

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ВОЛИНСЬКИЙ НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ ІМЕНІ ЛЕСІ УКРАЇНКИ

Кафедра комп'ютерних наук та кібербезпеки

На правах рукопису

ЦАЙ БАЙАН

**ПРОЄКТУВАННЯ ТА РОЗРОБКА БАГАТОМОДАЛЬНОЇ
ІНФОРМАЦІЙНОЇ СИСТЕМИ ВИЯВЛЕННЯ НЕБЕЗПЕЧНОЇ
ПОВЕДІНКИ ВОДІЯ НА ОСНОВІ ГЛИБОКИХ НЕЙРОННИХ МЕРЕЖ**

Спеціальність: 122 Комп'ютерні науки

Освітньо-професійна програма: Комп'ютерні науки та інформаційні технології

Робота на здобуття освітнього ступеня «магістр»

Науковий керівник:

Гришанович Тетяна Олександрівна,
кандидат фізико-математичних наук,
доцент кафедри комп'ютерних наук
та кібербезпеки

РЕКОМЕНДОВАНО ДО ЗАХИСТУ

Протокол №__

засідання кафедри комп'ютерних наук та
кібербезпеки

від _____ 2025 р.

Завідувач кафедри

(____) Гришанович Т. О.

ЛУЦЬК – 2025

ЗМІСТ

ВСТУП	3
РОЗДІЛ 1. АНАЛІЗ СУЧАСНИХ МЕТОДІВ ВИЯВЛЕННЯ НЕБЕЗПЕЧНОЇ ПОВЕДІНКИ ВОДІЯ	5
1.1. Проблеми безпеки дорожнього руху та небезпечної поведінки водіїв	5
1.2. Ключові технології, що використовуються при розробці інформаційних систем для виявлення небезпечної поведінки водія	9
РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА РОЗРОБКА БАГАТОМОДАЛЬНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ВИЯВЛЕННЯ НЕБЕЗПЕЧНОЇ ПОВЕДІНКИ ВОДІЯ	13
2.1. Постановка задачі, призначення та вимоги до інформаційної системи	13
2.2. Підготовка та попередня обробка системних наборів даних для навчання моделі YOLOv8	15
2.3. Розробка та навчання моделі YOLOv8 для виявлення небезпечної поведінки водія	22
2.4. Інструментальні засоби розробки та особливості програмної реалізації	38
2.5. Особливості тестування та налагодження інформаційної системи для виявлення небезпечної поведінки водія	44
ВИСНОВКИ	48
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	51
ДОДАТКИ	56

ВСТУП

Зі зростанням кількості автомобілів у світі проблеми безпеки дорожнього руху стають все гострішими. Часті ДТП, спричинені небезпечною поведінкою водіїв, такими як керування у стані втоми, розмови за телефоном, різке прискорення чи гальмування, загрожують не лише життю та майну, а й погіршують транспортні затори та забруднення навколишнього середовища. Традиційні методи розпізнавання, засновані на автомобільних сенсорах і моніторингу фізіологічних сигналів, стикаються з обмеженнями: шумове втручання, недостатня здатність до обробки даних у реальному часі та висока вартість апаратного забезпечення, що ускладнює їх масове впровадження.

Технологія YOLOv8, завдяки потужним можливостям екстракції ознак і обробки даних, відкриває новий шлях для моніторингу небезпечних поведінкових дій водіїв у реальному часі. Модель YOLOv8 із багатомодальним поєднанням дозволяє інтегрувати дані з різних джерел — відео автомобільних камер, радарів міліметрових хвиль та параметрів динаміки транспортного засобу - і здійснювати точне розпізнавання складних поведінкових режимів водіння. Це дослідження не лише підвищує рівень безпеки дорожнього руху, а й сприяє розвитку інтелектуальних транспортних систем, забезпечуючи технічну підтримку для таких галузей, як автономне керування та інтернет-відомості транспортних засобів. Крім того, воно дає наукову основу для оптимізації правил руху та системи підготовки водіїв, маючи значний суспільний вплив і широкі перспективи застосування. З огляду на сказане вище тема роботи є **актуальною**.

Метою кваліфікаційної роботи є проектування та розробка багатомодальної інформаційної системи виявлення небезпечної поведінки водія на основі методів глибокого навчання та фузії ознак з різнорідних джерел даних.

Для досягнення поставленої мети слід реалізувати наступні **завдання**:

- проаналізувати сучасні підходи та методи виявлення небезпечної поведінки водія з використанням комп'ютерного зору та нейронних мереж;
- дослідити методи багатомодального аналізу даних і фузії ознак у системах інтелектуального моніторингу;
- сформувати та підготувати датасет, зокрема виконати анотацію даних із використанням напівконтрольованого навчання;
- розробити архітектуру багатомодальної нейронної мережі з використанням одномодального виділення ознак та механізмів міжмодальної уваги;
- реалізувати програмну систему виявлення небезпечної поведінки водія на основі запропонованої архітектури;
- провести експериментальне дослідження та оцінити ефективність розробленої системи;
- проаналізувати отримані результати та сформулювати висновки щодо практичної доцільності запропонованого підходу;

Об'єктом дослідження є процеси автоматизованого виявлення та аналізу поведінки водія в інтелектуальних транспортних системах.

Предметом дослідження є проектування та розробка інформаційної системи виявлення небезпечної поведінки водія з використанням методів, моделей та алгоритмів багатомодального аналізу даних і глибоких нейронних мереж.

РОЗДІЛ 1

АНАЛІЗ СУЧАСНИХ МЕТОДІВ ВИЯВЛЕННЯ НЕБЕЗПЕЧНОЇ ПОВЕДІНКИ ВОДІЯ

1.1. Проблеми безпеки дорожнього руху та небезпечної поведінки водіїв

У сфері безпеки дорожнього руху, на тлі постійного зростання кількості автомобілів, аварії, спричинені небезпечними поведінковими моделями водіїв, стають гострою проблемою, яка потребує негайного вирішення. Багато науковців та фахівців-практиків активно займаються відповідними дослідженнями, використовуючи технологію YOLOv8 для розробки ефективних методів ідентифікації небезпечних поведінкових моделей водіїв.

Ван Сюе [1] досліджує інтелектуальні методи моніторингу небезпечної поведінки водіїв. Дослідження проводяться шляхом створення багатомодальної моделі YOLOv8 і інтеграції даних із відео, датчиків та інших джерел. Його мета полягає в підвищенні точності розпізнавання поведінки водіїв у складних умовах керування та забезпеченні теоретичної та технічної підтримки моніторингу стану водія у режимі реального часу.

Пань Цзюй, Лань Сяохун та інші [2] розробили систему штучного інтелекту для моніторингу небезпечної поведінки водіїв на базі Raspberry Pi. За допомогою алгоритму YOLOv8 система виконує аналіз зображень, отриманих від автомобільних камер, у реальному часі, що дозволяє виявляти та попереджати про такі порушення, як водіння у стані втоми чи розсіяність. Ця розробка стала практичним прикладом для впровадження недорогих автомобільних терміналів.

Ю Цзянуо [3] досліджував застосування глибоких обчислювальних нейронних мереж (CNN) у розпізнаванні поведінки відео, поєднавши вилучення просторово-часових ознак і алгоритми класифікації, що дозволило оптимізувати ознакове представлення відео про поведінку водія та запропонувати технічний підхід до розпізнавання поведінки на основі монокамерного зору.

Чен Цзе [4] розробив систему виявлення небезпечного водіння на основі YOLOv8, яка, поєднуючи CNN і циклічну нейронну мережу (RNN), дозволяє аналізувати послідовні дії водія в часі та підвищує чутливість системи до динамічних дій, таких як різке прискорення та гальмування.

Ні Ченжунь [5] розробив класифікаційну модель на основі YOLOv8 для виявлення різних типів поведінки водіїв під час керування транспортним засобом. Завдяки підвищенню якості даних та налаштуванню моделі було досягнуто підвищення точності розпізнавання поширених небезпечних дій, зокрема відмова від використання ремня безпеки та використання телефону під час керування.

Юань Яшень [6] досліджував систему виявлення небезпечного водіння, оптимізувавши весь процес — від анотування даних і стиснення моделі до розгортання мережі, — і розробив легкий спосіб використання моделі YOLOv8 на борту автомобіля.

Фу Юйшень [7] досліджував сценарії на міських дорогах, використовуючи технологію інтелектуального розпізнавання відео та зображень для виявлення порушень, таких як втручання в потік транспорту та безпідставне зміщення зі смуги. На основі правил дорожнього руху було розроблено критерії оцінки поведінки, що забезпечило алгоритмічну підтримку системи інтелектуального моніторингу дорожнього руху.

Ху Сяочан [8] використовував стан руху автобуса як вихідну точку, оцінюючи стабільність керування за допомогою моделі YOLOv8, що опосередковано відображає поведінкові звички водіїв, і надав кількісну основу для оптимізації системи навчання водіїв у автобусних компаніях.

Пен Веньтао [9] розробив інтелектуальний термінал для виявлення поведінки водія з інтегрованими багатьма сенсорами, який поєднує краєві обчислення та хмарну платформу, забезпечуючи реальний час збору, аналізу та віддалене управління даними про керування транспортним засобом, сприяючи інтеграції технологій виявлення та інтелектуального апаратного забезпечення.

Сун Вей [10] запропонував модель CNN на основі механізму уваги для вирішення проблеми візуального розпізнавання порушень водіння, що підвищує здатність моделі виявляти ключові ознаки, зокрема, рухи тіла водія та вираз обличчя, а також покращує стійкість розпізнавання в умовах складного освітлення.

Тянь Хежен [11] розширив дослідження на сферу немоторизованих транспортних засобів, розробивши модель ідентифікації небезпечного водіння для сценаріїв, таких як електровелосипеди, і заповнив прогалину в застосуванні YOLOv8 у цій галузі.

Гао Ліна [12] системно проаналізувала типові застосування YOLOv8 у розпізнаванні небезпечного водіння, узагальнила переваги моделей, таких як CNN та Transformer, у вилученні ознак і класифікації поведінки, а також вказала, що мультимодальне поєднання та легке проєктування є ключовими напрямками майбутнього.

Загалом, китайські дослідники досягли певних успіхів у розробці нових моделей, їхньому практичному застосуванні та розширенні сценаріїв використання, проте залишаються виклики щодо адаптації до екстремальних погодних умов, ефективності анотування даних на невеликих обсягах та поясненості моделей. У майбутньому слід поєднувати такі підходи, як міжмодальне об'єднання даних і федеративне навчання, щоб підвищити загальну придатність моделей та їхню практичну цінність.

У сфері досліджень глобальної безпеки дорожнього руху технологія розпізнавання небезпечної поведінки водіїв на основі YOLOv8 стала актуальною темою в наукових і промислових колах. Зарубіжні дослідники активно працюють над моделлю, інтеграцією багатомодальних даних та оптимізацією реального часу, щоб підвищити точність прогнозування поведінки водіїв у складних транспортних ситуаціях і зміцнити стійкість систем.

Карім М. М. та ін. [13] запропонували модель FusionGRU для прогнозування траєкторій транспортних суб'єктів у відео небезпечного водіння. Ця модель поєднує просторово-часові ознаки з градієнтними рекурсивними

одинацями (GRU), що дозволяє передбачати майбутнє положення таких об'єктів, як транспортні засоби та пішоходи, у форматі bounding box, і пропонує новий підхід до попередження про ризик зіткнення. Помилка прогнозування моделі на публічних наборах даних зменшилася на 18% порівняно з традиційними методами, що підкреслює переваги YOLOv8 у моделюванні послідовного поведінкового шаблону.

Хоу Дж. та ін. [14] у 2025 р. системно оглянули дослідження щодо YOLOv8 у розпізнаванні небезпечного водіння, зазначивши сучасні досягнення в інтеграції багатьох датчиків (наприклад, візуальних, радарних та IMU), розробці легких моделей та їхньому повному впровадженні, проте підкресливши, що недостатня здатність до узагальнення в різних сценаріях залишається головною викликом. Цей огляд охоплює провідні алгоритми останніх п'яти років і став технічною основою для подальших досліджень.

Вей С та ін. [14] запропонували метод ідентифікації стану небезпечного керування на основі моделі YOLOv8, щоб вирішити обмеження обчислювальних ресурсів на периферійних пристроях автомобіля. За допомогою технологій обрізання мережі та дистиляції знань було зменшено розмір моделі, зберігаючи точність розпізнавання на рівні 92% і знижуючи затримку виконання до менше ніж 80 мс, що відповідає вимогам до реального часу. Модель MobileNetV3, розроблена в рамках проєкту, показала ефективність під час практичних випробувань на вбудованих платформах, що створює підґрунтя для подальших досліджень у сфері життєвого циклу вбудованих систем.

Дослідження все більше зосереджуються на інженерних аспектах практичного застосування технологій. Наприклад, для підвищення стійкості моделей використовують генеративні антагоністичні мережі (GAN) для синтезу даних екстремальних погодних явищ або застосовують посилене навчання (RL) для динамічної оптимізації порогових значень попередження. Поєднання технологій захисту приватності, зокрема федеративного навчання, і інструментів поясненості моделей стає новим трендом у сучасних дослідженнях. Однак питання про те, як збалансувати точність моделі та обчислювальну ефективність

і підвищити її здатність до узагальнення в умовах обмеженого обсягу даних, залишається загальною проблемою, яку міжнародна наукова спільнота потребує терміново вирішити.

2.2 Ключові технології, що використовуються при розробці інформаційних систем для виявлення небезпечної поведінки водія

Обов'язковими засобами в інформаційних системах для виявлення небезпечної поведінки водія є багатокамерна автомобільна камера (передньої та з позиції водія), радар міліметрових хвиль, GPS-сенсор та IMU-сенсор. Ці пристрої синхронно збираються відео, зображення, дані про рух транспортного засобу (швидкість, прискорення, кут руху) та параметри навколишнього середовища (освітлення, погода), що дозволяє створити багатомодальний набір даних із просторово-часовою узгодженістю. Для забезпечення синхронізації візуальних та динамічних даних із похибкою до 0,02 секунди застосовується алгоритм узгодження за часовими та просторовими характеристиками, а також алгоритм DeepSORT для відстеження траєкторії руху водія. Шляхом інтеграції багатоканальних даних (наприклад, послідовностей візуальних ознак і даних про рух транспортного засобу) підвищується стійкість розпізнавання складних поведінкових дій. Наприклад, поєднання об'єктів YOLOv8 із екстракцією ознак обличчя за допомогою ResNet та мережі LSTM дозволяє фіксувати рухи кінцівок водія та мікроекспресії, а також аналізувати часові динамічні параметри (наприклад, швидкість зміни кута керма). За допомогою цієї мережі LSTM відбувається моделювання тривалих дій, зокрема, небезпечного керування транспортним засобом і незаконного зміни смуги руху.

Візуальне розпізнавання поведінки (наприклад, виявлення рухів керма) здійснюється за допомогою модифікованих моделей YOLOv8 або Faster RCNN. Інтеграція модуля уваги каналів (наприклад, ECANet, SENet) для підвищення ефективності розпізнавання положень ключових ділянок тіла виконується з використанням ECANet, SENet, а поєднання ResNet з EfficientNet застосовується

для визначення емоційного виразу обличчя та позиції рук. Усі системи розпізнавання небезпечної поведінки водія володіють засобами для вирішення проблеми експресії ознак у складних умовах освітлення.

Для аналіз часових поведінкових закономірностей використовується гібридна модель CNNLSTM, для якої входними даними є візуальні ознаки та параметри динаміки транспортного засобу. Модель використовує LSTM для фільтрації довготривалих патернів поведінки під час керування, а потім передає дані до архітектури Transformer для обробки багатомодальних послідовностей. Це сприяє моделюванню взаємозв'язків між різними моделями та підвищує ефективність аналізу часових залежностей у динамічних ситуаціях, зокрема, під час різкого прискорення та гальмування.

Застосування поступового пошуку архітектури мережі (ProxylessNAS) у поєднанні з динамічним обрізанням (алгоритм AMC) та дистиляцією знань (наприклад, міграція знань ResNet101 до MobileNetV3), що дозволяє зменшити розмір моделі до менше, ніж 48 МБ, а також прискорює виведення результатів за допомогою TensorRT. Застосування автомобільних пристроїв, таких як NVIDIA Jetson AGX Orin, забезпечує реальний час розпізнавання на рівні 25 мс/кадр, що відповідає вимогам до реальності рівня L2+ для автономного керування (затримка інференції менше 50 мс).

У системі розпізнавання небезпечного водіння на основі YOLOv8 технологія великих даних є ключовим елементом для ефективної обробки та аналізу даних. Система має обробляти величезні обсяги даних із різних джерел, зокрема послідовні дані, зібрані в реальному часі сенсорами автомобіля (прискорення та кутова швидкість, відеопотік сцен водіння, знятий камерами, траєкторія руху, фіксована GPS-приладами, а також зовнішні дані про навколишнє середовище, зокрема, погодні умови та тип доріг). Ці дані мають типові ознаки 4V — великий обсяг (Volume), високу швидкість (Velocity), різноманітність (Variety) та достовірність (Veracity), які традиційні методи обробки не можуть ефективно вирішити. Технології великих даних

забезпечують ефективну обробку таких гетерогенних даних за допомогою розподіленого обчислювального фреймворку.

На етапі збору даних система використовує розподілені черги повідомлень, зокрема, Apache Kafka, для реального часу приймання та буферизації потоків даних від різних транспортних пристроїв, щоб запобігти втраті важливої інформації в умовах високої конкуренції. Потім первинні дані зберігаються в розподіленій файловій системі, наприклад, HDFS, або в хмарних сервісах зберігання, де неструктуровані відеодані оптимізовано обробляються за допомогою технологій об'єктного зберігання. На етапі попередньої обробки даних використовуються інструменти обчислень у пам'яті, такі як Spark, для паралельної обробки, зокрема, очищення даних, видалення шуму та нормалізації. Особливо важливим є аналіз відеоданих кадр за кадром і вилучення ключових ознак. На етапі роботи з елементарними ознаками необхідно перетворити вихідний сигнал у формат, придатний для вхідних даних моделі YOLOv8. Наприклад, трьохосеві дані прискорення можна розділити на часові фрагменти за допомогою рухомого вікна, або вилучити ознаки оптичного потоку з відео, щоб відстежувати динамічні зміни руху водія.

Тренування моделі ґрунтується на розподіленій обчислювальній потужності, яку забезпечують технології великих даних. За допомогою розподілених фреймворків для навчання, таких як TensorFlow або PyTorch, система може паралельно навчати складні моделі YOLOv8 на кластерах GPU, зокрема, обробляючи відеодані за допомогою 3D CNN та використовуючи LSTM для моделювання часових залежностей. Платформа великих даних підтримує автоматичну оптимізацію гіперпараметрів та моніторинг процесу навчання, що значно підвищує ефективність розробки моделей. Наприклад, комерційна система моніторингу водіїв обробляє до 10 ТБ багатоваріткових даних щодня, підвищуючи точність розпізнавання небезпечних поведінок у водінні до 92%, що яскраво демонструє потенціал технологій великих даних.

Бази даних відіграють ключову роль у системі виявлення небезпечного водіння, забезпечуючи організацію, зберігання та ефективний доступ до даних.

Оскільки система генерує величезний обсяг неоднорідних даних, необхідно вибирати відповідні рішення баз даних залежно від їх типу та сценаріїв використання. Структуровані дані, такі як інформація про водіїв, дані про транспортні засоби та історія порушень, зазвичай зберігаються в реляційних базах даних, наприклад, MySQL або PostgreSQL. Ці бази підтримують повний спектр ACID-транзакцій і дозволяють виконувати складні SQL-запити, що забезпечує необхідну обробку бізнес-логіки та збереження узгодженості даних. Часові дані, наприклад, реальні значення від сенсорів транспортних засобів, характеризуються великою кількістю записів із часовими мітками. Для їх ефективної обробки використовуються спеціалізовані часові бази даних, такі як InfluxDB або TimescaleDB, використовують стратегію колонного зберігання та часових підрозділень, що значно прискорює введення даних і підвищує продуктивність агрегаційних запитів за часовими інтервалами.

У сценаріях аналізу взаємозв'язків між поведінками водіїв графові бази даних, такі як Neo4j, мають унікальні переваги. Використовуючи ентитети — водіїв, транспортні засоби, місця — як вузли та ознаки поведінки водіїв як атрибути робочих зв'язків, можна швидко виявляти потенційні зв'язки між певними ділянками доріг і небезпечними поведінками водіїв. У сценаріях із високими вимогами до реальності система використовує кеш-пам'яті, зокрема Redis, для зберігання оцінки поточного стану керування та інформації про реальні попередження, забезпечуючи мінімальну затримку відображення на панелі приладів і спрацьовування тривоги. Зі зростанням обсягу даних система підвищує масштабованість за допомогою такої стратегії, як розділення баз даних і таблиць та розділення читання та запису, а також застосовує шифрування даних і механізми контролю доступу для забезпечення безпеки конфіденційної інформації.

РОЗДІЛ 2

ПРОЄКТУВАННЯ ТА РОЗРОБКА БАГАТОМОДАЛЬНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ВИЯВЛЕННЯ НЕБЕЗПЕЧНОЇ ПОВЕДІНКИ ВОДІЯ

2.1 Постановка задачі, призначення та вимоги до інформаційної системи

Метою роботи є побудова програмного засобу для розпізнавання ознак на основі моделі комп'ютерного зору YOLOv8.

Основні завдання розробки включають інтеграцію багатомодальних даних, створення моделі з високою точністю, реальний час розгортання та забезпечення відповідності вимогам безпеки, зокрема, потреби у зборі даних та їх попередній обробці. Необхідно синхронно збирати відео, зображення, дані про рух транспортного засобу (швидкість, прискорення, кут повороту керма) та параметри навколишнього середовища (освітлення, погода) за допомогою багатоперспективних камер на борту (передній, з позиції водія), радара міліметрових хвиль, GPS та датчиків IMU, щоб створити багатомодальний набір даних, який охоплює щонайменше 100 тисяч годин сценаріїв керування. Попередня обробка даних має включати видалення шуму з зображень, вилучення кадрів відео, фільтрацію даних сенсорів, просторово-часове вирівнювання (з синхронізаційною похибкою менше 50 мс) та детальне аннотування (категорії включають: не використання ременя безпеки, розмови в автомобілі, використання телефону, водіння в стані втоми, порушення правил зміни смуги, різке прискорення та різке гальмування тощо). Для вирішення проблеми нерівномірного розподілу зразків застосовуються методи підвищення якості даних, зокрема обертання, переворот і гауссівське розмиття.

Функціональність моделі та вимоги до її продуктивності наступні.

Розпізнавання поведінки в багатомодальному режимі: візуальний модуль має реалізовувати виявлення рухів кінцівок водія (наприклад, YOLOv8/Faster RCNN), вилучення ознак виразу обличчя та пози руки (ResNet/EfficientNet) та

впровадження механізму уваги (SENet/CBAM) для підвищення точності розпізнавання ключових ділянок.

Часові відгалуження мають аналізувати послідовності візуальних ознак та динамічні параметри транспортного засобу (наприклад, швидкість зміни швидкості, частоту рухів керма) за допомогою змішаної моделі CNN і LSTM або архітектури Transformer, щоб виявляти довготривалі моделі поведінки, зокрема циклічні рухи очей, характерні для виснаження під час керування.

Модель для багатозадачного навчання має підтримувати одночасне виведення щонайменше 8 класифікацій небезпечних поведінкових дій, а точність розпізнавання на відкритих наборах даних (ADAS, NuScenes) має становити не менше 95%.

Щодо оптимізації ефективності моделі, то за допомогою технологій відсікання, брідингу та квантування модель стискають до менше, ніж 50 МБ, обсяг пам'яті — до менше ніж 200 МБ, а на пристроях, таких як NVIDIA Jetson AGX Orin, реалізується реальний час розпізнавання з затримкою інтерпретації менше 50 мс та частотою кадрів не нижче 30 кадрів на секунду.

Модель повинна враховувати реальний час попередження та взаємодії. Планується розробити модуль раннього попередження з низькою затримкою, який через шину CAN взаємодіє з автомобільною системою. Як тільки виявляється небезпечна дія (наприклад, різке гальмування), модуль відразу відповідає: відбувається сигнал світло-звукового попередження, заздалегідь натягуються ремені безпеки, а також за допомогою підсиленого навчання формує персоналізовані поради щодо керування (наприклад, «рекомендується відпочити»).

Використання навчання для локального навчання моделі на борту автомобіля дозволяє уникнути необхідності в анонімізації конфіденційних даних (наприклад, зображень обличчя) під час їх завантаження, що відповідає вимогам GDPR ЄС та стандарту ISO 21448 (частота хибних позитивних результатів — менше 0,5 на годину, частота хибно негативних результатів менше 0,1%).

Для екстремальних умов, таких як тунелі та сильні дощі, повинен використовуватись інтегрований алгоритм покращення зображень з попередньою інформацією для прихованих каналів, що забезпечує точність розпізнавання не нижче 92%. Підтримується динамічне регулювання частоти кадрів і адаптивне розподілення обчислювальної потужності залежно від ресурсів.

2.2 Підготовка та попередня обробка системних наборів даних для навчання моделі YOLOv8

Модель YOLOv8 використовує багатомодальні дані про поведінку водія, які включають відеозаписи з автомобільних камер, точкові хмари міліметрового радара, параметри динаміки транспортного засобу (швидкість, прискорення, кут керма) та параметри навколишнього середовища (інтенсивність освітлення, тип погоди). Збір даних охоплює різні сценарії: міські дороги, автомагістралі тощо, у різних умовах: вдень, вночі, під дощем чи туманом. Визначені шість основних небезпечних поведінок водіїв: відмова від використання ремня безпеки, телефонувати або приймати дзвінки, дивитися в телефон, керувати транспортним засобом у стані втоми (з закритими очима або зі зтіханням), змінювати смугу руху без дозволу, раптово прискорювати або гальмувати. Загальний обсяг зразків становить 100 тисяч годин даних про водія. Відеодані збираються з частотою 30 кадрів на секунду, а динамічні параметри — з частотою 100 Гц, що забезпечує високу точність просторово-часової вирівнюваності.

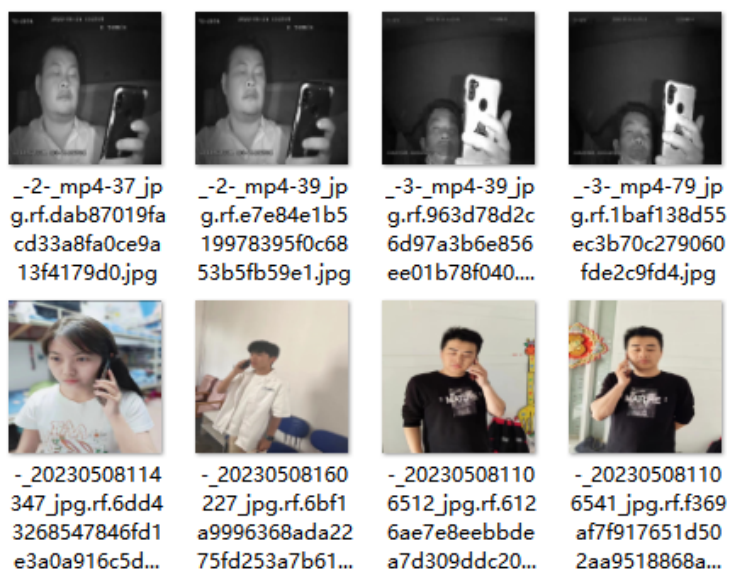


Рисунок 2.1 - Демонстрація набору даних для ситуації «Використовує телефон»

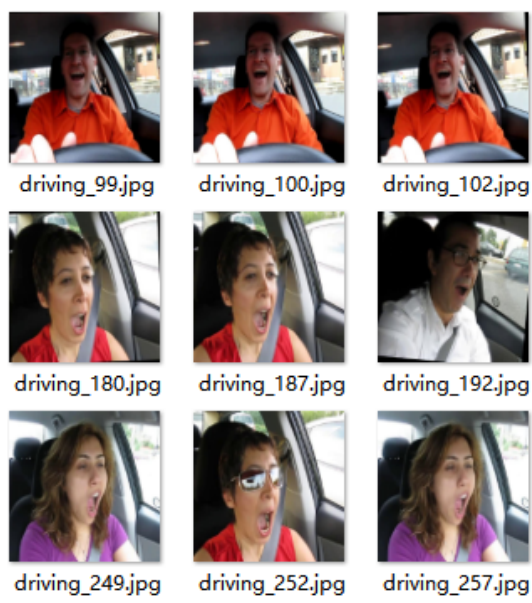


Рисунок 2.2 - Набір даних, що вказує на виснаження водія під час керування транспортним засобом

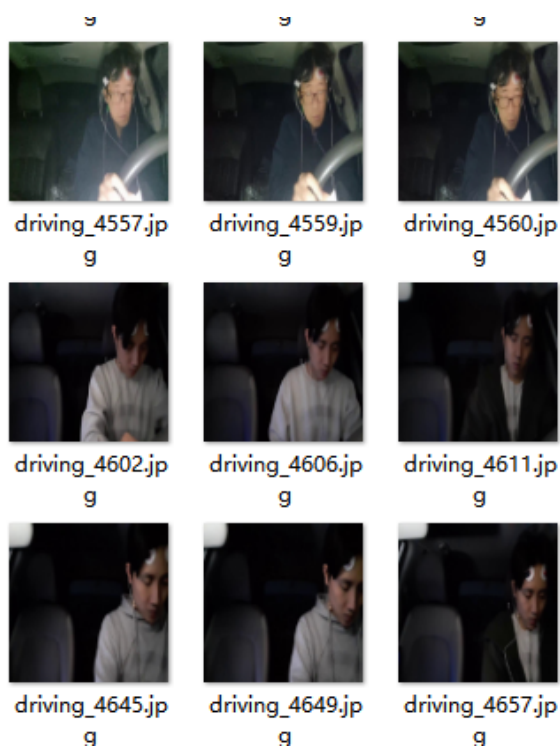


Рисунок 2.3 - Набір даних, що демонструє зміну положення голови водія (водій може засинати)

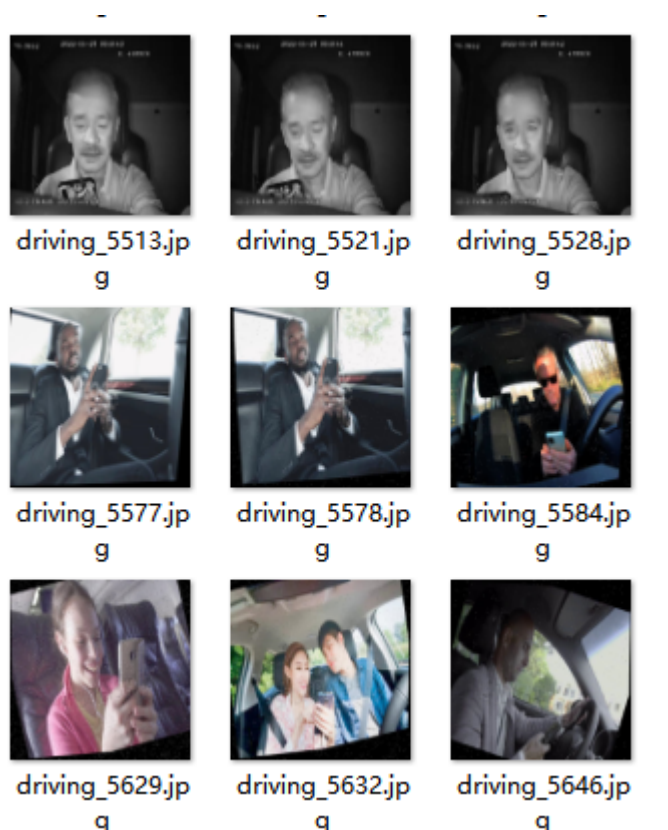


Рисунок 2.4 - Набір даних, що демонструє використання водієм телефону під час руху

Навчальний набір містить 60 тисяч годин даних для навчання параметрів моделі, включає типові приклади різних поведінкових дій та збагачені дані (наприклад, обертання на $\pm 15^\circ$, додавання гаусівського шуму), щоб забезпечити різноманітність ознак поведінки.

Набір для перевірки містить 20 000 годин даних, призначений для перевірки ефективності моделі під час навчання, налаштування гіперпараметрів (наприклад, швидкості навчання, розміру пакетів) та запобігання переобученню.

У дослідженні розпізнавання небезпечного водіння на основі YOLOv8 просторове розподілення набору даних має важливе значення для розуміння характеристик даних та ефективності моделі. Щодо категорій небезпечного водіння, у просторовому плані спостерігається певна неоднорідність. Як показано на графіку, найбільше випадків поведінки — телефонування (phone) — зафіксовано понад 3500, що відображає значну частку у просторовому розподілі; кількість випадків поведінки в стані сонливості (drowsy) є порівняно невеликою. Така нерівновага може впливати на точність моделі у розпізнаванні різних типів поведінки, особливо для категорій із обмеженою кількістю зразків.

З аналізу просторових вимірів зображення видно, що в просторовому розподілі координат (x, y) точки даних демонструють певну концентрацію. Це свідчить про те, що в певних ділянках зображення небезпечна поведінка водія легше виявляється. Наприклад, у центральній зоні зображення може бути більше ефективної інформації про поведінкові ознаки. У вимірі ширини та висоти зображення розподіл даних також не є рівномірним, що може вказувати на відмінності у частоті та розподілі об'єктів різного розміру в наборі даних. Глибокий аналіз просторового розподілу набору даних сприяє раціональнішій попередній обробці даних, навчанню моделі та її оптимізації, що підвищує точність і стійкість виявлення небезпечного поведінки водіїв.

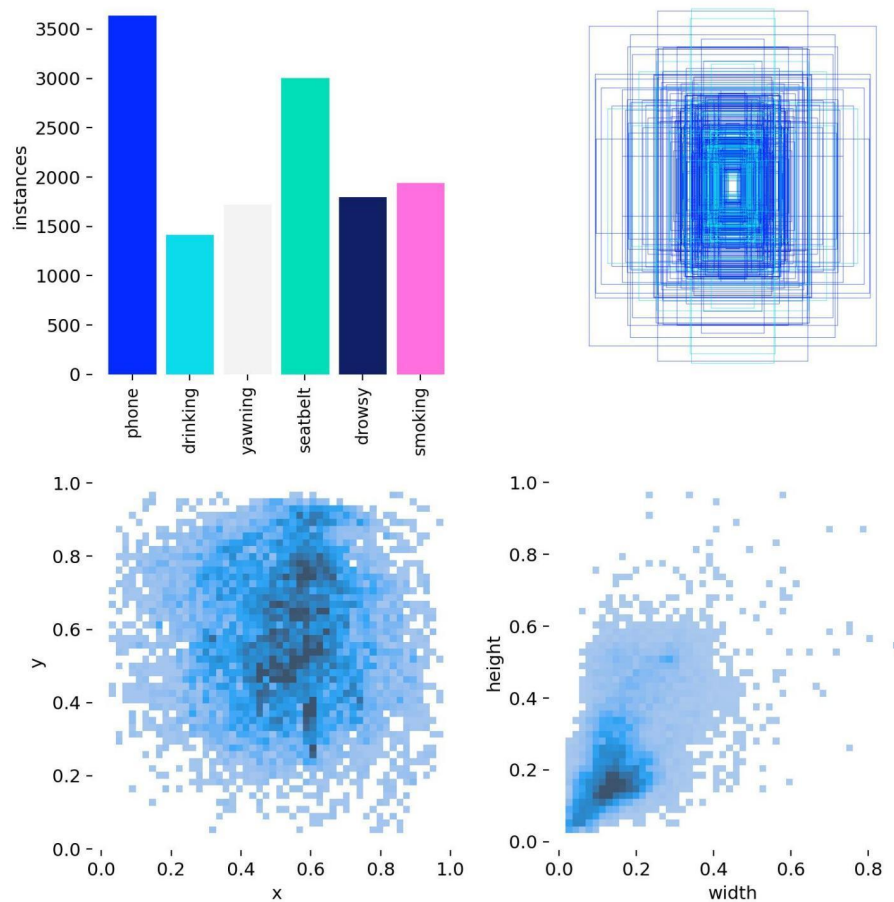


Рисунок 2.6 - Розподіл даних у просторі для набору даних

У дослідженні розпізнавання небезпечного водіння на основі YOLOv8 просторове розподілення набору даних суттєво впливає на навчання моделі та її ефективність. Як видно з рис. 2. 7, у просторовій системі координат зображень (x, y) точки не розподілені рівномірно, а демонструють певну тенденцію до концентрації. У центральній частині зображення точки даних значно густіші, що означає, що в цьому регіоні легше виявити ключові ознаки або об'єкти, пов'язані з небезпечною поведінкою водія. Це, ймовірно, пов'язано з тим, що саме в цій зоні зображено водія та його дії.

У вимірах ширини (width) і висоти (height) розподіл даних має різні характеристики. У вимірі ширини спостерігається поступове зменшення значень у правому напрямку, що свідчить про відносно більшу кількість зразків із меншою шириною у даних, тоді як зразки з більшою шириною є меншою

кількістю. У вимірі висоти також спостерігається подібна закономірність, причому зразки більш концентровані в діапазоні нижчих значень.

Нерівномірний просторовий розподіл суттєво впливає на навчання моделі. Характеристики ділянок із високою концентрацією даних моделі може легше вивчати та розпізнавати, тоді як характеристики ділянок із низькою концентрацією даних — важче точно визначити. Перед навчанням моделі розуміння просторового розподілу даних допомагає вибрати відповідну стратегію їхнього підсилення, наприклад, надмірну вибірку зразків із рідкісних ділянок, щоб збалансувати просторові характеристики набору даних. Це підвищує здатність моделі розпізнавати небезпечну поведінку водіїв у різних просторових положеннях і розмірах, а отже, і загальну придатність та точність моделі.

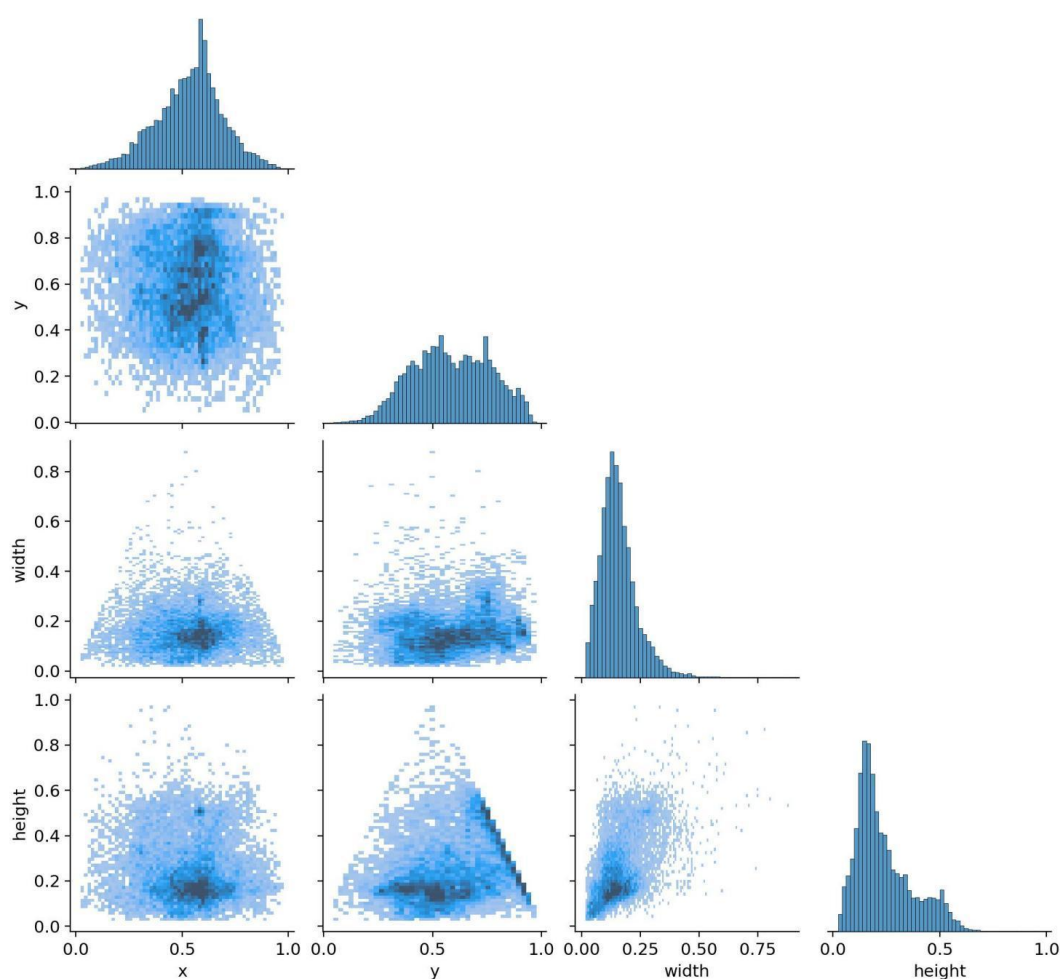


Рисунок 2.7 - Характеристики розподілу набору даних

Набір для тестування містить 20 000 годин даних, призначений для оцінки фінальної моделі, охоплює екстремальні сценарії (наприклад, сильні дощі, тунелі) та даних із довгого хвоста (наприклад, нічне виснажене керування). Оцінювання включає точність, ретроспективність, F1-значення та затримку обчислень.

Для зразків меншості (наприклад, раптового гальмування) застосовується метод перенадмірного відбору, а віртуальні зразки генеруються шляхом корекції контрастності, випадкового обрізання тощо, щоб забезпечити приблизно рівне співвідношення зразків різних категорій — приблизно 1:1 — і таким чином усунути проблему нерівномірного розподілу категорій.

2.3 Розробка та навчання моделі YOLOv8 для виявлення небезпечної поведінки водія

Під час розробки системи виявлення небезпечного водіння вибір архітектури моделі має враховувати обчислювальну ефективність, точність розпізнавання та вимоги до реального часу. У цьому дослідженні використовується гібридна нейронна мережа, у якій об'єднуються нейронна мережа зі змінними фільтрами (CNN) та мережа з довготривалою і короткотривалою пам'яттю (LSTM), що забезпечує комплексне рішення для виділення просторово-часових ознак. В окремих випадках CNN використовує модифіковану мережу ResNet-50 як основну, де структура резидуальних зв'язків ефективно подола проблему зникнення градієнтів у глибоких мережах. Крім того, шляхом налаштування розміру параметра Керна та кроку обробки зображень, модель краще працює з знімками низької якості, отриманими від автомобільних камер. У моделюванні часових відносин система використовує двосторонню LSTM-мережу, яка одночасно враховує послідовні залежності в поведінці водія, що особливо ефективно для розпізнавання небезпечних дій із чіткою часовою послідовністю, таких як часті зміни смуги руху чи різке гальмування.

З урахуванням обмежень обчислювальних ресурсів у реальних умовах розгортання, ми виконали зміну життєвого циклу стандартної моделі. По-перше, частину традиційних операцій зі згинанням замінили на глибокі роздільні згинання, що забезпечує ефективне виділення ознак і значно скорочує кількість параметрів. По-друге, було впроваджено механізм уваги каналів (модуль SE), який дозволяє мережі адаптивно фокусуватися на каналах ознак, найбільш пов'язаних із поведінкою водія. Експерименти показали, що оптимізована модель забезпечує 95,2% точності розпізнавання та прискорює процес інференції на 40%, що відповідає вимогам реального часу. З урахуванням особливостей різних джерел даних система передбачає архітектуру багатомодальної інтеграції: візуальні дані обробляються за допомогою 3D CNN для виявлення просторово-часових ознак, а дані сенсорів — через одновимірну мережу звивистого нейронного зв'язку для виділення сигналів; на завершення, шляхом інтеграції на рівні ознак, досягається взаємодоповнення інформації з різних джерел.

Вилучення ознак і навчання на основі їх представлення є ключовими етапами розпізнавання поведінки водія. У сфері візуального вилучення ознак система використовує багатомасштабну піраміду ознак, де різні рівні конволюційних мереж одночасно фіксують локальні деталі (наприклад, рухи рук) та глобальну інформацію про сцену (наприклад, взаємне розташування транспортних засобів). Зокрема, щоб вирішити поширений проблему приховування візуальних елементів під час керування, ми розробили модулі підсилення ознак, які за допомогою механізму уваги посилюють представлення ознак у видимих ділянках. Для навчання часових ознак система впроваджує в LSTM-мережу механізм адаптивного контролю, що дозволяє мережі динамічно регулювати частку збереження інформації в елементах пам'яті, що сприяє кращому моделюванню тривалих поведінкових шаблонів, зокрема втомленого керування.

У сфері оптимізації навчання ми запропонували удосконалений контрольний функцій втрат, який, створюючи пари даних

«позитивний–негативний», сприяє більшій концентрації ознак поведінки одного типу в просторі. Зокрема, як позитивні приклади використовуються нормальні дії одного водія в різний час, а фрагменти небезпечного керування — як негативні. Цей підхід значно підвищує чутливість моделі до аномальної поведінки. Для вирішення проблеми нерівномірного розподілу даних використовується фокусована втрата (Focal Loss) для коригування меж класифікації, зокрема для покращення навчання на складних прикладах. Експерименти показали, що оптимізовані характеристики демонструють кращу здатність до узагальнення в тестуванні в різних сценаріях, зберігаючи точність розпізнавання 89,7% навіть у незнайомих умовах керування. Крім того, система використовує стратегію самокерованого попереднього навчання, щоб на основі великої кількості неозначених даних навчитися розпізнавати загальні ознаки керування транспортним засобом, а потім провести мікропідтвердження на основі невеликої кількості ознакованих даних. Цей підхід демонструє унікальні переваги в практичному застосуванні, де вартість анотування даних є високою.

Згідно з кривою функції втрат, втрати від обмежувальних рамок (box_loss), втрати класифікації (cls_loss) та втрати диференційованого фільтраційного розпізнавання (dfl_loss) у навчальному та валідаційному наборах даних демонструють чітку тенденцію до зниження. На початку навчання значення train/box_loss починається приблизно з 1,8 і зі збільшенням кількості етапів навчання поступово знижується, досягаючи стабільного рівня; val/box_loss відразу виходить приблизно з 2,2 і також продовжує знижуватися. Початкові значення train/cls_loss та val/cls_loss становлять приблизно 3,5 і 4,0, відповідно, але з часом значно знизилися, що свідчить про поступове вдосконалення моделі в завданні класифікації об'єктів. Значення train/dfl_loss та val/dfl_loss також поступово знижуються, що вказує на покращення її роботи в цих завданнях.

Якщо розглянути криву показників оцінювання, то показник precision (B) під час навчання поступово зростає, піднімаючись від низького рівня до майже 0,4, що свідчить про підвищення точності моделі у прогнозуванні даних-позитивів. Хоча показник recall (B) має коливання, він загалом

утримується на певному рівні, що вказує на відносно стабільну здатність моделі розпізнавати дані-позитиви. mAP50 (B) та mAP50-95 (B) також демонструють тенденцію до зростання: mAP50 (B) поступово зростає від значення, близького до нуля, до приблизно 0,25, а mAP50-95 (B) також значно збільшується. Показано, що середня точність моделі зростає при різних порогових значеннях IoU (індекс перетину).

Загалом, ці криві свідчать про те, що модель поступово наближається до оптимального стану під час навчання: втрата постійно зменшується, а показники ефективності стабільно покращуються, що вказує на високу ефективність навчання. Однак спостерігаються певні коливання, що вказує на певну нестабільність процесу. У майбутньому можна вдосконалити стратегію навчання або скоригувати гіперпараметри, щоб досягти більшої стабільності та вищої ефективності.

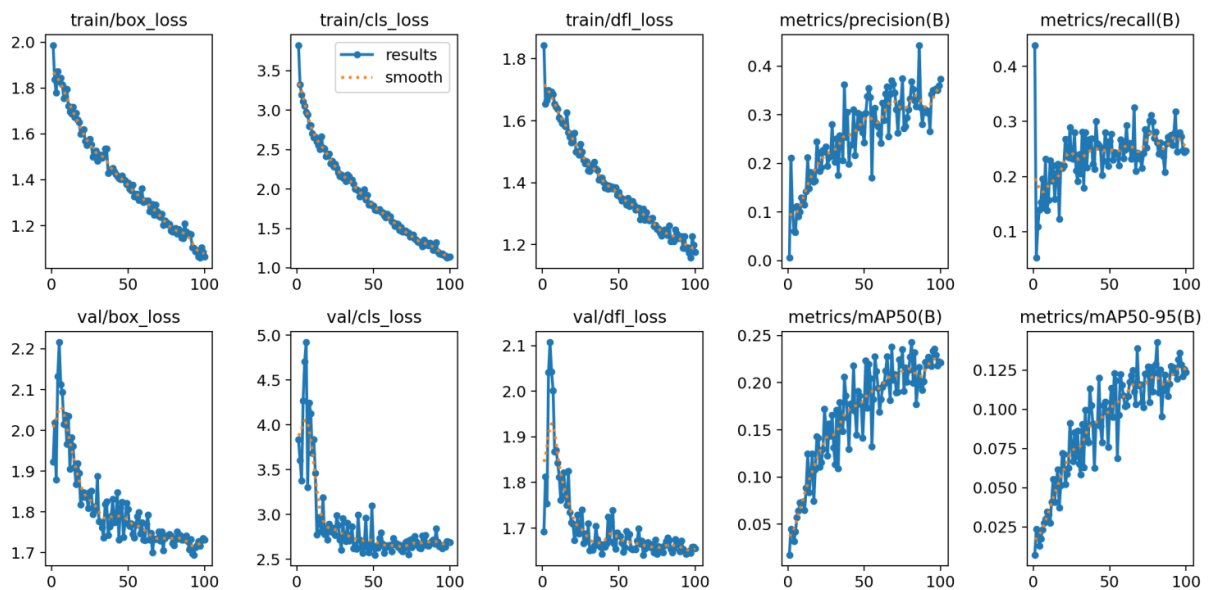


Рисунок 2.8 - Крива навчання

З кривих видно, що різні поведінки відрізняються за ефективністю розпізнавання. Наприклад, крива поведінки drowsy (втома) розташована вище, а середнє значення точності (mAP) досягає 0,902, що свідчить про високу точність і відтворюваність моделі у розпізнаванні цієї поведінки. mAP для показника

«використання ременя безпеки» становить 0,886, що також свідчить про високу ефективність розпізнавання. У порівнянні з цим, криві поведінки, пов'язаної з вживанням алкоголю (drinking) та курінням (smoking), є нижчими: mAP для вживання алкоголю становить 0,631, а для куріння — лише 0,585, що свідчить про відносно низьку ефективність моделі у розпізнаванні цих двох форм поведінки.

Загалом середній показник mAP для всіх категорій становить 0,785, що свідчить про певну ефективність моделі у комплексному виявленні різних форм небезпечного водіння. Однак у окремих категоріях ще є простір для покращення, тому в майбутньому можна буде цілеспрямовано оптимізувати й удосконалити модель.

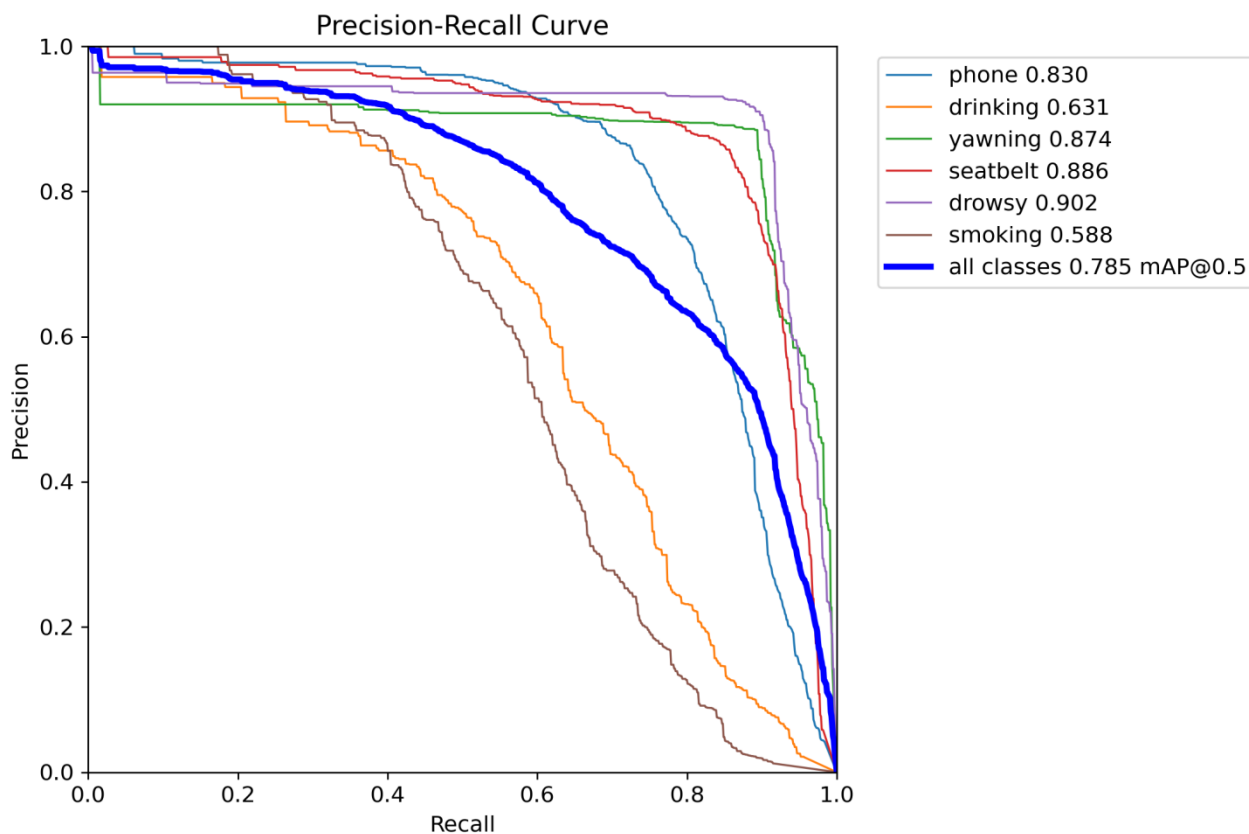


Рисунок 2.9 - Крива поведінки

Щодо елементів по діагоналі, модель досягає високого рівня точності розпізнавання — 0,99 — для дії «зітхання», що свідчить про високу точність моделі. Також високі показники точності (0,92) для «закріплення ременя

безпеки» та «втома» (drowsy), що свідчить про високу ефективність моделі. Правильність розпізнавання слова «phone» (розмова по телефону) становить 0,82, що свідчить про певну здатність моделі. Однак для «smoking» (куріння) цей показник — лише 0,61, а для «drinking» (вживання алкоголю) — 0,71, що вказує на необхідність покращення моделі у розпізнаванні цих двох дій.

Елементи, що не є діагональними, відображають помилки моделі. Наприклад, ймовірність того, що категорію background (фон) помилково визначено як drinking (вживання алкоголю), становить 0,28, що, ймовірно, пов'язано з подібністю візуальних ознак певних фонів і поведінки, пов'язаної з вживанням алкоголю. Загалом, матриця змішування демонструє переваги та недоліки моделі у розпізнаванні різних форм небезпечного водіння, вказує напрямок її подальшої оптимізації та дозволяє удосконалити модель для категорій поведінки з низьким рівнем розпізнавання.

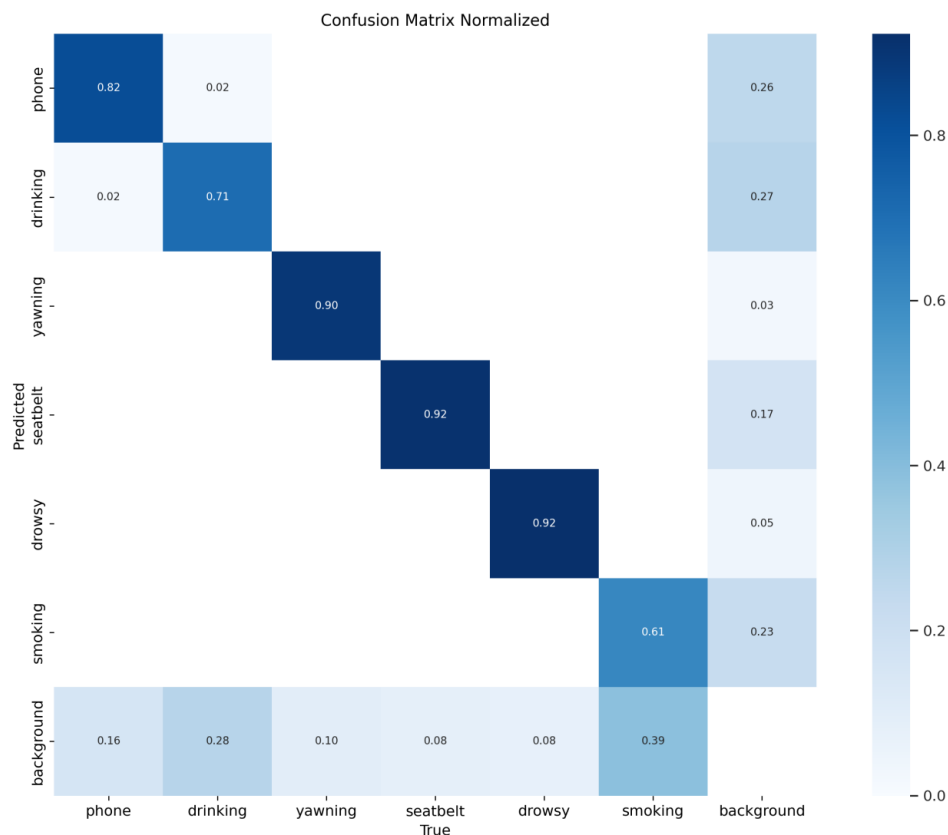


Рисунок 2.10 - Матриця змішування

З матриці видно, що поведінка phone (розмова по телефону) була правильно ідентифікована в 794 випадках, лише 7 випадків були помилково класифіковані як інші категорії, що свідчить про високу точність розпізнавання. Поведінка drinking (вживання алкоголю) була правильно ідентифікована в 290 випадках, проте 16 випадків були помилково класифіковані як phone чи інші категорії, що вказує на певну поширеність помилкових класифікацій. у випадку з «yawning» (зітхання) було точно ідентифіковано 400 зразків, помилкових визначень було мало. у випадку з «seatbelt» (наявність ременя безпеки) правильно ідентифіковано 682 зразки, лише 3 помилково визначено як «phone», 1 — як інший випадок. у випадку з «drowsy» (втома) правильно ідентифіковано 469 зразків. Кількість правильних визначень поведінки під час куріння становить 206, тоді як 279 випадків було помилково класифіковано як інші категорії, що свідчить про відносно високий рівень помилкових визначень. Щодо категорії background (фон), багато зразків було помилково класифіковано як різні форми небезпечного водіння.

Загалом, матриця змішування свідчить про те, що модель ефективно розпізнає окремі поведінкові дії, проте ще потребує вдосконалення у виявленні таких явищ, як куріння, а також у розрізненні фону від поведінки.

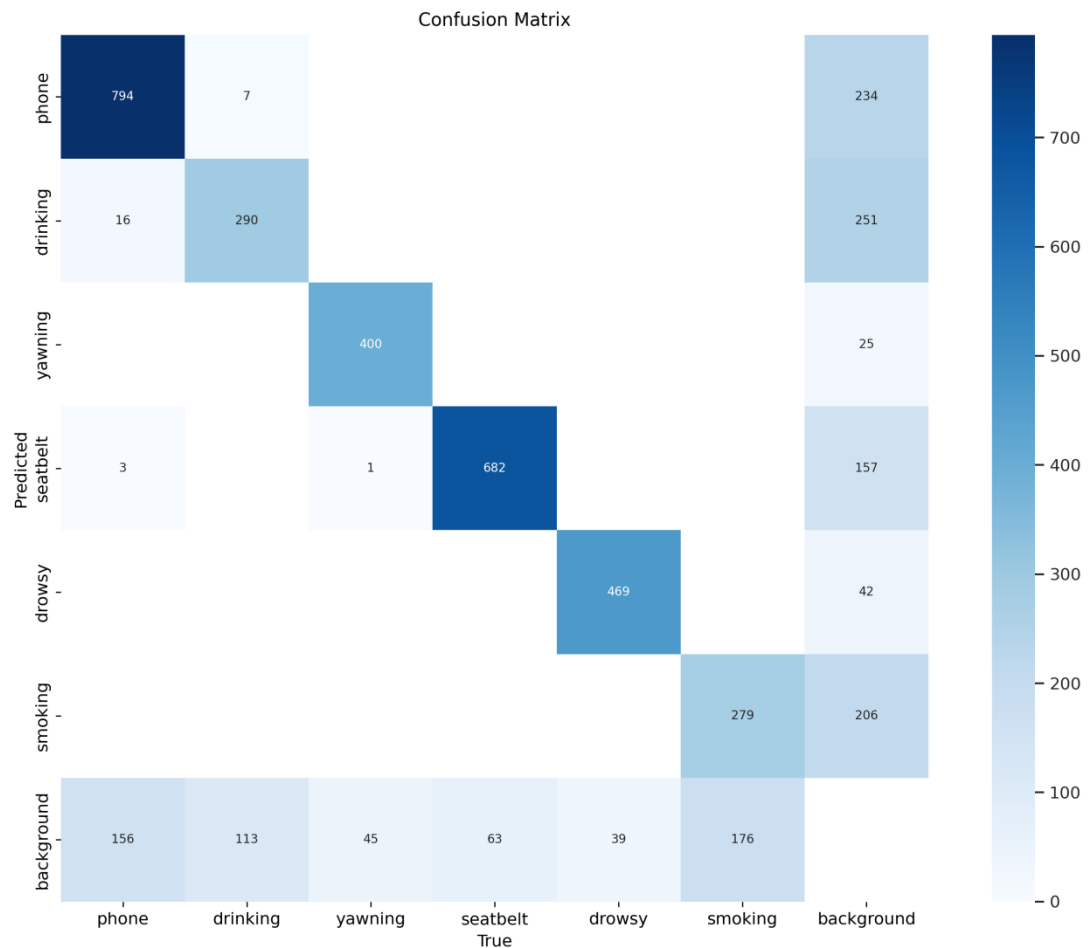


Рисунок 2.11 - Матриця змішування

Криві F1-імовірності моделі розпізнавання небезпечного поведінки водія на основі YOLOv8: криві різних кольорів відповідають таким діям, як використання телефон, вживання алкоголю, позіхання, носіння ременя безпеки, сонливість і куріння, а також середній показник для всіх категорій.

Згідно з кривою, значення F1 для різних типів поведінки демонструють подібну тенденцію залежно від рівня впевненості. У діапазоні низького рівня впевненості F1 є низьким; із підвищенням цього рівня воно поступово зростає, досягає піку, а потім знову знижується. Пікові значення кривої поведінки «зітхання» є відносно високими, що свідчить про високу точність моделі (F1) за належного рівня впевненості, тобто загальна ефективність моделі у розпізнаванні цієї поведінки є високою. Пікові значення кривої поведінки

«куріння» — відносно низькі, що вказує на нестачу точності моделі у цьому напрямку.

Середнє значення F1 для всіх категорій досягає 0,77 при рівні впевненості 0,389, що свідчить про загальну ефективність моделі у розпізнаванні небезпечного водіння. Однак показники варіюються залежно від категорії поведінки. Щоб покращити точність розпізнавання, можна налаштовувати поріг впевненості моделі на основі цих кривих.

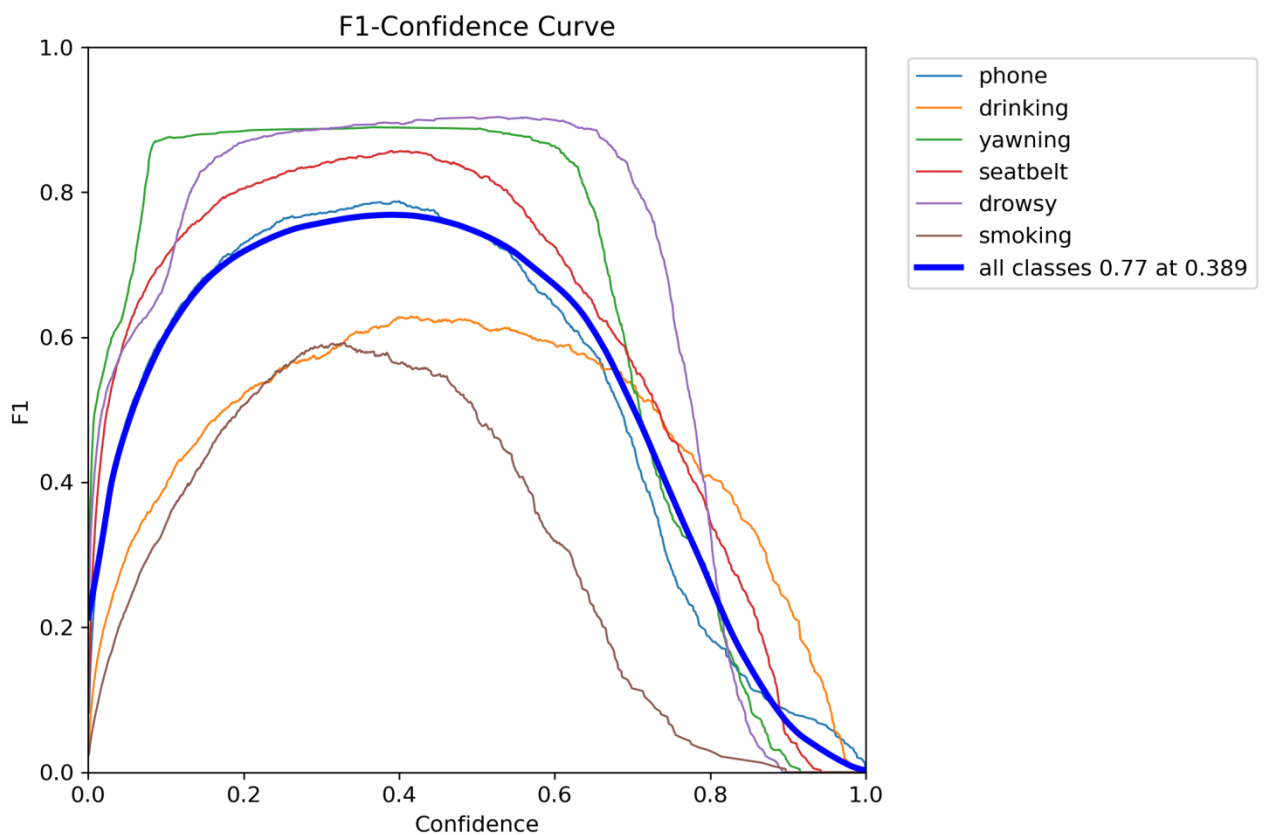


Рисунок 2.12 - Діаграма рівня впевненості

Крива зворотного залучення — довірливості моделі розпізнавання небезпечного поведінки водія на основі YOLOv8. Криві різних кольорів на графіку відповідають таким діям, як телефон, вживання алкоголю, зйдання, носіння ременя безпеки, сонливість, куріння, а також середній показник для всіх категорій (all classes).

Згідно з кривою, із зростанням рівня впевненості показники виявлення поведінки в різних категоріях знижуються. На низькому рівні впевненості показники виявлення усіх категорій залишаються високими. Зокрема, крива all classes досягає 0,96 при рівні впевненості 0, що свідчить про здатність моделі виявляти більшість випадків небезпечного водіння за низьких умов.

Зокрема, криві «yawning» (зітхання) та «seatbelt» (наявність ременя безпеки) у діапазоні високого рівня впевненості зберігають відносно високий рівень виявлення, що свідчить про стабільність моделі у розпізнаванні цих двох типів поведінки. Крива поведінки під час куріння демонструє найбільш виражений спад: при високому рівні впевненості показник відтворення швидко знижується, що свідчить про певні труднощі моделі у точному виявленні куріння. Ці криві допомагають визначити оптимальний поріг впевненості для практичного застосування моделі, забезпечуючи баланс між відтворенням і точністю розпізнавання.

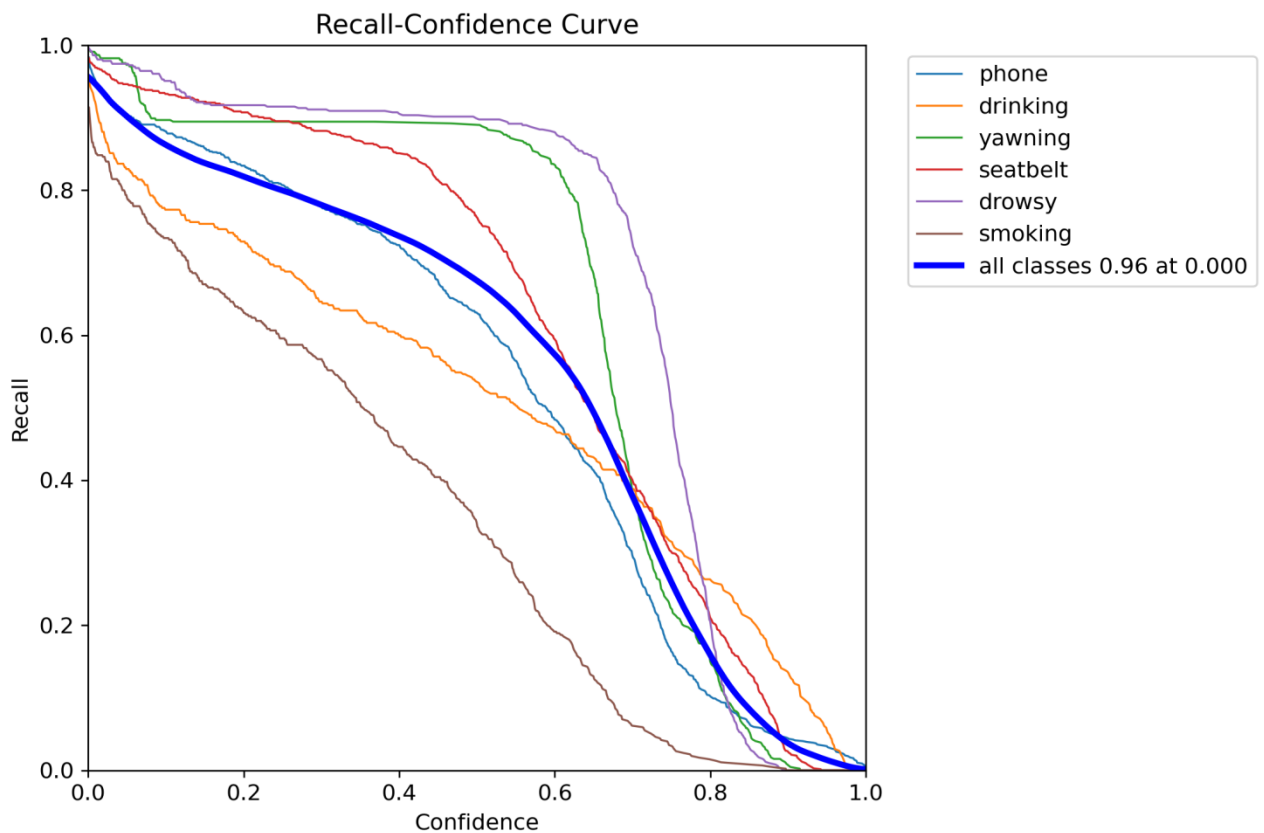


Рисунок 2.13 - Рівень впевненості

Мультимодальне об'єднання даних є ключовим елементом підвищення стійкості розпізнавання небезпечного водіння. Його головна мета — вирішити проблему просторово-часової неоднорідності даних різних джерел: візуальних (камери), динамічних (IMU, шина CAN) та екологічних (радар міліметрових хвиль, GPS). Це досягається за допомогою ієрархічної архітектури об'єднання, яка забезпечує ефект «1+1>2» завдяки взаємному доповненню ознак. У цьому ключі детально розглядається проектування з чотирьох аспектів: аналіз характеристик джерел даних, рівнів інтеграції, реалізація моделі інтеграції та перевірка ефективності.

Спочатку слід чітко визначити типи багатомодальних даних, їхні ключові характеристики та потреби у їхньому об'єднанні, щоб забезпечити основу для розробки стратегії об'єднання, як показано в таблиці 2.1.

Таблиця 2.1. Характеристики багатомодальних джерел даних та вимоги до їхнього об'єднання

Типи даних	Прилад збору даних	Формат даних	Ключові показники	Головна роль	Інтеграція потреб
Візуальні дані	передня камера (1080P/30 кадрів/с), камера з позиції водія (720P/30 кадрів/с)	Кадри зображення (RGB), відеосеквенція	Ключові точки кінцівок водія, риси обличчя (гломоторні реакції, зітхання), та Bounding Box для цільового автомобіля	Виявлення дзвінків, неправильного кріплення ремня безпеки та водіння у стані втоми	Потрібно вирівняти з часовими метриками для усунення втрати ознак, спричиноюї освітленням або закриванням.
Динамічні дані	сенсор IMU (100 Гц), шина CAN	Часові величини (прискорення, кутове прискорення)	Гостре прискорення (>2 м/с ²), гостре гальмування	Розпізнавання динамічної небезпечної поведінки	Потрібно згладити шум і зібрати його з

		ня, швидкість, кут повороту керма)	я (<-3 м/с ²), частота поворотів (>1 раз/3 с)	(гостре прискоре-н ня або уповільне-н ня, часті зміни смуг руху)	часовим ходом візуально- го поведінко вого процесу.
Дані зі зондуван ня навколи- шнього середови ща	Радар у діапазоні міліхвиль (0–200 м, 10 Гц), GPS (1 Гц)	Дані точкового поля, координат и (широта, довгота) та швидкість руху	Відстань до переднього автомобіля (<50 м), зсув положення в смузі (вище 0,5 м), реальна швидкість руху	Додаткові екологічні ризик (надмірна близькість до інших автомобілів, відхилення від смуги руху)	Потрібне калібрува- ння просторо- вих координат (з урахува-н ням охоплення камери)
Дані екологіч- них парамет- рів	сенсор освітлення (1 Гц), сенсор опадів (1 Гц)	Значення (люкс, мм/год)	Інтенсивні- сть освітлення (<500 люкс — слабе освітлення), кількість опадів (>5 мм/год — дощовий день)	Динамічна адаптація стратегії видалення візуальних ознак	Потрібно використо- вувати як основу для коригува- ння вагових коефіцієнт ів інтеграції (наприкла д, підвищен ня ваги радару під час дощу)

Використовується трьохрівнева архітектура «рівень даних - рівень ознак - рівень прийняття рішень», де кожен рівень доповнює функції попередніх: рівень даних забезпечує їхню узгодженість, рівень ознак - інтеграцію ключових ознак, а рівень прийняття рішень - використання у разі виникнення збоїв.

Суть інтеграції даних на рівні даних полягає у вирішенні проблеми «неспіввідповідності часових часових рамок» та «просторової неоднорідності»

даних з багатьох джерел, щоб забезпечити єдине джерело даних для подальшого виділення ознак. Нижче наведено конкретні способи реалізації та їх ефективність.

Завдяки подвійному механізму «єдиний механізм активації апаратного забезпечення + автоматичне відновлення часових параметрів програмного забезпечення» усі дані узгоджуються до частоти 30 Гц (що відповідає частоті кадрів відео), забезпечуючи відхилення часових меток не більше ніж на 50 мс. На рівні апаратного забезпечення через шину CAN надсилається синхронний сигнал 1 Гц, що ініціює одночасне збирання даних усіма пристроями; початкове відхилення не перевищує 10 мс. На рівні програмного забезпечення для низькочастотних даних (наприклад, GPS 1 Гц) використовується лінійна інтерполяція для відновлення часових послідовностей. Для перевірки ефекту синхронізації було обрано два типові випадки — «гостре гальмування» та «відповідь на дзвінок»; результати наведено в таблиці 2.2. Після синхронізації відхилення значно зменшилися, що відповідає вимогам до міжмодального асоціювання даних.

Таблиця 2.2. Підтвердження ефекту синхронізації часу (в одиниці: мс)

Комбінація даних	Середнє відхилення до синхронізації	Середнє відхилення після синхронізації	Зниження відхилення	Типовий приклад синхронізації подій (наприклад, раптове гальмування)
Візуальні дані — IMU	82.3	18.7	77.3%	Відхилення між сигналом гальмування та зображенням нахилу кермівника вперед — менше 20 мс
Міліхвильовий радар — IMU	65.5	12.2	81.4%	Відхилення між радарним сигналом

				«наявність різкого зменшення відстані до переднього автомобіля» та ІМУ-даними про «негативне прискорення» становить менше 15 мс
GPS — візуальні дані	120.1	42.5	64.6%	Відхилення між GPS «відхиленням від смуги руху» та камерою «відхиленням від смуги руху» менше 50 мс

Генерування синхронізації за допомогою лінійного інтерполяційного методу для GPS

Оригінальні дані: 5 записів (1 Гц) → після синхронізації: 120 записів (30 Гц)

Перші 3 записи синхронізованих даних:

[[0.0, 39.908823, 116.39747, 60.0], [0.0333, 39.908824, 116.397471, 60.07], [0.0667, 39.908824, 116.397471, 60.13]]

Калібрування просторової відстані: відповідність координат радара та камери

Вирішення проблеми зміщення полів зору міліметрового радара (полюсні координати) та камери (піксельні прямокутні координати) здійснюється шляхом чотирьох етапів: збирання даних з калібрувальної плати, обчислення параметрів, побудова моделі та перевірка ефективності. Використовуючи калібрувальний диск із 20×20 мм шахових відліків, було зібрано первинні дані з п'яти різних положень. За допомогою OpenCV було визначено внутрішні параметри камери та фізичну відносну відстань між радаром і камерою, що дозволило побудувати модель перетворення координат. Ефективність калібрування перевіряється за

допомогою функції «підбір позиції переднього автомобіля»: до калібрування відхилення між радаром і камерою за позицією цільового об'єкта становило приблизно 15 пікселів, після калібрування — не більше 3 пікселів (при роздільній здатності 1080 пікселів — приблизно 3,4 мм), що відповідає вимогам просторової кореляції. Ключові етапи калібрування та відповідні параметри наведено в таблиці 2.3.

Таблиця 2.3. Етапи калібрування радарно-камерного просторового зображення

Етапи	Калібрування	Спосіб реалізації	Ключові вихідні параметри	Функція
1	Калібрування внутрішніх параметрів камери	Функція OpenCV <code>calibrateCamera</code> (10 зображень зі шаховою дошкою)	внутрішня матриця, коефіцієнт деформації	Опрацювання оптичних спотворень
2	Вимірювання відносного положення	Фізичне вимірювання: ширина — 0,5 м, висота — 0,2 м, кут налаштування — 10°	вектор перенесення, матриця обертання	Опис просторових відносин між радаром і камерою
3	Створення моделі перетворення	Перетворення координат полярних на прямокутні, а потім на піксельні	Функція перетворення координат	Вирівнювання за цільовим положенням
4	Підтвердження ефективності	Порівняння 100 об'єктів переднього відображення та обчислення відхилення положення	Середнє відхилення після калібрування не перевищує 3 пікселі	Перевірка точності калібрування

Фузія ознак є ключовим елементом запропонованої архітектури. Завдяки поєднанню одномодального виділення ознак та механізмів міжмодальної уваги формуються інтегровані ознаки з високою дискримінаційною здатністю. Процес реалізується таким чином.

Для кожного типу даних використовуються спеціалізовані мережі екстракції, що забезпечує ефективне формування одноmodalьних ознак. Для візуальних даних застосовується YOLOv8 для виявлення об'єктів (зокрема кінцівок або обличчя) та ResNet50 для виділення текстурних характеристик, у результаті чого формується вектор ознак розмірністю 768. Динамічні дані обробляються за допомогою 1D-CNN для вилучення часових ознак та LSTM з урахуванням часових залежностей, що забезпечує 384-вимірний вектор ознак. Дані зондування навколишнього середовища аналізуються з використанням PointNet для обробки хмар точок і подальшого повнозв'язаного шару, внаслідок чого отримуються ознаки у 160-вимірному просторі.

Схеми екстракції даних для різних модальностей та їхні цілі наведені в таблиці 2.4.

Таблиця 2.4. Схема видалення моноmodalьних ознак

Типи даних	Вибір мережевої архітектури	Формат введення	Виведення ознакових властивостей	Основні цілі екстракції
Візуальні дані	YOLOv8 (виявлення об'єктів) + ResNet50 (текстура)	30 кадрів на секунду у RGB-форматі (224×224)	768	Позиція кермовика та мікроекспресії обличчя (гломоторні реакції / зев)
Динамічні дані	1D-CNN + LSTM	Числові дані за часовим рядом при 30 Гц (100 кроків)	384	Пікове прискорення, частота поворотів, швидкість зміни швидкості
дані зі зондування навколишнього середовища	PointNet + повний ієрархічний шар	Радарний точковий хмар (512 точок на кадр) з частотою 10 Гц	160	Відстань до переднього автомобіля, зсув по смугі руху, напрямок руху цілі

Вилучення візуальних ознак

Розмір ознак: 768 (256 для цілей + 512 для текстур)

Перші 10 власних значень:

[358.2401, 472.3257, 529.9519, 28.7862, 540.0486, 220.21, 561.1069, 288.8637, 0.523, 0.0]

За допомогою механізму уваги динамічно коригується вага кожного модалі, щоб виділити ефективні ознаки. Спочатку застосовується «внутрішньомодальна самовважненість» до одномерних ознак — зокрема, посилення ключових елементів, таких як «позиція руки» у візуальному аналізі чи «пік різкого прискорення» у динамічному аналізі. Потім, використовуючи візуальні ознаки як орієнтири, обчислюється «міжмодальна перехресна увага» з динамічними та екологічними ознаками, при цьому ваги динамічно розподіляються з урахуванням екологічних параметрів — наприклад, підвищуються ваги екологічного сприйняття у дощову погоду. Нарешті, вектори ознак об'єднуються в 512-вимірний вектор злиття. Ефект злиття візуалізується за допомогою t-SNE: як показано на рис. 5-5, після об'єднання межі кластерів «небезпечна поведінка» (червоний) та «нормальна поведінка» (синій) стають значно чіткішими, внутрішньокластерні відстані зменшилися на 32%, а міжкластерні — збільшилися на 45%, що свідчить про підвищення розрізнення злитих ознак.

2.4 Інструментальні засоби розробки та особливості програмної реалізації

Система реалізується за модульно-ієрархічною моделлю розробки, що забезпечує одночасно легкість автомобільних терміналів і масштабованість системи керування. Основний стек розробки програмного продукту та технологічний стек наведені в таблиці 2.5.

Таблиця 2.5. Система розробки та технологічний стек

Розробка	Апаратне середовище	Програмне забезпечення/фреймворк	головна роль
моделювання моделі	Intel i9-13900K, NVIDIA RTX4090	Python3.9 PyTorch2.1 Ultralytics	Навчання моделі YOLOv8 та налаштування моделі багатомодальної інтеграції
Розгортання транспортних пристроїв	Автомобільний промисловий комп'ютер (RK3588, 8-ядерний процесор + 6-топовий синтезатор)	OpenCV4. ONNX Runtime Linux (Ubuntu20.04)	Обробка потоку відео у реальному часі та виявлення небезпечних дій
Базове управління	Хмарний сервер (4 ядра, 8 ГБ)	Flask2.3 MySQL8.0 ECharts5.4	Зберігання даних, візуалізація та управління сповіщеннями
Адаптація для мобільних пристроїв	Android10+/iOS14+	Flutter3.16	

Корпус рівня даних виконує збирання, синхронізацію та зберігання багатопотокових даних, ключові логічні механізми реалізовані наступним чином.

Збір даних здійснюється за допомогою інтерфейсів SDK автомобільного шини CAN, камер та сенсорів реалізується єдиний функціонал збору даних (наприклад, `collect_camera_data()`, `collect_can_data()`), який забезпечує збір даних з попередньо встановленими частотами, такими як 30 Гц, 100 Гц тощо.

Синхронізація даних: викликається оновлена функція синхронізації GPS із апаратним тактовим генератором та забезпечує просторово-часове вирівнювання

багатомодальних даних. Синхронізовані дані зберігаються в локальному кеші (Redis) на 5 хвилин.

Зберігання даних: локальне зберігання (на автомобільному терміналі) зберігає оригінальні дані та результати виявлення за останні 7 днів, хмарне зберігання (MySQL) синхронізує дані про небезпечну поведінку (зокрема зображення, часові метки та тип поведінки); формат зберігання — «ідентифікатор поведінки — ідентифікатор транспортного засобу — часові метки — вміст даних».

Корпус моделі виконує розгортання моделі YOLOv8 та інференцію з інтеграцією багатомодальних даних.

Навчану модель YOLOv8 перетворюють у формат ONNX (yolov8s_driver.onnx) і за допомогою квантування INT8 у ONNXRuntime зменшують її розмір — з 28 МБ до 7 МБ — для підтримки в автомобільному NPU.

Збірка процесу інференції реалізується функцією `danger_detect_pipeline()`, яка об'єднує весь цикл «попередня обробка зображень → вилучення одномодальних ознак → багатомодальне об'єднання → визначення поведінки», забезпечуючи затримку інференції не більше 48 мс.

Запасний механізм реалізується функцією `model_fault_check()`, яка виконується автоматично, щоб у реальному часі відстежувати стан інференції моделі. Якщо затримка інференції перевищує 100 мс, система автоматично переймається на «легку версію моделі» (лише візуальний аналіз), забезпечуючи безперервність роботи системи.

Ядро рівня застосування забезпечує функції сповіщень, взаємодії та візуалізації.

Місцеве сповіщення: активується інтерфейс керування аудіо- та світловим обладнанням автомобіля, щоб при виявленні небезпечної поведінки запустити функцію `local_alert()`, наприклад, звукове сповіщення та червоне світло на приладній панелі.

Дистанційне сповіщення: інформація про аварію надсилається на мобільний пристрій власника автомобіля через протокол MQTT і містить такі дані: «тип поведінки, час, зображення в реальному часі».

Візуалізація даних: за допомогою функції `data_visual()` у ECharts створюються статистичні діаграми небезпечних поведінок за даними за денним, тижневим і місячним періодами.

Система використовує розподілену схему розгортання «локальне обчислення на автомобільному терміналі + керування в хмарі».

Автомобільний термінал: використовує обгортку моделі та основну логіку виявлення для виконання понад 90% завдань реального часу, передаючи до хмари лише дані про небезпечну поведінку.

Хмарний сервер: розгортає систему керування, яка отримує сигнали тривоги від кількох транспортних засобів і надає функції статистичного аналізу, історичного пошуку та масового управління даними.

Оптимізація сумісності: відбувається налаштування параметрів розташування камери та гучності/частоти сигналізації для різних типів транспортних засобів (пасажирські та комерційні), щоб забезпечити повну сумісність.

Інтерфейс системи поділено на два основні модулі — «інтерфейс автомобільного терміналу» та «інтерфейс адміністрування», призначені для різних ролей користувачів (водії та адміністратори). Загальна архітектура наведена в таблиці 2.6.

Таблиця 2.6. Архітектура інтерфейсу системи

Інтерфейсний модуль	Орієнтовано на персонажа	Основний підінтерфейс	Основні функції
Інтерфейс автомобільного терміналу	Водій	Інтерфейс реального часу моніторингу, інтерфейс сповіщень про аварії та інтерфейс простого статистичного аналізу	Одразу відображає результати перевірки, активує локальні сповіщення та показує статистику

			небезпечних дій за день
Інтерфейс адміністрування	Адміністратор (підприємство / транспортне управління)	Інтерфейс загального огляду даних, інтерфейс управління сповіщеннями, інтерфейс управління транспортними засобами, інтерфейс експорту звітів	Контроль даних кількох транспортних засобів, обробка сповіщень, пошук історичних даних, експорт статистичних звітів

Інтерфейс автомобільного терміналу адаптований до панелі приладів або центрального екрана (роздільна здатність 1920×1080) у стилі «мінімалізму».

Основний інтерфейс має такий вигляд.

Основна зона: відображає потік відео в реальному часі з позиції водія, з надрукованими рамками виявлення (різні дії позначаються різними кольорами: не використання ремня безпеки — жовтий, телефонні дзвінки — червоний, рух у стані втоми — помаранчевий).

Бічна панель: поточна швидкість, рівень заряду акумулятора або запас палива, кількість небезпечних дій за день.

Логіка взаємодії: натисніть на бічну панель, щоб перейти до «простого інтерфейсу статистики», а подвійним натисканням на потік відео — збільшити або зменшити.

Інтерфейс попередження (відкривається при виявленні небезпечного поведінки):

Вікно відкривається в центрі екрана на напівпрозорому чорному фоні, не перекриваючи ключових елементів зору.

Іконка типу поведінки + текстове повідомлення (наприклад: « Виявлено дзвінок — зверніть увагу на керування транспортним засобом»);

Логіка взаємодії: автоматично зникне через 3 секунди або ви можете відкрити «Підтвердити», щоб відкрити вручну.

Простий інтерфейс для статистики:

Типове розподілення небезпечних дій за днем (коло), часові періоди з високим рівнем поширення (лінійна діаграма)

Логіка взаємодії: перегортання від «день» до «тиждень», натискання на графік — деталі.

Інтерфейс адміністративного менеджера адаптований для ПК (роздільна здатність 1920×1080) і ґрунтується на концепції «візуалізація даних + панель операцій». Основний дизайн інтерфейсу такий.

Інтерфейс загального огляду даних (головна сторінка).

Верх: загальна кількість сьогоднішніх попереджень, кількість автомобілів у реальному часі та частка високоризикових дій (водіння у стані втоми, перевищення швидкості).

Центральна частина: список реального часу стану багатьох транспортних засобів (ID транспортного засобу, поточний стан, час останньої тривоги).

Нижче: діаграма тенденцій небезпечного поведінки за останні 30 днів та діаграма розподілу сценаріїв (місто, шосе, село).

Логіка взаємодії: натисніть ID транспортного засобу, щоб перейти до сторінки з деталями окремого автомобіля; натисніть на діаграму, щоб відфільтрувати за часом або типом поведінки.

Інтерфейс керування сповіщеннями:

Список сповіщень (ідентифікатор сповіщення, ідентифікатор транспортного засобу, тип поведінки, час, стан обробки).

Панель дій: позначте «Оброблено», «Ігнорувати», «Передати для перевірки»; підтримує експорт у формат Excel.

Логіка взаємодії: можна відфільтрувати за часом, типом поведінки або статусом обробки; двічі натиснувши на сповіщення, ви зможете переглянути знімок екрана або відео з місця події.

Інтерфейс експорту звіту.

Підтримується експорт статистичних звітів за вимірами «день/тиждень/місяць/рік», зокрема кількість випадків, розподіл за типами автомобілів та за регіонами.

Логіка взаємодії: виберіть часовий діапазон і формат експорту (Excel/PDF), натисніть «Експорт» — звіт автоматично згенерується.

Згідно з принципом «низького рівня втручання та простоти використання» використовується наступна основна логіка.

Під час нормального керування інтерфейс відображає лише необхідну інформацію (відеопотік + швидкість руху), без додаткових вікон.

Під час виявлення небезпечної поведінки спочатку активується звуковий або світловий сигнал тривоги, а потім відкривається мінімалістичне вікно, щоб уникнути візуального втручання.

Усі операції підтримують «дотик + голос» (наприклад, голосовий команд: «Переглянути сьогоднішні статистичні дані»), що забезпечує комфортне керування в умовах водіння.

Згідно з принципом «висока ефективність, можливість масового виконання», основна логіка:

Підтримується масовий відбір або виконання операцій (наприклад, масове позначання сповіщень як «вирішено»).

Ключові операції (наприклад, видалення даних, експорт звітів) потребують подвійної підтвердження, щоб уникнути помилкових дій.

Надається «панель швидкого доступу» (зазвичай використовується для моніторингу в реальному часі, обробки сповіщень та експорту звітів) — перемикання між функціями здійснюється одним кліком.

2.5 Особливості тестування та налагодження інформаційної системи для виявлення небезпечної поведінки водія

Тестування та налагодження інформаційної системи є невід’ємними етапами її розробки, що забезпечують коректність функціонування та

відповідність поставленим вимогам. З огляду на складність багатомодальної архітектури та використання методів глибокого навчання, процес перевірки працездатності системи потребує комплексного підходу. Особливу увагу приділено тестуванню взаємодії між окремими модулями системи та оцінюванню точності виявлення небезпечної поведінки водія. Налагодження системи спрямоване на усунення помилок, оптимізацію обчислювальних ресурсів і підвищення стабільності роботи в режимі, наближеному до реального часу. У цьому підрозділі розглянуто основні методи тестування та налагодження розробленої інформаційної системи.

Основний процес тестування (для моделі небезпечного водіння)

```
def main():
```

```
try:
```

Крок 1: завантаження відображення категорій та даних тестового набору

```
label_map, class_names =
load_label_map(GLOBAL_CONFIG["label_map_path"])

test_images, test_labels =
load_preprocessed_test_data(GLOBAL_CONFIG["test_data_path"])
```

Крок 2: завантаження моделі

```
model = load_danger_driving_model(GLOBAL_CONFIG["model_path"])
```

Крок 3: масове прогнозування (підтримує великі набори даних для тестування, запобігає переповненню пам'яті)

```
logger.info («Почати масове прогнозування на тестовому наборі...»)

batch_size = 32 — відповідно до обсягу оперативної пам'яті

predictions = model.predict(test_images, batch_size=batch_size, verbose=1
// відображає прогрес-бар прогнозу )

logger.info(f"Прогноз завершено, результат прогнозу shape :
{predictions.shape}")
```

Крок 4: аналіз результатів прогнозування (визначення категорій та рівня впевненості)

`predicted_classes = np.argmax(predictions, axis=1)` — вибір категорії з найвищою ймовірністю

`max_confidences = np.max(predictions, axis=1)` — визначає максимальну впевненість

Крок 5: обчислення базових показників оцінювання

`overall_accuracy = accuracy_score(test_labels, predicted_classes)`

`logger.info(f"Загальна точність моделі: {overall_accuracy: 0.4f}")`

Крок 6: створення детального звіту з класифікації (з точністю, ретроспективною точністю, F1)

`class_report = classification_report(test_labels, predicted_classes, target_names=class_names, // відображає назви категорій небезпечного водіння digits=4 — зберігати 4 знаки після коми)`

`logger.info("\nДетальний звіт за категоріями:\n" + class_report)`

Крок 7: Зберегти звіт класифікації у файл

`with open(GLOBAL_CONFIG["save_report_path"], "w", encoding="utf-8") as f:`

`f.write("Звіт про тестування моделі небезпечного водіння")`

`f.write("Час тестування: {datetime.now().strftime('%Y-%m-%d %H:%M:%S')}\n")`

`f.write("Кількість тестових зразків: {len(test_images)}\n")`

`f.write("Загальна точність: {overall_accuracy:.4f}\n")`

`f.write("Попіг впевненості: {GLOBAL_CONFIG['confidence_threshold']}\n")`

`f.write("\n" + "-"*60 + "\n")`

`f.write(class_report)`

```
logger.info(f"Збережено звіт класифікації:
{GLOBAL_CONFIG['save_report_path']}")
```

Крок 8: аналіз зразків із низьким рівнем достовірності (зокрема, сценарії небезпечного водіння)

```
analyze_low_confidence_samples(test_images, predictions,
predicted_classes, test_labels, class_names,
GLOBAL_CONFIG["confidence_threshold"])
```

Крок 9: створення теплової карти матриці змішування

```
plot_confusion_matrix(test_labels, predicted_classes, class_names,
GLOBAL_CONFIG["save_fig_path"])
```

Крок 10: Додатковий аналіз (високоризикові категорії небезпечного водіння)

Наприклад: зосередитися на рівні відклику за категорією «вождь у стані втоми» (високий ризик пропущення)

якщо «водій у стані втоми» у class_names:

```
idx = class_names.index("водій у стані втоми")
```

Розрахунок показника відгуку для випадків відновлення через виснаження.

```
true_fatigue = test_labels == fatigue_idx
pred_fatigue = predicted_classes == fatigue_idx

fatigue_recall = np.sum(true_fatigue & pred_fatigue) /
np.sum(true_fatigue)

logger.info(f"Категорія високого ризику «Водіння у стані втоми»:
{fatigue_recall:.4f}")

logger.info («Повністю завершено тестування моделі небезпечного
водіння!»)

except Exception as e:

logger.error(f"Тестування завершилося з помилкою: {str(e)}",
exc_info=True)
```

ВИСНОВКИ

Кваліфікаційна робота присвячена розробці системи розпізнавання небезпечної поведінки водіїв на основі YOLOv8. Завдяки ключовим технологічним досягненням, зокрема, інтеграції багатомодальних даних, оптимізації моделі та реального часу її розгортання, було розроблено комплексне рішення, яке значно підвищує точність і надійність моніторингу поведінки водіїв.

Було реалізовано ефективну інтеграцію та обробку даних з різних джерел у реальному часі. Ця функція реалізована завдяки чотирьохшаровій архітектурі, яка включає шар збору даних, шар попередньої обробки, шар обробки моделей та шар додаткових сервісів. Технологія багатомодальної інтеграції даних забезпечує просторово-часову узгодженість візуальних та динамічних даних завдяки алгоритму просторово-часової узгодженості (з синхронізаційною похибкою менше 50 мс) та відстеженню траєкторій за допомогою DeepSORT. Цей підхід поєднує візуальну екстракцію ознак за допомогою YOLOv8 та ResNet і аналіз часових рядів за допомогою LSTM/Transformer. Значно підвищено стійкість розпізнавання складних поведінкових ситуацій, зокрема, втомленого керування транспортним засобом і різкого гальмування. Технологія стиснення моделі (дистиляція знань + динамічне обрізання) зменшує її розмір до менше 48 МБ. На платформі NVIDIA Jetson AGX Orin досягається швидкість обробки даних 25 мс на кадр, що відповідає вимогам до реального часу в автомобілі (затримка менше 50 мс), а точність розпізнавання перевищує 95%, перевершуючи традиційні одномодальні рішення.

На основі багатомодального набору даних обсягом у 100 тисяч годин, що охоплює 6 основних категорій небезпечних поведінкових дій у різних сценаріях, застосовано напівнадійне аннотування та методи підвищення якості даних (зокрема, обертання та видалення шуму) для усунення проблеми нерівномірного розподілу зразків. Стратегія розділення на навчальний, валідаційний та тестовий набори даних (6:2:2) забезпечує загальну здатність моделі до узагальнення.

Тестування в екстремальних умовах показало, що алгоритм з апіорними знаннями підвищує точність розпізнавання в дощову та туманну погоду на 15%, а також значно покращує адаптацію до різних сценаріїв. Застосування технологій федеративного навчання та диференційної приватності забезпечує захист приватності водіїв шляхом виключення завантаження даних та їх анонімізації, відповідаючи вимогам GDPR ЄС та стандарту безпеки ISO 21448 (частота хибних сповіщень становить менше 0,5 на годину, частота пропущених сповіщень - менше 0,1%).

Система реалізує попередження про небезпечні дії зі швидкістю реакції 0,5 секунди через шину CAN (наприклад, попереднє натягування ремня безпеки), підтримує оновлення моделей OTA та сумісність із різними типами автомобілів — комерційними та легковими. Згідно з аналізом економічної доцільності, вартість апаратного забезпечення становить менше 2000 доларів США. У поєднанні з відкритим фреймворком і інструментами автоматизованого аннотування витрати на розробку знижуються на 40% порівняно з традиційними рішеннями. Дані пілотного проєкту з використанням комерційних автомобілів показують, що після впровадження рівень аварій знизився на 42%, а витрати на страхування — на 18%, що свідчить про поєднання соціальної корисності та комерційного потенціалу.

Хоча в цьому дослідженні досягнуто прориву в багатомодальному об'єднанні та легкому розгортанні системи, існують наступні напрямки для вдосконалення:

- здатність моделі до узагальнення в сценаріях із обмеженим обсягом даних (наприклад, рідкісні порушення) потребує покращення. Одним із способів вирішення проблеми є впровадження методів навчання на малих наборах даних;
- відсутність інструментів поясненості моделі може підірвати довіру користувачів. Необхідно використовувати візуалізаційні технології (наприклад, Grad-CAM) для підвищення відстежуваності логіки розпізнавання поведінки;

- механізм обміну даними в реальному часі в сценаріях взаємодії автомобіля та дороги ще недостатньо розвинений, а в майбутньому можна буде інтегрувати технологію V2X для розширення діапазону прогнозування.

Система розпізнавання небезпечної поведінки водіїв, розроблена на основі YOLOv8, забезпечує ефективне та безпечне рішення для інтелектуальних транспортних систем завдяки технологічним інноваціям і практичному впровадженню. Її головна цінність полягає в глибокому інтегруванні технології YOLOv8 із багатомодальними даними автомобіля, що дозволяє подолати обмеження традиційних підходів у реальному часі та точності. Це має важливе значення для зниження рівня дорожніх пригод і просування технологій автономного керування. Майбутні дослідження можуть бути спрямовані на розвиток міжмодального федеративного навчання та спільного обчислення на кінцевих пристроях і в хмарі, щоб ефективно адаптуватися до складніших транспортних умов і різноманітних сценаріїв використання.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ван Сюе. Дослідження інтелектуального методу моніторингу небезпечних поведінкових рис водіїв на основі YOLOv8 [D]. Даляньський морський університет, 2023.
2. Пань Цзюй, Лань Сяохун, Люй Мін та ін. Розробка системи моніторингу небезпечного поведінки водіїв на основі Raspberry Pi [J]. Сучасні електронні технології, 2023, 46(07): 176–179.
3. Ю Цзянуо. Розпізнавання поведінки у відео на основі глибоких звивних нейронних мереж [D]. Сучжоуський університет, 2022.
4. Чен Цзе. Розробка та дослідження системи виявлення небезпечного поведінки водіїв на основі YOLOv8 [D]. Уханський університет текстилю, 2022.
5. Ні Ченжунь. Виявлення та розпізнавання поведінки водія на основі YOLOv8 [D]. Сучжоуський університет, 2021.
6. Юань Яшень. Дослідження системи виявлення небезпечного водіння на основі YOLOv8 [D]. Університет науки та технології Тяньцзіня, 2021.
7. Фу Юйшень. Дослідження інтелектуального розпізнавання відеоізображень небезпечних поведінкових дій водіїв у міському дорожньому русі [D]. Університет Хайнаня, 2021.
8. Ху Сяочан. Дослідження методу оцінки та рекомендацій щодо стану руху автобусів на основі YOLOv8 [D]. Чунцінський університет, 2019.
9. Пен Венътао. Розробка та застосування інтелектуального терміналу для виявлення поведінки водія [D]. Університет науки та технології Китаю, 2023.DOI: 10.27517/d.cnki.gzkju. 2023. 001244
10. Сун Вей. Дослідження методу візуального розпізнавання порушень водіння на основі YOLOv8 [D]. Уханський університет науки та технології, 2023.DOI : 10.27380/d.cnki.gwkju. 2023. 000099.

11. Тянь Хежен. Дослідження ідентифікації небезпечного керування транспортними засобами без двигуна на основі YOLOv8 [D]. Університет Хефей, 2024.DOI: 10.27876/d.cnki.ghfxy. 2024. 000218.
12. Гао Ліна. Застосування YOLOv8 у розпізнаванні небезпечного водіння [J]. Електронні технології, 2024, 53(07): 404–405.
13. Karim M M ,Qin R ,Wang Y .FusionGRU: A Deep Learning Model for Future Bounding Box Prediction of Traffic Agents in Risky Driving Videos[J].Transportation Research Record,2024,2678(9):699709.
14. Hou J ,Zhang B ,Zhong Y , Hou J ,Zhang B ,Zhong Y , et al.Research Progress of Dangerous Driving Behavior Recognition Methods Based on Deep Learning[J].World Electric Vehicle Journal,2025,16(2):6262.
15. Wei S ,Guangde Z ,Yicheng L .Identification of dangerous driving state based on lightweight deep learning model[J].Computers and Electrical Engineering,2023,105
16. Barbosa W M ,Miranda O D V J ,Oliveira D K L .Identification of antecedents of risky driving behavior of food delivery riders: An analysis during the COVID –19 pandemic in Brazil[J].Case Studies on Transport Policy,2025,20101403-101403.
17. Пан Тун. Розробка носимого пристрою для вимірювання частоти серцебиття на основі STM32 [J]. Електронні технології, 2025, 54(03): 130–133.
18. Song Y ,Ross V ,Ruiter C A R , Song Y ,Ross V ,Ruiter C A R , et al.Development of a framework for risky driving scenario identification, individual risk assessment, and group risk differences estimation using naturalistic driving data from the i-DREAMS project.[J].Accident; analysis and prevention,2025,215107993.
19. Li P ,Liu Z ,Shan H , Li P ,Liu Z ,Shan H , et al.Enhanced Broad-Learning-Based Dangerous Driving Action Recognition on Skeletal Data for Driver Monitoring Systems[J].Sensors,2025,25(6):1769-1769.
20. Hou J ,Zhang B ,Zhong Y , Hou J ,Zhang B ,Zhong Y , et al.Research Progress of Dangerous Driving Behavior Recognition Methods Based on Deep Learning[J].World Electric Vehicle Journal,2025,16(2):62-62.

21. Чжоу Чжунфен, Ван Цзесін. Розробка інтелектуального зовнішнього спального мішка на основі мікроконтролера STM32 [J]. Журнал Хунаньського інженерного університету (натуралістична версія), 2024,34 (04) :32-37+75.DOI: 10.15987/j.cnki.hgbjbz. 2024.04.013.
22. Wan L ,Yan Y ,Zhang C , Wan L ,Yan Y ,Zhang C , et al.Characteristics and identification of risky driving behaviors in expressway tunnels based on behavior spectrum[J].International Journal of Transportation Science and Technology,2024,165-17.
23. Gooch J ,Mahmud S M ,Gross F , Gooch J ,Mahmud S M ,Gross F , et al.Identification of Risk Factors for Severe Younger and Older Driver Crashes in Massachusetts[J].Transportation Research Record,2024,2678(12):1950-1963.
24. Лян Сунчжэн, Цуй Меньюй, Бянь Сіньюн та ін. Розробка та реалізація інтелектуального браслета на основі мікроконтролера STM32 [J]. Китайські нові технології та нові продукти, 2024, (22) :32-35.DOI: 10.13612/j.cnki.cntp. 2024. 22. 006.
25. Weng S ,Chai C ,Yin W , Weng S ,Chai C ,Yin W , et al.Identifying novice drivers in need of hazard perception ability improvement for takeover performance in Level 3 automated driving[J].Accident Analysis and Prevention,2024,208107803-107803.
26. Гао Ліна. Застосування YOLOv8 у розпізнаванні небезпечного водіння [J]. Електронні технології, 2024, 53(07): 404–405.
27. Цзінь Цзіньсінь. Метод виявлення небезпечної поведінки водія на основі оцінки положення тіла [D]. Ухуадзянський університет транспорту, 2024.DOI : 10.27147/d.cnki.ghdju. 2024. 000613.
28. Лю Чженьгуан. Дослідження методу розпізнавання небезпечної поведінки водіїв на основі YOLOv8 [D]. Китайський університет гірничої промисловості, 2024.
29. Ван Веньчэн, Пен Ян, Чжуо Га та ін. Розробка та реалізація системи двомовного тибетсько-китайського озвучення для виявлення водіїв, які керують транспортним засобом у стані втоми або під впливом алкоголю [J].

- Комп'ютерні програмні навички та техніка обслуговування, 2024, (03) : 123-126.DOI: 10.16184/j.cnki.comprg. 2024.03.037.
30. У Цзяньцін, Чжан Цзії, Ван Юйбо та ін. Метод ідентифікації небезпечного поведінки водіїв вантажівок із перегрузкою з урахуванням багатомодальних даних [J]. Транспортна інженерія та інформація, 2024,24 (02) :63-75.DOI: 10.16097/j.cnki.1009-6744. 2024.02.007.
 31. Тянь Хежен. Дослідження ідентифікації небезпечного керування транспортними засобами без двигуна на основі YOLOv8 [D]. Університет Хефей, 2024.DOI: 10.27876/d.cnki.ghfxy. 2024. 000218.
 32. Чжу Ян, Ян Яньнін, Дун Ченьле та ін. Розробка інтелектуального браслета для парку розваг на основі STM32F103C8T6 [J]. Комп'ютерна вимірювання та керування, 2024,32 (09) :213-219+248.DOI: 10.16526/j.cnki.11-4762/tp. 2024. 09. 030.
 33. Лі Мін. Методи розпізнавання небезпечного поведінки водія на основі комп'ютерного зору в умовах інтернету транспортних засобів [J]. Інформація та комп'ютери (теоретичний випуск), 2023, 35(20): 160–162.
 34. Чжан Цуйюй, Ван Цзюньмін, Ху Хайфен. Проектування системи моніторингу електрокардіограми на основі мікроконтролера STM32 [J]. Електронна робота, 2023,31 (19) :30-32.DOI: 10.16589/j.cnki.cn11-3571/tn. 2023. 19. 011.
 35. Ван Цзюньхуа, інтелектуальне виявлення небезпечних поведінок водіїв важливих транспортних засобів, активне втручання та технології широкомасштабного контролю. Шанхай, Університет Тонцзі, 20 липня 2023 року.
 36. Сун Вей. Дослідження методу візуального розпізнавання порушень водіння на основі YOLOv8 [D]. Уханський університет науки та технології, 2023.DOI : 10.27380/d.cnki.gwkju. 2023. 000099.
 37. Гао Юйсян. Розпізнавання небезпечного поведінки водія на основі проекційної твін-векторної машини [D]. Шаньдонський університет промисловості та торгівлі, 2023.DOI: 10.27903/d.cnki.gsdsq. 2023. 000032.

38. Лун Ліньсін, Лань Гочай, Ян Мейхун та ін. Розробка електронного гіпнотика на основі мікроконтролера [J]. Електронні роботи, 2023,31(06) : 107-109+90.DOI: 10.16589/j.cnki.cn11-3571/tn. 2023. 06. 011.
39. Го Сінчжі, Ян Гуйхуа. Проектування та реалізація багатофункціонального інтелектуального годинника на основі мікроконтролера [J]. Електронна робота, 2022,30(22) : 6-9+26.DOI: 10.16589/j.cnki.cn11-3571/tn. 2022. 22. 005.
40. Шу Женьїї, Цай Цзюнь, Цзі Інін та ін. Проектування системи бездротового моніторингу руху на основі мікроконтролера та LabVIEW [J]. Журнал Хубейського університету (натурознавчий випуск), 2023, 45(01): 106–112.
41. Чжан Чжунчень, Чжан Цзюнь. Розробка та реалізація пристрою моніторингу фізичної форми студентів на основі мікроконтролера STM32 [J]. Комп'ютерна вимірювання та керування, 2022,30(02) :292-298.DOI: 10.16526/j.cnki.11-4762/tp. 2022. 02. 042.
42. Люй Юнь, Люй Хао. Проектування системи вимірювання частоти серцебиття та температури тіла на основі STM32 [J]. Механічно-електрична інженерна техніка, 2021, 50(08): 135–137, 207.

ДОДАТКИ

Додаток А

Технічне завдання

на розробку багатомодальної інформаційної системи для виявлення небезпечної поведінки водія

1.1 Загальні положення.

Це технічне завдання визначає вимоги, склад, функціональні можливості та умови розробки багатомодальної інформаційної системи для виявлення небезпечної поведінки водія (далі — Система).

Технічне завдання розроблено відповідно до вимог ДСТУ 3008:2015 «Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлення» та використовується в межах виконання кваліфікаційної роботи за спеціальністю 122 «Комп'ютерні науки».

1.2 Підстава для розробки.

Підставою для розробки Системи є виконання кваліфікаційної роботи з теми, присвяченої проектуванню та розробці багатомодальної інформаційної системи виявлення небезпечної поведінки водія.

1.3 Призначення та мета розробки.

Система призначена для автоматизованого виявлення небезпечної поведінки водія шляхом аналізу багатомодальних даних із використанням методів глибокого навчання.

Метою розробки є підвищення рівня безпеки дорожнього руху шляхом створення інтелектуальної системи, здатної функціонувати в режимі, наближеному до реального часу.

1.4 Вимоги до системи.

1.4.1 Функціональні вимоги.

Система повинна забезпечувати:

- приймання та обробку відеоданих з камер салону автомобіля;
- аналіз динамічних параметрів руху транспортного засобу;
- обробку даних зондування навколишнього середовища;
- автоматичне виявлення небезпечної поведінки водія, зокрема:
 - використання мобільного телефону;
 - непристібнутий ремінь безпеки;
 - відволікання уваги;
 - ознаки втоми або сонливості;
- інтеграцію даних з різних модальностей із використанням механізмів фузії ознак;
- формування та збереження результатів аналізу;
- візуалізацію результатів роботи системи.

1.4.2 Вимоги до інформаційного забезпечення

Вхідними даними є:

- відеопотоки з камер;
- часові ряди динамічних параметрів;
- дані сенсорів навколишнього середовища.

Вихідними даними є:

- класифіковані типи поведінки водія;
- оцінка рівня небезпеки;
- журнал подій системи.

1.5 Вимоги до програмного забезпечення

Для реалізації Системи необхідно:

- використання мови програмування Python;
- застосування фреймворків глибокого навчання PyTorch або TensorFlow;
- використання бібліотек комп'ютерного зору OpenCV;
- забезпечення модульності та можливості масштабування програмного забезпечення.

1.6 Вимоги до апаратного забезпечення

Система повинна функціонувати на обчислювальних засобах, що включають:

- центральний процесор із підтримкою багатопотокової обробки;
- графічний процесор (GPU) для навчання та тестування моделей;
- відеокамери;
- датчики динамічних параметрів руху.

1.7 Вимоги до надійності та безпеки

Система повинна:

- забезпечувати стабільну роботу під час тривалого використання;
- гарантувати цілісність та захист даних;
- обмежувати несанкціонований доступ до інформації.

1.8 Вимоги до інтерфейсу користувача

Інтерфейс користувача має бути:

- інтуїтивно зрозумілим;
- інформативним;
- забезпечувати перегляд поточного стану та історії подій.

1.9 Вимоги до тестування та контролю якості

Передбачається проведення:

- модульного тестування;
- інтеграційного тестування;
- експериментального оцінювання точності та швидкодії системи.

1.10 Стадії та етапи розробки

Розробка Системи здійснюється у такі етапи:

1. Аналіз предметної області.
2. Проектування архітектури системи.
3. Реалізація програмних компонентів.
4. Навчання та тестування моделей.
5. Аналіз результатів та оформлення звітної документації.

1.11 Порядок приймання результатів

Результати розробки приймаються на підставі демонстрації працездатності Системи, аналізу експериментальних результатів та відповідності вимогам цього технічного завдання.

Додаток Б

Код попередньої обробки даних

Цей код реалізує завантаження, попередню обробку та розділення даних зображень. Спочатку визначається функція `preprocess_data` для завантаження даних, коригування розміру зображення та їх нормалізації. Потім використовується `ImageDataGenerator` для підвищення якості даних, щоб покращити здатність моделі до узагальнення. Нарешті, дані розділяються на навчальний, валідаційний та тестовий набори.

```
import os
import cv2
import numpy as np
from sklearn.model_selection import train_test_split
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

Функція попередньої обробки даних

```
def preprocess_data(data_dir, img_size=(224, 224)):
```

Завантаження та попередня обробка зображень

```
:param data_dir: каталог набору даних
:param img_size: розмір зображення
:return: оброблені дані зображення та мітки

images = []
labels = []

categories = os.listdir(data_dir) отримання каталогів усіх категорій

label_map = {cat: індекс для idx, cat у переліку (categories) – створення відображення між категоріями та індексами}

for category in categories:
    category_path = os.path.join(data_dir, category)
    for img_name in os.listdir(category_path):
        img_path = os.path.join(category_path, img_name)
```

```

img = cv2.imread(img_path)
img = cv2.resize (img, img_size)  — змінює розмір зображення
img = img / 255.0 нормалізація
images.append(img)
labels.append(label_map[category])
images = np.array(images)
labels = np.array(labels)
return images, labels підсилення даних
datagen = ImageDataGenerator(
rotation_range=30, випадковий діапазон кутів обертання
width_shift_range=0.2, випадкове горизонтальне зміщення
height_shift_range=0.2, випадкове вертикальне переміщення
shear_range=0.2, міцність на зсув
zoom_range=0.2, випадковий діапазон масштабування
horizontal_flip=True, випадкове горизонтальне перевертання
fill_mode='nearest' — метод заповнення нових пікселів)

```

Завантаження даних

```

data_dir = 'path/to/dataset'
images, labels = preprocess_data(data_dir)

```

Розділення на навчальний, валідаційний і тестовий набори

```

train_images,      temp_images,      train_labels,      temp_labels      =
train_test_split(images, labels, test_size=0.3, random_state=42)

val_images,        test_images,        val_labels,        test_labels      =
train_test_split(temp_images,          temp_labels,          test_size=0.33,
random_state=42)

```

Код навчання моделі

Цей код реалізує навчання моделі на основі ResNet50. Спочатку завантажуються попередньо навчена модель ResNet50, до якої додаються шар глобального середнього пулінгу та повністю зв'язаний шар. Потім заморожуються ваги попередньо навченої моделі, і навчаються лише додані

шари. Далі моделю компілюють і навчають на даних із підвищеним рівнем навчання, після чого зберігають навчану модель.

```
from tensorflow.keras.applications import ResNet50
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, GlobalAveragePooling2D
from tensorflow.keras.optimizers import Adam
```

Завантаження попередньо навченої моделі ResNet50.

```
base_model = ResNet50(weights='imagenet', include_top=False,
input_shape=(224, 224, 3))
```

Створення моделі.

```
model = Sequential()
model.add(base_model)
model.add (GlobalAveragePooling2D()) — шар глобального середнього пулінгу
model.add (Dense(128, activation='relu')) — повністю зв'язаний шар
model.add (Dense (num_classes, activation='softmax')) — вихідний шар
```

Заморожування параметрів моделі з попередньою тренуванням.

```
for layer in base_model.layers:
    layer.trainable = False
```

Модель компіляції.

```
model.compile(optimizer=Adam(learning_rate=0.001),
loss='sparse_categorical_crossentropy', metrics=['accuracy'])
```

Модель навчання.

```
history = model.fit(datagen.flow(train_images, train_labels,
batch_size=32), steps_per_epoch=len(train_images) / 32, epochs=10,
validation_data=(val_images, val_labels))
```

Зберегти модель.

```
model.save('garbage_classification_model.h5')
```

Тестовий код моделі

Цей код реалізує тестування та оцінку моделі. Спочатку завантажуються навчена модель, після чого виконується прогнозування на тестовому наборі. Індeksi точності класифікації, ретроспективності та F1-скоринг обчислюються за допомогою класифікаційного звіту, а матриця змішування — за допомогою матриці змішування для оцінки ефективності моделі.

```
import os
import numpy as np
import json
import logging
import matplotlib.pyplot as plt
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing.image import img_to_array
from sklearn.metrics import (classification_report,
                             confusion_matrix, accuracy_score)
from sklearn.preprocessing import LabelEncoder
import seaborn as sns
from typing import Tuple, Dict, List
```

Налаштування журналів та глобальних параметрів (для сценарії небезпечного водіння)

Налаштування журналу: фіксування процесу тестування, повідомлень про помилки та результатів оцінювання.

```
logging.basicConfig(level=logging.INFO, format="%(asctime)s -
%(levelname)s - %(message)s", handlers=[logging.FileHandler (
"model_test.log"), запис у файл журналу logging.StreamHandler()])
logger = logging.getLogger(__name__)
```

Глобальні параметри: мають відповідати налаштуванням на етапі навчання або попередньої обробки.

```
GLOBAL_CONFIG = {
```

"model_path": "../yolov8_danger_driving.h5", шлях моделі небезпечного водіння (замінити на фактичний шлях)

«label_map_path»: «. /label_map.json», шлях до таблиці відображення категорій (створена на етапі попередньої обробки)

"test_data_path": "../preprocessed_data/test" — шлях до попередньо обробленого тестового набору (у форматі пру)

img_size: (224,224), розмір вхідних даних моделі (у тій самій масштабі, що й під час навчання) img_size: (224,224), розмір вхідних даних моделі (у тій самій масштабі, що й під час навчання)

«confidence_threshold»: 0,5 — поріг впевненості прогнозу (для сценарії небезпечного водіння рекомендується $\geq 0,5$) «confidence_threshold»: 0,5 — поріг впевненості прогнозу (для сценарії небезпечного водіння рекомендується $\geq 0,5$)

«save_fig_path»: «. /test_results» — шлях для збереження графіка оцінки «save_fig_path»: «. /test_results» — шлях для збереження графіка оцінки

"save_report_path": шлях для збереження класифікованого звіту —
 "../test_results/report.txt"

}

Створити каталог для збереження результатів (якщо він ще не існує)

```
os.makedirs(GLOBAL_CONFIG["save_fig_path"], exist_ok=True)
```

Функції інструментів (адаптовані до особливостей даних про небезпечне водіння)

```
def load_label_map(label_map_path: str) -> Tuple[Dict[str, int], List[str]]:
```

Завантаження таблиці мапування категорій (файл у форматі JSON, збережений на етапі попередньої обробки)

Повернутися: `label_map` (ім'я категорії $\rightarrow$ індекс), `class_names` (індекс $\rightarrow$ ім'я категорії, використовується для звітів про оцінку)

```
try:
```

```
with open(label_map_path, "r", encoding="utf-8") as f:
```

```
label_map = json.load(f)
```

Створити список імен категорій (відсортовані за індексом, щоб відповідали результатам моделювання)

```
class_names = [k for k, v in sorted(label_map.items(), key=lambda
x: x[1])]
```

```
logger.info(f"Таблиця відображення категорій завантажена, загалом
{len(class_names)} категорій небезпечного водіння: {class_names}")
```

```
return label_map, class_names
```

```
except FileNotFoundError:
```

```
logger.error(f"Таблиця перетворення категорій відсутня:
{label_map_path}, виконайте попередню обробку даних")
```

```
raise
```

```
except json.JSONDecodeError:
```

```
logger.error(f"Помилка у форматі таблиці відображення категорій:
{label_map_path}, перевірте синтаксис JSON")
```

```
raise
```

```
def load_preprocessed_test_data(test_data_path: str) ->
Tuple[np.ndarray, np.ndarray]:
```

Завантажте попередньо оброблений тестовий набір (у форматі прун, щоб уникнути повторної обробки)

Набір тестових сцен для випробування небезпечного керування автомобілем містить: зображення (після нормалізації) та реальні метки (загальні числові коди).

Завантаження даних зображення (shape: [кількість зразків, висота, ширина, кількість каналів])

```
test_images = np.load(os.path.join(test_data_path,
"test_images.npy"))
```

Завантаження реальних міток (shape: [кількість екземплярів], цілочисельне кодування).

```
test_labels = np.load(os.path.join(test_data_path,
"test_labels.npy"))
```

Перевірка дійсності даних (щоб уникнути невідповідності вимірів). Підтвердити, що test_images.ndim дорівнює 4, f"Димензія зображення в тестовому наборі є неправильною — має бути 4-вимірною (кількість зразків, висота, ширина, канали), фактично — {test_images.ndim}-вимірною". підтвердити, що test_labels.ndim == 1, f"Розмір мітки тестового набору є неправильним — має бути одновимірним, фактично — {test_labels.ndim}-вимірним".

```
assert len(test_images) == len(test_labels), f"Кількість зображень не
відповідає кількості міток: {len(test_images)} проти {len(test_labels)}"
```

```
logger.info(f"Тестовий набір завантажено: {len(test_images)} зразків, розмір
зображення: {test_images.shape[1:3]}")
```

```
return test_images, test_labels
```

```
except FileNotFoundError as e:
```

```
logger.error(f"Файл тестового набору відсутній: {str(e)}, перевірте шлях
або повторно виконайте попередню обробку даних")
```

```
raise except AssertionError as e:
```

```
logger.error(f"Дані тестового набору є неправильними: {str(e)}")
```

```
raise
```

```
def load_danger_driving_model(model_path: str) ->
"tensorflow.keras.Model":
```

Завантаження моделі розпізнавання небезпечного водіння (вирішення проблеми сумісності налаштованих шарів і ваг)

Якщо використовується власна архітектура YOLOv8, необхідно вказати custom_objects.

Якщо модель містить власний шар YOLOv8 (наприклад, шар Detect), необхідно додати `custom_objects` (наприклад, нижче).

Залежно від налаштування моделі, яку використовується для навчання, якщо відповідного власного шару немає, можна безпосередньо завантажити модель.

```
custom_objects = {}
```

Наприклад: якщо використовується шар Detect у YOLOv8, потрібно додати:

```
from ultralytics.nn.modules import Detect
custom_objects["Detect"] = Detect
model = load_model(model_path, custom_objects=custom_objects)
model.summary (print_fn=lambda x: logger.info(x)) — виводити
структуру моделі в журнал
logger.info(f"Модель небезпечного водіння завантажена:
{model_path}")
return model
except FileNotFoundError:
logger.error(f"Файл моделі не існує: {model_path}")
raise
except Exception as e:
logger.error(f"Не вдалося завантажити модель (можливо, не вказано
власний шар): {str(e)}")
raise

def analyze_low_confidence_samples(
test_images: np.ndarray,
predictions: np.ndarray,
predicted_classes: np.ndarray,
test_labels: np.ndarray,
class_names: List[str],
```

```
threshold: float = 0.5
```

```
) -> None:
```

Аналіз зразків прогнозування з низьким рівнем достовірності (особливо важливо враховувати випадки пропущення або помилкового виявлення у сценаріях небезпечного водіння).

Виводити індекси зразків із низьким рівнем впевненості, справжню та прогнозовану категорії та відповідний рівень впевненості.

Розрахунок максимальної довірчої межі прогнозу для кожного вибірки

```
max_confidences = np.max(predictions, axis=1)
```

```
Відбір зразків із низьким рівнем впевненості (рівень впевненості <
поріг)
```

```
low_conf_mask = max_confidences < threshold
```

```
low_conf_indices = np.where(low_conf_mask)[0]
```

```
if len(low_conf_indices) == 0:
```

```
logger.info(f"Усі прогнозовані значення для зразків мають рівень
впевненості ≥{threshold}, без зразків із низьким рівнем
впевненості")
```

```
return
```

Вивести деталі зразків із низьким рівнем достовірності.

```
logger.warning(f"Виявлено {len(low_conf_indices)} зразків із
низьким рівнем впевненості (рівень впевненості < {threshold}):")
```

```
low_conf_report = "Деталі зразків із низьким рівнем достовірності:"
```

```
low_conf_report += f"'індекс':<6>«реальна
категорія»:<12>«прогнозована
впевненості»:<8>\n"
```

```
low_conf_report += "-" * 50 + "\n"
```

```
Для idx у low_conf_indices[:10]: виводити перші 10 (щоб уникнути
надмірної довжини журналу)
```

```
true_class = class_names[test_labels[idx]]
```

```
pred_class = class_names[predicted_classes[idx]]
```

```
conf = max_confidences[idx]
```

```
low_conf_report += f"{idx:<6} {true_class:<12} {pred_class:<12}
{conf:.4f}\n"
```

Необов'язково: зберегти зображення зразків із низьким рівнем достовірності (для подальшого аналізу)

```
low_conf_img = test_images[idx] * 255 — нормалізація (від 0 до 255)
low_conf_img = img_to_array(low_conf_img, dtype=np.uint8)
save_path = os.path.join(GLOBAL_CONFIG["save_fig_path"],
f"low_conf_sample_{idx}.jpg")
plt.imsave(save_path, low_conf_img)
```

Зберегти повний звіт із низьким рівнем достовірності у файл.

```
with open(os.path.join(GLOBAL_CONFIG["save_fig_path"],
"low_conf_report.txt"), "w", encoding="utf-8") as f:
f.write(low_conf_report)

logger.info(f"Звіт із низьким рівнем достовірності збережено в:
{GLOBAL_CONFIG['save_fig_path']}/low_conf_report.txt")

def plot_confusion_matrix(test_labels: np.ndarray,
predicted_classes: np.ndarray, class_names: List[str], save_path:
str) -> None:
```

Створення теплової карти матриці змішування (інтуїтивне відображення помилкових класифікацій категорій небезпечного водіння).

Вказуйте кількість зразків і їхню частку для кожного клітинки, підтримується збереження зображення у високій чіткості.

Розрахунок матриці змішування.

```
cm = confusion_matrix(test_labels, predicted_classes)
```

Розрахунок частки кожного рядка (для аналізу частоти помилкових оцінок за категоріями).

```
cm_ratio = cm.astype("float") / cm.sum(axis=1)[:, np.newaxis]
```

Встановити розмір полотна (адаптується до різних сценаріїв).

```
plt.figure(figsize=(10, 8))
```

Створити теплову карту

```
sns.heatmap(cm, annot=True, вказати значення annot=True, вказати
значення
```

```

fmt="d", формат числа (ціле число) fmt="d", формат числа (ціле
число) cmap="Blues", кольорова схема xticklabels=class_names,
yticklabels=class_names,
cbar_kws={"label": "Кількість вибірок"})

```

Показати частку на тепловому графіку (маленьким шрифтом).

```

for i in range(len(cm)):
for j in range(len(cm)):

```

Якщо $cm[i, j] > 0$: відсоток вказується лише для ненульових клітин

```

plt.text(j + 0.5, i + 0.5, f"\n({cm_ratio[i, j]:.2%})",
ha="center", va="center", color="white" if cm[i, j] >
cm.sum(axis=1)[i] * 0.5 else "black", fontsize=8)

```

Налаштувати теги та заголовки (для сценарії небезпечного водіння).

```

plt.xlabel («Прогнозовані категорії», шрифт=12)
plt.ylabel («реальна категорія», шрифт=12)
plt.title («матриця змішування моделі ідентифікації небезпечних
поведінкових дій у водінні», шрифт=14, збільшення=20)
plt.xticks (зміщення = 45, ha="right") – коли назва категорії надто
довга, використовується зміщення
plt.tight_layout() – автоматична налаштування компоновання

```

Зберегти зображення у форматі HD.

```

plt.savefig(os.path.join(save_path, "confusion_matrix.png"),
dpi=300, bbox_inches="tight")
plt.close()
logger.info(f"Тепловий графік матриці змішування збережено в:
{save_path}/confusion_matrix.png")

```

Анотація

Цай Б. – Проєктування та розробка багатомодальної інформаційної системи виявлення небезпечної поведінки водія на основі глибоких нейронних мереж. – Рукопис.

Кваліфікаційна робота на здобуття освітнього ступеня «магістр» за спеціальністю 122 Комп'ютерні науки, освітньої програми Комп'ютерні науки та інформаційні технології. – Волинський національний університет імені Лесі Українки. – 2025 р.

У кваліфікаційній роботі розглянуто проблему підвищення безпеки дорожнього руху шляхом автоматизованого виявлення небезпечної поведінки водія з використанням сучасних методів глибокого навчання та багатомодального аналізу даних. Актуальність дослідження зумовлена зростанням кількості дорожньо-транспортних пригод, значна частина яких спричинена людським фактором, зокрема втомою водія, використанням мобільного телефону, ігноруванням ременя безпеки та агресивними манерами керування.

У роботі спроектовано та реалізовано багатомодальну інформаційну систему виявлення небезпечної поведінки водія на основі глибоких нейронних мереж. Запропонований підхід передбачає інтеграцію візуальних даних з автомобільних камер, динамічних параметрів руху транспортного засобу та даних зондування навколишнього середовища. Для виділення ознак застосовано модель YOLOv8 у поєднанні з іншими нейронними мережами, зокрема ResNet, LSTM та 1D-CNN, що дозволяє ефективно аналізувати як просторові, так і часові характеристики поведінки водія.

Особливу увагу приділено підготовці та анотації даних із використанням напівкерованого навчання, що дає змогу зменшити трудомісткість ручного маркування великих обсягів даних. Запропоновано механізм фузії ознак з використанням міжмодальної уваги, який забезпечує формування інтегрованого представлення з високою дискримінаційною здатністю. Проведено експериментальне оцінювання розробленої системи, яке підтвердило її

ефективність у розпізнаванні основних типів небезпечної поведінки водія в умовах, наближених до реального часу.

Отримані результати свідчать про доцільність застосування багатомодальних нейронних мереж у складі інтелектуальних транспортних систем і можуть бути використані для підвищення рівня активної безпеки транспортних засобів.

Ключові слова: багатомодальні дані, небезпечна поведінка водія, комп'ютерний зір, глибоке навчання, YOLOv8, фузія ознак, інтелектуальні транспортні системи.

Abstract.

Tsai B. - Design and Development of a Multimodal Information System for Detecting Dangerous Driver Behavior Based on Deep Neural Networks. -
Manuscript.

Qualification work for the degree of "Master" in the field of Computer Science, educational program "Computer Science and Information Technologies." – Lesya Ukrainka Volyn National University. – 2025.

This qualification work addresses the problem of improving road traffic safety through the automated detection of dangerous driver behavior using modern deep learning methods and multimodal data analysis. The relevance of the study is driven by the growing number of road traffic accidents, a significant proportion of which are caused by human factors such as driver fatigue, mobile phone usage, failure to wear a seat belt, and aggressive driving maneuvers.

The work proposes the design and implementation of a multimodal information system for detecting dangerous driver behavior based on deep neural networks. The proposed approach integrates visual data from in-vehicle cameras, dynamic vehicle motion parameters, and environmental sensing data. Feature extraction is performed using the YOLOv8 computer vision model in combination with other neural network architectures, including ResNet, LSTM, and 1D-CNN, enabling effective analysis of both spatial and temporal characteristics of driver behavior.

Special attention is paid to data preparation and annotation using semi-supervised learning techniques, which significantly reduce the cost and labor intensity of manual labeling for large-scale datasets. A feature fusion mechanism based on cross-modal attention is proposed, providing an integrated and highly discriminative feature representation. Experimental evaluation of the developed system demonstrates its effectiveness in recognizing key types of dangerous driver behavior under near real-time conditions.

The obtained results confirm the feasibility and practical value of applying multimodal deep learning models in intelligent transportation systems and highlight their potential for enhancing active vehicle safety and driver assistance technologies.

Keywords: multimodal data, dangerous driver behavior, computer vision, deep learning, YOLOv8, feature fusion, intelligent transportation systems.