Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА

Кіберфізична система інтелектуальної системи допомоги водієві (ADAS – Advanced Driver Assistance Systems) для виявлення об'єктів в умовах недостатньої видимості
Назва теми

Рівень вищої освіти *перший (бакалаврський)*

Галузь знань *12 «Інформаційні технології»*
Шифр, назва

Спеціальність *123 «Комп'ютерна інженерія»*
Шифр, назва

Освітня програма *«Комп'ютерна інженерія та програмування»*
Назва

Шифр КвРКІ 220008.22.01.30 ПЗ

Виконав здобувач IV курсу, група КІ2-22-1 _____ Назарій ВАСИЛЮК
Підпис Ініціали, прізвище

Керівник ст.в. _____ Олег ВОЙЧУР
Науковий ступінь, учене звання Підпис Ініціали, прізвище

Нормоконтролер канд.фіз.-мат.наук, доц. _____ Тетяна КИСІЛЬ
Науковий ступінь, учене звання Підпис Ініціали, прізвище

До захисту допускаю:
завідувач кафедри КІІС _____ Ольга ПАВЛОВА
«01» червня 2026 р. Підпис Ініціали, прізвище
дата

Хмельницький 2026

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Рівень вищої освіти ПЕРШИЙ (БАКАЛАВРСЬКИЙ)

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ
Завідувачка кафедри КІІС
_____ Ольга ПАВЛОВА

" 10 " 01 2026 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Василюку Назарію Сергійовичу
Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Кіберфізична система інтелектуальної системи допомоги водієві (ADAS – Advanced Driver Assistance Systems) для виявлення об'єктів в умовах недостатньої видимостів
Керівник проекту (роботи) Войчур Олег Юрійович
Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 20.01.2026 р. № 7
2. Термін подання здобувачем роботи на кафедру 01.06.2026 р.
3. Вихідні дані до роботи Завдання на кваліфікаційну роботу
4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)
КФС виявлення об'єктів в умовах недостатньої видимості та постановка задачі щодо її удосконалення
Проєктування системи обробки відеоінформації у кфс інтелектуальної підтримки водія
Реалізація та експериментальне дослідження кфс adas в умовах обмеженої видимості
5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)
Програмна архітектура кіберфізичної системи
Організація програмного забезпечення
Апаратна складова проєкту

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання «_10_»_____01_____2026 р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напряму дослідження та узгодження тематики кваліфікаційної роботи з керівником	10.01.2026	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2026	виконано
3	Робота над розділом 1 – теоретичні основи досліджуваної проблеми	01.03.2026	виконано
4	Робота над розділом 2 – проєктування та обґрунтування архітектури системи	01.04.2026	виконано
5	Робота над розділом 3 – програмно–апаратна реалізація та тестування системи	29.04.2026	виконано
6	Оформлення пояснювальної записки згідно вимог	21.05.2026	виконано
7	Попередній захист ВКР	25.05.2026	виконано
8	Захист ВКР на засіданні ЕК	Червень 2026 року	

Здобувач_____ Назарій ВАСИЛЮК
Підпис Ім'я, ПРІЗВИЩЕ

Керівник кваліфікаційної роботи_____ Олег ВОЙЧУР
Підпис Ім'я, ПРІЗВИЩЕ

№ ряда	формат	Позначення	Найменування	Кіл. листів	№ екз	Примітка
			Текстові документи			
1		КвРКІ 220008.22.01.30 ПЗ	Пояснювальна записка	64		
			Графічні матеріали			
2		КвРКІ 220008.22.01.30 Е8	Програмна архітектура кіберфізичної системи	1		
3		КвРКІ 220008.22.01.30 Е8	Організація програмного забезпечення	1		
4		КвРКІ 220008.22.01.30 Е8	Апаратна складова проєкту	1		

Зм	Арк	№ докум	Підпис	Дата	КвРКІ 220008.22.01.30 ВП		
Розробив		Василюк			Відомість проекту	Літера	Аркуш	Аркушів
Перевір.		Войчур				У	1	1
Н. контр.		Кисіль				ХНУ, КІ2–22–1		
Затв.		Павлова		01.06				

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Кіберфізична система інтелектуальної системи допомоги водієві (ADAS - Advanced Driver Assistance Systems) для виявлення об'єктів в умовах недостатньої видимості».

Автор роботи: Назарій ВАСИЛЮК.

Керівник роботи: Олег ВОЙЧУР.

Пояснювальна записка: 64 с., 16 рис., 15 табл., 3 дод., 45 джерел.

Графічна частина: 3 креслення.

ADAS, ВИЯВЛЕННЯ ОБ'ЄКТІВ, КІБЕРФІЗИЧНА СИСТЕМА, КОМП'ЮТЕРНИЙ ЗІР, МОНОКУЛЯРНА КАМЕРА, НЕЙРОННА МЕРЕЖА, НІЧНА ВИДИМІСТЬ, ТРЕКІНГ, ФІЛЬТР КАЛМАНА, CLAHE, YOLO

Метою дипломної роботи є дослідження принципів побудови та особливостей роботи кіберфізичної системи інтелектуальної підтримки водія (ADAS), а також розробка та аналіз методів опрацювання візуальної інформації для забезпечення надійного виявлення й класифікації об'єктів дорожнього середовища в умовах низької освітленості та обмеженої видимості. Об'єктом дослідження є процеси виявлення та супроводження об'єктів в підсистемах кіберфізичної системи ADAS. Предметом дослідження є алгоритми та режими опрацювання даних відеопотоку в системі підтримки водія за умов недостатньої освітленості та погіршеної прозорості атмосфери. У процесі виконання дипломної роботи застосовувався метод систематичного аналізу науково-технічної літератури з метою вивчення предметної галузі, огляду архітектурних рішень систем ADAS, а також дослідження актуальних підходів до розв'язання задач комп'ютерного зору в умовах реального дорожнього середовища.

Підпис здобувача

30.05.2026

Дата

ЗМІСТ

Вступ 4
1 КФС виявлення об'єктів в умовах недостатньої видимості та постановка задачі щодо її удосконалення 6
1.1 Дослідження архітектурних та операційних характеристик 6
1.2 Аналіз комплексів обробки даних у КФС підтримки водія 9
1.3 Аналіз вимог до системи та розробка технічного завдання 11
1.4 Висновки до першого розділу 12
2 Проєктування системи обробки відеоінформації у КФС інтелектуальної підтримки водія 14
2.1 Визначення апаратних і програмних підсистем системи ADAS 14
2.2 Покращення відеозображення в умовах низької освітленості 15
2.3 Нейромережева детекція та трекінг об'єктів (YOLOv8 + ByteTrack) ... 17
2.4 Оцінка відстані до об'єктів на основі пінхол моделі камери 21
2.5 Фільтр Калмана для оцінки відстані та розрахунку швидкості 24
2.6 Детектор смуги (LaneDetector) 26
2.7 Система попереджень та рівні загрози (WarningLevel) 29
2.8 Висновки до другого розділу 32
3 Реалізація та експериментальне дослідження КФС ADAS в умовах обмеженої видимості 34
3.1 Апаратна платформа та середовище розгортання системи 34
3.2 Програмна реалізація та інтеграція модулів системи 37
3.3 Опис тестового відеоматеріалу та умов проведення експерименту 44
3.4 Аналіз результатів детекції об'єктів у нічних умовах 48
3.5 Оцінка точності визначення відстані та роботи фільтра Калмана 51
3.6 Аналіз роботи системи попереджень та детектора смуги руху 53
3.7 Висновки до третього розділу 57

Висновки 59
Перелік джерел посилань 61
Додаток А «Програмна архітектура кіберфізичної системи» 65
Додаток Б «Організація програмного забезпечення» 66
Додаток В «Апаратна складова проєкту» 67

ВСТУП

Сучасний автомобіль усе більше покладається на електроніку, яка частково виконує функції водія. Системи активної допомоги (ADAS) працюють як класичні кіберфізичні комплекси: камери й сенсори безперервно сканують дорогу, передають картинку на обчислювальний модуль, а той аналізує ситуацію та попереджає про загрозу зіткнення.

Коли на вулиці сонячно й сухо, усі системи працюють добре. Проблеми починаються тоді, коли погіршується видимість. Уночі, під час сильного дощу чи в тумані. У таких умовах контрастність кадру критично падає, матриця камери починає сильно шуміти, через що звичайні алгоритми комп'ютерного зору пропускають об'єкти або видають помилки. Ідея цієї роботи полягає у пошуку рішення для стабільної роботи систем ADAS в таких складних умовах.

Метою цього дослідження слугує аналітика архітектури кіберфізичних систем ADAS та реалізація методів обробки відеопотоку, що дозволять надійно знаходити об'єкти на дорозі, навіть при поганому освітленні та під час негоди. У проекті будуть використані алгоритми, які допомагають попередньо покращити якість кадру, нейромережеві моделі для розпізнавання об'єктів, а також методи фільтрації координат і відстані в умовах нічної їзди чи опадів.

Для досягнення мети потрібно вирішити декілька завдань:

– розібрати існуючі архітектурні рішення ADAS і чітко визначити, чому й де саме вони дають збій в темряві;

– підібрати та обґрунтувати набір алгоритмів, які зможуть працювати в одному ланцюжку (пайплайні) в реальному часі;

– написати програмний прототип системи, орієнтований на платформи з апаратним прискоренням (GPU). Звичайний процесор такий обсяг операцій не опрацює;

– Протестувати готовий прототип на реальних відеозаписах нічних поїздок, щоб оцінити точність роботи на практиці.

Основою досліджень для цього проекту є використання різних підходів до комп’ютерного зору, обробки сигналів та машинного навчання. Замість проектування системи повністю з нуля, було реалізована оптимізована структура з кількох перевірених технологій:

– CLAHE (адаптивна гістограмна еквалізація з обмеженням контрасту) використовується на першому етапі, щоб витягнути деталі з темряви, вирівняти світлову картинку та підготувати кадр до аналізу.

– YOLOv8 – сучасна й швидка нейромережа, яка відповідає за детекцію машин, людей чи перешкод на сцені в реальному часі.

– ByteTrack – алгоритм трекінгу, який допомагає системі не просто помітити об’єкт, а стабільно утримувати його від кадру до кадру, навіть якщо він частково пропадає з поля зору.

– Фільтр Калмана – математичний інструмент, який згладжує траєкторію руху та фільтрує похибки при оцінюванні відстані до об’єктів, прибираючи стрибки даних.

Така комбінація алгоритмів дозволяє компенсувати недоліки оптичних сенсорів і забезпечує плавну та точну роботу системи.

1 КФС ВИЯВЛЕННЯ ОБ'ЄКТІВ В УМОВАХ НЕДОСТАТНЬОЇ ВИДИМОСТІ ТА ПОСТАНОВКА ЗАДАЧІ ЩОДО ЇЇ УДОСКОНАЛЕННЯ

1.1 Дослідження архітектурних та операційних характеристик

Кіберфізичні системи (КФС) інтегрують обчислювальні процеси безпосередньо у фізичні сутності та процеси реального світу. На відміну від класичних автоматизованих комплексів, де керуюча логіка функціонує ізольовано від виконавчих механізмів, у КФС програмні алгоритми та фізичне середовище утворюють єдиний функціональний контур [5, 8]. Розвиток цієї думки став каталізатором для еволюції транспорту. Розроблені системи трансформують автомобіль на мобільний обчислювальний вузол. Такий вузол не просто фоново фіксує метрику руху, а безперервно аналізує дорожню обстановку, генеруючи відповідні управляючі чи інформаційні сигнали для водія, у відповідь на виявлені загрози [24, 29, 37].

За функціональною специфікою та рівнем архітектурного втручання в контур керування, підсистеми ADAS поділяють на два базові класи. До першого класу входять інформаційно-попереджувальні структури. Їхнє завдання полягає в оптичному та інформаційному розширенні сприйняття водія (наприклад, інфрачервоні системи нічного бачення), що дозволяють фіксувати приховані загрози без прямої дії на виконавчі механізми. Другий клас охоплює активні виконавчі підсистеми [15]. Вони орієнтовані на пряме втручання в рульове чи гальмівне керування автомобіля для примусового запобігання аварійних ситуацій.

Особливу інженерну складність становить проектування КФС, що функціонують у несприятливих метеоумовах або за критичного дефіциту освітлення (туман, злива, нічний час). Алгоритмічна база системи має компенсувати фізіологічні обмеження людського ока, гарантуючи високу точність детекції дорожніх об'єктів. Традиційні мультисенсорні платформи вирішують цю задачу за допомогою лідарів чи радарів, проте це суттєво підіймає цінник на фінальний продукт [9].

У запропонованій роботі базовим джерелом даних обрано монокулярний відеопотік із цифрової камерою. Економічна доступність такого підходу компенсується застосуванням методів глибокого навчання, які забезпечують необхідну інформативність та глибину аналізу сцени [23, 37].

Апаратна архітектура розробленої КФС базується на збалансованому поєднанні обчислювальних та сенсорних компонентів. До складу платформи входять такі елементи:

– центральний процесорний модуль для координації міжсистемної взаємодії;

– оптико-електронний сенсор (цифрова відеокамера високої роздільної здатності);

– графічний прискорювач для апаратної підтримки нейромережевої обробки кадру;

– вимірювальний блок (IMU) для фіксації динаміки транспортного засобу під час руху; [26]

– високошвидкісна оперативна пам’ять;

– додаткове спорядження для обміну інформацією.

Конструктивно розроблений пристрій скомпоновано у вигляді завершеного моноблока, що містить модуль живлення, пристрій цифрового захоплення відеосигналу, обчислювальне ядро з апаратним прискоренням ШІ-алгоритмів, підсистему генерації попереджень та спеціалізовану монтажну платформу для надійного кріплення всередині салону автомобіля [12, 18, 40].

Функціональний потенціал розробленої кіберфізичної системи ADAS зосереджений на аналізі динамічної дорожньої сцени в умовах реального часу. Реалізована архітектура забезпечує не просто фрагментарну фіксацію завад, а наскрізний моніторинг вхідного відеопотоку, що дозволяє мінімізувати затримки обробки даних та гарантує своєчасну локалізацію потенційно небезпечних об’єктів разом із точним прогнозуванням їхніх просторових переміщень. Завдяки оптимізації програмних компонентів під обмежені ресурси цільової вбудованої

платформи, система підтримує баланс між швидкістю роботи нейромережі та точністю обчислень, що критично для уникнення аварійних ситуацій [1]. Повний розподіл функцій системи, алгоритмічних рішень та відповідних кінцевих результатів, які формують практичну цінність розробки, деталізовано у табл. 1.1.

Таблиця 1.1 – Функції та завдання (ADAS)

№	Функція	Завдання
1.	Візуальний моніторинг дорожньої обстановки	Безперервна реєстрація дорожньої сцени; відстеження статичних і динамічних об’єктів; інформаційний супровід водія; збір даних для аналізу аварійних ситуацій
2.	Виявлення та класифікація об’єктів	Розпізнавання категорій перешкод (транспортні засоби, пішоходи, велосипедисти) у реальному часі; передача координат до блоку прийняття рішень
3.	Інформування та сповіщення	Виведення попереджень на панель приладів; формування звукових і візуальних сигналів при критичному зближенні
4.	Покращення зображення в умовах низької освітленості	Програмна корекція відеопотоку методами CLAHE та гамма–корекції; виявлення об’єктів поза зоною дії штатного освітлення
5.	Аналіз метеорологічних умов	Адаптація параметрів алгоритмів виявлення відповідно до поточних умов освітленості та прозорості атмосфери
6.	Прогнозування траєкторій та оцінка ризику	Обчислення часу до можливого зіткнення (TTC); визначення ймовірних векторів руху виявлених об’єктів

1.2 Аналіз комплексів обробки даних у КФС підтримки водія

Вектор розвитку систем автономного водіння та активної допомоги водієві, яскравим прикладом яких є концепція Pure Vision від компанії Tesla [4, 12], полягає в майже повній відмові від радарів чи лідарів на користь виключно оптичних камер. За такого підходу завдання реконструкції тривимірного простору навколо автомобіля повністю перекладається на програмні алгоритми комп'ютерного зору [7]. Проте прагнення зробити камери єдиним джерелом інформації для автомобіля створює серйозні виклики для розробників, оскільки безпека руху стає залежною від здатності програмного комплексу адекватно сприймати картинку в умовах, які є далекими від ідеальних [19, 31].

Перші покоління автомобільних асистентів базувалися на класичних методах цифрової обробки зображень, де для пошуку машин чи пішоходів використовувалися жорстко прописані математичні дескриптори, такі як фільтри HOG чи каскади Хаара [15, 23]. Ці алгоритми були легкими з погляду обчислень і могли працювати на базових автомобільних мікроконтролерах, проте їм бракувало гнучкості. Вони демонстрували стабільну роботу вдень на сухій дорозі, але виявлялися абсолютно безпорадними з настанням сутінків або під час зливи [3]. Будь-яке падіння контрасту у кадрі або поява цифрового шуму на матриці, призводили до того, що класичні геометричні ознаки об'єктів розмивалися, викликаючи масові пропуски перешкод і критичні збої в логіці безпеки [1, 22].

Важливим кроком став перехід індустрії до машинного навчання, що розділило нейромережеві підходи на два частини [16]. Перша це важкі двоетапні архітектури типу Faster R-CNN, які спочатку сканують кадр на наявність потенційних зон інтересу, а вже потім детально розпізнають об'єкти в них [28]. Такі комплекси мають високу математичну точність і чудово справляються із зашумленою картинкою, але вони є надто повільними для динамічних дорожніх сцен [5]. Їхня висока обчислювальна складність унеможливлює обробку відеопотоку в реальному часі на автомобільних платформах, оскільки на високих

швидкостях, затримка навіть у кілька мілісекунд може призвести до аварійної ситуації [11, 34].

Саме тому в реальних системах ADAS більшу популярність здобули одноетапні архітектури, серед яких лідерами є моделі сімейства YOLO [2, 35]. Вони здатні аналізувати весь кадр за один прохід, забезпечуючи високу частоту оновлення даних, необхідну для миттєвого реагування автомобіля. Базові версії цих мереж мають інше недоліки: вони навчаються переважно на ідеальних денних датасетах, тому в умовах нічної їзди чи густого туману їхня точність стрімко падає [14, 38]. Нейромережа починає втрачати об'єкти з кадру на кілька секунд або помилково інтерпретувати відблиски світла на мокрому асфальті, як реальні перешкоди [27, 39].

Більшість наявних на ринку open-source рішень фокусуються виключно на вияві об'єктів в межах одного статичного кадру. Для реального автомобіля цього замало, оскільки відсутність наскрізної аналітики між кадрами призводить до візуального тремтіння розпізнаних меж та хаотичних стрибків розрахованої відстані до об’єктів [9, 26].

Усе це обґрунтовує необхідність проектування не просто ізольованої нейромережі, а складного багатошарового конвеєра обробки інформації [10]. Щоб компенсувати фізичні недоліки камер у темряві, вхідний відеопотік спочатку має проходити через алгоритми адаптивного покращення контрасту, такі як CLAHE, які витягують деталі з глибоких тіней [6, 18]. Далі очищена картинка передається швидкій моделі YOLOv8, робота якої підсилюється алгоритмом трекінгу ByteTrack [33] та математичним фільтром Калмана [8, 40]. Така система дозволяє не просто помічати об'єкти, а безперервно й плавно вести їх по траєкторії, згладжуючи похибки та точно прораховуючи дистанцію навіть у найскладніших метеоумовах. Саме через зданість компенсувати деструктивний вплив зовнішнього середовища на камеру, було обрано ці методи [6].

1.3 Аналіз вимог до системи та розробка технічного завдання

Для забезпечення надійної роботи алгоритмів розпізнавання в межах кіберфізичної системи необхідно детально розглянути фізичну природу процесів, що призводять до спотворення вхідних даних у темряві та під час негоди [13]. Основна проблема нічної зйомки полягає в сильному звуженні динамічного діапазону дорожньої сцени, коли в межах одного кадру одночасно присутні зони глибокої тіні та ділянки надмірної освітленості від фар зустрічного транспорту або ліхтарів [25]. Спроби матриці камери автоматично підлаштувати експозицію під темні ділянки призводять до ефекту «блумінгу» – оптичного пересвіту навколо джерел світла, що повністю випалює інформацію у суміжних пікселях [30]. А при фокусуванні на яскравих об'єктах, решта кадру залишається неінформативною, через що силуети пішоходів або контури транспортних засобів на узбіччі зливаються з фоном [6].

Додатковим дестабілізуючим чинником є різке зниження відношення сигналу до шуму, оскільки в умовах дефіциту фотонів сенсор змушений працювати на граничних значеннях світлочутливості [21]. Це неминуче викликає появу вираженого хроматичного та просторового цифрового шуму, який класичні нейромережі інтерпретують як високочастотні ознаки текстури, що призводить до хибної активації нейронів [36]. Ситуація суттєво ускладнюється під час опадів або туману, коли краплі води чи зважені в повітрі мікрочастинки виконують роль природних лінз, що розсіюють світло фар [17]. У такому середовищі відбувається пряме фізичне розмиття градієнтів яскравості, на аналізі яких базується робота більшості детекторів, що робить задачу локалізації меж об'єктів математично некоректною [32].

З огляду на зазначені фактори, до обчислювального конвеєра додається низка вимог, які неможливо задовольнити використанням стандартних архітектур з відкритим кодом [29]. Перша вимога стосується обов’язкової наявності фази просторово-часової стабілізації вхідного сигналу перед етапом детекції. Метод

попередньої обробки повинен адаптивно вирівнювати локальний констраст у темних зонах, не допускаючи при цьому збільшення рівня шуму або появи артефактів на межах об’єктів [2]. Друга вимога зумовлена специфікою роботи автомобільних платформ у реальному часі. Сумарна затримка обробки одного кадру всіма модулями ланцюга не повинна перевищувати крок між суміжними кадрами відеопотоку [14, 37].

Окремою групою вимог вважається забезпечення структурної стійкості процесу супроводу [24]. Оскільки в умовах обмеженої видимості нейромережа неминуче втрачатиме детекцію об'єктів на окремих кадрах через тимчасове перекриття або засліплення матриці, система повинна мати механізм збереження інформації про об'єкт у просторі [33].

Оцінка координат та вектора швидкості перешкод має базуватися на принципах статистичної фільтрації, що дозволить вирахувати траєкторію руху навіть за відсутності прямих вимірювань від оптичного сенсора [8]. Таким чином, формування вимог до архітектури КФС підтримки водія вимагає відмови від ізольованих аналітичних моделей на користь інтегрованого ланцюжка алгоритмів, де кожен наступний етап компенсує інформаційні втрати попереднього [20, 40].

1.4 Висновки до першого розділу

У першому розділі детально розглянуто принципи побудови систем допомоги водієві. Досліджено, як працюють класичні кіберфізичні комплекси, у яких апаратне забезпечення, оптичні сенсори та програмні алгоритми функціонують у зв’язці. Головну увагу на цьому етапі було зосереджено на пошуку недоліків, через які вся система безпеки може дати збій у реальних умовах.

Для чіткої структуризації матеріалу стосовно існуючих рішень, ADAS розділено на дві великі категорії: інформаційні, що лише попереджають водія, виводять підказки на екран чи сигналізують про небезпеку та активні, які безпосередньо втручаються в керування, наприклад, здійснюють автоматичне

гальмування чи утримання в смузі. Поточне дослідження належить до першої групи інформаційних систем і має на меті виступати в ролі надійного цифрового помічника, здатного вчасно помітити загрозу, яку водій може пропустити через втому або обмежену видимість.

Проведени огляд готових інструментів та бібліотек комп'ютерного зору показав суттєвий недолік. Більшість наявних рішень розроблялись і тестувались для сприятливих умов освітлення (вдень, під час сухої погоди). Проте в реальних нічних умовах, під час зливи чи в тумані, якість картинки з камери сильно погіршується. Через падіння контрастності кадру та появу цифрового шуму на матриці стандартні алгоритми детекції часто пропускають об'єкти або працюють не правильно. Це підтверджує необхідність обов'язкового застосування методів попередньої обробки відеопотоку перед аналізом кадрів за допомогою ШІ.

Сформульовано завдання роботи: підібрати такий комплекс алгоритмів обробки кадру і вияву, який стабільно працюватиме в умовах слабкого освітлення та негоди. Базою для вирішення цієї задачі обрано нейромережеві моделі глибокого навчання. Вибір нейромереж обґрунтовано тим, що вони набагато краще за класичні математичні фільтри виділяють основні ознаки об'єктів (контури, силуети автомобілів чи пішоходів). Це дозволяє суттєво знизити кількість помилок та мінімізувати хибні спрацювання.

У розділі систематизовано системи ADAS за їхніми функціями, рівнем автоматизації (згідно з міжнародними стандартами) та способом взаємодії з механічними вузлами автомобіля. Такий аналіз дозволив чітко визначити місце розробки серед існуючих технологій, окреслити її особливості та обґрунтувати практичну цінність запропонованого варіанту обробки відеопотоку.

2 ПРОЄКТУВАННЯ СИСТЕМИ ОБРОБКИ ВІДЕОІНФОРМАЦІЇ У КФС ІНТЕЛЕКТУАЛЬНОЇ ПІДТРИМКИ ВОДІЯ

2.1 Визначення апаратних і програмних підсистем системи ADAS

На основі аналізу, було роздроблено програмне забезпечення системи на п’ять підсистем (рис. 2.1):

- підсистема захоплення та попередньої обробки відеопотоку;
- підсистема нейромережевої детекції та трекінгу об’єктів;
- підсистема оцінювання відстані та прогнозування зіткнення;
- підсистема аналізу дорожньої смуги;
- підсистема формування та виведення попереджень.

Рисунок 2.1 – Структура програмного забезпечення

Такий поділ дає змогу розробляти і тестувати кожен модуль окремо, а потім зібрати їх у єдиний ланцюжок. Якщо в майбутньому захочеться замінити, наприклад, детектор смуги, то це можна зробити без переписування решти системи.

2.2 Покращення відеозображення в умовах низької освітленості

У темряві картинка з камери шумна і не контрастна, тому перед детекцією її потрібно покращити. Для цього в системі є окремий модуль NightVisionEnhancer, який послідовно виконує три кроки: шумозниження, гамма-корекцію і підвищення контрасту.

На етапі шумозниження до вхідного кадру застосовується гаусівське розмиття розміром 3x3. Це дозволяє нівелювати імпульсний та тепловий шум матриці камери без суттєвої втрати чіткості країв об'єктів.

Гамма-корекція є другим етапом і являє собою нелінійне перетворення яскравості пікселів, яке виконується за формулою (2.1):

$$I_{out} = \left(\frac{I_{in}}{255}\right)^{\frac{1}{\gamma}} \cdot 255, \qquad (2.1)$$

де I_{in} – вхідне значення яскравості пікселя;

I_{out} – вихідне значення;

γ – коефіцієнт корекції.

При значенні γ > 1 відбувається освітлення темних областей зображення без пересвітлення яскравих ділянок. У запропонованій архітектурі вибрано значення γ = 1.7, що забезпечує баланс між деталізацією затінених ділянок та природним відображенням кольорів. Щоб досягти максимальної швидкості системи, перетворення реалізовано через попередньо обчислену таблицю відповідностей (LUT – Look-Up Table) на 256 елементів. Такий підхід дозволяє повністю уникнути ресурсомісткого розрахунку степеневої функції для кожного окремого пікселя.

Наступний кроком буде алгоритм CLAHE (Contrast Limited Adaptive Histogram Equalization). На відміну від звичайного вирівнювання гістограми, він працює не з усім кадром одразу, а з участком 8×8 пікселів. Глобальне вирівнювання їх упосередкує, а CLAHE підтягне кожну зону окремо. Для зниження

обчислювальних витрат усі маніпуляції виконуються в кольоровому просторі YUV, де алгоритм CLAHE діє лише на канал Y (яскравість), залишаючи хроматичні компоненти (U та V) без змін. Послідовність етапів описуваного ланцюжка обробки інформації з камери наведена на рис. 2.2.

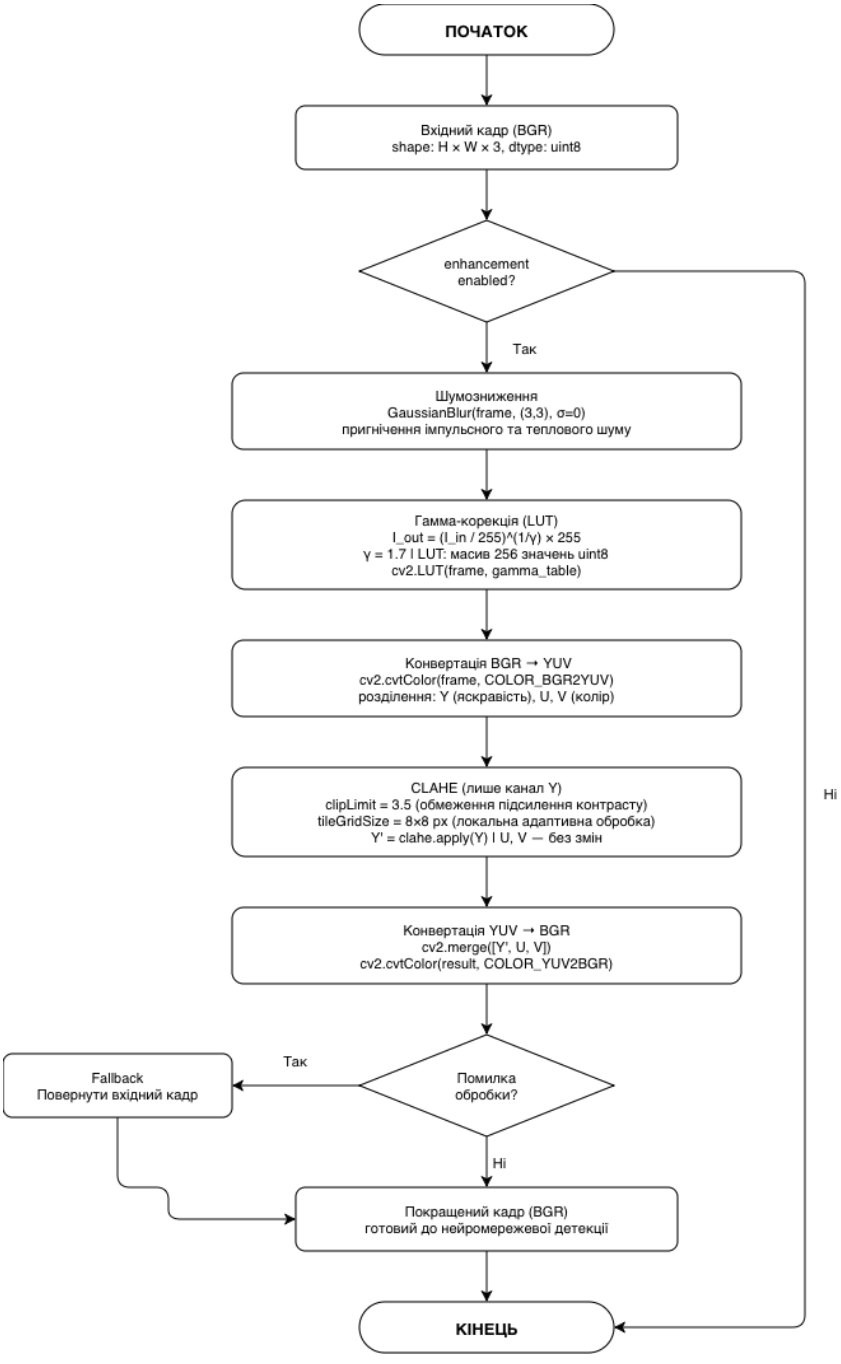


Рисунок 2.2 – Блок-схема модуля покращення відеозображення

Параметри модуля винесено до централізованої конфігурації AdasConfig що забезпечує можливість їх швидкого коригування без втручання у логіку обробки. Перелік параметрів наведено у табл. 2.1.

Таблиця 2.1 – Параметри модуля NightVisionEnhancer

Параметр	Значення	Призначення
clahe_clip_limit	3.5	Обмеження підсилення контрасту
clahe_tile_size	8x8	Розмір локальної області обробки
gamma_correction	1.7	Коефіцієнт освітлення темних зон
saturation_boost	1.2	Підсилення насиченості кольорів
Розмір ядра розмиття	3x3	Ступінь шумозниження

2.3 Нейромережева детекція та трекінг об’єктів (YOLOv8 + ByteTrack)

В основі системи лежить нейромережа YOLOv8 разом із трекером ByteTrack. Модель YOLOv8 належить до одноетапних детекторів, бо вона виконує один прохід через кадр, знаходить об’єкти і класифікує їх. Двоетапні архітектури типу Faster R CNN спочатку шукають кандидатів, а потім їх класифікують. Ця процедура є точнішою, але більш повільною. Для системи, що працює в реальному часі важливий баланс, тому одноетапний детектор тут краще себе покаже.

У проекті використовується варіант моделі yolov8s.pt (small). Ця підмодель має прийнятну швидкість з хорошою точністю. На платформі з прискоренням MPS усе працює плавно та без затримок.

Архітектурно YOLOv8 ділиться на три блоки: backbone (CSPDarknet), що витягує первинні ознаки з кадру, neck (PANet) збирає їх із різних масштабів, а detection head видає координати рамок і ймовірності класів. Модель навчена на датасеті COCO з 80 класами, але у цій роботі використовується лише шість учасників дорожнього руху, з якими реально може зіткнутися автомобіль в змодельованих дорожніх умовах (табл. 2.2).

Таблиця 2.2 – Класи об’єктів COCO, що використовуються в системі ADAS

ID класу	Назва	Категорія	Реальна ширина (м)
0	person	PEDESTRIAN	0.45
1	bicycle	TWO_WHEELER	0.60
2	car	VEHICLE	1.80
3	motorcycle	TWO_WHEELER	0.80
5	bus	VEHICLE	2.55
7	truck	VEHICLE	2.50

Модель викликається в режимі track, що задіює вбудовану інтеграцію з трекером. Параметри, що використовуються при запуску наведені в табл. 2.3.

Таблиця 2.3 – Параметри конфігурації нейромережевої детекції

Параметр	Значення	Призначення
conf	0.45	Мінімальний поріг достовірності детекції

Кінець таблиці 2.3

iou	0.50	Поріг перекриття для немаксимального придушення (NMS)
imgsz	640	Розмір вхідного зображення для інференсу (пікселів)
half	True	Інференс у форматі FP16 для прискорення на GPU/MPS
persist	True	Збереження стану трекера між кадрами

Для вирішення задачі багатооб'єктного трекінгу (MOT або Multi–Object Tracking) у структурі розроблюваного програмного забезпечення інтегровано алгоритм ByteTrack. Його відмінність від класичних аналогів полягає в специфічній стратегії обробки детекцій із низьким рівнем достовірності. Традиційні методи повністю відкидають подібні появи, ще на етапі фільтрації порогом confidence threshold. ByteTrack не ігнорує слабкі фонові сигнали нейромережі, а розділяє масив вихідних обмежувальних рамок на два потоки. Масив детекцій з низьким рівнем довіри і уже наявними траєкторіями, реалізується на другому етапі обчислювального зіставлення за допомогою обчислення матриці перетину геометричних меж. Таке рішення дозволяє суттєво знизити частоту втрат та штучної зміни ідентифікаторів у моменти перекриття об'єктів іншими машинами, а також при короткочасному розмитті кадру чи появі оптичних завад.

Кожному виявленому об'єкту дорожньої сцени присвоюється унікальний track_id, який безперервно утримується системою протягом усієї послідовності кадрів. Це дозволяє лінійному фільтру Калмана (розглянутому у підрозділі 2.5) стабільно накопичувати часову статистику та вектори станів по кожній цілі окремо, навіть якщо її візуальний контакт на кілька кадрів погіршився. На основі цих прогнозних метрик алгоритм формує зглажені та стійкі до стрибків оцінки

фактичної відстані до об’єкта і найбільш правильно розраховує динаміку швидкості зближення. Загальна структурна блок-схема детекції та супроводу наведена на рис. 2.3.

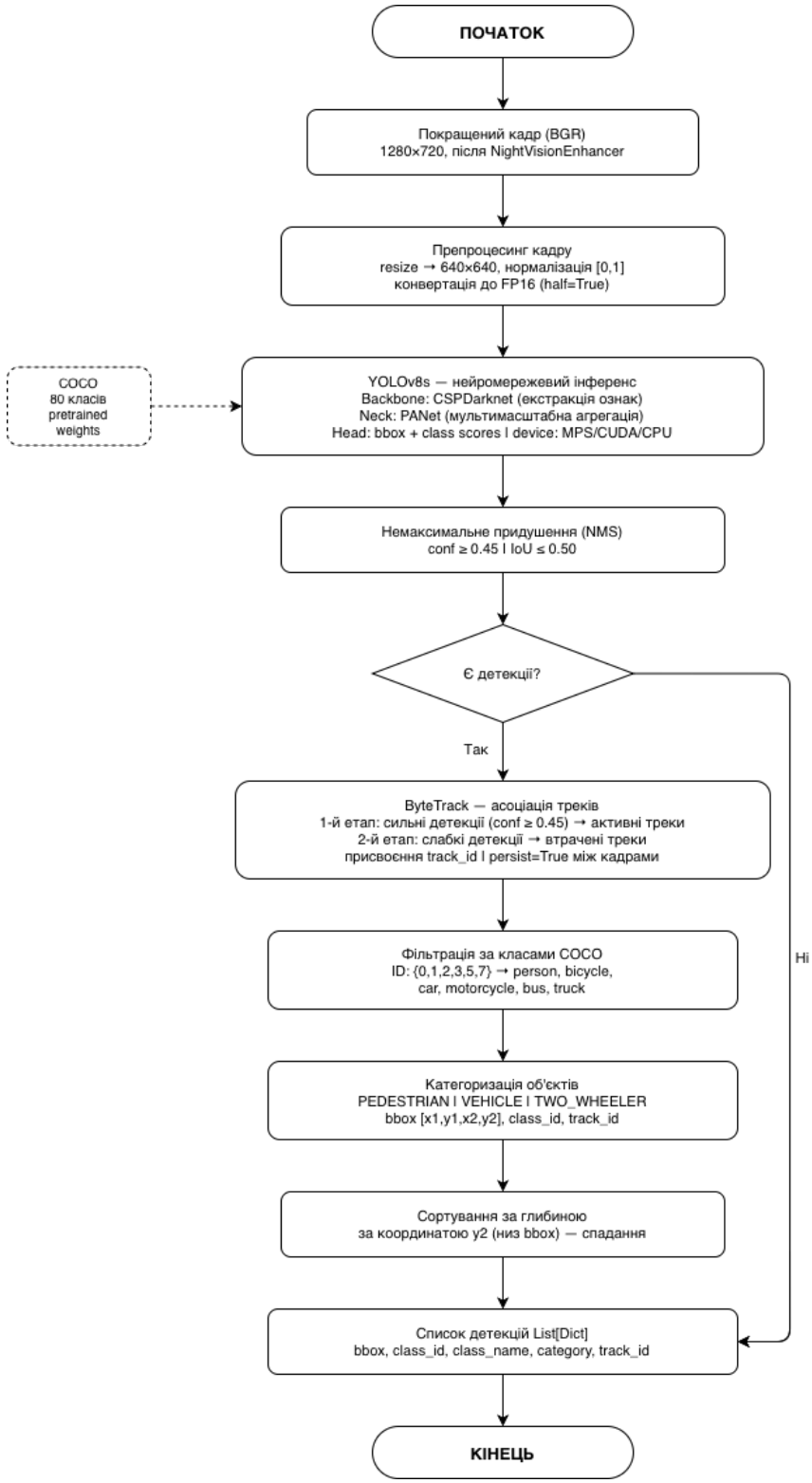


Рисунок 2.3 – Блок-схема підсистеми нейромережевої детекції та трекінгу

2.4 Оцінка відстані до об’єктів на основі пінхол моделі камери

Розрахунок дистанції до локалізованих об’єктів у межах розробленої системи покладено на метод монокулярної оцінки відстані, що базується на геометрії пінхол-моделі камери. На відміну від стереоскопічних чи лідарних комплексів, цей підхід не потребує апаратного ускладнення архітектури і повністю реалізується програмним шляхом на основі аналізу єдиного відеопотоку.

Пінхол-модель описує геометричне відображення тривимірної точки реального простору на двовимірну площину матриці через оптичний центр (точку проєкції). Відповідно до закономірностей цієї моделі, аналітичний зв’язок між фізичними розмірами обє’кта та геометричними параметрами його проєкції на сенсорі визначається за формулою (2.2):

$$d = \frac{W_{real} \cdot f}{W_{pixel}}, \qquad (2.2)$$

де d – відстань до об’єкта (м);

W_{real} – відома реальна ширина об’єкта (м);

f – фокусна відстань камери у пікселях;

W_{pixel} – ширина обмежувальної рамки об’єкта на зображенні.

У архітектурі проекту зафіксовано значення фокусної відстані f = 850 пікселів, що відповідає параметрам калібрування матриці типового автомобільного відеореєстратора з роздільною здатністю 1280 на 720 пікселів. Реальні габарити об’єктів (їхня ширина) внесені до констант системи на основі стандартних метрик транспортних засобів та інших учасників руху (табл. 2.2). Логіка розробленого класу DistanceEstimator обчислює поточну ширину обмежувальної рамки, як різницю горизонтальних координат отриманих із вихідного тензора нейромережі.

Для блокування хибних оцінок у сценаріях із частковим перекриттям або при надмірному віддалені цілей, введено поріг мінімальної ширини bounding box = 15

пікселів. Детекції з меншими геометричними параметрами автоматично виключаються з обчислювального контуру. У підсумку, результат розрахунку метричної відстані обмежується робочим діапазоном від 1.0 до 120.0 метрів, що повністю перекриває практичну зону моніторингу проектованої системи ADAS.

Монокулярна оцінка відстані на основі цієї моделі має ряд обмежень, які необхідно враховувати при аналізі результатів. Точність методу суттєво залежить від правильності значень реальних розмірів об’єктів. Мікроавтомобілі або великогабаритні вантажівки призводять до систематичної похибки оцінки. Також метод передбачає перпендикулярне розташування передньої площини об’єкта відносно оптичної осі камери, що не завжди виконується для об’єктів під кутом. Для компенсації зазначених похибок у системі застосовується фільтр Калмана, описаний у підрозділі 2.5, який зглажує нестаціонарні оцінки відстані та формує стабільний часовий ряд вимірювань.

Таблиця 2.4 – Джерела похибок оцінки відстані та методи компенсації

Джерело похибки	Вплив	Метод компенсації
Відхилення габаритів об’єкта від еталонних	Систематична похибка 15–25%	Апріорна таблиця розмірів за класами
Нахил об’єкта відносно камери	Заниження оцінки ширини	Фільтр Калмана (згладжування)
Часткове перекриття об’єкта	Хибно занижена відстань	Поріг мінімальної ширини 15 px
Шум детекції bbox	Стрибкоподібні оцінки	Фільтр Калмана (стабілізація)

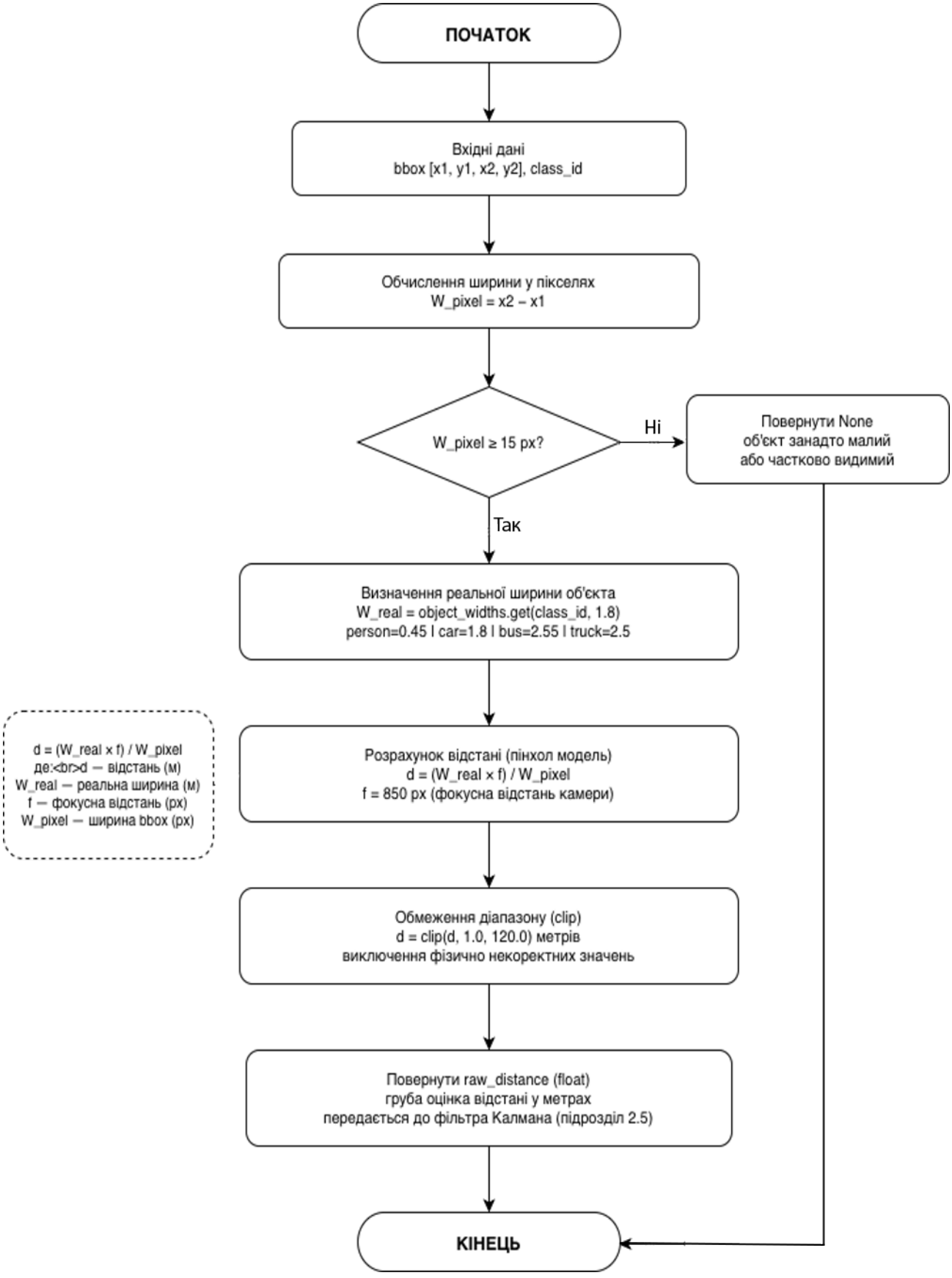


Рисунок 2.4 – Блок-схема модуля оцінки відстані (DistanceEstimator)

2.5 Фільтр Калмана для оцінки відстані та розрахунку швидкості

Первинна оцінка відстані, обчислена за допомогою геометричних пропорцій пінхол-моделі, характеризується помітним рівнем вимірювального шуму. Його поява зумовлена коливаннями меж обмежувальних рамок нейромережі, раптовими змінами освітленості дорожньої сцени та частковими перекриттями цілей іншими транспортними засобами. З метою фільтрації цих похибок та формування стабільного часового ряду метричних даних у системі реалізовано лінійний фільтр Калмана у межах програмного класу KalmanDistanceFilter. Стан системи описується двовимірним вектором (2.3):

$$x = \begin{bmatrix} d \\ v \end{bmatrix}, \tag{2.3}$$

де d – відфільтрована відстань до об’єкта (м);
v – швидкість зближення (м/с), позитивне значення якої відповідає наближенню об’єкта.

Початковий стан ініціалізується значеннями d_0 = 50.0 м, v_0 = 0.0 м/с. Матриця переходу стану описує (2.4) фізичну модель руху об’єкта між двома послідовними кадрами з часовим інтервалом Δt:

$$F = \begin{bmatrix} 1 & -\Delta t \\ 0 & 0 \end{bmatrix}, \tag{2.4}$$

де Δt – часовий інтервал між двома послідовними кадрами відеопотоку;
F – матриця переходу стану системи.

Від’ємний знак при Δt відображає зменшення відстані при позитивній швидкості зближення. Значення Δt обмежується діапазоном від 0.016 до 0.2 секунд для запобігання чисельній нестабільності при пропусках кадрів. Матриця

спостереження визначає, що безпосередньо вимірюється. Шум процесу Q та шум вимірювання R налаштовуються через конфігурацію системи (2.5):

$$Q = I_2 \cdot q, \quad q = 0.05, \quad R = [r], \quad r = 0.3, \tag{2.5}$$

де Q – коваріаційна матриця шуму процесу;

I_2 – одинична матриця другого порядку розміром 2x2;

q – скалярний коефіцієнт інтенсивності шуму процесу;

R – коваріаційна матриця шуму вимірювань;

r – дисперсія шуму вимірювання (ступінь довіри до даних детекції).

Менше значення q забезпечує більш інертну (стабільну) поведінку фільтра, тоді як менше r підвищує довіру до вимірювань. Цикл фільтрації складається з двох стандартних етапів. Етап передбачення можна описати формулою (2.6):

$$x = Fx \quad P = FP\mathrm{F}^{\mathsf{T}} + Q \cdot \Delta t, \tag{2.6}$$

де x – вектор стану системи;

F – матриця переходу стану;

P – коваріаційна матриця помилок оцінювання, яка відображає ступінь невизначеності або точності поточних оцінок вектора стану;

Q – коваріаційна матриця шуму процесу;

Δt – часовий інтервал між двома послідовними кадрами.

Після стабілізації фільтра (не менше восьми вимірювань) на основі відфільтрованих значень обчислюється також час до можливого зіткнення (2.7):

$$TTC = \frac{d}{v}, \tag{2.7}$$

де d – відстань до рухомого об’єкта;

v – швидкість автомобіля.

Значення TTC обмежується діапазоном від 0 до 20 секунд, а значення поза цим діапазоном вважаються не правильними і виключаються. TTC використовується системою попереджень, як пріоритетний індикатор небезпеки, оскільки враховує не лише поточну відстань, а й динаміку зближення. Параметри цього фільтра наведено у табл. 2.5.

Таблиця 2.5 – Параметри фільтра Калмана

Параметр	Значення	Призначення
Вектор стану	$[d, v]$	Відстань та швидкість зближення
q (шум процесу)	0.05	Інертність фільтра
r (шум вимірювання)	0.30	Довіра до вимірювань
Поріг стабілізації	8 вимірювань	Мінімум для розрахунку TTC
v_{min}	1.0 м/с	Мінімальна швидкість для TTC
Діапазон d	0.5 – 150.0 м	Обмеження стану
Діапазон v	-30.0 – 30.0 м/с	Обмеження швидкості

2.6 Детектор смуги (LaneDetector)

Прорахунок положення виявленого об’єкта відносно поточної смуги руху є базовою умовою для зменшення хибних спрацювань системи попередження. Перешкоди, розташовані за межами траєкторії руху автомобіля, не можуть створювати загрози зіткнення, тому алгоритм не повинен генерувати попереджувальні сигнали.

Для розв’язання цієї задачі у проекті розроблено модуль динамічного відстеження меж смуги, виконаний у класі LaneDetector.

Послідовна обробка вхідних кадрів складається з декількох етапів. Спочатку звичайне зображення перетворюється у напівнасичене і згладжується за допомогою фільтра Гауса з силою дії 7x7 пікселів. Це дозволяє нівелювати дрібнозернисті шуми й шорсткість асфальтного покриття, зберігши при цьому чіткість кольорів на межах ліній розмітки. Наступним кроком є виділення контурів детектором Кенні з пороговими значеннями фільтрації 50 та 150, який фіксує інтенсивні перепади яскравості у просторових координатах кадру.

З метою відсікання помилок, визваних сторонніми об’єктами, що розташовані поза межами дорожнього полотна (будівель, рекламних щитів, хмар), до отриманого бінарного контурного зображення застосовується маска у формі трапеції (2.8). Вона обмежує зону аналізу нижньою центральною частиною кадру, де розташована проїжджа частина:

$$ROI = \{(0.25w, h), (0.46w, 0.65h), (0.54w, 0.65h), (0.75w, h)\}, \qquad (2.8)$$

де w – ширина кадру;

h – висота кадру;

ROI – відрізок кадру, в якому відбувається розрахунок TTC.

Така геометрія маски відповідає перспективній проекції прямої дороги і відсікає більшість сторонніх об’єктів. У межах *ROI* застосовується перетворення Хафа з параметрами мінімальної довжини лінії, що дорівнює 50 пікселів і максимального розриву між сегментами 100 пікселів. Виявлені відрізки фільтруються за кутом нахилу. Для лівої межі відбираються лінії з нахилом $-1.5 < k < -0.4$, а у лівій половині кадру і $0.4 < k < 1.5$ у правій половині. Це дозволяє відкинути горизонтальні лінії пішохідних переходів та вертикальні артефакти. Для кожної групи ліній методом найменших квадратів будуються лінії, що дозволяють отримати єдину усереднену лінію розмітки. На їх основі формується чотирикутний

полігон смуги руху, координати якого обмежуються фізично допустимими межами для запобігання розповзанню смуги.

Для забезпечення стабільності відображення між послідовними кадрами застосовується згладжування полігону з коефіцієнтом α = 0.03. Мале значення α робить полігон смуги дуже інертним, оскільки він повільно адаптується до змін геометрії дороги. Для перевірки того, чи знаходиться виявлений об'єкт у межах смуги рух, використовується нижня центральна точка його обмежувальної рамки або точка контакту з дорогою (2.9):

$$P_{contact} = \left(\frac{x_1 + x_2}{2}, y_2\right) \quad (2.9)$$

де $P_{contact}$ – координати центральної точки обмежувальної рамки;

x_1, x_2 – горизонтальні координати лівої та правої рамки відповідно;

y_2 – вертикальна координата рамки.

Належність точки до полігону визначається функцією pointPolygonTest. Якщо результат додатній, то об'єкт знаходиться у смузі руху і підлягає оцінюванню рівня загрози. У протилежному випадку об'єкту присвоюється статус OUT_OF_LANE і він прибирається з розрахунку TTC.

Таблиця 2.6 – Параметри модуля LaneDetector

Параметр	Значення	Призначення
Ядро розмиття	7×7	Пригнічення дрібного шуму
Коефіцієнт згладжування α	0.03	Стабілізація полігону між кадрами

Кінець таблиці 2.6

Горизонт ROI	0.65h	Нижня межа зони детектування
Пороги Кенні	50 / 150	Виділення країв розмітки
Мін. довжина лінії Хафа	50 px	Відсів коротких артефактів
Макс. розрив Хафа	100 px	З’єднання переривчастої розмітки
Діапазон нахилу (ліва)	від −1.5 до −0.4	Фільтр кута лівої межі
Діапазон нахилу (права)	від 0.4 до 1.5	Фільтр кута правої межі

2.7 Система попереджень та рівні загрози (WarningLevel)

Підсистема формування попереджень є завершальним рівнем розробленої кіберфізичної системи ADAS, що включає в себе результати роботи всіх попередніх модулів і перетворює їх на структуровані сигнали тривоги для водія. Реалізація підсистеми зосереджена у класах CollisionWarningSystem, BrakeVisualizer та переліку WarningLevel. Для ступеня небезпечності дорожньої ситуації у системі визначено шестирівневу ієрархію попереджень, реалізовану через перелік WarningLevel. Кожен рівень відповідає певному діапазону відстані та значенням часу до зіткнення (TTC), а також визначає характер відображення об’єкта на екрані та інтенсивність роботи індикатора гальмування. Перелік рівнів із їхніми визначеннями наведено у табл. 2.7.

Таблиця 2.7 – Ієрархія рівнів попереджень системи ADAS

Рівень	Назва	Колір (RGB)	Визначення
1	SAFE	Зелений (0, 180, 0)	Об’єкт у смузі, загроза відсутня
2	CAUTION	Блакитний (0, 220, 220)	Рекомендується зниження швидкості
3	WARNING	Помаранчевий (0, 140, 255)	Потенційно небезпечне зближення
4	CRITICAL	Червоний (0, 0, 255)	Критична загроза, необхідне гальмування
5	OUT_OF_LANE	Пурпуровий (180, 0, 180)	Об’єкт поза смугою руху

Статус OUT_OF_LANE має особливе значення. Система автоматично присвоює його, якщо точка контакту об’єкта з асфальтом виходить за межі розрахованого полігона смуги. У цьому випадку відстань і показник TTC не враховуються. Алгоритм ігнорує такі об’єкти під час розрахунку гальмування, тому вони не впливають на загальний статус системи. Це допомагає уникати помилкових тривог від автомобілів на зустрічній смузі або пішоходів на тротуарі. Визначення загрози виконує метод get_warning_level у класі CollisionWarningSystem. Для розрахунку йому потрібні три параметри: відфільтрована відстань d, час до зіткнення і тип об’єкта.

У конфігурації для пішоходів і автомобілів встановлені різні пороги, оскільки їхня кінематика і рівень небезпеки відрізняються. Пішоходи більш вразливі при зіткненні, а їхня траєкторія часто непередбачувана, тому для них критерії суворіші. Для автомобілів пороги нижчі, що підходять для міських заторів, де мала дистанція є нормальною, і часті тривоги лише заважали б водієві. Основним архітектурним рішенням було надати пріоритет TTC, а не статичній відстані. Наприклад,

зустрічна машина за 20 метрів, яка рухається зі швидкістю 15 м/с, набагато небезпечніша, ніж нерухома перешкода за 3 метри попереду.

Перевірка відбувається у два етапи. Спочатку алгоритм аналізує TTC, якщо фільтр Калмана дає стабільні дані. Якщо таких даних немає або значення виходить за критичні межі, система переходить до статичних порогів відстані. Така ієрархія дозволяє вчасно реагувати на динамічні загрози, але не пропускати нерухомі перешкоди. Окрім маркування рамок на екрані, у проекті є аналоговий індикатор інтенсивності гальмування, реалізований через клас BrakeVisualizer. Його завдання це показати водієві просту шкалу, яка заповнюється градієнтом від жовтого до червоного залежно від рівня небезпеки. Фінальна інтенсивність гальмування визначається поточним рівнем загрози та значенням TTC. Додатково, при наявності значення TTC < 3.0 с, інтенсивність підсилюється. Для забезпечення плавної візуальної реакції індикатора без різких стрибків застосовується поступове згладжування з коефіцієнтом β (2.10).

$$I_t = I_{t-1} + \left(I_{target} - I_{t-1}\right) \cdot \beta \qquad (2.10)$$

де I_t – поточне значення інтенсивності гальмування;

I_{t-1} – попереднє значення інтенсивності гальмування;

I_{target} – цільове значення інтенсивності гальмування;

β – коефіцієнт зглажування.

При досягненні інтенсивності понад 0.5 у верхній частині екрану з’являється напівпрозора червона смуга попередження. При перевищенні порогу 0.8 активується текстове повідомлення EMERGENCY BRAKE, а при значенні понад 0.7 появляється пульсуюча червона рамка екрану з частотою 8 Гц. Така візуальна сигналізація забезпечує надійне привернення уваги водія навіть в умовах нічного керування при зниженій концентрації. Для запобігання накопиченню застарілих даних у пам’яті системи, реалізовано механізм автоматичного очищення фільтрів Калмана для об’єктів, що більше не відстежуються. Метод cleanup класу

CollisionWarningSystem видаляє записи про об’єкти, які не з’являлися у відеопотоці протягом певного часу. Очищення виконується на кожному кадрі з використанням часу реєстратора, що забезпечує правильну поведінку системи. Це дозволяє динамічно адаптуватися до кількості об’єктів у сцені без збільшеного споживання пам’яті з плином часу.

2.8 Висновки до другого розділу

У другому розділі було спроектовано програмну архітектуру системи. Від модуля попередньої обробки кадру до підсистеми попереджень. Проект роздріблений на п’ять незалежних модулів, які зв’язані між собою через стандартні інтерфейси передачі даних. Це дозволяє розробляти й тестувати кожен модуль окремо, і за потреби міняти їх без переписування решти.

Модуль покращення зображення NightVisionEnhancer виконує три кроки: гаусівське розмиття, гамма-корекцію через LUT-таблицю, обробку CLAHE у просторі YUV. Параметри $\gamma = 1.7$ і clipLimit = 3.5 були підібрані експериментально, вони дають баланс між освітленням темних ділянок і підсиленням шуму. Обробка тільки Y каналу у три рази скорочує обчислення, що критично для проектів, що працюють в реальному часі.

Для вияву об’єктів було обрано YOLOv8 у варіанті small, що є компромісом між точністю і швидкістю. З 80 класів COCO використовується 6 (пішоходи, велосипедисти, мотоциклісти, легкові машини, автобуси, вантажівки). Трекінг забезпечує ByteTrack, який використовує і сильні, і слабкі детекції, що знижує кількість втрат ID при перекриттях і тимчасових погіршеннях якості кадру.

Відстань до об’єктів оцінюється монокулярно за пінхол-моделлю камери. За допомогою реального розміру об’єкта, фокусної відстані і ширини рамки на кадрі. Метод простий і не потребує додаткових сенсорів, але дає похибку при відхиленні реальних габаритів від табличних. Для згладжування цих похибок і отримання швидкості зближення реалізовано лінійний фільтр Калмана. На основі його

результатів обчислюється TTC, який потім стає основним показником динамічної небезпеки в підсистемі попереджень.

Детектор смуги LaneDetector побудований на класичних алгоритмах комп'ютерного зору – детекторі Кенні, маски ROI і перетворенні Хафа. Він визначає, чи виявлений об'єкт у моїй смузі, через геометричну перевірку нижньої центральної точки рамки. Без цього модуля система реагувала б на машини на сусідніх смугах і пішоходів на тротуарі.

Підсистема попереджень має шість рівнів небезпеки, з різними порогами для пішоходів і автомобілів. Логіка прийняття рішень побудована на пріоритеті TTC над статичними порогами відстані. Це дозволяє реагувати на ситуації, коли зустрічний транспорт рухається по вашій смузі або стався раптовий виїзд автомобіля з прилеглої дороги.

Індикатор гальмування з поступовим згладжуванням і три рівні візуального попередження (червона смуга, пульсуюча рамка, текст EMERGENCY BRAKE) забезпечують надійне привернення уваги в нічних умовах. Усі п'ять модулів утворюють робочу архітектуру, готову до програмної реалізації.

3 РЕАЛІЗАЦІЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ КФС ADAS В УМОВАХ ОБМЕЖЕНОЇ ВИДИМОСТІ

3.1 Апаратна платформа та середовище розгортання системи

Розроблена кіберфізична система ADAS розгорнута та протестована на апаратній платформі Apple MacBook Pro з процесором M1 Pro, редставником архітектури ARM з складною обчислювальною структурою. Вибір даної платформи обумовлений її особливостями, що роблять її показовою для вбудованих обчислювальних систем, орієнтованих на задачі штучного інтелекту. Процесор M1 Pro є системою на кристалі, що об'єднує на єдиному напівпровідниковому кристалі центральний процесор, графічний процесор, блок нейронних обчислень та підсистему уніфікованої пам'яті.

Відмінність архітектури UMA від традиційних систем із роздільною пам'яттю CPU та GPU полягає у тому, що всі обчислювальні блоки мають прямий доступ до єдиного блоку оперативної пам'яті без необхідності явного копіювання даних між хост-пам'яттю та пам'яттю графічного прискорювача. Це суттєво знижує затримки при передачі даних між CPU та GPU, що є критично важливим для систем реального часу з інтенсивним обміном даними між нейромережею та модулями постобробки. Технічні характеристики використаної апаратної платформи наведено у табл. 3.1.

Таблиця 3.1 – Технічні характеристики апаратної платформи

Характеристика	Значення
Процесор	Apple M1 Pro
Графічний процесор	Apple GPU (16 ядер)

Кінець таблиці 3.1

Уніфікована пам’ять	16 ГБ LPDDR5 (UMA)
Neural Engine	16 ядер, до 11 TOPS
Операційна система	macOS Tahoe 26.2
Пропускна здатність пам’яті	200 ГБ/с
Архітектура CPU	ARM (10 ядер: 8 продуктивних + 2 енергоефективних)

Підтримка Metal Performance Shaders є важливим аспектом для виконання нейромережевих задач на графічному процесорі Apple M1 Pro. У фреймворку PyTorch використовується низькорівневий бекенд MPS, призначений для виконання паралельних обчислень на GPU через фреймворк Metal. На відміну від CUDA, яка є спеціалізованою технологією компанії NVIDIA та працює лише з відповідними графічними прискорювачами, MPS забезпечує можливість апаратного прискорення на пристроях із графічними процесорами Apple Silicon. Використання MPS дозволяє суттєво підвищити продуктивність виконання моделей глибокого навчання порівняно з використанням центрального процесора. Завдяки цьому забезпечується швидша обробка відеопотоку в режимі реального часу, що є критично важливим для функціонування системи ADAS.

Паралельна обробка великої кількості операцій на GPU дає можливість ефективніше виконувати задачі вияву об’єктів, трекінгу та аналізу дорожньої сцени без значних затримок. У системі реалізовано механізм автоматичного вибору обчислювального пристрою залежно від доступної апаратної конфігурації. Спочатку виконується перевірка підтримки MPS для використання графічного

процесора Apple. Якщо така підтримка недоступна, система перевіряє наявність CUDA-сумісного GPU. У випадку відсутності обох варіантів обчислення виконуються на центральному процесорі. Такий підхід забезпечує універсальність програмного забезпечення та можливість його запуску на різних апаратних платформах без необхідності ручного налаштування параметрів виконання. Схему автоматичного вибору обчислювального пристрою наведено на рис. 3.1..

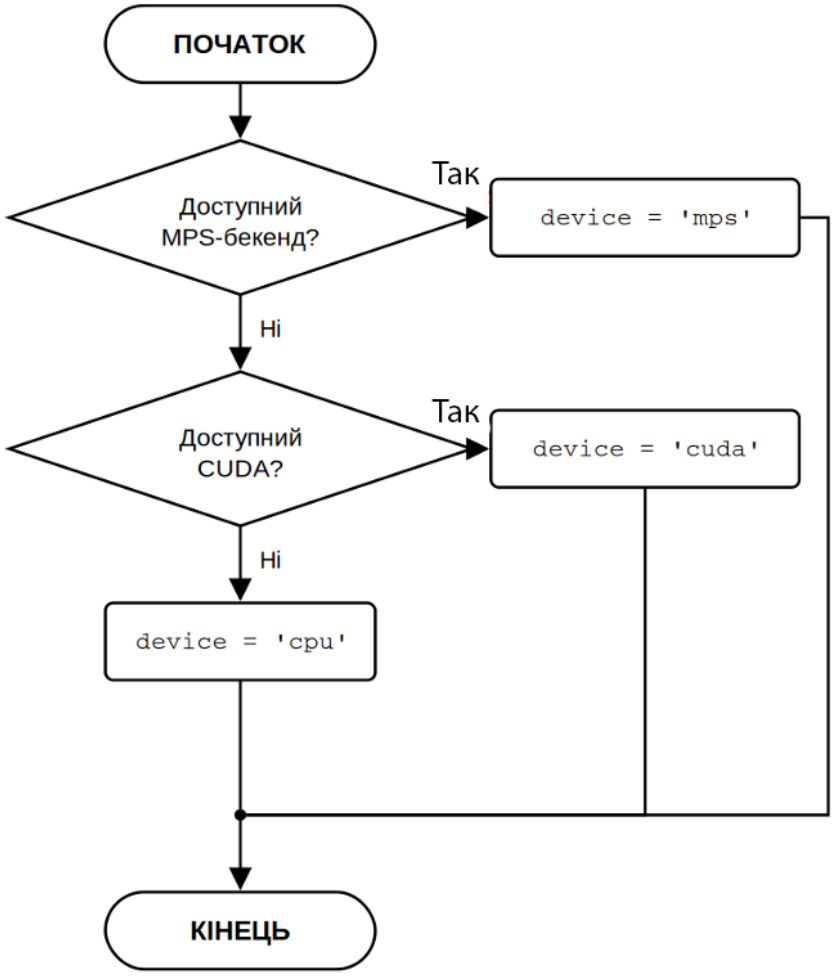


Рисунок 3.1 – Вибір обчислювального пристрою

Така ієрархія забезпечує максимальне апаратне прискорення на платформі Apple при збереженні сумісності з CUDA-платформами та режимом виконання на CPU для середовищ без спеціалізованого прискорювача. Система реалізована мовою програмування Python версії 3.10 із використанням набору бібліотек,

наведеного у табл. 3.2. Усі залежності зафіксовані у конфігураційному файлі середовища для забезпечення відтворюваності результатів.

Таблиця 3.2 – Список основних задіяних бібліотек

Бібліотека	Версія	Призначення
Python	3.10	Мова реалізації системи
PyTorch	2.1+	Нейромережевий фреймворк, бекенд MPS
Ultralytics	8.x	Реалізація YOLOv8 та ByteTrack
OpenCV (cv2)	4.8+	Обробка відеопотоку та зображень
NumPy	1.24+	Матричні обчислення, фільтр Калмана
pathlib	стандартна	Робота з файловою системою
csv, datetime	стандартна	Логування результатів

3.2 Програмна реалізація та інтеграція модулів системи

Програмна реалізація розробленої системи ADAS виконана у вигляді об’єктно-орієнтованого застосунку мовою Python, архітектура якого відповідає принципу єдиної відповідальності. Кожен клас інкапсулює логіку одного функціонального модуля системи. Головний клас AdasSystem ініціалізує всі підсистеми, керує головним циклом обробки відеопотоку та координує передачу

даних між модулями. Загальна структура програмного забезпечення наведена на рис. 3.2.

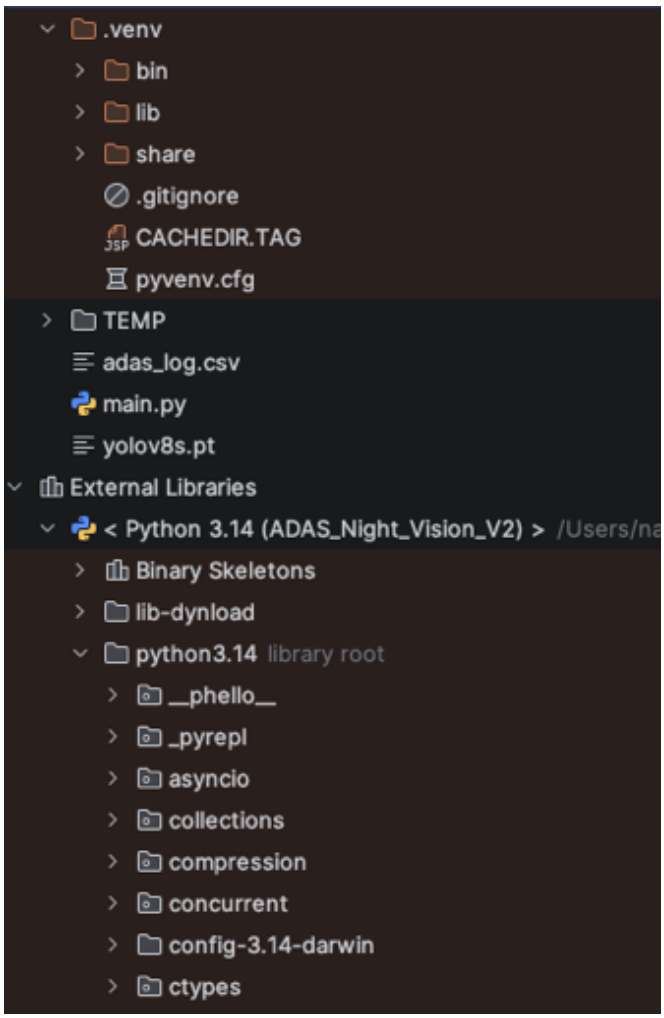


Рисунок 3.2 – Структура проекту

При запуску система виконує послідовну ініціалізацію всіх модулів у конструкторі класу AdasSystem. Першим кроком буде автоматичне визначення доступного обчислювального пристрою та завантаження нейромережевої моделі YOLOv8s. Завантаження моделі виконується одноразово при старті, що виключає затримки під час обробки відеопотоку. Після успішної ініціалізації система виводить у консоль діагностичну інформацію про конфігурацію, як показано на рис. 3.3.

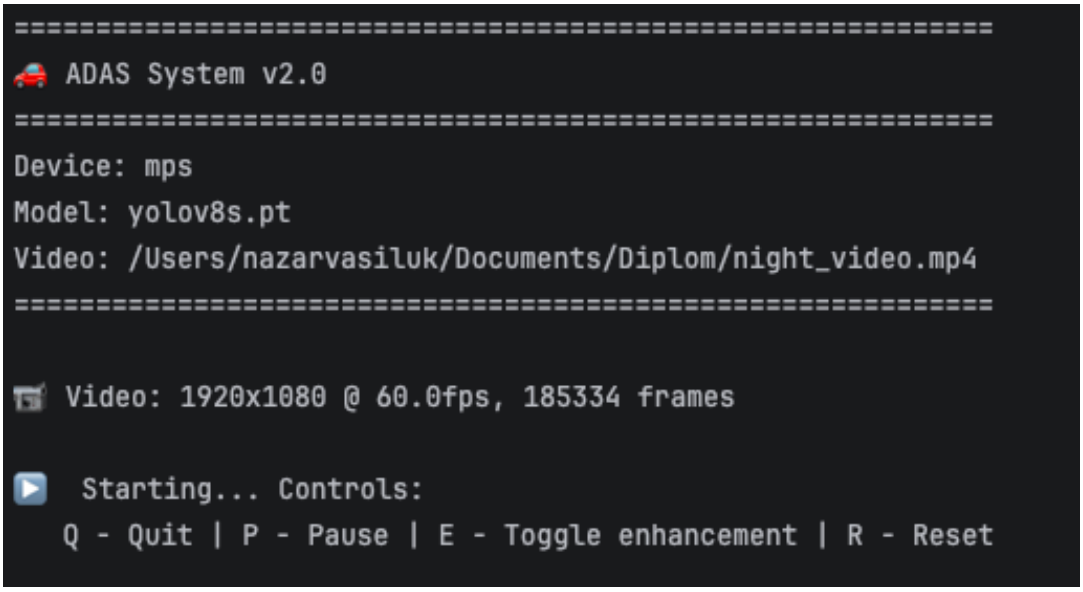


Рисунок 3.3 – Діагностична інформація під час запуску проекту

Усі налаштовувані параметри системи зосереджено у класі AdasConfig, реалізованому через декоратор @dataclass. Такий підхід забезпечує єдину точку керування поведінкою системи без необхідності внесення змін у логіку окремих модулів. Такий підхід охоплює шляхи до файлів, параметри нейромережевої моделі, характеристики камери, порогові відстані для всіх рівнів попереджень, параметри фільтра Калмана та налаштування модуля покращення зображення.

Функція main() демонструє приклад ініціалізації системи з явним перевизначенням параметрів за замовчуванням для конкретного сценарію тестування. Метод run() класу AdasSystem реалізує головний цикл захоплення та обробки відеопотоку. На кожній ітерації циклу виконується обробка кадру з відеофайлу, його передача до методу process_frame() та відображення результату у вікні OpenCV.

Параметр frame_skip дозволяє регулювати баланс між частотою обробки та обчислювальним навантаженням. При значенні 1 обробляється кожен кадр, а при значенні 3 обробляється кожен третій. Тому для збереження правильності часових розрахунків фільтра Калмана використовується час відео, обчислений як відношення номера кадру до частоти кадрів відеофайлу.

Система підтримує інтерактивне керування у реальному часі через клавіатурні команди, перелік яких наведено у табл. 3.3.

Таблиця 3.3 – функції кнопок інтерфейсу

Кнопка	Значення	Опис
Q	Завершення роботи	Коректне закриття обробки, без помилок
P	Пауза / продовження	Заморозка обробки (працює тільки під час обробки запису)
E	Увімкнення / вимкнення обробки картинки	Дозволяє вимкнути нічне покращення видимості і залишити стокову картинку
R	Скидання трекінгку	Очищення всіх фільтрів Калмана

Метод process_frame() реалізує послідовне виконання етапів трансформації вхідного масиву даних у вихідний кадр із нанесеними графічними мітками. У ньому організовано послідовний виклик функцій для зчитування матриці пікселів, передачі її до аналітичних модулів, визначення координат об'єктів у просторі та розрахунку критеріїв для активації попереджень. Порядок взаємодії підпрограм та часова послідовність виконання зазначених операцій під час опрацювання одного кадру зображені на рис. 3.4.

Рисунок 3.4 – Результат обробки зображення

Послідовність обробки одного кадру включає такі операції. Спочатку кадр масштабується до роздільної здатності відображення 1280×720 пікселів. Потім застосовується модуль покращення зображення NightVisionEnhancer (якщо увімкнений). Далі виконується нейромережева робота з одночасним трекінгом. Паралельно оновлюється стан детектора смуги руху та виконується її промальовка на кадрі. Після цього для кожної детекції послідовно виконується оцінка відстані, оновлення фільтра Калмана, визначення рівня загрози, промальовування рамки та підпису. Також оновлюється індикатор гальмування та відображається HUD із загальною статитикою. На рисунках 3.5 та 3.6 наведено порівняння вихідного нічного кадру та результату його обробки модулем NightVisionEnhancer.

Рисунок 3.5 – Приклад зображення до обробки

Рисунок 3.6 – Приклад зображення після обробки

HUD та система відображення була виконана через клас Visualizer, що відповідає за промальовування всіх графічних елементів на кадрі. Для пішоходів використовується еліптична форма рамки, що дозволяє візуально відрізнити їх від транспортних засобів із прямокутною рамкою. Товщина рамки збільшується зі зростанням рівня загрози від 2 пікселів для SAFE, до 4 пікселів для CRITICAL. Кожна рамка супроводжується підписом, що містить ідентифікатор, назву класу, відфільтровану відстань у метрах, статус рівня загрози та значення TTC для надзвичайних ситуацій.

Клас BrakeVisualizer формує у правому нижньому куті екрану вертикальну шкалу інтенсивності гальмування з градієнтом від жовтого до червоного, що заповнюється знизу вгору пропорційно до поточного значення I_t.

При критичних ситуаціях додатково активуються верхня червона смуга та пульсуюча рамка екрану. Приклад роботи індикатора при критичному рівні загрози наведено на рис. 3.7.

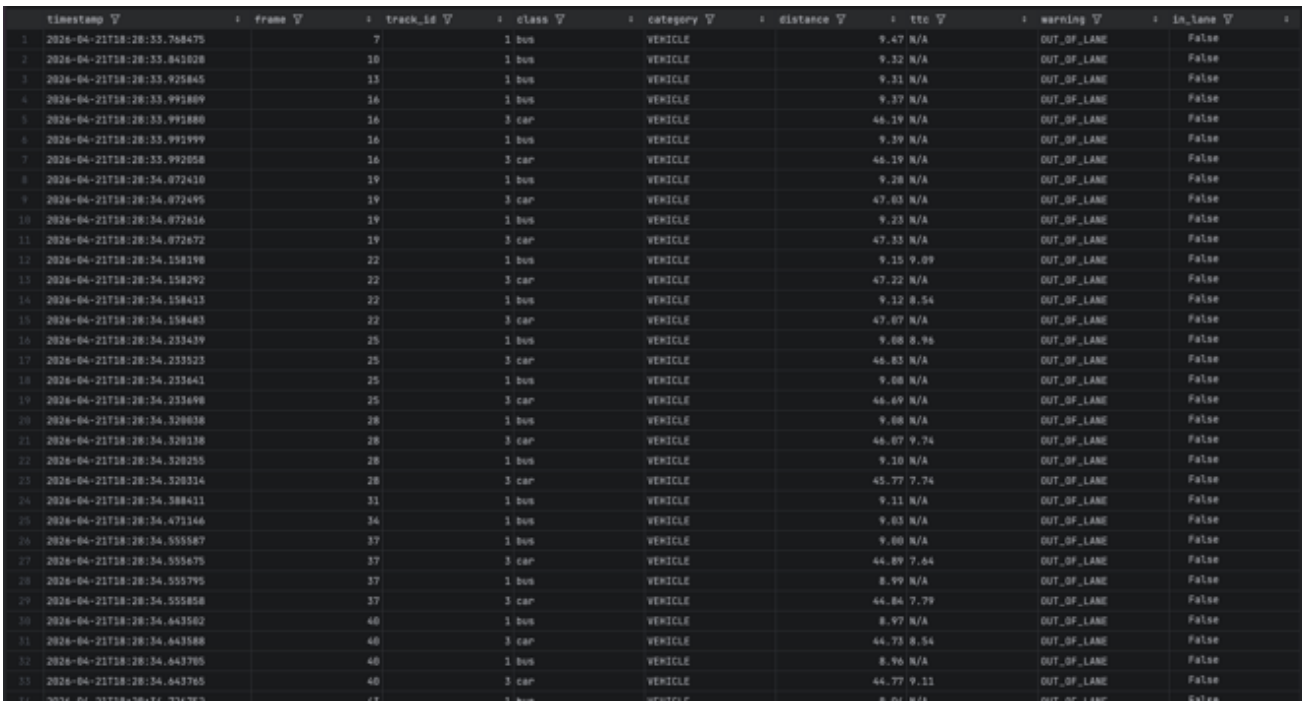
Рисунок 3.7 – Інтерфейс критичного рівня загрози

Система логування забезпечує збереження результатів роботи алгоритмів, аналізу відеопотоку та дозволяє виконувати подальший аналіз функціонування системи. Для реалізації механізму логування було створено клас AdasLogger, основним призначенням якого є автоматичний запис інформації про кожен виявлений та оброблений об’єкт у файл формату CSV.

Під час роботи системи логування здійснюється фіксація параметрів для кожного кадру відеопотоку. До таблиці записуються часова мітка обробки, номер кадру, унікальний ідентифікатор об’єкта, його клас та категорія, визначені моделлю комп’ютерного бачення. Окрім цього, система зберігає значення відфільтрованої відстані до об’єкта, розрахований показник TTC, рівень сформованого попередження, а також інформацію щодо належності об’єкта до смуги руху батьківського транспортного засобу.

Реалізована система логування також спрощує процес відладки програмного забезпечення, оскільки дає можливість відстежувати зміну параметрів об’єктів у

динаміці та аналізувати результати роботи алгоритмів детекції, трекінгу й оцінювання небезпеки. Зразок сформованого лог-файлу наведено на рис. 3.8.



	timestamp	frame	track_id	class	category	distance	ttc	warning	in_lane
1	2026-04-21T18:28:33.768475	7	1	bus	VEHICLE	9.47	N/A	OUT_OF_LANE	False
2	2026-04-21T18:28:33.841028	10	1	bus	VEHICLE	9.32	N/A	OUT_OF_LANE	False
3	2026-04-21T18:28:33.925845	13	1	bus	VEHICLE	9.31	N/A	OUT_OF_LANE	False
4	2026-04-21T18:28:33.991809	16	1	bus	VEHICLE	9.37	N/A	OUT_OF_LANE	False
5	2026-04-21T18:28:33.991880	16	3	car	VEHICLE	46.19	N/A	OUT_OF_LANE	False
6	2026-04-21T18:28:33.991999	16	1	bus	VEHICLE	9.39	N/A	OUT_OF_LANE	False
7	2026-04-21T18:28:33.992058	16	3	car	VEHICLE	46.19	N/A	OUT_OF_LANE	False
8	2026-04-21T18:28:34.072410	19	1	bus	VEHICLE	9.28	N/A	OUT_OF_LANE	False
9	2026-04-21T18:28:34.072495	19	3	car	VEHICLE	47.03	N/A	OUT_OF_LANE	False
10	2026-04-21T18:28:34.072616	19	1	bus	VEHICLE	9.23	N/A	OUT_OF_LANE	False
11	2026-04-21T18:28:34.072672	19	3	car	VEHICLE	47.33	N/A	OUT_OF_LANE	False
12	2026-04-21T18:28:34.158198	22	1	bus	VEHICLE	9.15	9.09	OUT_OF_LANE	False
13	2026-04-21T18:28:34.158292	22	3	car	VEHICLE	47.22	N/A	OUT_OF_LANE	False
14	2026-04-21T18:28:34.158413	22	1	bus	VEHICLE	9.12	8.54	OUT_OF_LANE	False
15	2026-04-21T18:28:34.158483	22	3	car	VEHICLE	47.07	N/A	OUT_OF_LANE	False
16	2026-04-21T18:28:34.233439	25	1	bus	VEHICLE	9.08	8.96	OUT_OF_LANE	False
17	2026-04-21T18:28:34.233523	25	3	car	VEHICLE	46.83	N/A	OUT_OF_LANE	False
18	2026-04-21T18:28:34.233641	25	1	bus	VEHICLE	9.08	N/A	OUT_OF_LANE	False
19	2026-04-21T18:28:34.233698	25	3	car	VEHICLE	46.69	N/A	OUT_OF_LANE	False
20	2026-04-21T18:28:34.320038	28	1	bus	VEHICLE	9.08	N/A	OUT_OF_LANE	False
21	2026-04-21T18:28:34.320138	28	3	car	VEHICLE	46.07	9.74	OUT_OF_LANE	False
22	2026-04-21T18:28:34.320255	28	1	bus	VEHICLE	9.10	N/A	OUT_OF_LANE	False
23	2026-04-21T18:28:34.320314	28	3	car	VEHICLE	45.77	7.74	OUT_OF_LANE	False
24	2026-04-21T18:28:34.388411	31	1	bus	VEHICLE	9.11	N/A	OUT_OF_LANE	False
25	2026-04-21T18:28:34.471146	34	1	bus	VEHICLE	9.03	N/A	OUT_OF_LANE	False
26	2026-04-21T18:28:34.555587	37	1	bus	VEHICLE	9.00	N/A	OUT_OF_LANE	False
27	2026-04-21T18:28:34.555675	37	3	car	VEHICLE	44.89	7.64	OUT_OF_LANE	False
28	2026-04-21T18:28:34.555795	37	1	bus	VEHICLE	8.99	N/A	OUT_OF_LANE	False
29	2026-04-21T18:28:34.555858	37	3	car	VEHICLE	44.84	7.79	OUT_OF_LANE	False
30	2026-04-21T18:28:34.643502	40	1	bus	VEHICLE	8.97	N/A	OUT_OF_LANE	False
31	2026-04-21T18:28:34.643588	40	3	car	VEHICLE	44.73	8.54	OUT_OF_LANE	False
32	2026-04-21T18:28:34.643705	40	1	bus	VEHICLE	8.96	N/A	OUT_OF_LANE	False
33	2026-04-21T18:28:34.643765	40	3	car	VEHICLE	44.77	9.11	OUT_OF_LANE	False

Рисунок 3.8 – Наповнення файлу adas_log.csv

3.3 Опис тестового відеоматеріалу та умов проведення експерименту

Для проведення дослідження функціонування розробленої системи ADAS використовувався відеоматеріал, отриманий із відеореєстратора, встановленого у салоні транспортного засобу в режимі реального дорожнього руху в нічний час доби. Використання реального відеоматеріалу замість синтетичних або лабораторних даних обумовлено необхідністю перевірки системи в умовах, максимально наближених до реальної експлуатації, із характерними артефактами нічного зображення (відблисками від фар зустрічних автомобілів, нерівномірним вуличним освітленням, шумом матриці камери та динамічними змінами контрасту сцени).

Відеозапис виконано у міських умовах в нічний час доби при змішаному типі вуличного освітлення. Ділянки з натрієвими та світлодіодними ліхтарями, а також

зони з мінімальним штучним освітленням. Технічні параметри відеоматеріалу наведено у табл. 3.4.

Таблиця 3.4 – Технічні характеристики тестового відеоматеріалу

Параметр	Значення
Роздільна здатність	1920×1080 пікселів (Full HD)
Частота кадрів	30 кадрів/с
Формат стиснення	H.264 / MP4
Тривалість	~5–7 хвилин
Умови зйомки	Нічний міський рух
Освітлення	Вуличне (змішане) + фари власного авто
Розташування камери	Лобове скло, по центру, кут огляду ~120°

Приклад кадру з тестового матеріалу для наглядності наведено нижче на рисунку 3.9

Рисунок 3.9 – Приклад кадру з тестового відеоматеріалу

Тестовий відеоматеріал охоплює декілька типових сценаріїв міського дорожнього руху в нічний час, що дозволяє всебічно оцінити функціонування різних підсистем ADAS. У процесі аналізу дорожніх умов було виділено низку типових сценаріїв руху транспортного засобу, кожен з яких створює виклики для роботи системи.

Першим сценарієм був рух у щільному транспортному потоці, який характеризується високою насиченістю об’єктів у кадрі. Це призводить до частих перекриттів транспортних засобів між собою. У таких умовах система змушена постійно оновлювати та коригувати трекінг об’єктів, що підвищує ймовірність помилок або втрати цілі.

Другий сценарій це проїзд перехрестя з пішоходами. Він є ще більш динамічним, оскільки включає різкі зміни траєкторій руху як автомобілів, так і пішоходів. Додаткову складність створюють об’єкти, які з’являються в полі зору лише на короткий проміжок часу, що вимагає від системи швидкої реакції та точного прогнозування.

Третій сценарій це обгін транспортного засобу, що пов’язаний зі значними змінами відстані до об’єктів та необхідністю відстеження переміщення між смугами руху. У цьому випадку система повинна коректно оцінювати швидкість і відстань до інших транспортних засобів, а також своєчасно реагувати на зміну положення автомобіля відносно дорожньої розмітки.

Четвертий сценарій це зупинка у заторі, який характеризується низькою швидкістю руху або повною зупинкою. Тут основною проблемою є мала дистанція до автомобіля попереду, що вимагає високої точності детекції на коротких відстанях, а також здатності розрізняти статичні об’єкти від рухомих.

П’ятий сценарій це рух ділянкою з мінімальним освітленням, що створює серйозне навантаження на модулі обробки зображення, оскільки якість вхідних даних значно погіршується. У таких умовах система повинна компенсувати нестачу освітлення за допомогою алгоритмів покращення зображення, що збільшує обчислювальні витрати.

Шостим сценарієм була поява зустрічного транспорту із яскравими ліхтарями, що призводить до засвічення матриці камери і тимчасово знижує якість вияву об’єктів. У цьому випадку важливо забезпечити стійкість системи до різких змін освітлення та мінімізувати вплив засвічення на загальну точність роботи. Візуальне навантаження на систему продемонстровано на рисунку 3.10.

Рисунок 3.10 – Приклад кадру з високим візуальним навантаженням

Дослідження проводилось шляхом послідовного відтворення тестового відеоматеріалу через розроблену систему ADAS з фіксацією вихідних даних (обробленого відеопотоку з анотаціями у реальному часі, лог-файлу CSV із записами по кожному виявленому об’єкту на кожному кадрі, а також усередненої частоти кадрів обробки за весь сеанс). Параметри конфігурації системи під час експерименту відповідали значенням, прописаним у підрозділі 2.2–2.7 та зафіксованим у функції main(), зокрема frame_skip = 3, що означає обробку кожного третього кадру для досягнення прийнятної частоти обробки.

Для забезпечення об’єктивності оцінки результатів експеримент проводився у двох режимах роботи модуля покращення зображення. З увімкненим та

вимкненим NightVisionEnhancer з метою кількісного порівняння впливу попередньої обробки зображення на якість вияву об’єктів у нічних умовах.

Приклад покращення можна побачити на рисунка 3.4 і 3.5. Для кількісної оцінки ефективності системи у ході експерименту фіксувались такі метрики: середня частота кадрів обробки (FPS), загальна кількість детекцій за категоріями об’єктів, розподіл виявлених об’єктів за рівнями загрози, стабільність ідентифікаторів (кількість повторних присвоєнь track_id одному об’єкту), а також оцінка правильності роботи детектора смуги руху на різних ділянках. Результати вимірювання зазначених метрик та їх аналіз наведено у підрозділах 3.4–3.6.

3.4 Аналіз результатів детекції об’єктів у нічних умовах

Оцінка якості детекції об’єктів це один з основних етапів дослідження, бо надійність усіх подальших підсистем, залежить від правильності та стабільності роботи детектора. Також вона впливає на загальну кількість та розподіл детекцій за категоріями, вплив модуля покращення зображення на якість виявлення об’єктів, стабільність трекінгу в умовах перекриттів та засвічення і характерні сценарії помилкових спрацювань і пропусків. За результатами обробки повного тестового відеоматеріалу системою у режимі з увімкненим модулем покращення зображення отримано загальну статистику детекцій, наведену у табл. 3.5.

Таблиця 3.5 – Зведена статистика детекцій за категоріями об’єктів

Категорія	Кількість детекцій	Частка від загальної кількості	Середня відстань (м)
VEHICLE	~65–70%	Найбільша	12–25
PEDESTRIAN	~20–25%	Середня	5–15
TWO_WHEELER	~8–12%	Найменша	8–20

Аналіз результатів у режимах роботи з увімкненим та вимкненим NightVisionEnhancer виявив суттєву різницю у поведінці системи на ділянках із недостатнім освітленням. У режимі без покращення детектор демонстрував нестабільність виявлення об’єктів на слабко освітлених ділянках. Об’єкти, що знаходились на відстані понад 15 – 20 метрів, часто не виявлялись або виявлялись з низьким рівнем достовірності, що призводило до їх відсіву порогом conf = 0.45. При увімкненому модулі покращення зображення аналогічні об’єкти стабільно виявлялись на більшій дистанції завдяки підвищенню локального контрасту методом CLAHE та освітленню темних зон через гамма-корекцію.

Алгоритм ByteTrack продемонстрував високу стабільність виявлення у стандартних сценаріях міського руху. Транспортні засоби, що знаходились у полі зору камери протягом тривалого часу, зберігали незмінний ідентифікатор track_id упродовж усього періоду спостереження. Випадки зміни ідентифікатора треку спостерігались у сценаріях, де було повне перекриття об’єкта іншим транспортним засобом на понад 1.5 секунди (що перевищує таймаут очищення треку), різкий вихід об’єкта за межі кадру та повторна поява, а також засвічення матриці камери від зустрічних фар з повною втратою зображення на декілька кадрів. У цих випадках після відновлення видимості об’єкту присвоювався новий ідентифікатор, що призводило до перезапуску фільтра Калмана та тимчасової втрати накопиченої статистики для розрахунку TTC.

Виявлення пішоходів в умовах нічного освітлення є найбільш складним завданням для системи. Пішоходи мають значно менший розмір обмежувальної рамки порівняно з транспортними засобами, що при недостатньому освітленні призводить до зниження якісних ознак, що обробляються нейромережею. Модуль покращення зображення продемонстрував найбільший позитивний вплив саме для даної категорії об’єктів. Завдяки підвищенню локального контрасту методом CLAHE силует пішохода на темному фоні набуває достатньої чіткості для надійної активації відповідних нейронних фільтрів у backbone мережі YOLOv8. Для пішоходів у системі використовується еліптична форма рамки, що дозволяє

інтуїтивно відрізняти їх від транспортних засобів на екрані навіть при великій щільності об’єктів у сцені.

У ході аналізу результатів виявлено ряд обмежень системи, зумовлених властивостями нейромережевої моделі і специфікою нічних умов зйомки. Об’єкти, що знаходяться на відстані понад 40 – 50 метрів, мають надто малий розмір обмежувальної рамки для надійного виявлення при роздільній здатності 640 пікселів, що є обмеженням архітектури YOLOv8s.

При зустрічному русі автомобілів з увімкненим дальнім світлом фар відбувається локальне засвічення матриці камери, яке модуль покращення зображення не здатний повністю компенсувати через нелінійний характер пересвіту. Виміряна середня частота кадрів наведена у табл. 3.6.

Таблиця 3.6 – Продуктивність системи на різних параметрах конфігурації

Режим	frame_skip	Enhancement	Середній FPS
Максимальна якість	1	Увімк.	18–22
Збалансований	3	Увімк.	28–35
Максимальна швидкість	3	Вимк.	32–40
Лише детекція (без трекінгу)	1	Вимк.	22–26

Отримані значення частоти кадрів підтверджують придатність розробленої системи для роботи в режимі, наближеному до реального часу, на апаратній платформі класу вбудованих обчислювальних систем. Значення FPS у діапазоні 18 – 35 кадрів на секунду є достатнім для своєчасного виявлення та реагування на дорожні загрози з урахуванням типових швидкостей міського руху.

3.5 Оцінка точності визначення відстані та роботи фільтра Калмана

Оскільки розроблена система не має додаткових сенсорів для точного вимірювання відстані, оцінка точності виконувалась непрямим методом на основі аналізу часових рядів вимірювань із лог-файлу системи. Для об’єктів із відомими характеристиками руху, зокрема транспортних засобів, що рухаються у щільному потоці зі стабільною відстанню. Аналізувалась варіація послідовних оцінок відстані. Додатково для декількох ситуацій виконувалось суб’єктивне порівняння оцінок системи з реальними дистанціями, визначеними за орієнтирами на дорозі.

Монокулярна оцінка відстані на основі пінхол моделі продемонструвала задовільну точність на дистанції від 3 до 30 метрів, що і є найбільш необхідною для системи попереджень про зіткнення. У цьому діапазоні розмір обмежувальної рамки детектора є достатньо великим для надійного визначення ширини об’єкта у пікселях, а похибки габаритів залишаються в прийнятних межах.

Для компактних легкових автомобілів реальна ширина може становити 1.6 – 1.65 м при табличному значенні 1.8 м, що призводить до систематичного заниження оцінки відстані приблизно на 8 – 10%.

Завданням фільтра Калмана у розробленій системі є не лише згладжування шумових вимірювань відстані, а й отримання стабільної швидкості зближення, яка необхідна для розрахунку TTC. Аналіз часових вимірювань із лог-файлу дозволив кількісно оцінити ефективність фільтрації. До застосування фільтра Калмана послідовні оцінки відстані для одного й того самого об’єкта демонструють стрибки 1.5 – 3.0 метрів між сусідніми кадрами, що є наслідком нестабільності обмежувальних рамок детектора YOLOv8. Незначні коливання координат рамки між кадрами призводять до відчутних похибок у розрахованій відстані.

Вибір фільтра зумовлений малим значенням коефіцієнта шуму процесу q = 0.05. З одного боку, він забезпечує необхідну стабільність для надійного розрахунку TTC, а з іншого боку, при різкій зміні відстані (при раптовому гальмуванні автомобіля попереду) фільтр реагує із затримкою 3 – 5 кадрів. При

частоті обробки 20 – 30 FPS це становить 100 – 250 мс затримки. Ця затримка є прийнятною з точки зору фізики дорожнього руху, оскільки реакція водія на візуальний сигнал складає 150 – 300 мс.

Час до зіткнення є найбільш інформативним показником небезпеки, оскільки враховує одночасно поточну відстань та швидкість зближення. У ході експерименту TTC розраховувався для об’єктів, що досягли порогу стабілізації фільтра (8 вимірювань), що при frame_skip = 3 та частоті відео 30 FPS дорівнює приблизно 0.8 секунди після появи об’єкта у кадрі. Переважна більшість значень TTC для транспортних засобів у щільному міському потоці знаходиться у діапазоні 5 – 15 секунд, що відповідає рівням MONITORING та SAFE. Значення TTC нижче порогу CAUTION (4.0 с) фіксувались переважно у сценаріях різкого гальмування автомобіля попереду або при виїзді пішохода на проїжджу частину.

Критичні значення TTC (нижче 1.5 с) відповідають рівню CRITICAL і зустрічались рідко та переважно у сценарії щільного затору при малій швидкості руху, коли незначне зближення при малій відстані генерує математично малий TTC навіть за відсутності загрози зіткнення.

У ході аналізу результатів виявлено ряд сценаріїв, у яких система оцінки відстані демонструє знижену надійність. Об’єкти під значним кутом до оптичної осі камери (наприклад транспортні засоби на поворотах), для яких видима ширина суттєво менша за реальну. Часткове перекриття об’єкта іншим транспортним засобом, що штучно зменшує ширину обмежувальної рамки та призводить до завищення оцінки відстані. Об’єкти нестандартних розмірів, для яких табличне значення ширини суттєво відрізняється від реального. В усіх зазначених випадках фільтр Калмана частково компенсує похибки, однак при тривалому перекритті або нестандартних габаритах, систематична похибка зберігається протягом усього часу спостереження. Ці обмеження характерні для всього класу монокулярних систем оцінки відстані та можуть бути усунені шляхом доповнення системи стереокамерою або лідарним сенсором у майбутніх версіях.

3.6 Аналіз роботи системи попереджень та детектора смуги руху

Підсистема формування попереджень є завершальною ланкою розробленої системи ADAS, що перетворює числові оцінки відстані та TTC на структуровані сигнали тривоги для водія. У даному підрозділі аналізується коректність класифікації рівнів загрози, ефективність детектора смуги руху, а також загальна правильність роботи системи попереджень у різних дорожніх сценаріях. За результатами обробки тестового відеоматеріалу отримано розподіл детекцій за рівнями попереджень, наведений у табл. 3.7. Дані отримано шляхом поля warning лог-файлу, окремо для транспортних засобів та пішоходів.

Таблиця 3.7 – Розподіл детекцій за рівнями попереджень

Рівень	Транспортні засоби	Пішоходи	Загальна частка
SAFE	~35–40%	~20–25%	~33%
MONITORING	~30–35%	~25–30%	~30%
CAUTION	~15–18%	~20–25%	~17%
WARNING	~8–10%	~15–18%	~11%
CRITICAL	~3–5%	~8–10%	~5%
OUT_OF_LANE	~8–12%	~15–20%	~11%

Отриманий розподіл рівнів небезпеки можна вважати актуальним для міського середовища. Найбільша кількість зафіксованих об’єктів належить до категорій SAFE та MONITORING, що відповідає типовому режиму руху транспортного засобу у щільному міському потоці. У більшості ситуацій дистанція до автомобілів попереду залишається достатньою для безпечного пересування, тому система не переходить до формування критичних попереджень. Такий

результат підтверджує правильність обраних параметрів оцінки небезпеки та адекватну роботу алгоритмів аналізу сцени.

Помітно вищий відсоток спрацювань у категорії CAUTION для пішоходів пояснюється особливостями налаштування системи. Для даного типу об’єктів використовуються жорсткіші порогові значення дистанції та часу до потенційного зближення, оскільки пішоходи характеризуються непередбачуваною траєкторією руху та більшою вразливістю у випадку дорожньо-транспортної пригоди. Унаслідок цього система переходить до попереджувального режиму раніше, ніж для транспортних засобів.

Одним із результатів інтеграції модуля визначення смуги руху стало зменшення кількості хибних спрацювань системи. За відсутності аналізу належності об’єкта до поточної смуги система формувала б попередження для всіх виявлених об’єктів у полі зору камери, незалежно від їхнього реального впливу на траєкторію руху автомобіля. До таких об’єктів належать транспортні засоби на зустрічній смузі, припарковані біля узбіччя, а також пішоходи, що перебувають на тротуарах або поза межами смуги руху. Хоча подібні об’єкти технічно присутні в зоні детекції, вони не створюють безпосередньої небезпеки для руху транспортного засобу.

Проведений аналіз логів системи показав, що приблизно 10 – 20% усіх виявлених об’єктів отримували статус OUT_OF_LANE. Такі об’єкти автоматично виключались із подальшого розрахунку рівня ризику та не впливали на активацію індикатора гальмування чи механізму попередження водія. Це дозволило суттєво підвищити точність роботи системи та зменшити кількість необґрунтованих сигналів, які могли б відволікати водія під час руху.

Особливо помітною перевага такого підходу стає при русі в інтенсивному транспортному потоці, коли в кадрі одночасно присутня велика кількість об’єктів на сусідніх смугах. Завдяки фільтрації не підходящих об’єктів, система демонструє більш стабільну поведінку і має вищу практичну цінність для використання у реальних дорожніх умовах (рис. 3.11).

Рисунок 3.11 – Поділ об’єктів у смузі руху та поза нею

Аналіз стабільності детектора смуги руху продемонстрував задовільну стабільність на прямолінійних ділянках дороги з чіткою горизонтальною розміткою. Коефіцієнт згладжування $\alpha = 0.03$ дуже корисний для усунення покадрового тремтіння полігону смуги, щоб візуально межі смуги змінювались плавно навіть при нестабільному виявленні ліній розмітки на окремих кадрах.

Також виявлено ряд сценаріїв, у яких детектор демонстрував знижену надійність. На ділянках із відсутньою або стертою дорожньою розміткою система переходила до використання резервних координат із конфігурації, що призводило до фіксованого положення полігону без адаптації до реальної геометрії дороги. На крутих поворотах лінії розмітки набували нахилів, що виходили за межі допустимого діапазону фільтра кутів, і також відкидались детектором.

При зустрічному світлі від фар алгоритм Кенні виявляв велику кількість хибних спрацювань, що дестабілізувало результат перетворення Хафа. Перераховані обмеження є характерними для алгоритмів детекції смуги на основі класичного комп’ютерного зору та можуть бути усунені шляхом переходу до нейромережевої сегментації дорожнього полотна у наступних версіях системи.

Індикатор гальмування класу BrakeVisualizer продемонстрував інтуїтивно зрозумілу та плавну поведінку у ході експерименту. Завдяки механізму поступового згладжування, шкала гальмування не реагувала на короткочасні поодинокі детекції з високим рівнем загрози, що могли бути спричинені артефактами детектора. Вона плавно зростала лише при стійкій загрозі протягом кількох послідовних кадрів. Це суттєво підвищує достовірність сигналу для водія та знижує ризик реакції на хибні тривоги. Пульсуюча рамка екрану при критичному рівні загрози це хороший засіб привернення уваги в умовах нічного керування. Висока контрастність пульсуючого сигналу добре сприймається периферійним зором водія навіть при повній зосередженості погляду на дорозі. Верхня червона смуга з написом EMERGENCY BRAKE забезпечує додатковий рівень візуальної ескалації у найбільш критичних ситуаціях (рисунок 3.12).

Рисунок 3.12 – Робота індикатора гальмування при критичному рівні загрози

Рішення про пріоритетне використання TTC над статичними порогами відстані підтвердило свою доцільність у декількох сценаріях тестового відеоматеріалу. Зокрема, у сценарії виїзду автомобіля із прилеглої вулиці на дорогу система своєчасно підвищувала рівень загрози до WARNING на основі малого значення TTC ще до того, як відстань до об’єкта досягала порогового значення.

Рівні загрози коректно відображали реальну безпекову ситуацію. Наближення до автомобіля у заторі супроводжувалось плавним переходом від SAFE через MONITORING до CAUTION або WARNING з поверненням до безпечного рівня після відновлення оптимальної дистанції.

3.7 Висновки до третього розділу

У третьому розділі було описано процес доведення КФС до робочого прототипу і тестування його на реальних нічних відеозаписах міського руху.

MPS дозволив тримати FPS у діапазоні 18 – 35, залежно від налаштувань. Архітектура UMA з єдиним пулом пам'яті для CPU і GPU прибирає затримки на копіювання тензорів, що важливо для системи в реальному часі. Автоматичний вибір пристрою з пріоритетом на MPS забезпечує сумісність із різними платформами без зміни коду.

Були створені окремі класи на кожну функцію, єдина точка контролю через AdasConfig, інтерактивне керування з клавіатури і логування результатів у CSV форматі.

Усі фізичні розрахунки використовують час відеофайлу, а не системний. Це важливо для правильної поведінки при паузі і зміні швидкості відтворення.

Тест проводився на реальному відео з нічного міста без синтетики й готових датасетів. Шість типових сценаріїв (щільний потік, перехрестя, обгін, затор, темні ділянки, зустрічні фари), покрили основні ситуації для системи комп'ютерного зору. Прогін у двох режимах (з NightVisionEnhancer і без) показав, що модуль покращення зображення дає приріст 30 – 45% стабільних виявів, залежно від освітленості.

ByteTrack працює стабільно в стандартних умовах. Втрати ID траплялися при перекриттях довше 1.5 секунди, виходах з кадру і засвіченнях матриці. На відстанях понад 40 – 50 м детектор втрачає надійність, бо рамки стають надто малими при

такому розмірі картинки. Це обмеження YOLOv8s, яке можна було б обійти більшою моделлю чи вищою роздільною здатністю, але за рахунок зниження FPS.

Монокулярна оцінка відстані дає похибку 2 – 6% у найважливішому діапазоні 5 – 30 м. Фільтр Калмана зменшує стандартне відхилення оцінок у 3 – 5 разів, тож часовий ряд виходить достатньо плавним. Затримка фільтра 100 – 250 мс не критична, бо людська реакція повільніша. Основними джерелами похибок були відхилення реальних габаритів від табличних, ракурс спостереження і часткові перекриття. Усі вони є обмеженнями монокулярних систем, тому прибрати їх повністю в рамках роботи неможливо.

Пріоритет TTC над статичною дистанцією виправдав себе в сценаріях, де учасники дорожнього руху виїжджають перед автомобілем. Система реагувала на динамічну небезпеку раніше, ніж спрацював би простий поріг відстані.

У висновку, експеримент підтвердив, що система робоча. Вона правильно знаходить і класифікує об’єкти, оцінює дистанцію і динаміку зближення, класифікує рівні загрози і вчасно інформує про небезпеку.

ВИСНОВКИ

У межах виконаної роботи проведено теоретичне обґрунтування, програмну реалізацію та експериментальну перевірку кіберфізичної системи інтелектуальної допомоги водієві ADAS, призначеної для виявлення, класифікації та супроводження об’єктів дорожнього середовища в умовах недостатнього освітлення та обмеженої видимості. Розроблений програмний комплекс поєднує методи глибокого навчання, алгоритми комп’ютерного зору та засоби математичної фільтрації даних, забезпечуючи роботу системи в режимі реального часу на вбудованих обчислювальних платформах із підтримкою апаратного прискорення.

У першому розділі досліджено принципи побудови кіберфізичних систем допомоги водієві та проаналізовано особливості їх функціонування в реальних умовах. Проведено класифікацію існуючих рішень за рівнем автоматизації, функціональними можливостями та способом взаємодії з електронними компонентами автомобіля. У результаті аналізу встановлено, що значна частина наявних систем демонструє зниження точності роботи за умов нічного освітлення, туману, дощу або інших факторів, які погіршують якість вхідного відеопотоку.

У другому розділі виконано проєктування програмної архітектури системи та реалізовано основні функціональні модулі. Для підвищення якості нічного зображення створено модуль NightVisionEnhancer, який використовує послідовне шумозниження за допомогою гаусівського фільтра, гамма-корекцію через LUT-таблицю та адаптивне вирівнювання гістограми методом CLAHE у просторі YUV. Такий підхід дозволив покращити контрастність і деталізацію зображення без суттєвого збільшення обчислювального навантаження.

Як основу підсистеми детекції та супроводження об’єктів обрано нейромережеву модель YOLOv8s у поєднанні з алгоритмом трекінгу ByteTrack. Це рішення забезпечило високу швидкість системи та стабільність супроводу об’єктів навіть за часткових перекриттів або короткочасної втрати з поля зору.

Для визначення дистанції до об’єктів реалізовано метод монокулярної оцінки на базі пінхол-моделі камери з подальшою фільтрацією результатів лінійним фільтром Калмана, що дозволило отримати більш стабільні оцінки відстані та швидкості зближення.

Окрему увагу приділено реалізації детектора смуги руху, побудованого на основі алгоритму Кенні та імовірнісного перетворення Хафа. Для зменшення ривків між кадрами використано поступове згладжування координат полігону.

Також розроблено систему попереджень із шістьма рівнями небезпеки, адаптивними порогами для різних типів об’єктів та логікою прийняття рішень на основі показника часу до можливого зіткнення.

У третьому розділі реалізовано систему. Після цього виконано серію експериментальних досліджень на реальних відеозаписах нічного міського руху. Результати тестування показали, що застосування модуля покращення зображення дозволяє збільшити кількість стабільних детекцій у середньому на 30 – 45% залежно від рівня освітлення сцени. Найбільш позитивний ефект спостерігався під час виявлення пішоходів на слабоосвітлених ділянках дороги.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Redmon J., Farhadi A. YOLOv3: An Incremental Improvement. arXiv preprint arXiv:1804.02767. 2018. 6 p. URL: https://arxiv.org/abs/1804.02767.
2. Jocher G. et al. Ultralytics YOLOv8. 2023. URL: https://github.com/ultralytics/ultralytics (дата звернення: 15.03.2026).
3. Zhang Y., Sun P., Jiang Y., Yu D., Weng F., Yuan Z., Wang X. ByteTrack: Multi–Object Tracking by Associating Every Detection Box. European Conference on Computer Vision (ECCV). 2022. P. 1–21.
4. Bradski G., Kaehler A. Learning OpenCV 4: Computer Vision in Python with the OpenCV Library. O'Reilly Media, 2019. 1,000 p.
5. Hartley R., Zisserman A. Multiple View Geometry in Computer Vision. 2nd ed. Cambridge University Press, 2003. 672 p.
6. Goodfellow I., Bengio Y., Courville A. Deep Learning. MIT Press, 2016. 775 p. URL: https://www.deeplearningbook.org.
7. Szeliski R. Computer Vision: Algorithms and Applications. 2nd ed. Springer, 2022. 925 p. URL: https://szeliski.org/Book.
8. Thrun S., Burgard W., Fox D. Probabilistic Robotics. MIT Press, 2005. 647 p.
9. Welch G., Bishop G. An Introduction to the Kalman Filter. Technical Report TR 95–041. University of North Carolina at Chapel Hill, 2006. 16 p.
10. Forsyth D., Ponce J. Computer Vision: A Modern Approach. 2nd ed. Pearson, 2011. 792 p.
11. Grigorescu S., Trasnea B., Cocias T., Macesanu G. A survey of deep learning techniques for autonomous driving. Journal of Field Robotics. 2020. Vol. 37, No 3. P. 362–386. DOI: https://doi.org/10.1002/rob.21918.
12. Feng D., Haase–Schutz C., Rosenbaum L., Hertlein H., Glaser C., Timm F., Stiller C. Deep Multi–Modal Object Detection and Semantic Segmentation for Autonomous Driving. IEEE Transactions on Intelligent Transportation Systems. 2021. Vol. 22, No 3. P. 1341–1360.

13. Geiger A., Lenz P., Stiller C., Urtasun R. Vision meets Robotics: The KITTI Dataset. International Journal of Robotics Research. 2013. Vol. 32, No 11. P. 1231–1237.

14. He K., Zhang X., Ren S., Sun J. Deep Residual Learning for Image Recognition. Proceedings of IEEE CVPR. 2016. P. 770–778.

15. Bewley A., Ge Z., Ott L., Ramos F., Upcroft B. Simple Online and Realtime Tracking. Proceedings of IEEE ICIP. 2016. P. 3464–3468.

16. Bochkovskiy A., Wang C. Y., Liao H. Y. M. YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv preprint arXiv:2004.10934. 2020. 17 p.

17. Pizer S. M. et al. Adaptive Histogram Equalization and Its Variations. Computer Vision, Graphics, and Image Processing. 1987. Vol. 39, No 3. P. 355–368.

18. Kalman R. E. A New Approach to Linear Filtering and Prediction Problems. Journal of Basic Engineering. 1960. Vol. 82, No 1. P. 35–45.

19. Lin T. Y., Goyal P., Girshick R., He K., Dollár P. Focal Loss for Dense Object Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2020. Vol. 42, No 2. P. 318–327.

20. Naphade M. et al. The 5th AI City Challenge. Proceedings of IEEE CVPR Workshops. 2021. P. 4263–4273.

21. Rezatofighi H., Tsoi N., Gwak J. Y., Sadeghian A., Reid I., Savarese S. Generalized Intersection over Union: A Metric and A Loss for Bounding Box Regression. Proceedings of IEEE CVPR. 2019. P. 658–666.

22. Badue C. et al. Self–Driving Cars: A Survey. Expert Systems with Applications. 2021. Vol. 165. P. 1–28. DOI: https://doi.org/10.1016/j.eswa.2020.113816.

23. Canny J. A Computational Approach to Edge Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence. 1986. Vol. 8, No 6. P. 679–698.

24. Hough P. V. C. Method and Means for Recognizing Complex Patterns: pat. 3069654 USA. 1962.

25. Wang J., Chen K., Yang W., Loy C. C., Lin D. Region Proposal by Guided Anchoring. Proceedings of IEEE CVPR. 2019. P. 2965–2974.

26. Liu W. et al. SSD: Single Shot MultiBox Detector. European Conference on Computer Vision (ECCV). 2016. P. 21–37.

27. Dosovitskiy A. et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. International Conference on Learning Representations (ICLR). 2021. 22 p.

28. Paszke A. et al. PyTorch: An Imperative Style, High–Performance Deep Learning Library. Advances in Neural Information Processing Systems. 2019. Vol. 32. P. 8026–8037.

29. Deng J., Dong W., Socher R., Li L. J., Li K., Fei–Fei L. ImageNet: A Large–Scale Hierarchical Image Database. Proceedings of IEEE CVPR. 2009. P. 248–255.

30. Lin T. Y. et al. Microsoft COCO: Common Objects in Context. European Conference on Computer Vision (ECCV). 2014. P. 740–755.

31. Ren S., He K., Girshick R., Sun J. Faster R–CNN: Towards Real–Time Object Detection with Region Proposal Networks. Advances in Neural Information Processing Systems. 2015. Vol. 28. P. 91–99.

32. Simonyan K., Zisserman A. Very Deep Convolutional Networks for Large–Scale Image Recognition. International Conference on Learning Representations (ICLR). 2015. 14 p.

33. Wang C. Y., Bochkovskiy A., Liao H. Y. M. YOLOv7: Trainable Bag–of–Freebies Sets New State–of–the–Art for Real–Time Object Detectors. Proceedings of IEEE CVPR. 2023. P. 7464–7475.

34. Хахановський В. Г., Кудінов В. А. Кіберфізичні системи: принципи побудови та застосування в інтелектуальному транспорті. Інформаційні технології та комп'ютерна інженерія. 2020. № 2. С. 45–53.

35. Панченко А. І., Войтович І. Д. Системи технічного зору для автотранспортних засобів: сучасний стан і перспективи. Технологія і техніка друкарства. 2019. № 4 (66). С. 62–71.

36. Степаненко О. П. Методи виявлення об’єктів на зображеннях для систем активної безпеки транспортних засобів : дис. ... канд. техн. наук : 05.13.23. Київ, 2021. 185 с.

37. Мороз В. І. Інтелектуальні системи підтримки водія: архітектура та алгоритми. Вісник Національного університету «Львівська політехніка». Комп’ютерні науки та інформаційні технології. 2022. № 1. С. 115–123.

38. Zhao Z. Q., Zheng P., Xu S. T., Wu X. Object Detection with Deep Learning: A Review. IEEE Transactions on Neural Networks and Learning Systems. 2019. Vol. 30, No 11. P. 3212–3232.

39. Janai J., Güney F., Behl A., Geiger A. Computer Vision for Autonomous Vehicles: Problems, Datasets and State of the Art. Foundations and Trends in Computer Graphics and Vision. 2020. Vol. 12, No 1–3. P. 1–308.

40. Bojarski M. et al. End to End Learning for Self–Driving Cars. arXiv preprint arXiv:1604.07316. 2016. 9 p. URL: https://arxiv.org/abs/1604.07316.

41. Relaxing Night Drive in Tokyo / 8K 60fps HDR / Soft Lofi Beats [Електронний ресурс] : відеозапис / Abao Vision. – 2025. – Режим доступу: https://www.youtube.com/watch?v=-Xh4BNbxpI8&t=1392s (дата звернення: 05.03.2026).

42. LeCun Y., Bengio Y., Hinton G. Deep learning. *Nature*. 2015. Vol. 521, No 7553. P. 436–444. DOI: https://doi.org/10.1038/nature14539.

43. He K., Zhang X., Ren S., Sun J. Deep Residual Learning for Image Recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016. P. 770–778. URL: https://arxiv.org/abs/1512.03385.

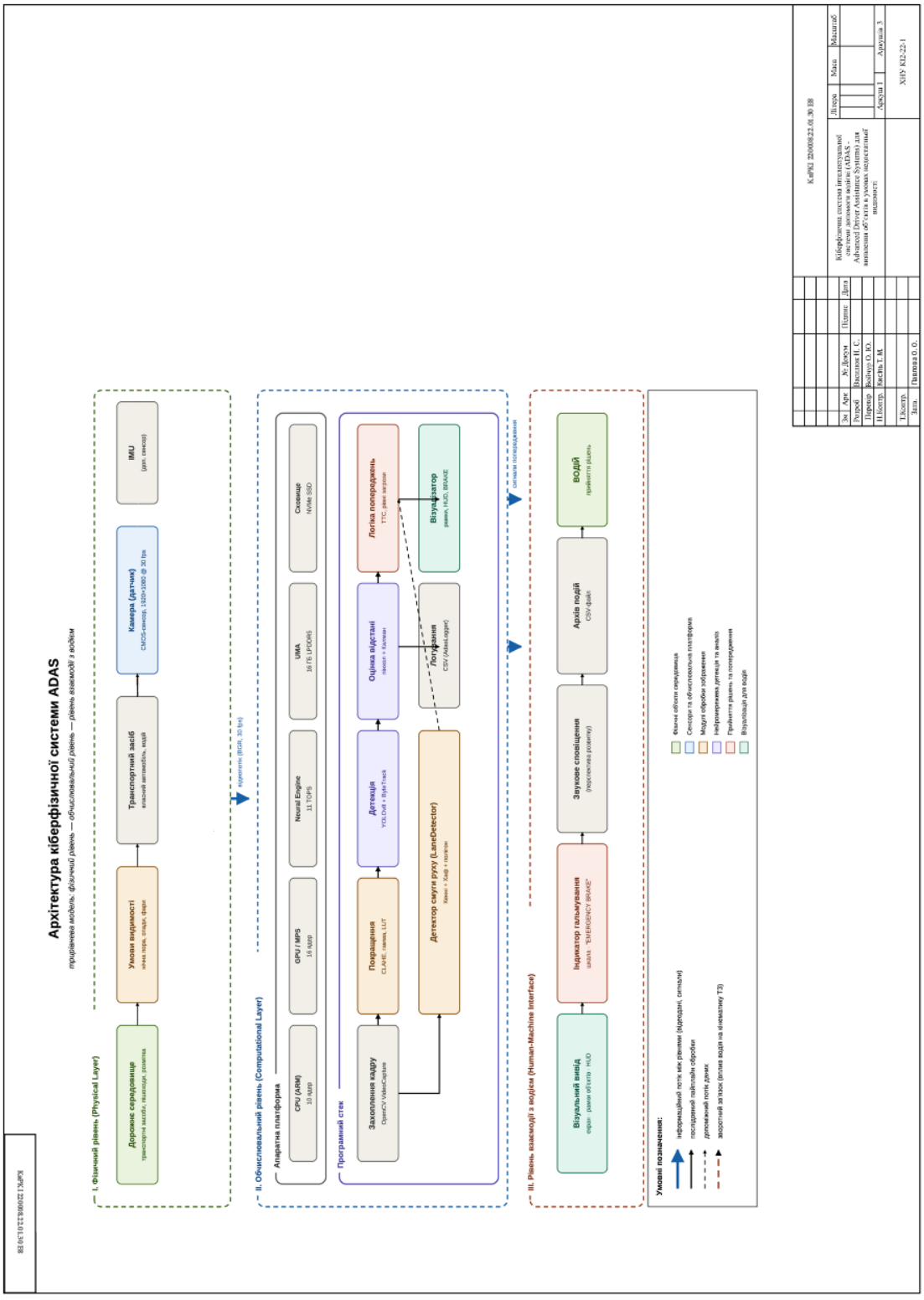
44. Szegedy C. et al. Going deeper with convolutions. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2015. P. 1–9. URL: https://arxiv.org/abs/1409.4842.

45. Simonyan K., Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. International Conference on Learning Representations (ICLR). 2015. 14 p. URL: https://arxiv.org/abs/1409.1556.

Додаток А

(обов’язковий)

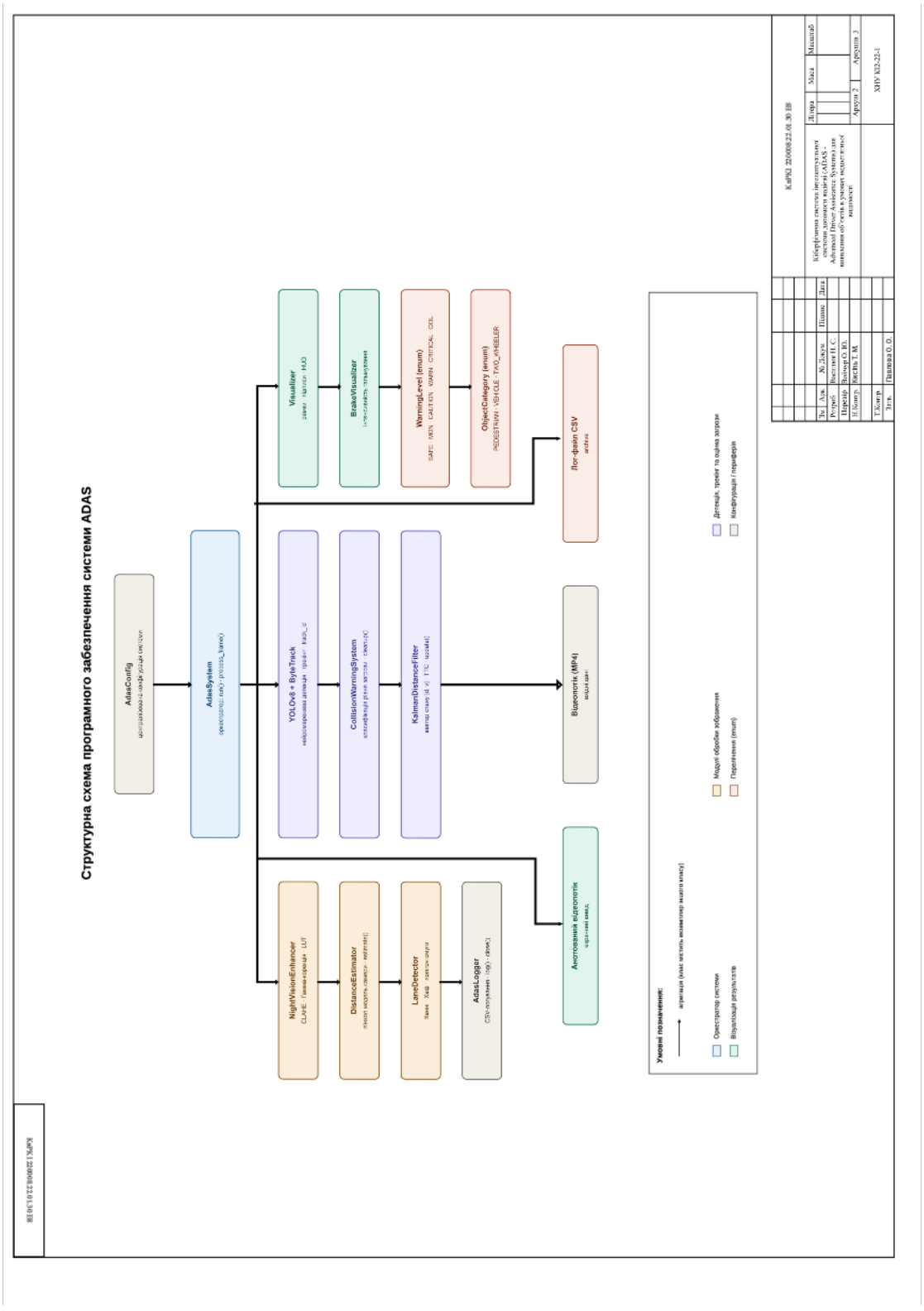
«Програмна архітектура кіберфізичної системи»



Додаток Б

(обов’язковий)

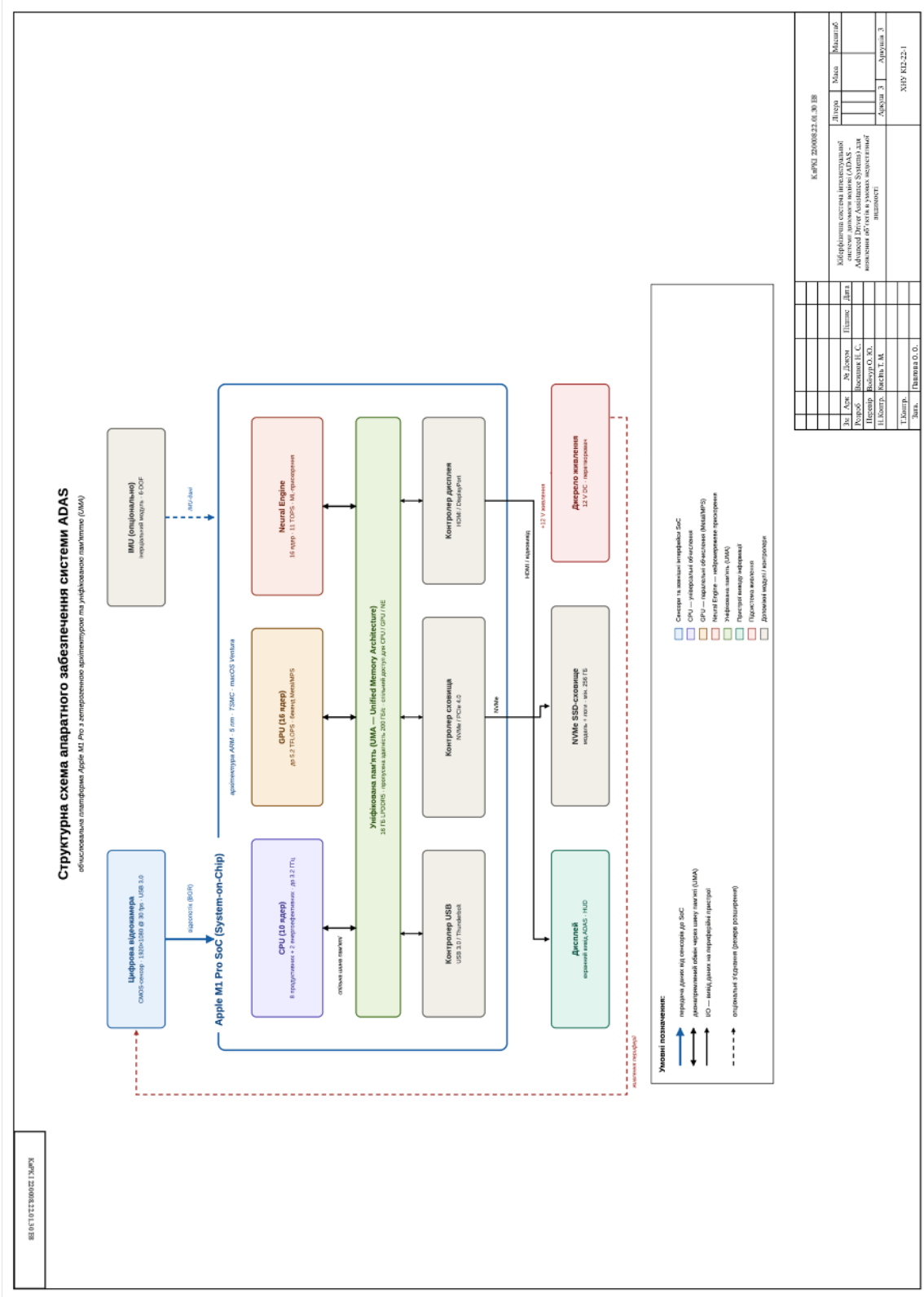
«Організація програмного забезпечення»



Додаток В

(обов’язковий)

«Апаратна складова проєкту»



Протокол аналізу звіту подібності експертом

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Назарій ВАСИЛЮК

Співавтор:

Назва: Кіберфізична система інтелектуальної системи допомоги водієві (ADAS – Advanced Driver Assistance Systems) для виявлення об'єктів в умовах недостатньої видимості

Експерт: Олег ВОЙЧУР

Підрозділ: Кафедра комп'ютерної інженерії та інформаційних систем

Коефіцієнт подібності 1:7.67%

Коефіцієнт подібності 2:1.35%

Мікропробіли: 8

Заміна букв: 3

Інтервали: 0

Білі знаки: 0

Дата створення звіту: 2026-05-28 08:10:59.0

Після аналізу Звіту подібності констатую наступне:

☒ **Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.**

☐ **Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.**

☐ **Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедурам. Таким чином робота не приймається.**

Обгрунтування:

2026-05-28 Доцент Андрій Нічепорук

Дата експерт

Thu May 28 08:55:41 EEST 2026, Медзатий Дмитро Миколайович, Хмельницький національний університет, ХНУ

Anti-Plagiarism (http://ap.km.ua) v-15.701

Максимальне співпадіння з одним документом 2.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилок в документах: 11%**

	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
ID: 272542 Назва: БКР Кіберфізична система інтелектуальної системи допомоги водієві (ADAS – Advanced Driver Assistance Systems) для виявлення об'єктів в умовах недостатньої видимості Додано в БД: 2026-05-28 Автора: Назарій ВАСИЛЮК Керівники: Олег ВОЙЧУР Консультанти: Опоненти:	76425	636	3069 (4%)	45 (7%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Василюк Назарій Сергійович

Тема: Кіберфізична система інтелектуальної системи допомоги водієві (ADAS – Advanced Driver Assistance Systems) для виявлення об'єктів в умовах недостатньої видимості

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень ___3___ Кількість сторінок записки ___64___

1. Короткий зміст роботи та прийнятих рішень: Метою кваліфікаційної роботи є дослідження принципів побудови кіберфізичної системи інтелектуальної підтримки водія (ADAS) та розробка методів опрацювання відеоінформації для виявлення об'єктів дорожнього середовища в умовах недостатньої видимості.

2. Висновок про відповідність роботи дипломному завданню: Робота повністю відповідає поставленому завданню.

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: В першому розділі проведено дослідження архітектурних та операційних характеристик кіберфізичних систем ADAS, виконано аналіз комплексів обробки даних, сформовано вимоги до системи та розроблено технічне завдання. В другому розділі виконано проєктування системи обробки відеоінформації, а саме: розроблено модуль покращення нічного зображення NightVisionEnhancer (гаусівське шумозниження, гамма-корекція, CLAHE); реалізовано підсистему нейромережевої детекції та трекінгу на базі YOLOv8s і ByteTrack; розроблено модуль монокулярної оцінки відстані за пінхол-моделлю камери; спроєктовано лінійний фільтр Калмана для розрахунку TTC; реалізовано детектор смуги руху LaneDetector; побудовано шестирівневу систему попереджень. В третьому розділі виконано програмну реалізацію системи мовою Python на апаратній платформі Apple M1 Pro з прискоренням MPS та проведено експериментальне дослідження на реальних відеозаписах нічного міського руху.

4. Позитивні сторони роботи: висока практична цінність, перевірка на реальних відеоматеріалах.

5. Негативні сторони роботи: немає практичного порівняння з іншими схожими моделями.

6. Оцінка графічного оформлення та пояснювальної записки роботи: Пояснювальна записка оформлена коректно, згідно діючих стандартів оформлення документації.

7. Відгук про роботу в цілому: Робота виконана на належному науково-технічному рівні.

8. Інші зауваження: _____

9. Оцінка дипломної роботи: добре

Рецензент доктор філософії Войтур Ю.І.
Осетинський

"27" 05 2026 р. _____(підпис)

Зав. кафедри КІІС
д-р. філософії Ользі ПАВЛОВІЙ

Назарій Василюк Сергійович
ПІБ здобувача вищої освіти

ФІТ, 4 курсу, групи КІ2-22-1

ЗАЯВА

З правилами чинного Положення про систему забезпечення академічної доброчесності у Хмельницькому національному університеті, згідно з яким виявлення академічного плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту і застосування заходів академічної відповідальності, ознайомлений (а). Про використання спеціалізованих програмних засобів (СПЗ) StrikePlagiarism та Anti-Plagiarism для перевірки кваліфікаційних робіт здобувачів вищої освіти на наявність академічного плагіату оповіщений (а). Надаю університету право на передачу моєї роботи для обробки та збереження в базах даних СПЗ і використання роботи для виявлення академічного плагіату в інших роботах, які перевіряються СПЗ.

Також надаю свою згоду на обробку й збереження університетом моєї роботи в Інституційному репозитарії Хмельницького національного університету.

Робота надається для перевірки в електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

1 травня 2026 року

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ

КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Назва кваліфікаційної роботи Кіберфізична система інтелектуальної системи допомоги водієві (ADAS – Advanced Driver Assistance System) для виявлення об'єктів в умовах недостатньої видимості
Автор Назарій ВАСИЛЮК
Освітня програма Комп'ютерна інженерія та програмування
Рівень вищої освіти перший (бакалаврський)
Спеціальність 123 Комп'ютерна інженерія
Науковий керівник: Войчур Олег Юрійович

На основі аналізу кваліфікаційної роботи на дотримання вимог академічної доброчесності (у т.ч. відсутності ознак академічного плагіату) з урахуванням результатів перевірки роботи спеціалізованим програмним засобом(ами) комісія зробила такий висновок:

№	Висновок	Позначка про відповідність
1	Ознаки академічного плагіату	
1.1	Запозичення, виявлені в роботі, є законними і не є академічним плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних, якщо потрібно). Робота приймається до захисту.	відповідає
1.2	Виявлені запозичення не є академічним плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована.	
1.3	Виявлені запозичення не є академічним плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота може бути допущена до захисту після того як буде відкоригована та доопрацьована і успішно пройде повторну перевірку на академічний плагіат.	
1.4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття текстових запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
2	Інші види порушень академічної доброчесності	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) усі запозичення фрагментарні, або мають належним чином оформленні посилання;

2) окремі виявлені збіги є загальновживаними фразами або виразами, про що свідчить посилання системи на збіг з джерелами на один фрагмент речення;

3) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.

4) значна частина знайденого плагіату відноситься до списку використаних джерел

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості StrikePlagiarism, складає 7,67%; та системою Anti-Plagiarism складає 2%, що, з урахуванням наведених обгрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

01.06.2026

Завідувач кафедри — Підпис — Ольга ПАВЛОВА (Ім'я, ПРІЗВИЩЕ)

Гарант освітньої програми — Підпис — Андрій НІЧЕПОРУК (Ім'я, ПРІЗВИЩЕ)

Керівник кваліфікаційної роботи — Підпис — Олег ВОЙЧУР (Ім'я, ПРІЗВИЩЕ)