

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Портативна IoT-система виявлення пошкоджень
дорожнього покриття

Виконав: студент IV курсу, групи СІ-42
спеціальності 123 «Комп'ютерна інженерія»

(шифр і назва спеціальності)

Курдичко С.В.
(підпис) (прізвище та ініціали)

Керівник Луцків А.М.
(підпис) (прізвище та ініціали)

Нормоконтроль Луцик Н.С.
(підпис) (прізвище та ініціали)

Завідувач кафедри Осухівська Г.М.
(підпис) (прізвище та ініціали)

Рецензент Стоянов Ю.М.
(підпис) (прізвище та ініціали)

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних систем та мереж
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Осухівська Г.М.

(підпис)

(прізвище та ініціали)

« 25 » квітня 2026 р

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня бакалавр
(назва освітнього ступеня)

за спеціальністю 123 «Комп'ютерна інженерія»
(шифр і назва спеціальності)

студенту Курдичку Сергію Володимировичу
(прізвище, ім'я, по батькові)

1. Тема роботи Портативна IoT-система виявлення пошкоджень дорожнього покриття

Керівник роботи Луцків Андрій Мирославович, к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 24 » квітня 2026 року № 4.9-188

2. Термін подання студентом завершеної роботи 19.06.2026 р.

3. Вихідні дані до роботи Види пошкоджень дорожнього покриття, моделі комп'ютерного зору, типи GPS-модулів, характеристики GSM-модемів

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ. 1. Аналіз принципів та вимог до процесу виявлення пошкоджень дорожнього покриття.

2. Проектування портативної IoT-системи виявлення пошкоджень дорожнього покриття.

3. Програмні модулі портативної IoT-системи виявлення пошкоджень дорожнього покриття

4. Безпека життєдіяльності, основи охорони праці. Висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

1. Фактори впливу на стан дорожнього покриття

2. Архітектура портативної IoT-системи виявлення пошкоджень дорожнього покриття

3. Діаграма потоків даних в портативній IoT-системі

4. Архітектурні рівні та алгоритм роботи портативної IoT-системи

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
<i>Безпека життєдіяльності, основи охорони праці</i>	<i>Сенчишин В.С., к.т.н., доц., каф. МТ</i>		

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	<i>Розробка технічного завдання</i>	<i>26.01 – 02.02</i>	
2.	<i>Робота над першим розділом «Аналіз принципів та вимог до процесу виявлення пошкоджень дорожнього покриття»</i>	<i>03.02 – 15.02</i>	
3.	<i>Робота над другим розділом «Проектування портативної IoT-системи виявлення пошкоджень дорожнього покриття»</i>	<i>20.04 – 25.04</i>	
4.	<i>Робота над третім розділом «Програмні модулі портативної IoT-системи виявлення пошкоджень дорожнього покриття»</i>	<i>26.04 – 05.05</i>	
5.	<i>Робота над четвертим розділом «Безпека життєдіяльності, основи охорони праці»</i>	<i>07.05 – 25.05</i>	
6.	<i>Оформлення пояснювальної записки і графічного матеріалу</i>	<i>26.05 – 7.06</i>	
7.	<i>Перевірка на академічний плагіат, перевірка керівником та консультантами</i>	<i>8.06 – 14.06</i>	
8.	<i>Попередній захист кваліфікаційної роботи бакалавра</i>	<i>15.06 – 21.06</i>	
9.	<i>Захист кваліфікаційної роботи бакалавра</i>	<i>26.06.2026</i>	

Студент

_____ (підпис)

Курдичко Сергій Володимирович

_____ (прізвище та ініціали)

Керівник роботи

_____ (підпис)

Луцків Андрій Мирославович

_____ (прізвище та ініціали)

АНОТАЦІЯ

Курдичко С.В. Портативна IoT-система виявлення пошкоджень дорожнього покриття: робота на здобуття ступеня бакалавра: спец. 123 – комп'ютерна інженерія. Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя, 2026.

Ключові слова: портативна IoT-система, дорожнє покриття, виявлення пошкоджень, комп'ютерний зір, NVIDIA Jetson Nano.

У кваліфікаційній роботі розроблено портативну IoT-систему виявлення пошкоджень дорожнього покриття. Система призначена для автоматизованого виявлення дефектів дорожнього покриття за допомогою камери, центрального обчислювального пристрою, GPS-модуля, модуля передавання даних та алгоритмів комп'ютерного зору.

У роботі проаналізовано проблему моніторингу стану дорожнього покриття, основні типи пошкоджень дорожнього полотна та автоматизовані методи їх виявлення. Обґрунтовано доцільність використання портативної IoT-системи, яка дозволяє зменшити залежність від ручного огляду доріг, забезпечити фіксацію координат дефектів і підвищити оперативність отримання інформації про стан дорожньої інфраструктури.

Для реалізації системи обрано центральний обчислювальний пристрій NVIDIA Jetson Nano, камеру на базі сенсора Sony IMX219, GPS-модуль NEO-6M та 4G USB-модем Huawei E3372h-320. Розроблено архітектуру системи, структурну схему апаратної частини та програмні модулі для отримання кадрів, попередньої обробки зображень, виявлення пошкоджень, визначення рівня небезпеки, GPS-прив'язки, збереження та передавання результатів. Результати роботи системи зберігаються у локальній базі даних SQLite і можуть передаватися через 4G-з'єднання для подальшої візуалізації та аналізу.

ANNOTATION

Kurdychko S.V. Portable IoT system for road surface damage detection: Bachelor's Graduation Thesis: speciality 123 — computer engineering. Ternopil: Ternopil Ivan Puluj National Technical University, 2026.

Keywords: portable IoT system, road surface, damage detection, computer vision, NVIDIA Jetson Nano.

The qualification thesis develops a portable IoT system for detecting road surface damage. The system is designed for automated detection of road surface defects using a camera, a central computing device, a GPS module, a data transmission module, and computer vision algorithms.

The thesis analyzes the problem of monitoring the condition of road surfaces, the main types of road pavement damage, and automated methods for their detection. The feasibility of using a portable IoT system is substantiated, as it makes it possible to reduce dependence on manual road inspection, ensure the recording of defect coordinates, and improve the timeliness of obtaining information about the condition of road infrastructure.

To implement the system, the NVIDIA Jetson Nano central computing device, a camera based on the Sony IMX219 sensor, the NEO-6M GPS module, and the Huawei E3372h-320 4G USB modem were selected. The system architecture, the structural diagram of the hardware part, and software modules for frame acquisition, image preprocessing, damage detection, hazard level assessment, GPS referencing, result storage, and data transmission were developed. The system results are stored in a local SQLite database and can be transmitted via a 4G connection for further visualization and analysis.

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1 АНАЛІЗ ПРИНЦИПІВ ТА ВИМОГ ДО ПРОЦЕСУ ВІЯВЛЕННЯ ПОШКОДЖЕНЬ ДОРОЖНЬОГО ПОКРИТТЯ	9
1.1 Проблеми моніторингу стану дорожнього покриття.....	9
1.2 Основні типи пошкоджень дорожнього полотна	12
1.3 Аналіз автоматизованих методів і засобів виявлення пошкоджень дорожнього покриття	16
РОЗДІЛ 2 ПРОЄКТУВАННЯ ПОРТАТИВНОЇ ІoT-СИСТЕМИ ВІЯВЛЕННЯ ПОШКОДЖЕНЬ ДОРОЖНЬОГО ПОКРИТТЯ	21
2.1 Аналіз вимог та особливостей реалізації портативної ІoT-системи виявлення дефектів дорожнього покриття.....	21
2.2 Проєктування архітектури портативної ІoT-системи виявлення дефектів дорожнього покриття	25
2.3 Центральний обчислювальний пристрій NVIDIA Jetson Nano.....	28
2.4 Вибір камери для отримання зображень дорожнього покриття.....	32
2.5 GPS-модуль NEO-6M	35
2.6 Вибір модуля передавання даних	36
РОЗДІЛ 3 ПРОГРАМНІ МОДУЛІ ПОРТАТИВНОЇ ІoT-СИСТЕМИ ВІЯВЛЕННЯ ПОШКОДЖЕНЬ ДОРОЖНЬОГО ПОКРИТТЯ.....	40
3.1 Архітектура портативної ІoT-системи на програмному рівні	40
3.2 Розробка алгоритму та його програмної реалізації при отриманні та попередній обробці відеокадрів	44

					КС КРБ 123.181.00.00 ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Курдичко С.В.			Літ.	Арк.	Аркушів
Перевір.		Луцків А.М.				6	
Реценз.		Стоянов Ю.М.			ТНТУ, каф. КС, гр. СІ-42		
Н. Контр.		Луцик Н.С.					
Затверд.		Осухівська Г.М.					

3.3	Використання моделі комп'ютерного зору для виявлення пошкоджень дорожнього покриття	49
3.4	Алгоритм класифікації дефектів і визначення рівня небезпеки	54
3.5	Обробка GPS-даних і прив'язка дефектів до координат	57
3.6	Збереження, передавання та візуалізація результатів роботи системи.....	59
РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ		62
4.1	Фактори трудової діяльності та умови праці користувачів ПК.....	62
4.2	Надзвичайні ситуації, викликані вибухами і способи захисту від них.....	65
ВИСНОВКИ		70
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....		71
Додаток А Технічне завдання		
Додаток Б Лістинг класу DamageStorage		
Додаток В Лістинг класу DamageSender		

					<i>КС КРБ 123.181.00.00 ПЗ</i>	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Стан дорожнього покриття є одним із важливих чинників безпеки дорожнього руху, комфорту пересування та ефективності функціонування транспортної інфраструктури. Наявність ям, тріщин, просідань, руйнувань асфальтобетонного шару та інших дефектів може призводити до пошкодження транспортних засобів, зниження швидкості руху, виникнення аварійних ситуацій і збільшення витрат на обслуговування доріг. Тому своєчасне виявлення пошкоджень дорожнього покриття є актуальним завданням для комунальних служб, дорожніх організацій та органів місцевого самоврядування.

Традиційні методи контролю стану доріг переважно ґрунтуються на візуальному огляді дорожнього полотна фахівцями або використанні спеціалізованих дороговартісних діагностичних комплексів. Такі підходи мають низку недоліків, зокрема значні часові витрати, залежність від людського фактору, потребу у залученні спеціального транспорту та обмежену оперативність оновлення інформації про стан дорожньої інфраструктури. У зв'язку з цим виникає потреба у створенні доступних, мобільних і автоматизованих систем, які можуть виконувати моніторинг дорожнього покриття з мінімальною участю людини.

Одним із перспективних напрямів розв'язання цієї задачі є використання технологій Інтернету речей, комп'ютерного зору та глибинного навчання. Поєднання камери, одноплатного комп'ютера, GPS-модуля та програмних алгоритмів аналізу зображень дає змогу створити портативну систему, здатну автоматично отримувати відеодані, виявляти дефекти дорожнього полотна, визначати їхній тип, орієнтовні розміри та місце розташування.

Метою кваліфікаційної роботи є розробка портативної IoT-системи виявлення пошкоджень дорожнього покриття з використанням засобів комп'ютерного зору, одноплатного комп'ютера та модуля геолокації.

					КС КРБ 123.181.00.00 ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1 АНАЛІЗ ПРИНЦИПІВ ТА ВИМОГ ДО ПРОЦЕСУ ВИЯВЛЕННЯ ПОШКОДЖЕНЬ ДОРОЖНЬОГО ПОКРИТТЯ

1.1 Проблеми моніторингу стану дорожнього покриття

Стан дорожнього покриття є важливим показником якості транспортної інфраструктури, оскільки безпосередньо впливає на безпеку дорожнього руху, комфорт перевезень, технічний стан транспортних засобів і загальну ефективність функціонування дорожньої мережі [1]. Дороги щоденно зазнають значних механічних навантажень від руху легкового, вантажного та громадського транспорту. Крім того, на їхній стан впливають кліматичні умови, сезонні коливання температури, опади, замерзання та відтавання води в мікротріщинах, а також якість матеріалів і технологія виконання дорожньо-будівельних робіт [1].

У процесі експлуатації дорожнього полотна поступово виникають різні типи пошкоджень: тріщини, вибоїни, просідання, деформації, руйнування країв покриття, сітчасті пошкодження та інші дефекти. На початкових етапах такі пошкодження можуть мати незначні розміри, однак за відсутності своєчасного виявлення та ремонту вони швидко збільшуються. Наприклад, невелика тріщина під дією навантажень і вологи може перетворитися на більший дефект, а згодом – на вибоїну, яка створює небезпеку для транспортних засобів і пішоходів. Тому своєчасне виявлення дефектів дорожнього покриття є важливою умовою ефективного планування ремонтних робіт.

Основні чинники, що впливають на погіршення стану дорожнього покриття, та можливі наслідки несвоєчасного виявлення дефектів наведено на рис. 1.1.

					КС КРБ 123.181.00.00 ПЗ			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>		<i>Курдичко С.В.</i>			<i>Аналіз принципів та вимог до процесу виявлення пошкоджень дорожнього покриття</i>	<i>Літ.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Перевір.</i>		<i>Луцків А.М.</i>					9	
<i>Реценз.</i>		<i>Стоянов Ю.М.</i>				<i>ТНТУ, каф. КС, гр. СІ-42</i>		
<i>Н. Контр.</i>		<i>Луцик Н.С.</i>						
<i>Затверд.</i>		<i>Осухівська Г.М.</i>						



Рисунок 1.1 – Основні чинники погіршення стану дорожнього покриття та наслідки несвоєчасного моніторингу

Традиційно моніторинг стану доріг виконується шляхом періодичних оглядів, під час яких спеціалісти візуально оцінюють дорожнє покриття, фіксують пошкодження та формують відповідні звіти. Такий підхід є відносно простим, однак має низку суттєвих недоліків. Насамперед він потребує значних витрат часу, особливо якщо необхідно перевірити велику кількість вулиць, доріг або міжміських трас. Крім того, якість оцінювання значною мірою залежить від досвіду та уважності фахівця, що створює ризик пропуску окремих дефектів або суб'єктивної оцінки їхнього рівня небезпеки [2].

Ще однією проблемою ручного моніторингу є низька оперативність оновлення інформації. Стан дорожнього покриття може змінюватися досить швидко, особливо після інтенсивних опадів, зимового періоду або значного транспортного навантаження. Якщо обстеження виконується нечасто, інформація про пошкодження може бути застарілою. У результаті ремонтні служби не завжди мають актуальні дані про місця виникнення дефектів, їхню кількість, розміри та пріоритетність усунення.

Для підвищення точності контролю стану доріг можуть використовуватися спеціалізовані діагностичні комплекси, обладнані високоточними камерами, лазерними сканерами, георадарами, інерційними датчиками та іншими вимірювальними засобами. Такі системи забезпечують детальне обстеження дорожнього покриття, однак їх використання часто пов'язане з високою вартістю обладнання, складністю експлуатації та потребою у спеціально підготовленому персоналі. Через це подібні комплекси не завжди є доступними для невеликих громад, комунальних підприємств або локальних дорожніх служб [2-4].

У сучасних умовах актуальним є пошук більш доступних і гнучких рішень, які могли б автоматизувати процес виявлення пошкоджень дорожнього покриття без використання дороговартісного спеціалізованого обладнання. Перспективним напрямом є застосування портативних IoT-систем. Такі системи можуть встановлюватися на автомобілі або стаціонарних опорах і виконувати збір даних під час звичайного руху або спостереження за певною ділянкою дороги.

Автоматизований моніторинг дорожнього покриття має низку переваг. Він дає змогу збирати дані регулярніше та з меншими витратами часу, а також зберігати результати у базі даних. Це дозволяє аналізувати динаміку погіршення стану дороги. Окрім цього, портативні системи можуть формувати звіти з інформацією про тип дефекту, його розташування, час виявлення та рівень небезпеки. Це спрощує прийняття рішень щодо першочерговості ремонтних робіт і підвищує ефективність управління дорожньою інфраструктурою [3].

Отже, проблема моніторингу стану дорожнього покриття полягає не лише у виявленні окремих пошкоджень, а й у забезпеченні своєчасного, систематичного та об'єктивного збору інформації про стан доріг. Розробка портативної IoT-системи виявлення пошкоджень дорожнього покриття є актуальним завданням, оскільки така система може поєднати доступність апаратних компонентів, можливості комп'ютерного зору та засоби геолокації для автоматизованого контролю дорожньої інфраструктури.

					КС КРБ 123.181.00.00 ПЗ	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

1.2 Основні типи пошкоджень дорожнього полотна

Для побудови автоматизованої системи виявлення пошкоджень важливо чітко визначити, які саме дефекти повинні розпізнаватися програмним модулем. Це необхідно для формування набору класів під час навчання моделі комп'ютерного зору, підготовки зображень, розмітки даних і подальшої класифікації результатів. У межах розроблюваної портативної IoT-системи будуть розпізнаватися ті типи пошкоджень, які можна виявити за допомогою камери та алгоритмів аналізу зображень [5].

Пошкодження дорожнього полотна можуть мати різну форму, глибину, площу поширення та ступінь впливу на безпеку руху. Одні дефекти проявляються у вигляді тонких лінійних тріщин, інші – як локальні руйнування, заглиблення або деформації поверхні. Для автоматизованої системи моніторингу важливо не лише зафіксувати наявність пошкодження, а й визначити його тип, оскільки від цього залежить оцінка небезпеки та подальше прийняття рішення щодо ремонту.

Одним із найбільш поширених видів дефектів є тріщини. Вони виникають унаслідок втоми асфальтобетонного шару, температурних деформацій, нерівномірного просідання основи, старіння матеріалу або циклічної дії транспортного навантаження. На ранніх етапах тріщини можуть бути малопомітними, проте з часом вони збільшуються, розгалужуються та сприяють проникненню води всередину дорожньої конструкції. У холодний період вода, що потрапляє у тріщини, замерзає, розширюється і пришвидшує руйнування покриття [3,4].

За напрямом поширення тріщини, зазвичай, поділяють на поздовжні та поперечні. Поздовжні тріщини розташовані вздовж напрямку руху транспорту. Їх поява може бути пов'язана з недостатньою міцністю основи, нерівномірним ущільненням дорожніх шарів або тривалим навантаженням від коліс транспортних засобів. Такі дефекти є важливими для виявлення, оскільки вони можуть свідчити про поступове ослаблення конструкції дороги.

					КС КРБ 123.181.00.00 ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

Поперечні тріщини, на відміну від поздовжніх, проходять поперек смуги руху. Вони часто виникають через температурне стискання та розширення асфальтобетону, усадку матеріалу або старіння покриття. Поперечні тріщини можуть погіршувати комфорт руху, а за відсутності ремонту поступово розширюватися та ставати джерелом подальших руйнувань [4].

Окрему групу становлять сітчасті або так звані “крокодилові” тріщини. Вони мають вигляд сукупності дрібних взаємопов’язаних ліній, які утворюють характерний візерунок на поверхні дороги. Такий тип пошкодження зазвичай свідчить про значне втомне руйнування дорожнього покриття або про втрату несучої здатності основи. Сітчасті тріщини є більш небезпечними порівняно з поодинокими лінійними дефектами, оскільки вони часто передують утворенню вибоїн і локальному руйнуванню асфальтобетонного шару.

Одним із найбільш небезпечних і помітних дефектів дорожнього полотна є вибоїни. Вибоїна являє собою локальне руйнування покриття, унаслідок якого на поверхні дороги утворюється заглиблення з нерівними краями. Причинами появи вибоїн можуть бути проникнення води у тріщини, руйнування асфальтобетону під дією коліс, недостатня якість ремонтних робіт або тривала експлуатація дороги без належного обслуговування. Наявність вибоїн створює значну небезпеку для транспортних засобів, оскільки може спричинити пошкодження шин, дисків, елементів підвіски та призводити до різкого маневрування водіїв [5].

До поширених дефектів дорожнього полотна також належать просідання та деформації. Просідання проявляється у вигляді локального зниження рівня дорожньої поверхні. Воно може бути спричинене недостатньою міцністю ґрунтової основи, порушенням водовідведення, розмиванням нижніх шарів або неякісним ущільненням матеріалів під час будівництва. Деформації можуть проявлятися у вигляді хвиль, нерівностей, напливів або зміни профілю покриття. Такі пошкодження ускладнюють рух транспорту та знижують комфорт пересування, однак їх автоматичне виявлення за допомогою звичайної відеокамери є складнішим, ніж виявлення тріщин або вибоїн.

					КС КРБ 123.181.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

Ще одним характерним дефектом є колійність. Вона формується у місцях постійного проходження коліс транспортних засобів і має вигляд поздовжніх заглиблень у смузі руху. Колійність найчастіше виникає на дорогах із високою інтенсивністю руху або значною часткою вантажного транспорту. Її поява може бути пов'язана з пластичними деформаціями асфальтобетону, недостатньою міцністю дорожньої основи або впливом високих температур. Особливу небезпеку колійність становить під час дощу, коли в заглибленнях накопичується вода, що підвищує ризик втрати керуваності автомобілем [5].

На краях проїзної частини можуть виникати крайові руйнування. Вони проявляються у вигляді обламування, розтріскування або поступового осипання крайових ділянок дорожнього покриття. Такі дефекти часто виникають через недостатню міцність узбіччя, відсутність належного водовідведення або рух транспорту близько до краю дороги. Крайові руйнування зменшують ефективну ширину проїзної частини та можуть створювати небезпеку для транспортних засобів на вузьких дорогах. На рис. 1.2 проілюстровано дефекти дорожнього покриття.



Рисунок 1.2 – Дефекти дорожнього покриття

Основні типи пошкоджень дорожнього полотна та їхні характерні ознаки систематизовано та представлено у табл. 1.1.

Таблиця 1.1 – Типи пошкоджень дорожнього полотна

Тип пошкодження	Характерні візуальні ознаки	Можливі причини виникнення	Орієнтовний рівень небезпеки
Поздовжня тріщина	Лінійний дефект, розташований уздовж напрямку руху транспорту	Повторюване навантаження від коліс, просідання основи, старіння покриття	Середній
Поперечна тріщина	Тріщина, що проходить поперек смуги руху	Температурні деформації, усадка матеріалу, втрата еластичності асфальтобетону	Середній
Сітчасте руйнування	Сукупність дрібних взаємопов'язаних тріщин, що утворюють сітчастий візерунок	Втомне руйнування покриття, недостатня несуча здатність основи	Високий
Вибоїна	Локальне заглиблення з нерівними краями та порушеною структурою покриття	Руйнування тріщин, дія води, інтенсивний рух транспорту, неякісний ремонт	Високий
Просідання	Локальне зниження рівня дорожньої поверхні	Слабка основа, порушення водовідведення, розмивання нижніх шарів	Середній
Колійність	Поздовжні заглиблення у місцях руху коліс транспортних засобів	Дія важкого транспорту, пластична деформація асфальтобетону, висока температура	Середній або високий
Крайове руйнування	Обламування або розтріскування країв проїзної частини	Недостатня міцність узбіччя, відсутність водовідведення, навантаження на край покриття	Середній

Наведена класифікація дозволяє визначити, які саме пошкодження доцільно враховувати під час розробки програмного забезпечення портативної IoT-системи. З погляду комп'ютерного зору найбільш придатними для автоматичного виявлення є дефекти, що мають чітко виражені візуальні ознаки на зображенні. До них належать поздовжні та поперечні тріщини, сітчасті руйнування і вибоїни. Вони відрізняються формою, текстурою та розмірами, тому можуть бути використані як окремі класи під час навчання або налаштування моделі розпізнавання.

Водночас такі пошкодження, як просідання чи колійність, можуть бути менш помітними на звичайному двовимірному зображенні, особливо за складних умов освітлення або за відсутності виражених контурів. Для їх більш точного виявлення можуть знадобитися додаткові сенсори, наприклад датчики відстані, інерціальні вимірювальні модулі або спеціалізовані системи тривимірного сканування.

Таким чином, основними класами пошкоджень для портативної IoT-системи доцільно обрати поздовжні тріщини, поперечні тріщини, сітчасті руйнування та вибоїни.

1.3 Аналіз автоматизованих методів і засобів виявлення пошкоджень дорожнього покриття

Більш точними системами при аналізі стану дорожнього покриття є спеціалізовані діагностичні комплекси. Такі системи можуть бути встановлені на автомобілях і оснащені високоякісним апаратним і програмним забезпеченням. Вони дозволяють отримувати детальну інформацію про рівність покриття, наявність тріщин, просідань та інших дефектів. Такі системи характеризуються високою точністю вимірювання та можливістю проведення комплексного аналізу. Водночас їх застосування пов'язане зі значними витратами на обладнання, обслуговування та підготовку персоналу.

Окрему групу становлять сенсорні методи контролю [6]. Вони ґрунтуються на використанні акселерометрів, гіроскопів, датчиків вібрації, інерціальних вимірювальних модулів або сенсорів, встановлених на транспортному засобі. Під час руху автомобіля такі датчики фіксують коливання, удари або зміну положення, які можуть свідчити про наявність нерівностей, вибоїн чи просідань. Сенсорні методи мають перевагу у простоті реалізації та можливості використання навіть у мобільних пристроях. Проте вони не завжди дозволяють точно визначити тип пошкодження, оскільки

					КС КРБ 123.181.00.00 ПЗ	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дата		

однакові вібраційні сигнали можуть виникати через різні причини: яму, стиковий шов, нерівність покриття або різке гальмування транспортного засобу.

Перспективним напрямом є використання методів, які базуються на відеоаналізі. У цьому випадку основним джерелом даних є камера, яка фіксує зображення дорожнього полотна. Отримані кадри можуть оброблятися програмними алгоритмами з метою пошуку характерних ознак дефектів: темних ділянок, лінійних структур, контурів, зміни текстур або локальних руйнувань поверхні. Класичні методи обробки зображень можуть включати перетворення кадру у відтінки сірого, фільтрацію шумів, підсилення контрасту, порогову обробку, виділення контурів і морфологічні операції. Такі методи є відносно простими та можуть працювати на обмежених обчислювальних ресурсах, однак їх ефективність значною мірою залежить від якості зображення, освітлення, погодних умов і стану дорожньої поверхні [6,7].

Класичні алгоритми комп'ютерного зору добре підходять для виявлення дефектів із чіткими геометричними ознаками, наприклад окремих тріщин або контрастних вибоїн. Проте дорожнє покриття часто має неоднорідну текстуру, плями, тіні, сліди ремонту, розмітку, забруднення або вологі ділянки. У таких умовах прості алгоритми можуть помилково сприймати сторонні об'єкти як дефекти або, навпаки, не розпізнавати пошкодження з низьким контрастом. Саме тому для більш надійного виявлення пошкоджень дедалі частіше застосовуються методи машинного навчання та глибинного навчання.

Методи машинного навчання передбачають використання набору ознак, які описують зображення або окремі його ділянки [8]. До таких ознак можуть належати форма, текстура, контраст, орієнтація ліній, статистичні характеристики яскравості або градієнти. Після виділення ознак виконується класифікація за допомогою алгоритмів машинного навчання. Перевагою такого підходу є можливість автоматизувати розпізнавання на основі попередньо підготовлених прикладів. Водночас якість роботи системи значною мірою залежить від правильного вибору ознак, що потребує додаткового аналізу та налаштування.

					КС КРБ 123.181.00.00 ПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

Сучаснішим підходом є використання глибинного навчання, зокрема згорткових нейронних мереж. Такі моделі здатні самостійно виділяти характерні ознаки зображень під час навчання, що робить їх ефективними для задач розпізнавання складних візуальних об'єктів. У контексті виявлення пошкоджень дорожнього покриття згорткові нейронні мережі можуть використовуватися для класифікації зображень, локалізації дефектів або сегментації пошкоджених ділянок. Завдяки цьому система може не лише визначати факт наявності дефекту, а й показувати його місце на кадрі.

Для практичних задач виявлення пошкоджень доріг особливо доцільним є застосування моделей детектування об'єктів. До таких моделей належать архітектури сімейства YOLO, SSD, Faster R-CNN та інші. Вони дозволяють знаходити на зображенні об'єкти певних класів і формувати для них обмежувальні рамки. У випадку дорожнього покриття такими об'єктами можуть бути вибоїни, поздовжні тріщини, поперечні тріщини або сітчасті руйнування. Перевагою моделей типу YOLO є висока швидкодія, що особливо важливо для портативної IoT-системи, яка повинна обробляти відеопотік у режимі, наближеному до реального часу [8].

Ще одним підходом є семантична або екземплярна сегментація зображень. На відміну від детектування об'єктів, де дефект позначається прямокутною рамкою, сегментація дозволяє виділити точніший контур пошкодженої області. Це корисно у випадках, коли потрібно оцінити площу дефекту або його форму. Наприклад, для аналізу вибоїн чи сітчастих руйнувань сегментація може дати більш детальну інформацію, ніж звичайна рамка. Проте такі моделі зазвичай потребують більше обчислювальних ресурсів і складнішої підготовки навчальних даних, оскільки для них необхідна розмітка не лише рамками, а й масками пошкоджень.

У портативних IoT-системах важливе значення має баланс між точністю виявлення та обчислювальною складністю [8]. Пристрій, побудований на базі одноплатного комп'ютера, має обмежені ресурси процесора, оперативної пам'яті та енергоспоживання. Тому для такої системи доцільно використовувати

					КС КРБ 123.181.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		18

«легкі» моделі комп'ютерного зору, які здатні забезпечити прийнятну точність без надмірного навантаження на апаратну платформу.

Порівняння основних методів виявлення пошкоджень дорожнього покриття наведено в табл. 1.2.

Таблиця 1.2 – Методи виявлення пошкоджень дорожнього покриття

Метод виявлення	Основний принцип роботи	Переваги	Обмеження
Візуальний огляд	Оцінювання стану дороги фахівцем під час безпосереднього обстеження	Простота реалізації, відсутність складного обладнання	Залежність від людського фактора, значні часові витрати, суб'єктивність оцінки
Діагностичні комплекси	Використання спеціалізованих автомобілів із камерами, лазерами та вимірювальними сенсорами	Висока точність, комплексне обстеження дорожнього полотна	Висока вартість, складність експлуатації, потреба у спеціальному персоналі
Сенсорні методи	Аналіз вібрацій, прискорень або зміни положення транспортного засобу	Відносна простота, можливість мобільного використання	Складність точного визначення типу дефекту, залежність від швидкості руху та характеристик автомобіля
Класична обробка зображень	Пошук контурів, ліній, темних ділянок і змін текстури на кадрі	Невисокі вимоги до обчислювальних ресурсів, зрозуміла логіка роботи	Чутливість до освітлення, тіней, шумів і неоднорідності дорожнього покриття
Машинне навчання	Класифікація зображень або ділянок дороги за попередньо виділеними ознаками	Можливість автоматизованого розпізнавання, краща адаптація до різних типів дефектів	Необхідність ручного вибору ознак, залежність від якості навчальних даних
Глибинне навчання	Використання нейронних мереж для автоматичного виділення ознак і виявлення дефектів	Вища точність, можливість виявлення складних візуальних структур, придатність для відеоаналізу	Потреба у навчальних даних, більші вимоги до обчислювальних ресурсів
Сегментація зображень	Виділення точного контуру пошкодженої області на зображенні	Можливість точнішої оцінки площі та форми дефекту	Складніша розмітка даних, вища обчислювальна складність

Аналіз наведених методів показує, що для розроблюваної портативної IoT-системи найбільш доцільним є використання методів відеоаналізу в поєднанні з

моделями глибинного навчання. Такий підхід забезпечує достатню гнучкість, дає змогу працювати зі звичайною камерою та дозволяє автоматично виявляти характерні типи пошкоджень дорожнього покриття. На відміну від сенсорних методів, аналіз зображень дає змогу не лише зафіксувати факт наявності нерівності, а й визначити її візуальний тип. Порівняно з ручним оглядом, автоматизована система забезпечує більшу регулярність збору даних і зменшує вплив суб'єктивного оцінювання.

Отже, методи виявлення пошкоджень дорожнього покриття відрізняються за складністю, вартістю, точністю та придатністю до використання в портативних системах. Для задачі створення доступної IoT-системи найбільш перспективним є поєднання камери, одноплатного комп'ютера та програмного модуля комп'ютерного зору. Такий підхід дозволяє реалізувати автоматизований моніторинг дорожнього полотна без використання дороговартісних діагностичних комплексів і створює основу для подальшого формування карти пошкоджень дорожньої інфраструктури.

					<i>КС КРБ 123.181.00.00 ПЗ</i>	Арк.
						20
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

РОЗДІЛ 2 ПРОЄКТУВАННЯ ПОРТАТИВНОЇ ІоТ-СИСТЕМИ ВІЯВЛЕННЯ ПОШКОДЖЕНЬ ДОРОЖНЬОГО ПОКРИТТЯ

2.1 Аналіз вимог та особливостей реалізації портативної ІоТ-системи виявлення дефектів дорожнього покриття

Розроблення портативної системи виявлення пошкоджень дорожнього покриття передбачає не лише вибір засобів отримання зображень, а й побудову цілісного технічного рішення, здатного працювати безпосередньо в умовах дослідження дорожнього середовища. На відміну від стаціонарних систем відеоспостереження або спеціалізованих дорожніх лабораторій, портативна система повинна бути мобільною, автономною, компактною та придатною для встановлення на транспортному засобі або використання під час виїзного огляду дороги. Саме тому доцільним є застосування ІоТ-технологій, які дають змогу поєднати засоби збирання даних, локальної обробки, передавання інформації та віддаленого моніторингу в межах єдиної комп'ютеризованої системи [8-10].

ІоТ-підхід у контексті контролю стану дорожнього покриття дозволяє розглядати кожний пристрій моніторингу як окремий інтелектуалізований вузол, який може отримувати візуальні дані, виконувати попередню обробку інформації, фіксувати координати виявлених дефектів і передавати результати до серверної або хмарної частини системи. Такий підхід є особливо актуальним для задач, де важливо не лише виявити пошкодження, а й прив'язати його до конкретного місця, сформувати запис у базі даних і забезпечити можливість подальшого аналізу стану дорожньої інфраструктури [9].

У рамках кваліфікаційної роботи портативна ІоТ-система розглядається як апаратно-програмний комплекс, що складається з камери, обчислювального

					КС КРБ 123.181.00.00 ПЗ			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>		<i>Курдичко С.В.</i>			<i>Проектування портативної ІоТ-системи виявлення пошкоджень дорожнього покриття</i>	<i>Літ.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Перевір.</i>		<i>Луцків А.М.</i>					21	
<i>Реценз.</i>		<i>Стоянов Ю.М.</i>				<i>ТНТУ, каф. КС, гр. СІ-42</i>		
<i>Н. Контр.</i>		<i>Луцик Н.С.</i>						
<i>Затверд.</i>		<i>Осухівська Г.М.</i>						

модуля, засобів позиціонування, модуля бездротового зв'язку, джерела живлення та програмного забезпечення для аналізу зображень.

Основним джерелом даних у такій системі є камера, яка фіксує фрагменти дорожнього покриття під час руху. Отримані зображення передаються на обчислювальний модуль із достатніми апаратними ресурсами, де може виконуватися попередня обробка кадрів, виділення області контролю та подальше виявлення ознак пошкоджень дорожнього покриття [10].

Використання IoT-технологій має кілька важливих переваг для такої системи. По-перше, забезпечується мобільність, оскільки пристрій може використовуватися на різних ділянках дороги без прив'язки до стаціонарної інфраструктури. По-друге, з'являється можливість автоматичного накопичення результатів моніторингу, що дозволяє формувати історію виявлених дефектів. По-третє, завдяки передаванню даних через бездротові мережі результати можуть бути доступними віддалено, наприклад для оператора, дорожньої служби або відповідального інженера. По-четверте, система може бути масштабована: за потреби можна використовувати декілька портативних вузлів, які передаватимуть інформацію до спільного сервера або бази даних [10].

Особливе значення має можливість географічної прив'язки виявлених пошкоджень. Для дорожнього моніторингу недостатньо лише встановити факт наявності тріщини, вибоїни чи просідання. Практична цінність системи суттєво зростає тоді, коли для кожного дефекту зберігаються координати, дата і час фіксації, тип пошкодження та, за можливості, зображення відповідної ділянки дороги. Це дає змогу створити електронний журнал дефектів, який надалі може використовуватися для планування ремонтних робіт, визначення пріоритетності обслуговування ділянок і порівняння змін стану покриття в часі.

Портативний характер системи також визначає підвищені вимоги до енергоспоживання та конструктивного виконання [11]. Пристрій повинен мати можливість працювати від акумулятора або бортової мережі транспортного засобу. Крім того, апаратна частина має бути достатньо компактною, щоб її можна було розмістити в корпусі або закріпити на транспортному засобі без суттєвого втручання в його конструкцію. Важливими є також стійкість до

					КС КРБ 123.181.00.00 ПЗ	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дата		

вібрацій, зміни освітлення, пилу, вологи та інших факторів, характерних для експлуатації в дорожніх умовах.

З погляду обробки даних система повинна забезпечувати отримання зображень дорожнього покриття з достатньою якістю для подальшого аналізу. Для цього необхідно враховувати роздільну здатність камери, кут огляду, висоту встановлення, стабільність зображення та умови освітлення. У разі використання алгоритмів комп'ютерного зору важливо, щоб кадри містили чітке зображення контрольованої ділянки дороги без надмірних спотворень, тіней або розмиття. Тому програмна частина системи має передбачати попередню обробку зображень, зокрема зміну розміру кадру, фільтрацію шумів, виділення області контролю та підготовку даних до подальшого аналізу [11].

Загальну логіку побудови портативної IoT-системи виявлення пошкоджень дорожнього покриття можна подати як послідовність взаємопов'язаних функціональних етапів: отримання візуальних даних, локальна обробка, визначення типу дефекту, фіксація координат, збереження результатів і передавання інформації користувачу або до серверної частини. Візуалізоване представлення такої послідовності показано на рис. 2.1.

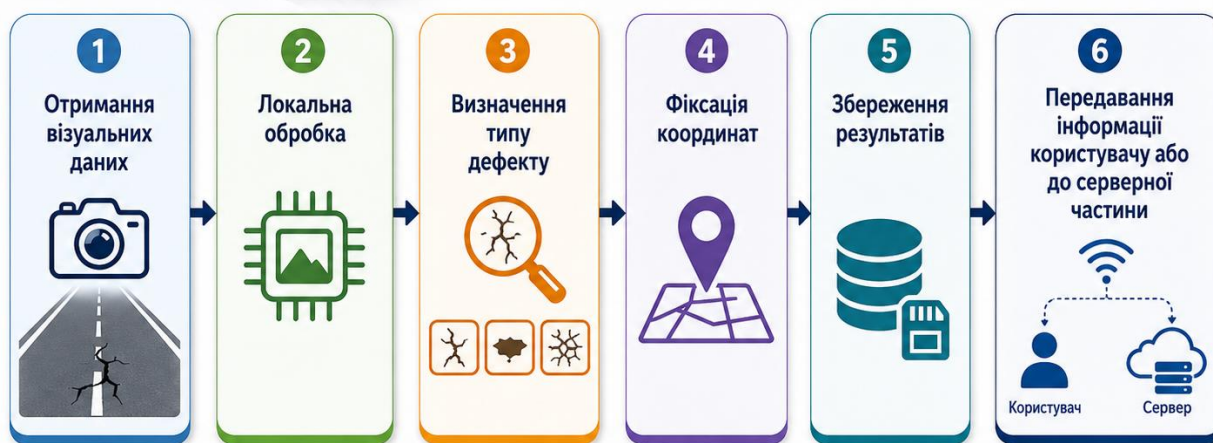


Рисунок 2.1 - Послідовність функціональних етапів у портативній системі виявлення дефектів дорожнього покриття

Підхід, зображений на рис. 2.1, дозволяє перейти від епізодичного ручного огляду дороги до більш системного та автоматизованого контролю.

Переваги застосування IoT-технологій у системі моніторингу дорожнього покриття наведено у табл. 2.1.

Таблиця 2.1 – Переваги використання IoT-технологій

Перевага IoT-підходу	Зміст переваги для розроблюваної системи
Мобільність	Можливість використання пристрою на різних ділянках дороги без стаціонарного монтажу
Автономність	Робота від акумулятора або бортового живлення транспортного засобу
Віддалений доступ	Передавання результатів моніторингу через бездротові канали зв'язку
Географічна прив'язка	Збереження координат місця виявлення дефекту
Масштабованість	Можливість використання декількох вузлів моніторингу
Накопичення даних	Формування бази даних дефектів дорожнього покриття
Оперативність	Швидке виявлення проблемних ділянок і передавання інформації оператору

На основі аналізу даних щодо призначення системи та умов її експлуатації можна сформувати основні вимоги до портативної IoT-системи виявлення пошкоджень дорожнього покриття.

Портативна IoT-система виявлення пошкоджень дорожнього покриття повинна забезпечувати повний цикл роботи з даними: від отримання зображення дороги до збереження результатів аналізу та передавання їх користувачу. Основним джерелом інформації є камера, яка фіксує контрольовану ділянку дорожнього полотна у вигляді окремих кадрів або відеопотоку. Отримані дані передаються до обчислювального модуля, де виконується їх попередня обробка та подальший аналіз.

Система має виявляти характерні візуальні ознаки пошкоджень дорожнього покриття. Для кожного виявленого пошкодження повинен формуватися запис щодо типу дефекту, дати і часу фіксації та інші дані.

Важливою особливістю такої системи є можливість передавання або відображення результатів моніторингу. Інформація про виявлені дефекти може зберігатися в локальній базі даних, передаватися на сервер або відображатися у графічному інтерфейсі користувача. У межах кваліфікаційної роботи

пропонується рішення, у якому результати аналізу записуються у базу даних або журнал подій із можливістю подальшого перегляду.

Оскільки система є портативною, її апаратна частина повинна бути компактною. Пристрій не повинен вимагати складного монтажу, а його підключення та запуск мають виконуватися без значних витрат часу.

Програмне забезпечення системи повинно працювати стабільно та забезпечувати обробку кадрів без значних затримок. Це особливо важливо під час використання пристрою на рухомому транспортному засобі, коли зображення дорожнього покриття швидко змінюється. Інтерфейс користувача має бути зрозумілим і не перевантаженим зайвими елементами, щоб оператор міг швидко переглядати результати, оцінювати стан зафіксованих ділянок і отримувати інформацію про місце виявлення дефекту.

2.2 Проєктування архітектури портативної IoT-системи виявлення дефектів дорожнього покриття

Портативна IoT-система виявлення пошкоджень дорожнього покриття повинна мати таку архітектуру, яка забезпечує отримання зображень дороги, їх локальну обробку, визначення координат виявлених дефектів, збереження результатів і передавання інформації користувачу.

Загальна архітектура портативної IoT-системи повинна відповідати принципу модульності. Це означає, що кожен апаратний або програмний компонент виконує окрему роль у процесі моніторингу дорожнього покриття.

Основними апаратними модулями системи є:

- камера – відповідає за отримання зображень або відеопотоку;
- обчислювальний модуль – виконує обробку кадрів і запуск алгоритму виявлення дефектів;
- GPS-модуль – забезпечує географічну прив'язку результатів;
- модуль зв'язку – використовується для передавання даних до серверної частини або інтерфейсу користувача.

					КС КРБ 123.181.00.00 ПЗ	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дата		

Для зберігання даних застосовується віддалене сховище. На рис. 2.2 показано архітектуру портативної IoT-системи.

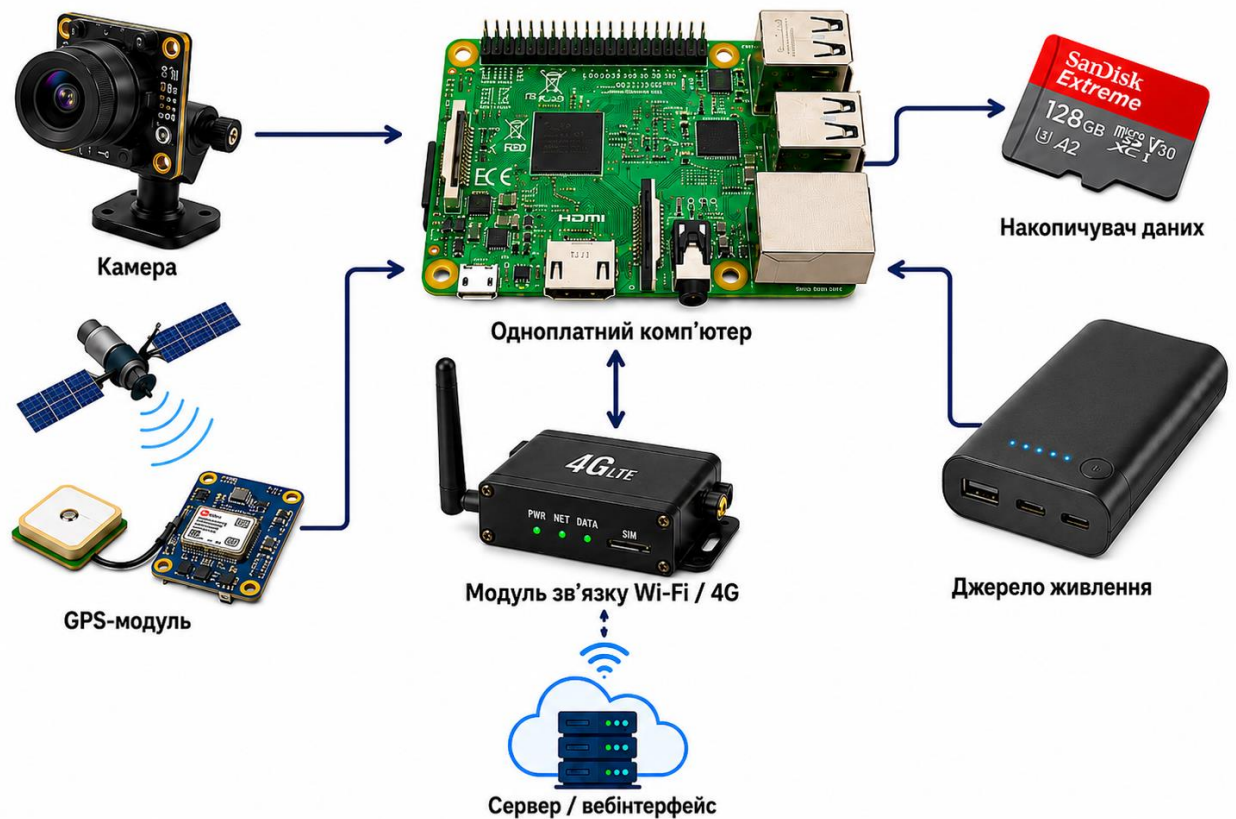


Рисунок 2.2 – Архітектура портативної IoT-системи виявлення пошкоджень дорожнього покриття

Центральним елементом архітектури є мінікомп'ютер, який забезпечує координацію роботи пристроїв та обробку зображень. До нього підключається камера, модуль геолокації, накопичувач даних і засіб бездротового зв'язку.

Під час роботи система отримує кадри дорожнього полотна, виконує їх попередню обробку, передає підготовлені зображення на програмний модуль комп'ютерного зору та аналізує наявність пошкоджень. Якщо дефект виявлено, система формує запис із результатами аналізу та доповнює його координатами, датою, часом і посиланням на збережене зображення.

На рис. 2.3 схематично проілюстровано потоки даних у проєктованій портативній IoT-системі.

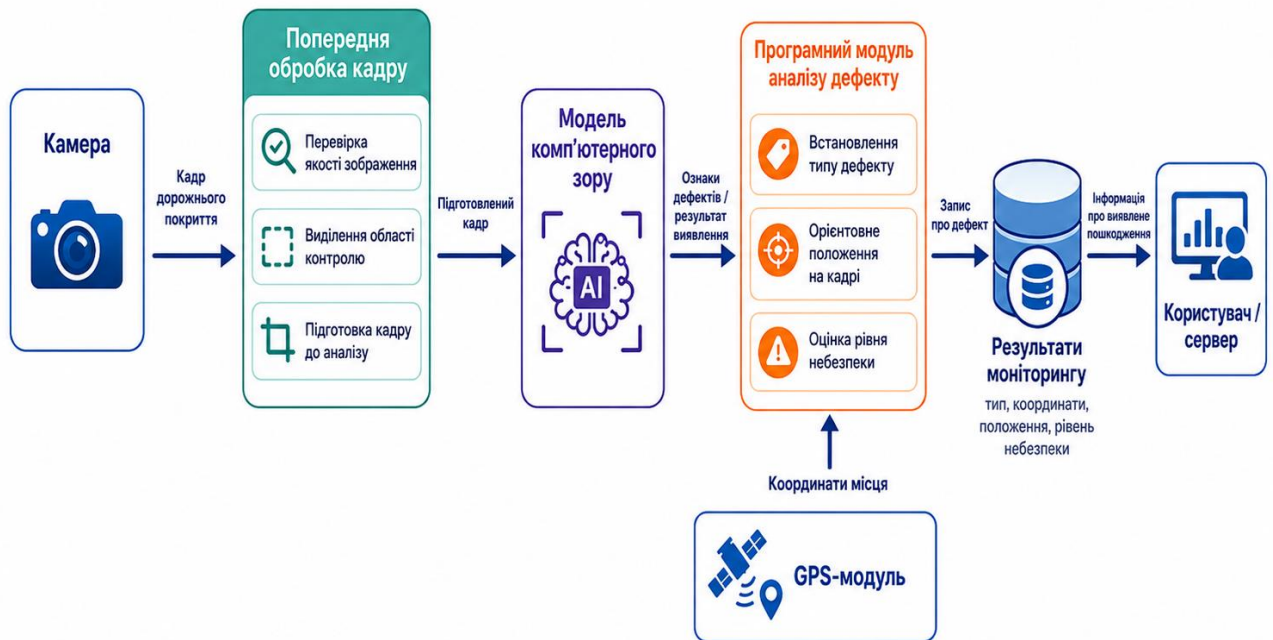


Рисунок 2.3 – Діаграма потоків даних у портативній IoT-системі

З погляду руху даних система працює послідовно. Спочатку камера фіксує ділянку дорожнього покриття. Після цього кадр надходить до обчислювального модуля, де виконується перевірка якості зображення, виділення області контролю та підготовка кадру до аналізу. Далі модель комп'ютерного зору визначає, чи є на зображенні характерні ознаки дефектів. У разі виявлення пошкодження програмний модуль встановлює його тип, орієнтовне положення на кадрі та рівень небезпеки. Паралельно з цим GPS-модуль надає координати місця, у якому було зафіксовано пошкодження.

Результати роботи системи зберігаються локально з можливістю передачі на віддалений сервер. Локальне збереження є доцільним у випадках, коли відсутнє стабільне підключення до мережі або система працює в автономному режимі.

Запропонована архітектура дозволяє розділити систему на кілька логічних рівнів. Перший рівень відповідає за отримання даних із зовнішнього середовища. До нього належать камера та GPS-модуль. Другий рівень виконує обробку інформації і включає одноплатний комп'ютер, програмний модуль попередньої обробки та модель комп'ютерного зору.

Третій рівень забезпечує збереження та передавання результатів, тобто взаємодію з базою даних, сервером або інтерфейсом користувача. На рис. 2.4 продемонстровано структуру системи за архітектурними рівнями.



Рисунок 2.4 – Структура системи за архітектурними рівнями

Перевагою такої архітектури є її гнучкість. За потреби окремі компоненти можуть бути замінені без повної зміни системи. Наприклад, замість однієї моделі камери можна використати іншу камеру з вищою роздільною здатністю, а замість локальної бази даних – віддалене сховище. Також система може бути доповнена додатковими сенсорами, наприклад інерціальним модулем або модулем вимірювання відстані. Це дає можливість поступово розширювати функціональність системи залежно від умов її використання.

2.3 Центральний обчислювальний пристрій NVIDIA Jetson Nano

При реалізації портативної IoT-системи в якості центрального обчислювального пристрою обрано NVIDIA Jetson Nano [12].

Ця платформа належить до класу вбудованих обчислювальних модулів, орієнтованих на задачі штучного інтелекту, комп'ютерного зору та обробки даних безпосередньо на периферійному пристрої. На відміну від звичайних одноплатних комп'ютерів, Jetson Nano має графічний процесор NVIDIA, що дозволяє ефективніше виконувати обчислення, пов'язані з нейронними мережами та аналізом зображень.

Вибір NVIDIA Jetson Nano є обґрунтованим, оскільки система повинна не лише отримувати кадри з камери, а й виконувати їх інтелектуальну обробку. Виявлення тріщин, вибоїн та інших пошкоджень дорожнього покриття потребує використання моделі комп'ютерного зору, яка аналізує зображення та визначає на ньому характерні дефекти. Для таких задач важливо мати апаратну платформу, здатну працювати з алгоритмами глибинного навчання та забезпечувати прийнятну швидкість обробки кадрів [12].

NVIDIA Jetson Nano побудований на базі чотириядерного процесора ARM Cortex-A57 та має графічний процесор NVIDIA Maxwell із 128 ядрами CUDA. Наявність GPU є важливою перевагою для задач комп'ютерного зору, оскільки значна частина операцій у нейронних мережах може виконуватися паралельно. Це дозволяє пришвидшити відповідь моделі порівняно з платформами, які покладаються переважно на центральний процесор. Для портативної системи виявлення дефектів це означає можливість швидше аналізувати кадри та своєчасно фіксувати пошкодження дорожнього полотна.

Платформа має 4 ГБ оперативної пам'яті LPDDR4, що є достатнім для роботи операційної системи, програмного забезпечення обробки зображень, моделі комп'ютерного зору та допоміжних сервісів [13-15]. У межах розроблюваної системи оперативна пам'ять використовується для зберігання поточних кадрів, проміжних результатів обробки, параметрів моделі та даних, які передаються до бази або інтерфейсу користувача. Для оптимальної роботи доцільно використовувати полегшені моделі виявлення об'єктів, які не створюють надмірного навантаження на пам'ять і обчислювальні ресурси.

Важливою особливістю NVIDIA Jetson Nano є підтримка сучасних програмних інструментів для розробки систем штучного інтелекту. Платформа

					КС КРБ 123.181.00.00 ПЗ	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

може використовуватися з бібліотеками комп'ютерного зору та фреймворками глибинного навчання, що дає змогу реалізувати алгоритми виявлення дефектів дорожнього покриття на основі попередньо навченої або адаптованої моделі.

Для роботи з відеопотоком Jetson Nano має необхідні інтерфейси підключення камер і периферійних пристроїв. До нього може бути підключена камера через CSI-інтерфейс або USB-камера, що спрощує вибір джерела зображення для прототипу. Крім того, платформа підтримує підключення накопичувача даних, GPS-модуля через UART або USB, а також засобів мережевого зв'язку [14]. Це дозволяє побудувати на її основі повноцінний портативний пристрій, який не лише аналізує зображення, а й зберігає результати та передає їх користувачу. На рис. 2.5 показано NVIDIA Jetson Nano.



Рисунок 2.5 – NVIDIA Jetson Nano

Окрім цього, NVIDIA Jetson Nano виконує роль локального вузла обробки даних. Такий підхід відповідає концепції периферійних обчислень, коли основний аналіз виконується безпосередньо на пристрої, а не на віддаленому сервері. Це має важливе значення для портативної системи, оскільки під час руху транспортного засобу стабільне інтернет-з'єднання не завжди гарантоване.

Локальна обробка дозволяє системі працювати автономно, фіксувати дефекти навіть без постійного доступу до мережі та передавати результати пізніше [15].

Ще однією перевагою Jetson Nano є придатність до роботи у складі компактних вбудованих рішень. Платформа має невеликі розміри та може бути розміщена у корпусі разом із камерою, модулем геолокації, накопичувачем і блоком живлення. Завдяки цьому її можна використати для створення переносного пристрою або системи, що встановлюється на транспортний засіб. Для стабільної роботи під час тривалого аналізу відеопотоку доцільно передбачити радіатор, активне охолодження та надійне джерело живлення.

Основні характеристики NVIDIA Jetson Nano, важливі для розроблюваної системи, наведено в табл. 2.2.

Таблиця 2.2 – Характеристики NVIDIA Jetson Nano

Характеристика	Значення для розроблюваної системи
Центральний процесор	Чотириядерний ARM Cortex-A57 використовується для керування системою, роботи операційної системи та виконання допоміжних програмних модулів
Графічний процесор	128-ядерний NVIDIA Maxwell забезпечує прискорення обчислень, пов'язаних із комп'ютерним зором і нейронними мережами
Оперативна пам'ять	4 ГБ LPDDR4 достатньо для роботи програмного забезпечення, обробки кадрів і запуску полегшених моделей виявлення дефектів
AI-продуктивність	До 472 GFLOPS, що дозволяє використовувати платформу для задач детектування об'єктів, класифікації та сегментації зображень
Підтримка камер	Можливість підключення CSI- або USB-камери для отримання відеопотоку дорожнього покриття
Підтримка периферії	Можливість підключення GPS-модуля, накопичувача даних, засобів зв'язку та додаткових сенсорів
Програмне середовище	Підтримка Linux-середовища, бібліотек комп'ютерного зору та фреймворків глибинного навчання
Придатність до edge computing	Дає змогу виконувати аналіз відеоданих локально, без постійної залежності від віддаленого сервера
Компактність	Дозволяє розмістити обчислювальний модуль у портативному корпусі системи

У контексті виявлення пошкоджень дорожнього покриття NVIDIA Jetson Nano доцільно використовувати разом із камерою, яка фіксує ділянку дороги перед транспортним засобом або під певним кутом до дорожньої поверхні.

Таким чином, NVIDIA Jetson Nano виступає не лише як пристрій для запуску нейронної мережі, а як центральний вузол, що об'єднує всі апаратні та програмні компоненти системи.

Застосування Jetson Nano також дозволяє забезпечити можливість подальшого розвитку системи без заміни апаратних складових. У майбутньому до неї можна додати складніші моделі комп'ютерного зору, підтримку сегментації дефектів, обробку відеопотоку з кількох камер або підключення додаткових сенсорів. Завдяки наявності GPU така платформа краще підходить для розвитку системи у напрямі інтелектуального аналізу дорожнього середовища

2.4 Вибір камери для отримання зображень дорожнього покриття

Камера є основним сенсорним елементом портативної IoT-системи виявлення пошкоджень дорожнього покриття. Саме вона формує вхідні дані для подальшої обробки, тому від її характеристик залежить подальші результати роботи системи. Для розроблюваної системи камера повинна забезпечувати достатню роздільну здатність, прийнятну частоту кадрів, сумісність із центральним обчислювальним пристроєм NVIDIA Jetson Nano та можливість встановлення у портативній конструкції [16].

Під час вибору камери для такої системи необхідно враховувати специфіку об'єкта спостереження. Тому камера повинна формувати кадри, придатні для подальшого аналізу засобами комп'ютерного зору.

Для портативної IoT-системи запропоновано використати камеру, яка підключається до NVIDIA Jetson Nano через CSI-інтерфейс. Такий спосіб підключення є зручним для вбудованих систем, оскільки дозволяє передавати відеодані без використання громіздких зовнішніх адаптерів. У порівнянні з USB-

					КС КРБ 123.181.00.00 ПЗ	Арк.
						32
Змн.	Арк.	№ докум.	Підпис	Дата		

камерою CSI-модуль має менші розміри, простіше розміщується у корпусі та краще підходить для компактного пристрою [16].

В якості камери для розроблюваної системи обрано модуль на базі сенсора Sony IMX219. Цей сенсор використовується в Raspberry Pi Camera Module 2 та сумісних CSI-камерах для NVIDIA Jetson Nano. Камера має роздільну здатність 8 Мп, що дозволяє отримувати деталізовані зображення дорожнього покриття. Така роздільна здатність є достатньою для фіксації основних типів дефектів. На рис. 2.6 показано модуль камери Sony IMX219.



Рисунок 2.6 – Модуль Sony IMX219

Важливою характеристикою камери є підтримка відеорежимів, придатних для обробки в реальному часі. Для задачі виявлення пошкоджень дорожнього покриття не завжди потрібно використовувати максимальну роздільну здатність сенсора, оскільки це збільшує навантаження на обчислювальний модуль. Практично доцільним є отримання відеопотоку у форматі Full HD або зменшеному розмірі кадру, який надалі подається на вхід моделі комп'ютерного зору. Наприклад, камера на базі IMX219 може використовуватися у режимі 1920 × 1080 з частотою 30 кадрів за секунду, що є достатнім для аналізу дорожньої поверхні під час руху з невеликою або середньою швидкістю.

Окрему увагу необхідно приділити куту огляду камери. Надто вузький кут огляду може обмежувати площу контрольованої ділянки, а надто широкий –

спричиняти геометричні спотворення, які ускладнюють оцінювання розмірів дефектів. Тому у даному випадку застосовуються налаштування камери зі стандартним або помірно широким кутом огляду, який забезпечує баланс між охопленням дороги та збереженням деталізації.

Під час встановлення камери важливо правильно вибрати її положення відносно дорожнього покриття. Якщо камера монтується на транспортному засобі, її можна розташувати у передній частині корпусу, на лобовій ділянці, на бампері або на спеціальному кронштейні. Вона повинна бути спрямована так, щоб у кадрі переважно відображалася контрольована ділянка дороги, а сторонні об'єкти не займали значну частину зображення. Для покращення роботи алгоритму доцільно програмно виділяти область контролю, тобто ту частину кадру, де найімовірніше розташоване дорожнє полотно.

Камера також повинна бути придатною до роботи в умовах змінного освітлення [17]. Під час реальної експлуатації якість зображення може змінюватися через сонячні відблиски, тіні від дерев і будівель, похмуру погоду, дощ або пил на дорожній поверхні. Основні характеристики камери на базі Sony IMX219, важливі з точки зору портативної IoT-системи, наведено в табл. 2.3.

Таблиця 2.3 – Основні характеристики камери на базі Sony IMX219

Характеристика	Значення для портативної IoT-системи
Тип сенсора	Sony IMX219, придатний для отримання зображень дорожньої поверхні
Роздільна здатність	8 Мп, що дозволяє фіксувати дрібні візуальні ознаки пошкоджень
Максимальна роздільна здатність зображення	3280 × 2464 пікселів, що забезпечує деталізовану фіксацію контрольованої ділянки
Відеорежим	1920 × 1080 при 30 кадрах за секунду може використовуватися для аналізу відеопотоку
Інтерфейс підключення	CSI-інтерфейс, зручний для підключення до NVIDIA Jetson Nano
Розміри модуля	Компактне виконання, придатне для встановлення у портативному корпусі
Призначення у системі	Отримання кадрів дорожнього покриття для подальшої обробки та виявлення дефектів

Змн.	Арк.	№ докум.	Підпис	Дата

КС КРБ 123.181.00.00 ПЗ

Арк.

34

Отже, для портативної IoT-системи виявлення пошкоджень дорожнього покриття обрано CSI-камеру, яка забезпечує отримання якісних зображень дорожньої поверхні, має компактні розміри, підтримує відеорежим, придатний для подальшого аналізу, і може бути інтегрована з NVIDIA Jetson Nano. Такий вибір відповідає вимогам до портативності, доступності та можливості реалізації прототипу системи комп'ютерного зору.

2.5 GPS-модуль NEO-6M

У розроблюваній системі для визначення координат застосовується GPS-модуль NEO-6M [18]. Цей модуль є поширеним рішенням для вбудованих систем, IoT-пристроїв і мобільних прототипів, у яких необхідно отримувати географічні координати об'єкта. Його можна підключити до NVIDIA Jetson Nano через UART або за допомогою USB-UART адаптера. Такий спосіб підключення є зручним, оскільки дозволяє отримувати координати у вигляді текстових повідомлень стандарту NMEA, які надалі можуть бути оброблені програмним модулем системи. На рис. 2.7 наведено модуль NEO-6M.

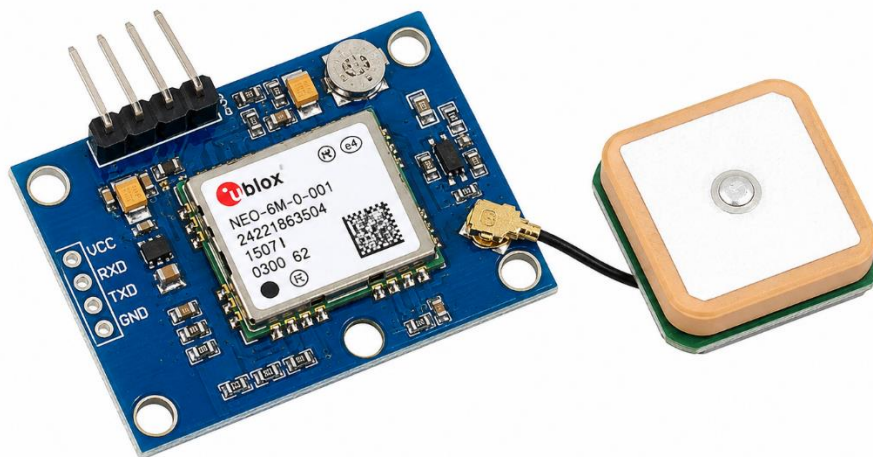


Рисунок 2.7 – GPS-модуль NEO-6M

Модуль NEO-6M приймає сигнали супутникової навігації та формує дані про поточне місце розташування пристрою. Для розроблюваної системи основними параметрами є широта, довгота, час, швидкість руху та якість

					КС КРБ 123.181.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		35

супутникового сигналу. Під час виявлення дефекту програмне забезпечення може зчитати актуальні координати з GPS-модуля та додати їх до запису про пошкодження. У результаті кожен дефект у базі даних буде мати не лише зображення і тип, а й просторову прив'язку [18].

Використання GPS-модуля є особливо важливим у випадку встановлення системи на транспортний засіб. Під час руху камера постійно фіксує нові ділянки дорожнього полотна, а координати змінюються в реальному часі. Якщо пошкодження виявлено на одному з кадрів, система повинна зафіксувати приблизне місце його появи. Для цього достатньо зберігати GPS-координати в момент виявлення дефекту або найближче доступне значення координат, отримане від модуля позиціонування.

Вибір NEO-6M є обґрунтованим завдяки простоті інтеграції, доступності та невеликому енергоспоживанню. Він не потребує складного налаштування та може працювати з типовими програмними бібліотеками для обробки NMEA-даних [18].

У загальній архітектурі системи GPS-модуль працює паралельно з модулем аналізу зображень. Поки камера передає кадри до NVIDIA Jetson Nano, модуль геолокації періодично оновлює поточні координати. Програмне забезпечення зберігає останнє коректне значення координат і використовує його в момент виявлення дефекту. Такий підхід є достатнім для задачі прототипування, оскільки дозволяє пов'язати кожне пошкодження з приблизним місцем його фіксації.

2.6 Вибір модуля передавання даних

Модуль передавання даних є важливим компонентом портативної IoT-системи виявлення пошкоджень дорожнього покриття, оскільки забезпечує передавання результатів моніторингу від пристрою до користувача, серверної частини або бази даних. Для розроблюваної системи це особливо важливо, оскільки пристрій може встановлюватися на транспортний засіб і працювати під час руху. У такому випадку використання лише локальної мережі Wi-Fi є

					КС КРБ 123.181.00.00 ПЗ	Арк.
						36
Змн.	Арк.	№ докум.	Підпис	Дата		

недостатнім, адже система повинна мати можливість передавати дані без прив'язки до конкретної точки доступу.

У розроблюваній системі як основний засіб передавання даних обрано 4G USB-модем Huawei E3372h-320 (рис. 2.8). Цей модем підтримує роботу в мережах 4G LTE, 3G та 2G і підключається до центрального обчислювального пристрою через USB-інтерфейс [19]. Такий спосіб підключення є зручним для NVIDIA Jetson Nano, оскільки не потребує складної зміни апаратної конфігурації системи. Модем може використовуватися разом із SIM-картою мобільного оператора, що дозволяє передавати результати моніторингу через мобільний інтернет.



Рисунок 2.8 – 4G USB-модем Huawei E3372h-320

Використання саме 4G-зв'язку є доцільним для портативної системи, оскільки пристрій не завжди працюватиме в зоні доступу локальної Wi-Fi мережі. Під час руху транспортного засобу система може фіксувати пошкодження на різних ділянках дороги, тому результати доцільно передавати через мобільну мережу. Це дозволяє надсилати дані до серверної частини майже в реальному часі або синхронізувати їх після появи стабільного з'єднання.

Huawei E3372h-320 належить до USB-модемів LTE Cat.4 і забезпечує швидкість приймання даних до 150 Мбіт/с та швидкість передавання до 50 Мбіт/с у мережі 4G. Для розроблюваної системи таких характеристик достатньо.

					КС КРБ 123.181.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		37

Особливістю мобільного зв'язку є те, що його якість може змінюватися залежно від місцевості, покриття оператора, швидкості руху транспортного засобу та навантаження на мережу. Тому система не повинна повністю залежати від постійної наявності інтернет-з'єднання. Для підвищення надійності передбачено режим локального збереження результатів.

Модем Huawei E3372h-320 має компактні розміри та підключається безпосередньо до USB-порту або через короткий USB-подовжувач [19]. Для покращення приймання сигналу доцільно розміщувати модем у верхній частині корпусу пристрою або ближче до зони з кращим радіосигналом. У разі встановлення системи всередині транспортного засобу варто враховувати, що металеві елементи кузова можуть погіршувати якість приймання. Тому за потреби можна використовувати USB-подовжувач або модем із можливістю підключення зовнішньої антени [20].

Основні характеристики вибраного модуля передавання даних наведено в табл. 2.5.

Таблиця 2.5 – Основні характеристики модема Huawei E3372h-320

Характеристика	Значення для портативної IoT-системи
Модель модуля	Huawei E3372h-320
Тип пристрою	4G USB-модем
Підтримувані мережі	4G LTE, 3G, 2G
Інтерфейс підключення	USB, що дозволяє підключити модем до NVIDIA Jetson Nano
Максимальна швидкість приймання даних	До 150 Мбіт/с у мережі 4G
Максимальна швидкість передавання даних	До 50 Мбіт/с у мережі 4G
Підтримувані LTE-діапазони	Bands 1/3/7/8/20, що відповідає поширеним частотам мобільного зв'язку
Основне призначення в системі	Передавання результатів виявлення дефектів до серверної частини або вебінтерфейсу
Режим роботи системи	Передавання JSON-записів, координат, часу фіксації та зображень виявлених пошкоджень
Додатковий режим	Локальне збереження результатів із подальшою синхронізацією після відновлення зв'язку

У загальній архітектурі системи 4G-модем взаємодіє з NVIDIA Jetson Nano як зовнішній мережевий інтерфейс. Передавання даних через 4G-модем дає змогу реалізувати мобільний сценарій використання системи.

Таким чином, обрана апаратна конфігурація портативної IoT-системи є технічно обґрунтованою та відповідає поставленим вимогам. Вона забезпечує отримання зображень дорожнього покриття, локальну інтелектуальну обробку даних, визначення координат дефектів, збереження результатів і передавання інформації через мобільну мережу. Поєднання NVIDIA Jetson Nano, камери Sony IMX219, GPS-модуля NEO-6M і 4G-модема Huawei E3372h-320 дозволяє створити прототип системи для автоматизованого моніторингу стану дорожнього покриття.

					<i>КС КРБ 123.181.00.00 ПЗ</i>	Арк.
						39
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

РОЗДІЛ 3 ПРОГРАМНІ МОДУЛІ ПОРТАТИВНОЇ ІоТ-СИСТЕМИ ВИЯВЛЕННЯ ПОШКОДЖЕНЬ ДОРОЖНЬОГО ПОКРИТТЯ

3.1 Архітектура портативної ІоТ-системи на програмному рівні

Програмне забезпечення портативної ІоТ-системи виявлення пошкоджень дорожнього покриття забезпечує узгоджену роботу всіх етапів обробки даних, починаючи від ініціалізації пристроїв і завершуючи збереженням інформації та її подальшим передаванням. На відміну від апаратної архітектури, яка визначає склад фізичних компонентів системи, програмна архітектура описує логіку взаємодії окремих програмних модулів і послідовність виконання операцій.

У загальному вигляді програмна частина системи складається з кількох взаємопов'язаних модулів:

- модуль захоплення кадрів;
- модуль препроцесингу зображень з камери;
- модуль виявлення пошкоджень дорожнього покриття;
- модуль інтерпретації результатів аналізу;
- модуль роботи з геолокацією;
- модуля збереження даних;
- модуль передавання результатів;
- модуль візуалізації.

Кожен із цих модулів має власне призначення, однак повноцінна робота системи забезпечується лише за умови їх послідовної взаємодії. На рис. 3.1 показано архітектуру програмного забезпечення системи виявлення пошкоджень дорожнього покриття.

					КС КРБ 123.181.00.00 ПЗ			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	<i>Програмні модулі портативної ІоТ-системи виявлення пошкоджень дорожнього покриття</i>	<i>Літ.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Розроб.</i>		<i>Курдичко С.В.</i>						
<i>Перевірив.</i>		<i>Луцків А.М.</i>					40	
<i>Реценз.</i>		<i>Стоянов Ю.М.</i>				<i>ТНТУ, каф. КС, гр. СІ-42</i>		
<i>Н. Контр.</i>		<i>Луцик Н.С.</i>						
<i>Затверд.</i>		<i>Осухівська Г.М.</i>						

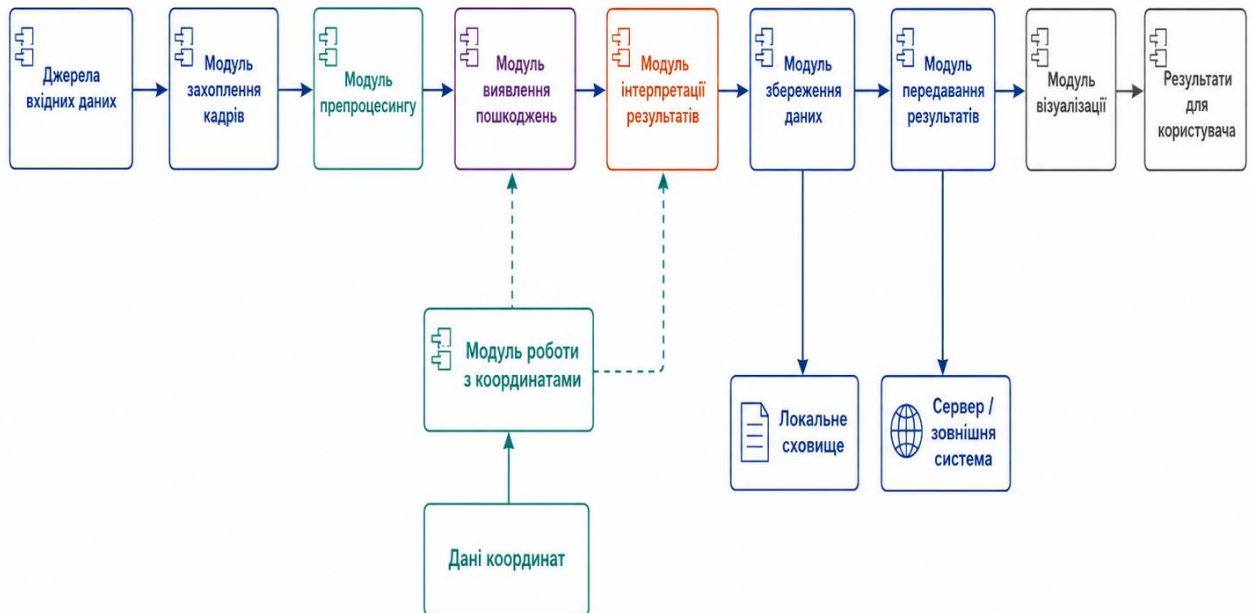


Рисунок 3.1 – Модульна архітектура програмного забезпечення портативної ІоТ-системи

Модуль захоплення кадрів відповідає за отримання зображень дорожнього покриття з відеопотоку. Його завданням є періодичне або безперервне зчитування кадрів і передавання їх на наступний етап обробки. У програмній реалізації цей модуль може працювати з відеопотоком у реальному часі або з попередньо записаним відеофайлом. Такий підхід дозволяє перевіряти програмну логіку навіть без постійного використання реального дорожнього середовища [20].

Модуль препроцесингу готує зображення перед його передачею на модель комп'ютерного зору. На цьому етапі може виконуватися зміна розміру кадру, обрізання області контролю, нормалізація значень пікселів, зменшення шумів і корекція яскравості. Попередня обробка необхідна для того, щоб вхідні дані мали стабільний формат і відповідали вимогам моделі розпізнавання [20].

Модуль виявлення пошкоджень дорожнього покриття виконує аналіз підготовленого кадру за допомогою моделі машинного навчання. Його результатом є набір знайдених об'єктів або областей, які можуть відповідати дефектам дорожнього покриття. Для кожного такого об'єкта модель формує тип пошкодження, координати області на зображенні та рівень упевненості. Ця

інформація ще не є кінцевим записом системи, оскільки потребує додаткової перевірки та інтерпретації.

Модуль інтерпретації результатів уточнює дані, отримані від моделі розпізнавання. Він може відкидати результати з низьким рівнем упевненості, об'єднувати близько розташовані області, визначати орієнтовний розмір дефекту та присвоювати йому рівень небезпеки. Саме цей модуль перетворює вихід моделі комп'ютерного зору на більш зрозумілу для користувача інформацію. Наприклад, замість набору координат рамки система формує запис про те, що на певній ділянці виявлено вибоїну або тріщину із заданим рівнем небезпеки.

Модуль роботи з координатами забезпечує прив'язку результатів аналізу до місця фіксації. Він отримує актуальні навігаційні дані, перевіряє їх коректність і додає координати до запису про дефект. Якщо в момент виявлення пошкодження координати тимчасово недоступні, система може використати останнє коректне значення або позначити запис як такий, що потребує уточнення. Такий підхід дозволяє уникнути втрати результату навіть у випадку короткочасних проблем із визначенням місця розташування.

Модуль збереження даних відповідає за формування і запис результатів у локальне сховище. Для кожного виявленого пошкодження зберігається тип дефекту, рівень небезпеки, дата й час фіксації, координати, шлях до зображення або фрагмента кадру, а також службовий статус запису. Наявність локального сховища дає змогу системі працювати автономно та не втрачати результати у випадку тимчасової відсутності мережевого з'єднання.

Модуль передавання результатів забезпечує обмін даними із зовнішньою частиною системи. Його завданням є підготовка структурованого повідомлення, перевірка доступності мережі, надсилання нових записів і повторна передача тих даних, які не були доставлені раніше. Доцільно передавати не весь відеопотік, а лише сформовані записи про дефекти та зображення пошкоджених ділянок. Це зменшує навантаження на канал зв'язку та робить систему придатною для мобільного використання.

					<i>КС КРБ 123.181.00.00 ПЗ</i>	Арк.
						42
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

Модуль візуалізації призначений для подання результатів у зручній для користувача формі. Він може відображати журнал виявлених пошкоджень, таблицю записів, окремі зображення дефектів або карту з позначеними координатами. Для кваліфікаційної роботи достатнім є опис або реалізація простого інтерфейсу, який дозволяє переглядати основні результати роботи системи: тип дефекту, час фіксації, координати та рівень небезпеки.

Алгоритм роботи портативної IoT-системи виявлення пошкоджень дорожнього покриття представлений на рис. 3.2.

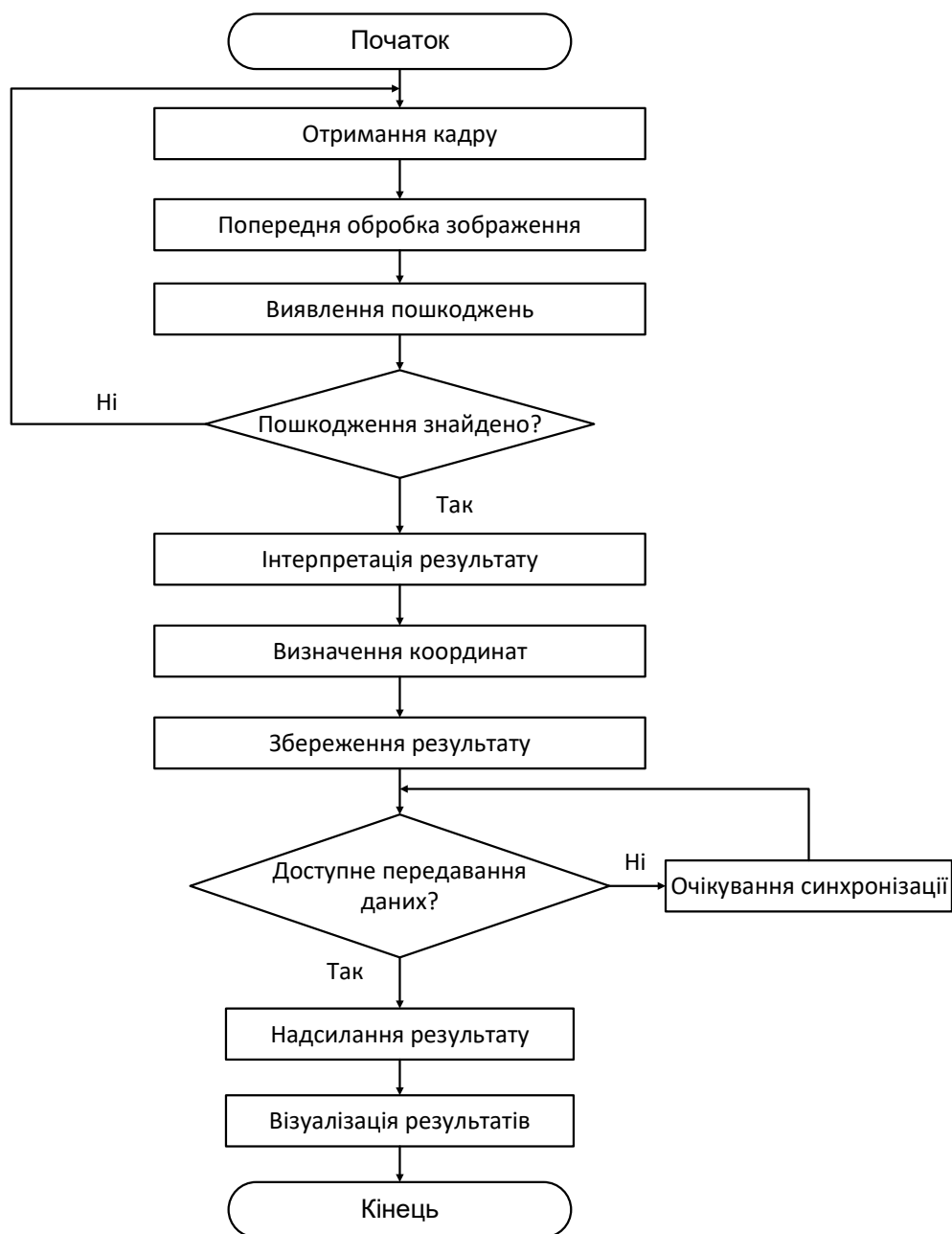


Рисунок 3.2 – Алгоритм роботи портативної IoT-системи

Змн.	Арк.	№ докум.	Підпис	Дата

Запропонована архітектура дозволяє відокремити основні задачі програмного забезпечення й уникнути надмірної залежності між окремими частинами системи. Це важливо для подальшого розвитку, оскільки кожен модуль може бути змінений або вдосконалений без повної перебудови всієї системи. Наприклад, можна замінити модель виявлення пошкоджень, змінити формат збереження даних або додати інший спосіб візуалізації результатів.

3.2 Розробка алгоритму та його програмної реалізації при отриманні та попередній обробці відеокадрів

Алгоритм попередньої обробки кадру складається з кількох послідовних операцій. Спочатку система отримує кадр із відеопотоку. Далі перевіряється коректність отриманого зображення. Якщо кадр не отримано або він пошкоджений, система переходить до зчитування наступного кадру. Якщо зображення є коректним, виконується обрізання області контролю. Це дозволяє залишити лише ту частину кадру, де розташована дорожня поверхня.

Після виділення області контролю зображення масштабується до розміру, який використовується моделлю комп'ютерного зору. Більшість моделей детектування об'єктів працюють із фіксованим розміром вхідного зображення, наприклад 640×640 пікселів. Зміна розміру кадру дозволяє стандартизувати вхідні дані та зменшити обчислювальне навантаження. Крім того, за потреби може виконуватися корекція яскравості, фільтрація шуму або перетворення колірного простору.

Для підвищення стабільності роботи моделі доцільно виконувати нормалізацію зображення. Вона полягає у приведенні значень пікселів до визначеного діапазону, зазвичай, в межах від 0 до 1. Така операція часто використовується у задачах глибинного навчання, оскільки дозволяє узгодити формат вхідних даних із тим, який застосовувався під час навчання моделі. Після цього кадр може бути переданий до модуля виявлення пошкоджень.

Загальний алгоритм отримання та попередньої обробки відеокадру наведено на рис. 3.3.

					<i>КС КРБ 123.181.00.00 ПЗ</i>	Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дата		

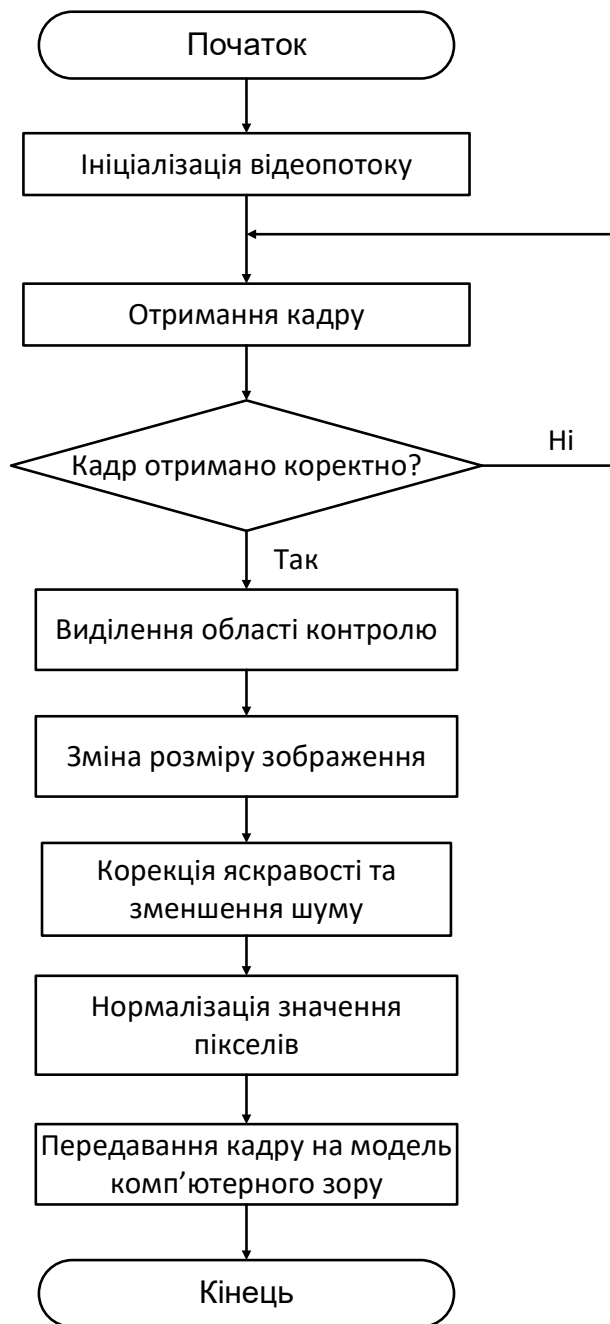


Рисунок 3.3 – Алгоритм захоплення кадру і виконання препроцесингу

Як видно з рис. 3.2, попередня обробка виконується перед етапом розпізнавання пошкоджень дорожнього покриття. Важливо, що алгоритм не змінює зміст зображення, а лише приводить його до такого вигляду, який є зручним для подальшої роботи моделі комп'ютерного зору.

Для реалізації цього алгоритму використано мову Python і бібліотеку OpenCV [21-23]. Нижче, на рис. 3.4, наведено реалізацію класу FramePreprocessor, який відповідає за попередню обробку кадрів перед їх передаванням до моделі комп'ютерного зору.

```

import cv2
import numpy as np

class FramePreprocessor:
    def __init__(self, input_size=(640, 640), roi_y_start=0.35):
        """
        input_size - розмір кадру, який подається на модель комп'ютерного зору;
        roi_y_start - частка висоти кадру, з якої починається область контролю.
        """
        self.input_size = input_size
        self.roi_y_start = roi_y_start

```

Рисунок 3.4 – Фрагмент оголошення класу FramePreprocessor з конструктором

Клас FramePreprocessor виконує роль окремого програмного модуля, який приймає на вхід кадр із відеопотоку та повертає два результати: підготовлене зображення для моделі комп'ютерного зору і покращене зображення для перегляду або збереження. У межах цього класу об'єднано кілька послідовних операцій: виділення області контролю, покращення якості зображення, зміна розміру кадру та нормалізація значень пікселів.

Метод `__init__()` є конструктором класу [22]. Він викликається під час створення об'єкта FramePreprocessor і задає основні параметри попередньої обробки. Параметр `input_size` визначає розмір зображення, яке буде подаватися на модель комп'ютерного зору. У наведеному фрагменті коду використано розмір 640×640 пікселів. Параметр `roi_y_start` визначає, з якої частини кадру починається область контролю дорожнього покриття. Наприклад, значення 0.35 означає, що верхні 35 % кадру відкидаються, а нижня частина використовується для аналізу.

Метод `extract_road_area()`, який показаний на рис. 3.5, призначений для виділення області контролю. Він отримує повний кадр і залишає лише ту його частину, у якій найімовірніше розташоване дорожнє покриття. Для цього спочатку визначаються висота та ширина кадру, після чого обчислюється координата `y_start`. Саме з цієї координати починається фрагмент зображення, який буде використано для подальшої обробки. Така операція дозволяє зменшити обсяг зайвої інформації у кадрі та не аналізувати ділянки, які не мають відношення до дорожнього полотна.

```

def extract_road_area(self, frame):
    """
    Виділення області контролю дорожнього покриття.
    Верхня частина кадру відкидається, оскільки зазвичай не містить
    корисної інформації про стан дорожньої поверхні.
    """
    height, width = frame.shape[:2]
    y_start = int(height * self.roi_y_start)

    road_area = frame[y_start:height, 0:width]

    return road_area

```

Рисунок 3.5 – Метод extract_road_area()

Метод enhance_image() виконує попереднє покращення якості зображення. Спочатку до кадру застосовується фільтр Гауса, який зменшує вплив дрібних шумів. Після цього зображення переводиться з колірного простору BGR у LAB. Такий перехід потрібний для того, щоб окремо опрацювати канал яскравості, не змінюючи суттєво кольорову складову зображення. Далі використовується метод CLAHE, який виконує локальне вирівнювання контрасту. Це дозволяє зробити темні або слабо помітні ділянки дорожньої поверхні більш виразними, що є корисним для подальшого виявлення тріщин і вибоїн. На рис. 3.6 наведено реалізацію методу enhance_image().

```

def enhance_image(self, frame):
    """
    Попереднє покращення якості зображення.
    Виконується зменшення шуму та вирівнювання яскравості.
    """
    blurred = cv2.GaussianBlur(frame, (3, 3), 0)

    lab = cv2.cvtColor(blurred, cv2.COLOR_BGR2LAB)
    l_channel, a_channel, b_channel = cv2.split(lab)

    clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8, 8))
    l_channel = clahe.apply(l_channel)

    enhanced_lab = cv2.merge((l_channel, a_channel, b_channel))
    enhanced_frame = cv2.cvtColor(enhanced_lab, cv2.COLOR_LAB2BGR)

    return enhanced_frame

```

Рисунок 3.6 – Метод enhance_image()

					<i>КС КРБ 123.181.00.00 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		47

Метод `prepare_for_model()`, який представлений на рис. 3.7, приводить зображення до формату, придатного для подачі на модель комп'ютерного зору. На цьому етапі кадр змінюється до заданого розміру `input_size`, після чого значення пікселів перетворюються у формат `float32` і нормалізуються шляхом ділення на 255. У результаті значення яскравості пікселів переходять у діапазон від 0 до 1. Такий формат є зручним для більшості моделей глибинного навчання, оскільки забезпечує стабільніший вхідний масштаб даних.

```
def prepare_for_model(self, frame):  
    """  
    Приведення кадру до формату, придатного для моделі.  
    """  
    resized = cv2.resize(frame, self.input_size)  
    normalized = resized.astype(np.float32) / 255.0  
  
    return normalized
```

Рисунок 3.7 – Метод `prepare_for_model()`

Метод `process()` об'єднує всі попередні операції в одну послідовність. Спочатку він викликає метод `extract_road_area()` і виділяє область контролю. Потім отриманий фрагмент передається до методу `enhance_image()`, де виконується покращення якості зображення. Після цього результат передається до методу `prepare_for_model()`, який формує нормалізований кадр потрібного розміру. Метод (рис. 3.8) повертає два значення: `prepared` – зображення, підготовлене для моделі, та `enhanced` – покращений фрагмент кадру, який може використовуватися для відображення або збереження.

```
def process(self, frame):  
    """  
    Повна послідовність попередньої обробки кадру.  
    """  
    road_area = self.extract_road_area(frame)  
    enhanced = self.enhance_image(road_area)  
    prepared = self.prepare_for_model(enhanced)  
  
    return prepared, enhanced
```

Рисунок 3.8 – Метод `process()`

					КС КРБ 123.181.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		48

Загалом реалізований клас забезпечує достатній набір операцій попередньої обробки відеокадру. Його використання дозволяє стандартизувати вхідні дані, зменшити кількість зайвої інформації в кадрі та частково компенсувати вплив шумів або нерівномірного освітлення. У подальшому цей клас можна розширити, додавши стабілізацію зображення, адаптивне визначення області контролю або додаткові методи фільтрації для складних умов зйомки.

3.3 Використання моделі комп'ютерного зору для виявлення пошкоджень дорожнього покриття

Після отримання та попередньої обробки кадру наступним етапом роботи програмного забезпечення є виявлення пошкоджень дорожнього покриття. На цьому етапі підготовлене зображення передається до моделі комп'ютерного зору, яка аналізує вміст кадру та визначає наявність характерних дефектів. На відміну від класичних алгоритмів обробки зображень, модель детектування об'єктів дає змогу не лише встановити факт наявності пошкодження, а й визначити його розташування на кадрі та віднести до певного класу.

У портативній IoT-системі виявлення пошкоджень дорожнього покриття використовується модель сімейства YOLO [24]. Такий підхід є зручним для задачі виявлення дефектів дорожнього покриття, оскільки модель працює з повним зображенням і формує список знайдених об'єктів у вигляді обмежувальних рамок. Кожна рамка містить координати області, у якій виявлено дефект, назву класу та значення впевненості моделі. Це дозволяє надалі виконати класифікацію пошкодження, оцінити його орієнтовний розмір і сформувати запис для бази даних.

У рамках кваліфікаційної роботи модель комп'ютерного зору може розпізнавати кілька основних класів пошкоджень: поздовжні тріщини, поперечні тріщини, сігчасті руйнування та вибоїни. Такий набір класів відповідає тим дефектам, які мають помітні візуальні ознаки на зображенні та можуть бути

					<i>КС КРБ 123.181.00.00 ПЗ</i>	Арк.
						49
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

виявлені за допомогою камери. Якщо модель повертає кілька знайдених об'єктів, програмний модуль обробляє кожен результат окремо.

Загальна логіка роботи модуля виявлення полягає у послідовному виконанні кількох операцій. Спочатку до модуля надходить попередньо підготовлений кадр. Далі виконується запуск моделі розпізнавання. Після цього результати детекції перевіряються за порогом впевненості. Якщо значення впевненості нижче встановленого порогу, такий результат відкидається, щоб зменшити кількість помилкових спрацювань. Якщо результат є достатньо надійним, система зберігає координати рамки, клас дефекту та значення впевненості.

Для реалізації модуля виявлення пошкоджень використано бібліотеку `ultralytics`, яка забезпечує зручний інтерфейс для роботи з моделями YOLO [24]. У програмному коді модель завантажується з файлу ваг, після чого до неї передається кадр. Результатом роботи є список виявлених об'єктів із координатами, класами та значеннями впевненості. На рис. 3.9 показано оголошення класу для виявлення пошкоджень дорожнього покриття.

```
from ultralytics import YOLO
import cv2

class RoadDamageDetector:
    def __init__(self, model_path, confidence_threshold=0.5):
        """
        model_path - шлях до файлу навченої моделі;
        confidence_threshold - мінімальний рівень впевненості,
        за якого виявлений об'єкт вважається достовірним.
        """
        self.model = YOLO(model_path)
        self.confidence_threshold = confidence_threshold

        self.class_names = {
            0: "longitudinal_crack",
            1: "transverse_crack",
            2: "alligator_crack",
            3: "pothole"
        }
```

Рисунок 3.9 – Оголошення класу `RoadDamageDetector`

					КС КРБ 123.181.00.00 ПЗ	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

У наведеному на рис. 3.9 програмному коді реалізовано клас RoadDamageDetector, який призначений для виявлення пошкоджень дорожнього покриття на окремому кадрі. Конструктор `__init__()` виконує початкове налаштування детектора. Він приймає шлях до файлу навченої моделі `model_path` і значення порогу впевненості `confidence_threshold`. У конструкторі завантажується модель YOLO, зберігається мінімальний допустимий рівень впевненості та задається словник класів пошкоджень: поздовжня тріщина, поперечна тріщина, сітчасте руйнування і вибоїна. У класі реалізовано метод `detect()`, програмний код якого представлено на рис. 3.10.

```
def detect(self, frame):
    """
    Виконання детекції пошкоджень на одному кадрі.
    """
    results = self.model(frame, verbose=False)
    detections = []

    for result in results:
        for box in result.boxes:
            confidence = float(box.conf[0])

            if confidence < self.confidence_threshold:
                continue

            class_id = int(box.cls[0])
            x1, y1, x2, y2 = box.xyxy[0].tolist()

            detection = {
                "class_id": class_id,
                "class_name": self.class_names.get(class_id, "unknown"),
                "confidence": confidence,
                "bbox": {
                    "x1": int(x1),
                    "y1": int(y1),
                    "x2": int(x2),
                    "y2": int(y2)
                }
            }

            detections.append(detection)

    return detections
```

Рисунок 3.10 – Програмна реалізація методу `detect()`

Метод `detect()` виконує безпосереднє виявлення дефектів на кадрі. На вхід він отримує зображення `frame`, передає його до моделі YOLO та обробляє

					КС КРБ 123.181.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		51

результати розпізнавання. Для кожного знайденого об'єкта метод перевіряє рівень впевненості, визначає клас пошкодження та координати обмежувальної рамки. Якщо впевненість нижча за встановлений поріг, результат ігнорується. У підсумку метод повертає список достовірно виявлених пошкоджень у вигляді структурованих записів.

Для наочного відображення результатів детекції пошкоджень дорожнього покриття додано функцію, яка наносить рамки та підписи на кадр. Це корисно під час налагодження системи, оскільки дозволяє швидко перевірити, чи правильно модель знаходить пошкожені ділянки. Фрагмент програмного коду цього методу показано на рис. 3.11.

```
def draw_detections(frame, detections):
    """
    Нанесення результатів детекції на зображення.
    """
    output_frame = frame.copy()

    for detection in detections:
        bbox = detection["bbox"]
        class_name = detection["class_name"]
        confidence = detection["confidence"]

        x1 = bbox["x1"]
        y1 = bbox["y1"]
        x2 = bbox["x2"]
        y2 = bbox["y2"]

        label = f"{class_name}: {confidence:.2f}"
```

Рисунок 3.11 – Фрагмент методу draw_detections()

Функція draw_detections() приймає початковий кадр і список знайдених дефектів. Для кожного результату вона зчитує координати рамки, назву класу та рівень впевненості. Після цього на зображенні малюється прямокутник навколо виявленої області та додається текстовий підпис. Такий кадр може використовуватися для перегляду оператором, збереження у файл або подальшого відображення в інтерфейсі користувача.

Для використання модуля детекції у загальному програмному циклі достатньо створити об'єкт класу RoadDamageDetector, передати до нього шлях до файлу моделі та викликати метод detect() для кожного підготовленого кадру. Застосування модуля детекції пошкоджень дорожнього покриття показано на рис. 3.12.

```
detector = RoadDamageDetector(  
    model_path="models/road_damage_yolo.pt",  
    confidence_threshold=0.5  
)  
  
image_path = "test_images/road_frame.jpg"  
frame = cv2.imread(image_path)  
  
if frame is None:  
    print("Не вдалося завантажити зображення")  
else:  
    detections = detector.detect(frame)  
  
    if len(detections) > 0:  
        result_frame = draw_detections(frame, detections)  
  
        for detection in detections:  
            print("Виявлено дефект:", detection["class_name"])  
            print("Впевненість:", detection["confidence"])  
            print("Координати:", detection["bbox"])  
            print("-" * 30)  
  
        cv2.imshow("Detection result", result_frame)  
        cv2.waitKey(0)  
        cv2.destroyAllWindows()  
    else:  
        print("Пошкоджень на кадрі не виявлено")
```

Рисунок 3.12 – Застосування модуля детекції пошкодження дорожнього покриття

Як видно з рис. 3.12, модель завантажується з файлу road_damage_yolo.pt. Після запуску методу detect() програма отримує список виявлених пошкоджень. Якщо список не порожній, результати можуть бути нанесені на кадр, виведені в консоль або передані до наступного модуля програмного забезпечення. Таким чином, модуль комп'ютерного зору виконує ключову роль у програмному забезпеченні системи.

					КС КРБ 123.181.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		53

3.4 Алгоритм класифікації дефектів і визначення рівня небезпеки

У портативній IoT-системі рівень небезпеки визначається на основі кількох ознак: типу пошкодження, площі області дефекту на кадрі та рівня впевненості моделі. Такий підхід є простим для реалізації та достатнім для прототипу.

Найбільшу небезпеку для транспортних засобів, зазвичай, становлять вибоїни та сітчасті руйнування, оскільки вони можуть свідчити про значне пошкодження дорожнього покриття. Лінійні тріщини можуть мати нижчий рівень небезпеки, однак за великої довжини або значної площі вони також потребують уваги. Тому в алгоритмі оцінювання враховується як клас дефекту, так і розмір області, у якій він був виявлений.

Площа дефекту в кадрі обчислюється за координатами обмежувальної рамки. Для більшої універсальності доцільно використовувати не абсолютну площу, а відносну площу, тобто частку кадру, яку займає виявлений дефект.

Для реалізації цього етапу можна створити окремий клас `DamageRiskEvaluator`. Він буде отримувати результати виявлення пошкодження дорожнього покриття, обчислювати площу обмежувальної рамки та додавати до кожного запису рівень небезпеки. Такий підхід дозволяє відокремити логіку оцінювання від модуля детекції, який відповідає лише за пошук дефектів. Оголошення класу `DamageRiskEvaluator` проілюстровано на рис. 3.13.

```
class DamageRiskEvaluator:
    def __init__(self, medium_area_threshold=0.03, high_area_threshold=0.08):
        """
        medium_area_threshold - поріг відносної площі для середнього рівня;
        high_area_threshold - поріг відносної площі для високого рівня.
        """
        self.medium_area_threshold = medium_area_threshold
        self.high_area_threshold = high_area_threshold
```

Рисунок 3.13 – Оголошення класу `DamageRiskEvaluator`

					<i>КС КРБ 123.181.00.00 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		54

У наведеному коді клас `DamageRiskEvaluator` реалізує оцінювання рівня небезпеки виявлених дефектів. Конструктор `__init__()` задає порогові значення відносної площі, які використовуються для розмежування середнього та високого рівнів небезпеки. Значення цих порогів можна змінювати залежно від роздільної здатності кадру, положення камери та практичних вимог до системи.

Метод `calculate_relative_area()` обчислює, яку частину кадру займає обмежувальна рамка виявленого дефекту. Для цього визначається площа рамки та ділиться на загальну площу кадру. Використання відносної площі робить алгоритм менш залежним від конкретного розміру зображення. Програмна реалізація методу `calculate_relative_area()` наведена на рис. 3.14.

```
def calculate_relative_area(self, bbox, frame_width, frame_height):  
    """  
    Обчислення відносної площі обмежувальної рамки.  
    """  
  
    box_width = bbox["x2"] - bbox["x1"]  
    box_height = bbox["y2"] - bbox["y1"]  
  
    box_area = box_width * box_height  
    frame_area = frame_width * frame_height  
  
    if frame_area == 0:  
        return 0  
  
    return box_area / frame_area
```

Рисунок 3.14 – Метод `calculate_relative_area()`

Метод `define_risk_level()`, який представлений на рис. 3.15, визначає рівень небезпеки за типом пошкодження та його відносною площею. Для вибоїн базовим рівнем є середній, оскільки навіть невеликі вибоїни можуть створювати небезпеку для транспортних засобів. Для сітчастих руйнувань базовим рівнем також є середній, а за більшої площі він підвищується до високого. Для поздовжніх і поперечних тріщин рівень небезпеки залежить переважно від розміру виявленої області.

```

def define_risk_level(self, class_name, relative_area):
    """
    Визначення рівня небезпеки за типом дефекту та його площю.
    """
    if class_name == "pothole":
        if relative_area >= self.high_area_threshold:
            return "high"
        return "medium"

    if class_name == "alligator_crack":
        if relative_area >= self.medium_area_threshold:
            return "high"
        return "medium"

    if class_name in ["longitudinal_crack", "transverse_crack"]:
        if relative_area >= self.medium_area_threshold:
            return "medium"
        return "low"

    return "unknown"

```

Рисунок 3.15 – Метод define_risk_level()

Фрагмент програмної реалізації методу evaluate() проілюстровано на рис. 3.16. Він приймає список результатів детекції та розмір кадру. Для кожного знайденого дефекту цей метод обчислює відносну площу, визначає рівень небезпеки та додає ці дані до результату. На виході формується оновлений список пошкоджень, який може бути переданий до модуля збереження, візуалізації або формування звіту.

```

def evaluate(self, detections, frame_shape):
    """
    Додавання рівня безпеки до результатів детекції.
    """
    frame_height, frame_width = frame_shape[:2]
    evaluated_detections = []

    for detection in detections:
        bbox = detection["bbox"]
        class_name = detection["class_name"]

        relative_area = self.calculate_relative_area(
            bbox,
            frame_width,
            frame_height
        )

```

Рисунок 3.16 – Фрагмент методу evaluate()

					КС КРБ 123.181.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		56

Запропонований алгоритм є правилом, тобто рівень небезпеки визначається не окремою нейронною мережею, а набором зрозумілих умов. Це є перевагою для прототипу, оскільки така логіка легко пояснюється, налаштовується та перевіряється.

3.5 Обробка GPS-даних і прив'язка дефектів до координат

Модуль роботи з GPS-даними отримує від навігаційного приймача поточні значення широти та довготи. Загальна логіка прив'язки дефекту до координат є простою: після виявлення пошкодження система отримує актуальні GPS-дані, формує запис із типом дефекту, рівнем небезпеки, часом фіксації та координатами, після чого передає цей запис до модуля збереження. У результаті кожне пошкодження має не лише візуальне підтвердження, а й просторову прив'язку.

Для програмної реалізації обробки GPS-даних можна використати окремий клас `GPSReader` [25]. Його завданням є зчитування координат, перевірка їх коректності та повернення останнього достовірного значення широти й довготи. Реалізація цього класу має вигляд, як показано на рис. 3.17.

```
from datetime import datetime

class GPSReader:
    def __init__(self):
        """
        Ініціалізація об'єкта для роботи з GPS-даними.
        """
        self.last_valid_location = None
```

Рисунок 3.17 – Оголошення класу `GPSReader`

У конструкторі `__init__()` створюється змінна `last_valid_location`, у якій зберігається останнє коректне значення координат.

До класу `GPSReader` входять два методи: `update_location()` та `get_current_location()`. Програмний код методу `update_location()` наведений на рис. 3.18.

```
def update_location(self, latitude, longitude):
    """
    Оновлення поточних координат.
    """
    if latitude is None or longitude is None:
        return self.last_valid_location

    if not (-90 <= latitude <= 90):
        return self.last_valid_location

    if not (-180 <= longitude <= 180):
        return self.last_valid_location

    self.last_valid_location = {
        "latitude": latitude,
        "longitude": longitude,
        "timestamp": datetime.now().strftime("%Y-%m-%d %H:%M:%S")
    }

    return self.last_valid_location
```

Рисунок 3.18 – Метод `update_location()`

Метод `update_location()` приймає значення широти та довготи. Спочатку він перевіряє, чи координати не є порожніми. Далі виконується перевірка допустимих меж: широта повинна бути в діапазоні від -90 до 90 градусів, а довгота — від -180 до 180 градусів. Якщо координати коректні, вони зберігаються разом із поточним часом. Якщо дані некоректні, метод повертає останнє достовірне значення.

Метод `get_current_location()` використовується для отримання останніх збережених координат. Його можна викликати в момент виявлення дефекту, щоб додати просторову прив'язку до запису про пошкодження. Програмну реалізацію цього методу показано на рис. 3.19.

					КС КРБ 123.181.00.00 ПЗ	Арк.
						58
Змн.	Арк.	№ докум.	Підпис	Дата		

```

def get_current_location(self):
    """
    Повернення останніх коректних координат.
    """
    return self.last_valid_location

```

Рисунок 3.19 – Метод get_current_location()

3.6 Збереження, передавання та візуалізація результатів роботи системи

Після виявлення пошкодження, визначення його типу, рівня небезпеки та координат система повинна сформувати повний запис про дефект. Такий запис є підсумковим результатом роботи попередніх програмних модулів і використовується для подальшого збереження, передавання та відображення користувачу. На цьому етапі важливо забезпечити цілісність даних, оскільки кожен виявлений дефект повинен мати не лише назву класу, а й час фіксації, координати, рівень небезпеки, шлях до зображення та статус передавання.

Для локального збереження результатів доцільно використати базу даних SQLite [25, 26]. У структурі запису про дефект доцільно передбачити службове поле sync_status. Воно показує, чи був запис уже переданий на сервер. Наприклад, значення pending може означати, що результат збережено локально, але ще не передано, а значення sent – що запис успішно синхронізовано. Такий механізм дозволяє уникнути втрати даних у разі нестабільного мобільного інтернету та забезпечує поступову синхронізацію накопичених результатів. Основні поля запису про виявлене пошкодження наведено в табл. 3.1.

Таблиця 3.1 – Структура запису про пошкодження дорожнього покриття

Поле запису	Призначення
id	Унікальний номер запису в локальній базі даних
class_name	Тип виявленого пошкодження
risk_level	Визначений рівень небезпеки дефекту
confidence	Рівень впевненості моделі комп'ютерного зору
latitude	Географічна широта місця фіксації

Поле запису	Призначення
longitude	Географічна довгота місця фіксації
detected_at	Дата і час виявлення пошкодження
image_path	Шлях до збереженого зображення або фрагмента кадру
sync_status	Стан передавання запису на сервер

Загальна логіка роботи модуля збереження та передавання результатів полягає в тому, що кожен новий дефект спочатку записується у локальну базу даних. Після цього програма перевіряє наявність мережевого з'єднання. Якщо зв'язок доступний, система формує структуроване повідомлення у форматі JSON, і передає його на сервер. Якщо передавання виконано успішно, статус запису змінюється на *sent*. Якщо ж з'єднання відсутнє або сервер тимчасово недоступний, запис залишається зі статусом *pending*. Для програмної реалізації цього етапу реалізовано окремий клас *DamageStorage* (додаток Б). Він відповідає за створення таблиці в базі даних, додавання нових записів, отримання непереданих результатів та оновлення їхнього статусу після синхронізації.

Для передавання результатів через 4G-з'єднання використовується HTTP-запит до серверної частини [27, 28]. У цьому випадку дані з локальної бази перетворюються у формат JSON і надсилаються на визначену адресу сервера. Програмну реалізацію цього класу наведено у додатку В. Результатом роботи портативної IoT-системи є сформований запис про виявлене пошкодження дорожнього покриття, структура якого при передачі на серверну частину представляється у форматі JSON, як показано на рис. 3.20.

```
{
  "class_name": "pothole",
  "risk_level": "high",
  "confidence": 0.91,
  "latitude": 49.5535,
  "longitude": 25.5948,
  "detected_at": "2026-06-01 10:35:20",
  "image_path": "images/damage_001.jpg",
  "sync_status": "sent"
}
```

Рисунок 3.20 – Структура JSON-результату

					КС КРБ 123.181.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		60

Окремим етапом є візуалізація результатів роботи системи. Для користувача результати можуть подаватися у вигляді таблиці, журналу подій або карти з позначенням координат пошкоджень. У найпростішому варіанті інтерфейс може відображати список виявлених дефектів із зазначенням типу пошкодження, часу фіксації, рівня небезпеки та координат.

У розширеному варіанті ці дані можуть наноситися на карту, що дозволяє швидко оцінити розташування проблемних ділянок дороги. Результат візуалізації даних про пошкодження показано на рис. 3.21.

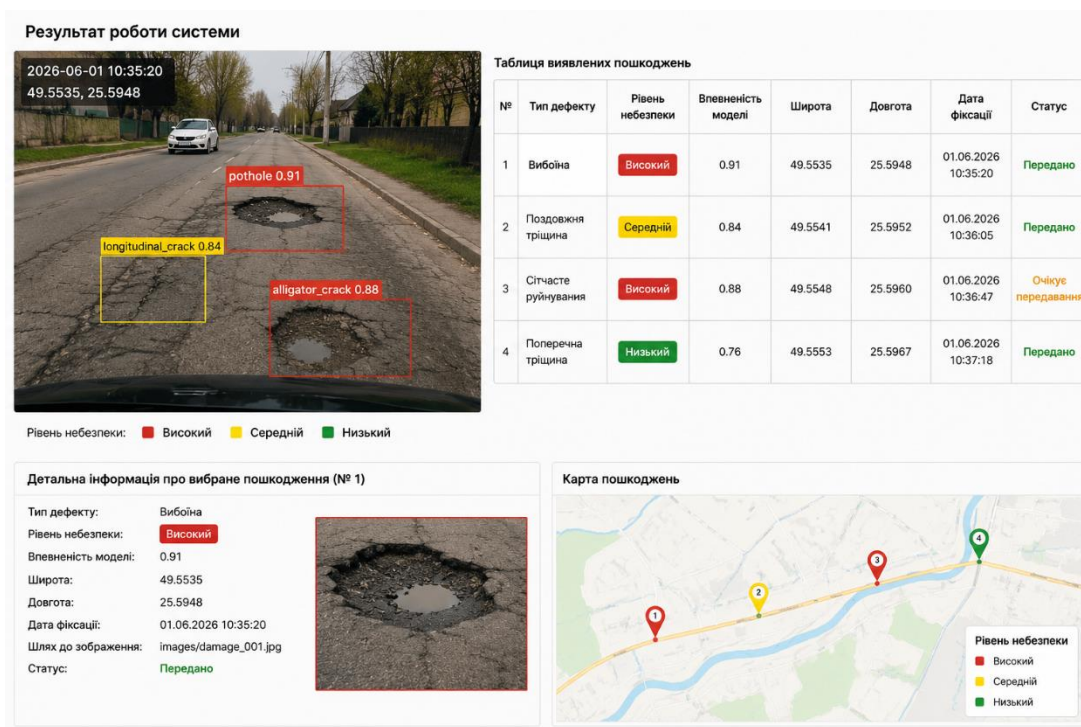


Рисунок 3.21 – Візуалізація даних про пошкодження дорожнього покриття

В інтерфейсі користувача результати можуть відображатися у вигляді журналу подій або карти пошкоджень. У журналі подій користувач бачить перелік усіх виявлених дефектів із датою, координатами та рівнем небезпеки. На карті кожен дефект позначений окремою точкою, колір або маркер якої відповідає рівню небезпеки. Такий спосіб візуалізації дозволяє швидко визначити ділянки дороги, які потребують першочергової уваги.

РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Фактори трудової діяльності та умови праці користувачів ПК

Дослідження, проведені фахівцями Всесвітньої організації охорони здоров'я (ВООЗ) показали, що у професійних операторів та канцелярських службовців, які у своїй діяльності використовують ПК, частіше зустрічаються порушення органів зору, опорно-рухового апарату, центральної нервової, серцево-судинної, імунної та статеві систем, захворювання шкіри. Необхідно зазначити, що вже в перші роки впровадження ВДТ в Європі та США була зафіксована значна кількість скарг операторського персоналу на загальне недомогання, передчасне стомлювання, головний біль, порушення функцій органів зору, які здійснювали несприятливий психофізіологічний вплив на самопочуття та працездатність операторів [30]. Однак, в той час основна увага приділялась розвитку техніки, а людина залишалась без необхідного захисту.

В умовах сучасного виробництва, яке характеризується масовим характером та широким застосуванням комп'ютерної техніки попередні пріоритети зазнали суттєвої трансформації. У центрі уваги вітчизняних та зарубіжних фахівців є питання щодо визначення характеру та умов праці користувачів комп'ютерів, функціональних змін у динаміці виконання трудових завдань, захворюваності та стану здоров'я, розробки засобів захисту [30].

Дослідження медиків-гігієністів, психологів, світлотехніків та фахівців з охорони праці та ергономіки показали, що сучасна професія користувача ПК належить до розумової праці, яка характеризується: високою напруженістю зорових функцій; одноманітною позою; великою кількістю стереотипних висококоординованих рухів, що виконуються лише м'язами кистей рук на фоні малої загальної рухової активності; значним нервово-емоційним компонентом,

					КС КРБ 123.181.00.00 ПЗ			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>		<i>Курдичко С.В.</i>			<i>Безпека життєдіяльності, основи охорони праці</i>	<i>Літ.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Перевірив.</i>		<i>Луцків А.М.</i>					62	
<i>Консульт.</i>		<i>Сенчишин В.С.</i>				<i>ТНТУ, каф. КС, гр. СІ-42</i>		
<i>Н. Контр.</i>		<i>Луцик Н.С.</i>						
<i>Затверд.</i>		<i>Осухівська Г.М.</i>						

особливо в умовах дефіциту часу; роботою з великими масивами інформації, що викликає активізацію уваги та інших вищих психічних функцій. Крім того, при роботі з дисплеями на електронно-променевих трубках виникає вплив на користувача цілої низки факторів фізичної природи — електростатичні поля, радіочастотне та рентгенівське випромінювання [30].

Встановлено, що стан організму користувача значно залежить від типу роботи з ПК та умов її виконання. В загальному усі користувачі комп'ютерів поділяються на професіоналів та непрофесіоналів. До останніх можна віднести осіб, які використовують комп'ютер епізодично і він є для них не основним, а тільки допоміжним засобом (науково-технічні працівники, бібліотекарі, студенти, школярі, торговельні працівники).

Діяльність професіоналів можна поділити на три групи:

- діяльність, яка пов'язана з виконанням нескладних багаторазово повторюваних операцій, що не вимагають великого розумового напруження (робота операторів комп'ютерного набору, працівників довідкових служб);
- діяльність, яка пов'язана із здійсненням логічних операцій, що постійно повторюються – це робота інженера-економіста, інженера-проектувальника, оператора автоматизованого виробництва;
- діяльність, коли в процесі роботи необхідно приймати рішення за відсутності заздалегідь відомого алгоритму (робота інженера-програміста, диспетчерів руху залізничного транспорту, аеропортів).

Необхідно зазначити, що такий поділ досить умовний, оскільки дане питання ще не достатньо розроблене і потребує детального вивчення. Проте, зрозуміло, що для кожної категорії користувачів комп'ютерів характерні свої особливості впливу комплексу несприятливих факторів трудового процесу та умов праці.

В Інституті медицини праці Академії медичних наук України проводились дослідження інтенсивності захворюваності осіб, що використовують у своїй роботі комп'ютер. Була вивчена захворюваність працівників з різною тривалістю використання комп'ютерів та характером діяльності самих користувачів. Розглядалися три групи користувачів: у першу

					КС КРБ 123.181.00.00 ПЗ	Арк.
						63
Змн.	Арк.	№ докум.	Підпис	Дата		

увійшли системні інженери-програмісти (тривалість роботи за комп'ютером більше 6 год. на день), у другу — інженери-економісти, які у своїй роботі використовують уже розроблене програмне забезпечення (тривалість роботи від 4 до 6 год.), у третю — математики-постановники завдань, які використовували комп'ютери не більше, ніж 2 год. на день [30]. Дані про захворюваність різних груп користувачів комп'ютерів та контрольної групи наведено у таблиці 4.1.

Таблиця 4.1 – Рівень захворюваності (%) осіб

Стан здоров'я	Користувачі ВДТ			Контрольна група
	1 група	2 група	3 група	
Функціональні порушення ЦНС (астенопічний синдром та ін.)	15,6	8,2	6,3	2,7
Хвороби системи кровообігу	57,7	60,3	29,2	23,0
Хвороби органів дихання	20,0	21,7	11,2	4,1
Хвороби органів травлення	40,0	38,6	29,8	18,9
Здорові	6,7	20,1	29,8	46,6

Так, здорових серед обстежених користувачів ВДТ виявилось у кілька разів менше, ніж у контрольній групі. З наведених даних видно, що фізіологічні порушення частіше спостерігаються у користувачів, які довше та інтенсивніше використовували ВДТ.

Крім того, за даними ряду авторів у користувачів, які інтенсивно використовують комп'ютер в умовах значних розумових напружень досить часто (40—70%) виникають психологічні та поведінкові порушення (нервозність, роздратування, тривога, нерішучість, замкнутість) [15]. Працівниками кафедри охорони праці та екології Української академії друкарства та Українського науково-дослідного інституту поліграфічної промисловості ім. Т. Шевченка проведені дослідження умов праці та особливостей трудового процесу на комп'ютеризованих робочих місцях низки

підприємств, що займаються видавничо-поліграфічною діяльністю (редакції, видавництва, друкарні). Було, зокрема, встановлено, що за суб'єктивними показниками (скарги) робота з ВДТ викликає різноманітні симптоми негативного впливу на здоров'я користувачів. У таблицях в [15] наведені характеристики скарг операторів комп'ютерного набору та редакторів і коректорів, які в процесі своєї роботи використовували комп'ютер. Результати досліджень наглядно ілюструють, що у працівників різних професійних груп, що працюють за відеотерміналом комп'ютера переважають "очні" симптоми. Часті також головний біль та загальна втома, особливо в кінці робочого дня. Причому у коректорів та редакторів такі симптоми зустрічаються частіше. Досить значний відсоток скарг пов'язаний з опорно-м'язовою системою (біль в області спини та шиї, втома м'язів рук), особливо в операторів комп'ютерного набору. Характерним також є той факт, що чим більший стаж роботи за комп'ютером, тим очевидніший його несприятливий вплив на здоров'я користувача.

4.2 Надзвичайні ситуації, викликані вибухами і способи захисту від них

Вибух – це вивільнення значної кількості енергії в обмеженому об'ємі за короткий проміжок часу. Вибух здійснюється частіше за все за рахунок вивільнення хімічної енергії вибухових речовин [31].

Вибухові речовини (ВР) – це порох, динаміт, тротил, нітрогліцерин та інші хімічні речовини, їх сполуки або суміші, здатні вибухнути без доступу кисню.

За складом ВР поділяються на вибухові хімічні сполуки і вибухові суміші, а за призначенням – на ініціювальні (первинні) та бризантні (вторинні). Ініціювальна ВР (азид свинцю, гримуча ртуть, тетразен тощо) - високочутлива до найпростіших імпульсів маса (удар, тертя, наколювання, електрична іскра тощо), яка застосовується для збудження вибухових перетворень у зарядах вторинних ВР. Бризантні ВР мають велику швидкість детонації, тобто швидкість вибухового перетворення - до 8 км/сек; серед них можна виділити гексоген, октоген, тротил, тетрил [16].

					КС КРБ 123.181.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		65

Під вибуховими пристроями (ВП) розуміють саморобні чи виготовлені промисловим способом виробу одноразового застосування, спеціально підготовлені і за певних обставин спроможні за допомогою використання хімічної, теплової, електричної енергії або фізичного впливу (вибуху, удару) створити вражаючі ефекти - спричинити смерть, тілесні ушкодження, істотну матеріальну шкоду шляхом вивільнення, розсіювання або впливу ударної хвилі, спалаху світла, теплової енергії, осколків, токсичних речовин, радіоактивних матеріалів, біологічних агентів. ВП можуть бути штатні та виготовлені з використанням побутових предметів – портфелів, іграшок, пакетів, електроприладів. Вони можуть спрацьовувати від натискування, падіння, піднімання, впливу радіохвиль певної частоти [31].

Основними елементами ВП є заряд ВР або вибухонебезпечної суміші, засоби ініціювання (підричник, детонатор), обладнання для приведення ВП у дію та корпус ВП, хоч останній може бути й безоболонковим.

Підрич вибухового пристрою може здійснюватись механічним, електричним або вогневим способами. Для першого використовують механічний підричник – детонатор, ударник, пружину та запобіжну чеку. Для другого способу підричу застосовують електродетонатори, електробатарей або інше джерело живлення: ініціювання підричу відбувається шляхом замкнення електромережі і підричу детонатора (дистанційно або за допомогою проводів). Третій спосіб підричу передбачає для ініціювання детонатора використання вогнепровідного шнура.

Вибух – це процес надзвичайно швидкого горіння, що супроводжується швидким зростанням тиску і має велику руйнівну силу.

Якщо у повітряному середовищі виникає така концентрація пилю, парів або газів, яка досягає межі вибуху, то при наявності відкритого джерела вогню відбудеться вибух (табл.4.2).

Пил деяких речовин має такі значення нижньої межі вибуху (г/м³): цукор - 8,9; торф - 10,1; сіно - 20,2; тирса - 65,0.

Найбільш небезпечним є високодисперсний пил, бо він має велику сумарну поверхню, що створює підвищену хімічну активність.

					КС КРБ 123.181.00.00 ПЗ	Арк.
						66
Змн.	Арк.	№ докум.	Підпис	Дата		

Вибухи та пожежі можуть виникати за таких обставин:

- у початковий період експлуатації виробництва — період притирання елементів технологічного обладнання (недоліки допущенні у процесі проектування, неякісне виконання монтажних робіт);
- в основний період експлуатації виробництва (через несправність контрольно-вимірювальних приладів та елементів обладнання, порушення вимог безпеки, недостатній нагляд і контроль, незадовільні планово-профілактичні ремонти);
- у період так званого "старіння" елементів технологічного обладнання (через корозію матеріалів, зношеність деталей, відсутність капітальних і поточних ремонтів).

Таблиця 4.2. – Концентраційні межі вибуху (займання) речовин

Речовина	Нижня межа		Верхня межа	
	% за об'ємом	г/м ³ при 20 ⁰ С	% за об'ємом	г/м ³ при 20 ⁰ С
Метиловий спирт	6,7	46,5	38,5	512,0
Етиловий спирт	3,61	50,0	19,0	363,0
Бутан	1,8	37,4	8,5	204,8
Метан	5,28	16,66	15,4	102,6
Пропан	2,31	36,6	9,5	173,8
Ацетилен	2,50	16,5	82,0	885,6
Пропілен	2,30	34,8	11,1	169,0
Етилен	3,11	35,0	35,0	406,0
Бензол	1,43	42,0	9,5	308,0
Ксилол	1,0	44,0	7,6	34,0
Толуол	1,25	38,2	7,0	268,0
Ацетон	2,91	38,6	13,0	314,0
Аміак	17,0	112,0	27,0	189,0
Сірководень	4,0	61,0	44,5	628,0
Бензин паливний	2,4	137,0	4,9	281,0
Бензин розчинник	1,9	-	5,1	-
Водень	4,09	3,4	80,0	66,4

Причинами виникнення пожеж можуть бути конструктивні недоліки та порушення експлуатації пічного опалення, електрообладнання та

електроустаткування, інженерних комунікацій, дефекти обладнання; організаційні порушення режимів технологічних процесів, похибки при виконанні технологічних процесів, необережні дії персоналу та низький рівень їх кваліфікації, порушення правил при поводженні з вогнем.

Всі різноманітні причини пожеж можна об'єднати у дві великі групи, що пов'язані:

– I група - з недопустимою, з точки зору пожежної безпеки, появою горючого середовища в умовах, де є джерело вогню (розрив трубопроводів в котельнях, що працюють на рідкому паливі, підтікання паливних ліній, аварійний стан обладнання, викиди бітуму при варінні);

– II група - з недопустимою появою джерел вогню при наявності горючої суміші і окисника.

Сюди відносяться пожежі обумовлені такими чинниками як:

– порушення режимів роботи технологічного та інженерно-технічного обладнання;

– недоліки монтажу та порушення експлуатації електрообладнання, електропроводки, електроапаратури (коротке замикання, іскріння, перенавантаження проводів);

– недоліки при облаштуванні та експлуатації опалювальних систем та установок;

– недопустиме підвищенням температури при адіабатичному стисненні (робота компресорного устаткування);

– перегрів речовин, що обробляються, коли їх температура досягає температури самозаймання;

– порушення режиму зберігання та обробки самозаймистих речовин і матеріалів;

– необережне поводження з вогнем, незнання правил та норм пожежної безпеки, недбале ставлення до своїх обов'язків, навмисний підпал, накопичення електростатичних розрядів.

Безпосередньо причиною пожежі може стати у непередбачений час, у непередбаченому місці, з точки зору пожежної безпеки, поява того чи іншого

					КС КРБ 123.181.00.00 ПЗ	Арк.
						68
Змн.	Арк.	№ докум.	Підпис	Дата		

компоненту, який бере участь в процесі горіння, а відтак, з метою профілактики, недопустимими є такі умови, що можуть призводити до неконтрольованих процесів горіння.

Основними вражаючими факторами вибухів згідно з [31] є:

- повітряна ударна хвиля (ПУХ), що виникає при ядерних вибухах, вибухах детонуючих та ініціюючих речовин, при вибухових перетвореннях хмар паливно-повітряних сумішей, вибухів резервуарів з перегрітої рідиною і резервуарів під тиском;

- осколкові поля, створювані уламками різного роду об'єктів.

Основними параметрами вражаючих факторів є:

- повітряна ударна хвиля – надлишковий тиск в її фронті;
- осколкове поле – кількість осколків, їх кінетична енергія і радіус розльоту.

У результаті дії вражаючих факторів вибуху відбувається руйнування або пошкодження будівель, споруд, обладнання, елементів комунікації, і загибель людей і тварин.

Вторинними наслідками вибухів є ураження об'єктів, які знаходяться всередині приміщень, уламками завалених конструкцій будівлі. У результаті вибухів можуть виникнути пожежі, витік небезпечних речовин з пошкодженого обладнання. При пожежах і вибухах люди отримують термічні і механічні травми. Характерні опіки верхніх дихальних шляхів, тіла, черепно-мозкові травми, множинні переломи і удари, комбіновані ушкодження.

Для запобігання пожеж і вибухів необхідно виключити можливість утворення горючого і вибухонебезпечного середовища, а також запобігти появі в цих середовищах джерел загоряння [31].

Уражаюча дія вибуху, незалежно від його природи, може призводити до значних матеріальних і людських втрат. Особливо актуальним є питання збереження життя людей і забезпечення функціонування підприємств стратегічного значення в Україні в умовах війни на сході.

					КС КРБ 123.181.00.00 ПЗ	Арк.
						69
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У кваліфікаційній роботі розроблено портативну IoT-систему виявлення пошкоджень дорожнього покриття, призначену для автоматизованого моніторингу стану доріг із використанням засобів комп'ютерного зору, геолокації та мобільного передавання даних. Запропоноване рішення дає змогу зменшити залежність від ручного огляду дорожнього полотна та забезпечити більш оперативне отримання інформації про дефекти покриття.

У першому розділі проаналізовано проблему моніторингу стану дорожнього покриття, розглянуто основні чинники його руйнування та типові види пошкоджень. Також проаналізовано методи виявлення дефектів дорожнього полотна та обґрунтовано доцільність використання відеоаналізу в поєднанні з моделями глибинного навчання.

У другому розділі спроектовано апаратну частину портативної IoT-системи. Як центральний обчислювальний пристрій обрано NVIDIA Jetson Nano, що забезпечує можливість локальної обробки відеокадрів і запуску моделей комп'ютерного зору. Для отримання зображень дорожнього покриття використано камеру на базі сенсора Sony IMX219. Для визначення місця фіксації дефектів передбачено GPS-модуль NEO-6M, а для передавання результатів – 4G USB-модем Huawei E3372h-320. Така конфігурація забезпечує мобільність, автономність і придатність системи до використання на транспортному засобі.

У третьому розділі розроблено програмну частину системи. Запропоновано архітектуру програмного забезпечення, яка включає модулі отримання кадрів, попередню обробку зображень, виявлення дефектів, оцінювання рівня небезпеки, обробку GPS-даних, збереження результатів і їх передавання. Розроблена програмна логіка дозволяє виявляти пошкодження дорожнього покриття, визначати їхній тип, обчислювати відносну площу дефекту та присвоювати рівень небезпеки.

У четвертому розділі розглянуто питання, які стосуються охорони праці та безпеки життєдіяльності.

					КС КРБ 123.181.00.00 ПЗ	Арк.
						70
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Жаровський Р.О., Луцик Н.С., Осухівська Г.М., Паламар А.М., Тиш Є.В.. Методичні вказівки до виконання кваліфікаційної роботи бакалавра, розроблені у відповідності з освітньою програмою «Комп'ютерна інженерія» першого (бакалаврського) рівня вищої освіти за спеціальністю 123 «Комп'ютерна інженерія» галузі знань 12 «Інформаційні технології». Тернопіль: ТНТУ, 2024. 39 с.

2. Буров Є.В., Митник М.М. Комп'ютерні мережі. Підручник. Том другий. Львів: «Магнолія 2006», 2024. 204 с.

3. Паламар М.І., Стрембіцький М.О., Паламар А.М. Проектування комп'ютеризованих вимірювальних систем і комплексів. Навчальний посібник. Тернопіль: ТНТУ. 2019. 150 с.

4. Лупенко С.А., Пасічник В.В., Тиш Є.В. Комп'ютерна логіка. Навчальний посібник. Львів: Видавництво «Магнолія 2006», 2024. 354 с.

5. Бородій, Г. Осухівська. Порівняльний аналіз архітектур обробки IoT-даних для екологічного моніторингу на Raspberry Pi 5. Вісник Хмельницького національного університету. Технічні науки. 2026. Т. 365, №3. 365(3), С. 474-479. <https://doi.org/10.31891/2307-5732-2026-365-67>

6. Панасенко С.М., Луцик Н.С. Інтеграція Rule-Based алгоритмів та методів машинного навчання для підвищення точності тестування електроніки. Актуальні задачі сучасних технологій: Матеріали XIV Міжнародної наук.-техн. конф. молодих учених та студентів ТНТУ ім. І.Пулюя (11-12 грудня 2025). Тернопіль, 2025. с.324

7. Кондратюк Р., Тиш Є. Машинний зір: сутність технології, принципи роботи та сфери застосування. XIII науково-технічна конференція «Інформаційні моделі, системи та технології» Тернопіль: ТНТУ. 2025. С.120.

8. Бойко О.В., Степаненко О.С. Основи Інтернету речей та вбудованих систем: навч. посіб. Київ: КПІ ім. Ігоря Сікорського, 2020. 156 с.

9. Buaya R., Dastjerdi A. V., Kaufmann M. Internet of Things: Principles and Paradigms. 2016. 378 p.

					КС КРБ 123.181.00.00 ПЗ	Арк.
						71
Змн.	Арк.	№ докум.	Підпис	Дата		

10. Gonzalez R.C., Woods R.E. Digital Image Processing. 4th ed. New York: Pearson, 2018. 1192 p.

11. Bradski G., Kaehler A. Learning OpenCV: Computer Vision with the OpenCV Library. Sebastopol: O'Reilly Media, 2008. 580 p.

12. Maeda H., Sekimoto Y., Seto T., Kashiya T., Omata H. Road Damage Detection and Classification Using Deep Neural Networks with Smartphone Images. Computer-Aided Civil and Infrastructure Engineering. 2018. Vol. 33, No. 12. P. 1127–1141. URL: <https://doi.org/10.1111/mice.12387> (дата звернення: 30.05.2026 р.).

13. OpenCV. OpenCV Documentation. URL: <https://docs.opencv.org/4.x/> (дата звернення: 01.06.2026 р.).

14. OpenCV. Image Filtering. URL: https://docs.opencv.org/4.x/d4/d86/group__imgproc__filter.html (дата звернення: 01.06.2026 р.).

15. OpenCV. Structural Analysis and Shape Descriptors. URL: https://docs.opencv.org/4.x/d3/dc0/group__imgproc__shape.html (дата звернення: 01.06.2026 р.).

16. Ultralytics. YOLO Python Usage Documentation. URL: <https://docs.ultralytics.com/usage/python/> (дата звернення: 01.06.2026 р.).

17. Ultralytics. YOLO Models Documentation. URL: <https://docs.ultralytics.com/models/> (дата звернення: 01.06.2026 р.).

18. NVIDIA. Jetson Nano Developer Kit User Guide. URL: https://developer.download.nvidia.com/assets/embedded/secure/jetson/Nano/docs/NV_Jetson_Nano_Developer_Kit_User_Guide.pdf (дата звернення: 02.06.2026 р.).

19. NVIDIA. JetPack SDK Documentation. URL: <https://developer.nvidia.com/embedded/jetpack> (дата звернення: 02.06.2026 р.).

20. NEO-6 u-blox 6 GPS Modules Data Sheet. URL: https://content.u-blox.com/sites/default/files/products/documents/NEO-6_DataSheet_%28GPS.G6-HW-09005%29.pdf (дата звернення: 02.06.2026 р.).

21. NEO-6 Receiver Description Including Protocol Specification. URL: https://content.u-blox.com/sites/default/files/products/documents/u-blox6_ReceiverDescrProtSpec_%28GPS.G6-SW-10018%29.pdf (дата звернення: 02.06.2026 р.).

					КС КРБ 123.181.00.00 ПЗ	Арк.
						72
Змн.	Арк.	№ докум.	Підпис	Дата		

22. Huawei. Huawei E3372 LTE USB Stick Product Information. URL: <https://consumer.huawei.com/> (дата звернення: 02.06.2026 р.).

23. Python Software Foundation. Python 3 Documentation. URL: <https://docs.python.org/3/> (дата звернення: 02.06.2026 р.).

24. Python Software Foundation. sqlite3 — DB-API 2.0 interface for SQLite databases. URL: <https://docs.python.org/3/library/sqlite3.html> (дата звернення: 02.06.2026 р.).

25. Python Software Foundation. datetime — Basic date and time types. URL: <https://docs.python.org/3/library/datetime.html> (дата звернення: 02.06.2026 р.).

26. Requests: HTTP for Humans Documentation. URL: <https://requests.readthedocs.io/en/latest/> (дата звернення: 02.06.2026 р.).

27. SQLite Consortium. SQLite Documentation. URL: <https://sqlite.org/docs.html> (дата звернення: 02.06.2026 р.).

28. Lutz M. Learning Python. 5th ed. Sebastopol: O'Reilly Media, 2013. 1648 p.

29. McKinney W. Python for Data Analysis. 3rd ed. Sebastopol: O'Reilly Media, 2022. 579 p.

30. Катренко Л.А., Катренко А.В. Охорона праці в галузі комп'ютерингу. Львів: Магнолія-2006. 2012. 544 с.

31. Гурик О.Я., Окіпний І.Б. Методичні вказівки для написання розділу «Безпека життєдіяльності, основи охорони праці» в кваліфікаційних роботах здобувачів освітнього рівня «бакалавр». Для студентів всіх форм навчання, рівень вищої освіти перший (бакалаврський). Тернопіль: ТНТУ імені Івана Пулюя. 2021. 20 с.

					КС КРБ 123.181.00.00 ПЗ	Арк.
						73
Змн.	Арк.	№ докум.	Підпис	Дата		

Додаток А
Технічне завдання

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії

Кафедра комп'ютерних систем та мереж

“Затверджую”

Завідувач кафедри КС

_____ Осухівська Г.М.

“ 02 ” лютого 2026 р.

ПОРТАТИВНА ІоТ-СИСТЕМА ВИЯВЛЕННЯ ПОШКОДЖЕНЬ

ДОРОЖНЬОГО ПОКРИТТЯ

ТЕХНІЧНЕ ЗАВДАННЯ

на 13 листках

Вид робіт:

Кваліфікаційна робота

На здобуття освітнього ступеня «Бакалавр»

Спеціальність 123 «Комп'ютерна інженерія»

«УЗГОДЖЕНО»

«ВИКОНАВЕЦЬ»

Керівник кваліфікаційної роботи

Студент групи СІ-42

_____ к.т.н., доц. Луцків А.М.

_____ Курдичко С.В.

« 02 » лютого 2026 р.

« 02 » лютого 2026 р.

Тернопіль 2026

1 Загальні відомості

1.1 Повна назва та її умовне позначення

Повна назва теми кваліфікаційної роботи: «Портативна IoT-система виявлення пошкоджень дорожнього покриття».

Умовне позначення кваліфікаційної роботи: КС КРБ 123.181.00.00

1.2 Виконавець

Студент групи СІ-42, факультету комп'ютерно-інформаційних систем і програмної інженерії, кафедри комп'ютерних систем та мереж, Тернопільського національного технічного університету імені Івана Пулюя, Курдичко Сергій Володимирович.

1.3 Підстава для виконання роботи

Підставою для виконання кваліфікаційної роботи є наказ по університету (№4/9-188 від 24.04.2026 р.)

1.4 Планові терміни початку та завершення роботи

Плановий термін початку виконання кваліфікаційної роботи – 26.01.2026 р.

Плановий термін завершення виконання кваліфікаційної роботи – 21.06.2026 р.

1.5 Порядок оформлення та пред'явлення результатів роботи

Порядок оформлення пояснювальної записки та графічного матеріалу здійснюється у відповідності до чинних норм та правил ISO, ЕСКД, ЕСПД та ДСТУ.

Пред'явлення проміжних результатів роботи з виконання кваліфікаційної роботи здійснюється у відповідності до графіку, затвердженого керівником роботи.

Попередній захист кваліфікаційної роботи відбувається при готовності роботи на 90% , наявності пояснювальної записки та графічного матеріалу.

Пред'явлення результатів кваліфікаційної роботи відбувається шляхом захисту на відповідному засіданні ЕК, ілюстрацією основних досягнень за допомогою графічного матеріалу.

2 Призначення і цілі створення системи

2.1 Призначення системи

Портативна IoT-система виявлення пошкоджень дорожнього покриття розробляється як апаратно-програмне рішення для автоматизованого моніторингу стану дорожнього полотна. Система призначена для отримання зображень дорожнього покриття, їх програмної обробки, виявлення характерних дефектів, визначення координат місця фіксації та передавання результатів користувачу або серверній частині.

Система повинна забезпечувати виявлення пошкоджень дорожнього покриття без постійної участі оператора. Основними дефектами, які можуть бути виявлені під час аналізу зображень, є поздовжні тріщини, поперечні тріщини, сітчасті руйнування та вибоїни. Для кожного виявленого дефекту система повинна формувати інформаційний запис із зазначенням типу пошкодження, рівня небезпеки, координат, дати й часу фіксації та посилання на збережене зображення.

Для роботи системи передбачається використання камери, центрального обчислювального пристрою, GPS-модуля, 4G USB-модема, локального накопичувача даних і програмного забезпечення для комп'ютерного зору. Камера отримує зображення дорожнього покриття, обчислювальний пристрій виконує обробку кадрів і запуск моделі виявлення дефектів, GPS-модуль визначає координати місця фіксації, а 4G-модем забезпечує передавання результатів через мобільну мережу.

Система призначена для підвищення оперативності моніторингу стану доріг, зменшення залежності від ручного огляду дорожнього покриття та формування структурованої бази даних пошкоджень.

2.2 Мета створення системи

Метою створення портативної IoT-системи виявлення пошкоджень дорожнього покриття є розроблення апаратно-програмного рішення, яке забезпечує отримання зображень дорожньої поверхні, автоматизоване виявлення дефектів засобами комп'ютерного зору, визначення рівня небезпеки пошкоджень, фіксацію GPS-координат і збереження результатів у структурованому вигляді.

Досягнення мети роботи можливе шляхом розв'язання таких задач:

- проведення аналізу проблеми моніторингу стану дорожнього покриття;
- визначення основних типів пошкоджень дорожнього полотна;
- аналіз автоматизованих методів виявлення дефектів дорожнього покриття;
- формування вимог до портативної IoT-системи;
- розроблення загальної архітектури системи;
- обґрунтування вибору апаратного забезпечення;
- розроблення алгоритму роботи програмного забезпечення;
- збереження результатів у локальній базі даних SQLite.

2.3 Характеристика об'єкту

Об'єктом проектування є портативна IoT-система виявлення пошкоджень дорожнього покриття, яка виконує автоматизований аналіз зображень дорожньої поверхні за допомогою камери, центрального обчислювального пристрою, GPS-модуля, модуля передавання даних і програмних алгоритмів комп'ютерного зору.

Система складається з апаратної та програмної частин. Апаратна частина забезпечує отримання зображень дорожнього покриття, локальну обробку даних, визначення координат і передавання результатів. Програмна частина виконує попередню обробку кадрів, виявлення пошкоджень, визначення рівня небезпеки, формування запису про дефект, збереження даних і їх передавання.

До складу апаратної частини системи входять мікроконтролер, камера, GPS-модуль, 4G USB-модем, накопичувач даних, джерело живлення, корпус, кріплення камери та з'єднувальні кабелі.

Мікроконтролер виконує роль центрального вузла системи. Він отримує відеокадри з камери, виконує попередню обробку зображень, запускає модель комп'ютерного зору, обробляє результати виявлення, отримує координати від GPS-модуля, формує запис про дефект і забезпечує збереження або передавання результатів.

Камера використовується для отримання зображень дорожнього покриття. GPS-модуль визначає координати місця фіксації пошкодження. 4G USB-модем забезпечує передавання результатів через мобільний інтернет. Локальна база даних використовується для збереження записів про дефекти у випадку автономної роботи або тимчасової відсутності зв'язку.

3 Вимоги до системи

3.1 Вимоги до системи в цілому

Основними вимогами до системи є:

- отримання зображень або відеокадрів дорожнього покриття за допомогою камери;
- попередня обробка кадрів перед подачею на модель комп'ютерного зору;
- виділення області контролю дорожнього покриття;
- зміна розміру зображення до формату, придатного для моделі;
- виявлення пошкоджень дорожнього покриття;
- класифікація дефектів за основними типами;
- визначення рівня небезпеки виявленого пошкодження;
- отримання GPS-координат місця фіксації дефекту;
- формування запису про виявлене пошкодження;
- збереження результатів у локальній базі даних;
- збереження зображень або фрагментів кадрів із виявленими дефектами;
- передавання результатів через 4G-з'єднання;
- повторна синхронізація даних після відновлення мережевого з'єднання;
- відображення результатів у вигляді таблиці або карти пошкоджень;
- забезпечення модульної структури програмного забезпечення.

Система повинна бути придатною для встановлення на транспортному засобі або використання як переносний засіб моніторингу. Апаратні компоненти повинні бути сумісними між собою, доступними для реалізації прототипу та достатніми для виконання задач комп'ютерного зору.

Програмне забезпечення повинно забезпечувати циклічну роботу системи. Після обробки одного кадру система повинна переходити до отримання наступного

кадру, а у випадку виявлення дефекту – формувати, зберігати та передавати відповідний запис.

3.1.1 Вимоги до способів та засобів зв'язку між компонентами системи

Основним пристроєм, який забезпечує обмін даними між компонентами, є мікроконтролер. До нього підключаються всі основні апаратні модулі системи.

Взаємодія між компонентами повинна виконуватися такими способами:

- камера підключається через CSI-інтерфейс;
- GPS-модуль підключається через UART або USB-UART адаптер;
- 4G USB-модем підключається через USB-інтерфейс;
- локальна база даних зберігається у файловій системі накопичувача;
- передавання результатів на сервер виконується через 4G-з'єднання;
- дані для передавання формуються у структурованому форматі JSON.

Під час підключення компонентів необхідно враховувати вимоги до живлення, стабільність з'єднань і можливість роботи системи під час руху транспортного засобу. Кабельні з'єднання повинні бути зафіксовані так, щоб вібрації не призводили до втрати сигналу або від'єднання модулів.

3.1.2 Вимоги по діагностуванню системи

Діагностування системи повинно передбачати перевірку працездатності основних апаратних і програмних компонентів.

Під час запуску системи повинні перевірятися:

- доступність камери;
- можливість отримання кадру з відеопотоку;
- коректність роботи модуля попередньої обробки зображень;
- доступність файлу моделі комп'ютерного зору;
- можливість запуску моделі виявлення пошкоджень;
- наявність зв'язку з GPS-модулем;

- коректність отримання GPS-координат;
- доступність 4G USB-модема.

У разі виникнення помилки система повинна виводити повідомлення, яке дозволяє визначити причину несправності.

До діагностичних заходів також належить перевірка фізичних з'єднань між контролером керування та периферійними модулями, зокрема CSI-інтерфейсу камери, USB-підключення модема, UART-з'єднання GPS-модуля та лінії живлення.

3.1.3 Перспективи розвитку, модернізація системи

Перспективи розвитку системи пов'язані з підвищенням точності виявлення дефектів дорожнього покриття, розширенням набору розпізнаваних пошкоджень і вдосконаленням засобів візуалізації результатів.

Одним із напрямів модернізації є використання точнішої моделі комп'ютерного зору або моделі сегментації, яка дозволить не лише формувати обмежувальну рамку навколо дефекту, а й точніше визначати контур пошкодженої ділянки. Це дасть змогу краще оцінювати площу пошкодження.

Іншим напрямом розвитку є використання камери з ширшим кутом огляду, вищою роздільною здатністю або кращою роботою в умовах слабкого освітлення. Також система може бути доповнена інерціальним модулем, датчиком освітленості або датчиком відстані для уточнення умов зйомки та підвищення достовірності результатів.

Система також може бути розширена шляхом створення веб-інтерфейсу для перегляду карти пошкоджень, журналу подій, статистики дефектів і збережених зображень. Додатковим напрямом модернізації є використання декількох портативних вузлів, які передаватимуть результати до спільної серверної частини.

3.1.4 Вимоги до надійності системи

Портативна IoT-система виявлення пошкоджень дорожнього покриття повинна забезпечувати стабільне виконання основних функцій під час отримання, обробки та передавання даних.

Надійність системи повинна забезпечуватися як на апаратному, так і на програмному рівнях. NVIDIA Jetson Nano повинен мати стабільне джерело живлення. Для тривалої роботи доцільно передбачити охолодження обчислювального пристрою, оскільки під час обробки відеокадрів і запуску моделі комп'ютерного зору система може працювати з підвищеним навантаженням.

Програмне забезпечення повинно передбачати обробку помилок. Якщо камера недоступна, GPS-модуль не повертає коректні координати, база даних не відкривається або сервер недоступний, програма повинна повідомити про помилку або перейти до резервного сценарію роботи.

Система повинна забезпечувати:

- стабільне отримання відеокадрів;
- коректну попередню обробку зображень;
- стабільне виконання моделі виявлення пошкоджень;
- коректне визначення рівня небезпеки дефекту;
- отримання або збереження останніх достовірних GPS-координат;
- запис результатів у локальну базу даних;
- збереження зображень дефектів;
- роботу в умовах тимчасової відсутності 4G-з'єднання;
- повторне передавання накопичених записів після відновлення зв'язку;
- можливість повторного запуску програми після помилки або зупинки.

3.1.5 Вимоги до функцій системи

Портативна IoT-система виявлення пошкоджень дорожнього покриття повинна забезпечувати повний цикл роботи: від отримання кадру дорожнього покриття до збереження та передавання результату виявлення дефекту.

До основних функцій системи належать:

- ініціалізація камери, GPS-модуля, бази даних і модуля передавання даних;
- отримання кадрів дорожнього покриття;
- перевірка коректності отриманого кадру;
- виділення області контролю;
- зміна розміру зображення;
- корекція яскравості та зменшення шумів;
- нормалізація кадру перед подаванням на модель;
- виявлення пошкоджень дорожнього покриття за допомогою моделі комп'ютерного зору;
- визначення типу дефекту;
- отримання координат обмежувальної рамки дефекту;
- визначення рівня впевненості моделі;
- обчислення відносної площі виявленого дефекту;
- визначення рівня небезпеки пошкодження;
- отримання GPS-координат місця фіксації;
- формування запису про виявлене пошкодження;
- збереження результату у базі даних SQLite;
- збереження зображення або фрагмента кадру;
- формування JSON-повідомлення для передавання;
- передавання результатів через 4G-з'єднання;
- збереження записів зі статусом очікування у випадку відсутності мережі;
- відображення результатів у вигляді таблиці, журналу подій або карти пошкоджень.

Мікроконтролер повинен виконувати такі задачі:

- отримувати кадри з камери;
- виконувати попередню обробку зображень;
- запускати модель виявлення пошкоджень;
- обробляти результати розпізнавання;
- взаємодіяти з GPS-модулем;
- формувати записи про дефекти;
- зберігати дані у SQLite;
- передавати результати через 4G-модем;
- забезпечувати циклічну роботу системи.

3.1.6 Вимоги до апаратного забезпечення

Апаратне забезпечення системи повинно бути доступним, сумісним між собою та достатнім для реалізації портативної IoT-системи.

Основними апаратними компонентами системи повинні бути:

- NVIDIA Jetson Nano;
- камера на базі сенсора Sony IMX219;
- GPS-модуль NEO-6M;
- 4G USB-модем Huawei E3372h-320;
- SIM-карта мобільного оператора для передавання даних;
- microSD-карта або USB-накопичувач;
- джерело живлення;
- DC-DC перетворювач для живлення від бортової мережі транспортного засобу;
- корпус для розміщення компонентів;
- кріплення для камери;
- з'єднувальні кабелі;
- елементи охолодження для NVIDIA Jetson Nano.

3.1.7 Вимоги до програмного забезпечення

Програмне забезпечення системи повинно бути реалізоване мовою Python і мати модульну структуру. Для реалізації програмного забезпечення повинні використовуватися:

- Linux-середовище на NVIDIA Jetson Nano;
- Python як основна мова програмування;
- OpenCV для обробки зображень;
- ultralytics для роботи з моделлю YOLO;
- sqlite3 для роботи з локальною базою даних SQLite;
- засоби обробки GPS-даних;
- файлове сховище для збереження зображень дефектів.

Програмне забезпечення повинно забезпечувати циклічну роботу системи. Після обробки одного кадру система повинна переходити до наступного кадру. У разі виявлення пошкодження програма повинна сформувати запис, зберегти його у базі даних і за можливості передати результат через 4G-з'єднання.

4 Вимоги до документації

Документація повинна відповідати вимогам ЄСКД та ДСТУ

Комплект документації повинен складатись з:

- пояснювальної записки;
- графічного матеріалу:
 - 1 Фактори впливу на стан дорожнього покриття.
 - 2 Архітектура портативної IoT-системи виявлення пошкоджень дорожнього покриття.
 - 3 Діаграма потоків даних в портативній IoT-системі.
 - 4 Архітектурні рівні та алгоритм роботи портативної IoT-системи.

*Примітка: У комплект документації можуть вноситися міни та доповнення в процесі розробки.

5 Стадії та етапи проектування

Таблиця 1 – Стадії та етапи виконання кваліфікаційної роботи бакалавра

№ етапу	Назва етапу виконання кваліфікаційної роботи	Термін виконання
1	Розробка технічного завдання	26.01 – 02.02
2	Робота над першим розділом «Аналіз принципів та вимог до процесу виявлення пошкоджень дорожнього покриття»	03.02 – 15.02
3	Робота над другим розділом «Проектування портативної IoT-системи виявлення пошкоджень дорожнього покриття»	20.04 – 25.04
4	Робота над третім розділом «Програмні модулі портативної IoT-системи виявлення пошкоджень дорожнього покриття»	26.04 – 05.05
5	Робота над четвертим розділом «Безпека життєдіяльності, основи охорони праці»	07.05 – 25.05
6	Оформлення пояснювальної записки і графічного матеріалу	26.05 – 7.06
7	Перевірка на академічний плагіат, перевірка керівником та консультантами	8.06 – 14.06
8	Попередній захист кваліфікаційної роботи бакалавра	15.06 – 21.06
9	Захист кваліфікаційної роботи бакалавра	26.06.2026

6 Додаткові умови виконання кваліфікаційної роботи

Під час виконання кваліфікаційної роботи у дане технічне завдання можуть вноситися зміни та доповнення.

Додаток Б
Лістинг класу DamageStorage

```
import sqlite3
from datetime import datetime

class DamageStorage:
    def __init__(self, database_path="road_damage.db"):
        """
        database_path - шлях до файлу локальної бази даних.
        """
        self.database_path = database_path
        self.create_table()

    def create_table(self):
        """
        Створення таблиці для збереження результатів виявлення.
        """
        connection = sqlite3.connect(self.database_path)
        cursor = connection.cursor()

        cursor.execute("""
            CREATE TABLE IF NOT EXISTS damages (
                id INTEGER PRIMARY KEY AUTOINCREMENT,
                class_name TEXT NOT NULL,
                risk_level TEXT NOT NULL,
                confidence REAL,
                latitude REAL,
                longitude REAL,
                detected_at TEXT,
                image_path TEXT,
                sync_status TEXT DEFAULT 'pending'
            )
        """)

        connection.commit()
```

```
connection.close()
```

```
def save_damage(self, damage_data):
```

```
    """
```

```
    Збереження запису про виявлене пошкодження.
```

```
    """
```

```
    connection = sqlite3.connect(self.database_path)
```

```
    cursor = connection.cursor()
```

```
    cursor.execute("""
```

```
        INSERT INTO damages (
```

```
            class_name,
```

```
            risk_level,
```

```
            confidence,
```

```
            latitude,
```

```
            longitude,
```

```
            detected_at,
```

```
            image_path,
```

```
            sync_status
```

```
        )
```

```
        VALUES (?, ?, ?, ?, ?, ?, ?, ?)
```

```
    """, (
```

```
        damage_data.get("class_name"),
```

```
        damage_data.get("risk_level"),
```

```
        damage_data.get("confidence"),
```

```
        damage_data.get("latitude"),
```

```
        damage_data.get("longitude"),
```

```
        damage_data.get(
```

```
            "detected_at",
```

```
            datetime.now().strftime("%Y-%m-%d %H:%M:%S")
```

```
        ),
```

```
        damage_data.get("image_path"),
```

```
        damage_data.get("sync_status", "pending")
```

```
    ))
```

```
    connection.commit()
```

```
    record_id = cursor.lastrowid
```

```

connection.close()

return record_id

def get_pending_records(self):
    """
    Отримання записів, які ще не були передані на сервер.
    """
    connection = sqlite3.connect(self.database_path)
    connection.row_factory = sqlite3.Row
    cursor = connection.cursor()

    cursor.execute("""
        SELECT *
        FROM damages
        WHERE sync_status = 'pending'
    """)
    records = [dict(row) for row in cursor.fetchall()]
    connection.close()

    return records

def mark_as_sent(self, record_id):
    """
    Оновлення статусу запису після успішного передавання.
    """
    connection = sqlite3.connect(self.database_path)
    cursor = connection.cursor()

    cursor.execute("""
        UPDATE damages
        SET sync_status = 'sent'
        WHERE id = ?
    """, (record_id,))

    connection.commit()
    connection.close()

```

Додаток В

Лістинг класу DamageSender

```
import requests

class DamageSender:
    def __init__(self, server_url):
        """
        server_url - адреса серверної частини для приймання
        результатів.
        """
        self.server_url = server_url

    def send_record(self, record):
        """
        Передавання одного запису на сервер.
        """
        try:
            response = requests.post(
                self.server_url,
                json=record,
                timeout=10
            )

            return response.status_code == 200

        except requests.RequestException:
            return False

    def synchronize(self, storage):
        """
        Передавання всіх локальних записів зі статусом pending.
        """
        pending_records = storage.get_pending_records()

        for record in pending_records:
            success = self.send_record(record)

            if success:
                storage.mark_as_sent(record["id"])
```