Телюк Артем Михайлович, студент магістратури

Національний університет “Львівська політехніка”, м. Львів

ORCID 0009**-**0000**-**1180**-**8492

**ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ ДОПОМОГИ ВОДІЮ
НА ОСНОВІ КОМП'ЮТЕРНОГО ЗОРУ ТА ГЛИБИННОГО НАВЧАННЯ**

Вступ. Актуальність теми. Системи допомоги водію (ADAS) відіграють ключову роль у підвищенні безпеки дорожнього руху, надаючи водіям критичну інформацію в режимі реального часу. Інформаційні системи в ADAS допомагають водіям краще орієнтуватися, залишаючи відповідальність за керування автомобілем. Незважаючи на потенціал повної автоматизації, сучасні технології все ще вимагають участі водія. Тому вдосконалення інформаційних систем ADAS є важливим для зниження кількості аварій та покращення обізнаності водіїв.

Формулювання проблеми. Деякі підсистеми ADAS, такі як розпізнавання дорожніх знаків та виявлення вибоїн, недостатньо розвинені для застосування в реальному часі. Крім того, системи розпізнавання смуг руху, транспортних засобів та пішоходів вимагають значних обчислювальних ресурсів, що ускладнює їх використання на недорогих пристроях із обмеженими можливостями. Виклик полягає у створенні ефективної інтегрованої системи, що працює на апаратному забезпеченні з низькими ресурсами без втрати точності та швидкості.

Мета дослідження. Метою статті є оцінка існуючих методів в ADAS та розробка нових підходів для підвищення їх ефективності та точності. Основна увага приділяється вдосконаленню розпізнавання смуг руху, дорожніх знаків і вибоїн із використанням моделей глибокого навчання, таких як YOLOv8n. Пропонується створення модульної архітектури системи з паралельною обробкою для покращення роботи в реальному часі на обмежених ресурсах.

Огляд літератури. Розвиток систем допомоги водію (ADAS) досяг значних успіхів у виявленні смуг руху, розпізнаванні об'єктів і моніторингу стану дорожнього покриття, які є критичними для підвищення безпеки водіїв. Виявлення смуг руху стикається з труднощами в умовах поганої погоди та нерівномірної розмітки. Kaur і Kumar [1] виявили обмеження традиційних методів, таких як перетворення Хафа, які показують низьку точність у реальних умовах. Для вирішення цих проблем Rachel та ін. [3] запропонували підхід на основі CNN, що забезпечує понад 97% точності навіть у складних ситуаціях. Saha та ін. [2] розробили методику flood**-**fill, яка адаптується до змін освітлення, але обмежена в екстремальних умовах.

Розпізнавання об'єктів є ключовою складовою ADAS, особливо для систем попередження зіткнень. Murthy та ін. [4] використовували YOLOv5 для створення реального часу системи, яка ефективно визначає пішоходів, транспортні засоби та інші перешкоди. Цей підхід перевершує старіші моделі, як**-**от R**-**CNN, за швидкістю та точністю, що робить його придатним для застосувань у реальному часі. Розпізнавання дорожніх знаків також є важливою функцією ADAS. Golgire [8] досліджував використання CNN для цієї задачі, показуючи здатність алгоритмів ефективно працювати в умовах слабкої видимості та змін освітлення при мінімальній попередній обробці даних. Оптимізація моделей через підбір гіперпараметрів дозволяє працювати на пристроях із обмеженими ресурсами, що важливо для реального часу [8]. Проте в недорогих ADAS рішеннях реальне розпізнавання знаків часто замінюється попередньо завантаженими базами даних. Виявлення вибоїн, як нова функція ADAS, спрямоване на моніторинг стану доріг. Buza та ін. [5] запропонували економічно ефективний метод на основі обробки зображень і спектральної кластеризації з точністю 81%. Joe та ін. [6] інтегрували виявлення вибоїн із Tiny YOLOv3 у мобільний додаток, що дозволяє моніторити стан доріг у реальному часі. Проте досягнення стабільної точності за різного освітлення залишається викликом [6]. Крім того, опції на основі смартфонів показують меншу реальну здатність у порівнянні з більш досконалими OEM**-**системами. Загалом дослідження підкреслюють еволюцію технологій ADAS: традиційні методи забезпечують базовий інструментарій, тоді як моделі на основі AI, такі як CNN та YOLO, суттєво покращують продуктивність і точність у реальному часі. Проте виклики оптимізації для середовищ із обмеженими ресурсами залишаються, що особливо важливо для доступних або вторинних ADAS**-**рішень [17][18].

Результати досліджень. Система попередження зіткнень в реальному часі попереджає водіїв про можливі зіткнення. Вона використовує комбінацію сенсорів, візуальних технологій та моделей машинного навчання для виявлення об'єктів та оцінки ризику зіткнення. Основна увага приділяється візуальним технологіям, які забезпечують ефективність та економічність у різних умовах.

Камера є основним сенсором для таких систем завдяки її адаптивності та низькій вартості. Найчастіше використовуються монокулярні камери, які знімають відео в реальному часі для аналізу об'єктів та оцінки відстаней. Для обробки даних застосовується обчислювальний пристрій, наприклад NVIDIA Jetson, який використовує моделі машинного навчання. Попередження подається через дисплей або аудіосистему.

Відео, зняте камерою, проходить етапи попередньої обробки: корекцію кольору, зміну розміру та зменшення шуму. Далі кадри аналізуються моделлю об'єктного виявлення, наприклад YOLO, яка визначає об'єкти, їхні класи та координати. На основі отриманих даних система оцінює ймовірність зіткнення, враховуючи відстань, швидкість і траєкторії об'єктів. У разі ризику зіткнення водію подається візуальне або звукове попередження.

Об'єктне виявлення за допомогою YOLO. Модель YOLO ефективна для реального часу завдяки швидкості (до 60 кадрів на секунду на потужних GPU) та точності. Модель ділить зображення на сітку, прогнозуючи прямокутники та ймовірності класів для кожної клітинки. YOLOv8, завдяки своїй оптимізації, забезпечує швидке виконання навіть на пристроях з обмеженими ресурсами, таких як Raspberry Pi 4. Зокрема, у форматі NCNN модель демонструє найкращі результати з часом обробки 414,73 мс на зображення.

Оцінка відстані. Для оцінки відстані часто використовується метод перспективної трансформації. Він дозволяє швидко та з мінімальними витратами ресурсів оцінювати відстані між транспортним засобом і об'єктами, базуючись на розмірі та положенні об'єктів на зображенні. Такий підхід особливо корисний для систем із монокулярними камерами.

Відстеження рухомих об'єктів у реальному часі забезпечується алгоритмами, такими як фільтр Калмана або SORT. Вони дозволяють точно оцінювати траєкторії об'єктів та розраховувати час до зіткнення за формулою $TTC =\frac{D}{V\_{r}}$, де D — відстань до об'єкта, $V\_{r} $— відносна швидкість. Якщо TTC менше критичного порогу, водію подається попередження.

Система розпізнавання смуг руху. Система розпізнавання смуг руху допомагає підтримувати автомобіль у межах смуги. Для цього застосовуються два основні підходи: традиційний комп’ютерний зір (CV) та глибоке навчання (DL).

Традиційний підхід (CV) – цей підхід використовує методи обробки зображень, такі як фільтрація, порогова обробка та алгоритми пошуку ліній. Переваги методу — висока швидкість і придатність для пристроїв з обмеженими ресурсами. Однак він чутливий до шуму і змін умов освітлення. Попередні етапи включають корекцію викривлення зображення, виділення області інтересу та згладжування шуму за допомогою фільтрів. Виявлення смуг здійснюється через методи, такі як перетворення Хафа, що дозволяють виявляти лінії за точками зображення. Підхід DL використовує сегментаційні моделі, такі як U**-**Net, які розділяють зображення на регіони смуг і несмуг. Моделі навчаються на великих наборах даних, що дозволяє їм працювати у складних умовах, зберігаючи високу точність. Однак вони вимагають більше ресурсів, хоча оптимізації, такі як прунінг або квантізація, роблять моделі придатними для пристроїв, як**-**от Raspberry Pi. Традиційний підхід менш вимогливий до обчислювальних ресурсів, але менш ефективний у складних умовах. Підхід DL забезпечує вищу точність, однак потребує значних обчислювальних потужностей і оптимізації для роботи в реальному часі.

Запропонований підхід: виявлення об’єктів для розпізнавання смуг руху. Запропонований метод використовує моделі виявлення об'єктів (OD), зокрема архітектуру YOLO, для вирішення завдання розпізнавання смуг руху. Лейблінг виконується у вигляді точок, які представляють смуги, а модель прогнозує їх як об’єкти у межах прямокутних областей. Кожна точка лейблу позначається прямокутною рамкою (bounding box), розміри якої налаштовуються залежно від характеристик набору даних. Для усунення надмірності використовується гіперпараметр, який контролює видалення областей з високим рівнем перекриття.

Метод базується на структурі набору даних CULane, де смуги поділяються на чотири класи: дальня права (RR), близька права (RC), близька ліва (LC) та дальня ліва (LL). Це дозволяє моделі прогнозувати як наявність смуги, так і її клас, що підвищує точність. Після навчання модель прогнозує прямокутники для точок смуг, які потім об'єднуються в лінії шляхом з'єднання їхніх центрів. Цей підхід є ефективним, особливо для пристроїв із обмеженими ресурсами. Модель YOLO може прогнозувати межі смуг навіть за відсутності чітких дорожніх ліній, використовуючи контекстні підказки. Це значна перевага перед традиційними методами, які часто не працюють у складних умовах, таких як погана видимість або неповні розмітки.

CULane — великий набір даних для досліджень у сфері розпізнавання смуг руху, який містить 133,235 кадрів, зібраних на шести автомобілях у різних умовах у Пекіні. Він поділений на 88,880 кадрів для навчання, 9,675 — для валідації, та 34,680 — для тестування. Тестовий набір включає категорії, такі як нормальні умови, тіні, відсутність розмітки та інтенсивний рух [16]. Кожен кадр містить вручну створені анотації навіть для смуг, які приховані транспортними засобами або погано видно, що дозволяє моделі прогнозувати смуги в складних умовах.

Для оцінки ефективності моделі YOLOv8n для розпізнавання смуг руху було проведено попереднє навчання на наборі даних із використанням фреймворку Ultralytics YOLOv8.2.79, Python 3.11.8 та PyTorch 2.2.1 на GPU NVIDIA GeForce RTX 3080 Ti. Модель, що має 168 шарів та 3,006,428 параметрів, обробляє 8.1 GFLOPs. Навчання тривало 75 епох (7.708 годин) із використанням оптимізатора SGD (швидкість навчання 0.01, момент 0.9) та зображеннями розміром 640 x 640 пікселів. Метрики оцінки включали Precision (P), Recall (R), mAP50 і mAP50**-**95. Результати показали високу точність для близьких смуг: Precision досяг 0.763 (LC) і 0.659 (RC), а mAP50 – 0.872 (LC) і 0.823 (RC). Далека права смуга (RR) мала значно нижчі показники (Precision 0.485, mAP50**-**95 0.295). Це підтверджує потенціал моделі для задач реального часу, але потребує вдосконалення для виявлення далеких смуг.

Система розпізнавання дорожніх знаків. Розпізнавання дорожніх знаків є важливою частиною ADAS, забезпечуючи водіїв інформацією про правила дорожнього руху та небезпеки. Багато систем використовують бази даних з попередньо завантаженими знаками для певних регіонів, проте ці бази мають обмеження: дані можуть бути застарілими, а реальні зміни (нові чи тимчасові знаки) не враховуються.

Для усунення недоліків статичних баз застосовується виявлення знаків у реальному часі за допомогою моделей, таких як YOLOv8. Процес включає два етапи: спочатку модель виділяє знаки із зображення, а потім класифікує їх за типом (обмеження швидкості, попередження тощо). Основна складність полягає у великій кількості типів знаків (понад 200), що ускладнює моделі для систем із обмеженими ресурсами.

Для зниження навантаження процеси виділення та класифікації знаків розділяють. Модель спочатку виявляє лише потенційні знаки, а класифікація виконується окремо або за потреби. У системах із низькими ресурсами можна просто показувати зображення знака водію без класифікації, що зберігає продуктивність.

DFG Traffic Sign Dataset містить 7,000 зображень із 200 категоріями знаків, зібраних у Словенії. Додатково створено набір із понад 30,000 аугментованих знаків, що підвищує стійкість моделі до складних умов.

Модель YOLOv8n навчалася 50 епох на DFG Traffic Sign Dataset із параметрами: оптимізатор AdamW (lr=0.002, momentum=0.9), розмір зображень 640x640 пікселів. Результати: Precision 0.977, Recall 0.956, mAP50 0.983, mAP50**-**95 0.854, швидкість інференсу 1.8 мс/зображення.

Система виявлення вибоїн. Система реального часу для виявлення вибоїн попереджає водіїв про пошкодження дороги за допомогою YOLOv8n. Основна проблема — відсутність відповідних датасетів, тому комбінували існуючі набори даних із власними, зібраними у Львові та Брюховичах.

 Остаточний набір включає понад 900 зображень із вибоїнами, анотації виконано у Roboflow. Додано аугментації для підвищення різноманітності: геометричні трансформації, зміна яскравості та контрасту.

Модель навчалась 100 епох із параметрами: оптимізатор AdamW (lr=0.002, momentum=0.9), розмір зображень 640x640 пікселів. Результати: Precision 0.782, Recall 0.641, mAP50 0.719, mAP50**-**95 0.385, швидкість інференсу 1.2 мс/зображення.

Передача інформації водію. Одним із головних викликів у впровадженні системи виявлення вибоїн на мінікомп’ютерах є швидкість обробки даних. Модель YOLOv8n оптимізована для реального часу, але навіть на обмежених ресурсах потрібен час на обробку кожного зображення. Інформація про вибоїни має передаватися швидко та ефективно, щоб забезпечити безпеку. Використання екранних попереджень, як**-**от на панелі приладів або HUD, не є ідеальним, оскільки це може відволікати водія під час критичної ситуації.

Пропонується замінити екранні попередження на інтуїтивну і менш відволікаючу систему: світлодіодну (LED) стрічку під лобовим склом у поєднанні з аудіопопередженнями. Якщо система виявляє вибоїну зліва, відповідна секція LED стрічки засвітиться червоним кольором із супровідним звуковим сигналом. Це дозволяє водієві інтуїтивно зрозуміти, де знаходиться небезпека, і сфокусуватися на дорозі в потрібному напрямку. LED стрічка відображає приблизне положення вибоїни відносно центру автомобіля, що дозволяє швидко реагувати без зайвих відволікань.

Наступні кроки — об’єднана система. Пропонується інтегрована система, яка об’єднує всі компоненти (попередження зіткнень, розпізнавання дорожніх знаків, смуг руху та вибоїн) у єдину систему допомоги водію. Її мета — виконувати аналіз зображень у реальному часі та надавати інформацію водієві, використовуючи пристрої з обмеженими ресурсами, такі як NVIDIA Jetson Nano або Xavier. При достатній оптимізації систему можна впровадити навіть на Raspberry Pi, що стане проривом у продуктивності на обмеженому обладнанні. Система складається з кількох ключових модулів. Камера, встановлена на панелі приладів, забезпечує безперервне захоплення відео. Це відео проходить попередню обробку, включаючи корекцію спотворень, обрізку та фільтрацію. Центральним елементом є універсальна модель YOLO, навчена для виявлення об’єктів (транспортних засобів, пішоходів, дорожніх знаків, вибоїн і смуг) у реальному часі. Для підвищення продуктивності кілька інстанцій YOLO можуть працювати паралельно, обробляючи різні кадри. Трекер об’єктів, як**-**от SORT або фільтр Калмана, дозволяє відслідковувати динамічні об’єкти, а перспектива перетворюється у вигляд «з висоти пташиного польоту» для кращого просторового розуміння. Додаткові модулі обробляють оцінку відстані та кривини смуг. Фінальний кадр формується з усіх вихідних даних, а модуль попереджень генерує сигнали при необхідності.

Для досягнення продуктивності в реальному часі система використовує модульну архітектуру, де кожен компонент працює незалежно і паралельно. Це зменшує затримку, усуваючи необхідність послідовної обробки. Наприклад, модуль попередньої обробки може працювати одночасно з моделлю об’єктного виявлення, тоді як модуль попереджень працює паралельно із трекером об’єктів. Багатопотоковість забезпечує масштабованість системи відповідно до доступних апаратних ресурсів. Якщо ресурси дозволяють, кілька інстанцій моделі YOLO можуть працювати одночасно, обробляючи різні кадри для підвищення FPS. У результаті система є модульною, масштабованою та адаптованою до апаратного забезпечення, забезпечуючи ефективність і продуктивність у реальному часі.

Запропонований підхід є перспективним, проте має значні виклики, зокрема оптимізацію для апаратного забезпечення з обмеженими ресурсами. Наприклад, запуск кількох інстанцій моделі YOLOv8n на Raspberry Pi вимагає агресивних методів оптимізації, таких як прунінг, квантизація та багатопотокова обробка. Особливу увагу потрібно приділити зниженню затримки модулів, насамперед модулів виявлення та трекінгу об'єктів, які найбільш вимогливі до ресурсів. Успішна реалізація цих оптимізацій дозволить створювати системи допомоги водію в реальному часі навіть на недорогих апаратних платформах, що суттєво підвищить доступність таких систем для масового ринку.

Висновки. Запропоновано вдосконалену систему ADAS, яка інтегрує компоненти реального часу: попередження зіткнень, розпізнавання смуг, дорожніх знаків та вибоїн, з використанням сучасних технологій машинного навчання. Модульна архітектура системи забезпечує роботу на пристроях із обмеженими ресурсами, таких як Raspberry Pi.

Система розпізнавання смуг покращена завдяки моделям виявлення об'єктів YOLO, що мінімізує попередню обробку. Розпізнавання дорожніх знаків розділяє процеси виявлення та класифікації для балансу між точністю та швидкістю. Метод виявлення вибоїн із YOLOv8 показав ефективність у реальному часі, а LED**-**стрічки як індикатори мінімізують відволікання водія.

Попередні результати підтверджують потенціал системи, проте потребують оптимізації моделей для вбудованих систем. Подальша робота зосередиться на підвищенні точності в складних умовах, оптимізації архітектури та розширенні датасетів. Успішна реалізація зробить ADAS доступнішою та підвищить безпеку дорожнього руху.

**Література**

1. Kaur, G., & Kumar, D. (2015). Lane detection techniques: A review. International Journal of Computer Applications, 112(10), 1**-**8.
2. Saha, A., Roy, D. D., Alam, T., & Deb, K. (2012). Automated road lane detection for intelligent vehicles. Global Journal of Computer Science and Technology, 12(6), 1**-**6.
3. Rachel, M. J. S., Kalaiselvi, S., & Salini, R. (2020). Lane detection using neural networks. International Research Journal of Engineering and Technology, 7(3), 3578**-**3582.
4. Murthy, J. S., Siddesh, G. M., Lai, W.**-**C., Parameshachari, B. D., Patil, S. N., & Hemalatha, K. L. (2022). ObjectDetect: A real**-**time object detection framework for advanced driver assistance systems using YOLOv5. Wireless Communications and Mobile Computing, 2022, 1**-**10. https://doi.org/10.1155/2022/9444360
5. Buza, E., Omanovic, S., & Huseinovic, A. (n.d.). Pothole detection with image processing and spectral clustering. Recent Advances in Computer Science and Networking, 48**-**53.
6. Joe, H., Blessingh, J., & Cherian, J. (2020). An intelligent pothole detection system using deep learning. International Research Journal of Engineering and Technology, 7(2), 1591**-**1594.
7. Jumaa, B. A., Abdulhassan, A. M., & Abdulhassan, A. M. (2019). Advanced driver assistance system (ADAS): A review of systems and technologies. International Journal of Advanced Research in Computer Engineering & Technology, 8(6), 231**-**234.
8. Golgire, V. (2021). Traffic sign recognition using machine learning: A review. International Journal of Engineering Research & Technology (IJERT), 10(5), 872**-**876.
9. Tyagi, H., Saroj, V. K., Shahzad, M., & Agarwal, A. (2023). Evolution of YOLO: Exploring the advancements in YOLOv8 for real**-**time wildlife detection. Journal of Computer Vision in Wildlife Monitoring, 1(2), 1**-**10.
10. Tabernik, D., & Skočaj, D. (2019). Deep learning for large**-**scale traffic**-**sign detection and recognition. IEEE Transactions on Intelligent Transportation Systems, 1524**-**9050. https://doi.org/10.1109/TITS.2019.2913588
11. Make ML. (n.d.). Potholes dataset. https://makeml.app/datasets/potholes
12. Jocher, G., Chaurasia, A., & Qiu, J. (2023). Ultralytics YOLOv8 (Version 8.0.0) [Computer software]. https://github.com/ultralytics/ultralytics
13. Rezaei, M., Terauchi, M., & Klette, R. (2015). Robust vehicle detection and distance estimation under challenging lighting conditions. IEEE Transactions on Intelligent Transportation Systems. https://doi.org/10.1109/TITS.2015.2421482
14. Budagam, D., Kumar, A., Ghosh, S., Shrivastav, A., Imanbayev, A., Akhmetov, I., Kaplun, D., Antonov, S., Rychenkov, A., Cyganov, G., & Sinitca, A. (2024). Instance segmentation and teeth classification in panoramic X**-**rays. *arXiv*. <https://doi.org/10.48550/arXiv.2406.03747>
15. Haque, M. R., Islam, M. M., Alam, K. S., Iqbal, H., & Shaik, M. E. (2019). A computer vision**-**based lane detection approach. *Khulna University of Engineering & Technology*. Received: 25 October 2018; Accepted: 17 January 2019; Published: 08 March 2019.
16. Pan, X., Shi, J., Luo, P., Wang, X., & Tang, X. (2018, February). Spatial as deep: Spatial CNN for traffic scene understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.
17. Souweidane, N., & Smith, B. (2023). State of ADAS, Automation, and Connectivity. Center for Automotive Research, Ann Arbor, MI.
18. Tomasch, E., & Smit, S. (2023). Naturalistic driving study on the impact of an aftermarket blind spot monitoring system on driver behavior of heavy goods vehicles and buses on reducing conflicts with pedestrians and cyclists. Accident Analysis and Prevention, 192, 107242. https://doi.org/10.1016/j.aap.2023.107242