

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра Інформатики
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

ДОСЛІДЖЕННЯ НЕЙРОМЕРЕЖНИХ ЗАСОБІВ ДЛЯ
ДЕТЕКТУВАННЯ ВІЗУАЛЬНИХ ОБ'ЄКТІВ НА ЗОБРАЖЕННІ

(тема)

Виконала:

студент 2 курсу, групи ІНФМ-21-1

Комаркова М.Д.

(прізвище, ініціали)

Спеціальності 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

Керівник проф. Гороховатський В.О.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри

(підпис)

Кобилін О.А.
(прізвище, ініціали)

2022 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)Кафедра Інформатики
(повна назва)Рівень вищої освіти другий (магістерський)Спеціальність 122 Комп'ютерні науки
(код і повна назва)Тип програми освітньо-професійнаОсвітня програма Інформатика
(повна назва освітньої програми)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«____» _____ 2022 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУстудентові Комарковій Марії Дмитрівні

(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження нейромережних засобів для детектування візуальних об'єктів на зображенні

затверджена наказом по університету від 9 листопада 2022 року № 1469Ст

2. Термін подання студентом роботи до екзаменаційної комісії 22 листопада 2022 р.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 _____

1. Огляд методів детектування згорткових мереж: CNN, R-CNN, Fast R-CNN, Faster R-CNN, SSD, YOLO.

2. Математична модель методу YOLO з використанням IoU.

3. Метод немаксимального придушення.

4. Архітектура та організація навчання YOLO.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) актуальність дослідження детектування об'єктів на зображеннях, тестові та результативні зображення з об'єктами, постановка задачі, вихідні дані, перспективи дослідження, висновки.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Консультант з дотримання діючих стандартів та норм	Доцент Творошенко І.С.		

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	09.11.2022	
2	Аналіз завдання, підбір літератури	09.11.22-10.11.22	
3	Аналіз літератури з досліджуваної проблеми	11.11.22	
4	Аналіз згорткових методів детектування об'єктів	12.11.22	
5	Проведення експериментального дослідження методу YOLO	12.11.22-13.11.22	
6	Програмна реалізація	13.11.22-14.11.22	
7	Оформлення пояснювальної записки	14.11.22-14.11.22	
8	Перевірка на плагіат	21.11.2022	
9	Рецензування	22.11.2022	
10	Підготовка презентації та доповіді	22.11.2022	
11	Занесення роботи в електронний архів	29.11.2022	
12	Попередній захист кваліфікаційної роботи	30.11.2022	

Дата видачі завдання 9 листопада 2022 р.

Студент _____
(підпис)

Керівник роботи _____
(підпис)

проф. Гороховатський В.О.
(посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 73 с., 2 табл., 34 рис., 51 джерело.

ДЕТЕКТУВАННЯ ОБ'ЄКТІВ, ЗГОРТКОВІ МЕТОДИ, КЛАСИФІКАЦІЯ, КЛЮЧОВІ ТОЧКИ, КОМ'ЮТЕРНИЙ ЗІР, НАВЧАННЯ З УЧИТЕЛЕМ, НЕЙРОМЕРЕЖНІ ЗАСОБИ.

Об'єктом дослідження є виявлення дорожніх знаків та розташування автомобілей за допомогою методу виявлення об'єктів на зображеннях YOLO.

Метою кваліфікаційної роботи є вивчення сучасного нейромережевого алгоритму, за допомогою якого можливо детектування візуальних об'єктів на зображенні. Вхідні дані включають в себе зображення з різними класами об'єктів.

Представлено результати вивчення методу YOLO для розпізнавання об'єктів, який базується на згортковому алгоритмі, що включає розділення зображення на непересічні блоки, регресійний аналіз обмежувальних рамок на підставі параметрів навчання, визначення ймовірностей для заданих класів.

В результаті експерименту перевірено точність методу на прикладних даних. Використано методи математичної статистики, розпізнавання образів, навчання з учителем, аналіз даних.

Перспективою дослідження є вивчення засобів для пришвидшення реалізації методу та навчання на довільних наборах даних.

DETECTION OF OBJECT, CONVOLUTION METHODS, CLASSIFICATION, KEY POINTS, COMPUTER VISION, PATTERN RECOGNITION, NEURAL NETWORK TOOLS.

The object of the research is the detection of road signs and the location of cars using the method of object detection in YOLO images.

The purpose of the research is to study a modern neural network algorithm, which can be used to detect visual objects in an image. The input data includes images with different classes of objects. The results of the includes division into non-intersecting blocks, bounding box regression, and the last coefficient that predicts bounding boxes.

An experimental assessment of the speed and performance of this algorithm was carried out, the accuracy of the method was checked on various input data. This research uses mathematical statistics, pattern recognition, model training with a teacher, mathematical prediction, data analysis.

The output of the further research is to study the acceleration of the algorithm, training on sets with a small resolution, and optimization of the method.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	6
Вступ	7
1 Аналіз проблеми розпізнавання візуальних об'єктів	9
1.1 Проблематика виявлення об'єктів на зображеннях	9
1.2 Розпізнавання за описом із дескрипторів ключових точок	17
1.3 Нейромережні засоби розпізнавання об'єктів	22
1.4 Постановка задач дослідження.....	25
2 Аналіз підходу YOLO для розпізнавання об'єктів на зображеннях.....	26
2.1 Принципи побудови сучасних нейромережових засобів розпізнавання	26
2.2 Метод YOLO	33
2.3 Архітектура та організація навчання у YOLO.....	43
2.4 Анотація зображень.....	47
3 Результати експериментального дослідження методу YOLO для класифікації.....	50
3.1 Вибір програмного середовища	50
3.2 Результати програмного моделювання.....	51
Висновки.....	66
Перелік джерел посилання.....	68

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

YOLO – You Only Look Once (однопрохідна нейронно-мережева архітектура для виявлення об'єктів)

R-CNN – Region Convolutional Neural Network (багатопохідна нейронно-мережева архітектура для виявлення об'єктів)

SOFTMAX – нормована експоненціальна функція

IOU – Intersection Over Union – ступінь перекриття двох прямокутників.
Чим більше область перекриття, тим більше IOU

КЗ – комп'ютерний зір

КТ – ключова точка

ROI – Region Of Interest Pooling (дія для виявлення об'єктів у згорткових нейронних мережах)

IoT – Internet Of Things

NMS – Non-Maximal Suppression

ВСТУП

Людська зорова система є достатньо швидкою (працює у реальному часі) та точною, що дозволяє миттєво розпізнавати розташування та взаємодію об'єктів на зображеннях. Сучасне комп'ютерне бачення, спрямоване на автоматизацію процесів аналізу зображень, набуло величезну популярність тільки за останні роки. Розпізнавання образів використовується в різних галузях, таких як розробка безпілотних роботів, автоматизоване керування руху об'єктів фізичного походження, системи для пошуку, використання в медицині, маркетингу та ін. [1-8].

Швидкі та точні алгоритми виявлення об'єктів дають можливість швидко розпізнавати образи та передавати дані в реальному часі про сцену допоміжним пристроям оброблення та користувачам. Наприклад, безпілотні автомобілі в змозі відрізнати дорожні знаки, витримувати дистанцію між іншими автомобілями, що підвищить надійність та безпеку даних пристроїв [4, 5].

Розпізнавання об'єктів також знайшло застосування в сучасних дронах, які впроваджуються з науковою, військовою та розважальною метою. Наприклад, безпілотний пристрій направляє об'єктив на користувача чи будь-якого іншого об'єкта та слідкує за його траєкторією руху чи змінами, що буде корисним для пошуку несправностей електропередач.

За останнє десятиріччя швидко розроблюються, розвиваються та поліпшуються нейромережеві методи розпізнавання образів, а потужність пристроїв за підтримки могутніх комп'ютерних фірм збільшується, що дозволяє обробляти все більші об'єми інформації та вдосконалювати вивчення моделей.

Сучасні методи розпізнавання образів базуються на апараті машинного навчання, в яких визначається регіон або підмножина точок зображення, що

можуть містити об'єкт, а далі об'єкти класифікуються в межах позицій визначеного регіону [4].

Актуальність кваліфікаційної роботи обумовлюється тим, що сьогодні розпізнавання образів із впровадженням нейромережових засобів застосовується у всіляких галузях, включаючи маркетинг та масштабні наукові дослідження.

У вхідних даних також присутні фотографії з маленьким масштабом обличчя або з частково видимими об'єктами.

Необхідно зробити наступні дослідження:

- вивчити нейромережові методи та підходи для виявлення об'єктів на зображеннях;
- створити та застосувати прикладний алгоритм для виявлення візуальних об'єктів;
- зробити висновки щодо прикладної результативності методу.

Практична значущість полягає в тому, що завдяки розробленим алгоритмам є можливість обробляти значні об'єми вхідних даних, що дозволяє швидко аналізувати візуальні дані та відрізнити класи об'єктів.

Для виявлення та класифікації об'єктів використовують спектр методів. Популярні підходи на основі глибокого навчання з використанням згорткових нейронних мереж CNN, таких як R-CNN і YOLO, автоматично навчаються виявляти та класифікувати об'єкти на зображеннях.

YOLO є сучасною моделлю комп'ютерного зору для виявлення та класифікації об'єктів на зображеннях, яка використовує алгоритми залишкових блоків, регресію обмежувальної рамки та критерій Intersection Over Union [1-8].

У даній роботі будуть вивчатися та удосконалюватися сучасні програмні моделі комп'ютерного зору для здійснення класифікації об'єктів на зображеннях.

1 АНАЛІЗ ПРОБЛЕМИ РОЗПІЗНАВАННЯ ВІЗУАЛЬНИХ ОБ'ЄКТІВ

1.1 Проблематика виявлення об'єктів на зображеннях

Людський зір та мозок миттєво розуміють, які об'єкти знаходяться на зображенні та як вони взаємодіють. Зорова система є швидкою та точною.

Проблема автоматичного розпізнавання образів, як для зображень, так і для відеорядів, виникає в різних сферах: у військових справах, системах безпеки, скануванні зображень, маркетингу, рекламі, комп'ютерних іграх. Автоматизація розв'язання цієї задачі залишається складною проблемою, як з теоретичної, так і з технічної точки зору.

Однією із моделей для розпізнавання образів є нейронні мережі [1-8]. Нейронна мережа – це напрям машинного навчання, що є основою для алгоритмів глибокого навчання, та імітує спосіб, яким біологічні нейрони передають сигнали один одному. Нейронні мережі спираються на навчальні дані. Нейронні мережі складаються з вузлів, які мають вхідний та вихідний рівень. Кожен вузол об'єднується з іншим, має поріг та вагу. Вузол – модель лінійної регресії, що складається з даних на вході, максимальних коефіцієнтів, граничного значення і вихідних даних [1-8]. Модель буде виглядати наступним чином:

$$\sum_{i=1}^p v_i d_i + b, \quad (1.1)$$

де v – вага;

m – вхідні дані;

b – зміщення.

Глибоке навчання є ключовою технологією для безпілотних автомобілів, що дозволяє їм розпізнавати знак зупинки або відрізнити пішохода від

ліхтарного стовпа. У глибокому навчанні комп'ютерна модель вчиться виконувати завдання класифікації безпосередньо з зображень, тексту чи звуку. Моделі глибокого навчання можуть досягти високої точності, іноді перевищуючи продуктивність людського рівня. Моделі навчаються за допомогою великого набору позначених даних і архітектур нейронних мереж, які містять багато рівнів.

Як правило, нейронні мережі представлені у вигляді графів, які складаються з вхідного, прихованого та вихідного шару (рис. 1.1).

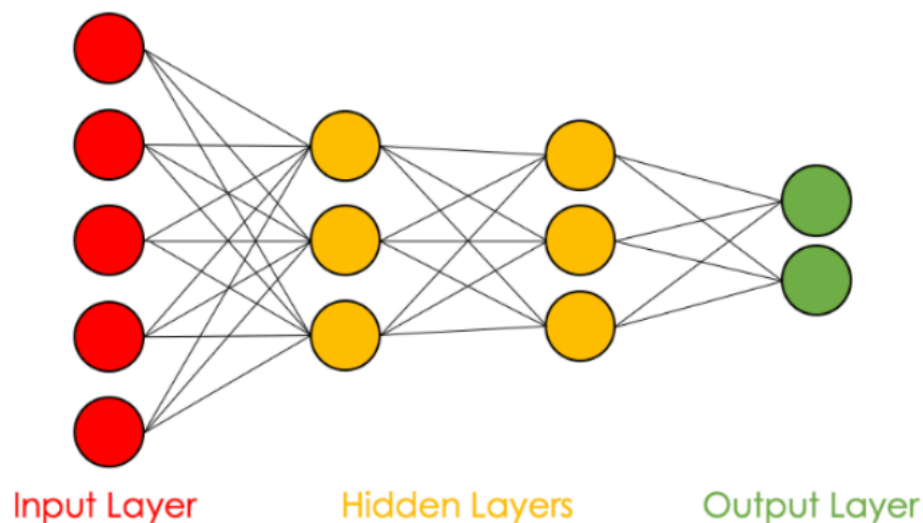


Рисунок 1.1 – Представлення нейронних мереж

Можна сформулювати формулу математичних операцій нейрону як (1.2)

$$y = \sum_i w_i * x_i + b, \quad (1.2)$$

де w – вага;

x – розмір вхідного нейрону;

b – зміщення.

Існує ряд поширених проблем із виявленням об'єктів. Наприклад, відмінність тварини від людини та подібні.

Такі проблеми можуть з'явитися під час виконання процесу розпізнавання об'єктів [1-8]:

- кут зору на об'єкт, наприклад, один об'єкт на зображенні може бути видний зверху, а інший збоку. Це одна з проблем із виявленням об'єктів, оскільки більшість детекторів навчаються із зображеннями лише з певної точки зору;

- деформація. Багато об'єктів, що представляють інтерес, не є твердими тілами і можуть бути деформовані екстремальними способами. Якщо детектор об'єктів навчено виявляти людину, яка включала лише людину, яка сидить, стоїть або йде, він може не виявити інших людей на цих зображеннях, оскільки характеристики на цих зображеннях можуть не збігатися з тими, які він вивчав;

- перекриття частини об'єкта, тобто на зображенні може бути лише частина видимого об'єкта;

- різні умови освітлення. За різних умов освітлення предмети мають різні кольори. Об'єкт може знаходитися при денному або нічному освітленні, що теж впливає на результати детектору;

- злиття фону з об'єктом. Об'єкти на зображеннях можуть зливатися з фоном, що ускладнює їх ідентифікацію;

- широкий обсяг подання класу об'єкту. Деякі об'єкти можуть мати дуже загальний клас, наприклад, будинок може мати багато різних типів цих об'єктів, і кожен матиме свій особливий вигляд;

- використання малих і надвеликих даних. Проблема полягає в тому, як краще використовувати невеликі навчальні дані та як продуктивність алгоритмів розпізнавання можна ефективно масштабувати за допомогою надвеликих даних. Для критично важливих програм, таких як автономне водіння, вартість помилок розпізнавання дуже висока. Настільки величезні набори даних, що містять сотні мільйонів зображень із багатими анотаціями, створені з надією, що точність навчених моделей можна значно підвищити [1-19].

Найвідомішою проблемою комп'ютерного зору є саме розпізнавання об'єктів на статичних зображеннях. Розпізнавання складається з класифікації зображення в одну з багатьох різних категорій. Одним із найпопулярніших наборів даних, які використовуються в академічних колах, є ImageNet, що складається з мільйонів класифікованих зображень. За останні роки окремі моделі комп'ютерної класифікації перевершили можливості людини [11].

Подібно до класифікації, локалізація визначає координати щодо розташування окремого об'єкта всередині зображення.

Локалізацію можна використовувати для багатьох прикладних проблем у реальному житті. Наприклад, інтелектуальне обрізання або навіть регулярне вилучення об'єкта для подальшої обробки за допомогою різних методів. Його можна поєднати з класифікацією, щоб не лише визначити місцезнаходження об'єкта, але й віднести його до однієї з багатьох можливих категорій.

Для виявлення об'єктів треба не лише знаходити об'єкти всередині зображення, але знаходити піксельну маску кожного з виявлених об'єктів. Це є проблемою сегментації екземпляра або об'єкта [11, 12].

При вирішенні проблеми локалізації та класифікації існує потреба виявлення та класифікації кількох об'єктів одночасно. Виявлення об'єктів – це проблема пошуку та класифікації змінної кількості об'єктів на зображенні. Важливою відмінністю є «змінна» частина. На відміну від таких проблем, як класифікація, результат виявлення об'єктів має змінну довжину, оскільки кількість виявлених об'єктів може змінюватися від зображення до зображення.

Щоб виявити об'єкти на зображенні, сучасні системи розпізнавання образів беруть за основу класифікатори та оцінюють їх в різних масштабах та місцях тестового зображення.

Розпізнавання образів відбувається шляхом порівняння істотних ознак, що характеризують певний клас або групу класів. Встановлюється відповідність між обраним на зображенні об'єктом і заданим шаблоном на основі співвідношення кінцевого набору певних властивостей і характеристик.

Існують такі підходи до розпізнавання образів [1, 13-18]:

- розпізнавання набору заздалегідь визначених об'єктів або класів об'єктів на зображенні;
- виявлення об'єкта, що полягає в перевірці зображення або його частини на відповідність певним умовам;
- ідентифікація об'єкта на зображенні з оцінкою його параметрів і прийняттям рішення.

В моделях деформованих частин класифікатор запускається в рівномірно розташованих місцях по всьому зображенню.

Сьогодні популярними методами розпізнавання об'єктів є структурні методи, що базуються на визначенні множини ключових точок та їх описові у вигляді числового чи бінарного вектора – дескриптора [13].

У структурних методах розпізнавання виконується обчислення міри релевантності між описами об'єкта та еталона, образи яких задані у вигляді множин векторів-дескрипторів, та оптимізації значень цієї міри для фіксованої бази еталонів [6].

Структурний підхід до розпізнавання образів дає можливість описувати велику кількість складних об'єктів шляхом використання невеликої кількості непохідних елементів та граматичних правил.

Методи розпізнавання, засновані на відповідності, представляють кожен клас за допомогою паттерну вектора прототипу. Набір шаблонів відомих класів називається навчальним набором. Набір шаблонів невідомих класів називається тестовим набором.

Невідомий шаблон призначається класу, до якого він найближче з точки зору попередньо визначеної метрики. Найпростішим підходом є класифікатор мінімальної відстані, який, як випливає з назви, обчислює (евклідову) відстань (1.3), де p та q є точками, між якими шукається відстань) між невідомим і кожним із векторів-прототипів. Потім обирається найменша відстань для прийняття рішення.

$$\rho(p, q) = \sqrt{(p_1 + q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}. \quad (1.3)$$

Виявлення та сегментація статичних об'єктів – це завдання для обробки відео, що займається ідентифікацією об'єкта і подальшим його відстеженням у часі, в тому числі його наявність, форму та положення.

Метод виявлення статичних об'єктів полягає у детектуванні руху на зображеннях, що досягається за допомогою виокремлення фону з подальшим певним відстеженням.

Розрахунок оптимального порогу за допомогою ітераційного методу шляхом мінімізації втрати пікселя при неправильній класифікації.

Ймовірність піксельного значення визначається формулою:

$$P(X) = P1 * p1(X) + P2 * p2(X) = P1 + P2, \quad (1.4)$$

де $p1(X)$ – фонові пікселі;

$p2(X)$ – пікселі об'єкту.

Віднімання фону – це широко використовувана техніка для сегментації областей переднього плану у відео-послідовностях, знятих зі статичної камери, яка в основному полягає у виявленні рухомих об'єктів за різницею між поточним кадром і фоновією моделлю. Щоб досягти хороших результатів сегментації, фоновіа модель повинна регулярно оновлюватися, щоб адаптуватися до змінних умов освітлення та стаціонарних змін сцени. Тому методів віднімання фону часто недостатньо для виявлення нерухомих об'єктів і тому доповнюється додатковим підходом.

Будь-яка надійна модель віднімання фону повинна бути в змозі обробляти зміни інтенсивності світла, повторювані рухи внаслідок довготривалих змін сцени. Аналіз такого підходу математично можна змоделювати за допомогою функції $P(x, y, t)$ як відеопослідовності, де t – часові

розміри, а x, y – розташування пікселів. Приклад $P(4,5,2)$ – інтенсивність пікселів у $4,5$ – піксельному місці зображення при $t=2$ у відеопослідовності.

Статичні методи розпізнавання об'єктів мають труднощі у виявленні об'єктів в натовпах, наприклад в аеропортах, залізничних станціях через те, що утворюються тіні, які відкидають рухомі об'єкти, що перетворює ініціалізацію об'єкта та відстеження на проблему, яку важко вирішити.

У статистичному розпізнаванні образів шаблон представлений набором d ознак або атрибутів, які розглядаються як d -вимірний вектор ознак. Добре відомі концепції статистичної теорії прийняття рішень використовуються для встановлення меж рішень між класами шаблонів. Ефективність простору представлення (набір функцій) визначається тим, як шаблони ну з різних класів можуть бути відокремлені. Дано набір шаблонів навчання з кожного класу, метою якого є встановити межі рішень у просторі функцій, які окремі візерунки, що належать до різних класів. В підхід теорії статистичного прийняття рішень, межі прийняття рішень визначаються розподілами ймовірностей шаблони, що належать до кожного класу, який повинен бути або визначені, або вивчені.

Існує також підхід до класифікації, заснований на дискримінантному аналізі: спочатку параметрична форма вказує межу рішення (наприклад, лінійну або квадратичну); тоді «найкраща» границя рішення зазначеної форми знаходиться на основі класифікації закономірностей навчання. Такі межі можна побудувати, використовуючи, наприклад, середнє значення критерій квадратичної помилки.

Вектор ознак – X , а мітки класів Y , правило Байєса говорить, що якщо у вас є спільний розподіл X і Y , і якщо X дано, при втратах $0-1$, оптимальним рішенням щодо Y є вибір класу з максимальною ймовірністю за X .

Дискримінантний аналіз належить до гілки методів класифікації, що називається генеративним моделюванням, де необхідно оцінити щільність X усередині класу за міткою класу. У поєднанні з попередньою ймовірністю

(безумовною ймовірністю) класів апостеріорну ймовірність Y можна отримати за формулою Байєса.

Ймовірність або гранична для класу k позначається як

$$\pi_k, \sum_1^K \pi_k = 1. \quad (1.5)$$

Існує набір навчальних даних і підраховується, який відсоток даних надходить з певного класу. Потрібна умовна щільність класу X . Щільність X , зумовлена класом k , або класом $G = k$.

Відповідно до правила Байєса необхідно обчислити апостеріорну ймовірність:

$$\Pr(G = k|X = x) = \frac{f_x(x)\pi_k}{\sum_1^k f_l(x)\pi_l}. \quad (1.6)$$

Щоб створити надійний детектор об'єктів, який може подолати поширені проблеми з виявленням об'єктів, необхідно переконатися, що існує надійна вибірка навчальних даних [14].

Найважливішим пунктом у виявленні у детектуванні об'єктів на зображеннях – це повне розуміння сцени. Невід'ємною частиною розпізнавання є визначення місця розташування об'єктів у сцені, висновок про зв'язки між об'єктами, ієрархії об'єктів від частини до загального розуміння, атрибути об'єктів. Це завдання передбачає не тільки сприйняття сцени, але також потребує когнітивного розуміння фізичного світу. На рисунку 1.2 зображено рівні сегментації об'єктів на зображеннях.

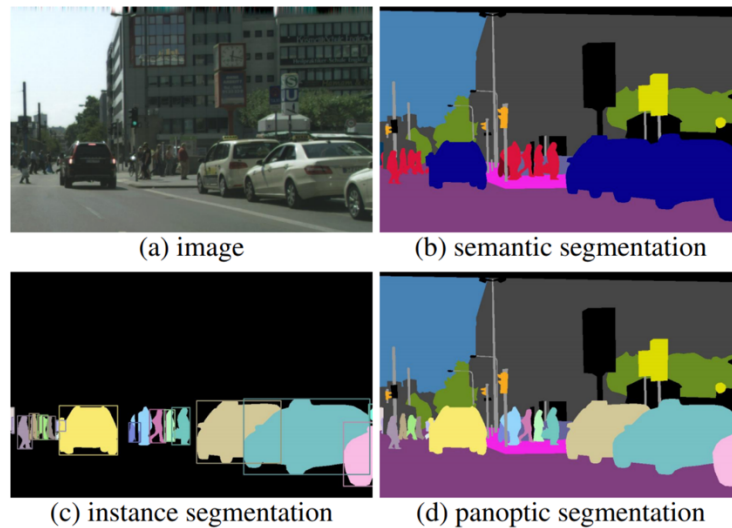


Рисунок 1.2 – Рівні сегментації об'єктів на зображеннях

1.2 Розпізнавання за описом із дескрипторів ключових точок

Структурні методи для розпізнавання об'єктів на статичних зображеннях використовують локальні ознаки зображень, які базуються на описові як множини ключових точок. Ключова точка описується числовим чи бінарним вектором, тобто дескриптором з властивостями вмісту яскравостей зображення для локальних околиць ключових точок [15, 16].

На рисунку 1.3 зображено множину координат ключових точок [17].



Рисунок 1.3 – Множина координат КТ

Простір образів – Ω , в якому $\omega \in \Omega$ – об’єкт на зображенні. Існує певна кількість класів на цьому зображенні $M[J] = \{1, 2, \dots, J\}$, де $1 \dots J$ – номери цих класів.

Простір описів образів описується, як $Z = \{z | z \in R^n\}$, де опис – це скінченна множина дескрипторів КТ. Існує відображення $\Theta : \Omega \rightarrow Z$ із простору Ω у простір Z .

Таким чином, отримуємо ланцюжок перетворень даних із простору образів у множину класів:

$$\Omega \rightarrow \theta(\Omega) \rightarrow Z \rightarrow \varphi(Z) \rightarrow M, \quad (1.7)$$

де $\varphi(Z) \rightarrow M$ – вирішальне правило, яке присвоює номер класу образу.

Сучасний підхід розпізнавання об’єктів за множиною дескрипторів ключових точок полягає в формуванні образів у наступних просторах – просторі сигналів, де формуються образи та у просторі ознак, в якому проходить реалізація прийняття рішення про клас образів, тобто в просторі сигналів задаються класи розпізнаваних об’єктів, а в просторі ознак вони розрізняються [18].

Виявлення ключових точок передбачає одночасне виявлення ознак візуальних сцен і локалізацію їхніх ключових точок. Це просторові розташування або точки на зображенні, які визначають, що є цікавим або що виділяється на зображенні детектором КТ. Дескриптори КТ інваріантні до обертання та масштабування зображення, згортання, перекладу, спотворення тощо.

Для побудови КТ в розпізнаванні образів використовуються наступні детектори – SIFT, SURF, ORB, BRISK та ін.

Метод SURF виглядає наступним чином $z(\omega) = \{z_i | z_i \in R^{64}\}_{i=1}^s$.

Базовий метод класифікації формується наступним чином:

- побудова опису $O \in Z$ для алгоритму розпізнавання, де Z – скінченна множина;
- обчислення значень релевантності $d_j(Z_j, O)$ для опису O на множині Z_1, Z_2, \dots, Z_J описів еталонів;
- оптимізація на множині значень $d_j(Z_j, O)$, $j=1, \dots, J$, і визначення номеру v класу еталона $v = \arg \underset{M}{\text{opt}} d_j(Z_j, O)$ з найкращим значенням релевантності [6].

Таким чином отримаємо класифікаційне правило $\hat{g}: Z \rightarrow M$ з використанням формування центрів класів $\{m_j\}_{j=1}^J$, $m_j \in Z$.

Класифікація $x \in Z$ здійснюється за допомогою конкурентного способу

$$v = \arg \min_{j=1, \dots, J} \rho(x, m_j), \quad (1.8)$$

де v – визначений клас КТ;

$\rho(x, m_j)$ – відстань між дескрипторами КТ.

У результаті чого до фіксованого опису $z(\omega)$ можна отримати трансформований опис як вектор $h[z(\omega)] = (h_1, h_2, \dots, h_J)$.

Даний вектор відповідає розподілу елементів вхідної множини дескрипторів за еталонними класами [19]. Матриця еталонних описів утворюється із набору векторів $h[Z_j]$.

Значення вектору $h[Z_j]$ також оцінює якість класифікації, наприклад, наступна умова $\arg \max_j \{h_j[Z_k]\} = k$ створює ідеальний класифікатор, тобто, при класифікації еталона з номером k максимальне значення у векторі h досягається для компоненти з номером k [20].

Критерій з часткою правильно класифікованих дескрипторів із множини елементів навчальної вибірки показує формула, наведена нижче:

$$\beta = \sum_{k=1}^J \frac{H_k[Z_k]}{s}. \quad (1.9)$$

Сьогодні популярності набули детектори КТ – ORB та BRISK. В основі їхнього підходу полягає обчислення дескриптору КТ у вигляді бінарного вектору з розмірністю, кратною ступеню двійки.

Бінарне представлення значно прискорює процес зіставлення дескрипторів за рахунок можливості застосування двійкових операцій і відповідно спрощує апаратну реалізацію системи розпізнавання. Крім того, бінарна арифметика дає потенцію застосувати ефективний апарат оброблення двійкових даних і синтезувати нові підходи для визначення подібності у просторі дескрипторів при побудові правил класифікації.

ORB та BRISK є не такими вимогливими до обчислювальних ресурсів у порівнянні з іншими, це завдяки спрощеності процедури побудови дескрипторів, завдяки чому існує вигреш у швидкості та точності обробки в порівнянні із SIFT та SURF. Отже, ORB та BRISK забезпечують більшу продуктивність за рахунок спрощення процесу оброблення і побудови дескрипторів бінарного типу [21].

SIFT та SURF дають можливість сформувати множину координат КТ та визначити їх дескриптори. SURF має більшу швидкодію оброблення даних ніж SIFT. Метод ORB визначає масиви більші розміром ОТ у порівнянні з SURF. У детекторі BRISK є можливість управління кількістю сформованих КТ [6].

Необхідно врахувати наскільки описи еталонів розрізняються в рамках застосовуваного методу для високої якості детектування довільних об'єктів.

Вища ймовірність правильного розпізнавання об'єктів забезпечує значніша відмінність елементів описів або суттєвіший їх склад.

Грануляція елементів множинного опису на підставі подібності його складових може забезпечити необхідне стиснення обсягу ознак, ефективну трансформацію простору, а також формування підмножин найбільш

інформативних ознак з метою ефективного зниження обчислювальних витрат [21].

Важливим фактором для забезпечення високої якості ідентифікації або розпізнавання довільних об'єктів є те, наскільки описи еталонів розрізняються в рамках методу. Чим значніша відмінність елементів описів, тим вище ймовірність правильного розпізнавання.

Грануляція елементів множинного опису на підставі подібності його складових може забезпечити необхідне стиснення обсягу ознак, ефективну трансформацію простору, а також формування підмножин найбільш інформативних ознак з метою ефективного зниження обчислювальних витрат.

Метод SURF описується, як $Z \subset R^n$, тобто підмножина $R^n = \{z \mid z \in R^n, \|z\| \approx 1\}$ n -вимірних дійсних векторів, евклідова норма яких дорівнює одиниці. SURF може мати сотні векторів з плаваючою комою. Щоб скоротити кількість векторів, існує стиснений опис на основі відображення, наприклад, формування підмножини значно меншої потужності шляхом застосування процедури відбору ознак із Z [22]. Це також називається формуванням підмножини найбільш значущих або інформативних ознак.

Метрика ρ оцінює подібності елементів, значення якої є критерієм еквівалентності двох дескрипторів, наприклад, використовуються евклідова або манхеттенська метрики із R^n [23, 24].

Два дескриптори вважаються еквівалентними, якщо виконано $\rho(z_1, z_2) \leq \delta_z$ [6], де поріг δ_z – це відсоток від максимально можливого значення, наприклад, 1%, 3%, 5%, 10%.

Існують чіткий і нечіткий підходи гранулювання та представляється, як уявлення опису у вигляді вектора через перехід до мультимножини шляхом класифікації його елементів, а також як угруповання подібних елементів на основі властивостей унікальності або кластерного аналізу.

Інформаційна гранула – підмножина універсуму, на якому визначено відношення подібності (нерозрізненості, еквівалентності), гранула

представляє собою об'єднання атомарних елементів [25]. В результаті опис існує як множина гранул. Міра гранули обчислюється як сума значень функції приналежності елементів:

$$d(A) = \sum_{a \in A} \mu_A(a), \quad (1.10)$$

де $\mu_A(a)$ – значення функції приналежності елемента гранулі;

$$\mu_A(a) \in [0,1].$$

Дії над ними виконуються згідно з законами теорії множин, вони можуть включати одна одну або утворювати ієрархії.

1.3 Нейромережні засоби розпізнавання об'єктів

Виявлення об'єктів можна виконувати за допомогою традиційних методів обробки зображень або впровадженням сучасних мереж глибокого навчання [26, 27].

Виявлення об'єктів є складним процесом і поєднує в собі два завдання, тобто класифікація та локалізація та формує обмежувальну рамку навколо кожного об'єкта на зображенні та призначає їм належність до класу.

Розпізнавання об'єктів відноситься до набору пов'язаних завдань для ідентифікації об'єктів на цифрових зображеннях.

Класифікація зображень – передбачення класу одного об'єкта на зображенні. Локалізація об'єкта визначає розташування одного чи кількох об'єктів на зображенні та формування прямокутника навколо їх розміру. Виявлення об'єктів поєднує ці два завдання та локалізує та класифікує один або декілька об'єктів на зображенні.

Класифікація зображення передбачає тип або клас об'єкта на зображенні.

На вході маємо зображення з одним об'єктом, наприклад зображення, а на виході мітка класу (наприклад, одне або кілька цілих чисел, які зіставляються з мітками класу).

Локалізація об'єкта визначає наявність об'єктів на зображенні та вказує на їх розташування за допомогою обмежувальної рамки. На вході маємо зображення з одним або кількома об'єктами, наприклад фотографія, а на виході одна або кілька обмежувальних рамок (наприклад, визначених точкою, шириною та висотою).

Виявлення об'єктів визначає наявність об'єктів за допомогою обмежувальної рамки та типів або класів розташованих об'єктів на зображенні.

Маємо зображення з одним або кількома об'єктами, наприклад фотографія, а в результаті одна або кілька обмежувальних рамок (наприклад, визначених точкою, шириною та висотою) і мітка класу для кожної обмежувальної рамки.

Традиційні методи обробки зображень, як правило, не вимагають історичних даних для навчання та є неконтрольованими за своєю природою. OpenCV є популярним інструментом для обробки зображень. Позитивні властивості даних методів полягають в тому, що для цих завдань не потрібні анотовані зображення, де люди позначили дані вручну (для навчання під наглядом).

Мінусами є те, що ці методи обмежені такими факторами, як складні сценарії (без одноколірного фону), оклюзія (частково приховані об'єкти), освітлення та тіні та ефект безладу.

Методи глибокого навчання зазвичай суттєво залежать від навчання під наглядом або без нагляду, при цьому методи під наглядом є стандартом у задачах комп'ютерного зору. Продуктивність обмежена обчислювальною потужністю графічних процесорів, яка з кожним роком стрімко зростає [27].

Технології комп'ютерного зору дозволяє виявляти об'єкти складних сцен та із складним освітленням точно.

Мінуси: потрібна величезна кількість навчальних даних; процес анування зображення трудомісткий і дорогий. Наприклад, позначення 500 000 зображень для навчання спеціального алгоритму виявлення об'єктів DL вважається невеликим набором даних. Однак багато контрольних наборів даних (MS COCO, Caltech, KITTI, PASCAL VOC, V5) забезпечують доступність позначених даних.

Сьогодні виявлення об'єктів на зображеннях широко прийнято дослідниками та прийнято компаніями комп'ютерного зору для створення комерційних продуктів.

Для виявлення та відстеження об'єктів став доступний широкий спектр програм комп'ютерного зору. Як наслідок, численні реальні програми, такі як моніторинг охорони здоров'я, автономне водіння, відеоспостереження, виявлення аномалій або комп'ютерний зор використовують глибоке навчання для виявлення об'єктів.

Технологія обробки зображень значно прогресувала за останні роки. Камери менші, дешевші та якісніші, ніж будь-коли раніше. Тим часом обчислювальна потужність різко зросла і стала набагато ефективнішою. В останні роки обчислювальні платформи рухалися до розпаралелювання за допомогою багатоядерної обробки, графічного процесора (GPU) і прискорювачів ШІ, таких як тензорні процесори (TPU).

Таке обладнання дозволяє виконувати комп'ютерне бачення для виявлення та відстеження об'єктів у реалізаціях майже в реальному часі. Отже, швидкий розвиток глибоких згорткових нейронних мереж (CNN) і розширена обчислювальна потужність графічного процесора є основними рушійними силами великого прогресу виявлення об'єктів на основі комп'ютерного зору.

1.4 Постановка задач дослідження

Розпізнавання об'єктів на зображеннях сьогодні є актуальною задачею із впровадженням нейромережових засобів застосовується у всіляких галузях, включаючи маркетинг та масштабні наукові дослідження.

Об'єктом дослідження є виявлення дорожніх знаків та розташування автомобілей за допомогою методу виявлення об'єктів на зображеннях YOLO.

Метою кваліфікаційної роботи є вивчення сучасного нейромережового алгоритму, за допомогою якого можливо детектування візуальних об'єктів на зображенні. Вхідні дані включають в себе зображення з різними класами об'єктів – людини, тварини, автівки та ін.

Основні завдання кваліфікаційної роботи полягають в наступному:

- проаналізувати сучасні нейромережові засоби розпізнавання об'єктів;
- ознайомитися та вивчити архітектуру та принципи роботи методу YOLO;
- розробити мережу для розпізнавання об'єктів на статичних зображеннях;
- провести експериментальне дослідження результативності методу на прикладних зображеннях.

2 АНАЛІЗ ПІДХОДУ YOLO ДЛЯ РОЗПІЗНАВАННЯ ОБ'ЄКТІВ НА ЗОБРАЖЕННЯХ

2.1 Принципи побудови сучасних нейромережових засобів розпізнавання

Згорткова нейронна мережа є алгоритмом, який може отримувати вхідне зображення та призначати його окремим фрагментам вагові значення, що є оцінками їх належності до набору класів для того, щоб мати можливість відрізнити один об'єкт від іншого. Архітектура згорткової нейронної мережі близька до структури підключення нейронів у людському мозку. Окремі нейрони реагують на стимули лише в обмеженій області поля зору, відомої як рецептивне поле. Набір таких полів перекривається, щоб охопити всю візуальну область. Згорткові нейронні мережі здатні за допомогою застосування фільтрів фіксувати просторові та часові залежності в зображенні. Тобто, мережу можна навчити краще розуміти склад зображення.

Розглянемо побудову найбільш популярних нейромережових методів CNN, Fast, Faster R-CNN та YOLO.

R-CNN використовує позиції регіону, щоб спочатку створити потенційні обмежувальні прямокутники на зображенні, а потім запустити класифікатор для цих запропонованих рамок. Після класифікації постобробка використовується для уточнення обмежувальних рамок, усунення дублікатів виявлення та повторної оцінки рамок на основі інших об'єктів у сцені. Ці складні конвеєри повільні та їх важко оптимізувати, оскільки кожен окремий компонент потрібно навчати окремо.

Принцип алгоритму R-CNN виглядає наступним чином (рис. 2.1).

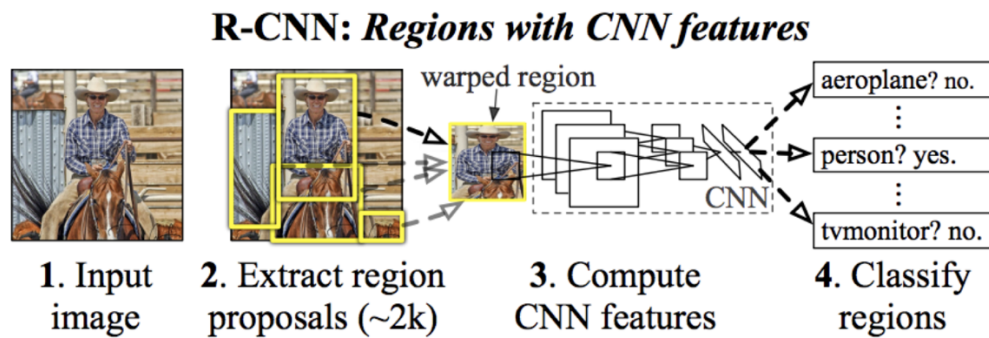


Рисунок 2.1 – Принцип роботи R-CNN

Р. Гіршик запропонував метод, за допомогою якого використовується вибірковий пошук, щоб виділити лише 2000 регіонів із зображення, і він назвав їх пропозиціями регіонів. Тому тепер замість того, щоб намагатися класифікувати величезну кількість регіонів, можна просто працювати з 2000 регіонами. Отже, даний алгоритм працює наступним чином:

- створення першої частини зображення, генерування регіонів-кандидатів;
- використання жадібного алгоритму для рекурсивного об'єднання подібних частин зображення у більші регіони;
- використання створених регіонів для створення остаточних пропозицій регіонів-кандидатів [28].

Розроблено різні підходи до виконання локалізації об'єкта в будь-якій процедурі виявлення об'єкта. Використання ковзних фільтрів різного розміру на зображенні для вилучення об'єкта із зображення може бути одним із підходів, який називається підходом вичерпного пошуку. В залежності від кількості фільтрів, визначаються обчислювальні зусилля для вичерпного пошуку.

Алгоритм вибіркового пошуку використовує вичерпний пошук, але замість того, щоб використовувати його окремо, він також працює з сегментацією кольорів, представлених на зображенні. Отже, вибірковий

пошук – це метод, який відокремлює об’єкти від зображення шляхом надання різних кольорів об’єкту.

Цей алгоритм починається зі створення багатьох маленьких фільтрів і використовує жадібний алгоритм для розширення регіону. Потім він знаходить схожі кольори в регіонах і об’єднує їх разом.

Подібність між регіонами можна розрахувати за допомогою формули:

$$S(a, b) = S_{\text{texture}}(a, b) + S_{\text{size}}(a, b), \quad (2.1)$$

де $S_{\text{texture}}(a, b)$ – це візуальна подібність;

$S_{\text{size}}(a, b)$ – подібність між регіонами.

Використовуючи цей алгоритм, модель продовжує об’єднувати всі регіони разом, щоб збільшити розмір регіонів.

Мережа Fast R-CNN приймає все зображення та набір пропозицій об’єктів як вхідні дані.

Архітектура даної моделі приймає фотографію набору пропозицій регіону як вхідні дані, які передаються через глибоку згорткову нейронну мережу. Попередньо підготовлений CNN використовується для виділення ознак. Кінець глибокого CNN – це спеціальний рівень, який називається «Рівень об’єднання регіонів інтересів» або «Об’єднання ROI», який виділяє функції, специфічні для даного регіону-кандидата вхідних даних.

Мережа спочатку обробляє все зображення з кількома згортковими і максимальними шарами об’єднання, щоб створити карту функцій conv. Потім для кожної пропозиції об’єкта шар об’єднання регіону інтересу (RoI) витягує вектор ознак фіксованої довжини з карти функцій. Кожен вектор ознаки подається в послідовність повністю зв’язаних шарів, які, нарешті, розподіляються на два однорідні вихідні рівні: один, який створює оцінки ймовірності і інший шар, який виводить чотири дійсні числа для кожного класу об’єктів. Вихідні дані CNN інтерпретуються повністю зв’язаним шаром,

після чого модель роздвоюється на два виходи: один для прогнозування класу через шар *softmax*, а інший – з лінійним виходом для обмежувальної рамки.

Потім цей процес повторюється кілька разів для кожної цікавої області на даному зображенні.

На рисунку 2.2 представлена архітектура моделі Fast R-CNN.

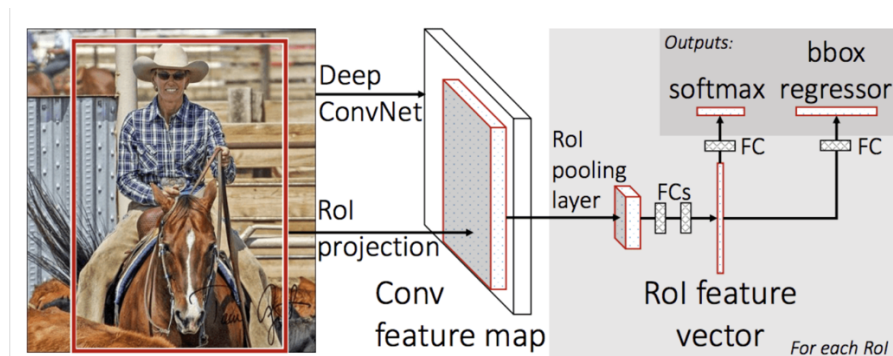


Рисунок 2.2 – Архітектура Fast R-CNN

Дана модель швидше ніж простий R-CNN, навчається та робить прогнози, але все ще вимагає набору регіонів-кандидатів, які пропонуватимуться разом із кожним вхідним зображенням.

У Faster R-CNN замість максимального об'єднання виконується об'єднання ROI для використання однієї карти функцій для всіх регіонів. Це деформує ROI в один шар; рівень об'єднання ROI використовує максимальне об'єднання для перетворення функцій.

У методі Faster R-CNN зображення надається як вхідні дані для згорткової мережі, яка забезпечує карту згорткових функцій. Замість використання алгоритму вибіркового пошуку на карті функцій для визначення пропозицій регіону для прогнозування пропозицій регіону використовується окрема мережа. На рисунку 2.3 можна побачити архітектуру Faster R-CNN.

Faster R-CNN використовує метод пропозиції регіонів для створення наборів регіонів. Він має додатковий CNN для отримання регіональної пропозиції, яка називається регіональною мережею пропозицій. У навчальному регіоні мережа пропозицій приймає карту функцій як вхідні дані

та виводить пропозиції регіону. І ці пропозиції надходять на рівень об'єднання ROI для подальшої процедури [29, 30].

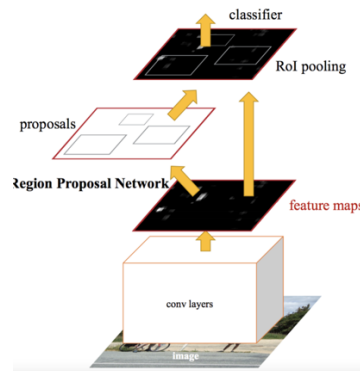


Рисунок 2.3 – Архітектура Faster R-CNN

SSD (Single Shot MultiBox Detector) призначений для обробки в реальному часі. SSD прискорює процес, усуваючи потребу в регіональній мережі. Щоб відновити втрату точності, SSD застосовує кілька вдосконалень, включаючи багатомасштабні функції та стандартні поля.

Метод SSD складається з 2 частин:

- видобування карти функцій;
- застосування фільтрів згортки для виявлення об'єктів.

Кожне передбачення складається з обмежувальної рамки та оцінки кожного класу [31].

Метод SSD робить один знімок, щоб побачити кілька об'єктів усередині зображення, тоді як підходи, засновані в основному на регіональній мережі пропозицій, такі як R-CNN, вимагають 2 знімки, один для створення регіональних пропозицій, інший для роботи з кожною пропозицією. Коли розмір об'єкта невеликий, продуктивність даного методу трохи знижується.

Метод використовує кілька шарів (багатомасштабні карти функцій) для незалежного виявлення, а також з нижчою роздільною здатністю для виявлення об'єктів більшого масштабу.

Для виявлення результативних блоків даних метод використовує градієнтний спуск для оптимізації моделі.

SSD використовує фазу зіставлення під час навчання, щоб узгодити відповідну рамку прив'язки з обмежувальними рамками кожного наземного об'єкта істинності в зображенні. Блок прив'язки з найвищим ступенем перекриття з об'єктом відповідає за прогнозування класу цього об'єкта та його розташування.

SSD ділить зображення за неперсічні блоки, і кожен блок сітки відповідає за виявлення об'єктів у цій області зображення. Якщо жодного об'єкта немає, він розглядається як фоновий клас, а розташування ігнорується. Кожній клітинці сітки в SSD можна призначити кілька прив'язок попередніх блоків. Вони попередньо визначені, і кожна з них відповідає за розмір і форму в клітинці сітки [32, 33].

Метод SSD використовує багатомасштабні карти функцій, тобто декілька шарів зображення, що необхідно для незалежного виявлення об'єктів. Оскільки CNN поступово зменшує просторовий вимір, резолюція карт функцій також зменшується. Наприклад, карти функцій 4×4 та 8×8 використовуються для об'єктів більшого масштабу рисунку 2.4 а) та б).

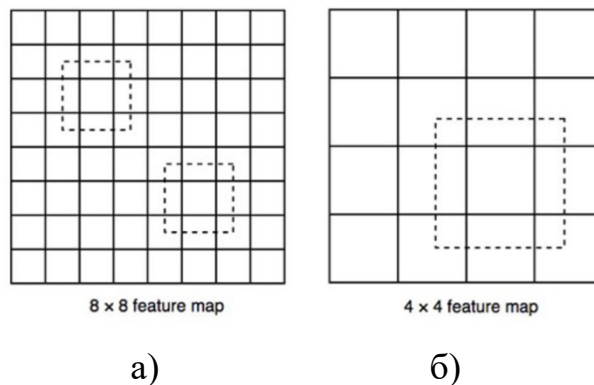


Рисунок 2.4 – Резолюція карт функцій

Необов'язково, щоб результативні блоки мали той самий розмір, що й неперсічні блоки. SSD складається з двох компонентів: опорної моделі та передньої частини SSD. Базовою моделлю є попередньо навчена мережа класифікації зображень як екстрактор ознак. Зазвичай це мережа ResNet,

навчена на ImageNet, з якої було видалено остаточно повністю підключений рівень класифікації [34, 35].

Стандартні межі непересічних результативних блоків рахуються наступним чином:

$$w = S \times \sqrt{\alpha}, h = \frac{S}{\sqrt{\alpha}}, \quad (2.2)$$

де S – масштаб;

α – співвідношення сторін.

Враховуючи велику кількість блоків, створених під час прямого проходу SSD під час результату, важливо відрізати більшу частину обмежувальної рамки, застосовуючи техніку, відому як неадекватне придушення: блоки з порогом втрати достовірності та IoU відкидаються, і зберігаються лише перші N прогнозів. Це гарантує, що мережа зберігає лише найвірогідніші прогнози, тоді як шумові видаляються (рис. 2.5).

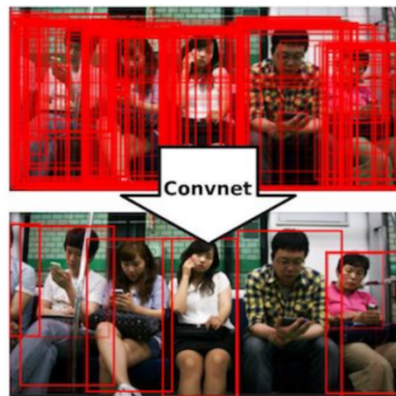


Рисунок 2.5 – Приклад роботи техніки неадекватного придушення

Чим більше існує непересічних блоків за замовчуванням, це призводить до більш точного виявлення. SSD може плутати об'єкти зі схожими категоріями, наприклад, тварини. Треба також виокремити, що SSD забезпечує гіршу продуктивність на менших об'єктах, оскільки вони можуть не відобразитися на всіх картах функцій.

2.2 Метод YOLO

Абревіатура YOLO розшифровується як You Only Look Once, YOLO є згортковою архітектурною нейронною мережею, яка призначена для виявлення об'єктів. Зараз існує 3 версії YOLO – версія 1, версія 2 та версія 3. Останні дві версії є покращеннями першої [36-38]. В алгоритмі YOLOv1 неможливо знайти маленькі об'єкти, якщо вони виглядають у вигляді кластера. Ця архітектура виявила труднощі в узагальненні точок об'єктів, якщо зображення має інші розміри, відмінні від навченого зображення. Основною проблемою є локалізація об'єктів у вхідному зображенні. В YOLOv2 була проведена пакетна нормалізація вхідного рівня, яка зменшує зсув у одиничному значенні в прихованому шарі, і таким чином покращує стабільність нейронної мережі. Також був збільшений розмір вхідних даних, а також введення Anchor Boxes, які виконують класифікацію та прогнозування в одній структурі. Ці блоки прив'язки відповідають за прогнозування обмежувальної рамки, і ці блоки прив'язки розроблені для певного набору даних за допомогою кластеризації.

YOLO розглядає проблему виявлення зображень як проблему регресії, а не як проблему класифікації, і підтримує одну згорткову нейронну мережу.

YOLO виявляє об'єкти безпосередньо від пікселів зображення до координат обмежувальної рамки та ймовірностей класу. Використовуючи даний алгоритм, модель дивиться один раз на зображення, щоб передбачити, що це за об'єкти присутні і де вони знаходяться.

Вищезгаданий алгоритм надзвичайно швидкий. Так як для виявлення кадрів не потрібен складний конвеєр, як для згаданих алгоритмів.

Згортка виконується для всього вхідного зображення лише один раз, щоб отримати прогнози, завдяки чому кодує контекстну інформацію про класи та їх появу.

Під час тестування множимо ймовірності умовного класу та прогнози достовірності окремих boxes, що дає оцінки достовірності для кожного класу.

Ці показники кодують як ймовірність того, що цей клас з'явиться в полі, так і те, наскільки добре прогнозований блок відповідає об'єкту (рис. 2.6).

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

Рисунок 2.6 – Оцінка достовірності для кожного класу

Даний алгоритм робить менше помилок під час прогнозування фонових виправлень об'єктів, оскільки переглядає всі зображення та причини глобально, а не локально [39]. Принцип системи YOLO полягає в тому, що одна згорткова мережа одночасно передбачає кілька обмежувальних прямокутників і ймовірності класів для цих рамок [40]. По-перше, змінюється розмір вхідного зображення, далі запускається згорткова мережа та фінальний пункт визначення порогового значення результуючих виявлень за впевненістю моделі. А саме алгоритм non-maximal suppression на рисунку 2.7. Non-maximal suppression є критичним кроком у виявленні об'єктів на зображенні. NMS – це класичний алгоритм, який аналізує кандидатів на виявлення та зберігає лише ті, що найбільш підходять. Одна згорткова мережа одночасно передбачає кілька обмежувальних прямокутників і ймовірності класу для цих рамок. YOLO тренується на повних зображеннях і безпосередньо оптимізує продуктивність виявлення. Ця уніфікована модель має кілька переваг над традиційними методами виявлення об'єктів.

Алгоритм YOLO працює шляхом поділу зображення на сітки, які мають рівномірну область. Кожна з цих сіток відповідає за виявлення та локалізацію об'єкта, який вона містить.

Відповідно, ці сітки передбачають координати обмежувальної рамки відносно координат їхніх клітинок разом із міткою об'єкта та ймовірністю присутності об'єкта в клітинці.

Цей процес значно зменшує обчислення, оскільки і виявлення, і розпізнавання обробляються клітинками із зображення, але це створює багато

дублікатів прогнозів через те, що кілька клітинок прогнозують той самий об'єкт з різними передбаченнями обмежувальної рамки [41].

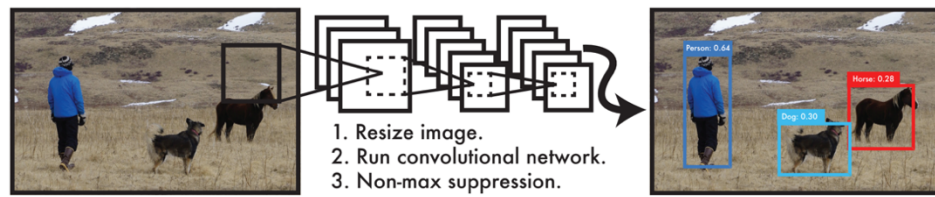


Рисунок 2.7 – Принцип розпізнавання образів за допомогою YOLO

Алгоритм YOLO працює на підґрунті наступних трьох принципів:

- залишкові блоки;
- регресія обмежувальної рамки;
- обчислення критерію Intersection Over Union (IOU).

Спочатку зображення розбивається на різні сітки з розміром $S \times S$, які мають рівномірне розміщення. На рисунку 2.8 показано, як вхідне зображення розділене сіткою.

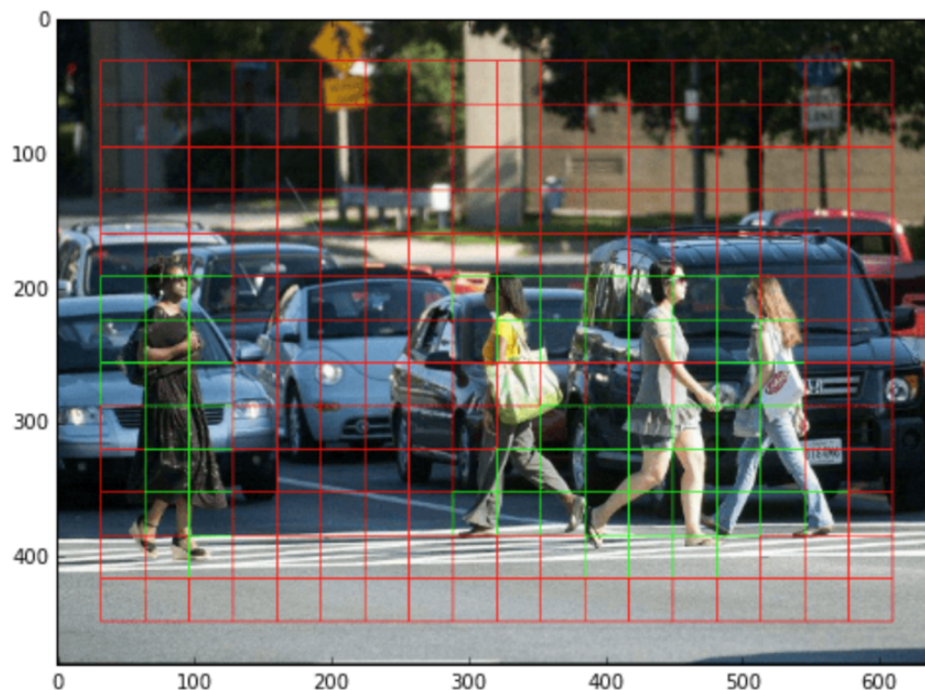


Рисунок 2.8 – Розділення зображення на сітку

На рисунку 2.8 маємо багато клітинок сітки однакового розміру. Кожна клітинка сітки виявляє об'єкти, які з'являються всередині неї. Наприклад, якщо об'єкт з'являється в певній клітинці сітки, то ця клітинка відповідатиме за його виявлення.

Обмежувальна рамка – це прямокутник, який виділяє об'єкт на зображенні.

Кожна обмежувальна рамка на зображенні характеризується наступними атрибутами:

- клас, наприклад, автівка, дорожній знак та ін.;
- висота;
- ширина;
- центр рамки обмежувальної).

На рисунку 2.9 показано приклад обмежувальної рамки. Обмежувальна рамка представлена жовтим контуром.

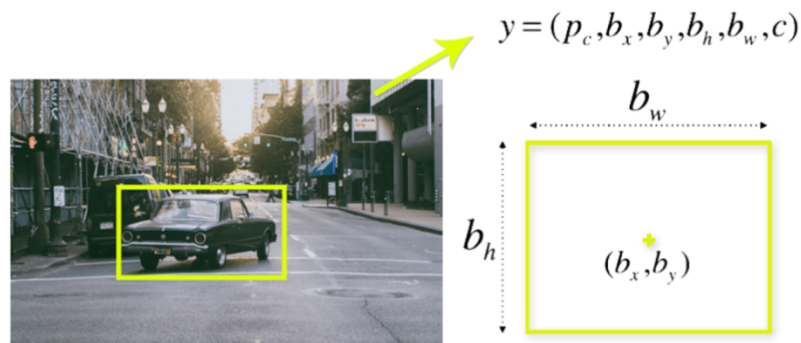


Рисунок 2.9 – Виділення об'єкту обмежувальною рамкою

YOLO використовує єдину регресію обмежувальної рамки для прогнозування висоти, ширини, центру та класу об'єктів. На рисунку 2.9 фіксується атрибут ймовірності появи об'єкта в обмежувальній рамці.

Intersection over union (перетин через об'єднання IOU) – це критерій для виявлення об'єктів, який описує, як блоки перекриваються. YOLO використовує IOU, щоб створити поле виводу, яке ідеально оточує об'єкти.

На рисунку 2.10 наведено простий приклад роботи IOU.

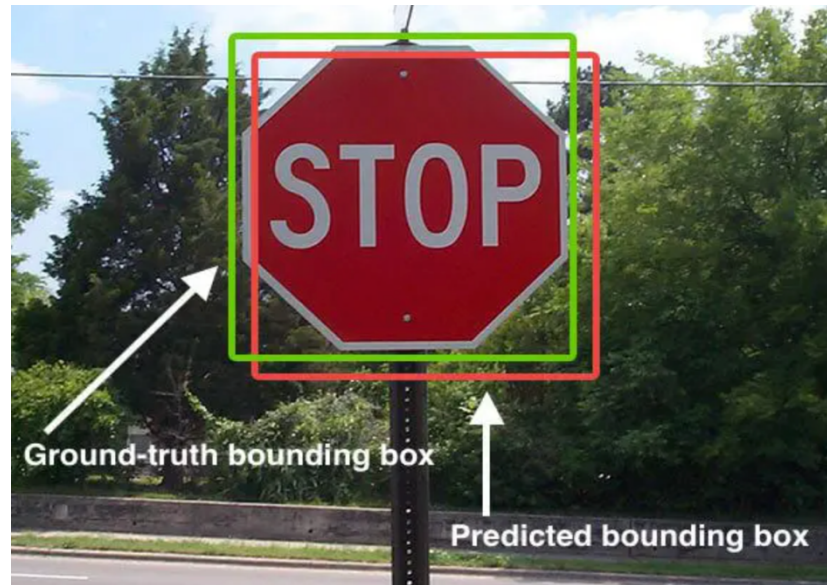


Рисунок 2.10 – Приклад роботи IOU

Кожна клітинка сітки розподілені із врахуванням прогнозу обмежених прямокутників та коефіцієнтів точності. IOU дорівнює 1, якщо передбачувана обмежувальна рамка збігається з реальною рамкою. Цей механізм усуває обмежувальні рамки, які не відповідають реальній рамці.

На рисунку 2.10 є дві обмежувальні рамки, одна зелена, а інша червона. Червоний прямокутник – це прогнозований контур, а зелений – реальний. YOLO гарантує, що дві обмежувальні рамки будуть досить близькі між собою.

IoU розраховується таким чином – площа перекриття, поділена на загальну площу, перетин множин, поділений на їх об'єднання (рис. 2.11).

Значення IoU варіюються від 0 до 1. Тут 0 означає відсутність перекриття, а 1 означає ідеальне перекриття. За допомогою порогового значення для IoU можна визначити, чи є прогноз істинно позитивним (*TP*), хибно-позитивним (*FP*) чи хибно-негативним (*FN*). У прикладі на рисунку 2.12 показано прогнози з пороговим значенням IoU a , встановленим як $a=0,5$. Перше передбачення є справді позитивним, оскільки поріг IoU становить 0,5.

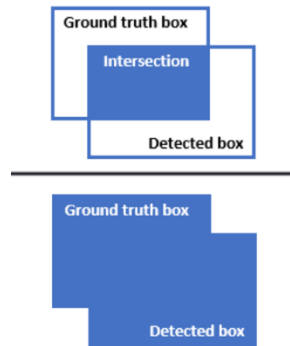


Рисунок 2.11 – Обчислення IoU

Якщо встановити поріг на рівні 0,97, тоді він стане хибно позитивним результатом.

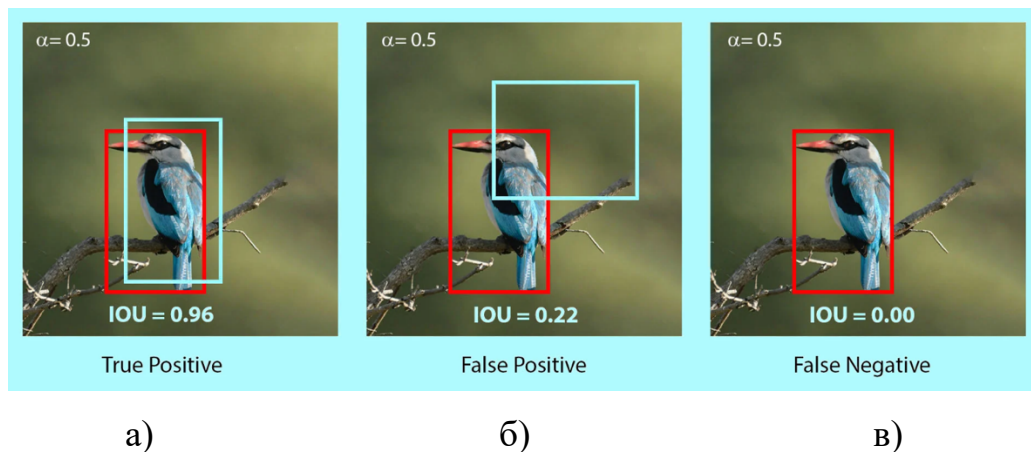


Рисунок 2.12 – Приклади прогнозів IoU

Подібним чином другий прогноз рисунку 2.12 а), б), в) є хибно-позитивним через порогове значення, але може бути істинно-позитивним, якщо ми встановимо поріг зі значенням 0,20.

Теоретично третій прогноз також може бути справді позитивним, враховуючи, що ми знижуємо поріг аж до 0.

IoU у виявленні об'єктів є допоміжною метрикою. Однак у сегментації зображення IoU є основним показником для оцінки точності моделі.

У випадку сегментації зображення область не обов'язково має бути прямокутною. Він може мати будь-яку правильну або неправильну форму. Це означає, що передбачення є масками сегментації, а не обмежувальними

рамками. Тому тут виконується попiксельний аналіз. Крім того, визначення TP , FP і FN децю відрізняється, оскільки воно не базується на попередньо визначеному пороговому значенні.

Рішення щодо виявлення як справжнього позитивного чи хибного позитивного результату повністю залежить від вимог до розроблюваного застосунку.

Кожний прогноз IoU обчислюється відносно кожного базового блоку істинності на зображенні.

IoU потім обмежуються певним значенням (зазвичай від 0,5 до 0,95), а прогнози зіставляються з основними полями правди за допомогою жадібноi стратегії (тобто найвищі IoU збігаються першими).

Потім для кожного класу об'єктів створюється крива точності-запам'ятовування (PR) і обчислюється середня точність (AP). Крива PR враховує продуктивність моделі щодо істинних позитивних, хибно-позитивних і хибно-негативних результатів у діапазоні довірчих значень.

Визначимо істинно позитивний результат:

$$TP = GT \times S, \quad (2.3)$$

де TP – істинно позитивний;

GT – основна істинність;

S – маска сегментації.

Хибно позитивний результат показує, що область знаходиться за межами сегментації зображення і обчислюється наступним чином:

$$FP = (GT + S) - GT. \quad (2.4)$$

Помилково-негативний результат показує кількість пікселів в зоні, яку модель не змогла передбачити і обчислюється наступним чином:

$$FN = (GT + S) - S. \quad (2.5)$$

У задачі виявлення об'єктів IoU – це відношення площі перетину до об'єднаної площі передбачення та базової істинності. Оскільки значення TP , FP і FN є нічим іншим, як площами або кількістю пікселів; ми можемо записати IoU наступним чином (2.6):

$$IoU = \frac{TP}{(TP + FP + FN)}. \quad (2.6)$$

На рисунку 2.13 наведено приклади значень балів для критерію Intersection over Union.



Рисунок 2.13 – Приклад обчислення IoU для різних обмежувальних рамок

Прогнозовані обмежувальні рамки, які значною мірою перекриваються з обмежувальними рамками реальності, мають вищі оцінки, ніж з меншим перекриттям. Це робить IoU результативним показником для оцінки діяльності будь-яких детекторів об'єктів.

Необхідно переконатися, що передбачувані і реальні рамки збігаються якомога точніше – IoU може врахувати це [42].

За допомогою трьох наведених підходів відбувається отримання кінцевого результату розпізнавання об'єктів – визначення рамки і належності до класу.

По-перше, використовуються залишкові блоки, суть яких полягає в розбитті зображення на рівномірні сітки розміром $S \times S$. Кожна з цих N сіток відповідає за виявлення та локалізацію об'єкта, який вона містить. Відповідно, ці сітки передбачають координати обмежувальної рамки відносно координат їхніх клітинок разом із міткою об'єкта та ймовірністю присутності об'єкта в клітинці.

Вважається, що об'єкт знаходиться в певній клітинці, лише якщо координати центру прив'язки знаходяться в цій клітинці. Завдяки цій властивості координати центру завжди обчислюються відносно комірки, тоді як висота та ширина обчислюються відносно розміру всього зображення.

Клас із максимальною ймовірністю вибирається та призначається цій конкретній комірці сітки. Подібний процес відбувається для всіх клітинок сітки на зображенні.

На рисунку 2.14 зображено використання наступних методів розпізнавання об'єктів на зображенні – залишкові блоки, регресія обмежувальної рамки та IOU.

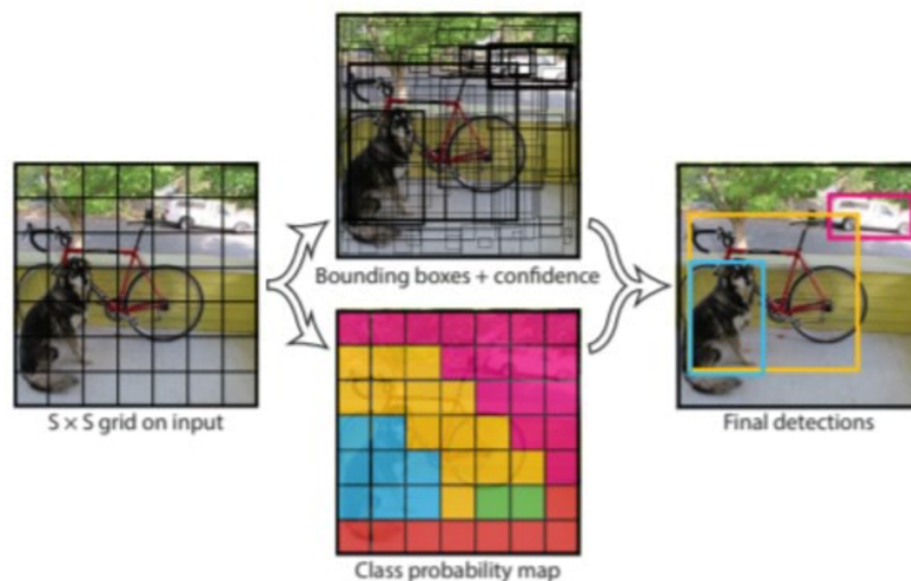


Рисунок 2.14 – Розпізнавання трьох класів об'єктів за допомогою YOLO

Завдяки наявності сітки можна виявляти один об'єкт на клітинку сітки замість одного об'єкта на зображення. Для кожної комірки сітки можливо закодувати вектор, який описуватиме комірку. Наприклад, перша клітинка зліва вгорі не містить жодного об'єкта.

Клітина, в якій знаходиться центр об'єкта, наприклад, центр собаки, є коміркою, в якій відбудеться оцінка достовірності наявності об'єкта в рамках даної області. Оцінка класифікації буде від «0,0» до «1,0», де «0,0» є найнижчим рівнем достовірності, а «1,0» – найвищим; якщо в цій комірці немає жодного об'єкта, оцінка достовірності має бути «0,0», а якщо модель повністю впевнена у своєму прогнозі, оцінка має бути «1,0». Ці рівні достовірності фіксують впевненість моделі в тому, що в цій чарунці існує об'єкт і що обмежувальна рамка точна. Кожна з цих обмежувальних рамок складається з 5 чисел: позиція x , y позиція, ширина, висота та надійність. Координати « (x, y) » представляють розташування центру передбачуваної обмежувальної рамки, а ширина та висота є частками відносно всього розміру зображення. Достовірність представляє IOU між передбачуваним обмежувальним прямокутником і фактичним обмежувальним прямокутником, що називається основним полем істинності. IOU розшифровується як Intersection Over Union і являє собою площу перетину передбачених і базових блоків істинності, поділену на площу об'єднання тих самих передбачених і базових блоків істинності.

До виведення обмежувальних рамок і балів достовірності, кожна клітинка передбачає клас об'єкта. Це передбачення класу представлено довжиною одного гарячого вектора C , кількістю класів у наборі даних. Однак важливо зазначити, що хоча кожна комірка може передбачити будь-яку кількість обмежувальних рамок і показників достовірності для цих рамок, вона передбачає лише один клас. Це обмеження самого алгоритму YOLO, і якщо в одній комірці сітки є кілька об'єктів різних класів, алгоритм не зможе класифікувати обидва правильно. Таким чином, кожне передбачення з чарунки сітки матиме:

$$C + B \times 5, \quad (2.7)$$

де C – кількість класів;

B – кількість передбачених обмежувальних рамок.

$$S \times S \times (C + B \times 5). \quad (2.8)$$

B помножено на 5, оскільки воно включає ($x, y, w, h, \text{доставірність}$) для кожного поля. Оскільки на кожному зображенні є комірки сітки $S \times S$, загальне передбачення моделі є тензором формули:

2.3 Архітектура та організація навчання у YOLO

Оригінальний YOLO був написаний Дж. Редмоном у спеціальному фреймворку під назвою Darknet. Darknet – це дуже гнучкий дослідницький фреймворк, написаний мовами низького рівня, який створив серію детекторів об'єктів у реальному часі в комп'ютерному зорі: YOLO, YOLOv2, YOLOv3 та YOLOv4.

Оригінальний YOLO був першою мережею виявлення об'єктів, яка поєднала проблему малювання обмежувальних прямокутників і ідентифікації міток класу в одній наскрізній диференційованій мережі.

YOLOv2 – зроблено ряд ітераційних удосконалень поверх YOLO, включаючи вищу роздільну здатність і блоки прив'язки.

YOLOv3 – базується на попередніх моделях, додавши оцінку об'єктності до передбачення обмежувальної рамки, додавши з'єднання з рівнями магістральної мережі та зробив прогнози на трьох окремих рівнях деталізації для покращення продуктивності менших об'єктів [43].

YOLOv4 – це двоступеневий детектор із кількома компонентами (рис. 2.15).

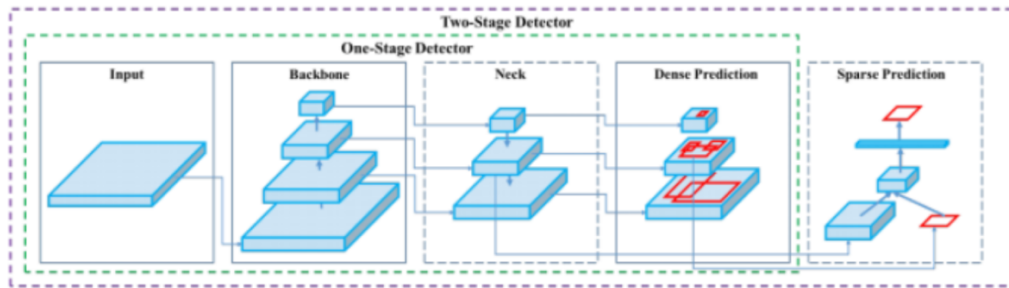


Рисунок 2.15 – Архітектура YOLO

Модель YOLO складається з трьох ключових компонентів: head (голови), neck (шиї) та backbone (хребта) [44].

Backbone – це архітектура глибокого навчання, яка в основному діє як екстрактор функцій та складається зі згорткових шарів. Усі базові моделі є моделями класифікації.

Neck використовує функції шарів згортки в хребті з повністю зв'язаними шарами для прогнозування ймовірностей і координат обмежувальної рамки. Він збирає карти функцій з різних етапів хребта. Отже, це агрегатор функцій.

Head також відомий як детектор об'єктів, він фактично знаходить область, де може бути присутній об'єкт, але не повідомляє, який об'єкт присутній у цій області.

Щоб детектор навчився розпізнавати об'єкти на зображеннях, йому потрібно надати вхідні навчальні дані. Це повинні бути дані з будь-якими об'єктами чи класами – тваринами, людьми або предметами. Наступним кроком є анотування обраних об'єктів, треба позначити цільові об'єкти, малюючи рамки навколо них.

Далі необхідно розпочати саме навчання моделі. Всього в даній системі на сьогодні розроблено 5 моделей. Починаючи від YOLOv5 nano (найменший і найшвидший) до YOLOv5 extra-large (найбільша і наймогутніша модель).

YOLOv5n – це нещодавно представлена наномодель, яка є найменшою в сімействі та призначена для периферійних пристроїв, Інтернету речей, а також із підтримкою OpenCV DNN. Це менше 2,5 МБ у форматі INT8 і близько 4 МБ у форматі FP32. Він підходить для мобільних рішень.

YOLOv5s – це маленька модель у сімействі з приблизно 7,2 мільйонами параметрів, яка ідеально підходить для виконання висновків на ЦП.

YOLOv5m – це модель середнього розміру з 21,2 мільйонами параметрів. Можливо, це найкраще підходить модель для багатьох наборів даних і навчання, оскільки вона забезпечує хороший баланс між швидкістю та точністю.

YOLOv5l – це велика модель сімейства YOLOv5 з 46,5 мільйонами параметрів. Вона підходить для наборів даних, де нам потрібно виявляти менші об'єкти.

YOLOv5x – це найбільша серед п'яти моделей, а також має найвищий mAP серед п'яти. Хоча він повільніший порівняно з іншими та має 86,7 мільйонів параметрів.

На рисунку 2.16 зображено архітектурні шари YOLO.

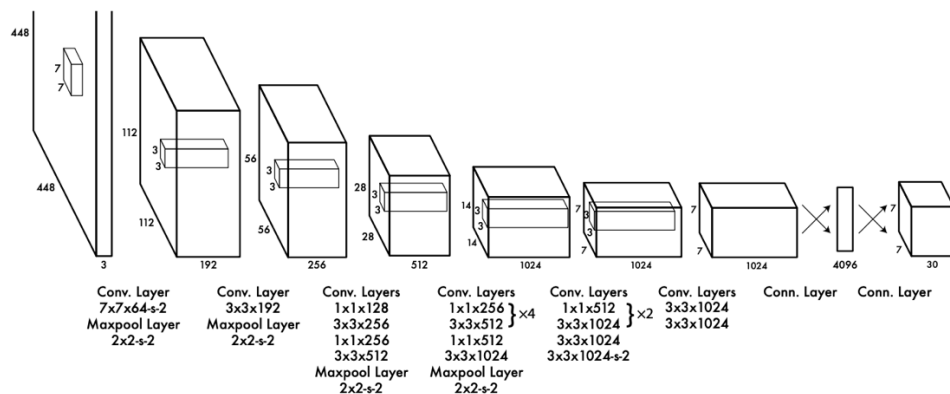


Рисунок 2.16 – Архітектура згорткової мережі

Отже, різниця даних версій залежить від вхідних даних моделей для тренування та вимог. Кожна версія ув'язнена до розміру вхідних даних, при виборі версії необхідно брати до уваги кількість даних, наприклад, YOLOv5n працює з невеликою кількістю вхідних даних, підходить для тренування IoT. YOLOv5s та YOLOv5l використовуються для масштабування моделі в глибину і ширину. Також, всі наведені моделі різняться за швидкістю, чим більше модель, тим швидше модель є.

Тренування та отримання висновків про клас об'єкту – це не весь функціонал, що надає YOLO.

YOLO надає можливість обрати модель з вищезгаданих 5-ох залежно від набору вхідних даних або чи це навчання детектора в режимі реального часу для периферії, чи розгортання найсучаснішої моделі виявлення об'єктів на хмарних графічних процесорах.

Дана модель також дозволяє експортувати навчену модель. Щоб конверсія виявлення об'єктів був повним, недостатньо лише навчання та виведення моделей. У випадках використання в реальному житті розгортання також є основною вимогою. Перед розгортанням нам переважно потрібно конвертувати навчену модель у правильний формат.

Є також функціонал для журналювання, що полегшує перегляд показників продуктивності в будь-який момент після навчання моделі.

YOLOv5 використовує генетичний алгоритм для створення блоків прив'язки, називається цей процес автоматичною прив'язкою, яка повторно обчислює блоки прив'язки відповідно до даних. Це використовується в поєднанні з алгоритмом *k-Means* для створення еволюційних опорних блоків *k-Means*. Це одна з причин, чому YOLOv5 так добре працює навіть на різноманітних наборах даних.

На рисунку 2.17 представлений цикл тренування моделі YOLO.

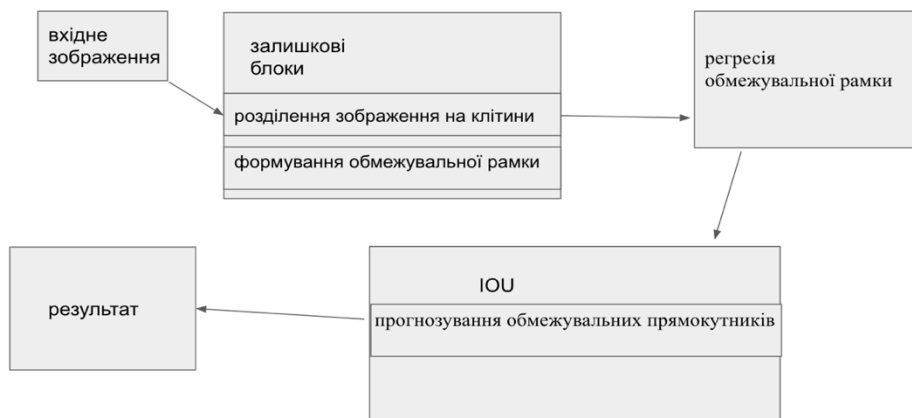


Рисунок 2.17 – Схема тренування моделі YOLO

2.4 Анотація зображень

Одним із найважливіших завдань комп'ютерного зору є маркування даних. Є кілька доступних інструментів, за допомогою яких можна завантажувати зображення, позначати об'єкти за допомогою сегментації для кожного екземпляра. Це допомагає точно локалізувати об'єкт за допомогою обмежувальних рамок або маскування за допомогою багатокутників. Ця інформація зберігається у файлах анотацій.

Основними форматами анотації є COCO та VOC Pascal.

COCO – це великомасштабні зображення із загальними об'єктами в контексті (COCO) для виявлення об'єктів, сегментації та набору даних для підписів. COCO має 1,5 мільйона екземплярів об'єктів для 80 категорій об'єктів

COCO має 5 типів анотацій, які використовуються, а саме для:

- виявлення об'єкта;
- виявлення ключової точки;
- сегментації речей;
- панорамної сегментації;
- підписів до зображень.

COCO зберігає анотації у файлі JSON. На рисунку 2.18 зображено приклад анотації зображення.

```
{  
  "info": info,  
  "licenses": [license],  
  "categories": [category],  
  "images": [image],  
  "annotations": [annotation]  
}
```

Рисунок 2.18 – Приклад анотації в форматі COCO

Основними будівельними блоками для файлу анотації JSON є:

- інформація: містить інформацію високого рівня про набір даних;
- ліцензії: містить список ліцензій на зображення, які застосовуються до зображень у наборі даних;
- категорії: містить список категорій. Категорії можуть належати до суперкатегорії;
- зображення: містить всю інформацію про зображення в наборі даних без обмежувальної рамки чи інформації про сегментацію. ідентифікатори зображень мають бути унікальними;
- анотації: список анотацій кожного окремого об'єкта з кожного зображення в наборі даних.

Pascal VOC надає стандартизовані набори даних зображення для виявлення об'єктів.

Pascal VOC – це файл XML, на відміну від COCO, який має файл JSON.

У середовищі Pascal VOC створюється файл для кожного зображення в наборі даних. У COCO існує по одному файлу для всього набору даних для навчання, тестування та перевірки.

Обмежувальна рамка в форматах даних Pascal VOC і COCO відрізняється.

Обмежувальна рамка COCO: (x-вгорі ліворуч, у-вгорі ліворуч, ширина, висота).

Обмежувальна рамка Pascal VOC: *xmin* – верхній лівий кут, *ymin* – верхній лівий, *xmax* – нижній правий, *ymax* – нижній правий).

На рисунку 2.19 зображено приклад анотації у форматі Pascal VOC.

```

<annotation>
  <folder>Kangaroo</folder>
  <filename>00001.jpg</filename>
  <path>./Kangaroo/stock-12.jpg</path>
  <source>
    <database>Kangaroo</database>
  </source>
  <size>
    <width>450</width>
    <height>319</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  <object>
    <name>kangaroo</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <bndbox>
      <xmin>233</xmin>
      <ymin>89</ymin>
      <xmax>386</xmax>
      <ymax>262</ymax>
    </bndbox>
  </object>
</annotation>

```

Рисунок 2.19 – Приклад анотації у форматі Pascal VOC

YOLO використовує формат Pascal VOC для анотації. Розглянемо вміст анотованого xml файлу:

- *folder* – папка, яка містить зображення;
- *filename* – назва фізичного файлу, який існує в папці;
- *name* – це назва об’єкта, який ми намагаємося ідентифікувати;
- *truncated* – вказує на те, що обмежувальна рамка, вказана для об’єкта, не відповідає повному об’єкту. Наприклад, якщо об’єкт видно на зображенні частково, встановлюється *truncated* рівню 1. Якщо об’єкт повністю видимий, тоді встановлюємо *truncated* рівним 0;
- *difficult* – об’єкт позначається як складний, якщо його важко розпізнати. Якщо об’єкт важко розпізнати, встановлюємо значення *difficult* в 1, інакше встановлюємо значення 0;
- *bndbox* – прямокутник, вирівняний по осі, що визначає розмір об’єкта, видимого на зображенні.

3 РЕЗУЛЬТАТИ ЕКСПЕРИМЕНТАЛЬНОГО ДОСЛІДЖЕННЯ МЕТОДУ YOLO ДЛЯ КЛАСИФІКАЦІЇ

3.1 Вибір програмного середовища

Моделювання виконано в середовищі Visual Studio Code 2022.2 із використанням засобів бібліотек OpenCV та darknet, мова програмування Python.

YOLOv5 інтегровано з Weights & Biases (W&B) для візуалізації в реальному часі та фіксуванні помилок або інформації та слідкування в хмаричне тренування моделі. Це дає змогу краще порівнювати прогони та самоаналіз, а також покращувати видимість і співпрацю для команд [45]. На рисунку 3.1 можна побачити середовище W&B для слідкування тренування моделі.

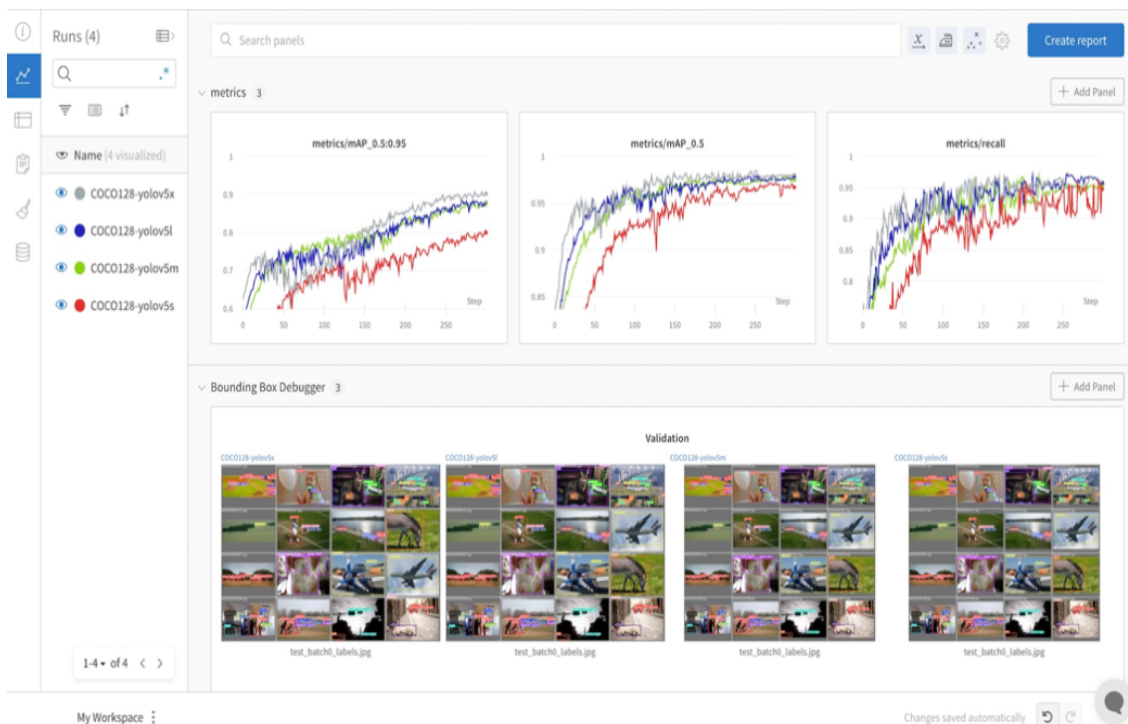


Рисунок 3.1 – Інтеграція W&B

3.2 Результати програмного моделювання

Перед тим, як починати тренування, необхідно обрати модель. Зараз існують наступні версії: v5n, v5s, v5m, v5l, v5x. На рисунку 3.2 порівняння даних моделей. Суть різниці даних версій залежить від розміру моделей. Час виведення моделі та складність моделі зростають від найнижчих значень у YOLOv5s моделі, до YOLOv5m, YOLOv5l і найвищих значень у моделі YOLOv5x. Також, впливають інші фактори, такі як розмір і кількість зображень набору даних, можливості середовища точність.

Оскільки нами використовується невеликий набір даних, то для тренування моделі оберемо версію YOLOv5s, вона є швидкою та добре працює з такими наборами даних.

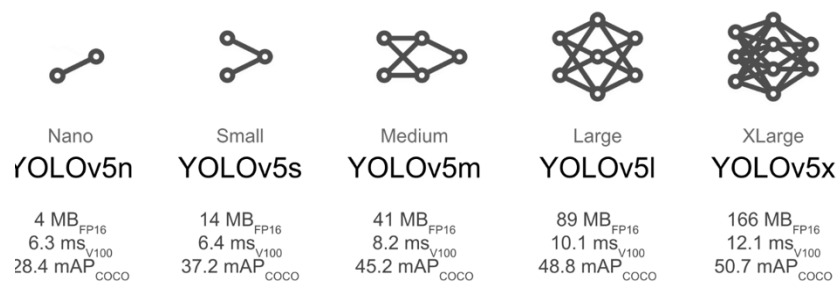


Рисунок 3.2 – Порівняння версій моделі YOLO

Для дослідження буде використовуватися датасет MakeML, який містить дорожні знаки, що належать до 4 класів, наведених на рисунку 3.3:

- світлофор;
- знак стоп;
- обмеження швидкості;
- пішохідний перехід.



Рисунок 3.3 – Приклади зображень з датасету MakeML

Кількість зображень в збірці досягає 877. Типове навчання такого набору займе менше півгодини для даного датасету.

Для експериментального дослідження клонуємо репозиторій YOLO і налаштуємо залежності, які необхідні для запуску YOLO.

Після завантаження репозиторію та даних необхідно перетворити анотації у формат, очікуваний для YOLO, тобто текстовий формат.

Набір MakeML вже мають анотовані зображення та дотримуються формату PASCAL VOC XML. YOLO в свою чергу чекає формат .txt для кожного зображення, де кожен рядок текстового файлу описує обмежувальну рамку.

Лістинг 3.1 приклад анотацій в форматі PASCAL VOC XML:

```

<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>

```

Наведений приклад описує файл, який має розміри 267×400×3. Він містить 3 теги об'єктів, які представляють 3 обмежувальні рамки. Нехай клас визначається тегом *name*, тоді як деталі обмежувальної рамки представлені тегом *bndbox*. Обмежувальна рамка описується координатами її верхнього лівого (*x_min*, *y_min*) кута та нижнього правого (*xmax*, *ymax*) кута.

У форматі маркування YOLO для кожного файлу зображення в одному каталозі створюється файл .txt із однаковою назвою. Кожен файл .txt містить анотації для відповідного файлу зображення, тобто клас об'єкта, координати об'єкта, висоту та ширину.

Лістинг 3.2 формат файлу:

```
<номер класу об'єкта> <x> <y> <ширина> <висота>.
```

Оскільки YOLO очікує файл .txt, то результат буде виглядати, як на рисунку 3.4.

```

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 3.4 – Перетворення анотації рисунка в текстовий файл

Нехай x_center та y_center – ширина і висота відповідно. Координати поля мають бути нормалізовані за розмірами зображення (тобто мати значення від 0 до 1) Кожен рядок представляє один із цих об'єктів. Специфікація для кожної лінії така, що існує один рядок на об'єкт.

Номери класів мають нульовий індекс, тобто починаються з 0.

Отже, перш за все, необхідно переписати з xml файлу координати об'єктів у формат txt, це можливо зробити з використанням онлайн сервісів або написати власну функцію.

Далі розподілимо зображення датасету на набори для навчання, перевірки та тестування. Для кожної категорії створимо окрему папку.

Розглянемо, які параметри для навчання моделі існують:

- `img`: розмір зображення. Зображення квадратне. Розмір оригінального зображення змінюється зі збереженням співвідношення сторін. Довша сторона зображення змінюється до цього числа. Коротша сторона підбита сірим кольором;

- `batch`: розмір кількості зображень;

- `epochs`: кількість епох для тренування;

- `data`: YAML-файл даних, який містить інформацію про набір даних (шлях до зображень, мітки);

- `workers`: кількість обробників CPU;

- `cfg`: архітектура моделі. Розмір і складність цих моделей у порядку зростання, і можливо вибрати модель, яка відповідає складності завдання виявлення об'єктів. Якщо необхідно задати власну архітектуру, то потрібно буде визначити файл YAML у папці `models` із зазначенням архітектури мережі;

- `weights`: попередньо підготовлені ваги, з яких можна почати тренування;

- `name`: різні речі про навчання, наприклад логування тренування. Ваги для тренувань зберігатимуться в папці під назвою `runs/train/name`;

- `hyp`: файл YAML, який описує вибір гіперпараметрів.

Деталі для набору даних, на якому навчається модель, визначаються файлом YAML конфігурації даних.

Нехай наступні параметри мають бути визначені у файлі конфігурації даних:

- `train`, `test` і `val`: розташування зображень тренування, тестування та перевірки;

- `nc`: кількість класів у наборі даних, `name`: імена класів у наборі даних. Індекс класів у цьому списку використовуватиметься як ідентифікатор для імен класів у коді.

Створимо новий файл під назвою `road_sign_data.yaml` і помістимо його в папку `yolov5/data`.

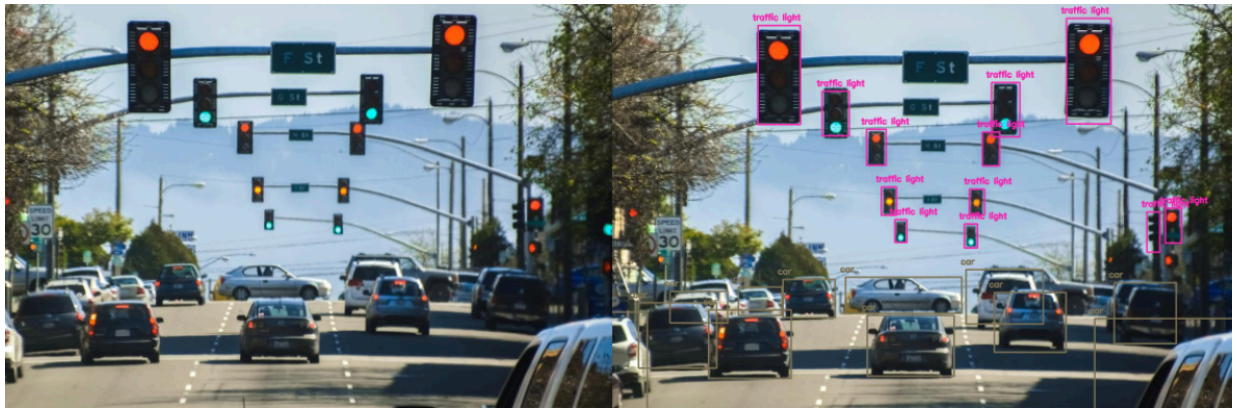
Оскільки набір даних невеликий, і немає великої щільності об'єктів на зображенні, почнемо з найменшої з попередньо підготовлених моделей YOLO. Розмір `batch` задамо 32, а розмір `image` – 640 і тренуємося на 100 епоchs.

Для тренування використовується 24 шари згорткової мережі.

Мережа розпізнавання складається з 24 згорткових рівнів, за якими слідують 2 повністю зв'язані рівні. Чергування 1×1 згорткові шари зменшують простір ознак із попередніх шарів.

Для експерименту на вході підготуємо зображення, яке має наступні параметри: W – ширина, H – висота, S – масштаб. Також маємо C – назви класів, що будуть використані для розпізнавання та додані у результативну рамку. `COLORS` – кольори, які генеруємо для результативної рамки, P – достовірність розпізнавання об'єкту на зображенні.

Окрім дорожніх знаків, дана програма також може розпізнавати авто, на рисунках 3.5, 3.6 а), б), в) зображено вхідні дані та результат виконання програми.



а)

б)

Рисунок 3.5 – Вхідне зображення та результат виконання алгоритму



а)

б)

Рисунок 3.6 – Вхідне зображення та результат виконання алгоритму

Лістинг 3.3 Вхідне зображення, а саме параметри W та H :

```
img = cv2.imread(ar.img)
```

```
w = img.sh[1]
```

```
h = img.sh[0]
```

Наступним кроком буде генерація кольорів та читання назви класів зручних для формування у список.

Лістинг 3.4 Зчитування ваги та файл конфігурації для створення мережі:

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

В архітектурі YOLO є кілька вихідних рівнів, які видають прогнози. Функція `get_output_layers()` надає назви вихідних шарів.

Лістинг 3.5 Вихідний шар не з'єднаний з наступним шаром:

```
def get_output_layers(n):
    names = n.getLayerName()
    result_l = [l_n[i[0] - 1] for i in n.getUnconnectedOutLayer()]
    return out
```

Також створимо функцію для малювання результативного прямокутника з написами класів об'єктів `draw_bounding_box()`.

Далі збираємо прогнози з вихідних рівнів. Для кожного розпізнавання об'єкту з кожного вихідного рівня (`outputs`), отримуємо достовірність виявлення об'єкту (P), ідентифікатор класу (C), параметри обмежувальної рамки, а також ігноруємо достовірність виявлення, якщо $P < 0,5$.

Лістинг 3.6 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(dtY[3] * h)
            xe = cenX - we / 2
            ye = cenY - he / 2
```

```

classId.append(class_id)
confC.append(float(conf))

```

Пройдемо кожне розпізнавання з кожного вихідного рівня та отримаємо ідентифікатор класу, достовірність і параметри обмежувальної рамки.

Наступним необхідним кроком є використання немаксимального придушення.

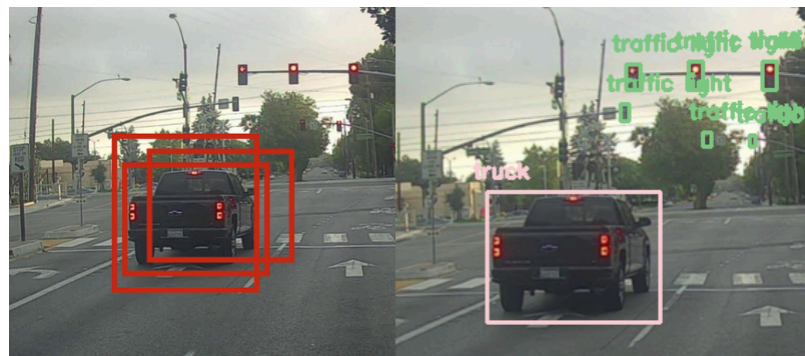
Лістинг 3.7 Виконання немаксимального придушення:

```

inds = cv2.dnn.NMSBoxes(boxes, confidences, confiden_threshold,
nms_threshold)

```

Адже, після пошуку об'єкту на зображенні може бути виявлено багато дублікатів результативної обмежувальної рамки і саме немаксимальне придушення видаляє блоки з великим перекриттям. На рисунку 3.7 а) можна побачити результати розпізнавання до використання немаксимального придушення та на рисунку 3.7 б) після його використання.



а)

б)

Рисунок 3.7 – Результат розпізнавання до використання немаксимального придушення та після

Отже, алгоритм немаксимального придушення прибирає зайві рамки.

Нехай NMS – немаксимальне придушення, B – результативні рамки виявлення об'єкту, S – рівень впевненості, а N – поріг перекриття обмежувальних рамок.

Отже, алгоритм немаксимального придушення полягає в наступному:

- обрати пропозицію з найвищим ступенем достовірності S та додати цей показник до результативного списку D ;
- далі необхідно порівняти результат минулого кроку з усіма пропозиціями, задля цього обчислимо IoU. Якщо IoU перевищує порогове значення N , видалимо цю пропозицію;
- повторюємо даний процес доки не видалимо всі пропозиції B .

Для навчання використовується Darknet Framework, далі перетворюємо модель для виконання виявлення.

Для виявлення об'єктів на зображенні з використанням алгоритму YOLO необхідно створити 3 функції:

- фільтрація неперсічних блоків на основі їх ймовірностей і порогу;
- обчислення IoU між двома блоками;
- розрахування функції Non-max suppression.

Нижче наведені дані функції відповідно.

Лістинг 3.8 Дані функції виявлення об'єктів:

```

check_filtr_box(box_conf, bxs, bx_cls, prbs, thrs = .6):
    bxSc = bxConf*bxClassS
    bxCl = K.argmax(box_scores,-1)
    box_class_scores = K.max(box_scores,-1)
    filterM = box_class_scores>threshold
    sc = t.bool_mask(box_class_scores,filtering_mask)
    bxs = t.bool_mask(bxs,filt_m)
    cls = t.bool_mask(box_classes,filtering_mask)
    return sc, boxes, cls

```

```

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[3] - box1[1]) * (box1[2] - box1[0])
    box2_area = (box2[3] - box2[1]) * (box2[2] - box2[0])
    unArea = box1_area + box2_area - int_ar[0] * int_ar[1]
    iou = int_ar[0] * int_ar[1] / unArea
    return iou

get_suppr(sc, bxs, clss, maximum_bxs = 10, iouvalue = 0.5):
    mxtensor = var(maximum_bxs, typeD='int32')
    K.get_session().run(tf.valn([mxtensor]))
    nm_ind
    tf.image.non_max_suppression(bboxes, scores, max_boxes, iou_threshold)
    scrs = K.gather(scrs, nm_ind)
    bxs = K.gather(bxs, nm_ind)
    clss = K.gather(clss, nm_ind)
    return scrs, bxs, clss

```

Візьмемо за основу рисунок 3.8.



Рисунок 3.8 – Зображення для проведення експерименту

Вхідне зображення має розмір 608×608 .

Кожен розпізнавальний об'єкт буде на виході мати результативну рамку.

YOLO формує результативні прямокутники для всіх об'єктів та класів одночасно [46-50]. Коли об'єкт знаходиться всередині блоку, цей блок відповідає за об'єкт. Формули для розрахунку прогнозованого прямокутника:

$$b_x = \sigma(t_x) + c_x, \quad (3.1)$$

$$b_y = \sigma(t_y) + c_y, \quad (3.2)$$

$$b_w = p_w + e^{t_w}, \quad (3.3)$$

$$b_h = p_h + e^{t_h}. \quad (3.4)$$

В результаті формуються прямокутники, які є кандидатами на присутність об'єктів в них, мережа не повинна передбачати остаточний розмір об'єкта, а повинна лише коригувати розмір найближчого прив'язки до розміру об'єкта. В YOLO ці прямокутники, які мають ширину та висоту – це розміри об'єктів на зображенні, які змінюються під розмір мережі. Чим більше прямокутників-кандидатів використовується, тим вище значення IoU (рис. 3.9).

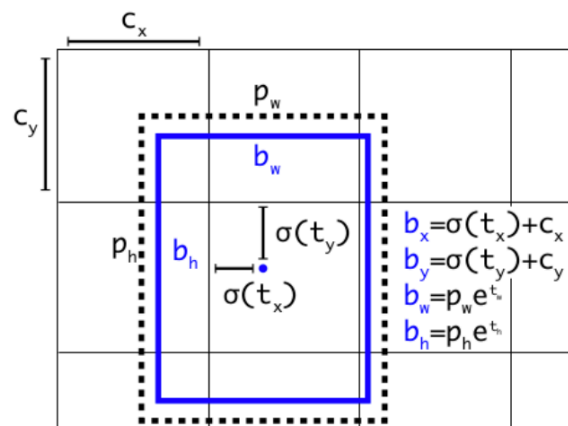


Рисунок 3.9 – Розрахунок обмежувальної рамки

Розглянемо розрахунки співвідношення істинно позитивного результату та шаблонних блоків.

$$r_w = \frac{w_{gt}}{w_{at}}, r_h = \frac{h_{gt}}{h_{at}}, r_w^{max} = \max\left(r_w, \frac{1}{r_w}\right), r_h^{max} = \max\left(r_h, \frac{1}{r_h}\right) \quad (3.5)$$

Таким чином, (рис. 3.10) результативний прямокутник успішно підібраних шаблонів прив'язки призначаються до відповідних клітинок.

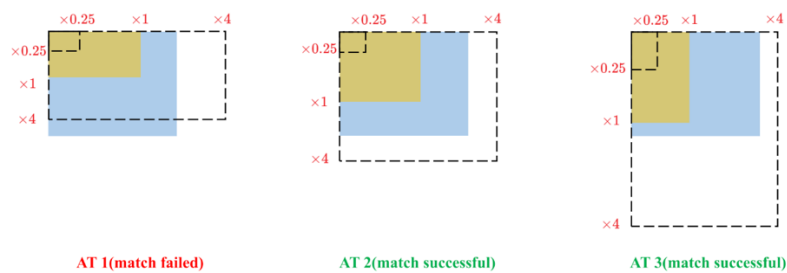


Рисунок 3.10 – Розрахунок істинно позитивного прямокутника

Останній рівень передбачає як ймовірності класу, так і координати обмежувальної рамки. Нормалізуємо обмежувальну рамку ширину та висоту за шириною та висотою зображення так, щоб вони знаходяться між 0 і 1. Параметризуємо обмежувальну рамку – x і координати y повинні бути зсувами певного розташування комірки сітки, тому вони також обмежені між 0 і 1.

Використовуємо функцію лінійної активації для останнього шару, а всі інші шари використовують наступну виправлену лінійну активацію (рис. 3.11):

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

Рисунок 3.11 – Функція лінійної активації

Визначимо та складемо таблицю оцінювання швидкодії та точності різних алгоритмів YOLO. На вході маємо зображення 640×640px. Вимірювання застосовуються з 2.3 GHz Quad-Core Intel Core i7.

Таблиця 3.1 показує результат виявлення різних версій YOLO.

Таблиця 3.1 – Результат детектування версій 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

На діаграмі рисунку 3.12 мета полягає в тому, щоб створити модель детектора об'єктів, яка є продуктивною (вісь x) відносно часу результату (вісь y).

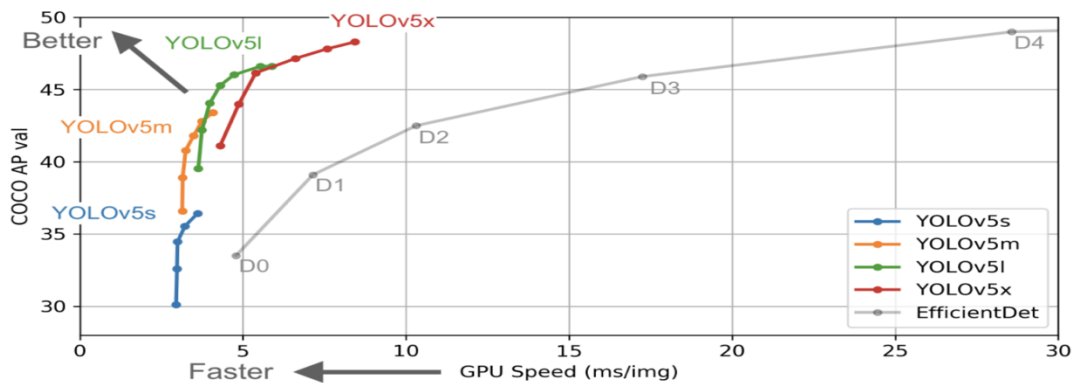


Рисунок 3.12 – Продуктивність моделей YOLO

Попередні результати показують, що YOLOv5 справляється з цією метою краще ніж попередні методи YOLO.

Розглянемо точність розпізнавання частково прикритих об'єктів на зображеннях таблиці 3.2.

Таблиця 3.2 – Точність розпізнавання об'єктів

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

Найточніша модель YOLOv5, YOLOv5x, може обробляти зображення в рази швидше з аналогічним ступенем точності, ніж інші моделі.

Для підвищення продуктивності моделі пропонуються наступні кроки:

- використання більших моделей від YOLOv5 підвищить продуктивність моделей навчання передачі;
- налаштування вхідних параметрів для тренування моделей залежно від вхідних даних.

Основним покращенням версії YOLOv5 на відміну від старих моделей є мозаїчна аугментація даних та автоматична прив'язка обмежувальної рамки.

Мозаїчна аугментація працює подібно наступним чином: чотири вихідні зображення об'єднуються в одне. Це дозволяє досягти кількох методів аугментації за один крок. Створюється чотири випадкові вирізки зображення, зберігаючи відносний масштаб об'єктів, цей метод покращує випадки, коли об'єкт на зображенні частково прикритий. Він також поєднує комбінації класів, які раніше не могли бути виявлені разом на одному зображенні. Це допомагає покращити точність моделі при незбалансованому набору вхідних даних.

Інша можливість моделі полягає в випадковому налаштуванні відтінку в кольорових зображеннях як етап попередньої обробки.

Отже, версія YOLOv5 працює набагато точніше та швидше ніж її попередники, завдяки впровадженню мозаїчної аугментації. Сьогодні це є найпопулярніша прикладна версія, що широко використовується в різних галузях. В новій версії можна побачити покращення детектування об'єктів на зображеннях, що частково перекриті.

ВИСНОВКИ

В результаті виконання дослідження було розглянуто методи розпізнавання об'єктів на зображеннях, вивчено згорткові нейронні мережі та виявлено напрямки їх результативного використання. Порівняно наступні методи розпізнавання об'єктів: CNN, Fast, Faster R-CNN, SSD та YOLO. Виявлено, що методи R-CNN засновані на багатопрохідному методі, тобто вимагають декілька знімків для регіональних позицій та для роботи пропозицій та використовують багатомасштабні карти функцій на відміну від YOLO та SSD, які використовують згортку лише один раз. Сформовано поради щодо випадків використання методу YOLO, експериментально порівняно версії YOLOv5.

Метод YOLO надзвичайно швидкий та використовується в тих випадках, коли об'єкт можна добре розглянути на зображенні, тобто він не є перекритим іншими об'єктами або не є близько розташованим до інших об'єктів, а зображення є чітким. YOLO надає можливість розпізнавання багатьох об'єктів на зображеннях, нові версії методу покращують точність детектування об'єктів з частковим перекриттям.

Виявлені вузькі місця, пов'язані з використанням та реалізацією методу YOLO, головною з яких є виявлення об'єктів з малим масштабом.

У експериментальній частині було використано алгоритм YOLOv5 для розпізнавання дорожніх знаків із використанням датасету MakeML з попередньо анотованими зображеннями. Вибір даної задачі обумовлюється тим, що метод YOLO найчастіше використовується для виявлення сигналів світлофору та фіксації дорожніх порушень.

Перспективою дослідження є вивчення засобів для пришвидшення реалізації методу та здійснення навчання на довільних наборах даних.

Результати дослідження апробовано у вигляді тези доповідей під час XXXIII Міжнародної науково-практичної конференції «TRENDS IN THE DEVELOPMENT OF SCIENCE IN THE MODERN WORLD» [51].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. You Only Look Once: Unified, Real-Time Object Detection / J. Redmon et al. URL: <https://arxiv.org/pdf/1506.02640.pdf> (дата звернення 09.11.2022).
2. Gandhi R. R-CNN, Fast R-CNN, Faster R-CNN, YOLO – Object Detection Algorithms. Medium. URL: <https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e> (дата звернення 09.11.2022).
3. Introduction to YOLO Algorithm for Object Detection. Engineering Education (EngEd) Program | Section. URL: <https://www.section.io/engineering-education/introduction-to-yolo-algorithm-for-object-detection> (дата звернення 09.11.2022).
4. Гороховатський В.О., Творошенко І.С. Аналіз багатовимірних даних за описом у формі множини компонент: монографія, Харків, ХНУРЕ, 2022, 124 с.
5. Гороховатський В.О., Гадецька С.В. (2020) Статистичне оброблення та аналіз даних у структурних методах класифікації зображень (монографія), Харків, ФОП Панов А.Н., 128 с.
6. R-CNN vs Fast R-CNN vs Faster R-CNN - A Comparative Guide. *Analytics India Magazine*. URL: <https://analyticsindiamag.com/r-cnn-vs-fast-r-cnn-vs-faster-r-cnn-a-comparative-guide/> (дата звернення 09.11.2022).
7. YOLO: Real-Time Object Detection Explained. *V7 - AI Data Platform for Computer Vision*. URL: <https://www.v7labs.com/blog/yolo-object-detection#h4> (дата звернення 09.11.2022).
8. Intersection Over Union IoU in Object Detection Segmentation. LearnOpenCV – OpenCV, PyTorch, Keras, Tensorflow examples and tutorials. URL: <https://learnopencv.com/intersection-over-union-iou-in-object-detection-and-segmentation/> (дата звернення 09.11.2022).

9. Aggarwal A. YOLO Explained. Medium. URL: <https://medium.com/analytics-vidhya/yolo-explained-5b6f4564f31> (дата звернення 09.11.2022).

10. Object Detection: An Overview in the Age of Deep Learning - KDnuggets. KDnuggets. URL: <https://www.kdnuggets.com/2017/09/object-detection-overview-deep-learning.html> (дата звернення 09.11.2022).

11. Rajput V. YOLO v4 explained in full detail. Medium. URL: <https://medium.com/aiguys/yolo-v4-explained-in-full-detail-5200b77aa825>

12. YOLOv5 Documentation. YOLOv5 Documentation. URL: <https://docs.ultralytics.com/> (дата звернення 09.11.2022).

13. Шапиро, Л., Стокман, Д., Богуславский, А. А., & Соколов, С. М. (2013). Компьютерное зрение.

14. Путятин, Е. П. (1990). Обработка изображений в робототехнике. Москва.

15. Wallraven, C., Caputo, B., & Graf, A. B. (2003). Recognition with Local Features: the Kernel Recipe. In ICCV (Vol. 3, pp. 257-264).

16. Путятін, Є. П., Гороховатський, В. О., & Матат, О. О. (2006). Методи та алгоритми комп'ютерного зору: навч. посіб. Харків: ТОВ «Компанія СМІТ».

17. Overcome These 6 Problems with Object Detection. *Edge AI and Vision Alliance*. URL: <https://www.edge-ai-vision.com/2022/02/overcome-these-6-problems-with-object-detection> (дата звернення 09.11.2022).

18. Image Recognition: Current Challenges and Emerging Opportunities - Microsoft Research. *Microsoft Research*. URL: <https://www.microsoft.com/en-us/research/lab/microsoft-research-asia/articles/image-recognition-current-challenges-and-emerging-opportunities/> (дата звернення 09.11.2022).

19. Mebsout I. Deep Learning's mathematics. Medium. URL: <https://towardsdatascience.com/deep-learnings-mathematics-f52b3c4d2576> (дата звернення 09.11.2022).

20. Using Background Subtraction Methods in Image Processing. Analytics India Magazine. URL: <https://analyticsindiamag.com/using-background-subtraction-methods-in-image-processing/> (дата звернення 09.11.2022).

21. Tyagi M. Image Segmentation : Part 1. Medium. URL: <https://towardsdatascience.com/image-segmentation-part-1-9f3db1ac1c50> (дата звернення 09.11.2022).

22. Hui J. SSD object detection: Single Shot MultiBox Detector for real-time processing. Medium. URL: <https://jonathan-hui.medium.com/ssd-object-detection-single-shot-multibox-detector-for-real-time-processing-9bd8deac0e06> (дата звернення 09.11.2022).

23. How single-shot detector (SSD) works? | ArcGIS API for Python. ArcGIS Developers. URL: <https://developers.arcgis.com/python/guide/how-ssd-works/#:~:text=SSD%20uses%20a%20matching%20phase,object's%20class%20and%20its%20location> (дата звернення 09.11.2022).

24. Forson E. Understanding SSD MultiBox – Real-Time Object Detection In Deep Learning. Medium. URL: <https://towardsdatascience.com/understanding-ssd-multibox-real-time-object-detection-in-deep-learning-495ef744fab> (дата звернення 09.11.2022).

25. Гороховатський В.О., Гадецька С.В., Стяглик Н.І. (2019) Вивчення статистичних властивостей моделі блочного подання для множини дескрипторів ключових точок зображень. *Радіоелектроніка, інформатика, управління*, No. 2, с. 100–107.

26. Gorokhovatsky, V.O. and Gadetska, S.V. (2019) Determination of Relevance of Visual Object Images by Application of Statistical Analysis of Regarding Fragment Representation of their Descriptions, *Telecommunications and Radio Engineering*, 78 (3), pp. 211–220.

27. 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.

28. Гороховатский В.А. (2003) Распознавание изображений в условиях неполной информации, Харків: ХНУРЭ, 112с.
29. 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.
30. 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.
31. 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.
32. 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.
33. 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.
34. 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.
35. 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.

36. 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.

37. 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.

38. 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.

39. 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.

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

41. 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.

42. 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.

43. 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.

44. 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.

45. Гороховатский В.А., Передрий Е.О. (2009) Корреляционные методы распознавания изображений путем голосования систем фрагментов. *Радиоелектроніка, інформатика, управління*, №1 (20), с.74–81.
46. 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.
47. Gorokhovatskyi O., Gorokhovatskyi V., Peredrii O. (2018) Analysis of Application of Cluster Descriptions in Space of Characteristic Image Features. *Data*, 3(4), 52.
48. 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.
49. 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.
50. Svyrydov, A., Kuchuk, H., Tsiapa, O. (2018) Improving efficiency of image recognition process: Approach and case study, *Proceedings of 2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies, DESSERT*, pp. 593-597.
51. Комаркова, М. (2022). Метод YOLO для детектування об'єктів на зображеннях.