

**Державний торговельно-економічний університет**

**Кафедра комп'ютерних наук та інформаційних систем**

**КВАЛІФІКАЦІЙНА РОБОТА**

на тему:

**«Розпізнавання номерних знаків та ідентифікація  
типов транспортних засобів з використанням deep  
learning»**

Студента 4 курсу, 7 групи,  
спеціальності  
F3 «Комп'ютерні науки»

*pідпис студента*

Санжаров  
Даніїл  
Русланович

Науковий керівник  
кандидат фізико-математичних наук,  
доцент

*pідпис керівника*

Філімонова Тетяна  
Олегівна

Гарант освітньої програми  
кандидат технічних наук, доцент

*pідпис керівника*

Демідов Павло  
Георгійович

**Київ 2025**

**Державний торговельно-економічний університет**

**Факультет інформаційних технологій**

**Кафедра комп'ютерних наук та інформаційних систем**

**Спеціальність F3 «Комп'ютерні науки»**

**Затверджую**

**Зав. кафедри Пурський О.І.**

**«16» грудня 2024 р.**

**Завдання  
на кваліфікаційну роботу студенту**

**Санжарову Даніїлу Руслановичу**

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи.

«Розпізнавання номерних знаків та ідентифікація типів транспортних засобів з використанням deep learning»

Затверджена наказом ректора від «13» грудня 2024 р. № 4080

2. Срок здачі студентом закінченої роботи 30 травня 2025 року

3. Цільова установка та вихідні дані до роботи

Мета роботи: розробка моделі комп'ютерного зору для виявлення транспортних засобів та системи для розпізнавання номерних знаків

Об'єкт дослідження: процеси розпізнавання та обробки зображень транспортних засобів з використанням методів deep learning

Предмет дослідження: моделі, методи та інформаційні технології комп'ютерного зору

4. Перелік графічного матеріалу \_\_\_\_\_

5. Консультанти по роботі із зазначенням розділів, за якими здійснюється консультування:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Філімонова Т.О.	22.12.2024 р.	22.12.2024 р.
2	Філімонова Т.О.	22.12.2024 р.	22.12.2024 р.
3	Філімонова Т.О.	22.12.2024 р.	22.12.2024 р.

6. Зміст кваліфікаційної роботи (перелік питань за кожним розділом)

**РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ РОЗПІЗНАВАННЯ ОБ'ЄКТІВ ЗА ДОПОМОГОЮ DEEP LEARNING**

1.1. Сутність та завдання комп'ютерного зору в контексті розпізнавання об'єктів

1.2. Методи ідентифікації та класифікації об'єктів для задач комп'ютерного зору

1.2.1 Традиційні методи

1.2.2 Згорткові нейронні мережі

1.2.3 You only look once

1.2.4. Зореві трансформери

1.2.5 Оптичне розпізнавання тексту

1.3. Огляд наявних рішень для розпізнавання номерних знаків

1.3.1 Автоматичне розпізнавання номерних знаків на основі YOLOv8 та EasyOCR

1.3.2 Автоматичне розпізнавання номерних знаків на основі R-CNN

1.3.3 Автоматичне розпізнавання номерних знаків на основі YOLOv2 та Fast-YOLO

**РОЗДІЛ 2. МОДЕЛІ ТА МЕТОДИ РОЗПІЗНАВАННЯ НОМЕРНИХ ЗНАКІВ ТА КЛАСИФІКАЦІЇ ТРАНСПОРТНИХ ЗАСОБІВ**

2.1. Особливості розпізнавання номерних знаків з використанням deep learning

2.2. Методи попередньої обробки даних

2.3. Проектування методу виявлення та розпізнавання

**РОЗДІЛ 3. РОЗРОБКА ТЕХНОЛОГІЇ ВИЯВЛЕННЯ ТА РОЗПІЗНАВАННЯ**

3.1. Збір та підготовка даних

3.2. Програмна реалізація технології розпізнавання

3.2.1 Розробка моделі комп'ютерного зору на основі YOLO

3.2.2 Програмна реалізація розпізнавання тексту

3.3. Технологія використання та оцінка отриманої моделі комп'ютерного зору

3.3.1 Аналіз результатів розпізнавання номерних знаків

3.3.2 Аналіз результатів ідентифікації транспортних засобів

**ВИСНОВКИ**

**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

7. Календарний план виконання роботи

№ Пор . .	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	
		За планом	фактично
1	2	3	4
1	<i>Вибір теми кваліфікаційної роботи</i>	<i>05.10.2024</i>	<i>05.10.2024</i>
2	<i>Розробка та затвердження завдання на кваліфікаційну роботу</i>	<i>22.12.2024</i>	<i>31.01.2025</i>
3	<i>Вступ</i>	<i>10.02.2025</i>	<i>10.02.2025</i>
4	<i>РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ РОЗПІЗНАВАННЯ ОБ'ЄКТІВ ЗА ДОПОМОГОЮ DEEP LEARNING</i>	<i>28.02.2025</i>	<i>28.02.2025</i>
5	<i>РОЗДІЛ 2. МОДЕЛІ ТА МЕТОДИ РОЗПІЗНАВАННЯ НОМЕРНИХ ЗНАКІВ ТА КЛАСИФІКАЦІЇ ТРАНСПОРТНИХ ЗАСОБІВ</i>	<i>04.04.2025</i>	<i>04.04.2025</i>
6	<i>РОЗДІЛ 3. РОЗРОБКА ТЕХНОЛОГІЙ ВИЯВЛЕННЯ ТА РОЗПІЗНАВАННЯ</i>	<i>10.05.2025</i>	<i>10.05.2025</i>
7	<i>Висновки</i>	<i>15.05.2025</i>	<i>12.05.2025</i>
8	<i>Здача випускної кваліфікаційної роботи на кафедру науковому керівнику</i>	<i>20.05.2025</i>	<i>20.05.2025</i>
9	<i>Попередній захист випускної кваліфікаційної роботи</i>	<i>27.05.2025</i>	<i>27.05.2025</i>
11	<i>Виправлення зауважень, зовнішнє рецензування випускної кваліфікаційної роботи</i>	<i>28.05.2025</i>	<i>28.05.2025</i>
12	<i>Представлення готової зшитої випускної кваліфікаційної роботи на кафедру</i>	<i>30.05.2025</i>	<i>30.05.2025</i>
13	<i>Публічний захист випускної кваліфікаційної роботи</i>	<i>За розкладом роботи ЕК</i>	

8. Дата видачі завдання «22» грудня 2024 р.

9. Керівник кваліфікаційної роботи

Філімонаова Т.О.

(прізвище, ініціали, підпис)

10. Гарант освітньої програми

Демідов П.Г.

(прізвище, ініціали, підпис)

11. Завдання прийняв до виконання студент

Санжаров Д.Р.

(прізвище, ініціали, підпис)

**12. Відгук керівника кваліфікаційної роботи**

У кваліфікаційній роботі реалізовано програмну систему на основі YOLOv11 для виявлення транспортних засобів і EasyOCR для розпізнавання тексту на номерних знаках. Проведено експериментальну оцінку ефективності розробленої моделі на реальних даних. Всі поставлені завдання виконані, робота може бути допущена до захисту.

---

---

Керівник кваліфікаційної роботи 30.05.2025 р.  
(*підпис, дата*)

**13. Висновок про кваліфікаційну роботу**

Кваліфікаційна робота студента Санжаров Д.Р.  
(*прізвище, ініціали*)

може бути допущена до захисту екзаменаційній комісії.

Гарант освітньої програми Демідов П.Г.  
(*підпис, прізвище, ініціали*)

Завідувач кафедри Пурський О.І.  
(*підпис, прізвище, ініціали*)

«\_\_\_\_\_» 2025 р.

## **Анотація**

У кваліфікаційній роботі розглядається проблема автоматичного розпізнавання номерних знаків та ідентифікації типів транспортних засобів із використанням сучасних методів глибокого навчання (deep learning). Проведено аналіз теоретичних основ комп'ютерного зору, традиційних і сучасних підходів до виявлення та класифікації об'єктів, зокрема згорткових нейронних мереж, архітектури YOLO та зорових трансформерів. Запропоновано та реалізовано програмну систему на основі YOLOv11 для виявлення транспортних засобів і EasyOCR для розпізнавання тексту на номерних знаках. Проведено експериментальну оцінку ефективності розробленої моделі на реальних даних.

**Ключові слова:** комп'ютерний зір, глибоке навчання, розпізнавання номерних знаків, ідентифікація транспортних засобів, YOLO, EasyOCR, згорткові нейронні мережі.

## **Annotation**

This qualification thesis addresses the problem of automatic license plate recognition and vehicle type identification using modern deep learning methods. Theoretical foundations of computer vision, as well as traditional and advanced approaches to object detection and classification, including convolutional neural networks, YOLO architecture, and vision transformers, are analyzed. A software system based on YOLOv11 for vehicle detection and EasyOCR for license plate text recognition is proposed and implemented. Experimental evaluation of the developed model on real-world data.

**Keywords:** computer vision, deep learning, license plate recognition, vehicle identification, YOLO, EasyOCR, convolutional neural networks.

## ЗМІСТ

<b>ВСТУП.....</b>	<b>9</b>
<b>РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ РОЗПІЗНАВАННЯ ОБ'ЄКТІВ ЗА ДОПОМОГОЮ DEEP LEARNING .....</b>	<b>12</b>
1.1. Сутність та завдання комп'ютерного зору в контексті розпізнавання об'єктів .....	12
1.2. Методи ідентифікації та класифікації об'єктів для задач комп'ютерного зору.....	13
1.2.1 Традиційні методи.....	13
1.2.2 Згорткові нейронні мережі .....	13
1.2.3 You only look once.....	14
1.2.4. Зореві трансформери .....	15
1.2.5 Оптичне розпізнавання тексту.....	16
1.3. Огляд наявних рішень для розпізнавання номерних знаків .....	16
1.3.1 Автоматичне розпізнавання номерних знаків на основі YOLOv8 та EasyOCR .....	17
1.3.2 Автоматичне розпізнавання номерних знаків на основі R-CNN.....	18
1.3.3 Автоматичне розпізнавання номерних знаків на основі YOLOv2 та Fast-YOLO .....	19
<b>РОЗДІЛ 2. МОДЕЛІ ТА МЕТОДИ РОЗПІЗНАВАННЯ НОМЕРНИХ ЗНАКІВ ТА КЛАСИФІКАЦІЇ ТРАНСПОРТНИХ ЗАСОБІВ.....</b>	<b>21</b>
2.1. Особливості розпізнавання номерних знаків з використанням deep learning .....	21
2.2. Методи попередньої обробки даних .....	22
2.3. Проектування методу виявлення та розпізнавання .....	22

<b>РОЗДІЛ 3. РОЗРОБКА ТЕХНОЛОГІЙ ВИЯВЛЕННЯ ТА РОЗПІЗНАВАННЯ .....</b>	<b>25</b>
3.1. Збір та підготовка даних .....	25
3.2. Програмна реалізація технології розпізнавання .....	26
3.2.1 Розробка моделі комп’ютерного зору на основі YOLO .....	26
3.2.2 Програмна реалізація розпізнавання тексту.....	29
3.3. Технологія використання та оцінка отриманої моделі комп’ютерного зору.....	32
3.3.1 Аналіз результатів розпізнавання номерних знаків.....	34
3.3.2 Аналіз результатів ідентифікації транспортних засобів .....	38
<b>ВИСНОВКИ .....</b>	<b>42</b>
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....</b>	<b>44</b>

## ВСТУП

Сучасний світ можна схарактеризувати у багато способів, але один з найвиразніших – це високий ступінь механізації, особливо в розвинутих країнах. Хоча стрімке зростання кількості автомобільних транспортних засобів є помітним лише в країнах, що розвиваються, розвинуті країни вже мають велику кількість у використанні. Автомобілі вимагають велику кількість інфраструктури та протягом всієї історії використання автомобіля інженери й науковці шукають як покращити безпеку та ефективність всієї системи. В цьому контексті однією з ключових задач є розпізнавання номерних знаків та ідентифікація типів транспортних засобів. Вона покриває різні сфери використання: збір плати за проїзд, контроль паркування, контроль дозволених транспортних засобів на окремих шляхах, оптимізація логістичних процесів тощо, що зумовлює **актуальність** обраної теми.

Традиційно для виконання таких задач використовувались живі оператори, що вручну спостерігали за трафіком напряму, наприклад охорона при в'їзді до логістичного центру, або через камери відео спостереження, наприклад службовці, що спостерігають за навантаженою ділянкою автомагістралі. Це був найпростіший та економічно виправданий метод, оскільки класичні методи комп'ютерного зору не ефективні в умовах реального світу за межами контролюваних виробничих приміщень. Ці методи засновані на ручному програмуванні ознак, що є не ефективним враховуючи велику кількість різноманітних моделей транспортних засобів та змінні умови відео- та фотознімання. В цьому контексті сучасні методи глибокого навчання (Deep Learning), зокрема нейронні мережі, відкривають нові можливості для створення моделей для комп'ютерного зору, що може адаптуватись до змінних умов реального світу та давати надійну оцінку транспортним засобам, навіть якщо їх немає в базі даних.

Дана робота розглядає теоретичну основу комп'ютерного бачення загалом та з використанням Deep Learning конкретно. Також вона надає приклад того як така система може бути реалізована з погляду програмного забезпечення.

Для програмної реалізації даної системи буде використано модель комп'ютерного бачення для виявлення сімейства YOLOv11 від компанії Ultralitics та бібліотека для мови програмування Python EasyOCR для розпізнавання тексту на номерних знаках. Пріоритетом є створення системи, що здатна на роботу в реальному часі.

**Мета і завдання дослідження.** Метою даного дослідження є: розробка моделі комп'ютерного зору для ідентифікації транспортних засобів та системи для розпізнавання номерних знаків. Для досягнення поставленої мети необхідно було вирішити наступні **завдання**:

- провести дослідження проблематики виявлення ТЗ та розпізнавання номерних знаків з використанням deep learning;
- дослідити наявні інтелектуальні методи виявлення та розпізнавання зображень;
- розробити модель для виявлення та класифікації транспортних засобів на зображені;
- розробити технологію для розпізнавання вмісту номерних знаків;
- провести оцінку якості розробленої моделі комп'ютерного зору.

**Об'єкт дослідження:** процеси розпізнавання та обробки зображень транспортних засобів з використанням методів deep learning.

**Предмет дослідження:** моделі, методи та інформаційні технології в системі для розпізнавання номерних знаків та ідентифікація типів транспортних засобів.

**Інформаційна база дослідження:** навчальні посібники, підручники, наукові статті, матеріалами конференцій, ресурси глобальної мережі Інтернет за тематикою комп'ютерного зору.

**Методи дослідження:** Для практичного розв'язання поставлених задач було використано такі методи:

- загальнонауковий аналітичний метод (розділ 1 та 2);
- методи збору та анотації вибірок даних;
- методи deep learning.

**Практичне значення.** Отримані результати дослідження можуть бути корисними для розв'язання задач управління транспортним потоком, відстежування транспортних засобів, забезпечення безпеки через ідентифікацію.

**Апробаційні результати дослідження.** Студентська науково-практична конференція «Інформаційні технології та кібербезпека в умовах воєнного часу».

**Публікації.** Матеріали студентської науково-практичної конференції.

**Структура та обсяг кваліфікаційної роботи.** Кваліфікаційна робота складається зі вступу, трьох розділів, висновків, списку використаних джерел із 21 найменувань, додатки, містить 32 сторінок основного тексту, 21 рисунків.

## **РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ РОЗПІЗНАВАННЯ ОБ'ЄКТІВ ЗА ДОПОМОГОЮ DEEP LEARNING**

### **1.1. Сутність та завдання комп'ютерного зору в контексті розпізнавання об'єктів**

Комп'ютерний зір – це галузь штучного інтелекту, що використовує машинне навчання та нейроні мережі з deep learning (далі *глибоке навчання*) і має на меті навчити комп'ютери та інші системи отримувати значущу інформацію з цифрових зображень, відео та інших візуальних джерел. [1] Список задач які вирішує комп'ютерний зір включає виявлення, розпізнавання, класифікація об'єктів та семантичне розуміння контексту. Ці задачі є широким узагальненням практичних задач, що потребують вирішення в різних сферах людської діяльності. В контексті даної роботи нас цікавить виявлення, розпізнавання та класифікація об'єктів, а саме номерних знаків та транспортних засобів.

Як зазначено в роботі [2] ці задачі комп'ютерного зору можна розділити на окремі підзадачі для кращого розуміння. Задачу виявлення можна розділити на виявлення окремих транспортних засобів та номерних знаків. Це є фундаментальною задачею на яку покладаються наступні задачі. Для класифікації транспортних засобів, нам спочатку треба виявити чи вони присутні на зображені в першу чергу або виокремити їх, якщо на зображені їх кілька, щоб класифікувати їх окремо. Аналогічно для розпізнавання номерних знаків треба спочатку виявити де вони перебувають на зображені, а потім використати методи оптичного розпізнавання тексту для того, щоб отримати значущу інформацію у вигляді тексту і спеціальних знаків. Наприклад на німецьких номерних знаках окрім літер та чисел використовуються спеціальні реєстраційні наліпки, що відрізняються між регіонами.

## **1.2. Методи ідентифікації та класифікації об'єктів для задач комп’ютерного зору**

Загадана вже робота [2], а також в роботах [3-6] надають глибокий огляд традиційних методів, згорткових нейронних мереж (*від англ.* Convolutional Neural Networks, CNNs; далі ЗНМ), you only look once (далі YOLO) та зорових трансформерів (*від англ.* Vision Transformers, ViTs; далі ЗТ).

### *1.2.1 Традиційні методи*

Традиційні методи спирались на визначені вручну ознаки та правила для виявлення об'єктів. Визначались такими методами як Scale-Invariant Feature Transform, Histogram of Oriented Gradients та Speeded-Up Robust Features. Хоча ці методи відрізняються в деталях, однак вони всі базуються на алгоритмічній обробці зображень та статистичному аналізі кольорової інформації, градієнтів та контрасту. Вони виділяють ключові точки та векторні ознаки, які використовуються як вхідні дані у класичних алгоритмах машинного навчання як SVM, k-NN та Random forest для розпізнавання та класифікації.

Хоча у цій сфері був значний прогрес протягом років, вони погано масштабуються для роботи в складних умовах: вразливі до шумів, змін в ракурсі, освітлені чи масштабі. Отже, вони не дуже добре підходять для виконання задачі ідентифікації ТЗ та розпізнавання номерних знаків.

### *1.2.2 Згорткові нейронні мережі*

ЗНМ є головною технологією сучасного комп’ютерного зору. Вона заснована на глибокому навченні, а її найбільша перевага над традиційними методами є здатність автоматично визначати релевантні ознаки на вхідних зображеннях. Це досягається використанням згорткових фільтрів, що дозволяють виділяти локальні закономірності, та шарів підвибірки, що робить модель більш стійкою до малої варіативності. Згорткові фільтри

використовуються для аналізу контурів, текстури, форми й патернів. ЗНМ є стійкими до зсувів, зміни масштабу і незначних трансформацій.

Популярні ЗНМ як ResNet демонструють високу точність у задачах класифікації, сегментації та виявлення. Однак ці моделі не дуже добре працюють з дрібними деталями як текст, оскільки на шарі підвибірки скорочується роздільність і втрачається суттєва інформація. Отже, ЗНМ добре підходять для задачі ідентифікації ТЗ, але потенційно можливі проблеми з виявленням номерних знаків через розмір об'єкта на більшому об'єкті.

### *1.2.3 You only look once*

YOLO є популярною архітектурою для задач виявлення об'єктів у реальному часі. Висока швидкість забезпечується тим, що виявлення та класифікація відбувається в один етап, коли як інші моделі зазвичай спочатку виявляють присутність якихось об'єктів, а лише потім намагається класифікувати їх. YOLO ділить зображення на сітку  $S \times S$  і на основі вмісту всього зображення намагається передбачити  $B$  рамок в кожній рамці та  $C$  класів. На виході отримуємо тензор передбачень. За основу на ранніх шарах використовується типова архітектура ЗНМ, а тому YOLO властиві схожі недоліки.

Через те сітку та функцію втрат, що надає пріоритет максимізації перетину передбаченої рамки та істини, YOLO має проблеми з виявленням малих об'єктів, особливо якщо вони близько один до одного. Однак YOLOv3 покращив архітектуру, що процес виявлення та класифікації відбувається кілька разів на різному масштабі, що значно покращило точність виявлення об'єктів різного розміру. Елементи трансформерів, що почали інтегрувати у версії 9 дозволили ще більше покращити загальну точність та глобальне розуміння вмісту дозволило значно покращити точність в складних умовах.

Загалом YOLO від 1-ї до поточної 11-ї версії, як описано в роботі [7], зазнало багато змін, що значно покращило точність моделі зберігши її

швидкість. Хоча робота [7] більше сфокусована на використанні YOLO в текстильній промисловості, покращення якісних показників повинні спостерігатись і в інших галузях. На сьогодні ця модель надає найкращий баланс між точністю та продуктивністю. Це робить цю модель чудовою для задачі виявлення та класифікації ТЗ і номерних знаків, однак все ще необхідно використовувати окрему технологію для розпізнавання вмісту номерних знаків.

#### *1.2.4. Зореві трансформери*

ЗР є новітнім підходом до розв'язання задач комп'ютерного зору. Вони засновані на технології трансформерів, які засновані на глибокому навчанні та були початково використані у галузі обробки натуральної мови. Недавно вони стали рушієм створення великих мовних моделей. На відміну від ЗНМ та YOLO, трансформери працюють з усім зображенням одночасно без змін у вигляді багатовимірних векторів. Цей підхід дозволяє моделі отримати контекст за ознаки напряму з зображення, а не з глибоко обробленого зображення з концентрованими даними. Завдяки цьому трансформери є дуже стійкими до шуму, перешкод та трансформаціям (різні кути та перспективи), а також здатні до високого рівня узагальнення.

Однак ціною високої точності, стійкості та узагальнення є дуже високі вимоги до пам'яті та розрахункових ресурсів. Точність значно зростає при використанні попередньої обробки зображень, що конфліктує з головною ідеєю про роботу з усім зображенням та контекстом. Також для найкращого результату модель потребує тонкого налаштування, що не є проблемою в контексті цієї роботи, оскільки нам не потрібно максимальне узагальнення, однак таким чином ускладнюється задача створення моделі для задачі виявлення та класифікації ТЗ та номерних знаків. А високі вимоги роблять їх не практичними для роботи в реальному часі.

В згаданих роботах [6-7] вказується, що комбінація ЗТ та ЗНМ дозволяє отримати кращі результати ніж чисті моделі покращуючи точність в

складних умовах зберігаючи продуктивність. Моделі YOLOv9 та новіше використовують

#### *1.2.5 Оптичне розпізнавання тексту*

Однією з сучасних бібліотек оптичного розпізнавання тексту для середовища Python є EasyOCR. Ця технологія створення виключно для розпізнавання тексту, а отже її треба використовувати у комбінації з іншими методами для виявлення та класифікації. Вона заснована на PyTorch, популярна бібліотека для глибокого навчання мовою Python, та підтримує багато природних мов, однак для нашої задачі це не має такого великого значення.

У роботах [8-9] продемонстровано, що EasyOCR здатна на високоточні результати навіть в не ідеальних умовах зовнішнього світу. В роботі [9] говориться, що TesseractOCR загалом має кращу точність та продуктивність, однак це спостерігається в першу чергу у відносно ідеальних умовах сканкопій друкованої літератури.

Існують більш новітні технології для оптичного розпізнавання тексту на основі ЗР як продемонстровано у роботі [10]. Вони мають високу точність роботи, однак вони працюють на рівні слів, а не окремих символів, що робить їх не придатними для мети цієї роботи.

### **1.3. Огляд наявних рішень для розпізнавання номерних знаків**

На тему автоматизованого розпізнавання номерних знаків було написано багато робіт. Це зумовлено актуальністю проблем які намагаються розв'язати ці технології. Існує багато способів реалізувати дану технологію, а також існує велика варіативність у номерних знаках між країнами та навіть окремими регіонами всередині цих країн, а тому є необхідність в спеціалізованих моделях. Для огляду було обрано три роботи на дану тему.

### *1.3.1 Автоматичне розпізнавання номерних знаків на основі YOLOv8 та EasyOCR*

У роботі [11] запропоновано метод на основі YOLOv8 та EasyOCR розпізнавання номерних знаків з високою точністю, однак вони не розпізнають вміст номерного знаку, а лише країну походження на основі маркування, що присутнє на самих знаках. Розпізнавання відбувається у кілька кроків:

1. Виявлення номерного знаку з використанням YOLOv8, що було натреновано на власній вибірці;
2. Масштабування зображення до розміру номерного знаку;
3. Попередня обробка зображення перед розпізнаванням тексту:
  - a. Кластеризація пікселів методом K-means;
  - b. Бінаризація зображення на основі кластерів;
  - c. Морфологічні операції для покращення тексту.
4. Використання EasyOCR для розпізнавання текстових позначок країн для класифікації.

Результати оцінки показали показник  $F_1$  99% для виявлення номерних знаків та точність 99.95% для розпізнавання тексту, однак вони не вказали при якому порозі впевненості чи перетину IoU. Також оськільки вони розпізнають тільки по кілька літер за раз, бо позначки країн в Європі зазвичай всього 1-2 літери та кілька виключень де 3, що означає меншу ймовірність для неправильного розпізнавання. Не було зазначено як швидко працює метод, однак, враховуючи кластеризацію для бінаризації, показник кадрів в секунду (FPS) не дуже високий, бо K-means не є дуже швидким алгоритмом, як буде доведено експериментальним шляхом.

Авторами роботи [11] було приділено багато уваги вибірці даних. Було зазначено, що вибірка складалась з 270 зображень з різних країн. Номерні знаки було видно з різних ракурсів та з різним освітленням. Було приділено багато уваги тому, щоб був високий рівень згоди між анотаторами. Вони повинні погоджуватись на тому, що належати до класу об'єкта, як саме

обирається обмежувальна рамка та в яких випадках можна проігнорувати об'єкт, наприклад номерний знак є замалим щоб на ньому можна було розпізнати вміст.

### *1.3.2 Автоматичне розпізнавання номерних знаків на основі R-CNN*

У роботі [12] запропоновано методологію на основі ЗНМ. Модель була створена для розпізнавання китайських номерних знаків у довільних умовах. Вибірка складається з 18699 зображень з різноманітним фоном та роздільній здатності.

Метод складається з двох основних етапів: виявлення та розпізнання. Виявлення починається з «*regional proposal network*», що відповідає за визначення загального положення об'єктів інтересу. Потім використовується R-CNN для класифікації об'єктів у регіоні зображення, що була запропонована попередньою мережею, а також визначення чотирьох кутів номерного знаку, що потім використовується для виправлення перспективи. Це є великою відмінністю від YOLO, що тільки пропонує обмежувальну рамку.

Коли було повністю виявлено номерний знак, починається процес розпізнавання. Першим кроком є «*spacial transformation network*», що відповідає за сегментацію зображення на окремі символи. Кожен символ направляється до класифікатора, що вже розпізнає їх.

Результати оцінки показали точність виявлення 0,7813 як середнє при IoU@0.5-0.9, де IoU це перетин між передбаченим сегментом та істиною. Цей показник був найвищим серед інших методів на основі ЗНМ з якими автори порівнювали власну методологію. Точність розпізнавання вмісту номерних знаків складала 0,89, однак якщо дозволити один невірно розпізнаний символ, то точність складає 0,9755, що говорить про простір для покращення метода корекції помилок. Показник FPS склав приблизно 11 на Nvidia K40 [13], що є професійним графічним прискорювачем з 2013 року і

був дуже дорогим. Логічно припустити, що ці показники будуть значно кращі на сучасному професійному та споживацькому обладнанні.

### *1.3.3 Автоматичне розпізнавання номерних знаків на основі YOLOv2 та Fast-YOLO*

У роботі [14] запропоновано методологію з трьох етапів: виявлення номерних знаків, сегментація та розпізнавання символів. Моделі YOLO використовуються для виявлення, а згорткові нейронні мережі для сегментації та розпізнавання. Для тренування та оцінки використовуються вибірки SSIG та UFPR-ALPR номерних знаків з Бразилії. Перша вибірка складається з досить схожих номерних знаків зі статичних камер та з простими фонами. Друга вибірка була створена авторами роботи [13] та складається з різноманітних транспортних засобів, що були зняті з різних ракурсів з рухомих камер та зі складними фонами. Вони використовуються для порівняння двох різновидностей YOLO.

Для етапу виявлення було використано YOLOv2 та Fast-YOLO. Автори очікували, що YOLOv2 покаже кращі результати на вибірці UFPR-ALPR, а Fast-YOLO – на SSIG. На вибірці SSIG Fast-YOLO показало повноту 100% та точність вище 99% під час виявленні транспортних засобах та 100% повноту і точність для номерних знаках. Однак треба мати на увазі, що ця вибірка досить проста та не зовсім показова. На вибірці UFPR-ALPR Fast-YOLO мало повноту 97.33%, а YOLOv2 99%. Обидва результати досить високі, але від їх якості залежить якість розпізнавання на наступних етапах.

Для етапів сегментації та розпізнавання символів використовується згорткова нейронна мережа, що дає можливість адаптувати методологію під конкретну країну та її стандарти. Також, якщо знати формат номерних знаків та проводити розпізнавання на відео, то можна підвищити точність використовуючи інформацію з кількох кадрів методом голосування. Наприклад якщо на 1 позиції знаходиться літера «A» на трьох з чотирьох кадрів, то з високою ймовірністю там буде літера «A». На вибірці SSIG було

отримано повноту 99.75% під час сегментації та точність розпізнання 85.45%, та 93.53%, якщо використовувати інформацію з кількох кадрів. На вибірці UFPR-ALPR під час сегментації було отримано повноту 95,97%, або 97,59%, якщо не враховувати не впізнані номерні знаки. Точність розпізнання символів склала 64,89% та 78,33% при індивідуальній та кількісній обробці кадрів. Автори зазначили, що при аналізі результатів стало очевидним, що найбільшою проблемою стало розпізнання номерних знаків на мотоциклах.

Показники FPS склали 47 та 35 для Fast-YOLO на SSIG та YOLOv2 на UFPR-ALPR відповідно. Експерименти проводились на графічному прискорювачі Nvidia Titan Xp [15], що аналогічно до вже загаданої K40 є професійним пристроєм, однак на значно новішій архітектурі та за нижчою ціною, що не напряму підтверджує моє припущення, що показники методу з роботи [12] повинні бути кращі на новішому обладнанні, однак складно сказати наскільки без прямих вимірювань. Показники у 35-47 FPS можна вважати достатніми для роботи в реальному часі, що більше пасує для спостереження над дорожнім рухом.

## **РОЗДІЛ 2. МОДЕЛІ ТА МЕТОДИ РОЗПІЗНАВАННЯ НОМЕРНИХ ЗНАКІВ ТА КЛАСИФІКАЦІЇ ТРАНСПОРТНИХ ЗАСОБІВ**

### **2.1. Особливості розпізнавання номерних знаків з використанням deep learning**

Номерні знаки – це об'єкти з простою геометрією та дизайном. Хоча існують значні винятки як країни Північної Америки, де в кожній адміністративній одиниці існують власні стандарти та широкий вибір графічного оформлення, що значно ускладнює розпізнання тексту, більшість країн мають уніфікований дизайн. У більшості випадків це прямокутна пластина з фоном одного кольору без малюнків чи візерунків, текстом чорного чи білого кольору та спеціальними позначками для ідентифікації країни походження чи регіону, тож вони досить однотипні. Вони встановлюються у спеціально визначені для цього місця на транспортному засобі, що робить їх досить виразними та відличними. Їх складно сплутати з чимось іншим в контексті.

Проте, як показало теоретичне дослідження у розділі 1, через хаотичність реального світу та відсутність контролю над навколоишнім середовищем у більшості випадків, використання традиційних методів не є доцільним, якщо взагалі можливим. Отже, використання технологій з глибоким навчанням є єдиним доцільним рішенням з розглянутих. Актуальні роботи [11, 12, 14], що були обговорені у підрозділі 1.3, використовують технології глибокого навчання, а деякі автори також наголошували, що традиційні методи не підходять для даної задачі.

Сучасні технології комп'ютерного зору на основі глибокого навчання роблять цю задачу досить тривіальною. Ключовим викликом стає збір якісної вибірки зображень для тренування моделі для ідентифікації. Вибірка повинна бути репрезентативною для кожного класу, показувати об'єкт під різними кутами та в різних умовах для забезпечення найкращої роботи в складних умовах.

## **2.2. Методи попередньої обробки даних**

Одними з найбільш поширеніх методів попередньої обробки зображень є кольорова корекція, нормалізація контрасту та яскравості, прибирання шумів та зміна розміру. Вони індивідуально не значні, але при правильній комбінації можуть суттєво змінити вигляд та вміст зображення. Вони були ключовими для традиційних методів комп'ютерного зору.

Проте, як показано в роботі [16], що проводить огляд різноманітних методологій з розпізнавання на основі глибокого навчання, вплив попередньої обробки не є однозначним. Якщо модель вже показує високі показники, то додаткова обробка не дасть значних покращень, якщо взагалі дасть як це показано у випадку з YOLOv3. Присутні приклади методологій, що не використовують попередню обробку у будь-якому вигляді та отримують високі показники.

Автором зазначається, що коли стоять вибір між використанням та невикористанням попередньої обробки перед розробниками, то варто визначити пріоритети. Вона може значно покращити результати розпізнавання, однак варто брати до уваги вартість реалізації та її складність. Якщо стоять задача зробити методологію якнайшвидшою, то варто обмежити використання цих методів.

## **2.3. Проектування методу виявлення та розпізнавання**

Методологія повинна виконувати ідентифікацію транспортних засобів та розпізнавання номерних знаків. В цій методології пропоную виокремити такі кроки:

1. Виявлення та класифікація транспортних засобів;
2. Розпізнавання номерних знаків;

Для реалізації даної методології доцільно використати модель архітектури YOLO, оскільки вона виконує виявлення та класифікацію в один крок, що і є необхідним. Хоча для задачі виявлення та класифікації транспортних засобів та класифікації використовується одна модель, для

кращого розуміння, було логічно розділено крок ідентифікації транспортних засобів та розпізнавання номерних знаків. Більше детально процес розпізнавання номерних знаків можна побачити на рис. 1.

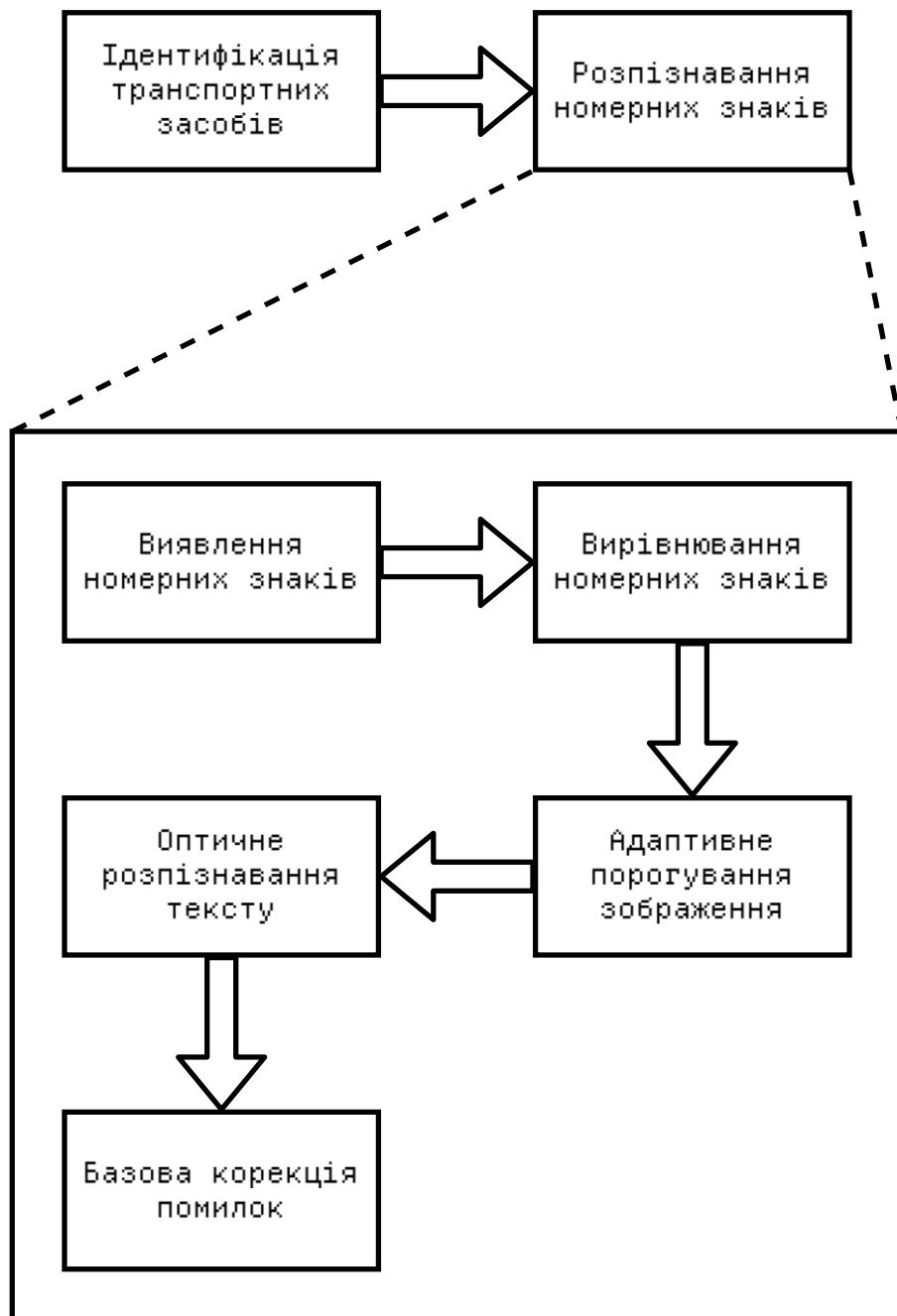


Рис. 1. Схематичне зображення кроків методології

Проаналізовані роботи в даній галузі використовують комплексні методи для розпізнавання вмісту номерних знаків, що містити розробку та використання методів сегментації та класифікації текстових символів.

Використання готових методів з відкритого доступу для розпізнавання тексту спрощує імплементацію без серйозних втрат точності.

Перед застосуванням методу розпізнавання символів пропонується зробити просту попередню обробку зображення. Ключовим є використання методу з вирівнювання тексту, оскільки текст вразливий до не прямих кутів. Однак використовуючи YOLO відсутня інформація про точні кути номерного знаку як це було в роботі [12], а тому рівень вирівнювання обмежений лише однією віссю. Для того, щоб зробити текст чіткішим використовується адаптивний алгоритм порогування, який автоматично визначає оптимальний поріг.

Такий підхід оптимізує баланс між складністю розробки та ефективністю кінцевого рішення, забезпечуючи при цьому достатню надійність для практичного застосування.

## РОЗДІЛ 3. РОЗРОБКА ТЕХНОЛОГІЙ ВИЯВЛЕННЯ ТА РОЗПІЗНАВАННЯ

### 3.1. Збір та підготовка даних

Оскільки технологія створюється для розпізнавання українських номерних знаків, то увага повинна бути зосереджена на зборі зображень з українськими номерними знаками. Логічним джерелом для подібних зображень є публічні дорожні камери відеоспостереження, однак, в умовах воєнного стану, доступ до них є обмеженим. Іншим загальнодоступним джерелом є онлайн платформи для продажу вживаних авто, наприклад <https://auto.ria.com/uk/> [17]. На подібних сайтах можна знайти широкий вибір транспортних засобів різних моделей з номерними знаками, що зображені з різних кутів.

Для кожного зображення треба створити анотації об'єктів. Кожен рядок в текстовому файлі анотації включає таку інформацію: номер класу, центр по осі  $x$ , центр по осі  $y$ , ширина, висота. Координати центру, ширина та висота задані у вигляді нормалізованих значень від 0 до 1, для того щоб роздільна здатність зображення не мала значення.

Для задачі класифікації було визначено такі класи: автомобіль (car), вантажівка (truck), мотоцикл (motorbike) та номерний знак (licence\_plate). Було зібрано вибірку з 467 зображень. На цих зображеннях було анотовано 420 номерних знаків, 287 автомобілів, 175 вантажівок та 57 мотоциклів. Треба зазначити, що до категорії вантажівок було віднесено пікапи, мінівени, фургони та звичайні вантажівки. До мотоциклів були віднесені двоколісні та триколісні мотоцикли.

В ідеальних умовах кожен вид транспортного засобу повинен бути репрезентований порівну. Однак під час збору зображень було прийнято рішення приділити увагу відповідно до кількості оголошень на момент збору. Приклад отриманих зображень з їх анотацією можна побачити на рис. 2.



Рис. 2. Наочний приклад зображень з вибірки

### 3.2. Програмна реалізація технології розпізнавання

Для реалізації було обрано мову програмування Python 3.12.6, оскільки саме для неї було створено програмні бібліотеки для створення моделей на основі YOLO від компанії Ultralitics, а також бібліотека EasyOCR для розпізнавання тексту.

#### 3.2.1 Розробка моделі комп’ютерного зору на основі YOLO

З програмної сторони реалізувати власну модель на основі YOLO досить просто. Компанія Ultralitics створила простий у використані

програмний інтерфейс у своїй бібліотеці для Python. Весь код, що необхідний для початку процесу тренування можна побачити на рис. 3.

```
1 from ultralytics import YOLO
2
3 model = YOLO("yolo11n.yaml")      # побудова моделі з нуля
4
5 results = model.train(
6     data='data.yaml',             # шлях до конфігураційного файлу з описом вибірки
7     imgsz=640,                   # розмір вхідних зображень
8     patience=20,                 # кількість епох для ранньої зупинки
9     cos_lr=True,                # використання косинусного затухання швидкості
10    nавчання
11    augment=True,               # використання аугментацій
12    device=0,                   # використання GPU з індексом 0
13    project='vehicle_detection', # назва проекту
14    name='train_nano'           # назва експерименту
15)
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
```

Python

Рис. 3. Код для тренування моделі YOLO

Як основу було використано конфігурацію для YOLO версії 11 розміру Нано. Розмір впливає на кількість параметрів, розмір моделі в пам'яті, точність та швидкість. Нано є найменшим розміром і було обрано для забезпечення найвищої швидкості.

Замовченням модель тренується 100 епох. Розмір зображення 640 є стандартним та задається у пікселях. Якщо зображення не є квадратним чи більшим за розміром, то воно буде автоматично стисненим. Такі маніпуляції призводять до втрати інформації, але оскільки задача не стоїть розпізнавати малі деталі, то не очікується критичний вплив.

Рання зупинка встановлена на 20 епох, а також включено затухання швидкості навчання. Ця конфігурація налаштована на забезпечення швидшої та кращої конвергенції, точки в якій подальша зміна параметрів не забезпечує значуще покращення результатів. Затухання означає, що на ранніх епохах одночасно можливі великі зміни до параметрів та високий хаос, коли на пізніших дозволені тільки менші зміни, що робить модель більш стабільною.

Аугментація це процес, що дозволяє штучно розширити розмір вибірки. Зображення з вибірки випадковим чином обрізають та відзеркалюють й створюють мозаїку, що дозволяє кілька разів використати ту саму інформацію в іншому вигляді без значного негативного впливу на перенавчання чи точність. Як виглядає така мозаїка можна побачити на рис. 4.

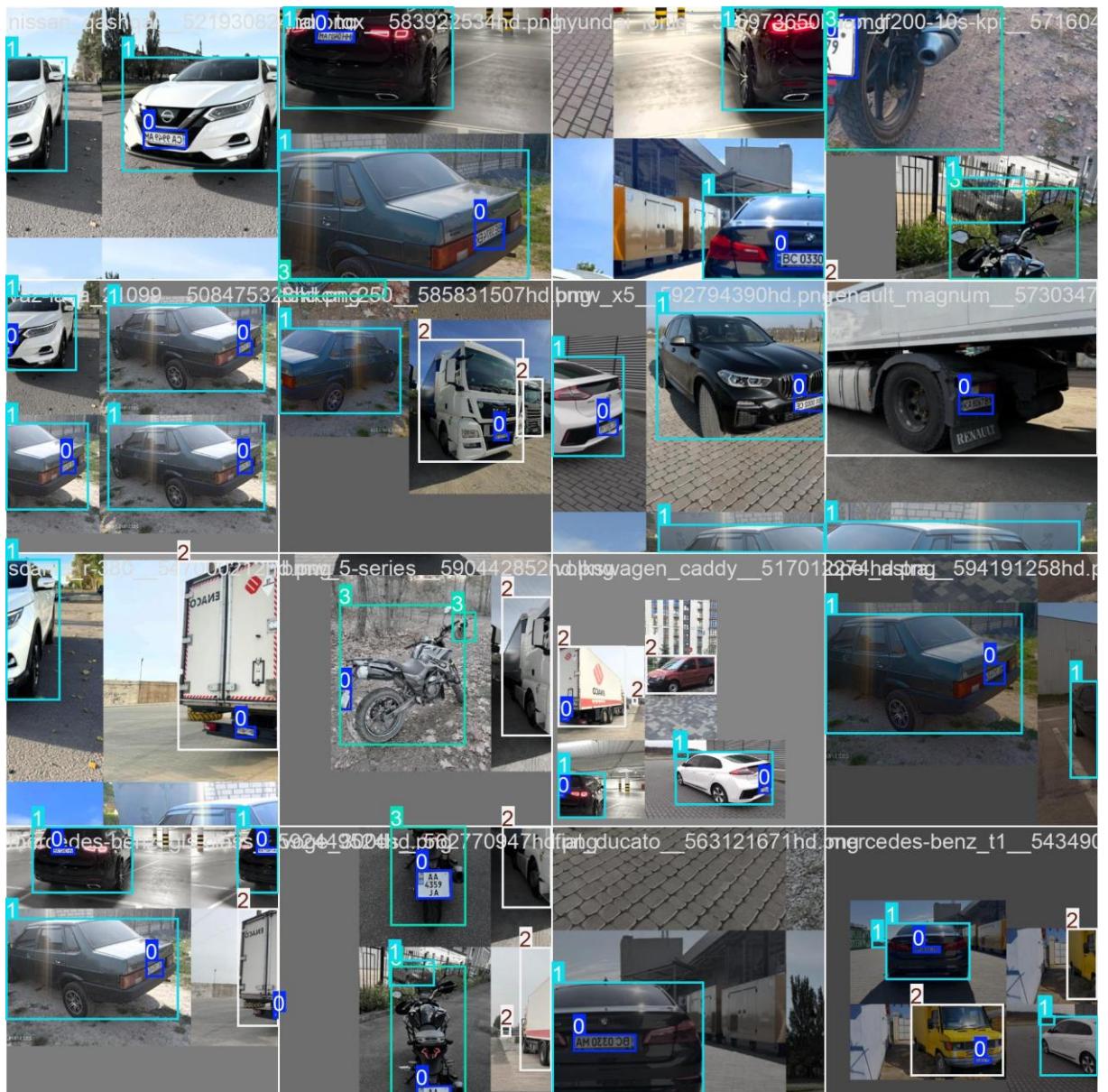


Рис. 4. Мозаїка аугментованих зображень, що використовуються для тренування

Для застосування моделі для ідентифікації об'єктів на зображені застосовується функція на рис. 5. Як аргумент для зображення використовується об'єкт зображення з бібліотеки OpenCV, що також потрібна для методів попередньої обробки у розділі 3.2.2. [18].

```
def predict_objects(image, model, confidence_threshold):
    results = model(image)[0]

    output = []
    for pred_object in results.bboxes.data.tolist():
        x1, y1, x2, y2, score, class_id = pred_object
        if score <= confidence_threshold: continue
        output.append({
            'box': [x1, y1, x2, y2],
            'score': score,
            'class_id': results.names[int(class_id)]})
    return output
```

Рис. 5. Функція для ідентифікації об'єктів

### 3.2.2 Програмна реалізація розпізнавання тексту

Для найкращого результату розпізнавання тексту перед використанням методів EasyOCR, необхідно провести певні маніпуляції над зображенням для найбільшої чіткості, а саме: вирівняти та використати адаптивне порогування. Як вже була загадано в розділі 2.3, в нашій методології немає можливості дізнатись точні координати чотирьох кутів номерного знаку, а тому можливість вирівняти номерний знак є обмеженою.

Для вирівнювання тексту було обрано готовий інструмент [19], оскільки це спрощує реалізацію, а результати роботи є задовільними. Результати вирівнювання можна побачити на рис. 6, а код на рис. 7.

Для адаптивного порогування застосовуються методи з OpenCV та їх результат можна побачити на рис. 6, а код на рис. 8.



Рис. 6. Демонстрація попередньої обробки

```
def deskew_text_validated(image_path):
    """
    Вирівнює зображення з використанням інструменту
    https://github.com/galfar/deskew
    """
    command = [DESKEW, '-o', 'deskewed.png', image_path]

    try:
        _ = subprocess.run(
            command,
            capture_output=True,
            text=True,
            check=True
        )

        deskewed = cv2.imread('deskewed.png')
        return deskewed
    except subprocess.CalledProcessError as e:
        print(f"Error running deskew: {e}")
        return cv2.imread(image_path)
```

Рис. 7. Код для вирівнювання тексту

```

def enhance_license_plate(warped_plate):
    # Перетворює на чорно-біле зображення
    if len(warped_plate.shape) == 3:
        gray = cv2.cvtColor(warped_plate, cv2.COLOR_BGR2GRAY)
    else:
        gray = warped_plate

    # Застосувати адаптивне порогування
    adaptive = cv2.adaptiveThreshold(gray, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C,
                                      cv2.THRESH_BINARY, 15, 5)
    _, otsu_adaptive = cv2.threshold(adaptive, 0, 255, cv2.THRESH_BINARY + cv2.THRESH_OTSU)
    return otsu_adaptive

```

Рис. 8. Код для порогування зображення

Деякі букви та цифри можна досить просто візуально сплутати, наприклад буква **О** дуже схожа на цифру **0**, чи **I** та **1**. За такою логікою можливо реалізувати досить простий алгоритм виправлення помилок, де відповідні символи будуть замінюватись, якщо вони не очікуються в тому сегменті номерного знаку. Код для виправлення помилок та розпізнавання тексту можна побачити на рис. 9.

Ми не можемо завчасно знати чи буде текст розпізнаний як один блок чи в якому порядку будуть виявлені кілька блоків. Для розв'язання цієї проблеми було прийнято рішення перебрати всі можливі комбінації текстових блоків, щоб спробувати виявити текст відповідний формату ДСТУ 4278:2019 [20].

```

def extract_text(image):
    """Проводить розпізнавання тексту"""
    result = OCR_READER.readtext(image, allowlist='ABCDEHIKMMNORTXYZJ0123456789| ', text_threshold = 0.3)

    # Створює всі можливі комбінації
    possible_combinations = [''.join(p) for p in permutations([r[1] for r in result])]

    return possible_combinations

def correct_chars(text):
    POSITION_RULES = {
        'initial_chars': { # Символи 0-1 та 6-7
            '0': '0',
            '1': 'I',
            '8': 'B',
            '2': 'Z',
            '7': 'Z',
            '|': 'I'
        },
        'middle_chars': { # Символи 2-5
            '0': '0',
            'I': '1',
            'B': '8',
            'Z': '7',
            '|': '1'
        }
    }

    corrected = list(text)

    for i, char in enumerate(corrected):
        if i in [0, 1, 6, 7]:
            corrected[i] = POSITION_RULES['initial_chars'].get(char, char)
        elif i in [2, 3, 4, 5]:
            corrected[i] = POSITION_RULES['middle_chars'].get(char, char)

    return ''.join(corrected)

def error_correct_plate(text):
    PATTERN = r'[A-Z01827\|\|]{2}[0-90IBZ\|\|]{4}[A-Z01827\|\|]{2}'

    for p in text:
        if not (match := re.search(PATTERN, p)): continue
        corrected = correct_chars(match.group())

    return (True, corrected)

    if not len(text[0]) == 8: return (False, text[0])
    return (False, correct_chars(text[0]))

```

Рис. 9. Код для розпізнавання тексту та виправлення помилок

### 3.3. Технологія використання та оцінка отриманої моделі комп’ютерного зору

Всі критичні функції для функціонування технології вже були реалізовані та описані у розділі 3.2. Останнім кроком є їх пов’язання в єдину технологію ідентифікації транспортних засобів та розпізнавання номерних знаків. Код реалізації можна побачити на рис. 10.

```
if __name__ == "__main__":
    results = {}
    speed = []

    for image_path in glob.glob(os.path.join(EVAL_DIR, '*.png')):
        start_time = time.time()
        image = cv2.imread(image_path)
        filename = os.path.splitext(os.path.basename(image_path))[0]

        predictions = predict_objects(image, YOLO_MODEL, CONFIDENCE)
        prediction_end_time = time.time()

        for ob in predictions:
            if not ob['class_id'] == 'licence_plate':
                image = draw_frame_with_label(image, ob['box'], ob['class_id'])
                continue
            plate_recognition_start_time = time.time()
            box = ob['box']
            cropped_plate = crop_image_to_box(image, box)

            processed_plate = process_license_plate('./cropped.png')
            preprocess_end_time = time.time()

            text = extract_text(processed_plate)
            ocr_end_time = time.time()

            content = error_correct_plate(text)
            error_correction_end_time = time.time()

            if content[0]:
                print(f"\u041f\u043e\u0437\u043d\u0430\u043b\u043e \u043f\u043e\u043b\u0438\u043d\u0430\u043b\u0438\u0437\u0430\u043d\u0438\u044f \u0434\u043b\u044f \u0434\u043b\u044f \u043f\u043e\u0434\u043b\u043e\u0436\u0435\u043d\u0438\u044f: {content[0]}")
            else:
                print(f"\u041f\u043e\u0437\u043d\u0430\u043b\u043e \u043f\u043e\u043b\u0438\u043d\u0430\u043b\u0438\u0437\u0430\u043d\u0438\u044f \u0434\u043b\u044f \u0434\u043b\u044f \u043f\u043e\u0434\u043b\u043e\u0436\u0435\u043d\u0438\u044f: {content[0]}")

            results[filename] = content
            image = draw_frame_with_label(image, ob['box'], content[1], (255, 255, 0))

        end_time = time.time()
        print(f"Ідентифікація {prediction_end_time - start_time:.4f} сек.")
        print(f"Попередня обробка {preprocess_end_time - plate_recognition_start_time:.4f} сек.")
        print(f"Розпізнавання тексту {ocr_end_time - preprocess_end_time:.4f} сек.")
        print(f"Загалом час на {image_path}: {end_time - start_time:.2f} сек.")
        speed.append(end_time - start_time)
        cv2.imwrite(OUTPUT_DIR + filename + '_labeled.png', image)

    total_time = sum(speed)
    fps = len(speed) / total_time if total_time > 0 else 0
    print(f"Оброблено {len(speed)} зображень за {total_time:.2f} сек. FPS: {fps:.2f}")
    with open('license_plate_results_with_preprocessing.yaml', 'w') as yaml_file:
        yaml.dump(results, yaml_file, default_flow_style=False)
```

Рис. 10. Реалізація використання технології

На виході створюються зображення з анотацією, що виглядають як на Рис. 11.

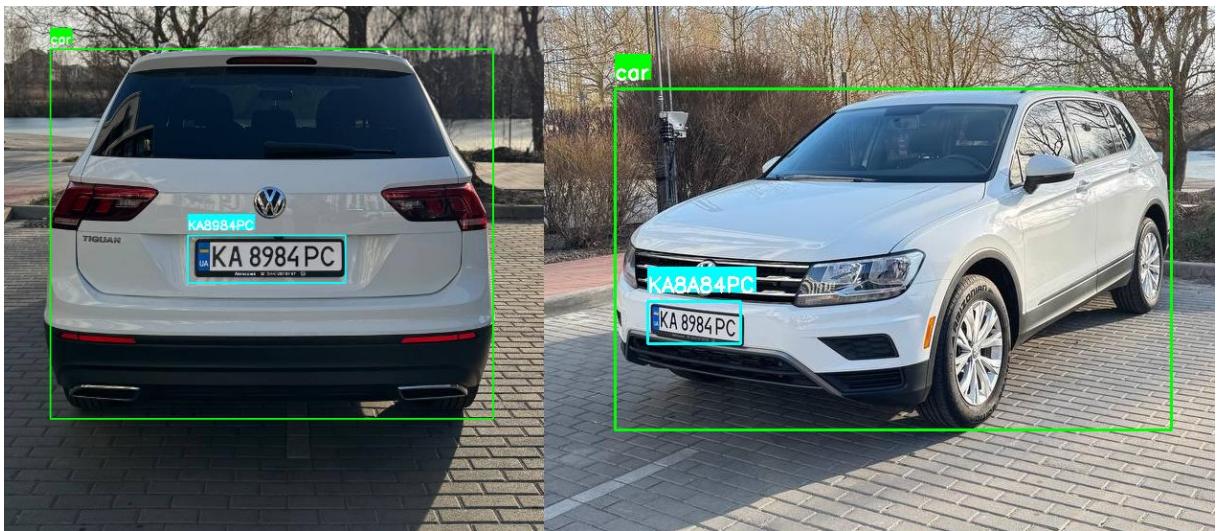


Рис. 11. Демонстрація візуалізації роботи технології

### 3.3.1 Аналіз результатів розпізнавання номерних знаків

На жаль, як можна побачити на демонстрації, EasyOCR не дуже добре справляється з розпізнаванням тексту, якщо номерний знак не зовсім рівно, навіть з використанням базового методу вирівнювання тексту. Результати розпізнавання тексту можна побачити на діаграмах на рис. 12 та 13. Результати було поділено на 3 категорії: правильно, неправильно та пропущено. Правильно це якщо було розпізнано знак відповідно до істини, не правильно якщо відповідає формату, але не істині, пропущено, якщо не відповідає стандарту або пусто.



Рис. 12. Діаграма результатів розпізнавання номерних знаків

## Результати розпізнавання номерних знаків за типом ТЗ

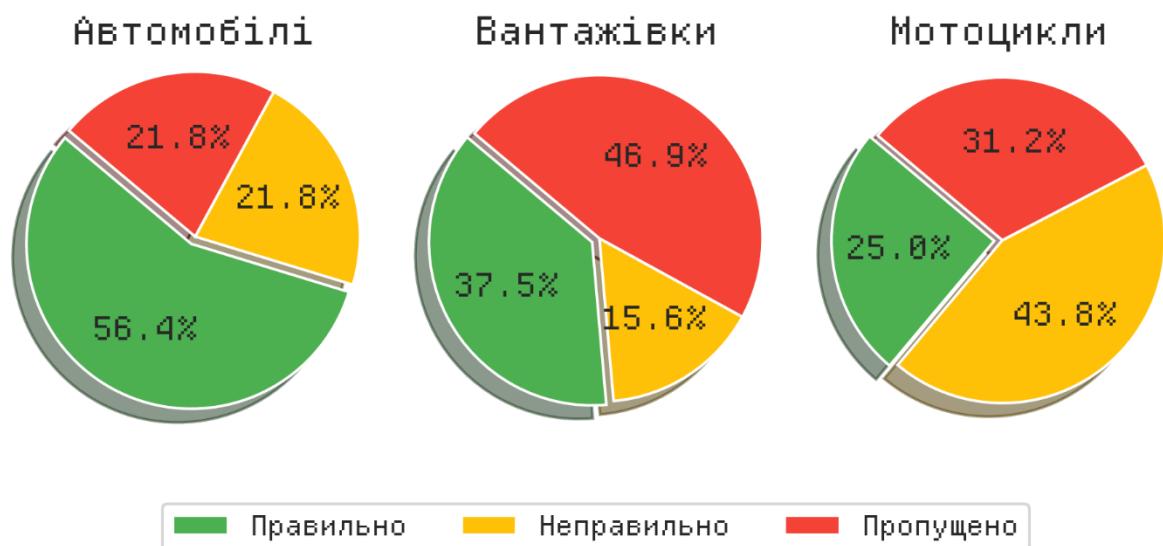


Рис. 13. Діаграма результатів розпізнавання номерних знаків за типом ТЗ

Мотоцикли мають вищу частку неправильних знаків оскільки порядок коду регіону та серії можуть бути сплутані, що призводить до правильного за форматом, але не вмістом знаку. Вантажівки мають вищу частку пропущених, оскільки у вибірці вантажівки часто сфотографовані під нетиповими кутами чи знаки мають перешкоди.



Рис. 14. Розподiл помилок у неправильних номерних знаках

Аналізуючи випадки неправильного розпізнання можна побачити, що у більшості випадків всього один символ є неправильним, як показано на діаграмі на рис. 14.

Якщо розширити аналіз і також включати пропущені номерні знаки, то можна побачити, що більшість випадків не мають нічого спільного з істиною. Однак все ще велику частку складають випадки, де тільки 1-3 помилки. Розподіл можна побачити на діаграмі на рис. 15.



Рис. 15. Розподіл помилок у неправильних та пропущених

Якщо провести аналіз матрицею плутанини на рис. 16 та 17, прибравши всі правильно розпізнані символи для простоти сприйняття, то складно визначити якусь закономірність. Помилки схожі на простий шум. Радіально симетричні ділянки низької щільноті пов'язані з алгоритмом виправлення помилок. Також, оскільки не застосовується складна методологія сегментації тексту, відсутня можливість провести комплексний аналіз з врахуванням пропущених символів. Тому у матриці аналізуються тільки ті випадки коли було розпізнано 8 символів.

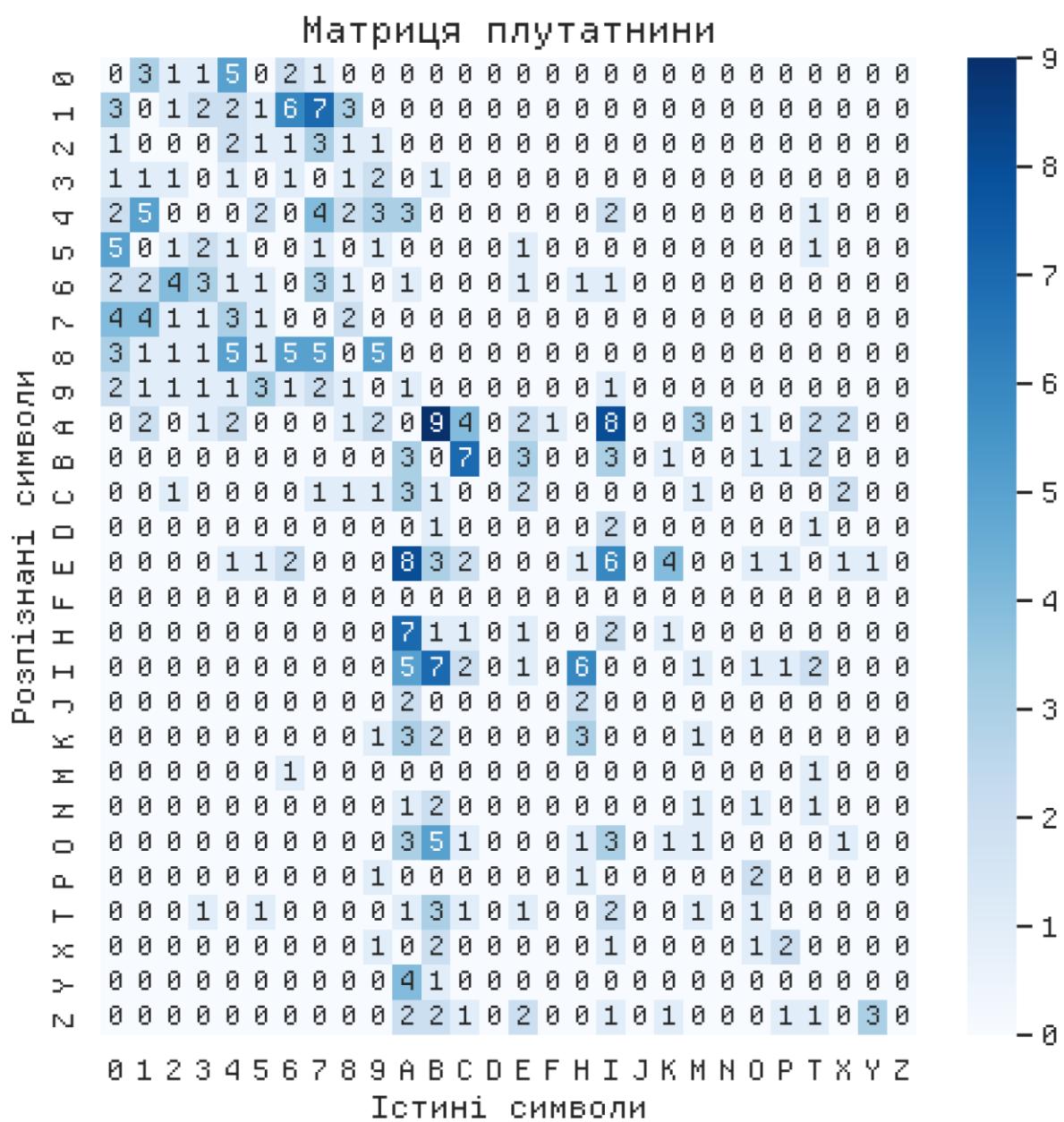


Рис. 16 Матриця плутанини розпізнання символів

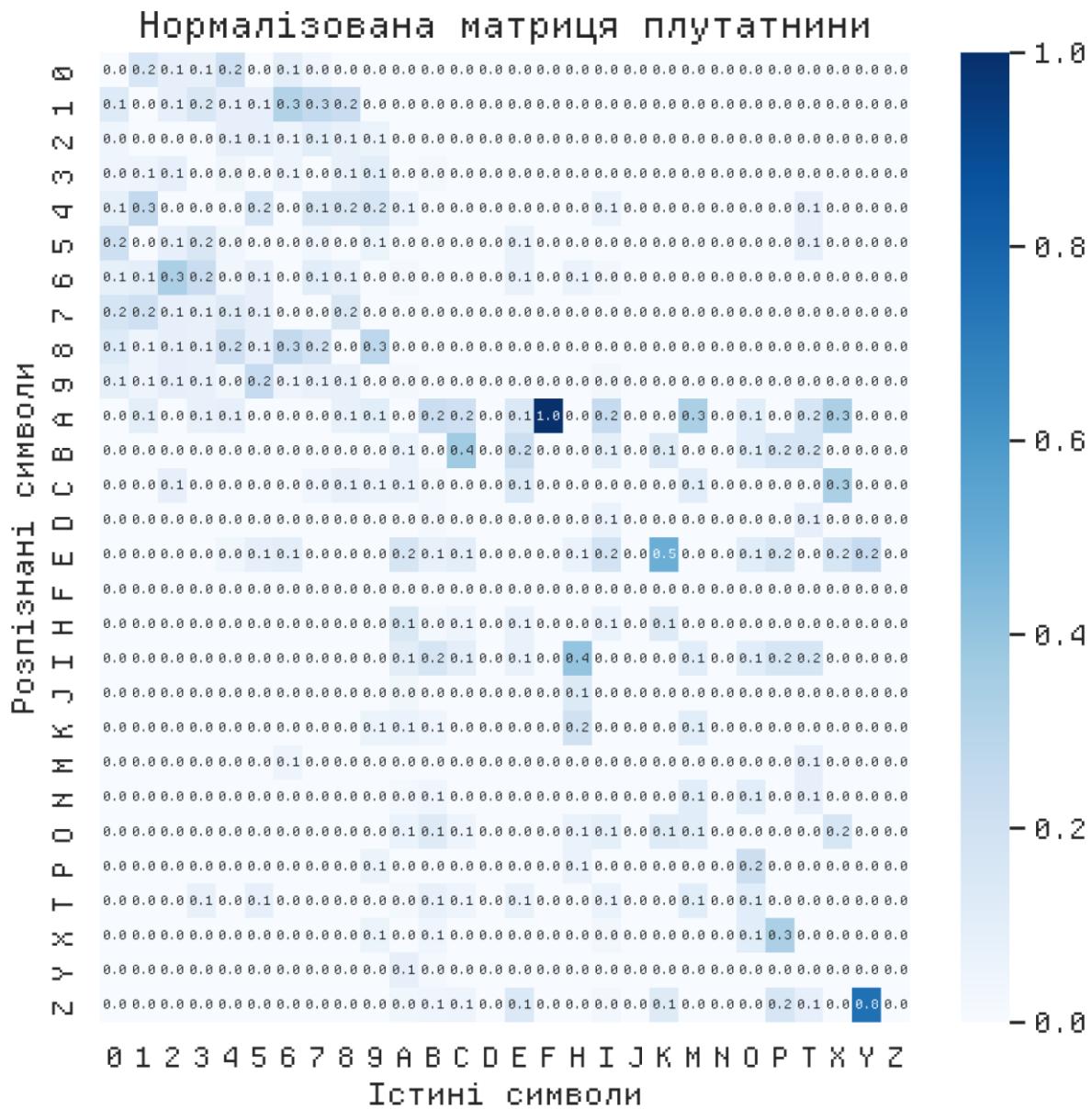


Рис. 17. Нормалізована матриця плутатини розпізнання символів

Літери F та Y є винятковими, оскільки їх застосування є обмеженим, а

тому вони є досить рідкісними.

### 3.3.2 Аналіз результатів ідентифікації транспортних засобів

Модель для ідентифікації транспортних засобів загалом має високі показники. На рис. 18 зображені графік залежності міри  $F_1$  від порогу впевненості моделі. Максимальний показник для всіх класів становить 0,92 при оптимальному порозі впевненості 0,682. З усіх класів, вантажівка

стабільно тримає нижчі показники, що пов'язано з узагальненою природою цього класу. Всі класи різко падають скоро після оптимального порогу впевненості.

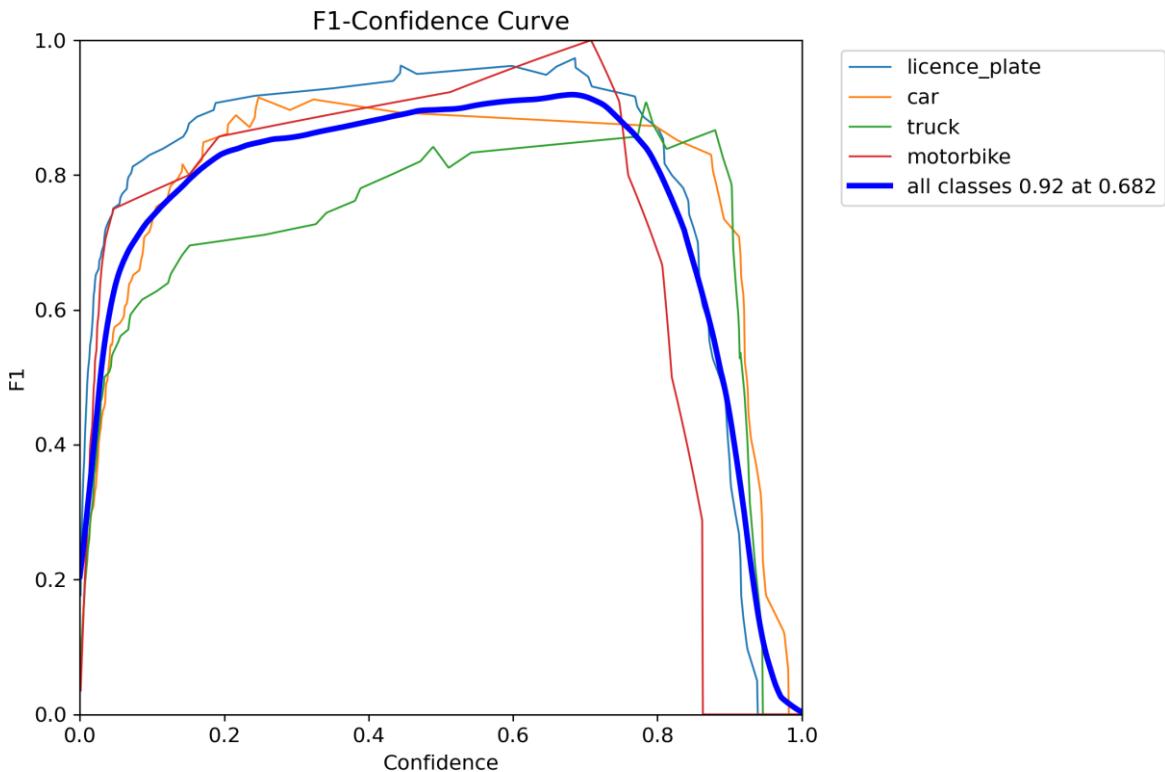


Рис. 18. Графік залежності міри  $F_1$  від порогу впевненості

Графік залежності точності від повноти на рис. 19 демонструє високу ефективність виявлення всіх цільових класів, хоча вантажівки все ще мають нижчі показники. Середнє значення точності при пороговому значенні IoU 0,5 становить 0,963, що говорить про високу якість виявлення та класифікації. Точність зберігає близько ідеальні показники до повноти у 0,8, де спостерігається різке падіння точності, однак точність для номерних знаків зберігається майже до показника повноти у 0,95.

Графік залежності точності від порогу впевненості на рис. 20 демонструє, що точність досягає максимального значення при порозі 0,940, однак для номерних знаків цей показник досягається при порозі приблизно

0,650. Також на цьому графіку знову спостерігається тренд нижчих показників для вантажівок.

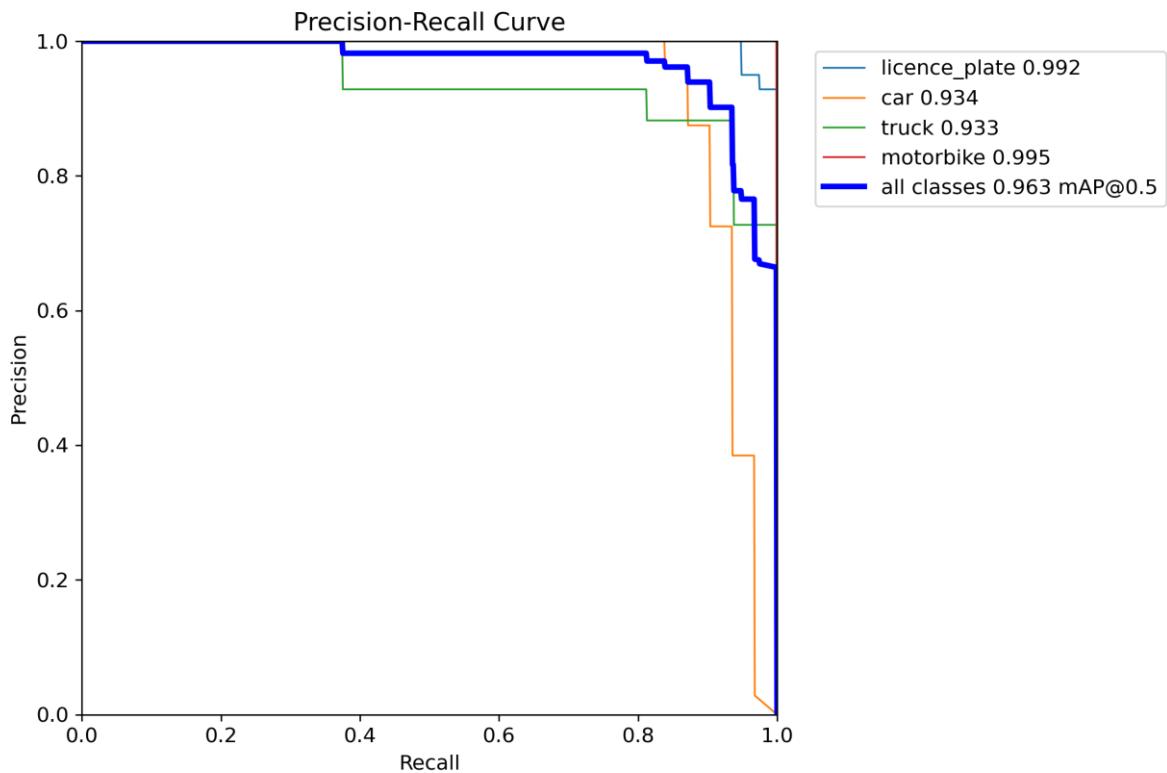


Рис. 19. Графік залежності точності від повноти

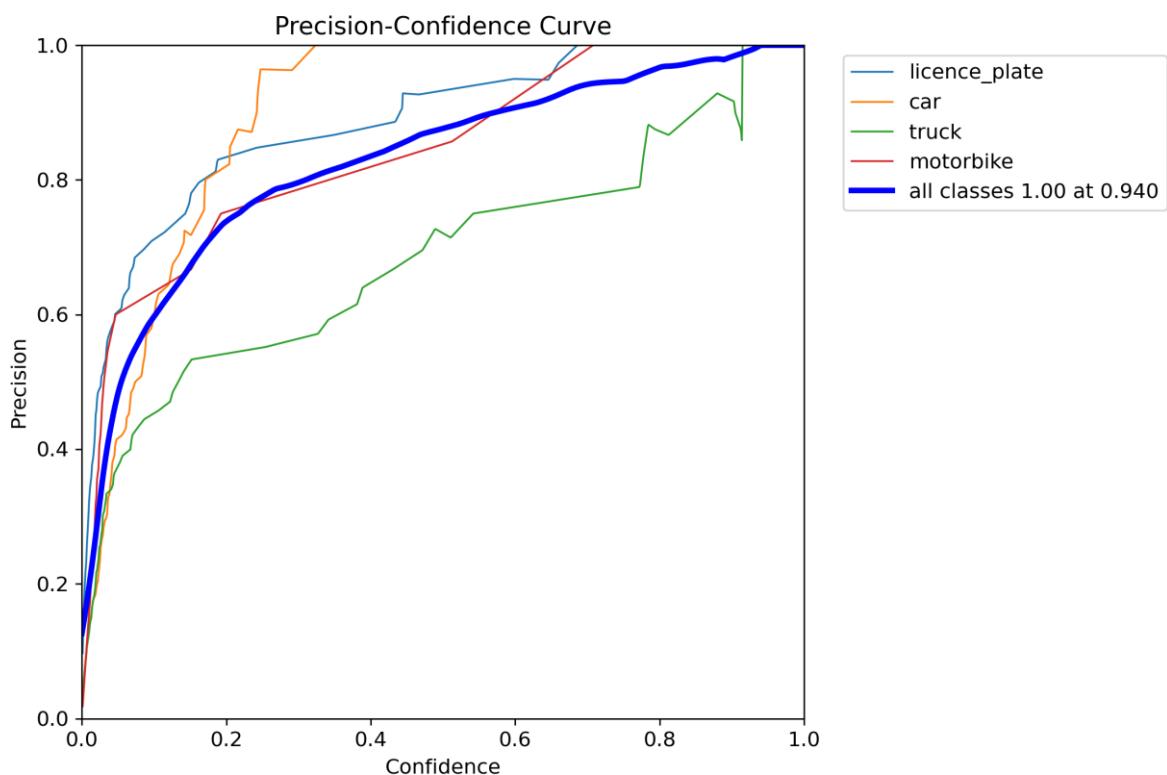


Рис. 20. Графік залежності точності від порогу впевненості

Графік залежності повноти від порогу впевненості на рис. 21 демонструє, що повнота 21 демонструє, що повнота 0,99 досягається при порозі впевненості 0. На відміну від інших графіків, найгірші показники має клас автомобіль, а не вантажівка. Повнота 1 зберігається для номерних знаків до порогу впевненості приблизно 0,45. Після порогу приблизно 0.75 повнота для всіх класів стрімко падає.

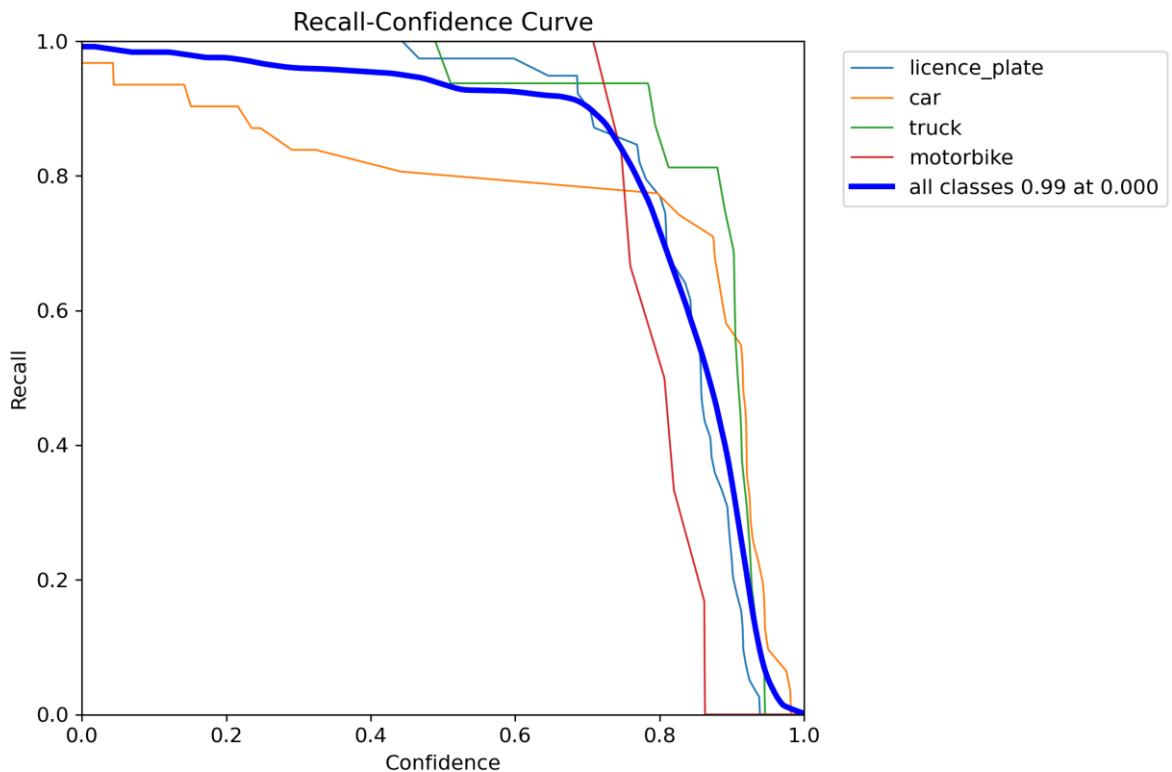


Рис. 21. Графік залежності повноти від порогу впевненості

Також технологія має показники швидкості 9,83 кадру в секунду і виконувалась на відеокарті Gigabyte RTX 4070 Super Windforce OC [21], що є відеокартою споживацького рівня. Цей показник не є достатнім для роботи в реальному часі. Але така швидкість достатня для виконання спостереження де швидкість транспортних засобів не є великою або проводиться аналіз відео чи фото в час відмінний від реального.

## ВИСНОВКИ

В ході цієї випускної кваліфікаційної роботи було розроблено технологію ідентифікації транспортних засобів та розпізнавання номерних знаків використовуючи модель You Only Look Once v11 та EasyOCR. Модель ідентифікації транспортних засобів має високі показники точності для всіх класів, а саме  $0,962 @ IoU 0.5$ . Однак метод розпізнавання номерних знаків на основі EasyOCR демонструє значно нижчі показники, а саме 0,412 для всіх видів ТЗ. Швидкість роботи склала 9,83 кадру в секунду на відеокарті Gigabyte RTX 4070 Super Windforce OC.

Показники ідентифікації транспортних засобів є високими, а модель є ефективним інструментом для розв'язання проблеми ідентифікації транспортних засобів різних видів на зображенні.

Показники розпізнавання тексту на номерних знаках є найслабшою частиною технології, що запропонована в цій роботі. Точність є досить низькою, особливо якщо номерний знак знаходиться під непрямим кутом. EasyOCR робить програмну реалізацію простою, але компромісом є значно нижчі показники в не ідеальних умовах.

Існує декілька напрямків для подальших дослідження та покращення технології:

1. Розширити технологію для можливості роботи з відео. Це дозволить не тільки зробити технологію більш практично, але також надасть можливість використання інформації з кількох кадрів для покращення точності розпізнавання тексту.
2. Розробити технологію розпізнавання тексту на основі YOLO, де кожен символ буде власним класом.
3. Покращити алгоритм виправлення помилок шляхом надання доступу до бази даних актуальних кодів регіонів та серій номерних знаків.

4. Розширити список класів типів транспортних засобів для більших можливостей аналізу вмісту.
5. Розширити можливості технології до розпізнавання номерних знаків для особливої техніки, причепів, військової техніки.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Гангало І. М., Лісовий Д. О., Жебка В. В. Розпізнавання об'єктів за допомогою технологій комп'ютерного зору. *Телекомунікації та інформаційні технології*. 2022. № 4. С. 46–52.  
URL: <https://doi.org/10.31673/2412-4338.2022.044652>.
2. Object detection with deep learning: a review / Z.-Q. Zhao et al. *IEEE transactions on neural networks and learning systems*. 2019. Vol. 30, no. 11. P. 3212–3232.  
URL: <https://doi.org/10.1109/TNNLS.2018.2876865>.
3. You only look once: unified, real-time object detection / J. Redmon et al. *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*. 2016. P. 779–788. URL: [https://www.cv-foundation.org/openaccess/content\\_cvpr\\_2016/papers/Redmon\\_You\\_O\\_nly\\_Look\\_CVPR\\_2016\\_paper.pdf](https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Redmon_You_O_nly_Look_CVPR_2016_paper.pdf).
4. Mopuri K. R., Garg U., Babu R. V. CNN fixations: an unraveling approach to visualize the discriminative image regions. *IEEE transactions on image processing*. 2018. Vol. 28, no. 5. P. 2116–2125.  
URL: <https://doi.org/10.1109/TIP.2018.2881920>.
5. A survey of the vision transformers and their CNN-transformer based variants / A. Khan et al. *Artificial intelligence review*. 2023. Vol. 56, Suppl. 3. P. 2917–2970. URL: <https://doi.org/10.1007/s10462-023-10595-0>.
6. Filipiuk M., Singh V. Comparing vision transformers and convolutional nets for safety critical systems. *Proceedings of the AAAI workshop on artificial intelligence safety*, Munich. 2022. URL: [https://ceur-ws.org/Vol-3087/paper\\_31.pdf](https://ceur-ws.org/Vol-3087/paper_31.pdf) (date of access: 20.05.2025).
7. Mao M., Hong M. YOLO object detection for real-time fabric defect inspection in the textile industry: a review of YOLOv1 to YOLOv11. *Sensors*. 2025. Vol. 25, no. 7. URL: <https://doi.org/10.3390/s25072270>.

8. Text Detection and Recognition from the Scene Images Using RCNN and EasyOCR / Y. L. Chaitra et al. *IOT with Smart Systems*. Singapore, 2023. P. 75–85. URL: [https://doi.org/10.1007/978-981-99-3761-5\\_8](https://doi.org/10.1007/978-981-99-3761-5_8).
9. Effectiveness of modern text recognition solutions and tools for common data sources / K. Smelyakov et al. *5th International conference on computational linguistics and intelligent systems*, Kharkiv, 22 April 2021. P. 154–165. URL: <https://ceur-ws.org/Vol-2870/paper15.pdf> (date of access: 20.05.2025).
10. TrOCR: Transformer-based optical character recognition with pre-trained models / M. Li et al. *The thirty-seventh AAAI conference on artificial intelligence (AAAI-23) : Proceedings of the AAAI conference on artificial intelligence*. 2023. P. 13094–13102.
11. Enhancing Automated Vehicle Identification by Integrating YOLO v8 and OCR Techniques for High-Precision License Plate Detection and Recognition / H. Moussaoui et al. *Scientific reports*. 2024. Vol. 14. URL: <https://doi.org/10.1038/s41598-024-65272-1>.
12. A CNN-based approach for automatic license plate recognition in the wild / M. Dong et al. *British machine vision conference*. 2017. URL: <https://bmva-archive.org.uk/bmvc/2017/papers/paper175/paper175.pdf> (date of access: 20.05.2025).
13. NVIDIA Tesla K40c Specs. *TechPowerUp*. URL: <https://www.techpowerup.com/gpu-specs/tesla-k40c.c2505> (date of access: 12.05.2025).
14. A robust real-time automatic license plate recognition based on the YOLO detector / R. Laroca et al. *2018 international joint conference on neural networks (IJCNN)*. 2018. P. 1–10. URL: <https://arxiv.org/pdf/1802.095> (date of access: 20.05.2025).

15. NVIDIA TITAN Xp Specs. *TechPowerUp*.  
URL: <https://www.techpowerup.com/gpu-specs/titan-xp.c2948> (date of access: 20.05.2025).
16. Koresh H. J. D. Impact of the preprocessing steps in deep learning-based image classifications. *National academy science letters*. 2024. Vol. 47. P. 645–647. URL: <https://doi.org/10.1007/s40009-023-01372-2>.
17. AUTO.RIA™ – Автобазар №1, купити та продати авто легко!. *AUTO.RIA*. URL: <https://auto.ria.com/uk/> (дата звернення: 20.05.2025).
18. OpenCV – Open Computer Vision Library. *OpenCV*. URL: <https://opencv.org> (date of access: 20.05.2025).
19. Galfar/deskew: Deskewing library for scanned documents. *GitHub*. URL: <https://github.com/galfar/deskew> (date of access: 12.05.2025).
20. ДСТУ 4278:2019. Дорожній транспорт. Знаки номерні транспортних засобів. Загальні вимоги. Правила застосування.
21. Gigabyte RTX 4070 SUPER WINDFORCE OC Specs. *TechPowerUp*. URL: <https://www.techpowerup.com/gpu-specs/gigabyte-rtx-4070-super-windforce-oc.b11487> (date of access: 20.05.2025).
22. Санжаров Д. Р., Філімонова Т. О. Ідентифікація транспортних засобів та номерних знаків з використанням моделі сімейства Ultralitics YOLO. *Інформаційні технології та кібербезпека в умовах воєнного часу* : Матеріали студент. науково-практ. конф., м. Київ, 23 квіт. 2025 р. С. 6–9.

## ДОДАТОК

Диплом І ступеня за участь у студентській науковій конференції

