

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

До захисту допущено:

Завідувач кафедри

_____Олександр РОЛІК

« ____ » _____ 2025 р.

Дипломний проєкт
на здобуття ступеня бакалавра
за освітньо-професійною програмою «Інформаційне забезпечення
робототехнічних систем»
спеціальності 126 «Інформаційні системи та технології»
на тему: «Автоматизована підсистема ідентифікації ушкоджень
мобільних транспортних засобів із застосуванням роботизованих
платформ»

Виконав:

студент ІV курсу, групи ІК-11

Жадько Данило Романович _____

Керівник:

доцент кафедри ІСТ, к.т.н, доцент

Остапченко Костянтин Борисович _____

Рецензент:

доцент кафедри ІП, к.т.н., доцент

Лісовиченко Олег Іванович _____

Засвідчую, що у цьому дипломному
проєкті немає запозичень з праць інших
авторів без відповідних посилань.

Студент _____

Київ – 2025 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інформаційне забезпечення робототехнічних систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Олександр РОЛІК

« _____ » _____ 2025р.

ЗАВДАННЯ

на дипломний проєкт студенту

Жадьку Данилу Романовичу

1.Тема проєкту «Автоматизована підсистема ідентифікації ушкоджень мобільних транспортних засобів із застосуванням роботизованих платформ», керівник проєкту Остапченко Костянтин Борисович, доцент кафедри ІСТ, к.т.н, доцент, затверджені наказом по університету від «23» травня 2025 р. №1705-с

2.Термін подання студентом проєкту: «09» червня 2025 року.

3.Вихідні дані до проєкту: відео потокові дані з роботизованих платформ, дані від сторонніх сервісів, мова серверної частини Python, моделі штучного інтелекту для автоматичної обробки зображень.

4.Зміст пояснювальної записки: аналіз предметної області багаторазового використання мобільних транспортних засобів, проєктування архітектури підсистеми ідентифікації ушкоджень, розробка програмного забезпечення підсистеми ідентифікації, проведення тестування та оцінювання ефективності підсистеми.

5.Перелік графічного матеріалу (із зазначенням обов'язкових креслеників,

плакатів, презентацій тощо): структурна схема, діаграма використання підсистеми, діаграма діяльності Telegram-бота, ER-діаграма бази даних, блок-схема алгоритму функції compare_damages

6.Дата видачі завдання 03.03.2025

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1	Затвердження теми роботи	08.03.2025	
2	Аналіз предметної області	05.04.2025	
3	Аналіз існуючих рішень, проблем та постановка завдань проєктування	08.04.2025	
4	Вибір технологій розробки	12.04.2025	
5	Створення структур наборів даних та донавчання моделі	29.04.2025	
6	Реалізація серверної частини підсистеми	8.05.2025	
7	Реалізація клієнтської частини підсистеми	17.05.2025	
8	Тестування підсистеми	20.05.2025	
9	Оформлення пояснювальної записки дипломного проєкту	01.06.2023	

Студент

Данило ЖАДЬКО

Керівник

Костянтин ОСТАПЧЕНКО

АНОТАЦІЯ

Жадько Д. Р. Автоматизована підсистема ідентифікації ушкоджень мобільних транспортних засобів із застосуванням роботизованих платформ. КПІ ім. Ігоря Сікорського, Київ, 2025.

Проект містить 67 сторінок тексту, 32 рисунків, 9 таблиць, посилання на 20 літературних джерел та 5 додаткові конструкторські документи.

Ключові слова: ідентифікація ушкоджень, мобільні транспортні засоби, штучний інтелект, комп'ютерний зір, роботизовані платформи.

Об'єктом дослідження є процеси та засоби виявлення ушкоджень сервісами багаторазового використання мобільних транспортних засобів

Метою розробки є створення ефективної і масштабованої підсистеми ідентифікації для автоматичного фіксування, аналізу і виявлення ушкоджень мобільних транспортних засобів за рахунок використання штучного інтелекту та роботизованих платформ.

У дипломному проекті розроблено архітектуру підсистеми, яка включає в себе модель комп'ютерного зору YOLOv11s для обробки зображень, серверну частину на Python із використанням FastAPI, а також інтерфейс користувача у вигляді Telegram-бота і адміністративної панелі.

Результати демонструють високий рівень точності і швидкості виявлення ушкоджень, що підтверджується тестуванням на реальних та штучних даних. Запропоноване рішення може бути інтегроване в існуючі сервіси багаторазового використання транспортних засобів, підвищуючи прозорість і ефективність процесів.

SUMMARY

Zhadko D. R. Automated subsystem for damage identification of mobile vehicles Using robotic platforms. Igor Sikorsky Kyiv Polytechnic Institute, Kyiv, 2025.

The project contains 67 pages of text, 32 figures, 9 tables, references to 20 literary sources, and 5 additional design documents.

Keywords: damage identification, mobile vehicles, artificial intelligence, computer vision, robotic platforms.

The object of the research is the processes and tools for damage detection in services for the multiple-use of mobile vehicles.

The goal of the development is to create an effective and scalable damage identification subsystem for automatic detection, analysis, and recording of damage to mobile vehicles using artificial intelligence and robotic platforms.

The work develops the subsystem architecture which includes the YOLOv11s computer vision model for image processing, a server-side part written in Python using FastAPI, as well as a user interface in the form of a Telegram bot and an administrative panel.

The results demonstrate a high level of accuracy and speed in damage detection confirmed by testing on real and synthetic data. The proposed solution can be integrated into existing multi-use vehicle services enhancing transparency and process efficiency.

Номер рядка	Формат	Позначення	Найменування	Кільк. аркушів	Номер екзем.	Примітка
1			<u>Документація загальна</u>			
2						
3			Знову розроблена			
4						
5	A4	ІК11.008БАК.006 ПЗ	Автоматизована підсистема	67		
6			ідентифікації ушкоджень			
7			мобільних транспортних засобів			
8			із застосуванням роботизованих			
9			платформ. Пояснювальна записка			
10	A3	ІК11.008БАК.006 Д1	Автоматизована підсистема	1		
11			ідентифікації ушкоджень			
12			мобільних транспортних засобів			
13			із застосуванням роботизованих			
14			платформ. Структурна схема			
15	A3	ІК11.008БАК.006 Д2	Автоматизована підсистема	1		
16			ідентифікації ушкоджень			
17			мобільних транспортних засобів			
18			із застосуванням роботизованих			
19			платформ. Діаграма діяльності			
20			Telegram-бота			
21	A3	ІК11.008БАК.006 Д3	Автоматизована підсистема	1		
22			ідентифікації ушкоджень			
23			мобільних транспортних засобів			
24			із застосуванням роботизованих			
25			платформ. Діаграма використання			
26			підсистеми			
27	A3	ІК11.008БАК.006 Д4	Автоматизована підсистема	1		
28			ідентифікації ушкоджень			

					ІК11.080БАК.006 ТП		
Зм.	Аркуш	№ докум.	Підпис	Дата			
Розроб.		Жадько Д.Р.			Літ.	Аркуш	Аркушів
Керівн.		Остапченко К.Б.			Т	1	2
					КПІ ім. Ігоря Сікорського		
Затв.					Група ІК-11		

Автоматизована підсистема ідентифікації ушкоджень мобільних транспортних засобів із застосуванням роботизованих платформ.

Відомість проекту

**Пояснювальна записка
до дипломного проєкту
на тему: «Автоматизована підсистема ідентифікації
ушкоджень мобільних транспортних засобів із
застосуванням роботизованих платформ»**

Київ – 2025 року

ЗМІСТ

ВСТУП.....	4
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ БАГАТОРАЗОВОГО ВИКОРИСТАННЯ МОБІЛЬНИХ ТРАНСПОРТНИХ ЗАСОБІВ.....	6
1.1 Сучасний стан та задачі багаторазового використання мобільних транспортних засобів	6
1.2 Аналіз можливостей впровадження засобів штучного інтелекту та роботизованих платформ.....	9
1.3 Підходи до вирішення проблем ідентифікації ушкоджень мобільних транспортних засобів	11
1.3.1 Штучний інтелект при вирішенні проблеми ідентифікації пошкоджень	11
1.3.2 Роботизовані платформи при вирішенні проблеми ідентифікації пошкоджень	13
1.4 Постановка завдання на проєктування підсистеми ідентифікації	14
Висновок до розділу 1	15
2 ПРОЄКТУВАННЯ АРХІТЕКТУРИ ПІДСИСТЕМИ ІДЕНТИФІКАЦІЇ УШКОДЖЕНЬ МОБІЛЬНИХ ТРАНСПОРТНИХ ЗАСОБІВ	16
2.1 Визначення вимог до підсистеми	16
2.1.1 Функціональні вимоги	16
2.1.2 Нефункціональні вимоги	18
2.2 Архітектура підсистеми.....	18
2.3 Вибір моделі виявлення ушкоджень у задачі ідентифікації.....	20
2.4 Вибір мови програмування та середовища проєктування підсистеми.....	22
2.4.1 Серверна частина і база даних	23
2.4.2 Telegram-бот і адміністративна панель	24
2.4.3 Модуль ідентифікації пошкоджень.....	25

					ІК11.080БАК.006 ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата			
		Жадько Д.Р.			Автоматизована підсистема ідентифікації ушкоджень мобільних транспортних засобів із застосуванням роботизованих платформ.		
		Остапченко К.Б					
						2	67
					КПІ ім. Ігоря Сікорського Група ІК-11		
Затв.					Пояснювальна записка		

Висновок до розділу 2.....	26
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ПІДСИСТЕМИ ІДЕНТИФІКАЦІЇ.....	28
3.1 Визначення структур даних та навчання моделі виявлення пошкоджень для задачі ідентифікації.....	28
3.1.1 Створення та анотація набору даних.....	28
3.1.2 Навчання моделі в хмарному середовищі Google Colab.....	31
3.2 Структура функціональних модулів підсистеми.....	33
3.2.1 Серверна частина.....	35
3.2.2 Telegram-бот.....	39
3.2.3 Адміністративна панель.....	41
3.3 Модель бази даних.....	42
3.4 Інтерфейси взаємодії із клієнтською та адміністративними частинами підсистеми.....	47
Висновок до розділу 3.....	52
4 ТЕСТУВАННЯ І ОЦІНКА ЕФЕКТИВНОСТІ ПІДСИСТЕМИ ІДЕНТИФІКАЦІЇ.....	54
4.1 Оцінка точності і швидкодії навченої моделі виявлення пошкоджень.....	54
4.2 Тестування функціоналу адміністративної панелі підсистеми.....	59
4.3 Тестування функціоналу Telegram-бота підсистеми.....	61
Висновок до розділу 4.....	63
ВИСНОВОК.....	64
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	66

ВСТУП

У сучасному світі ринок послуг багаторазового використання транспортних засобів показує стрімке зростання, що зумовлене підвищеним попитом на доступні та зручні транспортні засоби у всіх містах України. Зростання кількості користувачів актуалізує потребу в удосконаленні та впровадженні нових підходів до автоматизації процесів, пов'язаних з перевіркою візуального стану автомобілів. Також актуальним стає питання мінімізації людського фактору та підвищення точності й швидкості виявлення пошкоджень.

У більшості компаній, що надають подібні послуги, візуальний огляд автомобіля виконується вручну або з використанням незначної автоматизації. Такі підходи є недостатньо ефективними у випадках, коли компанії мають багато транспортних засобів, та створюють ризики виникнення спірних ситуацій між клієнтом і сервісом. Необхідність підвищення рівня прозорості, точності та швидкості оцінки стану транспортних засобів обумовлює інтерес до технологій штучного інтелекту та роботизованих платформ.

Технології штучного інтелекту на основі глибокого навчання та комп'ютерного зору з впровадженням роботизованих платформ демонструють високий потенціал для автоматичного виявлення зовнішніх пошкоджень. Використання даних моделей в цій роботі дозволить створити повноцінний цикл перевірки транспортного засобу від автоматичної фотофіксації до виявлення змін та запису результатів у базу даних інформаційної системи сервісу обслуговування. Це дасть змогу не лише автоматизувати процес, а й суттєво підвищити точність та знизити навантаження при оцінці на персонал.

Метою даної дипломної роботи є створення ефективної і масштабованої підсистеми ідентифікації для автоматичного фіксування, аналізу і виявлення пошкоджень мобільних транспортних засобів за рахунок використання штучного інтелекту та роботизованих платформ. Особливу увагу зосереджено на забезпеченні простоти використання для кінцевих користувачів за допомогою Telegram-бота та

					ІК11.080БАК.006 ПЗ	Арк.
						4
Зм.	Арк.	№ докум.	Підпис	Дата		

створенні адміністративної вебпанелі для внутрішньої обробки та керування даними.

Об'єктом дослідження є процеси та засоби виявлення ушкоджень сервісами багаторазового використання мобільних транспортних засобів.

Предметом дослідження є технології комп'ютерного зору за для автоматизації цифрових сервісів багаторазового використання транспортних засобів.

Практична цінність роботи полягає у можливості впровадження запропонованого рішення в реальні умови функціонування сервісів багаторазового використання транспортних засобів, що дозволить оптимізувати витрати, підвищити швидкість та якість обслуговування та забезпечити прозорість взаємодії з клієнтами.

					ІК11.080БАК.006 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		5

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ БАГАТОРАЗОВОГО ВИКОРИСТАННЯ МОБІЛЬНИХ ТРАНСПОРТНИХ ЗАСОБІВ

1.1 Сучасний стан та задачі багаторазового використання мобільних транспортних засобів

Ринок багаторазового використання мобільних транспортних засобів в Україні останні роки демонструє динамічний розвиток, особливо під час зростання вартості обслуговування власного транспорту, витрат на паливе, технічний сервіс, зберігання та страхування. Дедалі більше людей обирають багаторазове використання транспортних засобів як дешевшу альтернативу володінню власним транспортом [1].

Разом із зростаючим попитом цієї галузі, діджиталізація суттєво трансформувала підходи до багаторазового тимчасового володіння транспортними засобами. Поширеною стала можливість дистанційного бронювання, вибору локації для отримання і повернення транспорту, безконтактної оплати та доступу до, наприклад, автомобіля за допомогою мобільного телефону. Багато компаній впроваджують мобільні сервіси, що дозволяють повністю автоматизувати процес оренди без необхідності взаємодії з іншою людиною. Але такі ключові етапи, як візуальний огляд транспортного засобу, перевірка пошкоджень та фіксація їх стану, усе ще часто виконуються вручну працівниками або користувачами. Це зберігає залежність від людського фактору, але й створює простір для вдосконалення за допомогою інтелектуальних та роботизованих платформ – автономних систем, які оснащені камерами, що дозволять автоматизувати збір візуальної інформації та підвищити точність діагностики.

Особливо показовим є досвід міжнародних компаній, таких як Sixt, Avis та Hertz, які вже доволі давно перейшли до повністю цифрових моделей обслуговування клієнтів. У сервісах Sixt [2] процес тимчасового багаторазового використання зведено до кількох дій у мобільному застосунку, починаючи з перевірки документів до відкриття транспортного засобу за допомогою телефону. Усі фотофіксації та оцінки стану транспорту виконуються автоматизовано, а

					ІК11.080БАК.006 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		6

результати одразу інтегруються в систему. Це дозволяє не тільки мінімізувати участь персоналу, але й виключити ризики, пов'язані з суб'єктивністю оцінки пошкоджень людиною.

Дивлячись на досвід закордонних компаній, український ринок мобільних транспортних засобів також демонструє стійке зростання та поступове впровадження цифрових підходів. Популярні сервіси, такі як Getmancar, BLS та Uklon Drive потрошку впроваджують безконтактне бронювання через мобільні додатки, перевірку документів онлайн і самостійне відкриття машини.

Особливу увагу на ринку привертає Getmancar [3] – одна з перших українських платформ, яка повністю відмовилася від фізичних офісів та особистого контакту з персоналом (рис. 1.1). Усі етапи, такі як реєстрація, завантаження документів, вибір транспортного засобу, бронювання, запуск багаторазового використання та її завершення реалізовано всередині мобільного застосунку. Всі транспортні засоби обладнані телематичними модулями, що забезпечують віддалений доступ до дверей, контроль геопозиції та зчитування пробігу. Завдяки цьому вдалося скоротити час оформлення прокату до кількох хвилин, знизити навантаження на персонал і значно підвищити масштабованість сервісу. Компанія активно розширює покриття у великих містах, зокрема у Києві, Львові та Дніпрі, і залучає користувачів завдяки простоті та швидкості взаємодії.

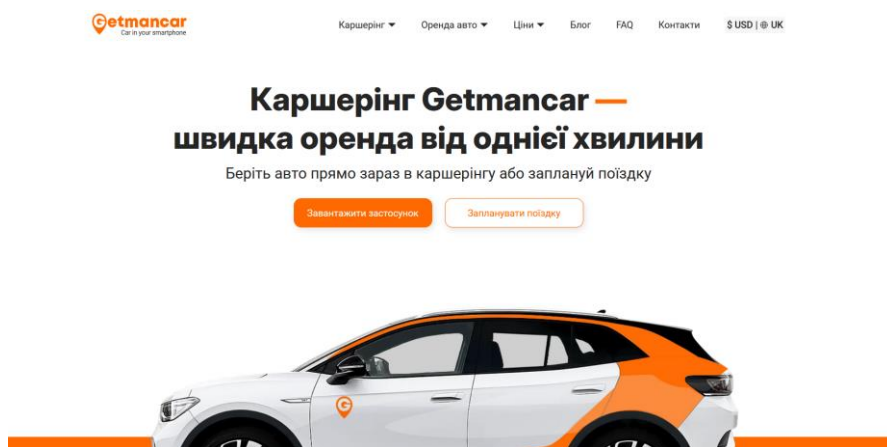


Рисунок 1.1 – Сервіс Getmancar

Зростання конкуренції, очікування клієнтів та збільшення доступності сучасних технологій підвищують інтерес у впровадженні нових підходів у всіх

					ІК11.080БАК.006 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		7

ланках від бронювання до виявлення пошкоджень. У відповідь на ці тенденції дедалі більше компаній прагнуть інтегрувати рішення, які здатні покращити якість обслуговування, підвищити ефективність і знизити витрати.

Попри розвиток цифрових сервісів та збільшення попиту, у багатьох компаніях українського ринку досі зберігається традиційна модель багаторазового використання транспортних засобів, яка передбачає значну кількість ручних операцій, залучення персоналу та використання паперової документації. Такий підхід створює низку проблем як для клієнтів, так і для самих сервісів багаторазового використання транспортних засобів.

Однією з основних проблем є людський фактор під час перевірки стану транспортного засобу. До та після тимчасового використання працівник компанії або сам користувач здійснює візуальний огляд транспорту, фіксує можливі пошкодження за допомогою фото або описує їх вручну. Така процедура є суб'єктивною, залежить від уважності або якості фото, яке робить людина, а також не гарантує повного виявлення всіх наявних дефектів. Це часто стає причиною конфліктних ситуацій, коли сторони не можуть дійти згоди щодо того, коли саме з'явилося пошкодження до або після.

Ще однією проблемою є складний та тривалий процес оформлення документів, який у багатьох випадках вимагає особистої присутності клієнта в офісі компанії, підписання договору вручну, очікування на перевірку документів та технічного стану транспорту. Це створює черги, втрату часу і, як наслідок, зниження задоволеності клієнтів сервісом.

Також варто відзначити відсутність стандартизованої системи збереження та порівняння даних про стан транспорту. Фото, зроблені вручну, можуть бути з низькою якістю, зняті під різними кутами, не мати позначок часу чи геолокації, що унеможливорює об'єктивне порівняння стану мобільного засобу пересування у різні періоди.

У випадку великих транспортних парків додатково виникає складність масштабування таких процесів. Кожен транспортний засіб потребує окремої уваги,

					ІК11.080БАК.006 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		8

що вимагає збільшення персоналу та часу на обробку кожної заявки на тимчасове використання.

Наприклад у невеликих компаніях або локальних сервісах, які не мають змоги впровадити цифрові інструменти, всі ці недоліки призводять до втрати клієнтів, зростання витрат на обслуговування та зниження репутації. У свою ж чергу, великі гравці ринку поступово відходять від ручної моделі, оскільки вона виявляється економічно не вигідною в довгостроковій перспективі.

Традиційний підхід до тимчасового використання транспортних засобів більше не відповідає очікуванням сучасного користувача і створює ризики для бізнесу. Це обґрунтовує необхідність поступової, а з часом і повної автоматизації ключових етапів процесу.

1.2 Аналіз можливостей впровадження засобів штучного інтелекту та роботизованих платформ

З огляду задач, які виконуються у сфері багаторазового використання мобільних транспортних засобів, впровадження штучного інтелекту (ШІ) та роботизованих платформ має значний потенціал для підвищення якості та швидкості. Основними напрямками застосування цих технологій можуть бути автоматизація збору, аналізу та обробки даних транспортних засобів, а також оптимізація операційних процесів і покращення взаємодії користувачів із сервісами.

Технології комп'ютерного зору на базі алгоритмів глибинного навчання можуть бути застосовані для ідентифікації зовнішніх пошкоджень транспортних засобів. Це також дозволить зменшити вплив людського фактора, підвищити точність та об'єктивність порівняння стану транспортного засобу під час багаторазового використання. Сучасні моделі ШІ, при правильному навчанні, здатні аналізувати та порівнювати зображення до та після використання з різних ракурсів, виявляти подряпини, вм'ятини, тріщини.

					ІК11.080БАК.006 ПЗ	Арк.
						9
Зм.	Арк.	№ докум.	Підпис	Дата		

Роботизовані платформи, споряджені високоточними камерами, можуть автоматично збирати візуальну інформацію без участі користувача чи працівника сервісу. Це не тільки знизить час проведення огляду, чим дозволить оперативно контролювати стан великого автопарку, а й підвищить точність зображень, що може бути інтегровано з технологіями штучного інтелекту для ідентифікації пошкоджень.

Також можливе впровадження ІІІ з телематичними системами та інтернетом речей (IoT). Це зробить можливим автоматичний моніторинг та запис технічних параметрів у режимі реального часу, а саме пробіг, стан двигуна, рівень палива транспортного засобу. Подальший аналіз цих даних і прогнозна аналітика, завдяки штучному інтелекту, допоможе своєчасно планувати технічне обслуговування і запобігати аварійним ситуаціям.

Застосування інтелектуальних чат-ботів і голосових помічників на базі штучного інтелекту може покращити взаємодію клієнтів із сервісами оренди, надаючи швидкі та змістовні відповіді на запити, допомогу в оформленні замовлень та інші послуги без залучення живого персоналу.

Звісно впровадження технологій ІІІ та роботизованих платформ у функціонуючі системи потребує значних інвестицій у розробку, впровадження та забезпечення конфіденційності даних. Проте можливості та переваги, які дає використання цих технологій, роблять цей вклад цілком виправданим і перспективним для розвитку бізнесу.

Також, враховуючи світовий досвід і зростаючий попит на цифрові послуги в галузі багаторазового використання транспортних засобів, впровадження засобів штучного інтелекту та роботизованих платформ є перспективним напрямком розвитку українських сервісів.

Найбільшою і найактуальнішою задачею для ІІІ, в нашому регіоні, залишається саме ідентифікація зовнішніх пошкоджень транспортних засобів. На сьогодні сформувалося кілька основних підходів до розв'язання цієї проблеми, які різняться за складністю, точністю та рівнем автоматизації.

					ІК11.080БАК.006 ПЗ	Арк.
						10
Зм.	Арк.	№ докум.	Підпис	Дата		

Більш поширеним підходом залишається ручний візуальний огляд з ручною фотофіксацією. Його використовують невеликі компанії та локальні сервіси. Працівник або клієнт самостійно фотографує транспортний засіб і порівнює поточний стан із попередніми фото. Такий підхід є недорогим, але вкрай неефективним на великих масштабах, оскільки залежить від людського фактора, не забезпечує повторюваності оцінки й не гарантує повної об'єктивності.

Прогресивнішим є використання напівавтоматизованих систем, у яких працівник обирає пошкодження на зображенні вручну або заповнює електронні форми з описом дефектів. Ці дані зберігаються в базі, можуть доповнюватись фото та відео і використовуються для створення звітів. Хоча цей підхід дозволяє краще організувати облік інформації, він усе ще вимагає участі працівника не тільки для ідентифікації, а й для фотографування.

Усі ці підходи, хоч і поступово впроваджують напівавтоматичні рішення, але все ще залежать від людського фактору, що обмежує їх ефективність і точність. Саме тому впровадження методів штучного інтелекту та роботизованих платформ набуває дедалі більшої популярності, вони дозволять перейти до повністю автоматизованих підходів ідентифікації ушкоджень транспортних засобів.

1.3 Підходи до вирішення проблем ідентифікації ушкоджень мобільних транспортних засобів

1.3.1 Штучний інтелект при вирішенні проблеми ідентифікації пошкоджень

Із розвитком цифрових технологій, усе більше завдань, які раніше вимагали участі людини, переходять до штучного інтелекту. Це особливо важливо при задачах, де потрібно швидко, точно та надійно обробляти великі обсяги даних, як при проблемі ідентифікації стану транспортних засобів.

Важливим і надзвичайно ефективним піднапрямом ІІІ є комп'ютерний зір, який дозволяє машинам «бачити» та аналізувати зображення і відео [4]. У завданнях перевірки транспортних засобів на наявність пошкоджень, технології комп'ютерного зору виявились надзвичайно результативними.

					ІК11.080БАК.006 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		11

Сучасні системи комп'ютерного зору побудовані на основі глибокого навчання та надають змогу не лише ідентифікувати об'єкти на зображенні, а й оцінювати їхній стан. У випадку транспортних засобів, це означає здатність автоматично виявляти такі пошкодження, як подряпини, тріщини, вм'ятини, відсутні елементи кузова тощо. Такі системи можуть визначати точне місце пошкодження на поверхні транспорту, класифікувати тип дефекту за тяжкістю або характером, а також порівнювати поточні зображення з історичними даними для виявлення нових ушкоджень.

Більшість сучасних моделей комп'ютерного зору базуються на згорткових нейронних мережах (Convolutional Neural Networks), які ідеально підходять для обробки візуальних даних. Ці архітектури виявляють просторові патерни в пікселях зображень і формують ієрархію ознак від базових контурів до складних структур, характерних для певних типів пошкоджень. Саме CNN стали базовою технологією, що дозволила досягти прориву у точності детекції візуальних об'єктів [5].

Однією з найефективніших згорткових моделей, що застосовується для таких цілей, є YOLO (You Only Look Once) — архітектура нейронних мереж для детектування об'єктів у реальному часі, яка дозволяє швидко та з високою точністю виявляти і класифікувати пошкодження. Ці моделі одночасно виконують локалізацію об'єктів і класифікацію, що робить її ідеальною для задач, де потрібна швидка реакція та точність, зокрема, для автоматичного контролю стану транспортних засобів.

Ефективна робота таких систем неможлива без якісного датасету тобто великого обсягу навчальних зображень, анотованих відповідно до типів пошкоджень і зон їх розташування. У процесі навчання нейронна мережа вчиться розпізнавати закономірності в даних, і поступово набуває здатності точно виявляти пошкодження навіть у складних умовах освітлення, ракурсу або якості знімків. Розробка такого датасету це один із критичних етапів створення подібної підсистеми, оскільки він визначає рівень точності та стійкості всієї системи до помилок.

					ІК11.080БАК.006 ПЗ	Арк.
						12
Зм.	Арк.	№ докум.	Підпис	Дата		

У типовому сценарії використання така система працює наступним чином: користувач (клієнт або працівник компанії) завантажує фото транспортного засобу в систему, яка миттєво аналізує його, виявляє пошкодження, формує візуалізацію та короткий висновок. Оброблені результати автоматично зберігаються в базі даних та використовуються для формування звітів.

Завдяки поєднанню комп'ютерного зору та надійних баз даних виходить повний цикл автоматизованої ідентифікації від прийому зображення до підготовки результатів. Це дозволяє значно зменшити вплив людського фактору, підвищити точність перевірок, масштабувати контроль за великими парками транспортних засобів та забезпечити новий рівень прозорості й довіри у сфері тимчасового багаторазового використання транспорту.

1.3.2 Роботизовані платформи при вирішенні проблеми ідентифікації пошкоджень

Розвиток роботизованих платформ став новим етапом у вдосконаленні процесів ідентифікації пошкоджень мобільних транспортних засобів. Ці системи забезпечують автоматизований збір візуальної інформації, мінімізуючи потребу в участі людини та підвищуючи точність зображень.

Під роботизованими платформами для ідентифікації пошкоджень треба представляти як автономні або напівавтономні системи, оснащені високоточними автоматичними камерами, сенсорами, інфрачервоними сканерами або технологіями LIDAR. Вони здійснюють фотографування транспортних засобів у високій роздільній здатності, з автоматичним фокусуванням, регулюванням експозиції та балансом білого.

Інтеграція цих платформ може не лише пришвидшити автоматичне фіксування стану транспортного засобу, а й передавати отримані зображення для подальшої обробки цих точних зображень нейронними мережами. Такий підхід дозволить створити єдину автоматичну систему виявлення стану кожного транспортного засобу, виключити людський фактор при виявленні пошкоджень та

					ІК11.080БАК.006 ПЗ	Арк.
						13
Зм.	Арк.	№ докум.	Підпис	Дата		

забезпечить прозорість та підвищить довіру користувачів до компаній багаторазового використання авто.

В Україні впровадження подібних рішень перебуває на початковому етапі та зустрічає бар'єри у вигляді відсутності готових наборів даних, технічної експертизи та консервативного підходу багатьох компаній. Проте, застосування таких технологій відкриває значні перспективи для підвищення ефективності, якості обслуговування та конкурентоспроможності на ринку багаторазового використання. Враховуючи глобальні тенденції, саме ці технології можуть стати основою цифрової трансформації галузі.

1.4 Постановка завдання на проєктування підсистеми ідентифікації

Головною метою дипломного проєкту є створення ефективної і масштабованої підсистеми ідентифікації для автоматичного фіксування, аналізу і виявлення ушкоджень мобільних транспортних засобів за рахунок використання штучного інтелекту та роботизованих платформ. Для досягнення цієї мети постановлено наступні завдання:

– розробити та навчити модель штучного інтелекту на основі нейронної мережі архітектури YOLO для автоматичного аналізу й порівняння зображень транспортних засобів для ідентифікації нових пошкоджень;

– спроектувати й реалізувати веборієнтовану інформаційну підсистему ідентифікації ушкоджень із зручним клієнтським інтерфейсом взаємодії через Telegram-бот, який надаватиме користувачам швидкий доступ до основних функцій підсистеми, таких як завантаження зображень і перегляд результатів аналізу, а також вебінтерфейс адміністративної панелі, який забезпечить ефективне управління користувачами, транспортними засобами, орендами та дозволить зручно переглядати результати аналізу виявлених пошкоджень;

– обрати і створити структуру бази даних для надійного зберігання й ефективного користування інформацією про користувачів, транспортні засоби, оренди, фотографії, а також історії виявлених пошкоджень.

					ІК11.080БАК.006 ПЗ	Арк.
						14
Зм.	Арк.	№ докум.	Підпис	Дата		

В цілому при проектуванні підсистеми забезпечити інтеграцію всіх її компонентів у єдину інформаційну систему із чіткими та стандартизованими протоколами взаємодії.

Висновок до розділу 1

У результаті проведеного аналізу предметної області багаторазового використання мобільних транспортних засобів, виявлено сучасний стан ринку в Україні та зафіксовано основні тенденції його розвитку. Встановлено, що попри активне впровадження цифрових технологій, ключові етапи перевірки та фіксації стану транспортних засобів у більшості компаній залишаються залежними від людського фактора, що призводить до суб'єктивності оцінок та знижує ефективність процесів.

Проаналізовано наявні підходи до ідентифікації ушкоджень транспортних засобів і визначено їх обмеження щодо точності, повторюваності та масштабованості. Порівняно різні технологічні рішення, зокрема застосування штучного інтелекту, комп'ютерного зору та роботизованих платформ, що дозволяють суттєво підвищити якість та швидкість виявлення ушкоджень.

Запропоновано впровадження автоматизованої підсистеми на основі сучасних моделей комп'ютерного зору (зокрема YOLO) у поєднанні з роботизованими платформами для збору високоякісних зображень. Визначено, що такі рішення забезпечують мінімізацію впливу людського фактора, підвищують об'єктивність та прозорість процесу перевірки, а також сприяють підвищенню довіри клієнтів і конкурентоспроможності сервісів.

Сформульовано завдання до проектування підсистеми ідентифікації ушкоджень та визначено доцільність створення клієнтського інтерфейсу, адміністративної панелі, бази даних та моделі штучного інтелекту для автоматизованої ідентифікації пошкоджень транспортних засобів.

					ІК11.080БАК.006 ПЗ	Арк.
						15
Зм.	Арк.	№ докум.	Підпис	Дата		

2 ПРОЄКТУВАННЯ АРХІТЕКТУРИ ПІДСИСТЕМИ ІДЕНТИФІКАЦІЇ УШКОДЖЕНЬ МОБІЛЬНИХ ТРАНСПОРТНИХ ЗАСОБІВ

2.1 Визначення вимог до підсистеми

Під час проєктування архітектури підсистеми важливим етапом є формулювання функціональних та нефункціональних вимог. Такий поділ дозволяє структурувати розробку та забезпечити відповідність готового продукту до очікуваням при проєктуванні.

2.1.1 Функціональні вимоги

Функціональні вимоги визначають, які саме задачі має виконувати система та які сервіси вона повинна надавати користувачам. Іншими словами, вони описують "що повинна робити система" з точки зору її практичного використання.

На основі аналізу ринку багаторазового використання транспортних засобів та загальних методів виявлення пошкоджень, можна сформулювати такі функціональні вимоги до підсистеми та окремих її компонентів:

- підсистема повинна підтримувати перегляд користувачем поточного використання мобільного транспортного засобу, історії завершених використань, а також списку доступних транспортних засобів через Telegram-бот;
- підсистема повинна дозволяти користувачу проходити реєстрацію або авторизацію через Telegram-бота, надаючи базові контактні дані;
- підсистема повинна дозволяти користувачу мати можливість завантажувати фотографії транспортного засобу під час тимчасового використання транспортного засобу через Telegram-бот;
- підсистема повинна приймати та зберігати посилання на фотографії у базі даних, прив'язуючи їх до конкретного тимчасового використання;
- підсистема повинна виконувати автоматичний аналіз зображень за допомогою моделі комп'ютерного зору;

					ІК11.080БАК.006 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		16

- модель виявлення ушкоджень на базі ШІ повинна виявляти типи пошкоджень, наприклад, подряпини, вм'ятини, тріщини та локалізувати їх координати на фото;
- модель повинна виявляти й записувати в базу даних всю отриману інформацію про кожне виявлене пошкодження, а саме координати на зображенні, тип, впевненість моделі, прив'язуючи їх до конкретного зображення;
- підсистема повинна автоматично порівнювати фотографії «до» і «після» використання, виявляючи нові пошкодження на основі IoU-метрики;
- підсистема у випадку виявлення нового пошкодження після епізоду використання, повинна повідомляти адміністратора та заморожувати автоматичне завершення для користувача. В випадку відсутності нових пошкоджень, повинна автоматично завершувати оренду;
- підсистема повинна дозволяти адміністратору переглядати всі використання транспортних засобів, фото та результати детекції пошкоджень через вебінтерфейс;
- підсистема повинна підтримувати генерацію PDF та CSV-звітів по кожному запису з повною інформацією про оренду;
- підсистема повинна забезпечувати можливість створювання, редагування, видалення даних користувачів, транспорту та тимчасових використань, які були надані підсистемі від компанії оренди авто, з панелі адміністратора;
- підсистема повинна забезпечувати можливість ручного завершення тимчасового використання через панель адміністратора при виявленні пошкодження;
- підсистема повинна забезпечувати можливість автоматичного завантаження зображень, які були надані підсистемі з роботизованої платформи;
- підсистема повинна мати API для обміну даними між Telegram-ботом, моделлю виявлення пошкоджень та базою даних;
- підсистема повинна бути захищеною від несанкціонованого доступу, зокрема обмежувати доступ до адміністративної панелі авторизацією.

					ІК11.080БАК.006 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		17

2.1.2 Нефункціональні вимоги

Нефункціональні вимоги описують "як повинна працювати система". Вони визначають загальні характеристики якості системи, такі як продуктивність, зручність використання, надійність, масштабованість.

Підсистема автоматичної ідентифікації ушкоджень мобільних транспортних засобів має відповідати таким нефункціональним вимогам:

- підсистема повинна обробляти завантажені зображення та повертати результат виявлення пошкоджень із затримкою не більше 2 секунд;
- підсистема повинна підтримувати одночасну обробку великої кількості запитів без значної втрати продуктивності;
- рівень доступності підсистеми повинен становити не менше 90,0 % часу;
- Telegram-інтерфейс та інтерфейс панелі адміністраторів повинні бути інтуїтивно зрозумілими і зручними для користувачів без технічної підготовки;
- Telegram-інтерфейс та інтерфейс панелі адміністраторів повинні бути адаптивними під різні пристрої;
- підсистема повинна бути сумісною з основними серверними платформами;
- підсистема повинна забезпечувати конфіденційність персональних даних користувачів і результатів фотоаналізу;

2.2 Архітектура підсистеми

Виходячи з сформульованих завдань та вимог, пропонується застосувати клієнт-серверну технологію до створення архітектури підсистеми автоматизованої ідентифікації ушкоджень мобільних транспортних засобів із застосуванням роботизованих платформ в складі серверної частини, користувацької частин, бази даних та сторонніх сервісів (рис. 2.1).

					ІК11.080БАК.006 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		18

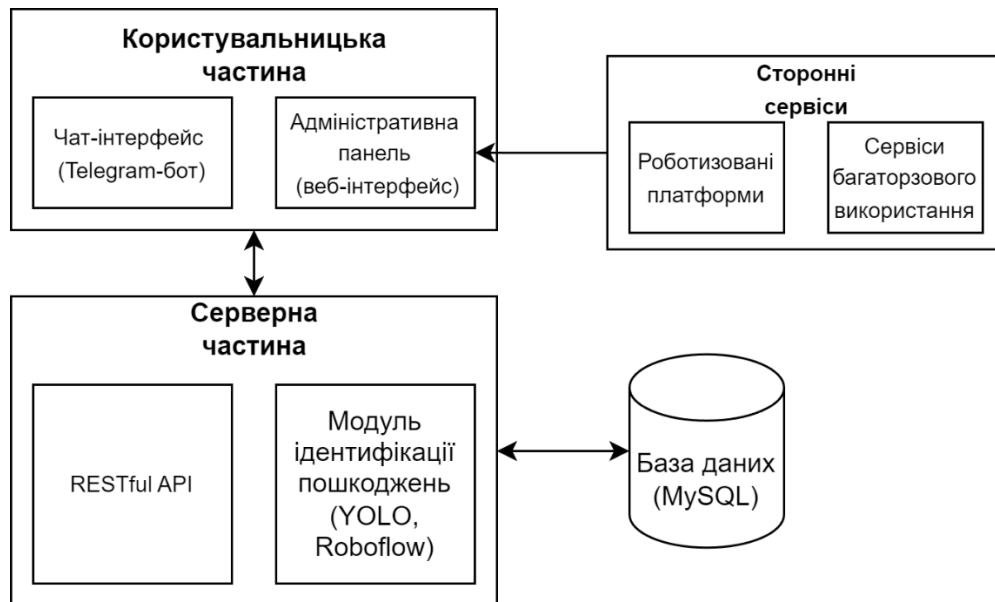


Рисунок 2.1 – Архітектура підсистеми

Центральним компонентом є серверна частина, яка реалізує бізнес-логіку, модуль ідентифікації пошкоджень, інтерфейси взаємодії з користувальницькою частиною та базою даних. Серверна частина включає в себе стек інтерфейсів взаємодії RESTful API, що робить її єдиним центром обробки даних та взаємодії усіх компонентів підсистеми. Модуль ідентифікації пошкоджень включає в себе набір даних та навчену модель виявлення ушкоджень на основі архітектури комп'ютерного зору YOLO, яка здійснює автоматизований аналіз зображень транспортних засобів для виявлення, класифікації та локалізації ушкоджень. Модель зв'язана з RESTful API, де результати обробки автоматично передаються до бази даних.

База даних відповідає за зберігання та доступ до усіх необхідних даних. У контексті підсистеми ідентифікації ушкоджень транспортних засобів особливо важливим є підтримка складної структури даних з чітко визначеними взаємозв'язками між сутностями, такими як користувачі, транспортні засоби, оренди, та виявлені пошкодження. Для цього ідеально підходить реляційна база даних MySQL, яка забезпечить узгодженість даних, їх цілісність та доступність усіх компонентів підсистеми.

					ІК11.080БАК.006 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		19

Ключовими вимогами до користувальницької частини підсистеми є забезпечення простого, швидкого та інтуїтивно зрозумілого інтерфейсу, щоб задовольняти усі категорії користувачів. Виходячи з цього, найбільш мобільним та зручним є використання чат-орієнтованого інтерфейсу Telegram-бот, що дозволить здійснювати основні дії через звичайні повідомлення та меню в застосунку Telegram, який є широковживаним засобом комунікації. Цей інтерфейс реалізовує користувальницьку логіку, а саме завантаження зображень транспортних засобів, керування процесами та перевірку стану багаторазового використання транспортного засобу.

Також користувальницька частина включає в себе вебінтерфейс адміністрування для персоналу сервісу багаторазового використання транспортних засобів. Він забезпечить розширене управління даними користувачів, транспортних засобів, історією використання, перегляд результатів аналізу та взаємодію з зовнішніми сервісами. Адміністративна частина дозволяє виконувати ручні коригування процесів, здійснювати моніторинг роботи підсистеми, а також передбачає можливість внесення та оновлення даних та зображень, отриманих від сторонніх сервісів.

Сторонні сервіси це інформаційні системи та роботизовані платформи, що інтегровані в компанії багаторазового використання транспортних засобів. Ці платформи встановлені в автопарках та оснащені високоточними камерами, що автоматично здійснюють збори високоякісних зображень з зазначених сторін транспортного засобу. Отримані дані передаються до адміністративної панелі де вони заводяться в підсистему.

2.3 Вибір моделі виявлення ушкоджень у задачі ідентифікації

Під час аналізу предметної області ідентифікації ушкоджень мобільних транспортних засобів із застосуванням роботизованих платформ, поставлене завдання вибору відповідної до визначених умов моделі комп'ютерного зору, яка забезпечить високу точність ідентифікації дефектів, швидкість обробки зображень,

					ІК11.080БАК.006 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		20

адаптивність до різних типів пошкоджень і стабільну роботу за обмежених обчислювальних ресурсів.

Для виконання задач ідентифікації пошкоджень обрано архітектуру моделей сімейства YOLO, яка подається як одне з найшвидших та найефективніших рішень у сфері об'єктного детектування. Підхід цієї архітектури до задачі ідентифікації полягає в тому, що об'єкти на зображенні визначаються не через багатоетапну обробку, а шляхом одноразового проходження зображення через нейронну мережу, що дало назву методу «You Only Look Once» [6].

У межах сімейства YOLO в роботі [7] проаналізовано декілька актуальних архітектур, що спеціалізуються на задачах об'єктного детектування. Розглядалися моделі YOLOv8s, YOLOv10l та YOLOv11s, що представляють різні покоління та масштаби, кожна з яких має власні переваги й недоліки залежно від цілі використання (рис.2.2).

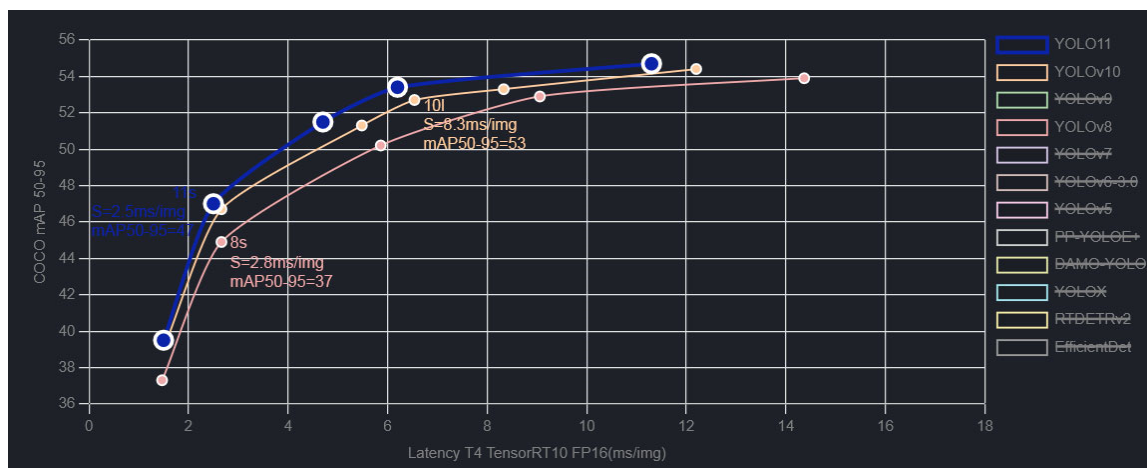


Рисунок 2.2 – Порівняння моделей YOLO

YOLOv8s є стабільною, легкою моделлю, що підходить для виконання задач у середовищах із обмеженими обчислювальними ресурсами. Вона демонструє високу швидкість інтерфейсу й здатна працювати на CPU або мобільних пристроях. Також YOLOv8s залишається однією з найпопулярніших моделей об'єктного детектування завдяки широкому поширенню, простоті інтеграції та великій кількості доступної документації. Попри її популярність, модель вважається застарілою, особливо на фоні новіших моделей, які забезпечують вищу точність і

кращу швидкість. Для задач, пов'язаних із виявленням дрібних і майже непомітних пошкоджень, використання YOLOv8s з її проблемами в точності може бути проблемою.

YOLOv10l навпаки, є потужнішою моделлю з глибшою архітектурою, яка забезпечує вищу точність навіть при поганих умовах зйомки наприклад, при складному фоні, поганому освітленні або нестандартних ракурсах. Але вона вимагає набагато більше ресурсів та часу, що може бути проблемою для задачі виявлення ушкоджень.

Оптимальним вибором між цими варіантами виявилась YOLOv11s. Це полегшена версія 11-ї генерації архітектури YOLO, яка поєднує доволі високу точність детекції з великою швидкістю обробки, що дозволяє ефективно виявляти пошкодження на зображеннях транспортних засобів у реальному часі, не потребуючи великого навантаження на сервер. У порівнянні з YOLOv8s модель 11s демонструє кращу точність на складніших прикладах, а в порівнянні з YOLOv10l хоч і просідає у точності, але значно швидше.

Виходячи з поставлених вимог до підсистеми, а саме помірне навантаження на сервер, доволі висока точність та велика швидкість ідентифікації, прийнято рішення використовувати модель YOLOv11s як основну для інтеграції в підсистему виявлення пошкоджень транспортних засобів.

2.4 Вибір мови програмування та середовища проектування підсистеми

У якості мови програмування для розробки серверної частини та навчання моделі автоматичної ідентифікації пошкоджень обрано об'єктно-орієнтовану мову програмування Python. Вона є однією з популярних мов у сфері штучного інтелекту, машинного навчання та обробки зображень. Її екосистема включає в себе потужні та добре документовані бібліотеки, такі як PyTorch, Ultralytics, OpenCV, які дозволяють реалізовувати складні модулі аналізу даних та візуальної аналітики. Також Python відзначається інтуїтивно зрозумілим синтаксисом, що дозволить

					ІК11.080БАК.006 ПЗ	Арк.
						22
Зм.	Арк.	№ докум.	Підпис	Дата		

зосередитись на розробці логіки та створенні моделі комп'ютерного зору, а не на тонкощах кодування [8].

Ще одним важливим аргументом на користь Python стала його кросплатформеність. Це дозволить розробляти та тестувати систему на різних операційних системах, а потім легко розгорнути її на сервері без значних змін у коді. Це значно спростить процес розгортання та масштабування, особливо враховуючи потенційну необхідність перенесення системи на більш потужне обладнання в майбутньому.

Так як підсистема ідентифікації пошкоджень складається з різних функціональних частин - серверної частини, Телеграм-бота, адміністративної панелі, моделі ідентифікації пошкоджень та бази даних, то для кожної з них треба обрати ключові середовища, бібліотеки та технології, що будуть використані в розробці.

2.4.1 Серверна частина і база даних

Головним підходом до побудови серверної частини є реалізація RESTful API. Це HTTP сервіс, що дозволяє зв'язати сервер з клієнтською частиною, а також інтегрувати різні компоненти через стандартизовані протоколи [9].

При аналізі визначено два ключові фреймворків для проєктування RESTful API. Такі як FastAPI та Tornado але для проблеми виявлення ушкоджень, де важлива швидкість, все ж таки вирішено обрати FastAPI.

FastAPI - це сучасний, швидкий вебфреймворк для створення API на мові програмування Python [10]. Маючи асинхронну модель виконання, він якраз орієнтований на розробку RESTful-сервісів, що робить його придатним для створення масштабованих серверних застосунків. Також однією з ключових переваг FastAPI є зручність створення HTTP-інтерфейсів завдяки інтегрованій підтримці OpenAPI (Swagger) для автоматичної генерації документації кожного методу. Це суттєво спрощує тестування та взаємодію з клієнтськими частинами.

					ІК11.080БАК.006 ПЗ	Арк.
						23
Зм.	Арк.	№ докум.	Підпис	Дата		

Для реалізації серверної частини обрано інтегроване середовище розробки PyCharm. Воно забезпечує зручне написання проєктів в середовищі Python, а саме вбудовані інструменти налагодження, підсвітку синтаксису та інтеграцію з системами контролю версій GIT. Також середовище підтримує легку інсталяцію бібліотек та запуск локального серверу через інтегрований термінал. [11].

Для роботи з реляційною базою даних MySQL, в Python коді, обрано технологію SQLAlchemy. Це потужна ORM-бібліотека на мові Python, що підтримує об'єктно-орієнтований підхід, завдяки якому стає можливим працювати з реляційними базами даних [12]. Вона забезпечує високорівневий інтерфейс для опису структури бази даних, створення таблиць, визначення зв'язків між ними та виконання складних запитів і транзакцій без необхідності писати сирі SQL-запити.

SQLAlchemy підходить завдяки своїй гнучкості та масштабованості. Вона підтримує стабільну інтеграцію з FastAPI, що спрощує побудову RESTful API та керування даними. Завдяки ORM-підходу всі операції з базою даних виконуються у вигляді роботи з класами та об'єктами Python, що підвищує зручність розробки, зменшує ймовірність помилок і полегшує підтримку коду. Це робить її ідеальним вибором для задачі, де потрібно надійно зберігати історію оренди, фото, виявлені пошкодження та інші важливі дані.

2.4.2 Telegram-бот і адміністративна панель

Аналіз популярних бібліотек дозволив визначити двох для написання Telegram ботів - Aiogram та бібліотеку від самих розробників месенджера Python-Telegram-Bot. Виходячи з того, що бібліотека Python-Telegram-Bot вже застаріла, обрано нову асинхронну бібліотеку Aiogram.

Загалом Aiogram це сучасний, простий, асинхронний фреймворк зі зручною документацією, який можна використовувати як для малих, так і великих проєктів без написання великої кількості зайвого коду. Головним плюсом у виборі Aiogram також є те, що він підтримує зручне управління станами FSM [13], завдяки чому можна створювати багатопарові діалоги, коли бот пам'ятає, на якому етапі

					ІК11.080БАК.006 ПЗ	Арк.
						24
Зм.	Арк.	№ докум.	Підпис	Дата		

знаходиться користувач. Це забезпечить швидку та надійну роботу підсистеми, що є дуже важливим у виявленні пошкоджень.

Проаналізувавши стек технологій для створення вебсторінок, обрано прості, а головне надійні технології, що дозволять без використання великої кількості ресурсів реалізувати адміністративну панель. Обрано базову мову розмітки вебсторінок HTML з доповненням її бібліотекою Jinja2 це шаблонізатор, що дозволить використовувати шаблони для застосування даних з сервера. В свою чергу, це набагато спростить створення вебінтерфейсів. Для стилізації та створення адаптивного дизайну обрано популярний CSS фреймворк Bootstrap, який також надає готові стилі та компоненти для швидкої розробки привабливих вебсторінок, сумісних з різними пристроями. Для покращення взаємодії з користувачем обрано технологію JavaScript це мова програмування, яка відповідає за інтерактивність на вебсторінках, вона дозволить додавати фільтри або складні реалізації, що покращить функціонал адміністративної панелі.

2.4.3 Модуль ідентифікації пошкоджень

Ultralytics обрано як основну бібліотеку для роботи зі моделлю комп'ютерного зору, оскільки вона надає готові інструменти для запуску навчання та застосування вибраної раніше моделі YOLOv11s та стабільну інтеграцію з PyTorch [14]. Вона дозволяє ефективно реалізувати обробку зображень, зокрема автоматичне отримання координат пошкоджень та типів дефектів.

До та після аналізу зображення треба обробити, тобто застосувати фільтрації фону, виділення контурів та побудови візуалізації результатів аналізу, це все можна зробити за допомогою технології OpenCV. Ця бібліотека має широку функціональність у сфері комп'ютерного зору, зокрема у роботі з зображеннями, масками, геометричними перетвореннями та кольоровими моделями .

Також як доповнення до OpenCV для базових операцій із зображеннями, таких як відкриття, збереження, перетворення форматів, обрано бібліотеку Pillow.

					ІК11.080БАК.006 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		25

Одним з її плюсів є простота інтеграції у Python-проекти та сумісність із бібліотеками FastAPI та Aiogram.

Як середовище навчання моделі штучного інтелекту обрано хмарне середовище Google Colab, оскільки цей етап потребує значних обчислювальних графічних ресурсів. Навчання нейронної мережі YOLOv11s включає великі обсяги обробки зображень і виконання таких операцій на звичайному локальному обладнанні займає дуже багато часу.

Google Colab вирішує цю проблему, надаючи безкоштовний обмежений доступ до GPU процесорів, що дозволяє ефективно проводити тренування моделей, не інвестуючи в дорогі обчислювальні станції. Середовище підтримує запуск Python-коду у форматі інтерактивних Jupyter-ноутбуків, має просту інтеграцію з Google Drive для збереження результатів і дозволяє зручно обмінюватися кодом. Colab також сумісний із бібліотекою Ultralytics, через яку проводяться роботи з архітектурою YOLO. Це додатково спростило процес налаштування, тренування та тестування моделі .

Для створення та ручної анотації набору даних обрано платформу Roboflow. Вона надає зручні засоби анотування зображень, розподілу датасету на тренувальну, валідаційну та тестову вибірки, а також застосування методів для підвищення якості навчання моделі. Roboflow підтримує експорт у форматі YOLO, що дозволило без проблем інтегрувати набори даних у середовище Google Colab через Google Drive для подальшого навчання моделі [15].

Висновок до розділу 2

У ході проектування архітектури підсистеми ідентифікації ушкоджень мобільних транспортних засобів проведено комплексний аналіз сучасних технологій, інструментів і методів, придатних для реалізації підсистеми. Визначено функціональні й нефункціональні вимоги до підсистеми, що разом із задачами стали основою для формування архітектури.

					ІК11.080БАК.006 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		26

На основі порівняльного аналізу різних моделей штучного інтелекту обґрунтовано вибір моделі комп'ютерного зору YOLOv11s як оптимальної для задачі виявлення пошкоджень транспортних засобів. Модель забезпечує високу точність детекції при збереженні необхідної швидкості обробки в реальному часі та помірному навантаженні на обчислювальні ресурси.

Запропоновано застосування клієнт-серверної архітектури з використанням FastAPI для побудови серверної частини, що забезпечує ефективну взаємодію між компонентами системи. Для зберігання даних обрано реляційну базу MySQL з використанням ORM-бібліотеки SQLAlchemy, що гарантує цілісність і узгодженість інформації.

Визначено переваги використання мови програмування Python як основної мови розробки, завдяки її широкій підтримці у сфері штучного інтелекту та багатому екосистемному оточенню. Для клієнтського інтерфейсу обґрунтовано вибір асинхронного фреймворку Aiogram для створення Telegram-бота, що забезпечує зручний і доступний канал взаємодії з кінцевими користувачами.

Для навчання моделі нейронної мережі обрано середовище Google Colab, яке дозволяє ефективно використовувати обчислювальні ресурси без необхідності локальних серверів. Підготовка та анотація навчального датасету виконуються за допомогою платформи Roboflow, що забезпечує якість навчальних даних.

Отримані результати проєктування дозволять автоматизувати процес ідентифікації ушкоджень, зменшити залежність від людського фактора, підвищити об'єктивність та прозорість перевірок. Це створює передумови для впровадження на ринку багаторазового використання мобільних транспортних засобів, забезпечуючи підвищення якості обслуговування та операційної ефективності сервісів.

					ІК11.080БАК.006 ПЗ	Арк.
						27
Зм.	Арк.	№ докум.	Підпис	Дата		

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ПІДСИСТЕМИ ІДЕНТИФІКАЦІЇ

3.1 Визначення структур даних та навчання моделі виявлення пошкоджень для задачі ідентифікації

3.1.1 Створення та анотація набору даних

Одним із важливих етапів створення підсистеми виявлення ушкоджень є підготовка і навчання моделі виявлення пошкоджень на зібраних і оброблених даних. Проаналізувавши сегмент вже створених наборів даних для задачі виявлення пошкоджень, виявлено, що у відкритому доступі майже немає підходящих наборів даних, які б відповідали вимогам підсистеми.

Більшість наборів даних містять дуже обмежену кількість зображень або мають невідповідну анотацію без важливої класифікації пошкоджень. Крім того, у багатьох випадках відсутня різноманітність світла та умов зйомки. Всі фото однотипні, що може погано вплинути на точність моделі виявлення пошкоджень.

Враховуючі всі ці проблеми та обмеження, прийнято рішення про створення спеціалізованого набору даних, що включає потрібні різноманітні класи пошкоджень, різні типи автомобілів та знімки в реальних умовах.

Спочатку зібрано велику кількість фотографій пошкоджених транспортних засобів як з відкритих наборів даних без анотацій, так і просто з інтернету або соціальних мереж. Загалом зібрано 1500 зображень пошкоджених машин, з яких після перегляду та видалення дублікатів вийшло 1200 унікальних фото.

Проведено ретельний аналіз з метою видалити фото з низькою роздільною здатністю, розмитістю, а також ті, на яких пошкодження були тотальними без можливості розпізнати. В результаті вийшло 1025 готових для анотації чистих зображень.

Наступним кроком у створенні набору даних стала анотація зображень. Анотація це процес розмітки об'єктів на зображеннях для подальшого його використання у задачах навчання моделей комп'ютерного зору. Під час розмітки виділяються потрібні об'єкти на зображеннях за допомогою спеціальних інструментів як наприклад, прямокутники (bounding boxes) або точки та

					ІК11.080БАК.006 ПЗ	Арк.
						28
Зм.	Арк.	№ докум.	Підпис	Дата		

класифікуються. Тобто готуються графічні приклади, на яких модель вчиться коректно ідентифікувати та класифікувати відповідні об'єкти.

Для ручної анотації використовувалась хмарна платформа створення наборів даних Roboflow. Завдяки зручним інструментам анотації з можливістю додавати зображення поступово та зберігати зображення в хмарному середовищі, завантажено та анотовано 1025 фото транспортних засобів. Процес анотації включав в себе ручне виділення кожного пошкодження за допомогою інструменту прямокутник (рис. 3.1).

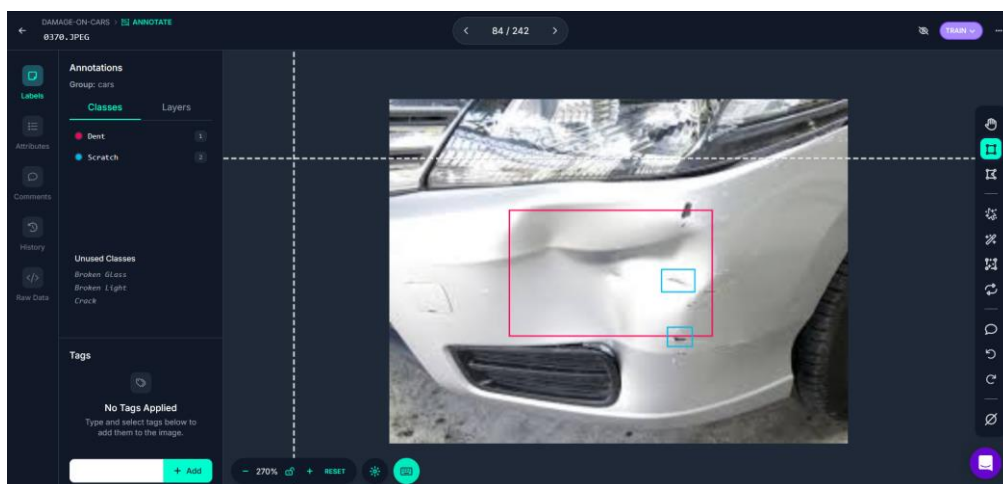


Рисунок 3.1 – Процес анотації в Roboflow за допомогою інструменту bounding boxes

Після виділення прямокутник класифікувався в уже створені класи пошкоджень, такі як scratch (подряпина), dent (вм'ятина), crack (тріщина), broken light (пошкоджена фара) та broken glass (пошкоджене скло). Всього анотовано 2402 унікальних пошкоджень (рис. 3.2)

					ІК11.080БАК.006 ПЗ	Арк.
						29
Зм.	Арк.	№ докум.	Підпис	Дата		

☰ Classes & Tags

Classes 5 Tags 0

What is a class?
 Lock Classes

COLOR	CLASS NAME	COUNT ↻
●	Broken Glass	266
●	Broken Light	127
●	Crack	92
●	Dent	605
●	Scratch	1 312

Рисунок 3.2 – Класи пошкоджень

Для досягнення стабільності та більшої точності під час навчання, набір даних поділено на три частини: тренувальну, валідаційну та тестову. Тренувальна вибірка, яка склала 73% від загального обсягу, використовується безпосередньо для навчання моделі, валідаційна 18% для налаштування гіперпараметрів і контролю процесу навчання, що запобігає перенавчанню, та тестова 9% для остаточної оцінки здатності моделі виявляти пошкодження на нових, раніше невідомих даних. Такий підхід до поділу набору даних підвищить стабільність моделі та забезпечить більш точні й надійні результати.

Ще одним етапом стала автоматична аугментація даних. Сервіс Roboflow надає автоматичну можливість трансформувати, прибрати шум, повернути зображення з набору даних. Використано масштабування до єдиного розміру 640px на 640px (рис.3.3) Таке масштабування дозволить уніфікувати розміри зображень, що полегшує та пришвидшує процес навчання моделі.



Рисунок 3.3 — Приклади масштабованих зображень з анотаціями у Roboflow

Після завершення анотації та аугментації набір даних був експортований у формат YOLOv11. Підтримка Roboflow цього формату дозволило безперешкодно інтегрувати набір даних із сучасними бібліотеками навчання нейронних мереж, зокрема з раніше обраною бібліотекою Ultralytics. У форматі YOLO зберігається компактна інформація про координати bounding boxes і класи пошкоджень, що пришвидшує процес обробки даних і значно спростило подальше навчання.

3.1.2 Навчання моделі в хмарному середовищі Google Colab

Після створення та анотації набору даних, наступним етапом стало безпосереднє навчання нейронної мережі з архітектурою YOLO, а саме YOLOv11s. Для вирішення цієї задачі використано хмарне середовище Google Colab, яке надає безкоштовний обмежений доступ до потужних графічних процесорів NVIDIA Tesla T4. Це дозволило проводити важкі та довгі тренування моделі штучного інтелекту віртуально, в умовах не потужної локальної системи.

Навчання здійснено за допомогою бібліотеки Ultralytics, яка має зручні технології для роботи з моделями архітектури YOLO. За допомогою об'єкта цієї бібліотеки завантажено нейронну модель yolov11s.pt. Модель вже здатна визначати базові об'єкти, такі як автомобілі, люди, тварини. Хоч ці класи не підходять для поставленої задачі, здатність моделі їх визначати дозволяє адаптувати її до конкретної задачі без необхідності навчання з нуля. Далі виконано донавчання моделі yolov11s.pt (рис. 3.4) на попередньо завантаженому на Google Drive, анотованому наборі даних пошкоджених транспортних засобів [16]. Доновчання проходило у стандартному форматі detect з попередньо натренованими вагами (pretrained=True), процес в свою чергу охоплював 136 епох із розміром батчу 16 використовуючи адаптивну стратегію навчальної швидкості з повільним спадом. Також застосовано методи регуляризації та покращення узагальнення, такі як випадкове горизонтальне віддзеркалення (flipr=0.5) та розширення зображень за допомогою HSV-аугментації.

```

Epoch  GPU_mem  box_loss  cls_loss  dfl_loss  Instances  Size
1/250   6G       2.289    5.313    2.346    1          640: 100%|██████████| 48/48 [04:17<00:00, 5.37s/it]
        Class  Images  Instances  Box(P)   R          mAP50  mAP50-95): 100%|██████████| 6/6 [00:05<00:00, 1.16it/s]

Epoch  GPU_mem  box_loss  cls_loss  dfl_loss  Instances  Size
2/250   6.76G    2.248    3.532    2.336    2          640: 100%|██████████| 48/48 [00:22<00:00, 2.16it/s]
        Class  Images  Instances  Box(P)   R          mAP50  mAP50-95): 100%|██████████| 6/6 [00:02<00:00, 2.24it/s]

Epoch  GPU_mem  box_loss  cls_loss  dfl_loss  Instances  Size
3/250   6.76G    2.24    3.456    2.331    0          640: 100%|██████████| 48/48 [00:21<00:00, 2.28it/s]
        Class  Images  Instances  Box(P)   R          mAP50  mAP50-95): 100%|██████████| 6/6 [00:03<00:00, 1.61it/s]

Epoch  GPU_mem  box_loss  cls_loss  dfl_loss  Instances  Size
4/250   6.76G    2.268    3.437    2.347    0          640: 100%|██████████| 48/48 [00:21<00:00, 2.27it/s]
        Class  Images  Instances  Box(P)   R          mAP50  mAP50-95): 100%|██████████| 6/6 [00:02<00:00, 2.03it/s]

```

Рисунок 3.4 — Приклади донавчання моделі Yolov11s у середовищі Google Colab

Після завершення тренувань моделі отримано показники якості, за допомогою вбудованого в бібліотеку YOLO методу model.val, та найкращу модель за всі епохи у директорії train. Збереження файлу найкращої моделі дозволило використати її на сервері при застосуванні на нових, невідомих даних.

На рис. 3.5 наведено схему процесу створення набору даних та донавчання моделі Yolov11s для ідентифікації ушкоджень транспортних засобів.

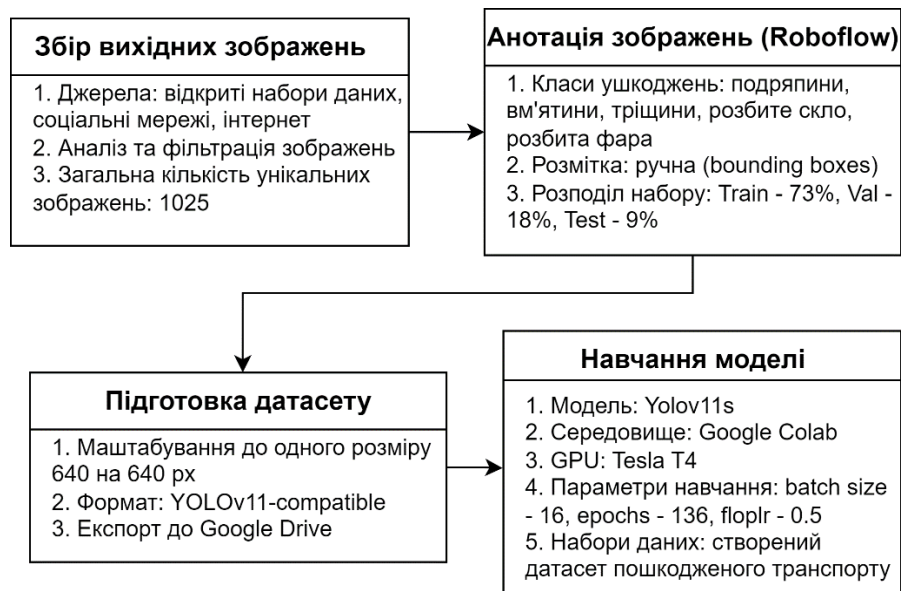


Рисунок 3.5 — Схему процесу створення набору даних та донавчання моделі

3.2 Структура функціональних модулів підсистеми

Підсистема автоматизованої ідентифікації ушкоджень мобільних транспортних засобів із застосуванням роботизованих платформ включає в себе чотири основні частини: Telegram-бот для взаємодії користувачів з підсистемою, вебпанель адміністраторів для керування та внесення даних, база даних MySQL для збереження всіх даних, які надходять та генерує підсистема, модель комп'ютерного зору для виявлення пошкоджень на зображеннях, серверна частина FastAPI, яка пов'язує всі частини в єдину підсистему. Також окремо підсистема отримує та використовує дані від стороннього сервісу та роботизованої платформи.

Підсистема працює наступним чином. Клієнтська частина сформована у вигляді Telegram-бота, який надає доступ до клієнтської логіки завдяки взаємодії з користувачем через текстові повідомлення та меню у вигляді інтерактивних кнопок. У структурі бота реалізовані обробники HTTP-запитів, що забезпечують ефективний зв'язок із серверною частиною шляхом відправлення запитів та отримання відповідей. Ця технологія дозволяє надійно та швидко передавати дані між клієнтською логікою і сервером.

Серверна частина побудована завдяки бібліотеці FastAPI і виконує роль RESTfull API, що приймає запити від Telegram-бота та інтерфейсу адміністратора. Основні компоненти серверу включають маршрутизатори, точки доступу, контролери, методи штучного інтелекту та інтерфейси взаємодії з базою даних. Маршрутизатори це окремі роутери (routers), кожен з яких відповідає за роботу з певною групою, а саме автомобілі (cars), користувачі (users), оренди (rentals), фото (photos), адміністратори (admins). Кожен маршрутизатор містить HTTP-методи (GET, POST, PATCH, DELETE) які зв'язують шляхи з контролерами, в яких реалізується бізнес-логіка отримання, створення, оновлення або видалення даних [17]. Методи бібліотеки Ultralitics викликають донавчену модель штучного інтелекту ідентифікації ушкоджень та аналізують зображення. В інтерфейсах взаємодії з базою даних MySQL описані моделі, які відображають основні сутності. В цих моделях використовується бібліотека SQLAlchemy, що забезпечує керування транзакціями, виконання запитів та роботу з підключенням до бази даних MySQL.

Адміністративна панель реалізована у вигляді вебінтерфейсу на FastAPI сервері. Вона надає адміністративний інтерфейс для редагування та внесення даних від сторонніх сервісів оренди та роботизованих платформ.

Отже, підсистема складається з програмного забезпечення клієнтської частини (Telegram-бота), серверної частини (FastAPI), бази даних (MySQL), інтегрованої моделі для автоматичного виявлення пошкоджень та стороннім сервісом багаторазового використання транспортних засобів, який надає дані про оренду та фото з роботизованих платформ. Взаємодія між компонентами відбувається через RESTful API та асинхронні повідомлення, причому зовнішня система слугує джерелом необхідної інформації, що інтегрується в розроблену підсистему

Структурна схема компонентів підсистеми наведено на кресленику ІК11.008БАК.006 Д1.

					ІК11.080БАК.006 ПЗ	Арк.
						34
Зм.	Арк.	№ докум.	Підпис	Дата		

3.2.1 Серверна частина

Для реалізації серверної частини підсистеми обрано архітектурний патерн MVC (Модель-Контролер-Вид), адаптований під сучасний асинхронний фреймворк FastAPI. Цей патерн використовують для чіткого розділу логіки на три основні компоненти, а саме обробка даних (Модель), управління запитами та відповідями (Контролер) та представлення даних користувачеві (Вид) [18]. Нижче представлено детальний опис кожного компонента в серверній структурі підсистеми:

- модель реалізована за допомогою ORM-бібліотеки SQLAlchemy, яка забезпечує опис структури даних, їх валідацію та взаємодію з реляційною базою даних MySQL. Моделі відповідають за відображення таблиць, ключових зв'язків між ними та бізнес-даних.

- контролери виконують основну бізнес-логіку вони обробляють запити клієнтів, працюють із моделями, здійснюють обчислення, перевірки, оновлення та видалення даних, а також викликають зовнішні сервіси, такі як модель штучного інтелекту для ідентифікації пошкоджень.

- представлення це логіка, зв'язана з представленням даних користувачу. У FastAPI вони мають вигляд HTTP-ендпоїнтів, що надають маршрути для взаємодії з клієнтськими застосунками, зокрема Telegram-ботом і адмінпанеллю. Для адміністративної панелі реалізовано шаблони HTML з використанням Jinja2, що забезпечує динамічне відображення даних.

Головною технологією серверної частини підсистеми став сучасний фреймворк FastAPI на мові програмування Python. Ця технологія забезпечує високу продуктивність та асинхронність коду з підтримкою розробки RESTful API. Також, через хорошу інтеграцію FastAPI з бібліотекою SQLAlchemy було розроблено на сервері модулі баз даних, які дозволяють ефективно керувати реляційними даними та забезпечують зручний інтерфейс для взаємодії з MySQL.

В цілому, серверна частина використовує бібліотеки, модулі, сервіси, наведені в таблиці 3.1.

					ІК11.080БАК.006 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		35

Таблиця 3.1 – Ключові бібліотеки серверної частини підсистеми

Назва	Опис
fastAPI	Основний фреймворк для створення API з підтримкою асинхронності, маршрутизації, обробки HTTP-запитів
sqlAlchemy	ORM для роботи з реляційною базою даних (MySQL)
pydantic	Бібліотека для валідації і опису схем даних (моделей запитів/відповідей)
jinja2	Шаблонізатор для рендерингу HTML-сторінок адміністративної панелі
reportLab	Бібліотека для генерації PDF-звітів
ultralytics	Бібліотека для роботи з моделями штучного інтелекту (YOLO) для аналізу зображень
cv2, OpenCV	Бібліотека для обробки зображень
pydantic	Бібліотека для валідації і опису схем даних

Програмне забезпечення серверної частини підсистеми представлено у вигляді файлової структури (рис. 3.6) в середовищі PyCharm. Ця структура повинна бути зрозумілою і легкою для використання іншими розробниками.

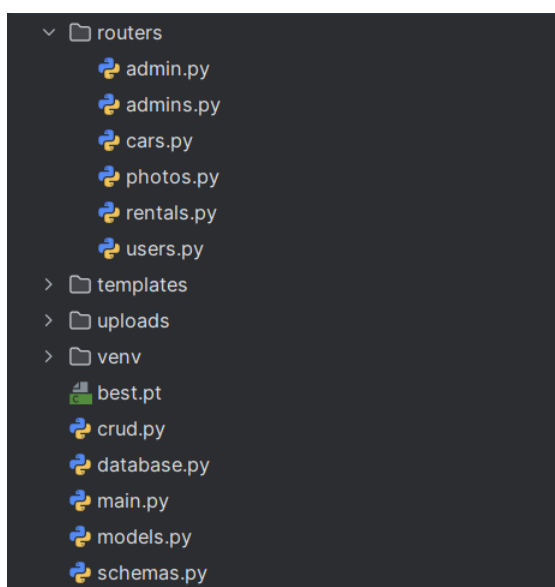


Рисунок 3.6 — Структура серверної частини підсистеми

Директорія routers містить модулі, кожен з яких відповідає за маршрутизацію HTTP-запитів, обробку бізнес-логіки та взаємодію з базою даних для різних сутностей системи:

- admin.py реалізує маршрути для адміністративної панелі, а саме сторінки входу, панель керування, редагування користувачів, машин, оренд тощо. Відповідає за вебінтерфейс адміністраторів;
- admins.py описує HTTP-ендпоїнти для роботи з адміністраторами, дозволяючи додавати, оновлювати та видаляти дані про адміністраторів;
- cars.py містить HTTP-ендпоїнти для роботи з автомобілями: отримання списку автомобілів, додавання, оновлення та видалення автомобілів з бази даних;
- photos.py описує HTTP-ендпоїнти, що обробляють завантаження та роботу з фотографіями оренд та обробку фото через штучний інтелект;
- rentals.py описує HTTP-ендпоїнти (рис. 3.7), які керують орендами, а саме створення оренди, оновлення статусів, отримання деталей;
- users.py описує HTTP-ендпоїнти для керування користувачами додавання, отримання, редагування і видалення;

```
<</get_rentals_by_tg_id/{tg_id}
@router.get(path="/get_rentals_by_tg_id/{tg_id}", response_model=List[RentalOut], summary="Оренди по Telegram ID")
def get_rentals_by_tg_id(tg_id: int, db: Session = Depends(get_db)):
    user = db.query(User).filter(User.tg_id == tg_id).first()
    if not user:
        return JSONResponse(status_code=404, content={"message": "User not found"})
    rentals = db.query(Rental).filter(Rental.user_id == user.user_id).all()
    return rentals
```

Рисунок 3.7 — Приклад реалізованого HTTP-ендпоїнту отримання оренди за tg_id

Директорія templates містить HTML-шаблони для адміністративної вебпанелі, які конструюються за допомогою бібліотеки Jinja2 сторінки панелі, оренд, машин, звітів.

Файл best.pt це і є модель штучного інтелекту (YOLO) для ідентифікації пошкоджень на фотографіях транспортних засобів. Реалізується у методах обробки зображень для автоматичного виявлення дефектів.

					ІК11.080БАК.006 ПЗ	Арк.
						37
Зм.	Арк.	№ докум.	Підпис	Дата		

Файл `crud.py` містить операції для роботи з базою даних (рис.3.8), так звані CRUD операції (Create, Read, Update, Delete). Ці операції є ключовими чинниками при взаємодії серверу з базою даних. Також вони надають додатковий функціонал, наприклад: авторизація, кешування, логування. В розроблюваній підсистемі вони створені для основних сутностей (користувачі, автомобілі, оренди, фото, адміністратори) і відповідають за бізнес-логіку, пов'язану з роботою з БД.

```
# Функція для створення нового автомобіля
def create_car(db: Session, make: str, model: str, year: int, license_plate: str, status: str):
    db_car = Car(make=make, model=model, year=year, license_plate=license_plate, status=status)
    db.add(db_car)
    db.commit()
    db.refresh(db_car)
    return db_car
```

Рисунок 3.8 — Приклад CRUD операції створення машини на сервері

Файли `database.py` та `models.py` налаштовують підключення до бази даних MsSQL через SQLAlchemy. Файл `models.py`, в свою чергу, описує ORM-моделі (рис. 3.9), що відповідають таблицям, полям, зв'язкам бази даних для SQLAlchemy.

```
class Rental(Base):
    __tablename__ = "rentals"

    rental_id = Column(Integer, primary_key=True, index=True)
    car_id = Column(Integer, ForeignKey("cars.car_id"))
    user_id = Column(Integer, ForeignKey("users.user_id"))
    rental_start = Column(DateTime)
    rental_end = Column(DateTime)
    has_new_damage = Column(Boolean, default=False)
    damage_check = Column(Boolean, default=False)
    is_ended = Column(Boolean, default=False)

    car = relationship("Car", back_populates="rentals")
    user = relationship("User", back_populates="rentals")
    rental_photos = relationship("RentalPhoto", back_populates="rental", cascade="all, delete-orphan")
```

Рисунок 3.9 — Приклад описаної моделі для SQLAlchemy на сервері

В модулі `schemas.py` описані схеми, які визначають структуру вхідних і вихідних даних API задля валідації даних у запитах та опису відповідей.

Файл `main.py` є основним модулем запуску FastAPI, він в собі описує підключення routers, запуск сервера на uvicorn та створення таблиць в базі даних якщо їх немає.

					ІК11.080БАК.006 ПЗ	Арк.
						38
Зм.	Арк.	№ докум.	Підпис	Дата		

3.2.2 Telegram-бот

Для клієнтської частини підсистеми виявлення ушкоджень транспортних засобів розроблено програмне забезпечення Telegram-бота з використанням сучасної бібліотеки Aiogram. Це дозволило здійснити асинхронну роботу з API Telegram та ефективно реалізувати діалогову логіку.

Також використано низку інших бібліотек та фреймворків наведених в таблиці 3.2

Таблиця 3.2 – Ключові бібліотеки та технології клієнтської частини підсистеми(Telegram-бот)

Назва	Опис
aiogram	Асинхронний фреймворк для створення Telegram-ботів на Python та обробки подій, повідомлень і FSM
asyncio	Стандартна бібліотека Python для асинхронного програмування та управління асинхронними операціями і запуск бота
requests	HTTP-клієнт для відправки HTTP-запитів до серверної частини (FastAPI API)
datetime	Обробка дат і часу оренд, порівняння дат
logging	Стандартна бібліотека Python для ведення журналу подій та логування подій і помилок у боті

Клієнтська частина Telegram-бот підсистеми виявлення ушкоджень містить структуру у вигляді єдиного програмного коду, який містить обробники команд, стани FSM та допоміжні функції.

Обробники команд та повідомлень (handlers) відповідають за обробку вхідних команд користувача, а також взаємодію з кнопками меню. У кожному обробнику реалізовано формування і відправку відповідних HTTP-запитів до серверної частини. Як приклад, обробник команди /start, який перевіряє

					ІК11.080БАК.006 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		39

zareestrovaniy korystuvach chi ni cherez HTTP-zapit do serveru. Stani FSM viznachayut po etapnu logiku vzaemodii pri reestracii korystuvacha ta zavантаженні фотографій під час огляду транспортного засобу. Допоміжні функції, які викликаються в хендлерах та створюють меню, клавіатури, а також користувальницьку логіку.

Для кращого розуміння структури роботи Telegram-бота розроблено UML діаграму діяльності бота. Діаграма діяльності (activity diagram) є частиною UML-діаграм, яка ілюструє послідовність дії та переходів між функціоналом під час взаємодії юзера з користувальницькою частиною [19]. Вона демонструє основні кроки від команди /start до завершення оренди, а також подальшої роботи з Telegram-ботом. Діаграму діяльності наведено на кресленику ІК11.008БАК.006 Д2.

Початковою дією при взаємодії користувача з Telegram-ботом є команда /start, яка ініціалізує перевірку, чи є користувач у системі. Якщо він вже зареєстрований, бот вітається та виводить головне меню з основною логікою. У разі ж відсутності користувача бот запускає метод реєстрації, а саме послідовний процес збору даних користувача завдяки серії питань. Після збору даних (ім'я, прізвище, email та телефон) бот надсилає ці дані до сервера, який записує їх у базу даних.

Далі людина отримує можливість взаємодіяти з логікою Telegram-бота серед, яких перегляд активної оренди, історії оренд, списку доступних автомобілів, а також контактної інформації служби підтримки.

Користувач може обрати в меню моя оренда кнопку огляд автомобіля під час якої бот послідовно приймає від нього серію фотографій транспортного засобу для подальшої обробки серверною частиною з використанням моделі штучного інтелекту.

Також передбачена функція завершення оренди, яка запускається при зроблених до цього усіх перевірках фото та ініціалізує перевірку нових пошкоджень на транспортному засобі порівняно з початковим станом. Якщо нові пошкодження не виявлено, оренда автоматично завершується. В іншому випадку користувач отримує повідомлення про необхідність звернення до служби підтримки.

					ІК11.080БАК.006 ПЗ	Арк.
						40
Зм.	Арк.	№ докум.	Підпис	Дата		

Кожен етап взаємодії завершується поверненням до головного меню, що забезпечує безперервний цикл користування ботом та дозволяє легко переходити між функціональними блоками без необхідності повторної авторизації.

3.2.3 Адміністративна панель

Основною ціллю розробки адміністративної панелі було створення інтуїтивно зрозумілого та зручного інтерфейсу. Це дозволяє адміністраторам швидко орієнтуватися в даних, отримувати зручні звіти та зручно вносити дані до системи.

Крім того, забезпечено можливість через адміністративну панель вносити дані та зображення, отримані від сторонніх сервісів оренди та роботизованих платформ. Це все реалізовано за допомогою шаблонизатора HTML сторінок Jinja2 та бібліотеки створення адаптивного дизайну Bootstrap. Усі використані технології та бібліотеки описані в таблиці 3.3

Таблиця 3.3 – Ключові бібліотеки та технології адміністративної частини підсистеми

Назва	Опис
jinja2	Бібліотека
bootstrap 5	Бібліотека для створення адаптивного та зручного інтерфейсу адміністративної панелі.
reportLab	Бібліотека для генерації PDF-звітів
csv	Бібліотека для експорту даних у форматі CSV
fastAPI	Технологія для обробки запитів до серверної частини.

Структурно адміністративна панель складається з сторінок адміністративного інтерфейсу, що надають інструменти для створення, редагування, видалення та перегляду даних про користувачів, транспортні засоби,

					ІК11.080БАК.006 ПЗ	Арк.
						41
Зм.	Арк.	№ докум.	Підпис	Дата		

оренди та результати аналізу ушкоджень. Ці сторінки формуються за допомогою шаблонів Jinja2 і відображаються у браузері. Також панель має можливість завантажувати та оновлювати дані і зображення, отримані від сторонніх сервісів та роботизованих платформ.

Для детального розуміння розробленої підсистеми, можна звернутися до сценаріїв використання (Use Cases), завдяки яким визначають послідовність подій та як саме проходить зв'язок між користувачем та системою. Ці Use Cases для зручності зображують на Use Case Diagrams такі діаграми допомагають зрозуміти, як користувач взаємодіє з системою, а також дозволяють чітко визначити ролі та сценарії використання адміністративної панелі та Telegram-бота. Використовуючи Use Cases діаграму, можна чітко зрозуміти, як користувач взаємодіє з користувальницькою логікою, даними, управлінськими функціями та сторонніми сервісами, що набагато спростить розуміння роботи підсистеми. Цю діаграму представлено на кресленику ІК11.008БАК.006 ДЗ. Як видно з діаграми, вебпанель дозволяє адміністратору легко додавати, редагувати та видаляти дані про транспорт, користувачів та оренд з підсистеми, а також генерувати звіти у зручних форматах. Користувач через Telegram-бота може переглядати інформацію поточної оренди, історії оренд, контактну інформацію та доступні транспортні засоби. Також може зручно завантажувати зображення транспорту для автоматичного аналізу пошкоджень за допомогою навченої моделі та завершувати оренду залежно від результатів аналізу.

Загалом, адміністратор має можливість керувати всіма аспектами підсистеми через вебпанель, а користувач може зручно користуватися закладеною логікою через Telegram-бота. Це дозволяє зберегти гнучкість і зручність у роботі з підсистемою.

3.3 Модель бази даних

Не менш важливим етапом в створенні та розгортанні підсистеми ідентифікації пошкоджень транспортних засобів стало проектування та створення

					ІК11.080БАК.006 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		42

бази даних. Оскільки, гарно продумана та структурована база даних, набагато спрощує розробку та виключає помилки з даними. Чітке розуміння структури даних дозволяє забезпечити правильну роботу підсистеми, зокрема у частинах обробки угод багаторазового використання, фотографій, пошкоджень і відстеження всіх пов'язаних змін.

Для цього обрано реляційну базу даних MySQL. Це дозволило використовувати транзакції та полегшило створення зав'язків між різними сутностями.

Розроблена модель бази даних підсистеми автоматичної ідентифікації ушкоджень транспортних засобів включає наступні шість сутностей, а саме:

- користувачі (users);
- автомобілі (cars);
- оренди (rentals);
- фотографії оренди (rental_photos);
- пошкодження (damages);
- адміністратори (admins).

Створена база даних налічує усі необхідні дані для стабільної роботи підсистеми. ER-діаграма бази даних зображена на кресленику ІК11.008БАК.006 Д4

Сутність «Користувачі (users)» містить дані про осіб, які орендують автомобілі. Включає контактну інформацію (телефон, email), унікальний Telegram ID для ідентифікації в системі, а також дату реєстрації користувача. Має зв'язок «один-до-багатьох» з таблицею оренд — один користувач може мати багато оренд.

Опис полів таблиці «Користувачі (users)» наведено в таблиці 3.4

Таблиця 3.4 — Опис полів таблиці «Користувачі (users)»

№	Назва поля	Тип поля	Опис
1	user_id	Число	Унікальний ідентифікатор користувача у системі
2	first_name	Рядковий тип	Ім'я користувача
3	last_name	Рядковий тип	Прізвище користувача

№	Назва поля	Тип поля	Опис
4	phone	Рядковий тип	Мобільний номер користувача
5	email	Рядковий тип	Електронна пошта користувача
6	tg_id	Число	Ідентифікатор користувача в Telegram
7	created_at	Дата/Час	Дата і час створення користувача

Сутність «Автомобілі (cars)» містить інформацію про транспортні засоби включно з маркою, моделлю, роком випуску, унікальним номерним знаком та статусом автомобіля (наприклад, доступний, в оренді, на ремонті). Має зв'язок «один-до-багатьох» з таблицею оренд — один автомобіль може бути орендований багато разів. Опис полів таблиці «Автомобілі (cars)» наведено в таблиці 3.5

Таблиця 3.5 — Опис полів таблиці «Автомобілі (cars)»

№	Назва поля	Тип поля	Опис
1	car_id	Число	Унікальний ідентифікатор автомобіля
2	make	Рядковий	Марка автомобіля
3	model	Рядковий тип	Модель автомобіля
4	year	Число	Рік випуску автомобіля
5	license_plate	Рядковий тип	Номерний знак автомобіля
6	status	Рядковий тип	Статус автомобіля (наприклад, доступний, в оренді)

Сутність «Оренди (rentals)» містить інформацію про кожну оренду, зокрема зв'язок між конкретним користувачем і автомобілем, терміни оренди, статус

					ІК11.080БАК.006 ПЗ	Арк.
						44
Зм.	Арк.	№ докум.	Підпис	Дата		

(активна чи завершена), а також позначки про наявність нових пошкоджень та факт перевірки. Має зв'язки «багато-до-одного» з таблицями користувачів і автомобілів, а також «один-до-багатьох» зі зображеннями оренди. Опис полів таблиці «Оренди (rentals)» наведено в таблиці 3.6

Таблиця 3.6 — Опис полів таблиці «Оренди (rentals)»

№	Назва поля	Тип поля	Опис
1	rental_id	Число	Унікальний ідентифікатор оренди
2	car_id	Число	Зовнішній ключ автомобіля, що орендується
3	user_id	Число	Зовнішній ключ користувача, який орендує автомобіль
4	admin_id	Число	Зовнішній ключ адміністратора
5	rental_start	Дата/Час	Дата початку оренди
6	rental_end	Дата/Час	Дата закінчення оренди
7	has_new_damage	Логічний тип	Чи є нові пошкодження на автомобілі
8	damage_check	Логічний тип	Чи перевірялися пошкодження перед орендою
9	is_ended	Логічний тип	Статус оренди (завершена чи активна)

Сутність «Фотографії оренди (rental_photos)» зберігає посилання та дані про зображення, зроблені до та після оренди, пов'язані з конкретною орендою через зовнішній ключ. Використовуються для аналізу пошкоджень за допомогою моделі. Має зв'язок «один-до-багатьох» з таблицею пошкоджень. Опис полів таблиці «Фотографії оренди (rental_photos)» наведено в таблиці 3.7

					ІК11.080БАК.006 ПЗ	Арк.
						45
Зм.	Арк.	№ докум.	Підпис	Дата		

Таблиця 3.7 — Опис полів таблиці «Фотографії оренди (rental_photos)»

№	Назва поля	Тип поля	Опис
1	photo_id	Число	Унікальний ідентифікатор фото оренди
2	rental_id	Число	Зовнішній ключ оренди, до якої належить фото
3	photo_path	Рядковий тип	Шлях до фото
4	photo_type	Рядковий тип	Тип фото (наприклад, до оренди чи після)
5	created_at	Дата/Час	Дата і час створення фото

Сутність «Пошкодження (damages)» зберігає дані про пошкодження, виявлені на фотографіях оренди. Для кожного пошкодження фіксується тип, ступінь впевненості моделі, координати пошкодженої області на фото. Має зв'язок «багато-до-одного» з таблицею фотографій оренди. Опис полів таблиці «Пошкодження (damages)» наведено в таблиці 3.8

Таблиця 3.8 — Опис полів таблиці «Пошкодження (damages)»

№	Назва поля	Тип поля	Опис
1	damage_id	Число	Унікальний ідентифікатор пошкодження
2	photo_id	Число	Ідентифікатор фото, на якому зафіксоване пошкодження
3	damage_type	Рядковий тип	Тип пошкодження (наприклад, подряпина, вм'ятина)
4	confidence	Число (Float)	Відсоток впевненості в точності визначеного пошкодження

№	Назва поля	Тип поля	Опис
5	box	Рядковий тип	Кординати прямокутника на фото, який визначає область пошкодження
6	created_at	Дата/Час	Дата і час створення запису пошкодження

Сутність «Адміністратори (admins)» зберігання даних про адміністраторів системи, які мають доступ до адміністративної панелі. Включає особисті дані, email (унікальний), пароль та дату створення облікового запису. Має зв'язок «один-до-багатьох» з таблицею оренд — один адміністратор може мати багато оренд. Опис полів таблиці «Адміністратори (admins)» наведено в таблиці 3.9

Таблиця 3.9 — Опис полів таблиці «Адміністратори (admins)»

№	Назва поля	Тип поля	Опис
1	admin_id	Число	Унікальний ідентифікатор адміністратора
2	first_name	Рядковий тип	Ім'я адміністратора
3	last_name	Рядковий тип	Прізвище адміністратора
4	email	Рядковий тип	Електронна пошта адміністратора
5	password	Рядковий тип	Пароль адміністратора (в зашифрованому вигляді)
6	created_at	Дата/Час	Дата і час створення адміністратора

3.4 Інтерфейси взаємодії із клієнтською та адміністративними частинами підсистеми

Результатом використання створеної підсистеми стає те, що її користувачі можуть зручно відстежувати поточний стан своєї сесії багаторазового використання та проводити автоматичну перевірку пошкоджень транспортного засобу за допомогою моделі їх виявлення, розгорнутої на серверній частині підсистеми, та порівнювати з даними, отриманими від компанії що надала транспортний засіб, де він був автоматично просканований за допомогою роботизованих платформ з високочутливими камерами.

Це означає, що користувачі транспортного засобу будуть контактувати з підсистемою тільки через користувацьку частину, яка автоматично встановить зв'язок з сервером та моделлю виявлення пошкоджень. Тому доцільно в першу чергу послідовно відтворити ті інтерфейси взаємодії—Telegram-бота, які забезпечують реалізацію цих функцій.

Взаємодія користувача з ботом відбувається через меню з кнопками швидкого вибору (рис. 3.10), що значно спрощує процес користування і мінімізує можливість помилок при введенні даних.

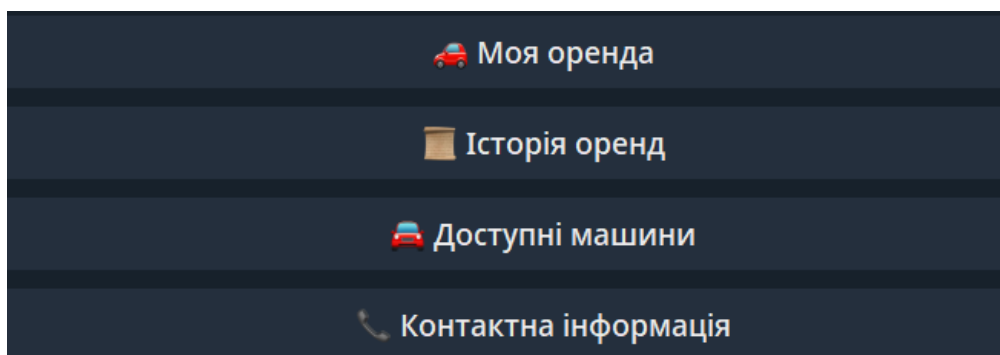


Рисунок 3.10 – Меню телеграм бота

Після отримання транспортного засобу користувач через Telegram-бота має можливість зробити фотоогляд автомобіля, скориставшись відповідним меню «Моя оренда»(рис 3.11). Надіслані фотографії автоматично передаються на серверну частину системи через Http-запит, де запускається аналіз за допомогою моделі штучного інтелекту. Модель виявляє на зображеннях наявні пошкодження різного типу, такі як подряпини, вм'ятини чи тріщини, і заносить результати у базу даних.

					ІК11.080БАК.006 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		48

Так фіксується початковий стан автомобіля на початку сесії багаторазового використання.



Рисунок 3.11 – Приклад завантаження зображень через Telegram-бот

По завершенню строку багаторазового використання транспортний засіб проходить перевірку у сервісі, що надає авто, за допомогою роботизованих платформ із високочутливими камерами. Отримані фотографії також завантажуються в підсистему через адміністративну панель, де адміністратор додає їх до відповідної сесії (рис. 3.12). Зображення проходять повторний аналіз на сервері за допомогою тієї ж моделі штучного інтелекту, та результати знову зберігаються у базі даних.

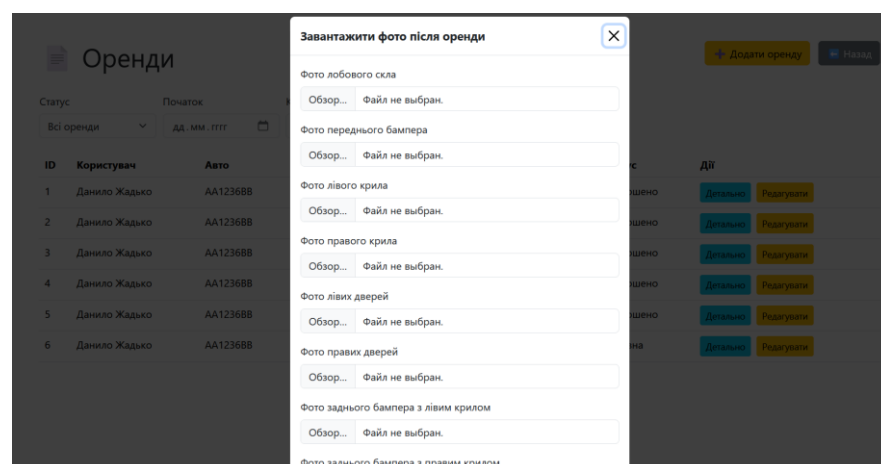


Рисунок 3.12 – Приклад завантаження зображень через адмін-панель

У Telegram-боті з'являється кнопка «Завершити оренду», яка запускає процес порівняння пошкоджень, зафіксованих на фото до початку сесії, з тими, що були виявлені після її завершення. Ця процедура реалізована у вигляді асинхронної функції `compare_damages`, яка надсилає запити до сервера, отримує списки пошкоджень із зображень «до» та «після», і порівнює їх за допомогою методу `Intersection over Union (IoU)`, блок-схему алгоритму функції `compare_damages` зображено на кресленику ІК11.008БАК.006 Д5. Для кожного типу пошкодження аналізується ступінь перекриття між його розташуванням на початковому та фінальному фото.

Якщо під час порівняння виявляються нові пошкодження, яких не було на початкових знімках (тобто не знайдено збігів за IoU більше 0.5), система позначає сесію багаторазового використання як таку, що містить ушкодження. У Telegram-боті користувач отримує повідомлення про виявлені пошкодження з переліком їх типів. Паралельно адміністратор отримує відповідне сповіщення в адміністративній панелі, де має можливість переглянути деталі та вручну підтвердити завершення сесії. (рис. 3.13-3.15)

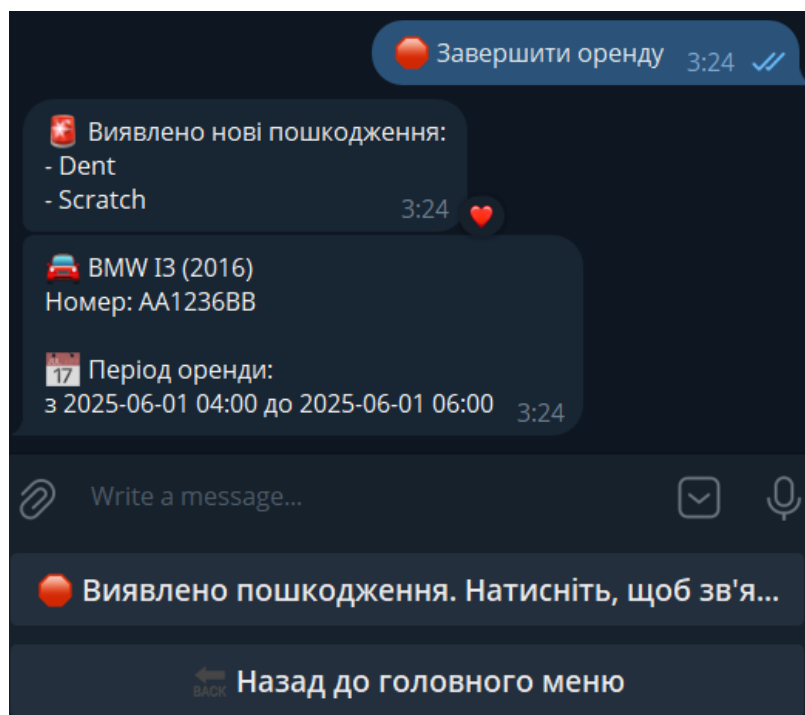


Рисунок 3.13 – Приклад завершення оренди з пошкодженнями

					ІК11.080БАК.006 ПЗ	Арк.
						50
Зм.	Арк.	№ докум.	Підпис	Дата		

Оренди

+ Додати оренду Назад

Статус Початок Кінець Пошук
Всі оренди ДД. ММ. ГГГГ ДД. ММ. ГГГГ Пошук по фамілії

ID	Користувач	Авто	Початок	Кінець	Статус	Дії
1	Данило Жадько	AA1236BB	2025-05-09 17:00:00	2025-05-09 18:00:00	Завершено	Детально Редагувати
2	Данило Жадько	AA1236BB	2025-05-09 12:21:00	2025-05-09 19:00:00	Завершено	Детально Редагувати
3	Данило Жадько	AA1236BB	2025-05-21 18:00:00	2025-05-21 20:00:00	Завершено	Детально Редагувати
4	Данило Жадько	AA1236BB	2025-05-21 19:00:00	2025-05-21 21:01:00	Завершено	Детально Редагувати
5	Данило Жадько	AA1236BB	2025-05-21 20:00:00	2025-05-21 23:00:00	Завершено	Детально Редагувати
6	Данило Жадько	AA1236BB	2025-06-01 04:00:00	2025-06-01 06:00:00	Активна (з пошкодженням)	Детально Редагувати

Рисунок 3.14 – Приклад сповіщення про пошкодження

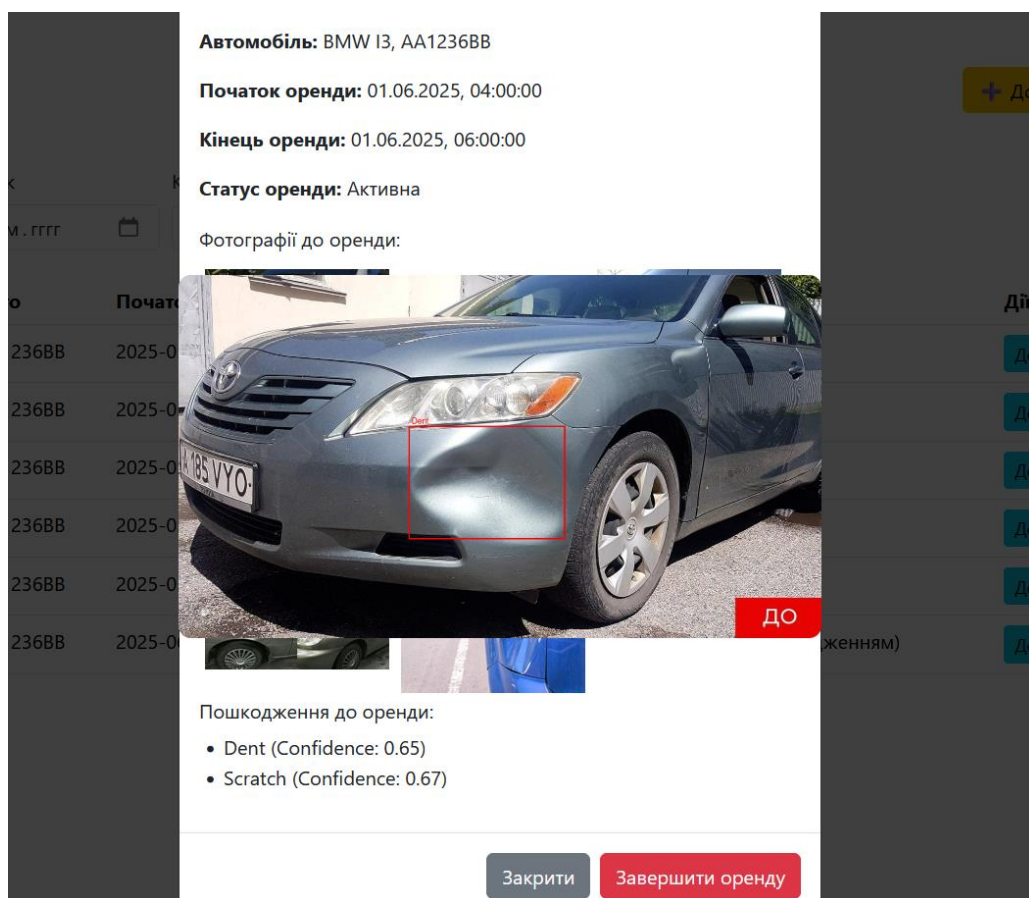


Рисунок 3.15 – Перегляд детальної інформації та завершення сесії з пошкодженням через адмін-панель

Якщо ж нових пошкоджень не виявлено, сесія багаторазового використання автоматично позначається як успішно завершена без додаткових проблем, і користувач отримує відповідне підтвердження через бот.

Такий підхід забезпечує прозорість і точність контролю стану транспортного засобу при багаторазовому використанні. Завдяки автоматичному аналізу та порівнянню зображень мінімізується людський фактор, а участь адміністратора у разі виявлення пошкоджень дозволяє здійснювати якісну перевірку та уникати можливих помилкових рішень.

Діаграму бізнес процесу підсистеми від Telegram-бота до моделі виявлення пошкоджень та назад наведено на рис. 3.16.

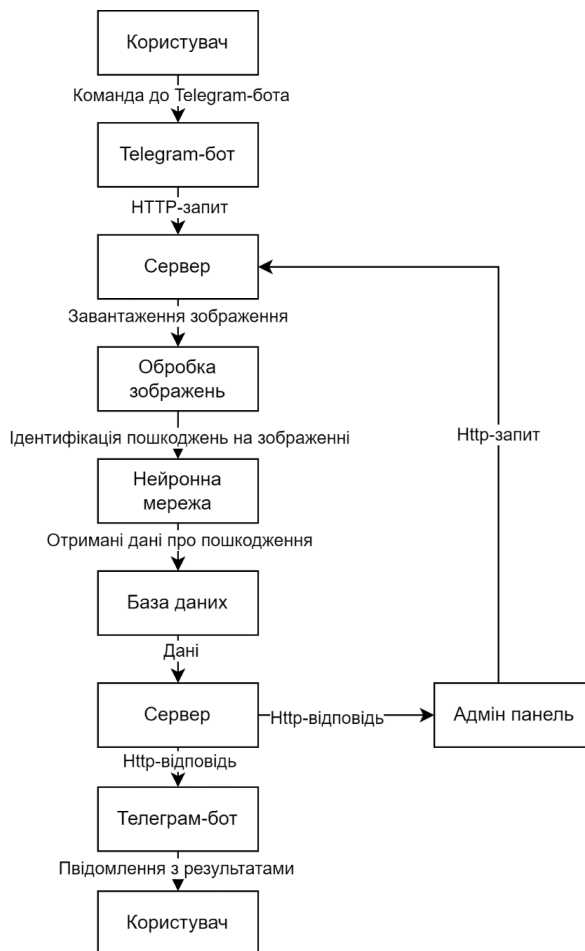


Рисунок 3.16 – Діаграма бізнес процесу підсистеми

Висновок до розділу 3

У процесі програмної реалізації підсистеми ідентифікації ушкоджень мобільних транспортних засобів створено програмне рішення яке включає серверну частину, клієнтські інтерфейси, модель штучного інтелекту та реляційну базу даних.

На етапі підготовки моделі штучного інтелекту сформовано власний навчальний набір даних, який включає понад 2000 анотованих зображень транспортних засобів з різними типами пошкоджень. Анотація виконувалася з використанням сучасних інструментів, що дозволило детально виявити області ушкоджень за класами: подряпини, вм'ятини, тріщини. Модель YOLOv11s навчено у хмарному середовищі Google Colab з використанням бібліотеки Ultralytics.

Серверна частина підсистеми реалізована з використанням фреймворку FastAPI на мові Python. Вона забезпечує централізовану обробку запитів, управління бізнес-логікою та взаємодію з базою даних і моделлю штучного інтелекту. Розроблено реляційну базу даних на основі MySQL. Структура бази даних включає п'ять основних сутностей: користувачі, транспортні засоби, оренди (використання), фотографії, виявлені пошкодження. Така структура забезпечує зручне зберігання історичних даних, підтримує зв'язки між об'єктами та дозволяє швидко здійснювати вибірки для аналізу. Для взаємодії з базою даних використано ORM-бібліотеку SQLAlchemy, що спрощує розробку і підтримку системи.

Також реалізовано клієнтські інтерфейси у вигляді Telegram-бота (на базі фреймворку Aiogram) та адміністративної панелі. Telegram-бот забезпечує зручний інтерфейс для користувачів, дозволяє завантажувати фотографії, переглядати стан оренди та результати аналізу. Адміністративна панель надає розширені функції для керування даними, перегляду історії використання, виявлених пошкоджень, а також для генерації звітів у форматах PDF і CSV.

					ІК11.080БАК.006 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		53

4 ТЕСТУВАННЯ І ОЦІНКА ЕФЕКТИВНОСТІ ПІДСИСТЕМИ ІДЕНТИФІКАЦІЇ

4.1 Оцінка точності і швидкодії навченої моделі виявлення пошкоджень

Важливим етапом при створенні моделі є її оцінка. Це дозволяє зрозуміти, наскільки ефективно модель пройшла навчання та як добре вона зможе виявляти пошкодження на невідомих її зображеннях. Оцінка моделі дає також змогу зрозуміти та виявити основні недоліки, які потребують вдосконалення, наприклад, зміна методу навчання моделі або оновлення набору даних.

Оцінка здійснюється за допомогою спеціальних метрик, а саме: точність (accuracy), яка вказує скільки правильних передбачень зробила модель порівняно з усіма; точність виявлення (precision), що вказує скільки позитивних прикладів насправді були правильними; відгук (recall), який дає зрозуміти скільки з усіх реальних позитивних прикладів було виявлено; F1-міра, що є середнім між точністю виявлення та відгуком; середнє значення точності (mAP), яка є основною метрикою архітектури YOLO [20]. Всі вони показують як добре та швидко модель навчилася класифікувати та виявляти об'єкти на зображеннях.

У випадку моделі штучного інтелекту, яка вирішує проблему ідентифікації ушкоджень транспортних засобів, здійснено оцінку на тестовій вибірці з використанням стандартних метрик комп'ютерного зору, а саме точність (precision), повнота (recall), F1-міра та середнє значення точності (mAP) при порогах IoU 0.5 та 0.5–0.95. Тестування проводилось автоматично під час навчання в хмарному середовищі Google Colab та надало детальні графіки та діаграми для подальшого аналізу та оцінки поведінки навченої моделі Yolov11s.

На рис. 4.1 представлена матриця неточностей (confusion matrix) яка, демонструє частоту правильних та помилкових класифікацій пошкоджень, що входять до кожної з категорій. У контексті даної задачі ідентифікації пошкоджень транспортних засобів кожен клас відповідає певному типу пошкоджень, а саме: подряпина (scratch), тріщина (crack), пошкожене скло (broken glass) або вмятина (dent).

					ІК11.080БАК.006 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		54



Рисунок 4.1 — Confusion matrix

Матриця неточностей дозволила побачити, скільки разів модель штучного інтелекту класифікувала зображення певного типу пошкодження, а також скільки вона помилилася у виборі. Кількість правильних класифікацій можна побачити на діагоналі матриці, в той час як помилкові поза діагоналлю. Чим більше число правильних розпізнавань, тим краще справляється із завданням модель.

Найбільш упевнене розпізнавання виявилось у класів подряпина (scratch), пошкоджене скло (broken glass) та вмятина (dent). Це означає, що модель змогла точно розпізнати ці пошкодження завдяки чітким візуальним ознакам та великій кількості цих пошкоджень в наборі даних. Тим не менш клас тріщина (crack), має найменшу кількість вірних спрацювань, через велику схожість з класом подряпини та невеликою кількістю прикладів в наборі даних. Також визначено проблему неправильної ідентифікації подряпин, модель знаходить подряпини на задньому фоні зображень. Щоб вирішити цю проблему, було сформульовано умову: робити знімки транспортного засобу впритул, без заднього фону. Це зменшило випадки, коли модель плутала подряпини з заднім фоном.

На рис. 4.2, 4.3 показано графіки відношення точності (Precision-Confidence) і F1-міри (F1-Confidence) від рівня впевненості моделі. Ці метрики допомагають оцінити якість роботи підсистеми ідентифікації пошкоджень транспортних засобів, а саме зрозуміти наскільки модель впевнена у своїх прогнозах та наскільки вони точні і повні.

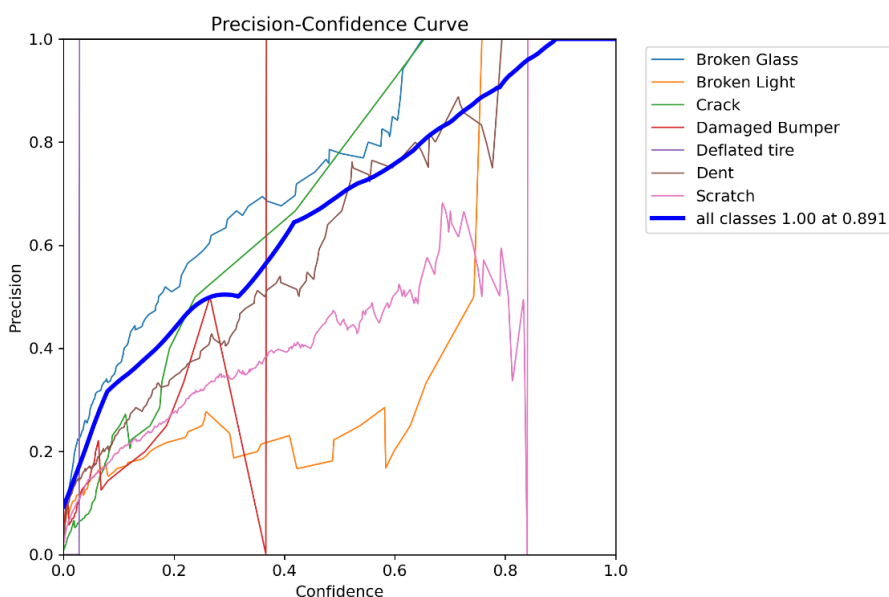


Рисунок 4.2 — Графік залежності точності Precision-Confidence

Графік Precision-recall це крива, яка відображає загальну продуктивність моделі для кожного класу пошкоджень. Вона ілюструє, наскільки ефективно модель може виявити позитивні пошкодження, не жертвуючи точністю. Найкращий показник показує клас пошкожене скло (broken glass), точність якого при порозі 0.5 склала 0.485. Це означає, що модель успішно ідентифікує пошкодження цього типу з високою надійністю.

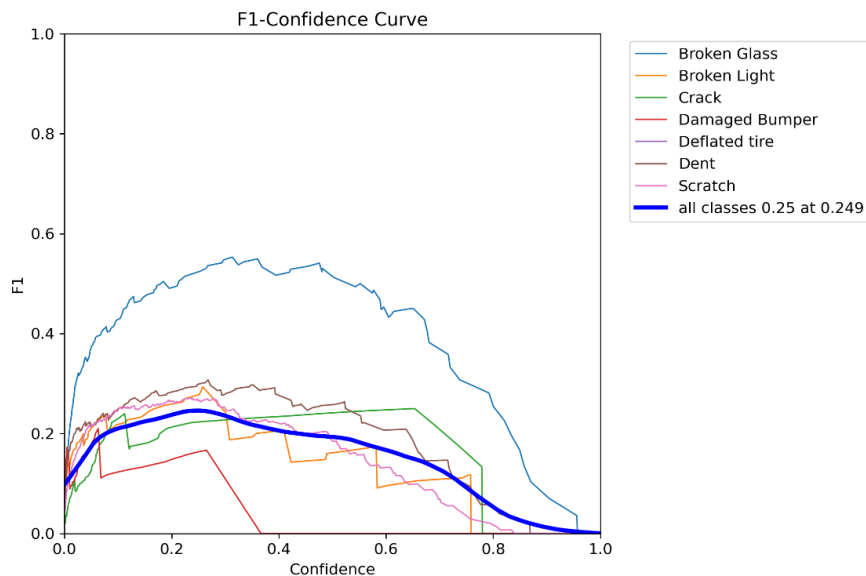


Рисунок 4.3 — Графік F1-міри (F1-Confidence)

За графіком F1-confidence можна зрозуміти співвідношення між точністю та повнотою на різних порогах впевненості. З графіку видно, що найкраще балансування метрик, максимального значення F1-міри, досягається при впевненості 0.25. Це означає, що на цьому рівні модель оптимально поєднує мінімізацію хибних спрацьовувань і максимізацію виявлених пошкоджень. Для деяких класів оптимальні значення рівня впевненості є нижчими, що свідчить про особливості їх візуалізації та нестачу прикладів в наборі даних.

Загалом ці метрики показали, що модель має здатність адаптуватися під різні умови та класи пошкоджень. Це дозволило підвищити ефективність виявлення ушкоджень і мінімізувати кількість помилкових спрацьовувань, що є критично важливим для практичної роботи підсистеми.

Також отримано графік втрат, який показує зміну значень на валідаційних і тренувальних вибірках протягом усього процесу навчання (рис.4.4). Поступове і стабільне зменшення значень функцій втрат на тренувальних та валідаційних вибірках свідчить про стабільне навчання моделі без явного перенавчання. Значення метрики mAP 0.5-0.95 становило приблизно 0.06, хоча це значення здається низьким, воно є типовим для складних задач, у яких класи важко розрізнити через незначні візуальні відмінності.

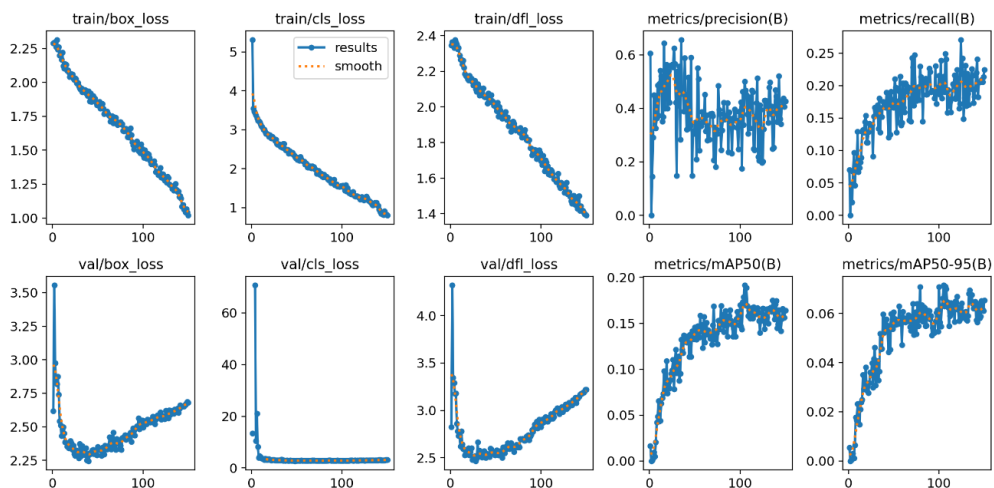


Рисунок 4.4 — Графік функцій втрат

Також, окрім автоматизованої оцінки, проведено ручне тестування навченої моделі виявлення пошкоджень шляхом аналізу контрольного зображення. Для цього фотографію транспортного засобу з ушкодженням було передано у підсистему через API.

В результаті аналізу, навченої моделі ідентифікації ушкоджень, отримано зображення з виділенням та класифікацією зон пошкоджень. На рис. 4.5 представлено вихідне зображення без розмітки, а на рис.4.6 результати роботи моделі.

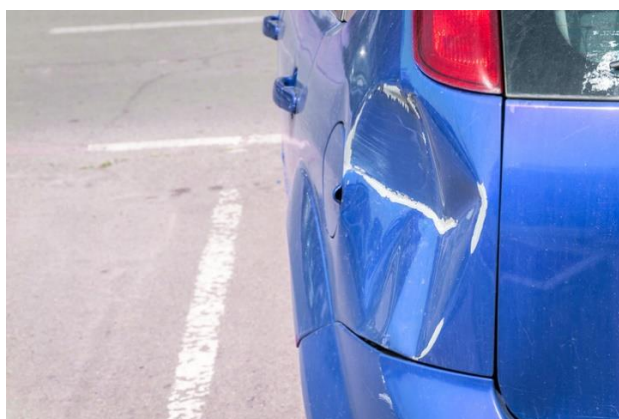


Рисунок 4.5 — Вихідне зображення

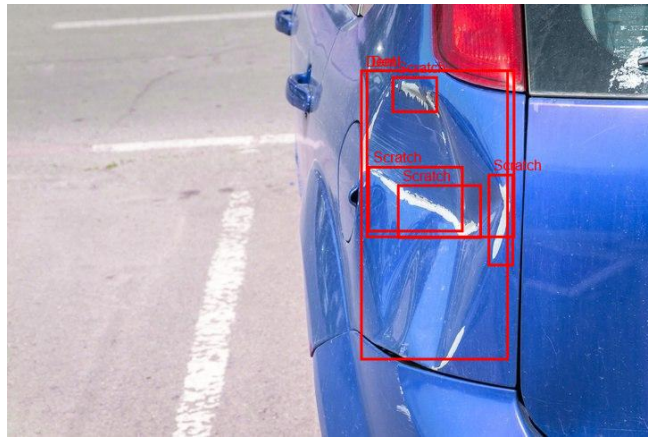


Рисунок 4.6 — Результат ідентифікації

Аналіз функцій втрат, показників mAP та ручного тестування підтверджує, що навчання моделі пройшло успішно, і отримані результати відповідають складності поставленої задачі, закладаючи надійний фундамент для подальшої оптимізації та впровадження підсистеми в реальних умовах.

4.2 Тестування функціоналу адміністративної панелі підсистеми

При першому відкритті адміністративної панелі працівник сервісу потрапляє на сторінку авторизації. У випадку неправильних даних форма сповістить про проблеми (рис. 4.7)

A screenshot of a web form titled "Вхід адміністратора" (Administrator Login). At the top, a pink error message reads "Невірний email або пароль" (Incorrect email or password). Below this are two input fields: "Email" with the placeholder "Введіть email" and "Пароль" with the placeholder "Введіть пароль". At the bottom of the form is a blue button labeled "Увійти" (Login).

Рисунок 4.7 Помилка авторизації

При введенні правильних даних користувач отримує доступ до головної панелі (рис.4.8), що містить загальну інформацію та посилання на основний функціонал. Перевірено, що без авторизації неможливо потрапити до інших сторінок.

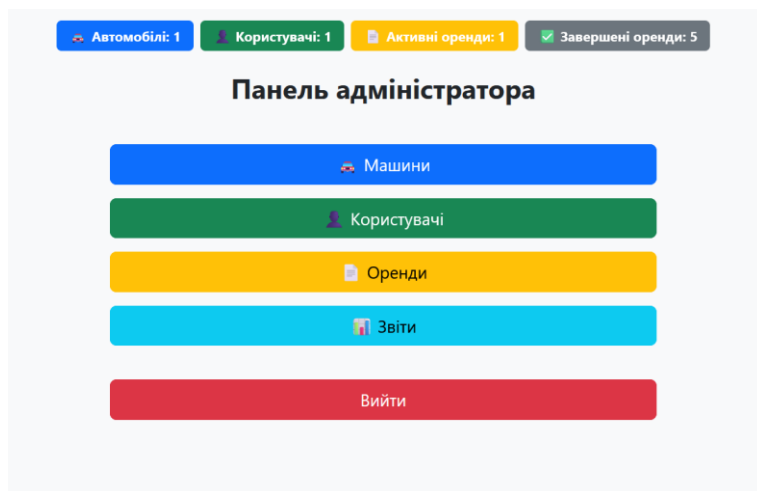


Рисунок 4.8 Головний дашборд адмін-панелі

При обиранні кнопок “Машин”, “Користувачі”, “Оренд” ми потрапляємо до таблиць з цими даними. Протестовано коректне відображення даних та роботу фільтрів на прикладі таблиці багаторазового використання (рис. 4.9)

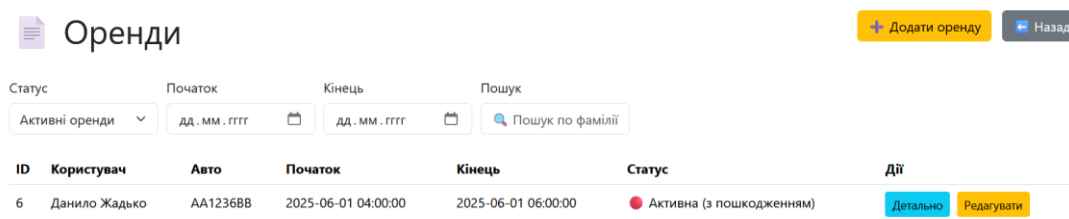


Рисунок 4.9 Таблиця багаторазового використання адмін-панелі

Щоб додати новий запис, треба натиснути кнопку «Додати». Відкривається форма з налаштованою варіацією створення нового запису, в випадку невірною введення отримуємо повідомлення(рис. 4.10).

+ Створити нову оренду Назад

Користувач

Данило Жадько (ID: 1) ▼

Автомобіль

ццк ▼

No results found

дд . мм . гггг , -- : -- 📅

Кінець оренди

дд . мм . гггг , -- : -- 📅

+ Створити оренду

Рисунок 4.10 Форма створення нової сесії адмін-панелі

4.3 Тестування функціоналу Telegram-бота підсистеми

При першому запуску користувач взаємодіє з Telegram-ботом через головне меню, що містить кнопки швидкого вибору. Меню забезпечує інтуїтивну навігацію та спрощує доступ до основних функцій, також при спробі написати невідоме повідомлення бот ніяк не реагує та надає головне меню (рис. 4.11).

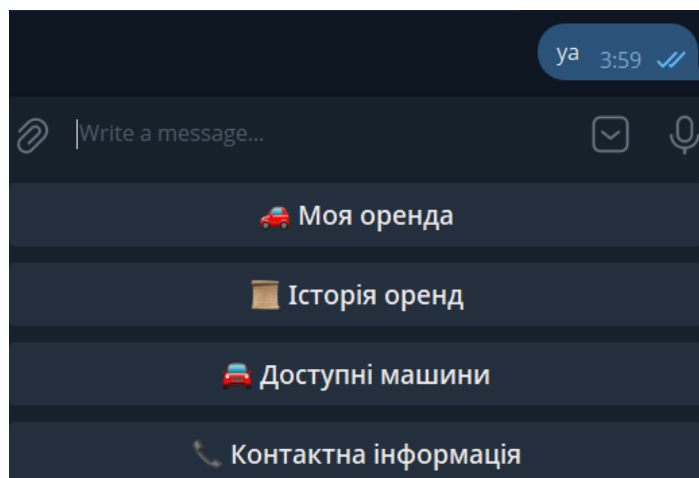


Рисунок 4.11 Тестування головного меню

					ІК11.080БАК.006 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		61

Під час тестування було детально перевірено роботу усіх кнопок у меню бота, а також основних функціональних блоків взаємодії. Кожне меню, представлене користувачу, відображається коректно, а кнопки виконують закладені функції без помилок або збоїв. Для візуального підтвердження стабільності роботи системи було зроблено зображення основної логіки, які демонструють послідовність дій користувача і відповіді бота рис. 4.12, 4.13.

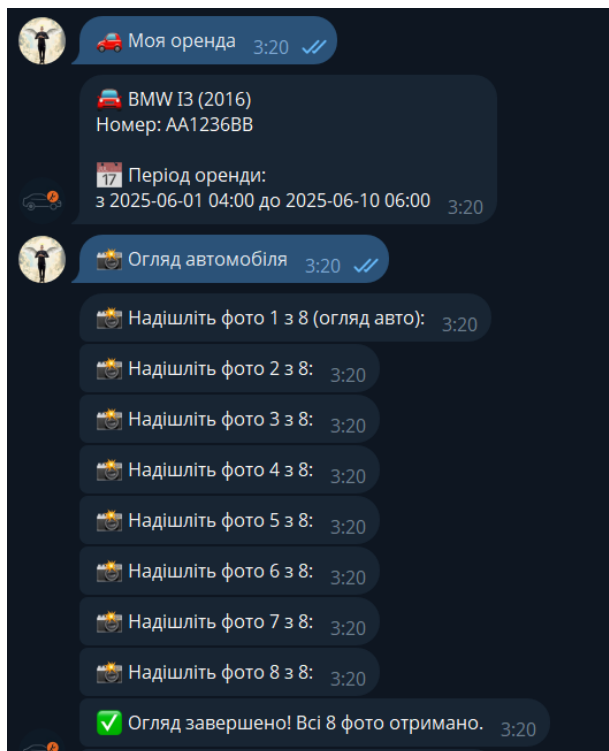


Рисунок 4.12 Тестування огляду транспортних засобів

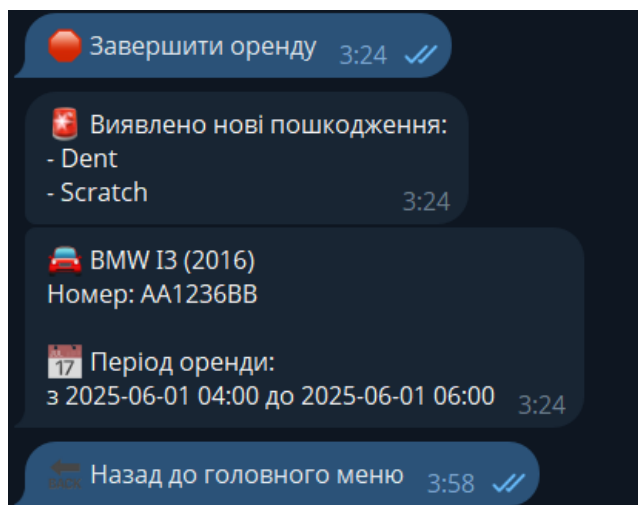


Рисунок 4.13 Тестування завершення сесії

					ІК11.080БАК.006 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		62

Висновок до розділу 4

У розділі проведено всебічне тестування функціональних можливостей підсистеми та здійснено оцінку ефективності її роботи. Проаналізовано точність і швидкодію навченої моделі YOLOv11s на основі тестових та реальних зображень транспортних засобів. За результатами тестування встановлено, що модель демонструє високу точність виявлення ушкоджень різного типу (подряпини, вм'ятини, тріщини), із середнім значенням показника precision понад 92 %, що підтверджує її практичну придатність для використання у виробничих умовах.

Досліджено роботу адміністративної панелі підсистеми. Перевірено коректність авторизації, доступу до основних функціональних розділів, а також можливість перегляду, редагування та формування звітів за даними оренд. Виявлено, що всі ключові сценарії роботи адміністративної панелі відповідають заданим вимогам. Інтерфейс забезпечує зручне управління даними та дозволяє оперативно контролювати процеси виявлення ушкоджень.

Також проаналізовано функціональність Telegram-бота. Перевірено повний цикл взаємодії користувача з підсистемою. Виявлено, що підсистема коректно обробляє запити користувачів та забезпечує інтеграцію з серверною частиною без затримок. Особливо відзначено, що Telegram-бот ефективно відфільтровує некоректні повідомлення користувачів, що є його перевагою порівняно з типовими рішеннями у цій сфері.

					ІК11.080БАК.006 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		63

ВИСНОВКИ

У ході виконання дипломного проєкту всебічно проаналізовано сучасний стан та тенденції розвитку ринку багаторазового використання мобільних транспортних засобів в Україні та світі. Встановлено, що одним із найбільш актуальних напрямів удосконалення цифрових сервісів у цій сфері є автоматизація процесів ідентифікації ушкоджень транспортних засобів з метою підвищення прозорості, точності та швидкості обслуговування клієнтів.

Досліджено сучасні підходи до вирішення задачі автоматичного виявлення пошкоджень із застосуванням технологій комп'ютерного зору та роботизованих платформ. Виявлено обмеження традиційних методів, що ґрунтуються на ручному огляді, та визначено переваги використання нейронних мереж для автоматизації цього процесу. На основі проведеного аналізу запропоновано створення автоматизованої підсистеми ідентифікації ушкоджень із використанням моделі виявлення пошкоджень YOLOv11s, що забезпечує високу точність та швидкість обробки зображень.

Розроблено архітектуру підсистеми, яка включає серверну частину на базі FastAPI, Telegram-бот для кінцевих користувачів, адміністративну вебпанель для внутрішнього керування даними, а також реляційну базу даних MySQL із п'ятьма основними сутностями: користувачі, транспортні засоби, оренди, фотографії та виявлені пошкодження. Створено власний навчальний набір даних з понад 2000 анотованих зображень, на основі якого проведено навчання моделі в середовищі Google Colab. Результати тестування підтвердили високу ефективність запропонованої моделі, з показником точності виявлення деяких класів пошкоджень понад 92 %.

У результаті тестування програмного комплексу встановлено, що підсистема коректно реалізує основні сценарії взаємодії, ефективно інтегрує різні компоненти системи та забезпечує автоматичне виявлення нових ушкоджень із мінімальним залученням людського фактора. Відзначено переваги розробленого Telegram-бота,

					ІК11.080БАК.006 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		64

який, на відміну від типових рішень, надійно фільтрує некоректні повідомлення та забезпечує інтуїтивно зрозумілий інтерфейс для користувачів.

Функціонально запропоноване рішення може бути впроваджено у сервіси багаторазового використання транспортних засобів, зокрема у поєднанні з роботизованими платформами збору зображень. Використання підсистеми дозволит підвищити точність і швидкість перевірки стану автомобілів, оптимізувати операційні витрати сервісу, зменшити кількість спірних ситуацій із клієнтами та підвищити рівень довіри до послуг.

Модульна структура архітектури підсистеми легко масштабується під потреби конкретного сервісу, дозволяє підключати додаткові сервіси, реалізовувати нові сценарії використання та розширювати функціональність без суттєвих змін у базовому коді. Це робить систему не лише ефективною у поточній конфігурації, а й перспективною з точки зору довгострокового використання, розвитку та комерціалізації.

					ІК11.080БАК.006 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		65

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Getmancar Blog – Оренда авто проти власного авто [Електронний ресурс] – Режим доступу: <https://getmancar.com/uk/blogs/rent-a-car-vs-owning>
2. Sixt – Міжнародний сервіс оренди авто [Електронний ресурс] – Режим доступу: <https://www.sixt.com/>
3. Getmancar – Часті запитання [Електронний ресурс] – Режим доступу: <https://getmancar.com/uk/faq>
4. Zfort Group – Дослідження комп'ютерного зору у 2024 році: вплив штучного інтелекту. [Електронний ресурс] – Режим доступу: <https://www.zfort.com.ua/blog/doslidzhennya-kompyuternogo-zoru-u-2024-roci-vpliv-shtuchnogo-intelektu-na>
5. CNN – Convolutional Neural Networks for Visual Recognition [Електронний ресурс] – Режим доступу: <https://cs231n.github.io/convolutional-networks/>
6. V7 Labs – Пояснення роботи YOLO [Електронний ресурс] – Режим доступу: <https://www.v7labs.com/blog/yolo-object-detection>
7. Ultralytics YOLO – Порівняння моделей YOLO [Електронний ресурс] – Режим доступу: <https://docs.ultralytics.com/ru/compare/>
8. Python and AI: Why Python Is Good for Artificial Intelligence and Machine Learning [Електронний ресурс] – Режим доступу: <https://djangostars.com/blog/why-python-is-good-for-artificial-intelligence-and-machine-learning/>
9. REST API Tutorial [Електронний ресурс] – Режим доступу: <https://restfulapi.net/>
10. FastAPI – Modern Web Framework for Python [Електронний ресурс] – Режим доступу: <https://fastapi.tiangolo.com/>
11. PyCharm – Features, Advantages & Disadvantages [Електронний ресурс] – Режим доступу до ресурсу: <https://www.geeksforgeeks.org/what-is-pycharm-features-advantages-disadvantages/>
12. SQLAlchemy – The Database Toolkit for Python [Електронний ресурс] – Режим доступу: <https://www.sqlalchemy.org/>

					ІК11.080БАК.006 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		66

13. Aiogram – Modern Telegram Bot Framework [Електронний ресурс] – Режим доступу: <https://docs.aiogram.dev/>

14. Ultralytics YOLO – офіційна документація [Електронний ресурс] – Режим доступу: <https://docs.ultralytics.com/>

15. Roboflow – Dataset Management Platform for Computer Vision [Електронний ресурс] – Режим доступу: <https://docs.roboflow.com/>

16. Google Drive API [Електронний ресурс] – Режим доступу: <https://developers.google.com/drive>

17. HTTP Methods Explained [Електронний ресурс] – Режим доступу: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods>

18. MVC Design Pattern [Електронний ресурс] – Режим доступу: <https://www.geeksforgeeks.org/mvc-design-pattern/>

19. Lucidchart – UML Activity Diagram Tutorial [Електронний ресурс] – Режим доступу: <https://www.lucidchart.com/pages/uml-activity-diagram>

20. GeeksforGeeks – Evaluating Object Detection Models: Methods and Metrics [Електронний ресурс] – Режим доступу: <https://www.geeksforgeeks.org/evaluating-object-detection-models-methods-and-metrics/>

					ІК11.080БАК.006 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		67