'єктів [21].

YOLOv4 - це двоетапний детектор з кількома компонентами (рис. 3.15).

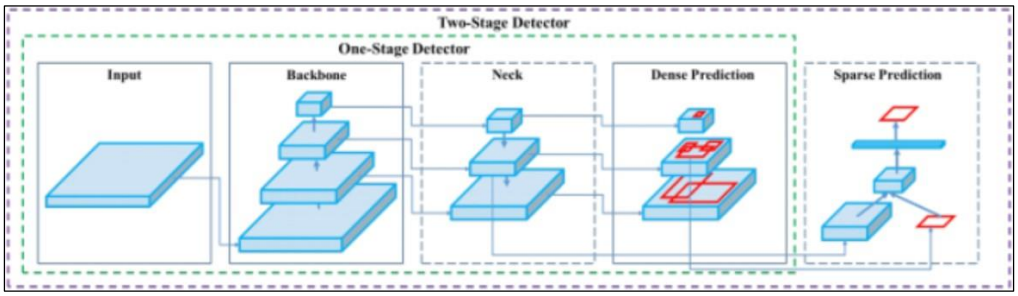


Рисунок 3.15 – Архітектура YOLO

Модель YOLO складається з трьох основних компонентів: голови (head), шиї (neck) та хребта (backbone) [22]. Хребет є основною архітектурою глибокого навчання, яка виконує функцію екстракції ознак і складається зі згорткових шарів. Всі базові моделі вважаються моделями класифікації.

Шия використовує функції згорткових шарів хребта разом з повністю зв'язаними шарами для передбачення ймовірностей і координат обмежувальних рамок. Вона об'єднує карту ознак з різних ступенів хребта, функціонуючи як агрегатор ознак. Головний компонент, відомий також як детектор об'єктів, фактично визначає область, де може перебувати об'єкт, але не вказує, який саме об'єкт присутній у цій області.

Для навчання детектора розпізнаванню об'єктів на зображеннях необхідно надати навчальні дані. Це можуть бути дані з будь-якими об'єктами або класами - тваринами, людьми або предметами. Наступним кроком є анотування обраних об'єктів, позначаючи цільові об'єкти за допомогою обведення їх рамками.

Далі наступним кроком буде почати навчання моделі. У цій системі наразі розроблено 5 різних моделей. Починаючи з YOLOv5 nano, яка є найменшою і найшвидшою, і закінчуючи YOLOv5 extra-large, яка є найбільшою і найпотужнішою. YOLOv5n – це представлена наномодель, спроектована для використання на периферійних пристроях, в Інтернеті речей, із підтримкою OpenCV DNN. Ця модель займає менше 2,5 МБ у форматі INT8 і приблизно 4 МБ у форматі FP32, ідеально підходить для мобільних рішень. YOLOv5s – це компактна модель з приблизно 7,2 мільйонами параметрів, ідеальна для виконання на ЦП. YOLOv5m – модель середнього розміру з 21,2 мільйонами параметрів, можливо, найбільш підходяща для багатьох наборів даних і завдань навчання, оскільки вона забезпечує хороший баланс між швидкістю та точністю. YOLOv5l – це велика модель з 46,5 мільйонами параметрів, призначена для виявлення менших об’єктів. Нарешті, YOLOv5x – найбільша модель серед п'яти, з найвищим серед них mAP. Хоча вона менш швидка порівняно з іншими моделями і має 86,7 мільйонів параметрів. На рисунку 3.16 представлені архітектурні шари YOLO.

Отже, різниця між цими версіями залежить від вхідних даних моделей для навчання та вимог. Кожна версія обмежена розміром вхідних даних, тому при виборі версії важливо враховувати їх кількість. Наприклад, YOLOv5n придатний для тренування в інтернеті речей, оскільки працює з невеликою кількістю даних.

YOLOv5s та YOLOv5n використовуються для масштабування моделі в глибину і ширину. Крім того, всі ці моделі відрізняються за швидкістю: чим більше модель, тим швидше вона працює.

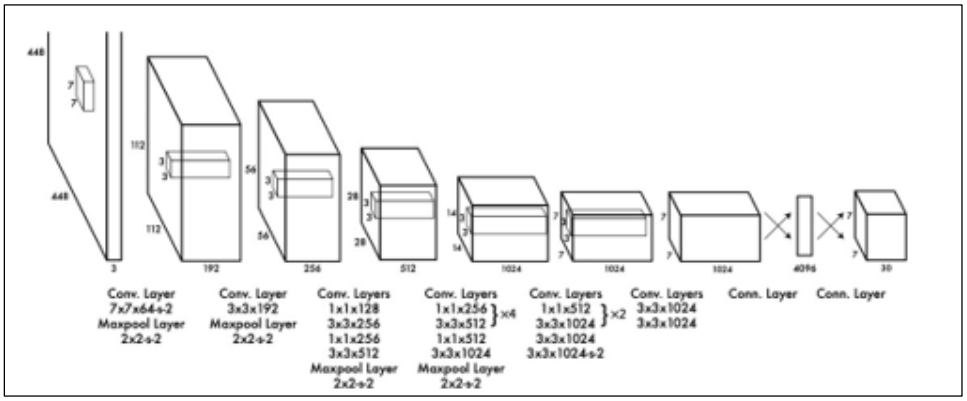


Рисунок 3.16 – Архітектурна конфігурація згорткової мережі

YOLOv5 використовує генетичний алгоритм для створення блоків прив’язки, цей процес називається автоматичною прив’язкою. Вона перераховує блоки прив’язки відповідно до даних у поєднанні з алгоритмом k-Means для створення еволюційних опорних блоків k-Means. Це одна з причин успішності YOLOv5 навіть на різних наборах даних. На рисунку 3.17 зображено цикл тренування моделі YOLO.

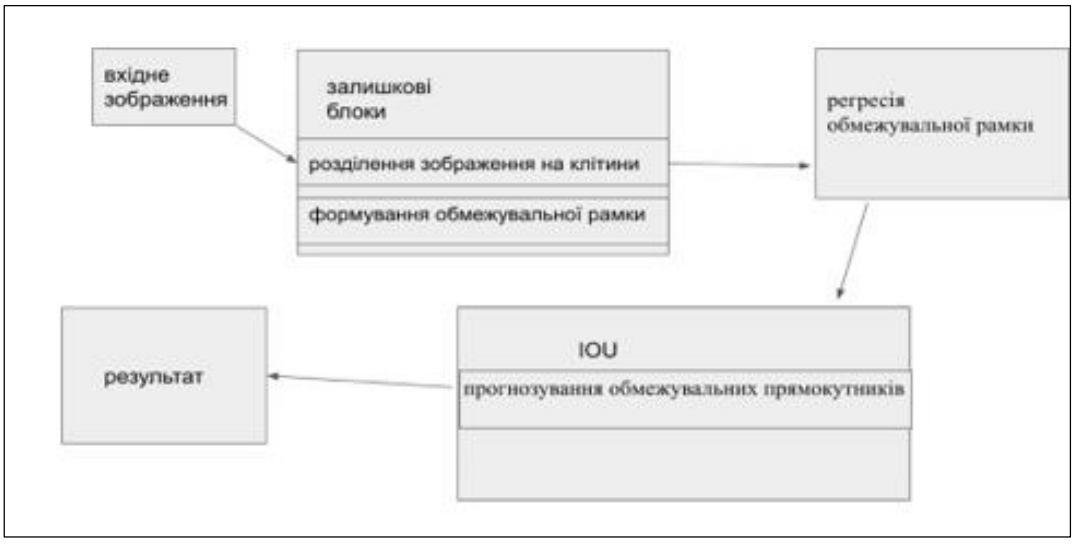


Рисунок 3.17 – Тренувальний процес моделі YOLO

YOLO дозволяє вибирати модель з п'яти варіантів, враховуючи вхідні дані, чи це навчання детектора в режимі реального часу для периферії, або розгортання

передової моделі виявлення об'єктів на хмарних графічних процесорах. Крім того, ця модель дозволяє експортувати навчену модель. Проте для повного функціонування конвеєра виявлення об'єктів необхідно не лише навчання та вивід моделей, а й розгортання, особливо при застосуванні у реальних умовах. Перед розгортанням намагаємося конвертувати навчену модель у відповідний формат. Крім того, існує функціонал для журналювання, який полегшує аналіз показників продуктивності в будь-який момент після завершення навчання моделі.

4 РЕЗУЛЬТАТИ ЕКСПЕРИМЕНТАЛЬНОГО ДОСЛІДЖЕННЯ МЕТОДУ YOLO ДЛЯ КЛАСИФІКАЦІЇ ОБ'ЄКТІВ

4.1 Програмне середовище

Моделювання було виконано в середовищі Visual Studio Code 2022.2 з використанням бібліотек OpenCV та darknet, мовою програмування Python. YOLOv5 інтегровано з платформою Weights & Biases (W&B) для візуалізації в реальному часі, реєстрації помилок та інформації, а також відстеження процесу тренування моделі у хмарному середовищі. Це дозволяє краще порівнювати різні запуски, проводити самоаналіз та покращувати видимість і співпрацю в команді [23]. На рисунку 4.1 представлено середовище W&B для відстеження тренування моделі.

Рисунок 4.1 – Використання W&B (Weights & Biases)

4.2 Результати програмної моделі

Перш ніж розпочати тренування, необхідно вибрати модель. Наразі існують такі версії: v5n, v5s, v5m, v5l, v5x. На рисунку 4.2 показано порівняння цих моделей. Основна відмінність між версіями полягає в розмірі моделей. Час

виведення та складність моделей збільшуються від найменших значень у моделі YOLOv5s, до YOLOv5m, YOLOv5l і найвищих значень у моделі YOLOv5x. Також важливими факторами є розмір і кількість зображень у наборі даних, а також можливості середовища та точність.

Оскільки у нас невеликий набір даних, для тренування моделі ми виберемо версію YOLOv5s, яка є швидкою та добре працює з такими наборами даних.

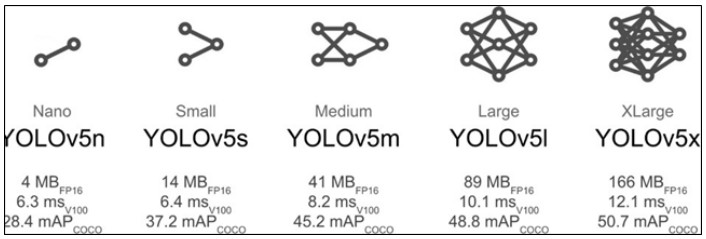


Рисунок 4.2 – Порівняння різних версій моделі YOLO

Для проведення аналізу буде використаний набір даних MakeML, в колекції міститься 1000 зображень. Для звичайного навчання цього набору потрібно менше половини години. Для експериментального дослідження ми клонуємо репозиторій YOLO та налаштовуємо необхідні залежності для його запуску. Після завантаження репозиторію та даних потрібно конвертувати анотації в формат, прийнятний для YOLO, а саме у текстовий формат. Набір MakeML вже містить анотовані зображення і відповідає формату PASCAL VOC XML. Для YOLO необхідно перетворити формат у .txt для кожного зображення, де кожен рядок текстового файлу визначає обмежувальну рамку. Нижче наведено приклад анотацій у форматі PASCAL VOC XML (див. рисунок 4.3).

```
<annotation>
<folder>images</folder>
<filename>file.png</filename>
<size>
<width>267</width>
<height>350</height>
<depth>3</depth>
<name>trafficlight</name>
<pose>Unspecified</pose>
<truncated>0</truncated>
<occluded>0</occluded>
<difficult>0</difficult>
<bndbox>
<xmin>16</xmin>
<ymin>12</ymin>
<xmax>13</xmax>
<ymax>22</ymax>
</bndbox>
</object>
</annotation>
```

Рисунок 4.3 – Анотації у форматі PASCAL VOC XML

Приклад, що наводиться, описує файл з розмірами 267×400×3, що включає 3 теги об'єктів, представлені 3 обмежувальними рамками. Клас визначається тегом "name", а деталі обмежувальної рамки - тегом "bndbox", що описуються координатами верхнього лівого (x_min, y_min) та нижнього правого (x_max, y_max) кутів. У форматі маркування YOLO для кожного файлу зображення створюється файл .txt з анотаціями, такими як клас об'єкта, його координати, висота та ширина. Оскільки YOLO вимагає наявності файлу .txt, структура даних буде виглядати, як показано на рисунку 4.4.

```
0  0.480109 0.631250 0.692969 0.713278
0  0.741016 0.522222 0.314844 0.933333
27 0.785937 0.506944 0.039062 0.151030
```

Рисунок 4.4 – Перетворення опису малюнка у текстовий документ

Нехай x_center та y_center відповідають за центр об'єкта, використовуючи нормалізовані координати поля. Кожен об'єкт відображається у вигляді окремого рядка, де координати нормалізовані від 0 до 1. Кожному класу надається індекс, що починається з 0. Отже, спочатку необхідно перетворити координати об'єктів з формату XML у TXT, що можна зробити за допомогою онлайн-сервісів або власної функції. Після цього зображення датасету розділяється на набори для навчання, перевірки та тестування, кожен з яких має окрему папку для кожної категорії об'єктів. Ррозглянемо параметри, які використовуються під час навчання моделі:

- img: розмір зображення. Зображення завжди квадратне, і розмір оригінального зображення змінюється з урахуванням збереження пропорцій. Довша сторона зображення автоматично змінюється до цього розміру, а коротша сторона доповнюється сірим кольором;
- batch: розмір партії зображень;
- epochs: кількість епох для навчання;
- data: YAML-файл даних, що містить інформацію про набір даних (шляхи до зображень, мітки);

- workers: кількість обробників CPU, які використовуються;
- cfg: архітектура моделі. Моделі впорядковані за складністю, і можна вибрати ту, яка відповідає складності завдання виявлення об'єктів. Якщо потрібно визначити власну архітектуру, то слід створити YAML-файл у папці моделей з описом архітектури мережі;
- weights: попередньо підготовлені ваги, які можуть бути використані для початку тренування;
- name: різні налаштування, пов'язані з навчанням, такі як логування процесу тренування. Ваги для тренування будуть збережені у папці з назвою "runs/train/name";
- hyp: YAML-файл, що описує вибір гіперпараметрів;

Файл конфігурації даних у форматі YAML визначає деталі набору даних, який використовується для навчання моделі. Він містить такі параметри:

- місця розташування зображень для тренування, тестування та перевірки, позначені як train, test і val відповідно;
- параметри nc, який вказує кількість класів у наборі даних, та name, який містить імена цих класів. Індекси у списку name використовуються як ідентифікатори класів у коді.

Для створення нового файлу з назвою road_sign_data.yaml потрібно виконати наступні кроки:

- створити файл і помістити його у папку yolov5/data;
- оскільки обсяг набору даних невеликий, а на зображеннях немає великої кількості об'єктів, будемо використовувати найменшу з попередньо підготовлених моделей YOLO;
- визначимо розмір пакета (batch) 32 та розмір зображення 640. Навчимо модель протягом 100 епох.

Для тренування буде використано 24 шари згорткової мережі.

Мережа розпізнавання складається з 24 згорткових шарів, за якими слідують 2 повністю зв'язані шари. Використання згорткових шарів розміром 1x1 допомагає зменшити простір ознак з попередніх шарів.

Для проведення експерименту на вході буде використано зображення з такими параметрами: ширина (W), висота (H), масштаб (S). Крім того, у нас є набір класів (C), які будуть використані для розпізнавання та включені у вихідний результат. Кольори (COLORS), що генеруються для вихідного результату, і достовірність розпізнавання об'єкту на зображенні (P) також враховані.

Поза розпізнаванням дорожніх знаків, програма також може ідентифікувати автомобілі. На рисунке 4.5 представлені вхідні дані та результат роботи програми.

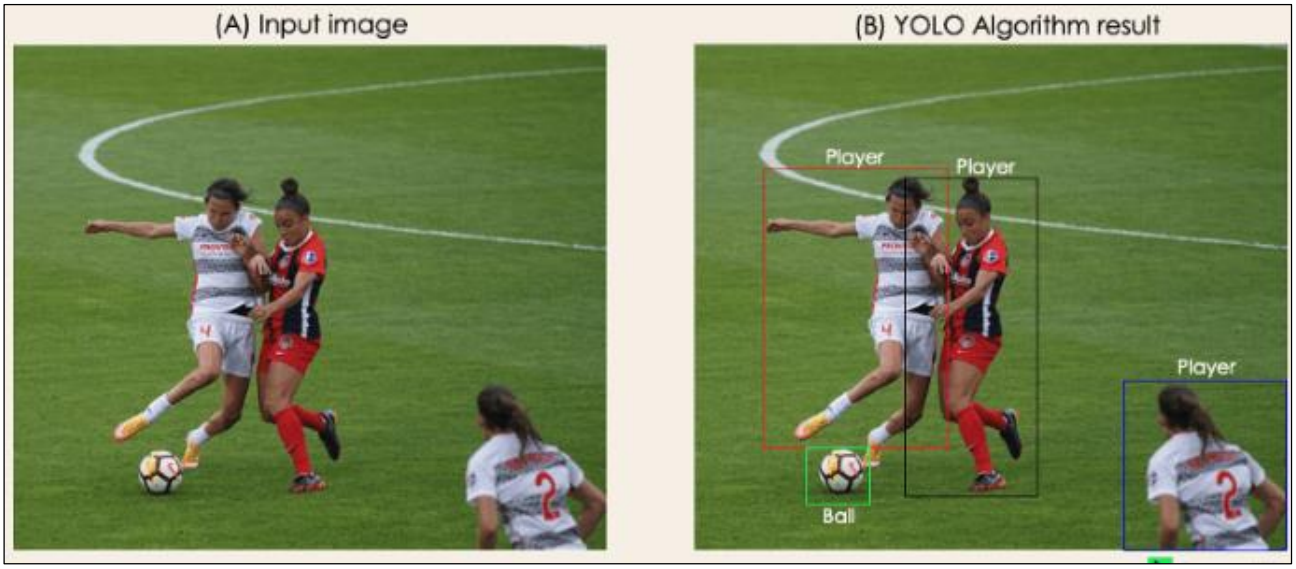


Рисунок 4.5 – Зображення на вході та вихідний результат виконання алгоритму

Вихідне зображення, з якого ми отримуємо параметри W та H, відображається таким чином:

```
img = cv2.imread(ar.img)
w = img.shape[1]
h = img.shape[0]
```

Потім ми генеруємо кольори та читаємо назви класів, що зручно формуємо у список.

Читаємо ваги та файл конфігурації для створення мережі:

```
n = cv2.dnn.readNet(as.w, ar.conf)
```

Архітектура YOLO має кілька вихідних рівнів, які дають прогнози.

Функція get_output_layers() повертає назви вихідних шарів.

Вихідний шар не з'єднаний з наступним:

```
def get_output_layers(n):
names = n.getLayerNames()
result_l = [names[i[0] - 1] for i in n.getUnconnectedOutLayers()]
return result_l
```

Також ми створимо функцію для малювання прямокутника результату з підписами класів об'єктів draw_bounding_box().

Потім ми збираємо прогнози з вихідних рівнів. Для кожного розпізнаваного об'єкту з кожного вихідного рівня (outputs), ми отримуємо достовірність виявлення об'єкту (P), ідентифікатор класу (C), параметри обмежувального прямокутника, ігноруючи виявлення, якщо P < 0,5.

Прогнози вихідних рівнів:

```
ots = net.forward(get output layers(net)) for o in ots:
for detY in o: s = dtY[6:]
clId = n.argmax(s) conf = s[clId]
if conf > 0.6:
cenX = int(dtY[0] * w) cenY = int(dtY[1] * h) we = int(dtY[2] * w) he =
int(dtH[3] * h) xe = cenX - we / 2
ye = cenY - he / 2
clId.append(class id) confC.append(float(conf))
```

Ми отримаємо ідентифікатор класу, достовірність і параметри обмежувальної рамки для кожного розпізнавання на кожному вихідному рівні.

Для навчання використовується Darknet Framework, який потім перетворюється для виконання виявлення.

Для виявлення об'єктів на зображенні за допомогою алгоритму YOLO потрібно створити 3 функції:

- фільтрація неперекриваючихся блоків на основі їх ймовірностей та порогу;
- обчислення перетину об'єктів між двома блоками;
- реалізація функції Non-max suppression.

Нижче подані відповідні функції виявлення об'єктів (рис 4.6)

```
def check filtr box(box conf, bxs, bx clss prbs, thrs=0.6):
    bxSc = box conf * bx clss prbs
    bxCl = K.argmax(bxSc, -1)
    box class scores = K.max(bxSc, -1)
    filterM = box class scores > thrs
    sc = t.bool mask(box class scores, filterM)
    bxs = t.bool mask(bxs, filterM)
    clss = t.bool mask(bxCl, filterM)
    return sc, bxs, clss

def iou(box1, box2):
    k1 = max(box1[0], box2[0])
    p1 = max(box1[1], box2[1])
    k2 = min(box1[2], box2[2])
    p2 = min(box1[3], box2[3])
    int ar = (k2 - k1) * (p2 - p1)
    box1 area = (box1[2] - box1[0]) * (box1[3] - box1[1])
    box2 area = (box2[2] - box2[0]) * (box2[3] - box2[1])
    unArea = box1 area + box2 area - int ar
    iou = int ar / unArea
    return iou

def get suppr(sc, bxs, clss, maximum bxs=10, iouvalue=0.5):
    mxtensor = tf.Variable(maximum bxs, dtype='int32')
    K.get session().run(tf.variables initializer([mxtensor]))
    nm ind  =  tf.image.non max suppression(bxs,  sc,  maximum bxs,
iouvalue)
    scrs = K.gather(sc, nm ind)
    bxs = K.gather(bxs, nm ind)
    clss = K.gather(clss, nm ind)
    return scrs, bxs, clss
```

Рисунок 4.6 – Функція виявлення об'єктів

Скористаємось рисунком 4.7 як вихідним матеріалом.

Рисунок 4.7 – Зображення, призначене для використання у експерименті

Розмір вхідного зображення складає 608×608 пікселів. Після обробки кожного об'єкту буде виділено результативну рамку. Метод YOLO одночасно створює прямокутники для всіх об'єктів і класів [24-28]. Кожен об'єкт, який знаходиться всередині блоку, оброблюється саме цим блоком. Формули для визначення розмірів прогнозованого прямокутника (формули 4.1-4.4):

$$b_x = \sigma(t_x) + c_x, \tag{4.1}$$

$$b_y = \sigma(t_y) + c_y, \tag{4.2}$$

$$b_0 = p_0 + e^{t_w}, \tag{4.3}$$

$$b_2 = p_2 + e^{t_h}. \tag{4.4}$$

У результаті утворюються прямокутники, які можуть містити об'єкти, і мережа не має передбачати окремий остаточний розмір об'єкта. Замість цього, вона лише коригує розмір найближчого прямокутника так, щоб він відповідав розміру об'єкта. У YOLO ці прямокутники, що представляють собою ширину та висоту об'єктів на зображенні, змінюються відповідно до розміру мережі. Чим більше прямокутників-кандидатів використовується, тим вище значення IoU (рис. 4.8).

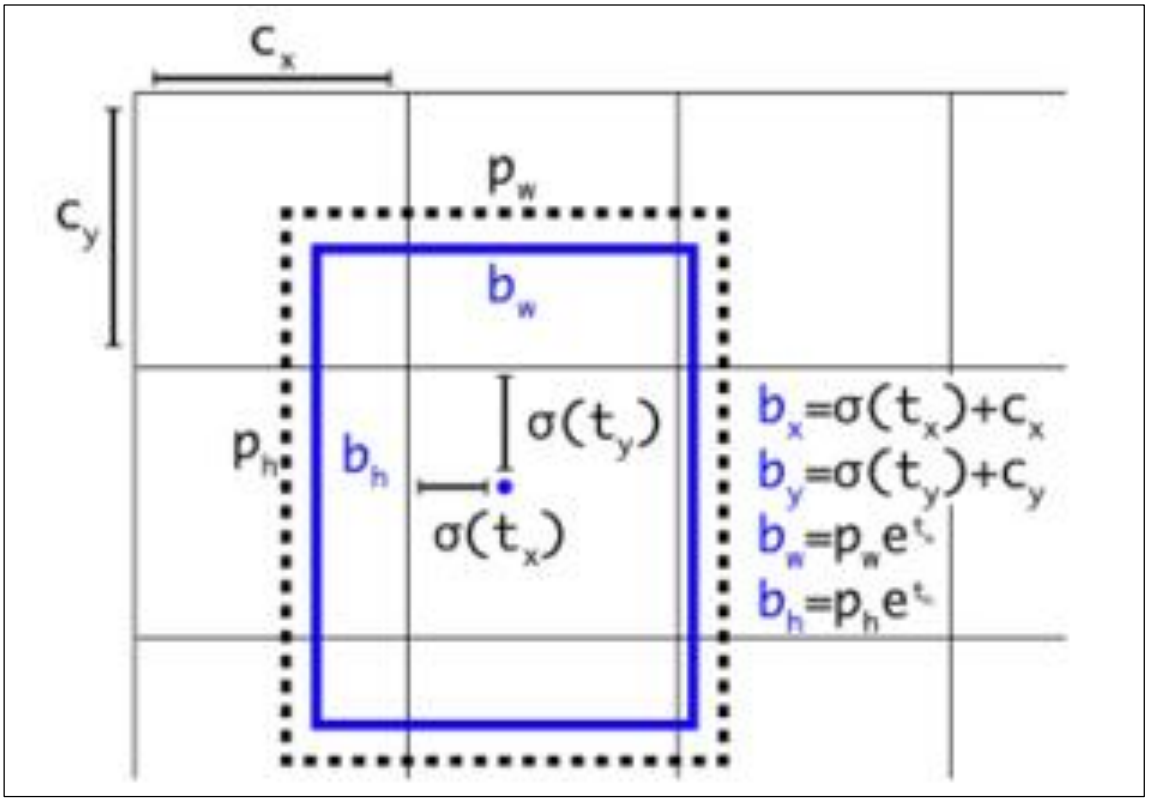


Рисунок 4.8 – Обчислення межі обмежень

Отже, прямокутник, який виникає в результаті успішного вибору шаблонів прив'язки, призначається відповідним клітинкам (див. рис. 4.9).

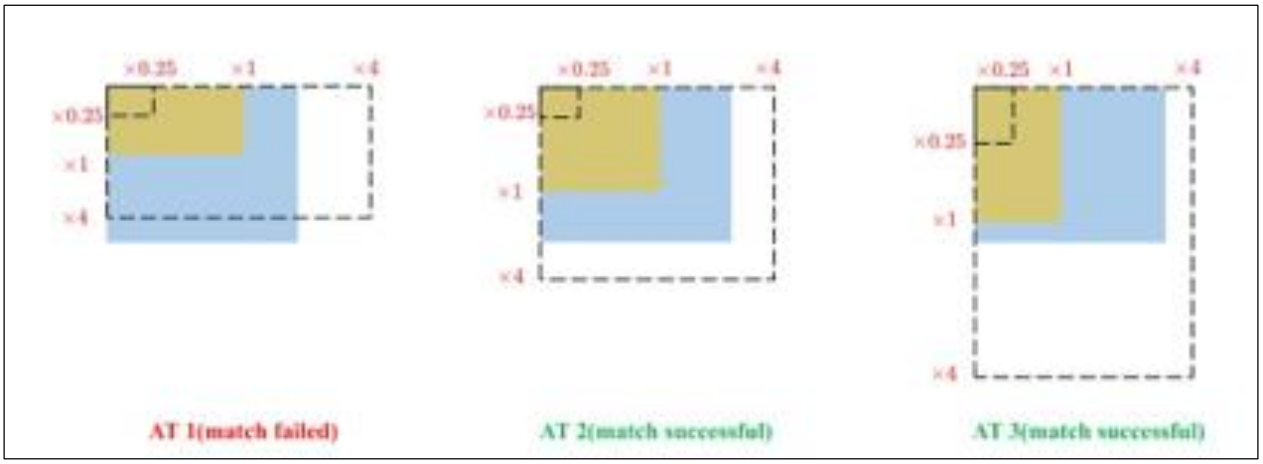


Рисунок 4.9 – Розрахунок площі істинно позитивного прямокутника

Останній рівень враховує як ймовірності класу, так і координати обмежувальної рамки. Змінюємо розміри обмежувальної рамки (ширина і висота) на відповідні пропорції зображення, щоб вони були в межах від 0 до 1. Координати x і y обмежувальної рамки параметризуються як зсуви відносно розташування комірки сітки, тому вони також обмежені між 0 і 1. Для останнього шару застосовуємо функцію лінійної активації, тоді як усі інші шари використовують наступну виправлену лінійну активацію (рис. 4.10):

$$\phi(x) = \begin{cases} x, & \text{if } x > 0 \\ 0.1x, & \text{otherwise} \end{cases}$$

Рисунок 4.10 – Лінійна активаційна функція

Визначимо та створимо таблицю для оцінки швидкодії та точності різних версій алгоритму YOLO. В якості вхідних даних ми використовуємо зображення розміром 640×640 пікселів. Вимірювання проводяться на процесорі з тактовою частотою 2.3 ГГц із чотирма ядрами Intel Core i7. У таблиці 4.1 наведено результати виявлення різних версій алгоритму YOLO.

Таблиця 4.1 – Виявлені версії алгоритму YOLO

Модель	Швидкодія, ms	Розмір пікселів
YOLOv3	493	640
YOLOv4	588	640
YOLOv5s	99	640
YOLOv5m	230	640
YOLOv5l	450	640
YOLOv3	789	1280
YOLOv4	890	1280
YOLOv5s	560	1280
YOLOv5m	490	1280
YOLOv5l	678	1280

Мета діаграми у рисунку 4.11 полягає в створенні продуктивної моделі детектора об'єктів, де вісь x відображає час, а вісь y - результат.

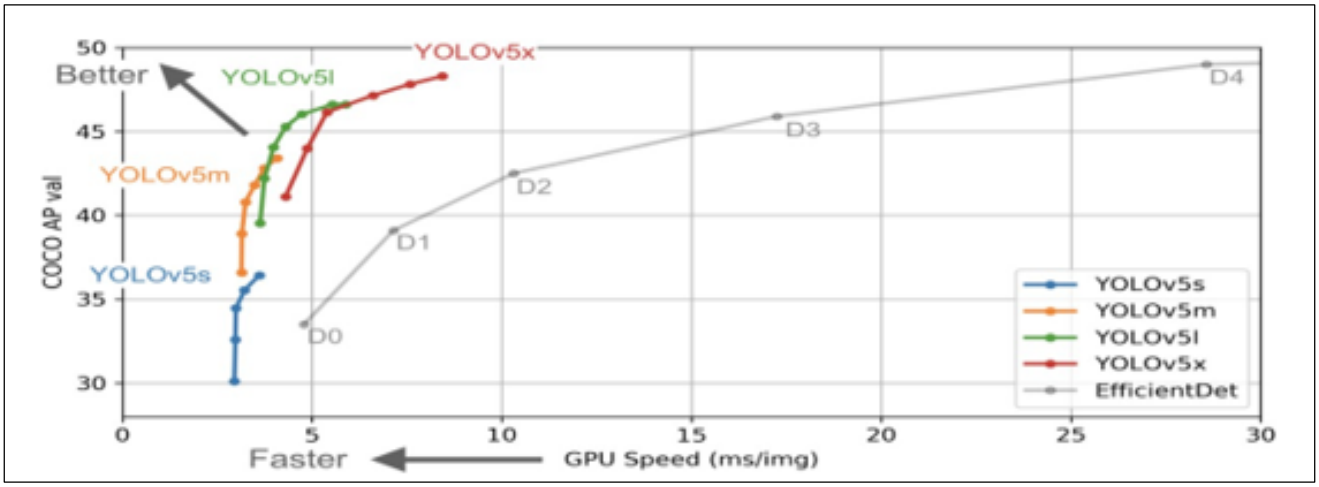


Рисунок 4.11 – Продуктивність моделей YOLO

Попередні дані демонструють, що YOLOv5 ефективніше досягає цієї мети порівняно з попередніми версіями алгоритму YOLO. Давайте розглянемо точність виявлення частково закритих об'єктів на зображеннях, яка представлена у таблиці 4.2.

Таблиця 4.2 – Рівень точності у визначенні об'єктів

Модель	Точність розпізнавання, %
YOLOv1	17
YOLOv2	22
YOLOv3	24
YOLOv4	34
YOLOv5	87

Найточніша версія YOLOv5, YOLOv5x, може обробляти зображення значно швидше з аналогічним рівнем точності, ніж інші варіанти. Щоб підвищити продуктивність цієї моделі, пропонуються такі заходи:

- використання більших моделей, ніж YOLOv5, для поліпшення продуктивності моделей передачі навчання;
- налаштування вхідних параметрів для тренування моделей залежно від вхідних даних.

Однією з ключових покращень у версії YOLOv5, порівняно з попередніми моделями, є впровадження мозаїчної аугментації даних та автоматична прив'язка обмежувальної рамки. Мозаїчна аугментація працює наступним чином: чотири вихідні зображення об'єднуються в одне, що дозволяє застосовувати кілька методів аугментації за один крок. Вона створює чотири випадкові вирізки зображення, зберігаючи відносний масштаб об'єктів, що покращує виявлення частково прихованих об'єктів на зображеннях. Крім того, це дозволяє комбінувати класи, які раніше не можна було виявити разом на одному зображенні, що підвищує точність моделі, особливо в умовах незбалансованих наборів даних.

Ще однією можливістю моделі є випадкове налаштування відтінку в кольорових зображеннях як етапу попередньої обробки.

Отже, версія YOLOv5 працює значно точніше та швидше за свої попередники завдяки впровадженню мозаїчної аугментації. У цій версії спостерігаються покращення у виявленні об'єктів на зображеннях, що частково перекриваються.

ВИСНОВКИ

У рамках даної роботи було проведено дослідження ефективності сучасних методів глибокого навчання для визначення положення тіла людини на відеопотоці з безпілотного літального апарату (БПЛА). Дослідження охопило огляд існуючих методів відстеження об'єктів на відеопотоці, вибір та порівняння різних архітектур глибокого навчання, а також проведення експериментів з визначенням їхньої ефективності.

Результати експериментів свідчать про великий потенціал застосування глибокого навчання для відстеження положення тіла людини на відеопотоці з БПЛА. Методи, засновані на конволюційних нейронних мережах (CNN), виявили високу точність визначення ключових точок тіла, що робить їхнє використання перспективним для практичних застосувань.

Однак, важливо враховувати, що ефективність моделей глибокого навчання може залежати від конкретного сценарію використання, особливостей вхідних даних та інших факторів. В процесі дослідження було виявлено, що оптимальний вибір архітектури та параметрів навчання може значно підвищити точність визначення положення тіла.

Загальні висновки дозволяють стверджувати, що глибоке навчання має великий потенціал для застосування в системах відстеження об'єктів на відеопотоці з БПЛА. Проте, для практичного впровадження цих методів, необхідно продовжувати дослідження та оптимізацію, враховуючи конкретні умови використання та вимоги до продуктивності.

Це дослідження може слугувати основою для подальших робіт у галузі відстеження об'єктів на відеопотоці та розробки систем безпілотного контролю, що використовують передові технології глибокого навчання для оптимізації функціональності та точності визначення положення об'єктів на землі.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Deep Learning (Part 1):Understanding Basic Neural Networks URL - https://medium.com/@Coursesteach/deep-learning-part-1-86757cf5a0c3 (дата звернення: 17.12.2023).

2. Deep Learning Applications for Computer Vision URL - https://serokell.io/blog/deep-learning-for-computer-vision (дата звернення: 17.12.2023).

3. Usage of Unmanned Aerial Vehicles in Medical Services URL - https://www.researchgate.net/publication/362136836_Usage_of_Unmanned_Aerial_Vehicles_in_Medical_Services_A_Review (дата звернення: 17.12.2023).

4. Drone and Controller Detection and Localization URL - https://www.researchgate.net/publication/366156655_Drone_and_Controller_Detection_and_Localization_Trends_and_Challenges (дата звернення: 17.12.2023).

5. Classification in Machine Learning: An Introduction URL - https://www.datacamp.com/blog/classification-machine-learning (дата звернення: 17.12.2023).

6. What's the difference between CNN and RNN? URL - https://www.telusinternational.com/insights/ai-data/article/difference-between-cnn-and-rnn (дата звернення: 17.12.2023).

7. Гороховатский В.А. (2003) Распознавание изображений в условиях неполной информации, Харків: ХНУРЭ, 112с.

8. Gorokhovatskyi V., Putyatin Y., Gorokhovatskyi O, Peredrii O. (2018) Quantization of the Space of Structural Image Features as a Way to Increase Recognition Performance. The Second IEEE International Conference on DataStream Mining & Processing 21-25 August 2018, Lviv, Ukraine. pp. 464 – 467.

9. Gorokhovatskyi, V., Tvoroshenko, I., & Chmutov, Y. (2022). Застосування систем ортогональних функцій для формування простору ознак у методах класифікації зображень. Advanced Information Systems, 6(3), 5-12.

10. Гороховатський В.О., Гадецька С.В., Стяглик Н.І., Власенко Н.В. (2020) Класифікація зображень на підставі ансамблю статистичних розподілів за

класами еталонів для компонентів структурного опису. Радіоелектроніка, інформатика, управління, №4 , с. 85–94.

11. Gorokhovatskyi V., Gadetska S., Ponomarenko R. (2020) Recognition of Visual Objects Based on Statistical Distributions for Blocks of Structural Description of Image. Lecture Notes in Computational Intelligence and Decision Making. Proceedings of the XV International Scientific Conference “Intellectual Systems of Decision Making and Problems of Computational Intelligence” (ISDMCI’2019), Ukraine, pp. 501-512.

12. Gadetska, S.V., Gorokhovatskyi, V. O., Stiahlyk, N. I., Vlasenko, N.V. (2021) Statistical data analysis tools in image classification methods based on the description as a set of binary descriptors of key points. Radio Electronics, Computer Science, Control, №4, pp. 58-68.

13. Gorokhovatskyi, V., Stiahlyk, N., Tsarevska, V. (2021). Combination method of accelerated metric data search in image classification problems. Advanced Information Systems, 5 (3), pp. 5–12.

14. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., and Al-Dhaifallah M. (2022) Classification of Images Based on a System of Hierarchical Features, Computers, Materials & Continua, 72(1), pp. 1785-1797.

15. Tvoroshenko I., and Gorokhovatskyi V. (2022) The Application of Hybrid Intelligence Systems for Dynamic Data Analysis, International Journal of Engineering and Information Systems, 6(2), pp. 40-48.

16. Gadetska S., Gorokhovatskyi V., Stiahlyk N., Vlasenko N. (2022) Aggregate Parametric Representation of Image Structural Description in Statistical Classification Methods. In CEUR Workshop Proceedings: Computer Modeling and Intelligent Systems (CMIS-2022), 3137, pp. 68-77.

17. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., and Zeghid M. (2022) Cluster representation of the structural description of images for effective classification, Computers, Materials & Continua, 73 (3), pp. 6069–6084.

18. Gorokhovatskyi, V., Vlasenko, N. (2021). Редукція опису зображення у складі множини дескрипторів на основі метричного критерію інформативності. Advanced Information Systems, 5(4), pp. 10-16.

19. Гороховатский, В. А. (2014). Структурный анализ и интеллектуальная обработка данных в компьютерном зрении.

20. Gorokhovatskyi, V., Rusakova, N., Tvoroshenko, I. (2020) The application of image analysis methods and predicate logic in applied problems of magnetic monitoring. Telecommunications and Radio Engineering, 79 (20), pp. 1801-1811.

21. Gorokhovatsky, V.A., Putyatin Y. P. (2009) Image Likelihood Measures of the Basis of the Set of Conformities. Telecommunications and Radio Engineering, 68 (9), p. 763–778.

22. M. A. Ahmad, V. Gorokhovatskyi, I. Tvoroshenko, N. Vlasenko, S. K. Mustafa (2021) The Research of Image Classification Methods Based on the Introducing Cluster Representation Parameters for the Structural Description, International Journal of Engineering Trends and Technology, 69(10), pp. 186-192.

23. Gorokhovatskyi V.A. (2018) Image Classification Methods in the Space of Descriptions in the Form of a Set of the Key Point Descriptors. Telecommunications and Radio Engineering, 77 (9), pp. 787-797.

24. Гороховатский В.А., Передрий Е.О. (2009) Корреляционные методы распознавания изображений путем голосования систем фрагментов. Радіоелектроніка, інформатика, управління, №1 (20), с.74–81.

25. Gorokhovatskyi V.A., Zamula A.A. (2016) Employment of Intelligent Technologies in Multiparametric Control Systems. Telecommunications and Radio Engineering. Vol. 75, No 19, p. 1775–1785.

26. Gorokhovatskyi O., Gorokhovatskyi V., Peredrii O. (2018) Analysis of Application of Cluster Descriptions in Space of Characteristic Image Features. Data, 3(4), 52.

27. Гороховатський В.О., Пупченко Д.В., Солодченко К.Г. (2018) Аналіз властивостей, характеристик та результатів застосування новітніх детекторів для визначення особливих точок зображення. Системи управління, навігації та зв’язку. С. 93–98.

28. Gorokhovatsky, A.V., Gorokhovatsky, V.A., Vlasenko, A.N., Vlasenko N.V. (2014) Quality Criteria for Multidimensional Object Recognition Based Upon Distance Matrices, Telecommunications and Radio Engineering, Vol. 73, No 18, pp. 1661 – 1670.

29. N. V. Bilous, I. A. Ahekian, and V. V. Kaluhin, “DETERMINATION AND COMPARISON METHODS OF BODY POSITIONS ON STREAM VIDEO,” RIC, no. 2, p. 52, Jun. 2023, doi: 10.15588/1607-3274-2023-2-6

30. Н. В. Білоус, І. А. Агекян, О. В. Рассоха, and О. В. Грамм, “ДОСЛІДЖЕННЯ МЕТОДІВ ДЛЯ РОЗРОБКИ ПРОГРАМНОЇ СИСТЕМИ РОЗПІЗНАВАННЯ ЕМОЦІЙ ТА ВИЗНАЧЕННЯ СТАНУ ЗДОРОВ’Я ЛЮДИНИ,” Біоніка інтелекту. – Харків : ХНУРЕ, vol. №1 (94), pp. 65–70, 2020, doi: 10.30837/bi.2020.1(94).