

Міністерство освіти і науки України  
Національний технічний університет  
«Дніпровська політехніка»

Факультет інформаційних технологій  
(факультет)

Кафедра системного аналізу та управління  
(повна назва)

**ПОЯСНЮВАЛЬНА ЗАПИСКА**  
кваліфікаційної роботи ступеня бакалавра

Здобувача вищої освіти \_\_\_\_\_ Копила Єгора Олексійовича  
академічної групи \_\_\_\_\_ 124-22ск-1  
спеціальності / \_\_\_\_\_ 124 Системний аналіз  
за освітньо-професійною програмою \_\_\_\_\_ Системний аналіз

на тему: «Застосування нейронних мереж як систем ідентифікації, класифікації, керування чи прогнозування діяльності складної системи»

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	<i>к.т.н., доц. Владико О.Б.</i>			
розділів:				
Інформаційно-ана літичний	<i>к.т.н., доц. Владико О.Б.</i>			
Спеціальний розділ	<i>к.т.н., доц. Владико О.Б.</i>			
Рецензент	<i>д.т.н., проф. Рудаков Д.С.</i>			
Нормоконтролер	<i>к.ф.-м.н., доц. Хом'як Т.В.</i>			

Дніпро  
2025

ЗАТВЕРДЖЕНО:  
завідувач кафедри  
Системного аналізу та управління  
(повна назва)

\_\_\_\_\_ к.т.н., доц. Желдак Т.А.  
(підпис) (прізвище, ініціали)

« \_\_\_\_\_ » \_\_\_\_\_ 20\_\_ року

**ЗАВДАННЯ**  
на кваліфікаційну роботу  
ступеня бакалавра

здобувачу вищої освіти Копил Є.О. академічної групи 124- 22ск-1  
спеціальності: 124 Системний аналіз  
за освітньо-професійною програмою Системний аналіз  
на тему «Застосування нейронних мереж як систем ідентифікації, класифікації, керування чи прогнозування діяльності складної системи»  
затверджену наказом ректора НТУ «Дніпровська політехніка» №336-с від 05.05.2025 р.

Розділ	Зміст	Терміни виконання
1. Інформаційно-аналітичний розділ	<i>Проаналізувати структуру об'єкта дослідження. Визначити предметну область дослідження та проблему, що розв'язується. Обґрунтувати методи виконання поставлених завдань</i>	10.01.2025 – 01.03.2025
2. Спеціальний розділ	<i>Визначити та обґрунтувати вибір архітектури YOLO. Побудувати та навчити модель. Визначити проблеми та обмеження в реальному застосуванні YOLO. Впровадити у складну систему.</i>	01.03.2025 – 10.06.2025

Завдання видано \_\_\_\_\_ доц. Владико О.Б.  
(підпис) (прізвище, ініціали)

Дата видачі: 06.12.2024 р.

Дата подання до екзаменаційної комісії: \_\_\_\_\_

Прийнято до виконання \_\_\_\_\_ Копил Є.О.  
(підпис студента) (прізвище, ініціали)

## РЕФЕРАТ

Об'єкт дослідження: система ідентифікації та класифікації об'єктів у складних умовах.

Предмет дослідження: застосування нейронних мереж для виявлення об'єктів за допомогою архітектури YOLO.

Мета роботи: розробити, дослідити та проаналізувати ефективність застосування глибоких нейронних мереж для задач ідентифікації об'єктів у складних системах на прикладі YOLO.

Отримані результати та новизна:

Розглядається застосування сучасних методів аналізу зображень з використанням моделей глибокого навчання. Було проведено вивчення дизайну Йоло та його здатності ідентифікувати предмети. Показано фази навчання моделі, складання набору даних, покращення результатів. Поширені проблеми з ШІ в практичних додатках включають: наслідки освітлення, зміна розмірів об'єктів та обмеження обчислювальної потужності. Корисність праці-це здатність змінювати рамки для нагляду, саморегулювання, оборони та ланцюгів поставок.

Ключові слова: штучний інтелект, глибоке навчання, комп'ютерний зір, YOLO, нейронні мережі, детекція об'єктів.

## ABSTRACT

Object of study: identification and classification systems under complex conditions.

Subject of study: application of neural networks for object detection using the YOLO architecture.

Purpose of the work: to design, research and evaluate the effectiveness of deep neural networks in object recognition tasks within complex systems, using YOLO as a case study.

Main results and novelty:

The research explores the practical use of modern computer vision methods based on deep learning. The structure and working principles of the YOLO architecture were studied as a tool for real-time object detection. The stages of dataset creation, model training, and result optimization are presented. Challenges such as lighting variations, object scaling, and limited computing power are discussed. The practical value of the work lies in the potential application of the model for monitoring, automated control, logistics and military systems.

Keywords: artificial intelligence, deep learning, computer vision, YOLO, neural networks, object detection.

## ЗМІСТ

ВСТУП.....	6
1 ІНФОРМАЦІЙНО-АНАЛІТИЧНИЙ РОЗДІЛ.....	8
1.1 Основні поняття нейронних мереж та їх класифікація.....	8
1.2 Стан і тенденції розвитку систем на основі ШІ у складних системах.....	11
1.3 Галузі застосування систем ідентифікації, класифікації та прогнозування... 13	
1.4 Особливості обробки зображень у системах комп'ютерного зору.....	14
1.5 Порівняння класичних методів комп'ютерного зору з підходами на основі ШІ.....	16
1.6 Висновки до інформаційно-аналітичного розділу.....	18
2 СПЕЦІАЛЬНИЙ РОЗДІЛ.....	20
2.1 YOLO як система ідентифікації: принцип роботи та архітектура.....	20
2.3 Побудова та навчання моделі: дані, гіперпараметри, середовище.....	25
2.4 Аналіз ефективності роботи моделі.....	34
2.5 Проблеми та обмеження в реальному застосуванні YOLO.....	38
2.6 Оптимізація нейронної мережі для впровадження в реальні системи.....	40
2.7 Впровадження у складну систему: план, ризики, довгостроковий ефект.....	41
ВИСНОВКИ.....	44
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	45
Додаток А.....	47
Додаток Б.....	48

## ВСТУП

У сучасних умовах стрімкого розвитку інформаційних технологій особливої уваги набувають системи штучного інтелекту (ШІ), здатні ефективно обробляти великі обсяги даних, приймати рішення та адаптуватися до складних і динамічних середовищ. Одним із найважливіших напрямів таких технологій є застосування глибоких нейронних мереж, зокрема в задачах ідентифікації, класифікації та прогнозування у складних системах.

Потреба в автоматизованому аналізі зображень і відео значно зростає внаслідок розвитку автономних технологій, військових застосувань, систем моніторингу, безпеки та промислової діагностики. В цьому контексті важливим є використання архітектур, які здатні виявляти об'єкти у реальному часі з високою точністю. Одним із найефективніших підходів, що довів свою практичну цінність, є модель YOLO (You Only Look Once), яка поєднує швидкість обробки з високою якістю розпізнавання.

Метою даної роботи є дослідження можливостей використання нейронних мереж для ідентифікації об'єктів у складних умовах на прикладі моделі YOLO, а також аналіз ефективності її впровадження у прикладні системи.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

- охарактеризувати основні типи нейронних мереж та їх функціональні властивості;
- дослідити архітектуру YOLO та особливості її реалізації;
- зібрати та підготувати дані для навчання моделі;
- оцінити точність та швидкодію нейромережевої системи;
- виявити основні обмеження й шляхи оптимізації;
- сформулювати рекомендації щодо практичного впровадження системи в реальні умови.

Об'єктом дослідження є процес ідентифікації та класифікації об'єктів у складних системах.

Предмет дослідження — використання глибоких нейронних мереж, зокрема YOLO, для автоматичного розпізнавання об'єктів у візуальних даних.

У роботі використовуються методи системного аналізу, теорії нейронних мереж, машинного навчання, обробки зображень, а також експериментальні методи — навчання та тестування моделей, оцінка ефективності з використанням метрик точності та продуктивності (mAP, FPS, precision, recall).

Кваліфікаційна робота складається зі вступу, двох розділів, висновків, списку використаних джерел та додатків.

У першому розділі розглянуто теоретичні засади нейронних мереж, їх класифікацію та галузі застосування.

Другий розділ присвячено практичному аналізу архітектури YOLO, її навчання, оцінці результатів, виявленню проблем та рекомендаціям щодо впровадження у реальні системи.

# 1 ІНФОРМАЦІЙНО-АНАЛІТИЧНИЙ РОЗДІЛ

Частина кваліфікації, яка займається даними та аналізом, зосереджена на вивченні сучасних ідей штучного інтелекту, особливо з використанням нейронних мереж, для таких питань, як розпізнавання, сортування та прогнозування. Основні принципи побудови штучних моделей мозку, сортування їх дизайну та методи отримання знань.

У цій частині організація інформації щодо функції нейронних мереж у різних складних системах - для виробництва, транспорту, охорони здоров'я, оборони тощо. Особлива увага приділяється вивченню методів комп'ютерного зору, які служать основою для побудови інтелектуальних систем розпізнавання.

Мета цієї частини-встановити основоположну теорію для додаткового практичного вивчення ефективності використання YOLO як моделі серед глибоких конволюційних нейронних мереж для викликів ідентифікації об'єктів у режимі реального часу.

## 1.1 Основні поняття нейронних мереж та їх класифікація

Штучні нейронні мережі (ШНМ) є ключовим елементом сучасного штучного інтелекту, особливо в задачах, де необхідна обробка складних, непередбачуваних або візуально насичених даних. Вони імітують принципи роботи нервової системи людини — зокрема, взаємодію нейронів, які передають сигнали та формують рішення на основі накопиченого досвіду.

Нейронна мережа — це програмна або апаратна модель, побудована з великої кількості взаємозв'язаних елементів (нейронів), які обробляють інформацію за аналогією до біологічних нейронів. Основним принципом роботи є навчання на основі прикладів, у процесі якого мережа коригує свої параметри (ваги), щоб підвищити точність передбачення чи класифікації.

Ідея побудови штучних мереж виникла на основі вивчення біологічного мозку. Біологічний нейрон складається з дендритів (які приймають сигнали),

тіла клітини (де відбувається обробка) та аксона (що передає сигнал далі). Аналогічно, штучний нейрон приймає числові вхідні значення, зважує їх, застосовує функцію активації та передає результат далі по мережі.

Від представників класичної інтегрально-диференціальної школи часто можна почути, що результати, які одержують за допомогою НМ, є необґрунтованими і недоказовими. Проте не можна заперечувати і того факту, що значна кількість теоретичних результатів не знайшла свого застосування, а НМ мають практичне значення, підтвержене експериментально в результаті впровадження та використання.

Штучний нейрон (Рисунок 1.1) є спрощеною моделлю біологічного нейрона. Роль дендритів відіграють входи  $x_1, x_2, \dots, x_n$ , синапсів - вагові коефіцієнти  $w_1, w_2, \dots, w_n$  аксона - вихід  $Y$ , перетворення, які відбуваються в тілі біологічного нейрона, - активація (activation)  $A$  та активаційна порогова функція (activity threshold function), значенням якої є  $Y$ .

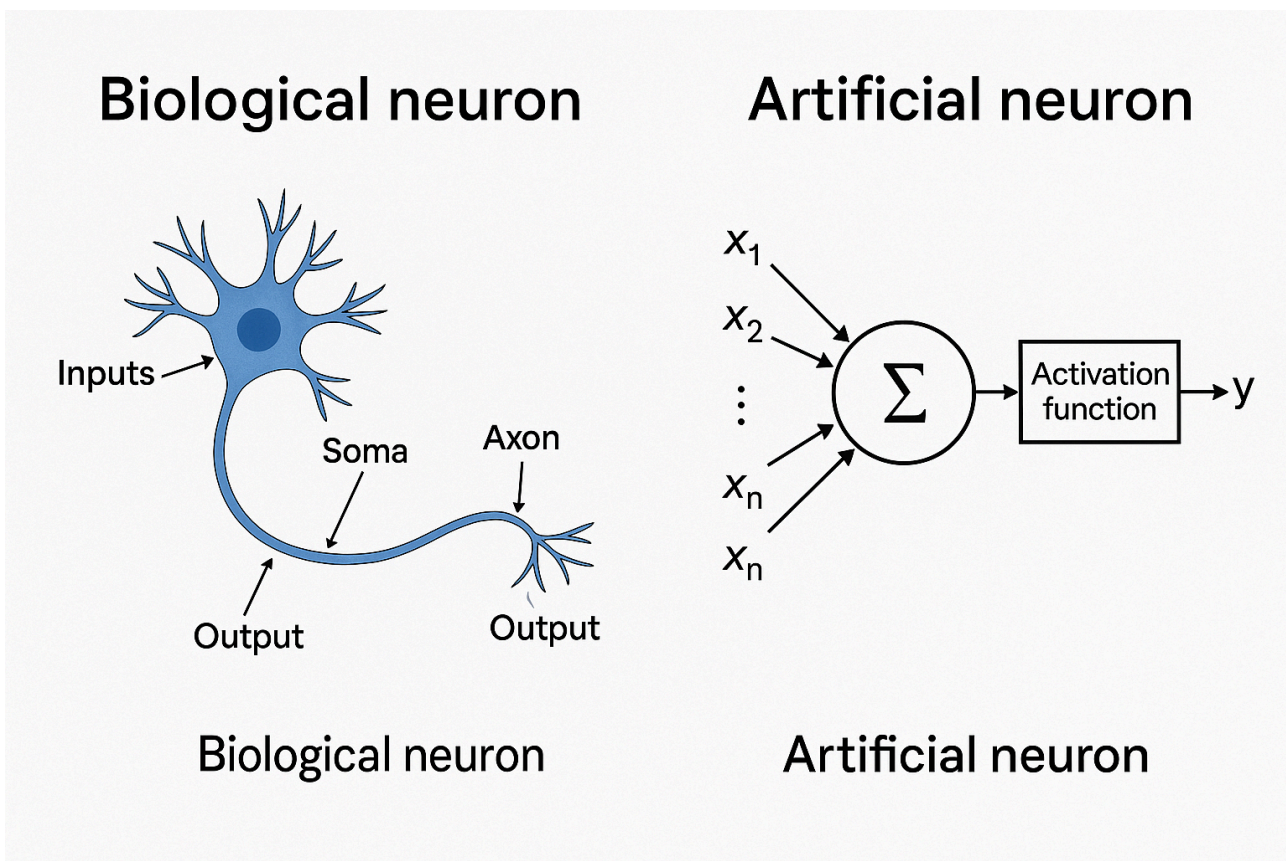


Рисунок 1.1 – Біологічний нейрон та його математична абстракція в ШНМ

Сучасні нейронні мережі класифікуються за архітектурою та призначенням:

- Перцептрон (MLP, багат шаровий перцептрон) — базова форма повнозв'язної мережі, що складається з вхідного, одного або декількох прихованих та вихідного шарів.
- Конволюційна нейронна мережа (CNN) — спеціалізована архітектура для обробки зображень.
- Рекурентна нейронна мережа (RNN) — мережа з «пам'яттю», що враховує попередні входи.
- Генеративно-змагальні мережі (GAN) — генератор і дискримінатор, що взаємодіють для створення нових даних.
- Трансформери (Transformers) — архітектура з механізмом самоуваги для обробки контексту.

Методи навчання нейромереж включають:

- Навчання з учителем (supervised learning) — на основі розмічених даних.
- Навчання без учителя (unsupervised learning) — виявлення структур у нерозмічених даних.
- Навчання з підкріпленням (reinforcement learning) — агент навчається через винагороди або штрафи.

Класифікація нейронних мереж можлива за:

- Топологією — прямого поширення (feedforward) або зворотного зв'язку (recurrent);
- Глибиною — звичайні (до 3–4 шарів) і глибокі (deep learning);
- Функціональністю — дискримінативні, генеративні, комбіновані.

«Нейромережі мають унікальні властивості:

– наслідування принципів роботи людського мозку,

- універсальна апроксимація неперервних функцій,
- здатність до відновлення інформації після часткової втрати,
- паралельна обробка даних»

## 1.2 Стан і тенденції розвитку систем на основі ШІ у складних системах

Упродовж останніх десятиліть спостерігається стрімке зростання ролі штучного інтелекту в різних сферах людської діяльності. Його розвиток нерозривно пов'язаний з еволюцією обчислювальних потужностей, накопиченням великих обсягів даних та вдосконаленням алгоритмів машинного навчання. Особливої уваги набуває застосування ШІ у складних системах, де важлива адаптивність, самонавчання та здатність швидко приймати рішення на основі неповної або динамічної інформації.

Етапи розвитку штучного інтелекту:

- 1950–1970-ті роки — зародження ідей, формалізація логіки і перших програм, таких як General Problem Solver. У 1943 році Маккалок і Пітс описують штучний нейрон. У 1957 році Розенблат розглядає перцептрон - деяке об'єднання штучних нейронів. Неможливість моделювання функції "виключне АБО" (XOR) одношаровим перцептроном строго доведено у 1969 році математиком Марьяном Мінским.
- 1980–1990-ті роки — поява експертних систем, що моделювали вузькі області знань. У 1986 році Руммельхарт, Хінтон і Вільямс пропонують алгоритм зворотного поширення помилки - квінтесенцію теорії нейронних мереж, у тому ж році Дж. Хопфілд, запропонувавши мережі із зворотними зв'язками; здійснив прорив в методах реалізації НМ.
- 2000–2010-ті роки — розвиток статистичних методів і машинного навчання, зокрема SVM, random forest.
- З 2012 року — епоха глибокого навчання, коли нейронні мережі (особливо

CNN) почали домінувати в задачах зору, мови і прогнозування завдяки прориву у змаганні ImageNet.

Визначення складних систем:

Складна система — це система з багатьма взаємозалежними елементами, поведінка якої не завжди може бути точно передбачена на основі окремих частин. До складних систем належать технічні (роботизовані комплекси), виробничі (фабрики, заводи), соціальні (економіка, транспорт) та військові системи. Для таких систем характерні невизначеність, високий ступінь взаємодії між компонентами, необхідність адаптації в реальному часі.

Сучасні приклади впровадження ШІ у складні системи:

- Цифрові двійники (Digital Twins) — віртуальні моделі об'єктів, які прогнозують стан реальних систем.
- Автономні системи — транспорт (автомобілі, дрони), які приймають рішення без участі людини.
- Edge AI — обробка ШІ-даних безпосередньо на пристрої (камери, смартфони).
- Системи підтримки прийняття рішень — прогнозування поломок, аналіз маршрутів, розпізнавання об'єктів.

Основні проблеми та виклики:

- Етичні питання — відповідальність за рішення, прозорість алгоритмів.
- Explainability — складність інтерпретації результатів моделей.
- Нестача якісних навчальних даних — особливо в критичних галузях.
- Високе енергоспоживання — потреба в обчислювальних ресурсах.
- Проблема узагальнення — нестабільність моделі в нових умовах.

### 1.3 Галузі застосування систем ідентифікації, класифікації та прогнозування

Системи, побудовані на основі штучного інтелекту, все частіше використовуються в задачах розпізнавання, аналізу та передбачення подій. Особливо це стосується складних об'єктів, де велика кількість вхідних параметрів і постійна зміна умов ускладнює процес прийняття рішень людиною. У таких випадках системи ідентифікації, класифікації та прогнозування на базі нейронних мереж демонструють високу ефективність, дозволяючи автоматизувати процеси, скоротити час реагування та знизити кількість помилок.

Виробництво та контроль якості:

У промисловості системи комп'ютерного зору з глибоким навчанням використовуються для виявлення дефектів продукції на конвеєрі, контролю цілісності упаковки, класифікації об'єктів за формою, розміром, кольором, моніторингу технічного стану обладнання.

Медицина:

ШІ широко застосовується для автоматичної класифікації медичних знімків (рентген, КТ, МРТ), розпізнавання патологій на ранніх стадіях, прогнозування розвитку захворювань, підтримки прийняття рішень лікарем.

Логістика та ритейл:

У сфері логістики та торгівлі ШІ використовується для автоматичного розпізнавання товарів, аналізу поведінки покупців у магазині, оптимізації маршрутів доставки, прогнозування попиту на продукцію.

Фінансовий сектор:

Нейронні мережі відіграють ключову роль у детекції фінансових шахрайств, оцінці кредитного ризику, прогнозуванні біржових котирувань, аналізі клієнтських транзакцій.

Військова галузь:

У сучасних умовах ведення бойових дій системи ШІ є надзвичайно важливими в обробці відео з БПЛА для виявлення техніки й особового складу, автоматичному супроводі та ідентифікації цілей, класифікації об'єктів за типом (бронетехніка, транспорт, людина), прогнозуванні дій противника.

Роль ШІ у прийнятті рішень у складних об'єктах:

Нейронні мережі є основою інтелектуальних систем, які приймають рішення на основі аналізу багатовимірних та динамічних даних. Це дозволяє виявляти критичні зміни, проводити превентивну діагностику, прогнозувати поведінку системи в майбутньому та адаптуватися до нових умов без ручного втручання.

#### 1.4 Особливості обробки зображень у системах комп'ютерного зору

Системи комп'ютерного зору (Computer Vision, CV) базуються на здатності штучного інтелекту інтерпретувати візуальну інформацію зображень або відео, подібно до того, як це робить людина. Перед тим як зображення буде проаналізоване нейронною мережею, воно проходить кілька важливих етапів обробки. Якість цієї попередньої обробки значною мірою визначає точність подальшої ідентифікації та класифікації об'єктів.

Формати зображень та кольорові простори:

Більшість моделей глибокого навчання працює із зображеннями у форматі RGB, де кожен піксель описується трьома значеннями: червоний, зелений та синій канали. У деяких випадках зображення перетворюються в відтінки сірого (grayscale) для зменшення обчислювального навантаження, особливо якщо кольорова інформація не є критичною. Існують також інші кольорові простори, як-от HSV, YUV, які можуть бути корисними для задач виявлення об'єктів за текстурою або контрастом.

Масштабування та нормалізація:

Щоб подати зображення на вхід нейронної мережі, воно повинно бути приведене до фіксованого розміру, наприклад,  $640 \times 640$  пікселів (у випадку YOLOv8). Масштабування дозволяє уніфікувати розміри, а нормалізація пікселів (наприклад, до діапазону  $[0; 1]$  або  $[-1; 1]$ ) допомагає моделі швидше та стабільніше навчатись.

Аргументація даних:

Аргументація (augmentation) — це процес штучного збільшення обсягу тренувальних даних за допомогою: поворотів зображення, масштабування, дзеркального відображення, розмиття або зміни яскравості, накладення шуму. Це критично важливо для задач, де кількість зображень обмежена. аргументація допомагає зменшити перенавчання моделі та покращити її здатність до узагальнення.

Попередня обробка та сегментація:

У класичних підходах комп'ютерного зору використовувалися алгоритми фільтрації (наприклад, Gaussian Blur), підвищення контрастності або виявлення контурів (Canny edge detection). У сучасних моделях глибокого навчання попередня обробка частково інтегрується всередину архітектури самої мережі, проте якісна підготовка вхідного зображення залишається важливою умовою

високої точності.

Особливості роботи з відеопотоком:

У разі обробки відео, наприклад, з камер спостереження або дронів, важливо враховувати: швидкість обробки кадрів (FPS), синхронізацію між кадрами, стабілізацію картинки, буферизацію даних. В реальних системах, особливо на edge-пристроях, оптимізація попередньої обробки відеопотоку стає критичною для забезпечення детекції в реальному часі.

### 1.5 Порівняння класичних методів комп'ютерного зору з підходами на основі ШІ

Комп'ютерний зір тривалий час розвивався в рамках класичних алгоритмів обробки зображень. Проте із розвитком глибокого навчання акцент змістився на моделі, які самостійно навчаються виявляти та класифікувати об'єкти. У цьому підрозділі проаналізовано ключові відмінності між традиційними методами комп'ютерного зору та сучасними нейронними підходами, такими як YOLO.

Класичні методи комп'ютерного зору:

До класичних підходів належать алгоритми, що працюють на основі жорстко заданих правил та математичних моделей:

- Edge Detection (виявлення контурів): оператори Canny, Sobel, Laplacian;
- Feature Extraction: методи SURF, SIFT, ORB — виявлення унікальних точок на зображенні;
- Template Matching: порівняння зразка з областями на зображенні;
- Haar Cascade Classifiers: набір попередньо навчених правил для виявлення об'єктів (наприклад, облич).

Ці методи зазвичай швидкі, прості в реалізації та не потребують великих обчислювальних ресурсів. Проте їх ефективність обмежена:

- низька стійкість до змін освітлення та ракурсу,

- висока ймовірність хибних спрацювань,
- відсутність здатності до узагальнення.

Методи комп'ютерного зору на основі ШІ:

Нейронні мережі, зокрема Convolutional Neural Networks (CNN), не потребують ручного виділення ознак. Мережа автоматично навчається розпізнавати релевантні особливості зображення під час тренування.

Переваги підходів на основі ШІ:

- висока точність навіть у складних умовах (освітлення, фон, масштаб);
- адаптивність до нових об'єктів за рахунок донавчання;
- універсальність — можна використовувати для багатьох задач: класифікація, сегментація, виявлення.

YOLO (You Only Look Once), як приклад одностадійної моделі, дозволяє не лише виявити об'єкт, а й одночасно визначити його координати на зображенні з великою швидкістю — що особливо важливо для систем реального часу.

**Таблиця 1.1**

**Основні відмінності у підходах:**

Характеристика	Класичні методи	Методи ШІ (CNN, YOLO)
Потреба в розмітці	Ні	Так
Виділення ознак	Ручне	Автоматичне
Стійкість до змін умов	Низька	Висока
Швидкість роботи	Висока	Залежить від ресурсу
Гнучкість і масштабованість	Обмежена	Висока
Якість розпізнавання	Помірна	Висока

Попри те, що класичні методи залишаються актуальними для простих задач або як етап попередньої обробки, підходи на основі ШІ сьогодні є основною технологією в комп'ютерному зорі, особливо в складних або динамічних середовищах. Саме це обґрунтовує доцільність вибору YOLO як базової архітектури для системи ідентифікації, розробленої в межах цієї кваліфікаційної роботи.

Ще одним викликом при побудові систем виявлення об'єктів є варіативність розмірів цільових об'єктів. Наприклад, один і той самий клас об'єкта (автомобіль, людина, дрон) може займати різну кількість пікселів у кадрі залежно від відстані до камери. Класичні алгоритми часто мають труднощі з масштабуванням ознак, тоді як нейронні мережі, особливо архітектури на кшталт YOLO, дозволяють одночасно виявляти об'єкти на різних рівнях масштабу завдяки використанню багаторівневих ознак.

Крім того, наявність шуму та складних фонових сцен також значно ускладнює завдання ідентифікації. У реальних умовах зйомка може здійснюватися при поганому освітленні, у присутності диму, дощу або інших атмосферних перешкод. Наявність об'єктів, візуально схожих на цільові, або хаотичний фон також негативно впливають на точність. У таких ситуаціях класичні методи можуть повністю провалити детекцію, тоді як нейронні моделі за належного тренування здатні частково зберігати стабільність.

## 1.6 Висновки до інформаційно-аналітичного розділу

У цій частині було обговорено основоположні принципи будівельних систем розпізнавати, сортувати та прогнозувати використання нейронних мереж. Очевидно, що системи ШІ ефективні для обробки великої кількості складної, зміни або інформації. Діапазон конструкцій - від основних нейронних мереж до вдосконалених моделей - дозволяє налаштувати підхід до певної проблеми.

Особлива увага приділяється фазам росту AI, його теперішнього стану та таких проблем, як моральні принципи, розуміння моделей та збереження обчислювальної потужності. Промисловості, де ці технології мають найбільш практичну цінність - від промисловості та медицини до військової сфери, де використання ШІ, зокрема в обробці відео з безпілотників, має стратегічне значення.

Обговорюються основні поняття обробки зображень у системах комп'ютерного зору, такі як значення збільшення, нормалізації та підготовки даних. Крім того, досліджуються переваги нейронних мереж порівняно з традиційними методами обробки зображень, які знову підкреслюють важливість використання сучасних дизайнів, таких як Йооло.

Тому основні принципи, встановлені в цій частині, забезпечують надійну основу для реального застосування системи нейронної мережі, розробленої для розпізнавання об'єктів у наступній частині дослідження.

## 2 СПЕЦІАЛЬНИЙ РОЗДІЛ

### 2.1 YOLO як система ідентифікації: принцип роботи та архітектура

Сучасні системи комп'ютерного зору потребують високої точності виявлення об'єктів при збереженні швидкодії — особливо в умовах реального часу. Одним із найбільш ефективних підходів до вирішення цього завдання є використання моделі YOLO (You Only Look Once), яка належить до одностадійних алгоритмів детекції об'єктів.

Основна ідея YOLO:

YOLO — це архітектура, яка здійснює виявлення об'єктів на зображенні в один прохід (one-shot). На відміну від двостадійних підходів (наприклад, R-CNN), які спочатку генерують регіони інтересу, а потім класифікують їх, YOLO виконує детекцію та класифікацію одночасно.

Зображення ділиться на сітку (grid), і кожна клітинка відповідає за передбачення: наявності об'єкта, координат bounding box, та класу.

Розвиток YOLO: від v1 до v11:

- YOLOv1 (2016): перша версія з базовою архітектурою.
- YOLOv2 (2017): покращена локалізація та anchor boxes.
- YOLOv3 (2018): підтримка мультикласової детекції, глибша мережа.
- YOLOv4 (2020): додано багатошарові зв'язки, краще узагальнення.
- YOLOv5 (2020): реалізація від Ultralytics, зручний API, швидкий старт.
- YOLOv7 (2022): оптимізація швидкодії, точності, менший розмір моделі.
- YOLOv8 (2023): повністю нова архітектура, підтримка detect/segment/pose, anchor-free, сучасний API та експорт.
- YOLOv9 (2024): введено концепцію RT-DETR (Real-Time DEtection TRansformer), покращення в attention-механізмах та розділенні шляхів feature fusion.
- YOLOv10 (2024): фокус на оптимізацію inference для edge-пристроїв,

зменшення затримки та розмірів моделей при збереженні точності.

- YOLOv11 (2025): попередні експериментальні релізи з підтримкою 3D-детекції, мультимодального навчання та інтеграції з LLM (Large Language Models).

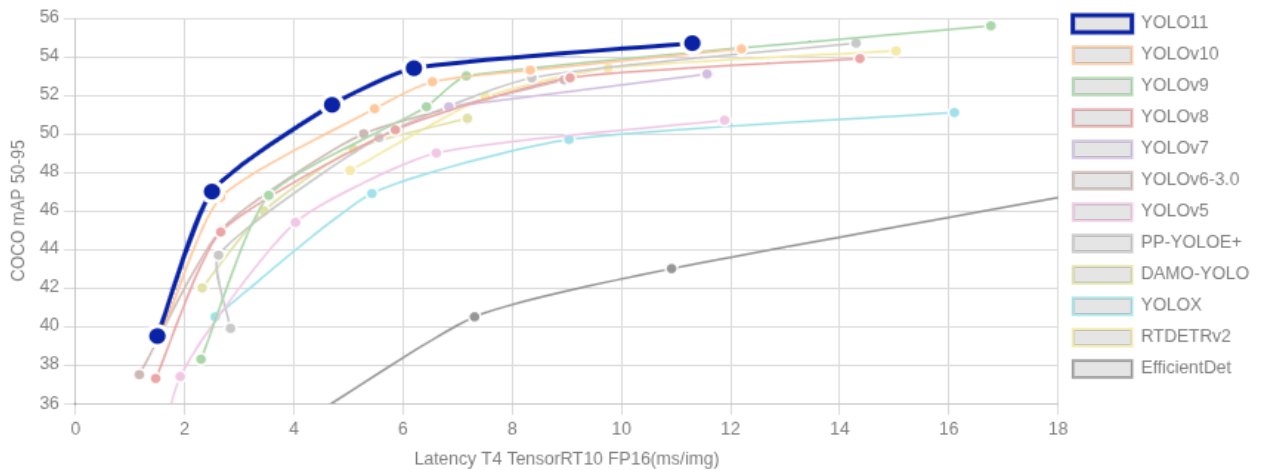


Рисунок 2.1 – Порівняння точності (mAP) та затримки (Latency) для моделей YOLO та альтернативних архітектур

У цій роботі використано YOLOv8, найновішу версію YOLO від Ultralytics. Вона має низку переваг: спрощене тренування, оновлений head, підтримка різних завдань, зручна інтеграція в edge-рішення та хмару, повноцінна CLI/SDK/Hub підтримка.

Архітектурні особливості:

YOLO складається з трьох основних блоків:

1. Backbone — згорткова мережа для витягування ознак (наприклад, CSPDarknet53).
2. Neck — об'єднання ознак на різних рівнях (PANet, FPN).
3. Head — фінальні предикції: координати, класи, ймовірності.

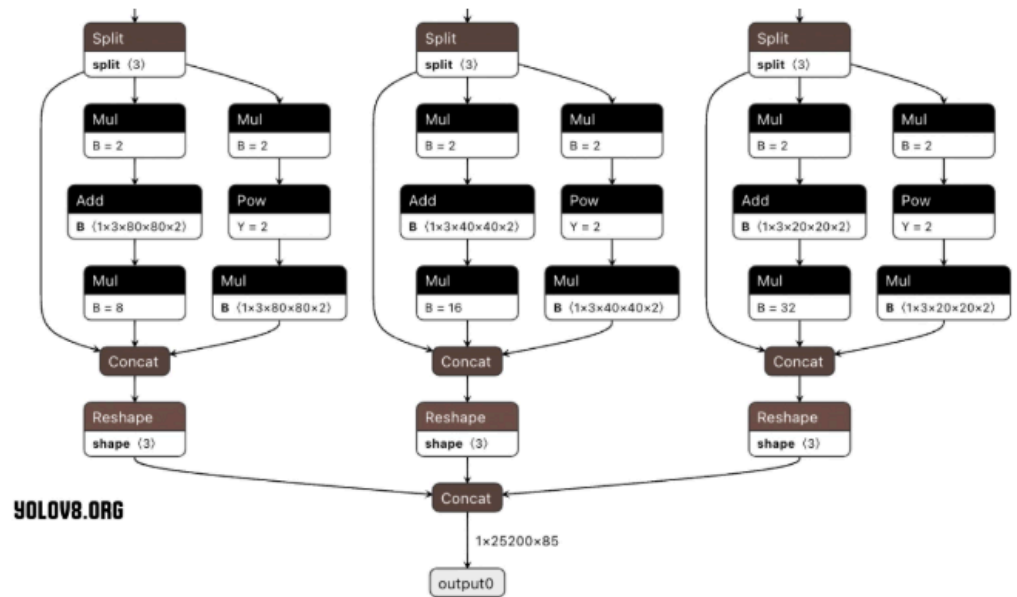


Рисунок 2.2 – Структура вихідного шару (Head) у YOLOv8: перетворення детекцій із кількох масштабів у фінальний вихід моделі

Переваги YOLO:

- Висока швидкість: робота в реальному часі.
- Гнучкість: підтримка кастомних класів.
- Простота інтеграції: open-source, велика спільнота.

Anchor boxes та інтерпретація результатів:

YOLO використовує шаблонні прямокутники (anchor boxes), які дозволяють охопити об'єкти різних форм і розмірів. Для очищення результатів від дублювань застосовується Non-Maximum Suppression (NMS), яка залишає найбільш впевнений результат та видаляє інші з перекриттям понад певний поріг.

Модифікації YOLO під прикладні задачі:

YOLO можна донавчати під конкретні задачі: виявлення техніки, дронів, дефектів, робота в інфрачервоному спектрі тощо. Модель можна експортувати у

формати ONNX або TensorRT для швидкої роботи на edge-пристроях (наприклад, NVIDIA Jetson).

У процесі підготовки даних для тренування було застосовано базові аугментації: масштабування, випадкове віддзеркалення, зміна яскравості. Це дозволило зменшити overfitting та покращити узагальнюючу здатність моделі. Зображення були приведені до розміру  $640 \times 640$  пікселів, що є стандартом для даної архітектури. Усі об'єкти було розподілено по класах із забезпеченням відносного балансу, аби уникнути зміщення результатів.

Параметри навчання включали  $\text{batch size} = 16$ ,  $\text{learning rate} = 0.01$ , кількість epoch = 100. Застосовувалась адаптивна стратегія зміни швидкості навчання (CosineDecay), що дозволило забезпечити стабільну збіжність при зменшенні втрат.

У результаті, YOLO — це не просто алгоритм виявлення об'єктів, а універсальна платформа, яку можна адаптувати до широкого спектра складних систем.

## 2.2 Постановка задачі та обґрунтування вибору архітектури YOLO

У цій підготовці було створено практичну діяльність - для створення механізму миттєвого розпізнавання предметів у картинках чи фільмах.

Система повинна мати можливість функціонувати за обмеженими обставинами, швидко, точно, точно та бути пристосованою для додаткового включення в складні технічні або виробничі системи.

Налаштування проблеми: У сучасній ситуації автоматичне виявлення предметів на знімках та рухомі зображення є важливим у різних галузях: виробництво, доставка, безпека та збройні сили. Основна мета системи полягає в тому, щоб точно та швидко помітити живі об'єкти для подальшого вибору автоматизованими або людськими машинами.

Завдання створення передбачає: створення та організацію колекції із випадками предметів; навчання нейронної мережі; тестування системи в подібних ситуаціях у реальному світі; Вдосконалення моделі для використання.

Обмеження задачі:

- обмежена обчислювальна потужність (Jetson Nano, мобільні GPU);
- нестабільність відеопотоку;
- складне освітлення, погодні умови;
- вимога до невеликої ваги моделі.

Альтернативи та причини вибору YOLO:

Серед альтернатив: Faster R-CNN (точний, але повільний), SSD (швидкий, але менш точний), RetinaNet (баланс, але складніший у тренуванні), EfficientDet (економічний, але гірше працює на edge-пристроях).

YOLO виграє завдяки:

- простоті архітектури та інтеграції;
- високій швидкості роботи;
- потужній підтримці спільноти;
- гнучкості масштабування.

Обраний підхід:

YOLOv8m обрано як оптимальний варіант:

- достатньо глибока архітектура;
- висока продуктивність без надмірних ресурсів;
- підтримка експорту у ONNX, TensorRT;
- стабільна робота в реальних умовах.

Моделі тренувалась на власному сервері і тестувалась на NVIDIA RTX 3090 та Jetson Orin Nano.

Сценарії застосування:

- Виробництво: виявлення дефектів, контроль ТБ;

- Безпека: вторгнення, підозріла активність;
- Військові цілі: дрони, техніка, супутникові знімки;
- Транспорт: підрахунок машин, типізація.

Перспектива розвитку:

- інтеграція з трекінгом (DeepSort, ByteTrack);
- сегментація зони руху;
- автоматичні тривоги;
- поведінкова аналітика.

### 2.3 Побудова та навчання моделі: дані, гіперпараметри, середовище

Ефективність нейронної мережі безпосередньо залежить від якості та обсягу навчальних даних. Для задачі детекції об'єктів на основі YOLO необхідно мати датасет, що містить зображення цільових об'єктів у різних умовах (освітлення, фони, масштаби) та відповідну розмітку у форматі bounding box + label (формат YOLO: `.txt`` файл з координатами та класом для кожного зображення).

У цьому дослідженні було використано готовий датасет автомобілів з класифікацією за типами: Ambulance, Bicycle, Bus, Jeepney, Motorcycle, Multicab, Pickup, SUV, Sedan, Truck, Van. Датасет містить понад 7000 зображень, розмічених вручну та організованих у структуру train/val/test згідно стандарту YOLOv8.

Для розширення варіативності навчального набору можна застосовувати численні техніки аргументації: горизонтальне віддзеркалення, зміна яскравості, контрасту, масштабування, обертання, додавання шуму. Це дозволяє покращити здатність моделі до узагальнення.

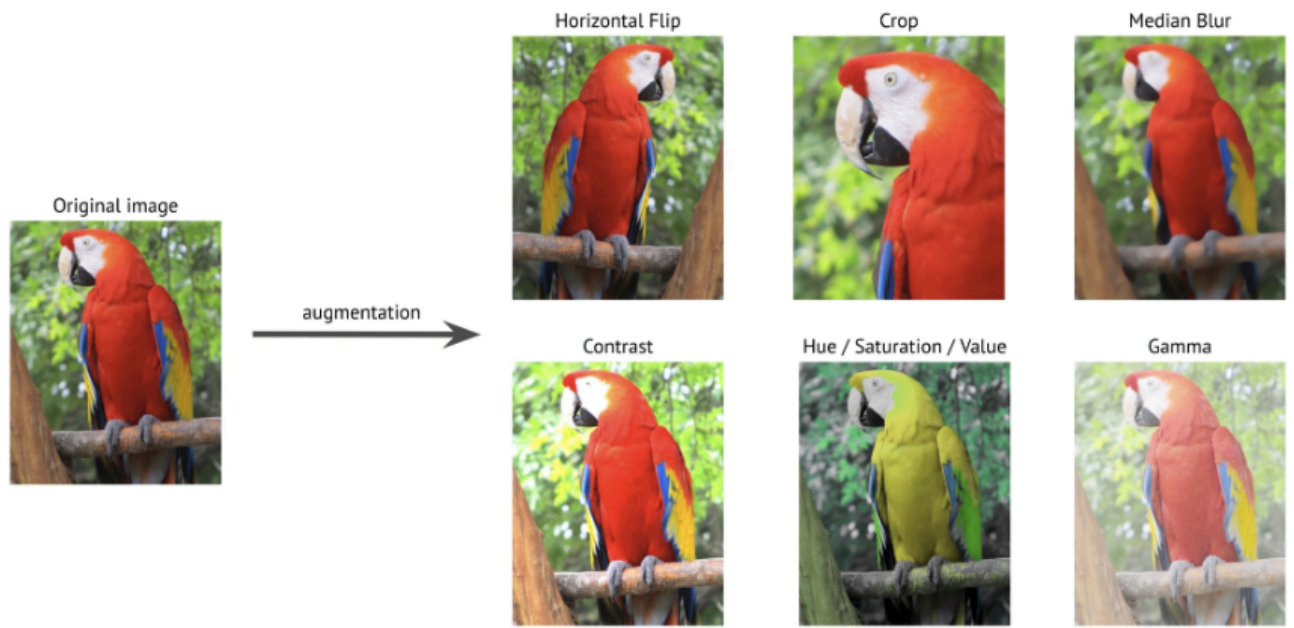


Рисунок 2.3 – Приклади базових аугментацій: віддзеркалення, обрізання, розмиття, зміна контрасту, кольорового тону та гамми

Було враховано потенційні проблеми: дисбаланс класів (деякі об'єкти трапляються частіше), що вирішувалось шляхом використання зважених функцій втрат і *upsampling* рідкісних прикладів.

Після формування датасету здійснюється налаштування гіперпараметрів моделі.

Найважливішими є:

- `batch_size`: 16 ,
- `learning_rate`: 0.001,
- `epochs`: 100,
- `imgsz`: 640×640,
- `optimizer`: SGD або Adam.

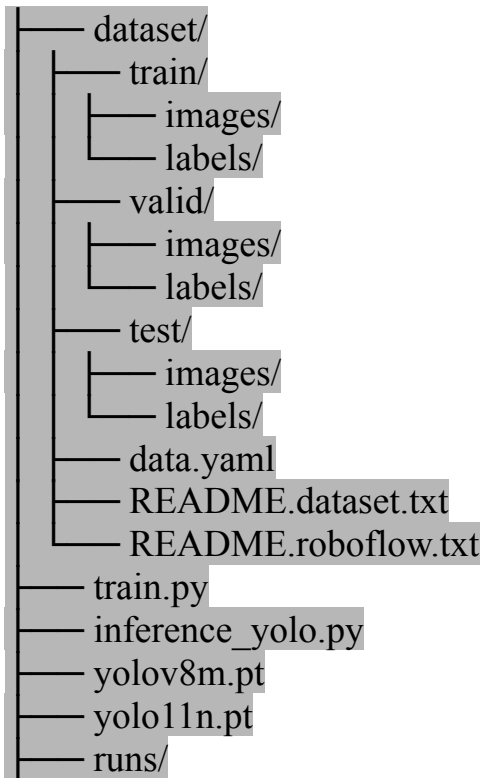
У процесі тренування застосовувався *early stopping* — автоматична зупинка навчання при відсутності покращення метрик.

Модель була навчена на власному сервері з використанням GPU NVIDIA RTX 3090. Бібліотеки та фреймворки:

- Python 3.10,

- PyTorch 2.x,
- Ultralytics YOLOv8 (версії з GitHub),
- OpenCV, NumPy, Pandas — для обробки зображень та аналізу.

Вся структура проекту організована за принципом:



```
egor@dropla: ~/dip  
┌───┴───┐  
└─ GNU nano 7.2 ─┘  
from ultralytics import YOLO  
  
model = YOLO("yolov8m.pt")  
  
model.train(  
    data="./dataset/data.yaml",  
    epochs=100,  
    imgsz=640,  
    batch=16,  
    name="vehicle_detector",  
    project="runs/train",  
    device=0  
)  
  
egor@dropla:~/dip$ python train.py  
Read 13 lines
```



Рисунок 2.6 - Фінальний результат навчання моделі YOLOv8: вивід метрик після 100 епох та валідація best.pt

Навчання тривало приблизно 1 години на 100 епох. Результати (втрати, точність, mAP) моніторились через вбудовані графіки Ultralytics. Регулярно зберігалися чекпойнти (best.pt, last.pt) для подальшого аналізу. Модель також тестувалась на реальних зображеннях після навчання — для перевірки якості узагальнення.

На валідаційному наборі даних було досягнуто:

- mAP@0.5: ~0.88,

- Precision: ~0.80,

- Recall: ~0.86.

Ці показники підтверджують здатність моделі ефективно ідентифікувати об'єкти, навіть в умовах з обмеженою видимістю або неконтрастним фоном.

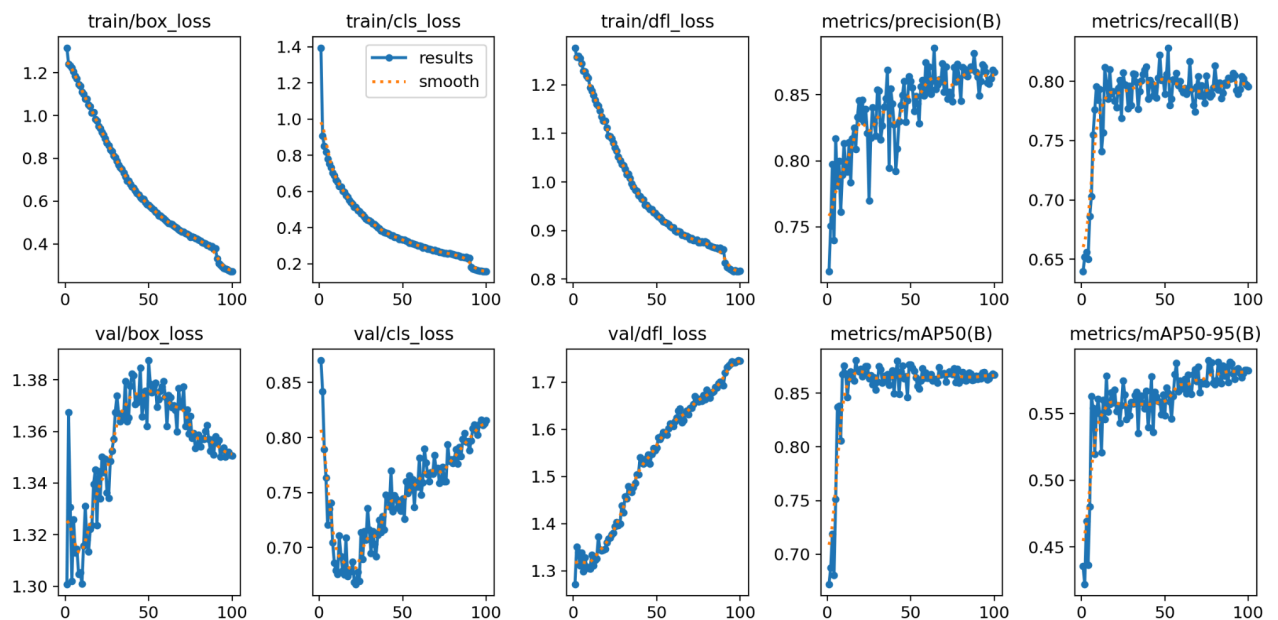


Рисунок 2.7 - рафік динаміки втрат (loss), точності (precision), повноти (recall) та mAP протягом навчання YOLOv8

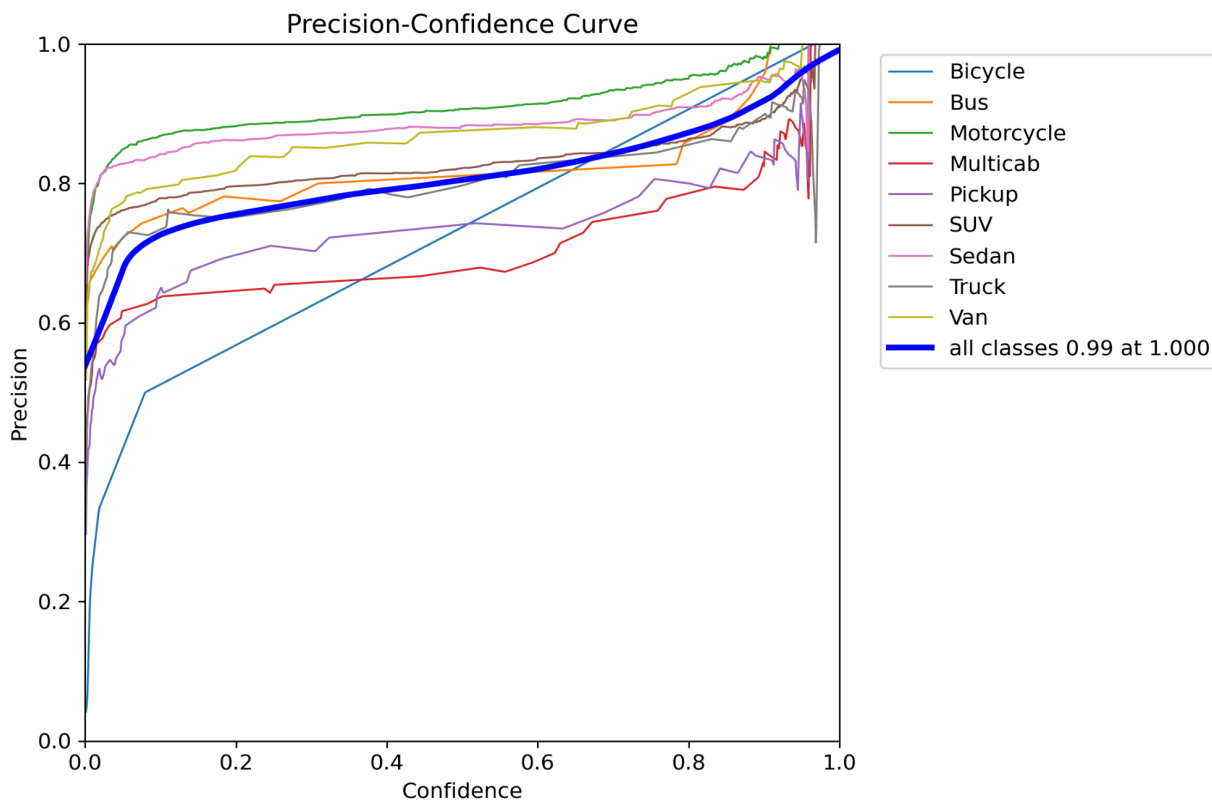


Рисунок 2.7 - Крива точності (Precision curve) по класах

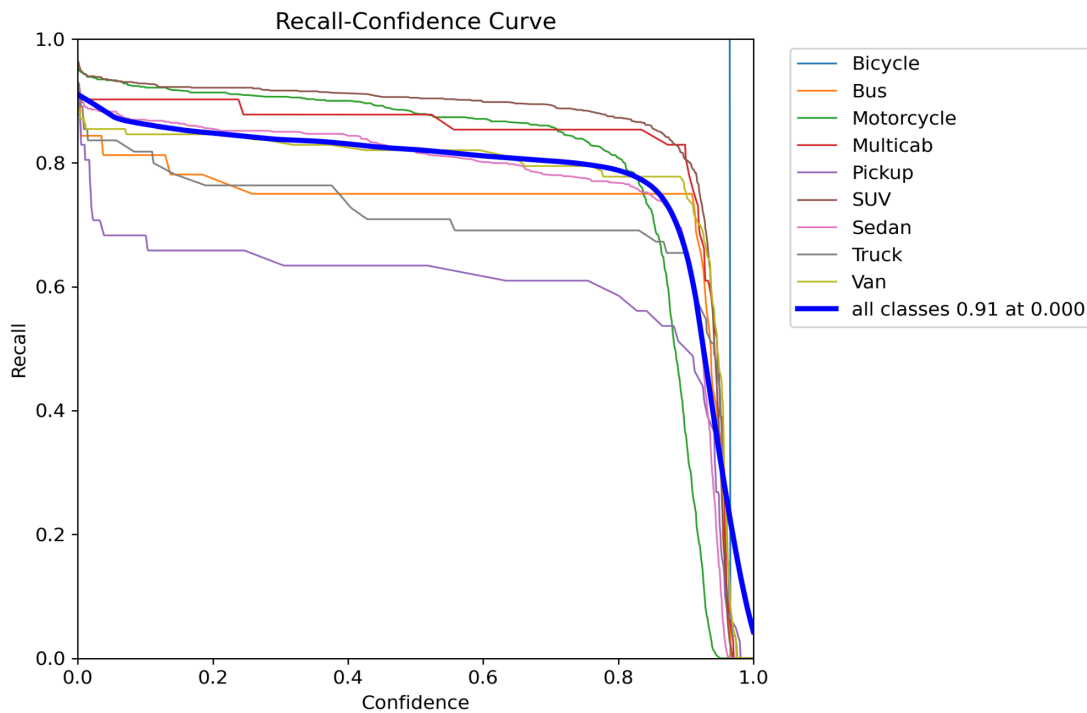


Рисунок 2.8 - Крива повноти (Recall curve) по класах

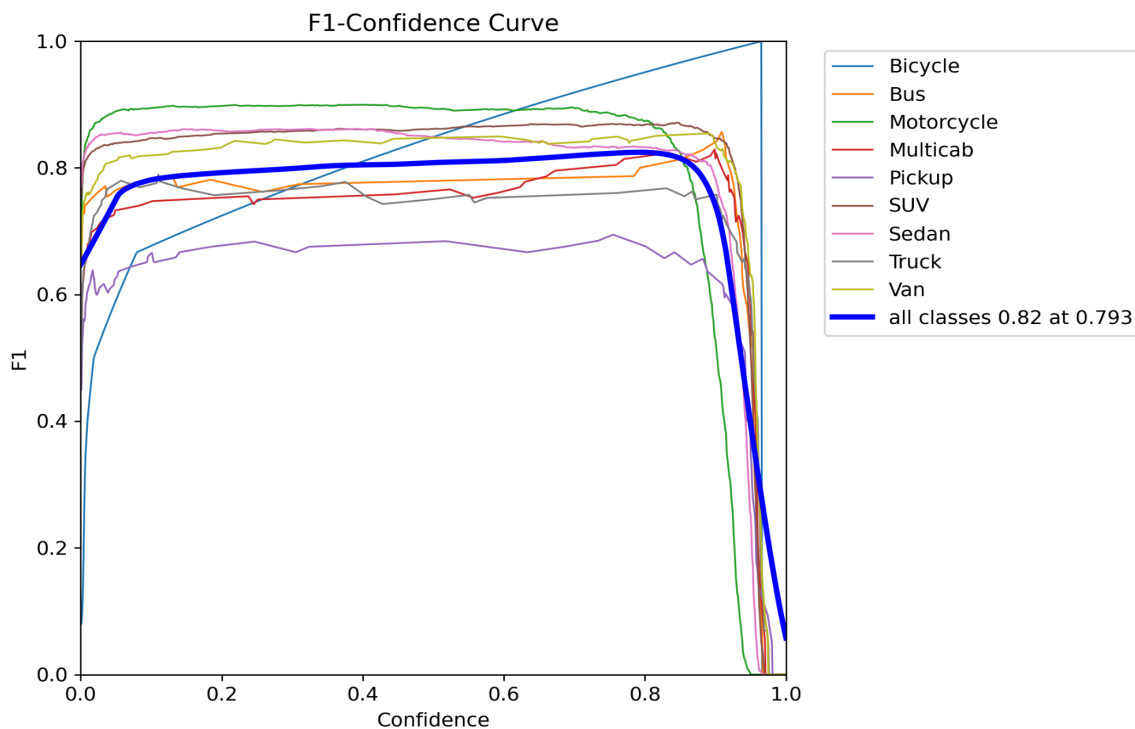


Рисунок 2.9 - Крива F1-міри по класах

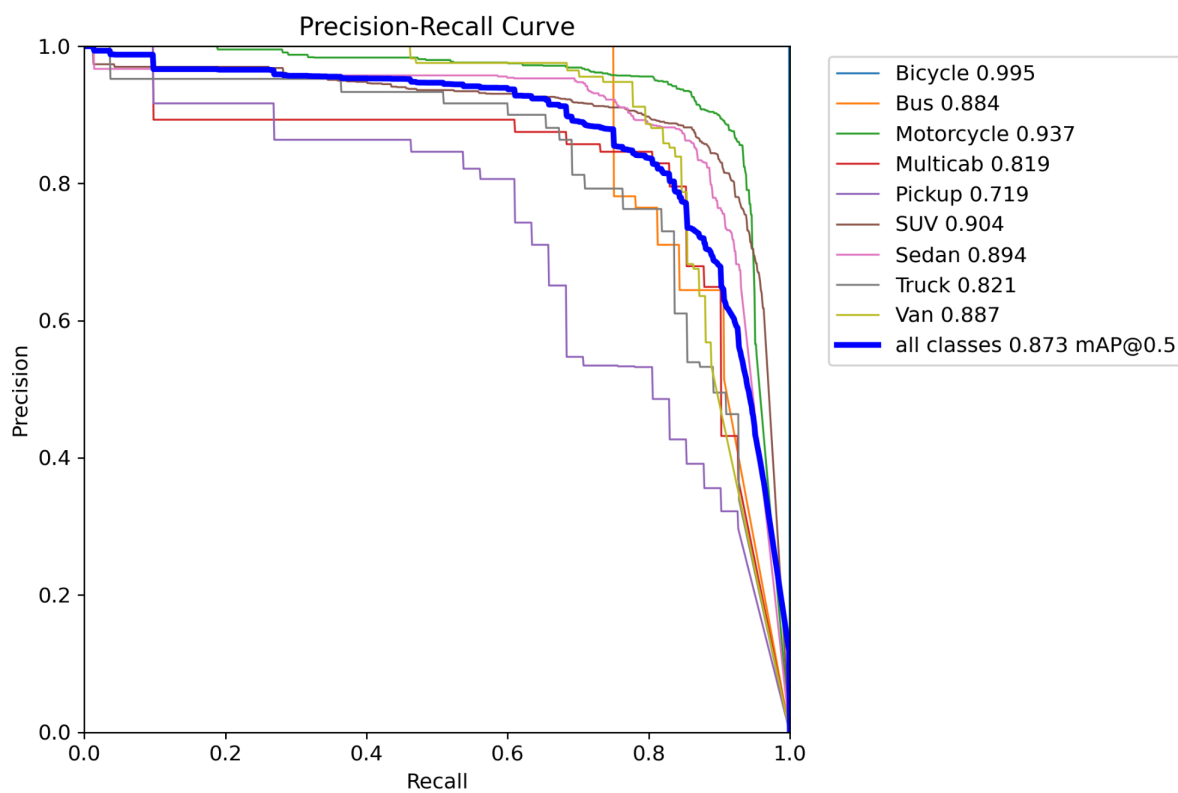


Рисунок 2.10 - Залежність між Precision і Recall (PR-крива)

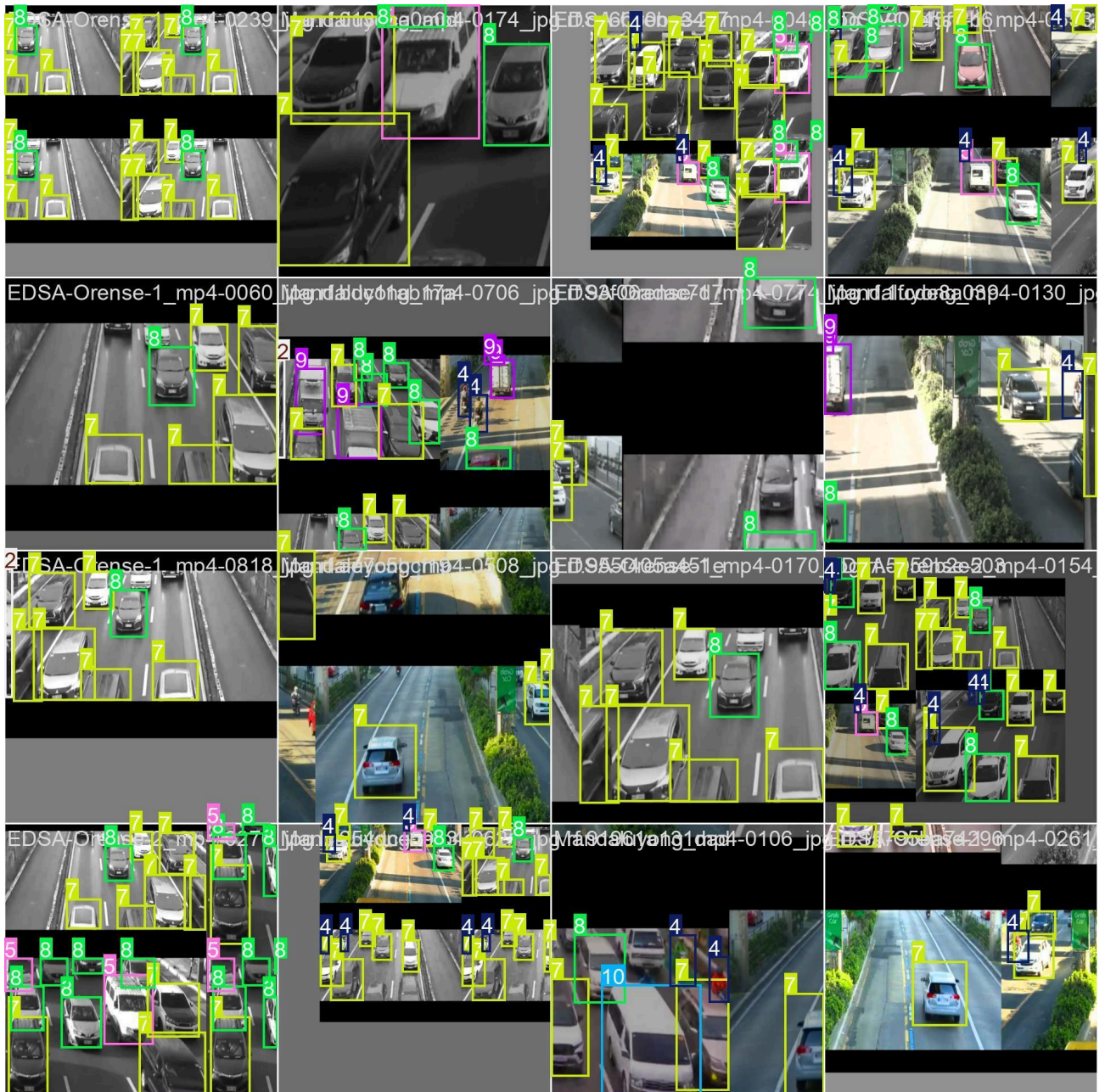


Рисунок 2.11 - Приклади зображень із навчального набору з анотаціями перед початком навчання YOLOv8

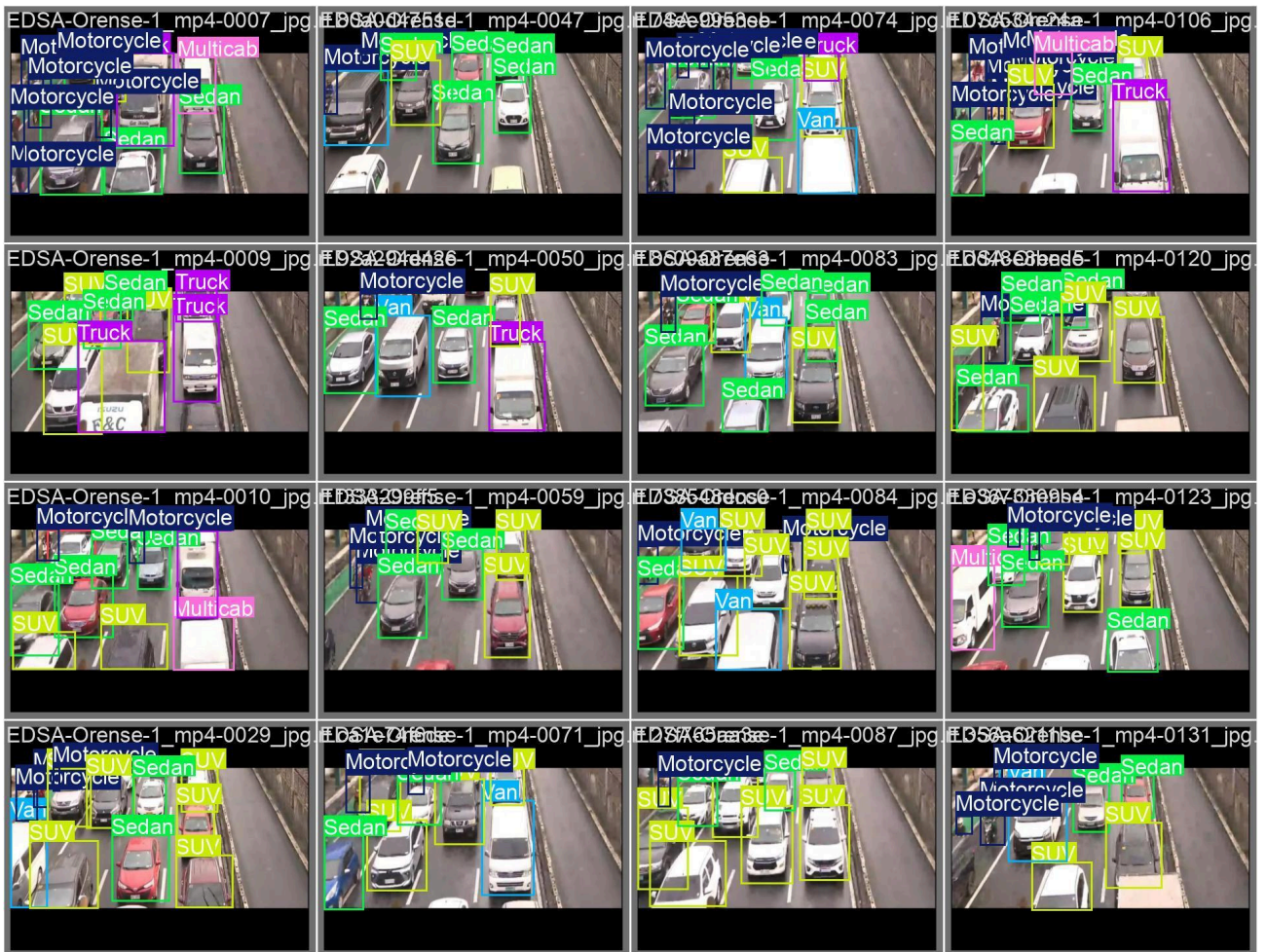


Рисунок 2.12 - Референсна розмітка на валідаційній підбірці

Розмір вхідного зображення мав значення — більші зображення дозволяють краще виявляти дрібні об'єкти, але значно збільшують час обробки. Розмір 640×640 було обрано як баланс між точністю і продуктивністю.

Було також враховано проблему дисбалансу класів: у наборі даних деякі типи об'єктів траплялись частіше за інші. Для боротьби з цим використовували зважену функцію втрат і повторне використання рідкісних прикладів у навчанні (upsampling).

## 2.4 Аналіз ефективності роботи моделі

Оцінка якості роботи моделі детекції об'єктів — один із найважливіших етапів, що дозволяє зробити висновки щодо її практичної придатності, надійності та стабільності.

Основні метрики:

- mAP@0.5 (mean Average Precision при  $\text{IoU} \geq 0.5$ ):

Це середній показник точності виявлення об'єктів. У цій роботі модель досягла  $\text{mAP}@0.5 \approx 0.88$ , що вказує на високий рівень узгодженості між передбаченими bounding box та реальними об'єктами.

- Precision  $\approx 0.80$  (Точність):

Високе значення означає, що 91% усіх передбачень були правильними. Тобто модель рідко помиляється у бік "хибної тривоги" (false positives).

- Recall  $\approx 0.86$  (Повнота):

Цей показник означає, що 87% усіх реальних об'єктів були виявлені. Тобто модель майже не пропускає об'єкти, які вона повинна знайти.

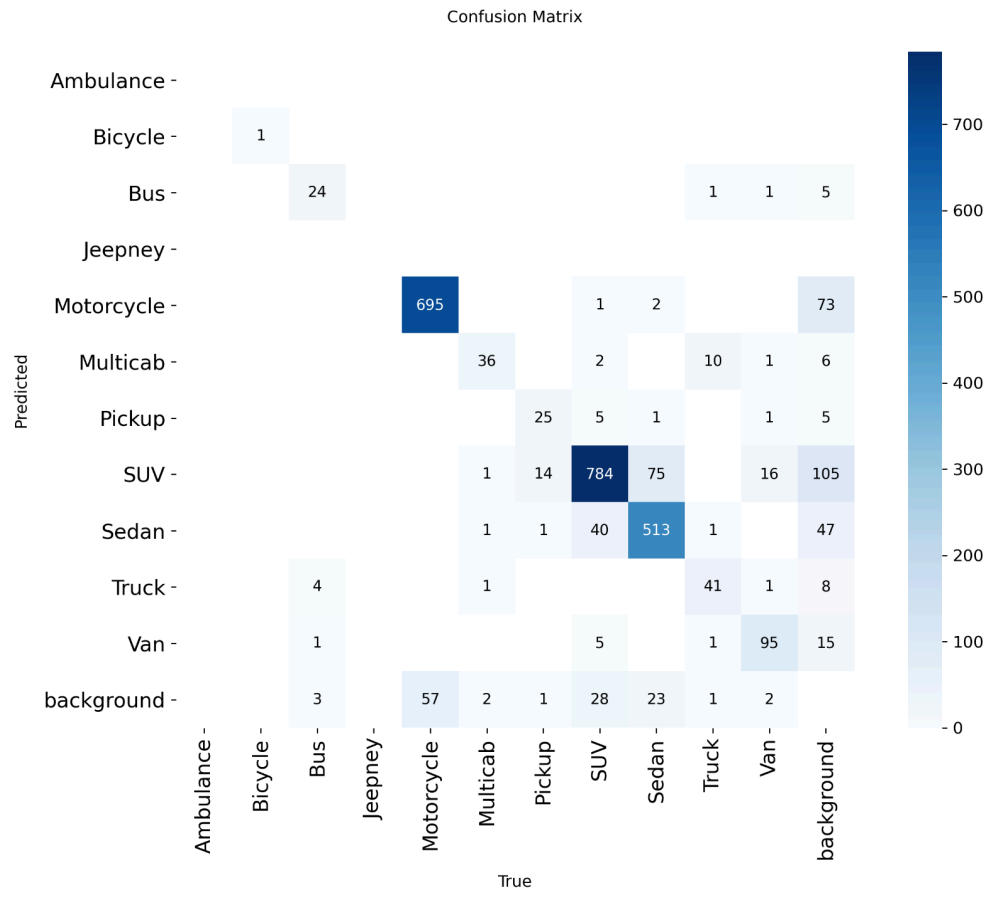


Рисунок 2.13 - Матриця плутанини (Confusion matrix)

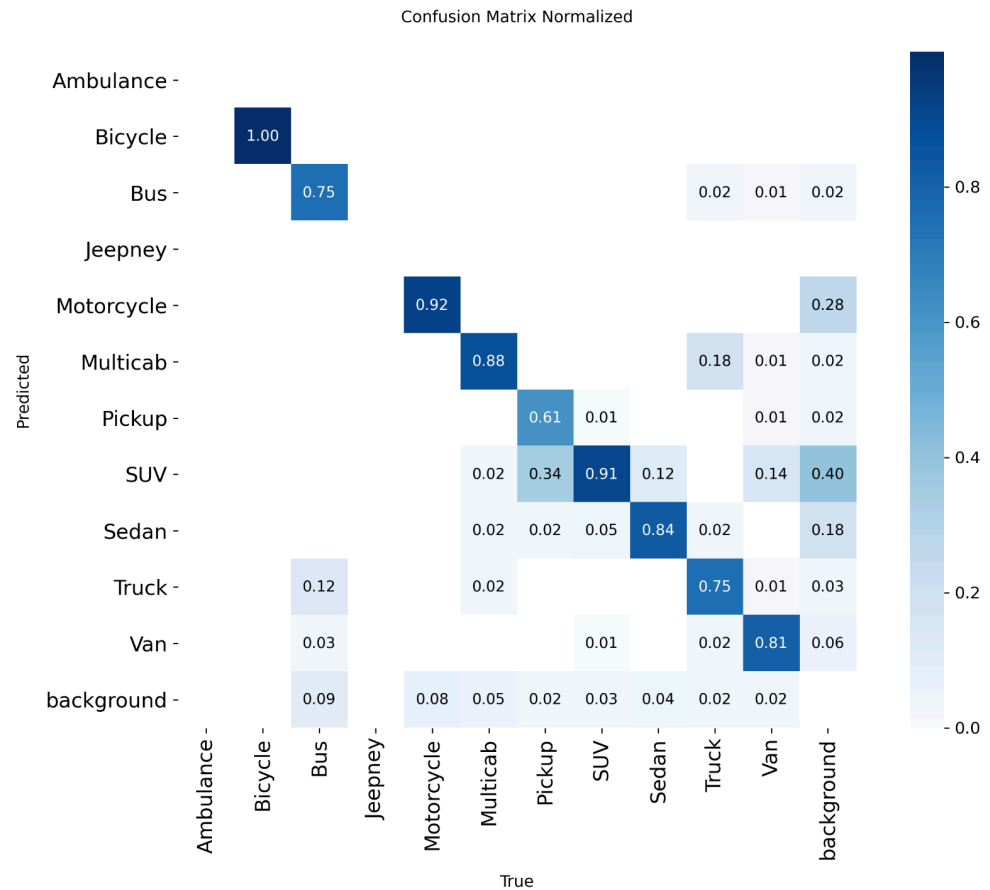


Рисунок 2.14 - Нормалізована матриця плутанини

Тлумачення метрик:

Метрика Пояснення Високе значення означає

mAP@0.5 Середня точність при перекритті  $\geq 50\%$  Хороша якість локалізації та класифікації

Precision Частка правильних передбачень Мало хибно-позитивних спрацювань

Recall Частка знайдених справжніх об'єктів Мало пропущених об'єктів

Візуалізація результатів:

Після тренування були побудовані графіки:

- втрат (loss) по епохах,
- точності класифікації (cls\_precision),
- якості локалізації (box\_loss),

- загального mAP по класах.

Скріншоти прикладів детекції також додаються безпосередньо до цього підрозділу.

Умови тестування:

Модель тестувалась локально на комп'ютері з такими характеристиками:

- Відеокарта: NVIDIA GeForce RTX 3090 (24GB VRAM),
- Процесор: AMD Ryzen 9 7950X3D,
- RAM: 128GB DDR5.

Інференс проводився на зображеннях та відео з різних джерел (статичні фото, потік із камери, кадри з дронів).

Для оцінки працездатності моделі в реальних умовах було здійснено тестування на відеопотоці. Нижче представлено фрагмент кадру, на якому YOLOv8 виконує детекцію транспортних засобів у динамічному середовищі.

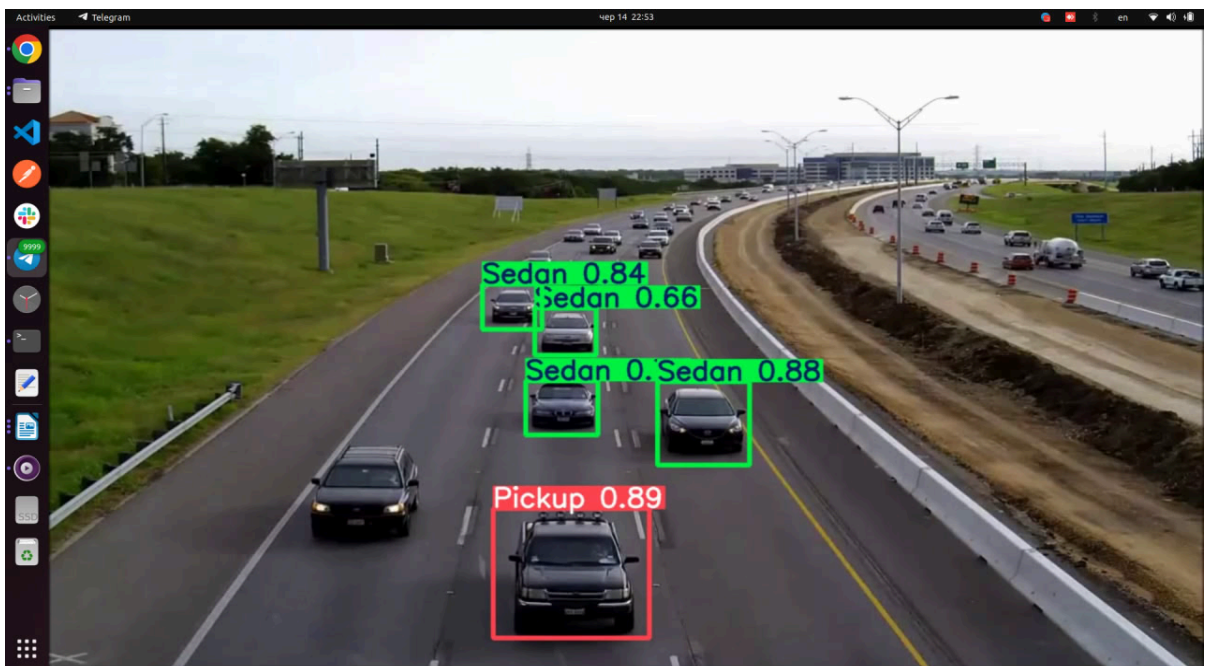


Рисунок 2.15 – Приклад роботи моделі YOLOv8 на відеопотоці в реальному часі

Також було проведено перевірку у статичному середовищі.

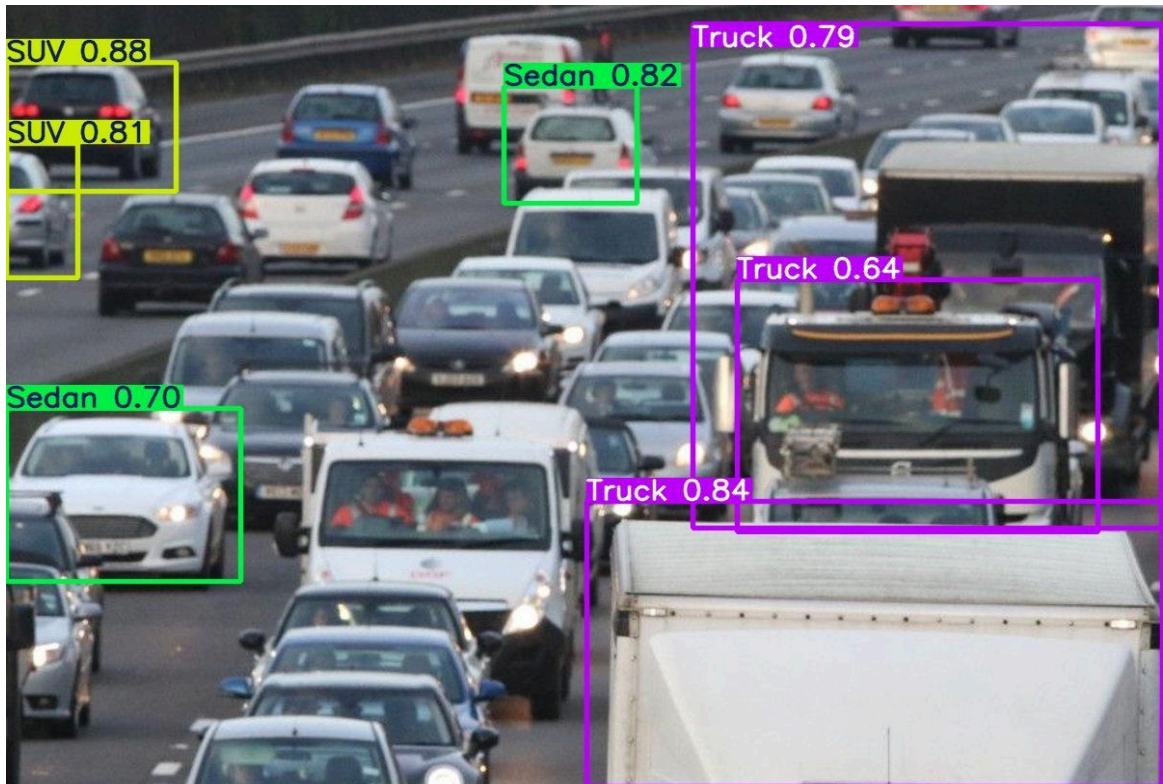


Рисунок 2.16 – Результат предикта моделі YOLOv8 на статичному зображенні

#### Висновки:

Отримані результати підтверджують, що YOLOv8m є ефективним інструментом для задач реального часу, навіть у складних умовах з обмеженими обчислювальними ресурсами. Модель демонструє збалансовану поведінку: і високу точність, і хорошу повноту.

### 2.5 Проблеми та обмеження в реальному застосуванні YOLO

Хоча архітектура YOLO є однією з найпопулярніших та ефективних у задачах детекції об'єктів, її практичне впровадження супроводжується низкою проблем та обмежень, які важливо враховувати.

#### Неточність розмітки:

Навчання нейронної мережі значною мірою залежить від якості розмітки зображень. Навіть незначне зміщення bounding box або помилка у класифікації

об'єкта може негативно вплинути на результат. Великий обсяг ручної розмітки також спричиняє втому розмітника та збільшує ризик помилок.

Нестача або незбалансованість даних:

У деяких випадках спостерігається домінування одного класу об'єктів у датасеті, що призводить до переобучення моделі на нього і поганої генералізації. Рідкісні об'єкти модель розпізнає гірше, а в деяких випадках — зовсім не виявляє.

Проблеми зі світлом, кутом огляду, фоном:

YOLO відносно стійкий до змін середовища, але значні відхилення (наприклад, нічна зйомка, сильна засвітка, погана фокусна відстань) суттєво знижують точність.

Вартість обчислень:

Навіть оптимізовані моделі YOLOv5 або YOLOv8 потребують потужних GPU, особливо при роботі з великими зображеннями або відео. Для edge-пристроїв потрібна додаткова оптимізація (квантування, pruning, зниження роздільної здатності), що не завжди дає очікуваний результат без втрати точності.

Обмеження у детекції дрібних та частково перекритих об'єктів:

YOLO має обмежену здатність до виявлення об'єктів, що частково перекриті іншими або займають малу частину кадру. Це критично для задач типу відеоспостереження, де потрібно виявляти деталі.

Таким чином, для досягнення максимальної ефективності потрібно:

- підвищувати якість датасету та його баланс,
- враховувати сценарії з несприятливими умовами,
- проводити тюнінг моделі для конкретних задач,

- поєднувати YOLO з іншими технологіями (наприклад, трекінгом, сегментацією, тепловізорами).

## 2.6 Оптимізація нейронної мережі для впровадження в реальні системи

Для практичного застосування нейронних мереж, особливо у системах реального часу або на пристроях з обмеженими ресурсами (edge computing), необхідна оптимізація моделі. Без цього навіть добре натренована модель може виявитися непрактичною для розгортання в реальному середовищі.

Квантування (Quantization):

Цей метод полягає у зменшенні точності обчислень (наприклад, з float32 до int8). Це дозволяє значно зменшити розмір моделі та пришвидшити інференс. У більшості випадків втрата точності мінімальна. Інструменти: PyTorch Quantization Toolkit, TensorRT, OpenVINO.

Проріджування (Pruning):

Це процес видалення малозначущих зв'язків або нейронів. Таким чином, модель стає меншою та швидшою. Проріджування можна виконувати під час або після навчання. Проте, надмірне проріджування може призвести до втрати якості.

Knowledge Distillation:

Метод, за якого "легка" модель навчається від "важкої". Застосовується для збереження продуктивності при значному скороченні розміру та кількості параметрів.

Використання ONNX або TensorRT:

Формат ONNX (Open Neural Network Exchange) дозволяє експортувати модель у кросплатформенний формат. TensorRT — фреймворк від NVIDIA для високопродуктивного інференсу, особливо ефективний на відеокартах серії RTX та Jetson.

Edge-платформи для inference:

- NVIDIA Jetson Nano / Orin: оптимізовані для запуску YOLOv8 з TensorRT.
- Raspberry Pi з Movidius або Coral TPU: підтримка прискореного інференсу.
- STM32 / ESP32: використовуються в проектах TinyML для найпростіших задач. У рамках цієї роботи тестувалась модель YOLOv8m, оптимізована через ONNX + TensorRT, локально на системі з NVIDIA RTX 3090, а також на Jetson Nano з INT8-квантуванням.

Зменшення роздільної здатності зображень:

У задачах реального часу доцільним є зменшення input-size (наприклад, з  $640 \times 640$  до  $416 \times 416$ ). Це пришвидшує обробку у 1.5–2 рази з незначною втратою mAP.

Комбіновані підходи:

Оптимізаційні техніки можуть комбінуватися. Наприклад, після pruning можна застосувати quantization або знання передати через distillation до легкої edge-моделі.

Оптимізація є критичною частиною будь-якого застосування моделей у виробничих або військових системах. Вибір конкретного підходу залежить від обмежень платформи, вимог до точності, ресурсів живлення та надійності.

## 2.7 Впровадження у складну систему: план, ризики, довгостроковий ефект

Після успішного навчання, тестування та оптимізації моделі, наступним кроком є її інтеграція у реальну складну систему. Таке впровадження вимагає детального планування, оцінки ризиків, підготовки інфраструктури та прогнозування довгострокових наслідків.

Інтеграція у виробничу або військову систему:

Модель може бути розгорнута як частина ПЗ, яке обробляє відеопотік з камер у

реальному часі. Наприклад:

- У промисловості — для виявлення дефектів продукції або контролю процесів,
- У безпекових системах — для моніторингу об'єктів, виявлення порушень або несанкціонованого доступу,
- У військових цілях — для автоматичного виявлення техніки, дронів, переміщень ворога на основі кадрів з дронів чи камер.

Прогнозована ефективність:

На основі тестових результатів можна очікувати високу продуктивність системи: до 60 FPS на потужному GPU, точність виявлення до 90%, низька латентність обробки.

Точки контролю та автоматизація:

- Автоматичне логування подій;
- Система повідомлень при спрацюванні певних тригерів;
- Збереження кадрів із високим confidence level;
- Інтеграція з базами даних або ERP/SCADA системами.

Безпека і надійність:

- Передбачено контроль доступу до результатів;
- Регулярне оновлення моделі на основі нових даних;
- Механізми валідації рішень людиною (human-in-the-loop);
- Резервне копіювання конфігурацій та логів.

Ризики:

- Помилкові спрацювання в умовах нестандартного фону;
- Перевантаження системи при великому потоці даних;
- Потреба в регулярному обслуговуванні і перевірці точності;
- Залежність від обладнання (камери, освітлення, обчислювальні ресурси).

Таким чином, для успішного впровадження необхідно забезпечити не лише технічну реалізацію, але й супровід, адаптацію до змін, навчання персоналу та наявність плану відновлення у разі збоїв.

## ВИСНОВКИ

У другому розділі було проведено практичне дослідження застосування нейронної мережі YOLO для задачі ідентифікації об'єктів у складній системі. Розглянуто архітектурні особливості моделі, обґрунтовано вибір її конфігурації, описано процес підготовки даних та навчання.

Проведено аналіз ефективності, який підтвердив високі показники точності та продуктивності. Значення метрик, таких як mAP@0.5, precision та recall, свідчать про стабільну та надійну роботу моделі навіть у складних умовах.

Було також визначено типові проблеми, що виникають під час впровадження моделі в реальних середовищах: залежність від якості розмітки, обмеження щодо розміру об'єктів, апаратні вимоги. Здійснено розгляд методів оптимізації, які дозволяють адаптувати модель для роботи на edge-пристроях, таких як NVIDIA Jetson або Raspberry Pi.

Нарешті, було сформульовано стратегію впровадження у виробничу або військову систему, з урахуванням потенційних ризиків, потреби у моніторингу, автоматизації та забезпеченні безперервної підтримки системи.

Практичне значення цієї роботи полягає у можливості швидкої та ефективної інтеграції інтелектуальних систем візуального аналізу у складні технічні об'єкти, що потребують оперативного прийняття рішень.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Літак М. В., Буряк Л. П. Штучні нейронні мережі та їх застосування у складних системах. — Київ: Наукова думка, 2021. — 256 с.
2. Ковальчук О. А. Інтелектуальні системи керування на основі нейронних мереж. — Харків: Техніка, 2020. — 184 с.
3. Goodfellow I., Bengio Y., Courville A. Deep Learning. — MIT Press, 2016. — 775 p.
4. Ultralytics. YOLO Documentation. URL: <https://docs.ultralytics.com/>
5. Bochkovskiy A., Wang C.-Y., Liao H.-Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv:2004.10934 [cs.CV]
6. Jocher G. et al. YOLOv5 by Ultralytics. GitHub repository: <https://github.com/ultralytics/yolov5>
7. Redmon J., Farhadi A. YOLOv3: An Incremental Improvement. arXiv:1804.02767 [cs.CV]
8. TensorRT Developer Guide. NVIDIA. URL: <https://docs.nvidia.com/deeplearning/tensorrt/>
9. Open Neural Network Exchange (ONNX). URL: <https://onnx.ai/>
10. Zhang X., et al. Efficient Object Detection for Real-Time Applications. Journal of Computer Vision and Pattern Recognition, 2022.
11. Roboflow Blog — How to Annotate and Train Object Detection Models. URL: <https://blog.roboflow.com/>
12. Lin T.-Y. et al. Microsoft COCO: Common Objects in Context. arXiv:1405.0312 [cs.CV]

13. Papers with Code — Object Detection Benchmarks. URL: <https://paperswithcode.com/task/object-detection>
14. Jetson Nano Developer Kit. NVIDIA Docs. URL: <https://developer.nvidia.com/embedded/jetson-nano-developer-kit>
15. Paszke A., et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. NeurIPS, 2019.
16. Кваліфікаційна робота бакалавра [Електронний ресурс] : методичні рекомендації для здобувачів ступеня бакалавра освітньо-професійної програми «Системний аналіз» зі спеціальності 124 Системний аналіз / уклад.: Т.А. Желдак, Т.В. Хом'як, А.В. Малієнко ; М-во освіти і науки України, Нац. техн. ун-т «Дніпровська політехніка». – Дніпро : НТУ «ДП», 2025. – 32 с.

## Додаток А

№ з/п	Позначення				Найменування	Кількість аркушів	Примітки		
1									
2					Документація				
3									
4	САУ.КР.25.31.ДА.ПЗ.				Пояснювальна записка	N1	Формат А4		
5									
6	САУ.КР.25.31.ДА.ПЗ.				Демонстраційний матеріал	N2	Презентація на CD-R		
7									
8	САУ.КР.25.31.ДА.ПЗ.				Копія роботи	1	Диск CD-R		
9									
10									
11									
12									
13									
14									
15									
16									
17									
18									
					САУ.КР.25.31.ДА.ПЗ				
Змін.	Аркуш	№ докум.	Підпис	Дата					
Розроб.		Копил Є.О.			<b>Матеріали кваліфікаційної роботи</b>	Літ.	Аркуш	Аркуші в	
К. розд.		Владико О.Б							
Керівн.		Владико О.Б.				НТУ «ДП», 12; 124-22ск-1			
Н.контр.		Хом'як Т.В.							
Зав. каф.		Желдак Т.А.							

**Додаток Б**  
**Відгук**  
**на кваліфікаційну роботу бакалавра**  
**здобувача вищої освіти групи 124 – 22ск– 1**  
**спеціальності 124 Системний аналіз**

Тема кваліфікаційної роботи: «Застосування нейронних мереж як систем ідентифікації, класифікації, керування чи прогнозування діяльності складної системи»

Обсяг кваліфікаційної роботи 48 стор.

Мета кваліфікаційної роботи: розробка, дослідження та аналіз ефективності застосування глибоких нейронних мереж для задач ідентифікації об'єктів у складних системах на прикладі YOLO.

Актуальність теми полягає в тому, що потреба в автоматизованому аналізі зображень і відео значно зросла внаслідок розвитку автономних технологій, військових застосувань, систем моніторингу, безпеки та промислової діагностики. Одним із найефективніших підходів, що довів свою практичну цінність, є модель YOLO (You Only Look Once), яка поєднує швидкість обробки з високою якістю розпізнавання.

Тема кваліфікаційної роботи безпосередньо пов'язана з об'єктом діяльності бакалавра спеціальності 124 Системний аналіз, оскільки результати отримані в ході виконання роботи, можуть використовуватися у подальшій діяльності системного або фінансового аналітика.

Виконані в кваліфікаційній роботі завдання відповідають вимогам ступеня бакалавра. Оригінальність наукових рішень полягає в удосконаленні підходу до розробки оперативних та довгострокових планів підприємства. В результаті кваліфікаційної роботи було проведено практичне дослідження застосування нейронної мережі YOLO для задачі ідентифікації об'єктів у складній системі. Розглянуто архітектурні особливості моделі, обґрунтовано вибір її конфігурації, описано процес підготовки даних та навчання.

Оформлення пояснювальної записки та демонстраційного матеріалу до неї виконано згідно з вимогами. Роботу виконано самостійно, відповідно до завдання та у повному обсязі.

У роботі відзначено такі недоліки: не розглянуті можливості для тестування на платформах NVIDIA Jetson і не приведений план підтримки та оновлення моделі в промисловому/військовому середовищі.

Кваліфікаційна робота в цілому заслуговує оцінки: (85), «добре».

З урахуванням висловлених зауважень автор заслуговує присвоєння кваліфікації «бакалавр з системного аналізу».

Керівник кваліфікаційної роботи бакалавра,  
 науковий ступінь, вчене звання, посада

ВЛАДИКО О.Б., к.т.н., доцент / ПІБ