

О. БИЗКРОВНИЙ, К. СМЕЛЯКОВ

МОДЕЛІ КЛАСИФІКАТОРА ПЕРЕДУМОВ ВИНИКНЕННЯ ДТП ДЛЯ ПЕРЕДБАЧЕННЯ НЕБЕЗПЕЧНИХ СИТУАЦІЙ НА ПЕРЕХРЕСТЯХ

Предметом статті є передумови виникнення дорожньо-транспортних пригод на перехрестях та ділянках з обмеженою видимістю; використання моделей комп'ютерного зору для завдань класифікації передумов виникнення ДТП і визначення ефективності їх застосування для роботи в режимі реального часу. **Мета дослідження** – порівняння моделей комп'ютерного зору для завдання класифікації передумов виникнення ДТП для роботи в режимі реального часу; порівняння моделей з використанням платформи *Jetson TX2*; визначення ефективності роботи зазначеного підходу з метою створення сигналів для водія про небезпеку в режимі реального часу. **Завдання:** дослідження моделей комп'ютерного зору *Detectron2*, *YOLOv7* для завдання класифікації передумов ДТП на зображенні в розрізі швидкодії моделей, зручності створення набору даних, тренування моделей та їх використання; порівняння *YOLOv8* і *DetectNet_v2* на одноплатному комп'ютері *Jetson TX2* у розрізі швидкодії. **Методи дослідження:** тренування та використання моделей машинного навчання та методи моделювання небезпечних ситуацій за допомогою програмного забезпечення *BeamNG.tech*, *CARLA*; порівняльний аналіз результатів застосування моделей із застосуванням метрик для оцінювання їх ефективності. Основними **результатами дослідження** є виявлення найбільш ефективної моделі для завдань класифікації на одноплатному комп'ютері *Jetson TX2-DetectNet_v2*; позитивна оцінка ефективності використання запропонованого підходу для попередження водія про небезпеку в режимі реального часу, проте з огляду на розмір навчального набору даних та складність його підготовки. **Висновки.** Розглянуто алгоритми комп'ютерного зору, зокрема *Detectron2*, *YOLOv7* та *DetectNet_v2*. Виявлено, що модель *YOLOv7* є кращою порівняно з *Detectron2* для завдання виявлення передумов ДТП на зображенні, проте зі свого боку *DetectNet_v2* є більш ефективним для застосування на одноплатному комп'ютері *Jetson TX2*, якщо порівнювати з *YOLOv7*. Додатково на підставі експериментів з'ясовано, що застосування окресленого підходу для передбачення передумов ДТП є проблематичним через складнощі у створенні навчального набору даних – варіативність сценаріїв передумов.

Ключові слова: класифікація об'єктів на зображенні; комп'ютерний зір; машинне навчання; штучний інтелект; інформаційні технології; *Nvidia Jetson*; передбачення ДТП; *YOLO*; *DetectNet_v2*; *Detectron2*.

Вступ

Зростання кількості транспортних засобів та їх швидкості є одним із чинників підвищення частоти дорожньо-транспортних пригод, що зі свого боку також є фактором збільшення кількості жертв і постраждалих унаслідок ДТП. Сучасні методи запобігання небезпечних ситуацій на дорогах спрямовані переважно на безпеку окремих автомобілів і не беруть до уваги учасників дорожнього руху, які перебувають поза зоною дії вбудованих механізмів безпеки транспортних засобів.

Однією з проблем індивідуальних засобів безпеки автомобіля є неможливість попередження водія про перешкоди, розташовані за межами досяжності окремих засобів безпеки. Отже, виникає необхідність у розробленні технологій, що дають змогу виявляти та попереджати водіїв про потенційні небезпеки на дорогах у режимі реального часу, які можуть працювати поза межами зони

застосування індивідуальних засобів забезпечення безпеки автомобіля.

З огляду на сучасні тенденції у сфері технологій зазначена система має бути розроблена з використанням машинного навчання. Ще одним важливим фактором безпеки дорожнього руху є здатність водія швидко й адекватно реагувати на зміну ситуації та отриману інформацію. Адже швидкість виявлення та спосіб повідомлення водія також має вплив на ефективність системи.

Отже, розроблення та дослідження нових методів прогнозування небезпечних ситуацій і оперативного оповіщення про них є актуальним і важливим завданням.

Аналіз останніх досліджень і публікацій

Методи виявлення передумов і чинників, що викликають автомобільні аварії, використовують досягнення в сфері комп'ютерного зору та глибокого

навчання. Зростання кількості досліджень у цьому напрямі свідчить про важливість раннього виявлення потенційно аварійно небезпечних умов. Визначені передумови застосовують для управління безпекою дорожнього руху. Інноваційне дослідження [1] зосереджене на виявленні різних видів дорожніх перешкод, таких як лежачі поліцейські та вибоїни, зображення, отримані за допомогою камер ZED і глибокої нейронної мережі.

Окреслений підхід може використовуватись для виявлення перешкод на певних ділянках руху й за допомогою додаткового механізму, сповіщати

інших водіїв про виявлені небезпеки на дорожньому полотні, що дасть змогу учасникам дорожнього руху знати більше про дорожні умови, що сприяє кращому прийняттю рішень і зменшенню аварійності.

Крім цього, у роботі [2] було застосовано акселерометричні дані в поєднанні з обробленням зображень для покращення аналізу дорожнього покриття та виявлення перешкод, що можуть бути причинами аварій. У зазначеному дослідженні наголошено на важливості як візуальних, так і фізичних чинників для аналізу, які можуть попередити водія про потенційну небезпеку (рис. 1).

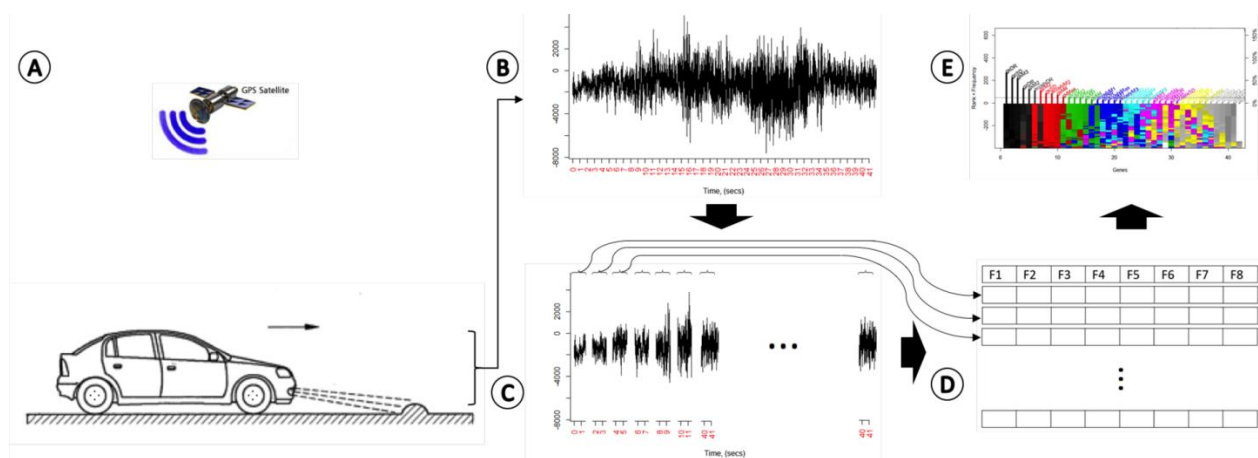


Рис. 1. Поєднання даних із сенсорів та візуального складника для завдань прогнозування небезпек [2]

Такі інтегровані методи дають змогу здійснювати моніторинг і допомагають водієві реагувати на перешкоди в режимі реального часу. Наявні дослідження сприяють розвитку та створенню автономних транспортних засобів.

У праці [3] використовується непараметричний метод оцінювання випадкової величини [4] для аналізу просторового розподілу ДТП на основі інформації, отриманої за допомогою камер спостереження. Цю методологію використовують для визначення кластерів аварій, виявлення ділянок, що потребують більш суворого контролю, і потенційних поліпшень інфраструктури. Зазначене наголошує на необхідності поєднання оцінювання історичних даних із сучасними методами оброблення зображень для більш повного розуміння аварійно небезпечних середовищ.

У роботі [5] запропоновано новий метод автоматичного виявлення дорожньо-транспортних пригод за допомогою машинного зору, зокрема аналізу записів із придорожніх камер спостереження для встановлення пошкоджених транспортних засобів, що свідчать про аварію. Цей тип аналізу дає

змогу швидко повідомляти аварійні служби про ДТП порівняно з традиційними методами інформування. Аналогічно підхід, розроблений у роботі [6], упроваджує глибоке навчання для вилучення візуальних ознак із подальшою ідентифікацією часових патернів, досягаючи вражаючого рівня точності у виявленні дорожньо-транспортних пригод на основі відеоданих. Ці дослідження надають важливого значення обробленню зображень у реальному часі для підвищення ефективності реагування на надзвичайні ситуації на дорогах. Описана в дослідженні [7] інтеграція підходів сегментації зображень з *fuzzy*-теорією [8] для оцінювання зон пошкоджень автомобілів у аваріях також показує, як оброблення зображень може бути використане для детального аналізу пошкоджень після аварії, забезпечуючи більш чітку картину наслідків зіткнення. Поєднання ефективних методів сегментації з *fuzzy*-логікою підвищує точність ідентифікації пошкоджених ділянок транспортними засобами й цим надає цінну інформацію для реконструкції та аналізу аварії.

Застосування систем кооперативної транспортної інфраструктури, як зазначено в дослідженні [9], відтворює, як дані зображень і датчиків можуть допомагати в автоматичному виявленні ДТП і внаслідок цього підвищувати ефективність реагування й безпеку дорожнього руху загалом.

У праці [10] запропоновано метод, що інтегрує аналіз зображень за допомогою згорткових нейронних мереж (CNN) для оптимізації механізмів реагування на ДТП, покращуючи як управління дорожнім рухом, так і заходи безпеки.

Інтеграція різних типів даних за допомогою передових технологій має важливе значення для сучасних систем управління дорожнім рухом, спрямованих на зниження аварійності. Оскільки ця галузь продовжує розвиватися, долучення оброблення зображень у методологію запобігання аваріям обіцяє не лише покращити прогнозування потенційних інцидентів, але й оптимізувати екстрене реагування на аварії. У розглянутих дослідженнях наголошено на вирішальній ролі технологічного прогресу в зменшенні ризиків, пов'язаних із дорожньо-транспортними пригодами.

Нерозв'язані проблеми та мета дослідження

У проаналізованих вище роботах розглянуто ДТП та реагування на неї як на постфактум-подію, тобто коли дорожньо-транспортна пригода вже відбулася. Необхідна технологія, яка зможе попередити водія про небезпеку в режимі реального часу, щоб запобігти аварії. Такий підхід дасть змогу суттєво зменшити кількість ДТП, оскільки інформація надходитиме безпосередньо з місця розташування автомобіля й транслюватиме ситуацію, що виникає безпосередньо в певний проміжок часу, а не базуватиметься на історичних даних певних ділянок руху.

Мета дослідження – порівняння моделей комп'ютерного зору для завдання класифікації передумов виникнення ДТП з метою роботи в режимі реального часу. Порівняння моделей з використанням платформи *Jetson TX2* [11]; визначення ефективності роботи зазначеного підходу для створення сигналів для водія про небезпеку в режимі реального часу. Для досягнення мети необхідно **виконати такі завдання**: дослідити моделі комп'ютерного зору *Detectron2*, *YOLOv7* для завдання класифікації передумов ДТП на зображенні в розрізі швидкодії

моделей, зручності створення набору даних, тренування моделей та їх використання; порівняння *YOLOv8* та *DetectNet_v2* на одноплатному комп'ютері *Jetson TX2* у розрізі швидкодії.

Матеріали й методи

З метою порівняння моделей комп'ютерного зору для завдань класифікації в режимі реального часу розглядаються моделі *Detectron 2* [12] та *YOLOv7* [13]. Після визначення кращого алгоритму в цій статті обрану модель або сімейство буде розгорнуто на *Jetson TX2* для оцінювання ефективності роботи на одноплатному комп'ютері.

Створення набору даних

Для створення типових сценаріїв дорожніх ситуацій було використано програмне забезпечення *BeamNG.tech* [14], де розроблялися скрипти, за якими автомобілі мали рухатись і влаштовувати небезпечні ситуації. Для спрощення класифікації небезпечних ситуацій було визначено, що для експерименту застосовуватиметься небезпечна ситуація повороту ліворуч на перехресті через декілька зустрічних смуг, де одна з них пропускає водія, що повертає, проте крайня права смуга пуста, або нею хтось рухається і не знає, чому сусідня смуга зупинена (рис. 2).

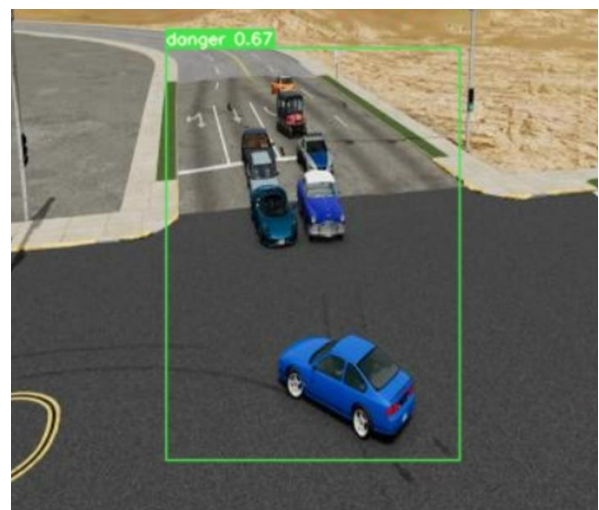


Рис. 2. Типова ситуація, під час якої водії, що мають пріоритет у русі, призупиняються, щоб дати іншому транспортному засобу зробити поворот

Описана ситуація поширена та керується більш етичними нормами, ніж правилами дорожнього руху.

У деяких ситуаціях такий вияв поваги до водія, який хоче повернути ліворуч, завершується трагічно.

За допомогою створених сценаріїв ДТП з використанням програмного забезпечення *BeamNG.tech* було записано відео, яке потім застосовувалося для формування тренувального набору даних. Стратегія створення набору даних для дослідження передбачала такі кроки:

- симуляція щонайменше п'яти різних перехресть із різними конфігураціями автомобільного руху та різною кількістю зустрічних смуг;
- класифікація зображень лише за двома категоріями: "небезпека" й "відсутність небезпеки".

Навчальний набір даних містить 300 зображень, з яких 150 належать до класу "небезпека", а 150 – до класу "відсутність небезпеки". На рис. 3 наведено приклад зображення з набору даних.

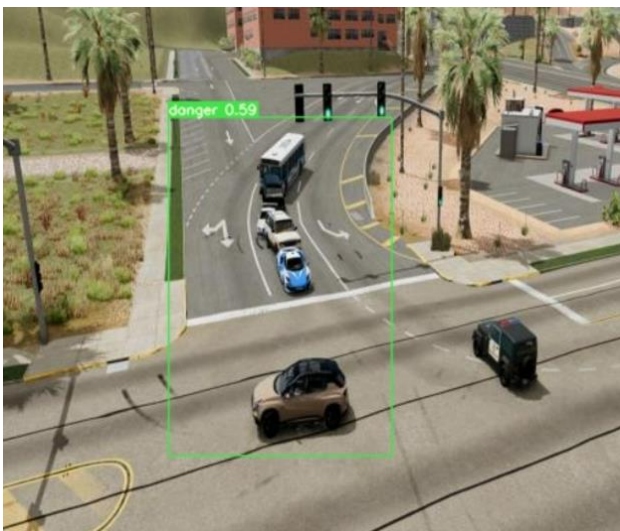


Рис. 3. Приклад зображення з тренувального набору даних, який описує клас "небезпека"

Тобто створений набір даних слугує для визначення патернів того, що є передумовами виникнення зазначеного типу ДТП.

Метрики валідації моделей

Результати проведеного експерименту були визначені за допомогою наведених далі метрик: *mAP* (*Mean Average Precision*) *with IoU*, час тренування моделей, використана пам'ять відеокарти, розмір тренуваної моделі, *Loss Box*, *Loss Cls*, час для розпізнавання класів на зображенні, *Recall*.

Mean Average Precision (*mAP*) *with IoU* – це метрика, що узагальнює якість класифікації об'єкта за кількома критеріями: наскільки правильно модель ідентифікує об'єкти (класифікація) та наскільки просторово точно вона їх локалізує (локалізація через *IoU*).

IoU між передбаченим об'єктом B_p та істинною розміткою B_g визначається як відношення:

$$IoU(B_p, B_g) = \frac{area(B_p \cap B_g)}{area(B_p \cup B_g)}, \quad (1)$$

де $area(B_p \cap B_g)$ – площа перетину двох прямокутників; $area(B_p \cup B_g)$ – площа об'єднання двох прямокутників.

IoU (*Intersection over Union*) є показником оцінювання ступеня перекриття двох обмежувальних рамок або сегментаційних масок. Якщо передбачення ідеальне, тоді *IoU* дорівнює 1, а за повної невідповідності – 0. Значення *IoU* визначається рівнем перекриття між цими межами. Тому для оцінювання точності модель аналізують у разі різних значень *IoU*, щоб з'ясувати, за яких умов досягається найкраща продуктивність.

Метрика точності (*Precision*) має такий вигляд:

$$P = \frac{TruePositive}{TruePositive + FalsePositive}, \quad (2)$$

де *TruePositive* – кількість правильно передбачених об'єктів; *FalsePositive* – кількість хибнопозитивних передбачень, коли модель передбачила об'єкт, якого насправді немає.

Формула *Recall* (повноти) використовується для оцінювання того, наскільки добре модель виявляє всі релевантні об'єкти. Вона визначається як відношення кількості правильно передбачених позитивних випадків (*True Positives*) до загальної кількості фактичних позитивних випадків (сума *True Positives* і *False Negatives*):

$$Recall = \frac{TP}{TP + FN}, \quad (3)$$

де *TP* – кількість правильно передбачених об'єктів; *FN* – кількість втрачених об'єктів. Реальний об'єкт існує, проте модель не змогла його класифікувати.

Loss Box – це функція втрат, що оцінює, наскільки точно передбачені обмежувальні рамки прилягають до реального об'єкта. Зазвичай використовується регресійна втрата, така як *L1* або

smooth L1. Формула для обчислення цієї втрати має такий вигляд:

$$L_{\text{box}}(t^u, v) = \sum_{i \in \{x, y, w, h\}} L_1^{\text{smooth}}(t_i^u - v_i), \quad (4)$$

де $t^u = (t_x^u, t_y^u, t_w^u, t_h^u)$ – позначає вектор передбачених параметрів рамки для об'єкта класу u ;

$v = (v_x, v_y, v_w, v_h)$ – вектор відповідних істинних координат;

$i \in \{x, y, w, h\}$ – описує ітерування над компонентами координат;

L_1^{Smooth} – функція втрат.

Loss Cls (classification loss) оцінюють правильність класифікації кожного передбаченого класу. Зазвичай для цього використовується функція втрат на основі крос-ентропії. Формула для обчислення цієї втрати має такий вигляд:

$$L_{\text{cls}}(p, y) = -[y \log(p) + (1 - y) \log(1 - p)], \quad (5)$$

де $p \in \{0, 1\}$ – передбачена класифікація; $y \in \{0, 1\}$ – істинна класифікація, де 1 – позитивний клас, 0 – негативний.

Для порівняння моделей після розгортання їх на одноплатному комп'ютері *Jetson TX2* було використано

інструментарій *trtexes*, що дає змогу досягти швидкодії та інші характеристики після запуску моделей комп'ютерного зору на *Jetson TX2* для порівняння.

Результати досліджень

У табл. 1 подано результати тренування та порівняння моделей *YOLOv7* і *Detectron2*. Для тренування було використано *Google Colab* середовище з GPU: *Tesla T4*.

Графіки метрик втрат моделі *YOLOv7* та *Average Precision with IoU* зображені на рис. 4 і 5.

Таблиця 1. Порівняння отриманих метрик під час тренування моделей *YOLOv7* та *Detectron2*

Характеристика	YOLOv7	Detectron2
System RAM, GB	5.7	3.7
GPU RAM, GB	11.4	8.5
SSD, GB	24.9	25.9
Training time, mins	15.9	51
AP@.5:95	0.583	0.651
AP@.5	0.926	0.909
Inference time, s	0.1916668	0.198093
Model size, MB	74.8	815

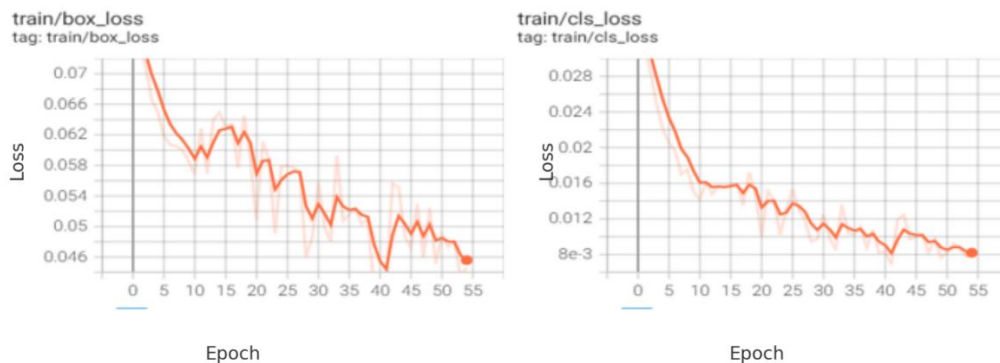


Рис. 4. Графіки функцій втрат моделі *YOLOv7*

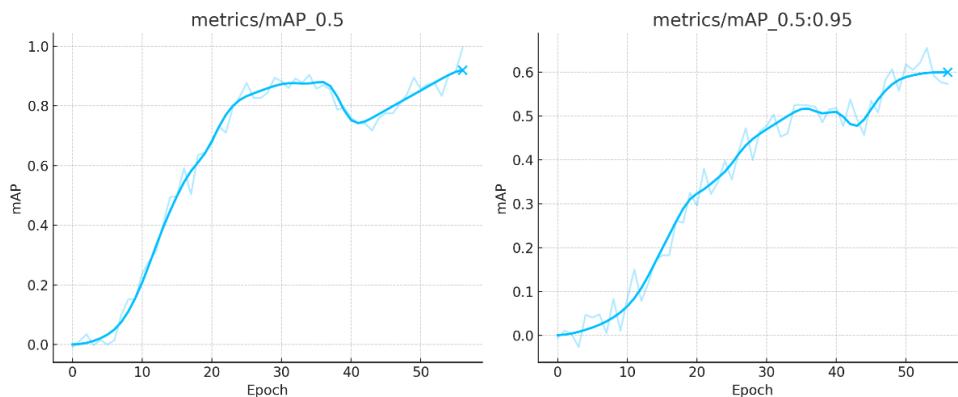


Рис. 5. Графіки mAP з $IoU = 0.5$ та $IoU = 0.95$ моделі *YOLOv7*

Протягом сорокової ітерації тренування більшість метрик мали спад у показниках. Такі значення можуть свідчити про надмірне пристосовування моделі до тренувальних даних, що може негативно

впливати на здатність розпізнавання класів, що загалом знижує точність моделі, але й фіктивно зменшує кількість помилок. Результати верифікації моделі подані на рис. 6.



Рис. 6. Результат розпізнавання моделлю YOLOv7 повороту ліворуч як небезпечної ситуації

Графіки *Loss Box* моделі *Detectron2* зображені на рис. 7.

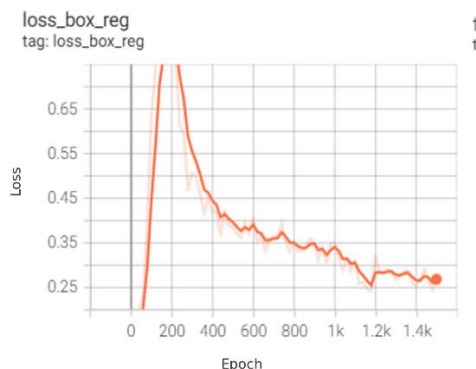


Рис. 7. Графік функції втрат отриманої під час тренування моделі *Detectron*

Результат валідації тренуваної моделі *Detectron2* поданий на рис. 8.



Рис. 8. Результат валідації моделі *Detectron2* [15]

Висновки щодо тренування та порівняння моделей YOLOv7 і *Detectron2*

З огляду на результати тренувань, валідації моделей і метрик можна сказати, що на цьому наборі даних важко виявити кращу модель за точністю. Швидкість роботи / класифікації є дуже близькою одна до одної. Проте можна зауважити, що модель YOLOv7 значно швидше навчається, та їй необхідно менше ітерацій для досягнення того результату, що був поданий в експерименті. Також розмір моделі YOLOv7 суттєво менший, що може вплинути на її розгортання на апаратному забезпеченні, що має обмежені можливості з боку оперативної пам'яті та пам'яті зберігання.

Розгортання моделей YOLOv8 і *DetectNet_v2* на *Jetson TX2*

Під час розгортання моделі YOLOv8 на *Jetson TX2* виникли складнощі із застосуванням натренованої моделі. Розгортання моделей YOLOv8 безпосередньо на платформі *Jetson TX2 NX* є неможливим. Розробник YOLOv8 – компанія *Ultralytics* – використовує *Python 3.10*, тоді як остання доступна версія *Python* на *Jetson TX2* – 3.6.9. Хоча існує можливість зібрати *Python 3.10* з вихідного коду для архітектури *aarch64*, відсутні готові *pip*-пакети для встановлення *torchvision*, сумісного з цією

версією *JetPack* і оновленим *Python*. Крім того, для роботи *Ultralytics* потрібен *CUDA Toolkit 11*, тоді як *Jetson TX2* підтримує лише версію 10.2. Оновлення *CUDA* на цій платформі є складним завданням через сильну залежність компонентів на низькому рівні. Однак існує спосіб розгортання моделі *YOLOv8* на *Jetson TX2*. Загалом він полягає у використанні плагіна *DeepStream* для запуску моделі. Процес передбачає два основні етапи:

- 1) генерація ваг моделі з *.pt*-файлу за допомогою скрипту *get_wts_yoloV8.py* на іншому комп'ютері, що підтримує *Python 3.10*;
- 2) використання згенерованих файлів для виконання моделі в тестових застосунках безпосередньо на *Jetson TX2 NX*.

Порівняння швидкодії *YOLOv8* та *DetectNet_v2*

Під час проведення експерименту швидкодії моделі *YOLOv8* [16] на відео було виявлено суттєву деградацію її швидкодії. Використання ресурсів відеокарти становило 100 %, проте швидкодія – п'ять кадрів за секунду.

Це свідчить про неможливість використання запропонованої моделі на одноплатному комп'ютері *Jetson TX2* для завдань класифікації в реальному часі, оскільки швидкодія не дає змогу обробляти зображення вчасно.

Проте компанія *Nvidia* надає розроблені нею моделі, які є оптимізованими для застосування

на одноплатних комп'ютерах *Nvidia Jetson*. До такої моделі належить *DetectNet_v2*, яку можна натренувати за власним набором даних.

Швидкодія тренованої моделі, яка працює на основі *ResNet-10*, що виявляє автомобілі, є досить високою: 74 кадри за секунду, що може вважатися більш ніж достатнім для завдань розпізнавання в реальному часі (рис. 9).

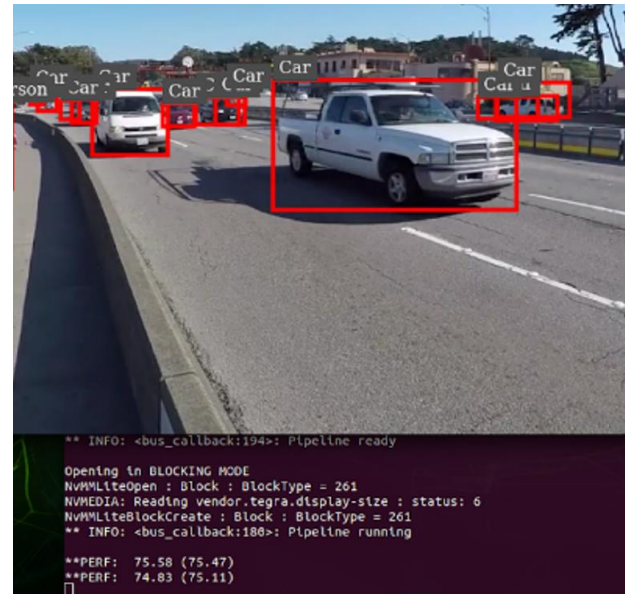


Рис. 9. Швидкодія тренованої моделі *DetectNet_v2*

Із використанням інструментарію *trtexec* було обчислено, наскільки модель *DetectNet_v2* краще працює на *Jetson TX2*, ніж *YOLOv8* (табл. 2).

Таблиця 2. Порівняння *YOLOv8* і *DetectNet_v2*

Metric Name	Yolov8m	DetectNet_v2	DetectNet_v2 Nano
Average on 10 runs – GPU latency, ms	155.703	6.927	4.180
Average on 10 runs – Host latency, ms	155.991	7.088	4.270
Throughput, qps	6.407	140.517	233.145
Latency: min, ms	153.808	7.052	4.244
Latency: max, ms	157.84	8.735	5.289
Latency: mean, ms	156.049	7.109	4.282
End-to-End Host Latency: min, ms	153.817	7.057	4.250
End-to-End Host Latency: max, ms	157.854	8.741	5.295
End-to-End Host Latency: mean, ms	156.058	7.115	4.288
GPU Compute Time: min, ms	153.52	6.892	4.154
GPU Compute Time: max, ms	157.556	8.575	5.196
GPU Compute Time: mean, ms	155.761	6.948	4.192

Метрики, що надає інструментарій *trtexec*, мають таке пояснення:

- *GPU latency (average on 10 runs, ms)*: середній час, необхідний для обчислень класифікації винятково

на графічному процесорі, незважаючи на перед- і післяоброблення або копіювання даних між CPU та GPU;

- *Host latency (average on 10 runs, ms)*: середній час, заміряний на стороні CPU, що містить виклики

до GPU, передавання вхідної інформації, очікування результату та його зчитування;

- *Throughput (qps)*: продуктивність системи, що визначається як кількість запитів (інференсів), які можуть бути оброблені за секунду (*queries per second*);

- *Latency (min / max / mean, ms)*: мінімальна, максимальна та середня затримка під час класифікації, що дає змогу оцінити варіативність часу виконання;

- *End-to-End Host Latency (min / max / mean, ms)*: загальна затримка, виміряна на стороні хоста, яка передбачає повний цикл оброблення – від моменту надсилання вхідної інформації до досягнення кінцевого результату;

- *GPU Compute Time (min / max / mean, ms)*: час фактичного виконання обчислень на GPU (ядро виконання), що демонструє завантаження та ефективність використання графічного процесора.

Порівняння моделей *YOLOv8m*, *DetectNet_v2* та *DetectNet_v2 Nano* показало суттєву різницю у швидкодії. *YOLOv8m* має найвищу затримку й найнижчу пропускну здатність, тому потребує потужних GPU та не є оптимальною для вбудованих систем. *DetectNet_v2* забезпечує кращий баланс між точністю та продуктивністю й може застосовуватись

на одноплатних комп'ютерах *Jetson*. Найефективнішою для *Jetson*-платформ є *DetectNet_v2 Nano* завдяки найменшій затримці (~4.2 мс) та найвищій продуктивності (~233 qps).

Обговорення досягнутих результатів

Проведені експерименти вказують на те, що є можливість створення моделі комп'ютерного зору, яка зможе класифікувати передумови ДТП в режимі реального часу. Для покращення цього підходу можна застосувати такі спрощення:

- вимкнення фонові інформації на зображенні, оскільки вона не впливає на процес виконання маневру;

- ігнорування типу транспортного засобу, адже характер маневру залишається незмінним незалежно від того, чи його здійснює легковий автомобіль, вантажівка або велосипед;

- додавання характеристик, які сприятимуть кращій класифікації зображень, зокрема інформації про дорожню розмітку, яка може бути відсутньою або малопомітною.

Приклад описаних покращень зображень на рис. 10.

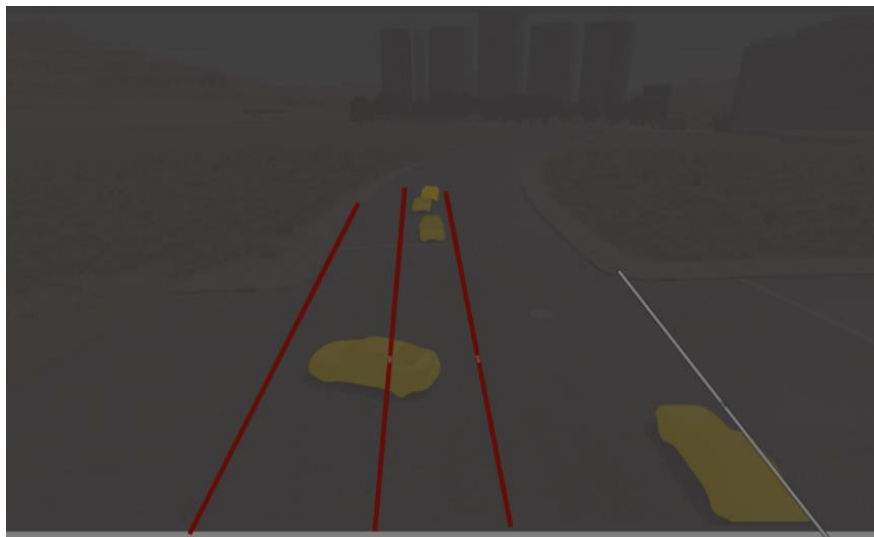


Рис. 10. Приклад змінених елементів зображення для покращення результатів класифікації

Однак цей підхід потребує значних ресурсів для підготовки, оскільки анування зображень виконується вручну, і тільки людина може коректно визначити, які лінії та об'єкти необхідно позначати, а які не мають значення. Також значна кількість різних варіацій небезпечних ситуацій зумовлює розширення навчального набору даних.

Висновки

У межах цього дослідження розглянуто сімейства алгоритмів комп'ютерного зору *YOLO*, *Detectron2* та *DetectNet_v2*, а також їх можливість у застосуванні на одноплатному комп'ютері *Jetson TX2*.

Було створено навчальний набір даних для класифікації небезпечної ситуації повороту ліворуч, а також проведено навчання моделей і порівняно їх. За результатами експериментів визначено, що *DetectNet_v2* є найкращою моделлю для роботи на одноплатному комп'ютері *Jetson TX2* з-поміж

розглянутих. Подальші кроки в розробленні цього підходу можна здійснювати в розрізі зміни навчального набору даних, щоб оптимізувати й допомогти моделі у визначенні того, які ситуації вважаються небезпечними, а які ні, а також зміни гіперпараметрів моделей під час тренування.

Список літератури

1. Speed Bump and Pothole Detection Using Deep Neural Network with Images Captured through ZED Camera / J.-E. Peralta-López та ін. *Applied Sciences*. 2023. Vol. 13, № 14. 8349 p. DOI: <https://doi.org/10.3390/app13148349>
2. Speed Bump Detection Using Accelerometric Features: A Genetic Algorithm Approach / J. Celaya-Padill. *Sensors*. 2018. Vol. 18, № 2. 443 p. DOI: <https://doi.org/10.3390/s18020443>
3. Eslami Nezhad S. A., Delavar M. R. An integrated network-constrained spatial analysis for car accidents: a case study of tehran city, iran. *ISPRS – International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. 2019. XLII-4/W18. P. 335–342. DOI: <https://doi.org/10.5194/isprs-archives-xlii-4-w18-335-2019>
4. Contributors to Wikimedia projects. Kernel density estimation – Wikipedia. Wikipedia, the free encyclopedia. URL: https://en.wikipedia.org/wiki/Kernel_density_estimation (дата звернення: 19.04.2025).
5. Ravindran V., Viswanathan L., Rangaswamy S. A Novel Approach to Automatic Road-Accident Detection using Machine Vision Techniques. *International Journal of Advanced Computer Science and Applications*. 2016. Vol. 7, № 11. DOI: <https://doi.org/10.14569/ijacsa.2016.071130>
6. Robles-Serrano S., Sanchez-Torres G., Branch-Bedoya J. Automatic Detection of Traffic Accidents from Video Using Deep Learning Techniques. *Computers*. 2021. Vol. 10. № 11. 148 p. DOI: <https://doi.org/10.3390/computers10110148>
7. Amirfakhrian M., Parhizkar M. Integration of image segmentation and fuzzy theory to improve the accuracy of damage detection areas in traffic accidents. *Journal of Big Data*. 2021. Vol. 8, № 1. DOI: <https://doi.org/10.1186/s40537-021-00539-2>
8. Zimmermann H. J. Fuzzy set theory. *Wiley Interdisciplinary Reviews: Computational Statistics*. 2010. Vol. 2, № 3. P. 317–332. DOI: <https://doi.org/10.1002/wics.82>
9. An Automatic Car Accident Detection Method Based on Cooperative Vehicle Infrastructure Systems / D. Tian та ін. *IEEE Access*. 2019. Vol. 7. P. 127453–127463. DOI: <https://doi.org/10.1109/access.2019.2939532>
10. Wang S. Traffic Accident Prediction based on CNN and Time of Commuting after Accident. *Highlights in Science, Engineering and Technology*. 2024. Vol. 118. P. 65–71. DOI: <https://doi.org/10.54097/5t9mba88>
11. Suzen A. A., Duman B., Sen B. Benchmark Analysis of Jetson TX2, Jetson Nano and Raspberry PI using Deep-CNN. *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, Ankara, Turkey, 2020 p. 2020. DOI: <https://doi.org/10.1109/hora49412.2020.9152915>
12. Detection, instance segmentation, and classification for astronomical surveys with deep learning (DeepDISC): Detectron2 implementation and demonstration with hyper supprime-cam data / G. Merz та ін. *Monthly Notices of the Royal Astronomical Society*. 2023. DOI: <https://doi.org/10.1093/mnras/stad2785>
13. Wang C.-Y., Bochkovskiy A., Mark Liao H.-Y. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *Computer Vision and Pattern Recognition*. 2022. DOI: <https://doi.org/10.48550/arxiv.2207.02696>
14. BeamNG.tech Contributors. BeamNG.tech. *BeamNG.tech*. URL: <https://beamng.tech/> (дата звернення: 25.02.2025).
15. Comparison of potential road accident detection algorithms for modern machine vision system / O. Byzkrovnyi та ін. *Environment. technologies. resources. Proceedings of the International Scientific and Practical Conference*. 2023. Vol. 3. P. 50–55. DOI: <https://doi.org/10.17770/etr2023vol3.7299>
16. Comparison of Object Detection Algorithms for the Task of Person Detection on Jetson TX2 NX Platform / O. Byzkrovnyi та ін. *2024 IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream)*, Vilnius, Lithuania, 2024 p. 2024. DOI: <https://doi.org/10.1109/estream61684.2024.10542592>

References

1. Peralta-López, J., Morales-Viscaya, J., Lázaro-Mata, D., Villaseñor-Aguilar, M., Prado-Olivarez, J., Pérez-Pinal, F., Gutiérrez, A. (2023), "Speed bump and pothole detection using deep neural network with images captured through zed camera". *Applied Sciences*, № 13(14), 8349 p. DOI: <https://doi.org/10.3390/app13148349>
2. Celaya-Padilla, J. M., Galván-Tejada, C. E., López-Monteagudo, F. E., Alonso-González, O., Moreno-Báez, A., Martínez-Torteya, A., Gamboa-Rosales, H. (2018), "Speed bump detection using accelerometric features: a genetic algorithm approach". *Sensors*, № 18(2), 443 p. DOI: <https://doi.org/10.3390/s18020443>
3. Eslaminezhad, S. A. and Delavar, M. R. (2019), "An integrated network-constrained spatial analysis for car accidents: a case study of tehran city, iran". *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-4/W18, P. 335–342. DOI: <https://doi.org/10.5194/isprs-archives-xlii-4-w18-335-2019>
4. Contributors to Wikimedia projects, (2005), "Kernel density estimation – Wikipedia. *Wikipedia, the free encyclopedia*", available at: https://en.wikipedia.org/wiki/Kernel_density_estimation (last accessed 25.02.2025)
5. Ravindran, V., Viswanathan, L., & Rangaswamy, S. (2016), "A novel approach to automatic road-accident detection using machine vision techniques". *International Journal of Advanced Computer Science and Applications*, №7(11). DOI: <https://doi.org/10.14569/ijacsa.2016.071130>
6. Robles-Serrano, S., Torres, G. S., & Branch, J. W. (2021), "Automatic detection of traffic accidents from video using deep learning techniques". *Computers*, №10(11), 148 p. DOI: <https://doi.org/10.3390/computers10110148>
7. Amirfakhrian, M. and Parhizkar, M. (2021), "Integration of image segmentation and fuzzy theory to improve the accuracy of damage detection areas in traffic accidents". *Journal of Big Data*, №8(1). DOI: <https://doi.org/10.1186/s40537-021-00539-2>
8. Zimmermann, H. (2010), "Fuzzy set theory". *WIREs Computational Statistics*, №2(3), P. 317–332. DOI: <https://doi.org/10.1002/wics.82>
9. Tian, D., Zhang, C., Duan, X., & Wang, X. (2019), "An automatic car accident detection method based on cooperative vehicle infrastructure systems". *IEEE Access*, № 7, P. 127453-127463. DOI: <https://doi.org/10.1109/access.2019.2939532>
10. Wang, S. (2024), "Traffic accident prediction based on cnn and time of commuting after accident". *Highlights in Science, Engineering and Technology*, №118, P. 65–71. DOI: <https://doi.org/10.54097/5t9mba88>
11. Süzen, A. A., Duman, B., & Şen, B. (2020). "Benchmark analysis of jetson tx2, jetson nano and raspberry pi using deep-cnn". *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, № 1-5. DOI: <https://doi.org/10.1109/hora49412.2020.9152915>
12. Merz, G., Liu, Y., Burke, C. J., Aleo, P., Liu, X., Kind, M. C., Liu, Y. (2023), "Detection, instance segmentation, and classification for astronomical surveys with deep learning (deepdisc): detectron2 implementation and demonstration with hyper supprime-cam data". *Monthly Notices of the Royal Astronomical Society*, № 526(1), P. 1122–1137. DOI: <https://doi.org/10.1093/mnras/stad2785>
13. Wang, C., Bochkovskiy, A., & Liao, H. M. (2022), "Yolov7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors". *Computer Vision and Pattern Recognition*. DOI: <https://doi.org/10.48550/arxiv.2207.02696>
14. "BeamNG.tech Contributors, BeamNG.tech". 2023. available at: <https://beamng.tech/> (last accessed: 25.02.2025)
15. Byzkrovnyi, O., Smelyakov, K., Chupryna, A., Savulionienė, L., & Sakalys, P. (2023), "Comparison of potential road accident detection algorithms for modern machine vision system". *Environment. technologies. resources*. Proceedings of the International Scientific and Practical Conference, №3, P. 50–55. DOI: <https://doi.org/10.17770/etr2023vol3.7299>
16. Byzkrovnyi, O., Smelyakov, K., Chupryna, A., & Lanovyy, O. (2024), "Comparison of object detection algorithms for the task of person detection on jetson tx2 nx platform". *IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream)*. DOI: <https://doi.org/10.1109/estream61684.2024.10542592>

Надійшла (Received) 15.04.2025

Відомості про авторів / About the Authors

Бизкровний Олександр Миколайович – Харківський національний університет радіоелектроніки, аспірант кафедри програмної інженерії, Харків, Україна; e-mail: oleksandr.byzkrovnyi@nure.ua; ORCID ID: <https://orcid.org/0000-0001-9335-442X>

Смеляков Кирило Сергійович – доктор технічних наук, професор, Харківський національний університет радіоелектроніки, завідувач кафедри програмної інженерії, член НТР, Харків, Україна; e-mail: kyrylo.smelyakov@nure.ua; ORCID ID: <https://orcid.org/0000-0001-9938-5489>

Byzkrovnyi Oleksandr – Kharkiv National University of Radio Electronics, PhD Student at the Department of Software Engineering, Kharkiv, Ukraine.

Smelyakov Kyrylo – Doctor of Sciences (Engineering), Professor, Kharkiv National University of Radio Electronics, Head at the Department of Software Engineering, Kharkiv, Ukraine.

MODELS OF THE CLASSIFIER OF PREREQUISITES FOR THE OCCURRENCE OF ROAD ACCIDENTS TO PREDICT DANGEROUS SITUATIONS AT INTERSECTIONS

The **subject matter**: this study focuses on the preconditions for the occurrence of road traffic accidents at intersections and areas with limited visibility; the use of computer vision models for classifying the preconditions of road traffic accidents and evaluating the effectiveness of their use in real-time operation. The **goal** of the study is to compare computer vision models for the task of classifying the preconditions of traffic accidents for real-time operation. The study involves comparing models using the Jetson TX2 platform and determining the effectiveness of this approach for generating real-time warning signals for drivers.

Tasks: explore computer vision models such as Detectron2 and YOLOv7 for the task of classifying traffic accident preconditions in terms of model performance, ease of dataset creation, model training, and deployment. Compare YOLOv8 and DetectNet_v2 on a single-board computer Jetson TX2 in terms of processing speed. The **methods** used include training and using machine learning models, as well as simulating hazardous situations using software such as BeamNG.tech and CARLA. A comparative analysis of the application results of the models was conducted using performance evaluation metrics. The **main results** of the study include identifying the most effective model for classification tasks on the Jetson TX2 single-board computer – DetectNet_v2; a positive evaluation of the effectiveness of this approach for real-time driver warning, although certain limitations were noted regarding the size of the training dataset and the complexity of its preparation. **Conclusions**. The following computer vision algorithms were examined: Detectron2, YOLOv7, and DetectNet_v2. It was found that YOLOv7 outperforms Detectron2 in detecting the preconditions of traffic accidents in images. However, DetectNet_v2 was found to be better suited for deployment on the Jetson TX2 single-board computer compared to YOLOv7. Additionally, based on experimental findings, it was concluded that the application of this approach to predicting accident preconditions is problematic due to difficulties in creating a training dataset – specifically, the variability of precondition scenarios.

Keywords: Object classification on an image; Computer vision; Machine Learning; Artificial Intelligence; Information Technologies; Nvidia Jetson; road traffic accidents prediction; YOLO; DetectNet_v2; Detectron2.

Бібліографічні описи / Bibliographic descriptions

Бизкровний О. М., Смельяков К. С. Моделі класифікатора передумов виникнення ДТП для передбачення небезпечних ситуацій на перехрестях. *Сучасний стан наукових досліджень та технологій в промисловості*. 2025. № 2 (32). С. 177–187. DOI: <https://doi.org/10.30837/2522-9818.2025.2.177>

Byzkrovnyi, O., Smelyakov, K. (2025), "Models of the classifier of prerequisites for the occurrence of road accidents to predict dangerous situations at intersections", *Innovative Technologies and Scientific Solutions for Industries*, No. 2 (32), P. 177–187. DOI: <https://doi.org/10.30837/2522-9818.2025.2.177>