

СТЕЦЬ СЕРГІЙ

Чернівецький національний університет імені Юрія Федьковича

<https://orcid.org/0009-0007-0231-9970>e-mail: stets.serhii@chnu.edu.ua

АНАЛІЗ ТОЧНОСТІ ТА ШВИДКОДІЇ ДЕТЕКЦІЇ АВТОМОБІЛІВ ЗА ДОПОМОГОЮ НЕЙРОННИХ МЕРЕЖ YOLOV8 ТА YOLOV11

Предметом дослідження є застосування згорткових нейронних мереж YOLOv8 та YOLOv11 для виявлення зображень автомобілів. Сформовано спеціалізований датасет, який містить зображення автомобілів для різних умов отримання. Створений датасет розділено на навчальний, контрольний та тестовий набори даних. Програмне забезпечення для навчання нейромережі та детекції об'єктів розроблено на мові Python із використанням хмарних технологій. Для підвищення швидкості навчання моделей використано паралельні обчислення, які реалізовано на GPU. Проведено донавчання моделей YOLOv8 та YOLOv11 на створеному датасеті. Оцінку точності детекції виконано за метриками повноти, точності, IOU, mAP50 та mAP60-95. Доновчені моделі продемонстрували вищу точність при виявленні зображень автомобілів. Точність моделей YOLOv8 та YOLOv11 після донавчання практично співпадає. Показано, що модель YOLOv11 забезпечує вищу швидкодію при детекції об'єктів і потребує меншого обсягу пам'яті. Моделі нейромереж успішно протестовано при трекінгу автомобілів на реальних відеозаписах.

Ключові слова: виявлення автомобілів, згорткові нейронні мережі, YOLO, хмарні технології, програмне забезпечення, паралельні обчислення.

STETS SERHII

Yuriy Fedkovych Chernivtsi National University

ANALYSIS OF ACCURACY AND SPEED OF VEHICLE DETECTION USING NEURAL NETWORKS YOLOV8 AND YOLOV11

The subject of the study is the application of convolutional neural networks (CNN) YOLOv8 and YOLOv11 for the task of detecting car images. For comparative analysis, medium-sized YOLO models were used, namely the YOLOv8m and YOLOv11m models. A specialized dataset was formed, which contains images of vehicles for different scales, acquisition conditions, lighting levels and quality. The dataset was annotated manually using the Roboflow tool. The size of the initial dataset was increased by data augmentation methods. The resulting dataset was divided into training, control and test datasets. The software for training the neural network and object detection was developed in Python using cloud technologies. To increase the speed of model training, parallel calculations were used, which are implemented on the GPU. The YOLOv8m and YOLOv11m models were further trained on the created dataset. Using the test dataset, the accuracy and speed of YOLO models of different versions before and after retraining were compared. The detection accuracy was compared using the metrics of Recall, Precision, IOU, mAP50, mAP60-95. The experimental results indicate that the retrained models demonstrate higher accuracy in detecting car images compared to models without retraining. The accuracy of the YOLOv8m and YOLOv11m models after retraining is practically the same. It is shown that the YOLOv11m model provides higher speed in detecting objects and requires less memory, which is a significant advantage in resource-constrained environments. Neural network models have been successfully tested in tracking cars on real video recordings from traffic surveillance cameras. The results obtained demonstrate the practical effectiveness of retraining YOLO neural networks, as well as the feasibility of further research into improving YOLO architectures.

Keywords: car detection, CNN, YOLO, cloud technologies, software, parallel computing.

Стаття надійшла до редакції / Received 17.07.2025

Прийнята до друку / Accepted 15.08.2025

Постановка проблеми

У сучасних системах комп'ютерного зору завдання виявлення (детекції) транспортних засобів, зокрема, автомобілів, є критично важливим для широкого спектра застосунків: від систем відеоспостереження до інтелектуальних транспортних систем та автономного водіння [1, 2]. Складність завдання детекції зображень автомобілів зумовлена змінами освітлення, погодними умовами, високою інтенсивністю руху, різними розмірами та ракурсами автомобілів. З цієї причини для виявлення автомобілів на зображеннях широко застосовують штучні нейронні мережі (ШНМ). Одним із найбільш ефективних засобів для виявлення автомобілів на зображеннях є згорткові нейронні мережі (CNN – Convolutional Neural Network) з архітектурою YOLO (You Only Look Once – ви дивитеся лише раз) [3]. Назва YOLO означає, що нейронна мережа одноходова, а відповідно володіє високою швидкодією. Моделі YOLO попередньо навчені (Pre-trained) на датасеті COCO (Common Objects in Context – Загальні Об'єкти в Контексті) [4], який містить 80 класів найбільш поширених об'єктів (у тому числі й автомобілів). Проте, існує ряд версій YOLO (на даний час найновіша версія – 12), тому виникає проблема вибору найбільш відповідної версії YOLO для конкретної задачі детекції. Крім цього, попередньо навчені моделі YOLO можуть забезпечувати відносно низьку точність детекції для зображень дорожнього руху певного типу, тому виникає потреба у донавчанні (Fine-tuning) нейромереж на спеціально підібраних репрезентативних наборах даних (датасетах).

Аналіз досліджень та публікацій

У сфері комп'ютерного зору задачі виявлення транспортних засобів активно досліджуються, зокрема, у контексті адаптації нейронних мереж до змінних умов середовища.

Дослідження [5] представляє покращену методику детекції транспортних засобів у реальному

часі на основі модифікованої мережі YOLOv5, яка забезпечує зменшення помилкових детекцій об'єктів. Розглянуто можливості підвищення точності детекції транспортних засобів на основі зображень, отриманих для складних умов дорожнього руху, наприклад, при оклюзії (перекриванні) зображень об'єктів. Однак, у даній роботі розглянуто детекцію зображень об'єктів із використанням YOLOv5, тому існує потреба у дослідженні більш сучасних версій YOLO (зокрема, v8 та v11).

У роботі [6] запропоновано покращену модель нейронної мережі YOLOX_S, яка призначена для вирішення проблем неточної детекції та пропуску малих об'єктів (віддалених транспортних засобів) у дорожніх сценах. Дослідники оптимізували оригінальну мережу YOLOX_S, видаливши надлишкові частини архітектури. Такий підхід демонструє важливість адаптації нейронної мережі для специфічних завдань детекції транспортних засобів, що може реалізуватися шляхом зміни архітектури нейронної мережі або її донавчання на спеціалізованих датасетах.

Застосування нейронних мереж YOLO для детекції об'єктів в автономних транспортних засобах у реальному часі розглянуто в роботі [7]. Показано можливості застосування YOLO для виявлення транспортних засобів, пішоходів, дорожніх знаків та перешкод. Дослідження підтверджує практичну важливість та перспективність застосування нейронних мереж YOLO для розвитку технологій автономного водіння, демонструє широкий спектр застосувань YOLO у транспортній сфері.

У роботі [8] досліджено можливості застосування нейронних мереж YOLO для детекції транспортних об'єктів, показано їх переваги у високій точності детекції та високій швидкодії (достатній для обробки відеопотоку в реальному часі). Проте, автори також виявили деякі обмеження YOLO: низька точність детекції на зображеннях зі значними варіаціями масштабу та великою кількістю об'єктів. Дослідження підтверджує доцільність застосування моделі YOLOv8 для детекції транспортних засобів та вказує на необхідність донавчання нейромережі для обробки спеціальних зображень об'єктів.

Дослідження [9] підтверджує високі показники точності моделі YOLOv8 для детекції та ідентифікації різних типів транспортних засобів. Автори підкреслюють ефективність моделі YOLOv8 та перспективність її практичного впровадження у системи нагляду та управління дорожнім рухом. Дослідження підтверджує високу точність моделі YOLOv8 після донавчання.

Аналіз сучасних досліджень демонструє активний розвиток засобів детекції зображень автомобілів за допомогою нейронних мереж з архітектурою YOLO. Показано, що результати детектування об'єктів моделями YOLO різних версій можуть суттєво відрізнятися, тому існує проблема вибору оптимальної версії нейромережі. З метою підвищення точності детектування застосовується оптимізація архітектури нейромережі для детекції специфічних об'єктів, донавчання нейромереж на спеціалізованих датасетах та впровадження додаткових алгоритмів для покращення точності детекції об'єктів з малими розмірами [10]. В еволюції архітектури YOLO доцільно виділити версію YOLOv8 (після цієї версії відбулися суттєві зміни API) та версію YOLOv11 (одна з найновіших та найпоширеніших).

Метою роботи є аналіз ефективності детекції автомобілів за допомогою нейронних мереж YOLOv8 та YOLOv11, обґрунтування вибору оптимальної версії нейромережі за критеріями точності та швидкодії, а також дослідження впливу донавчання моделей YOLO на точність детекції.

1. Оцінка точності детектування об'єктів на зображеннях

З метою комплексної оцінки точності детектування зображень об'єктів за допомогою різних версій нейронної мережі YOLO використано ряд поширених метрик, які дають змогу об'єктивно оцінити переваги та недоліки моделей нейромереж при детекції об'єктів. Використано такі метрики: повнота (Recall), точність (Precision), IOU, mAP50, mAP60-95 [2, 11].

Метрика повноти (Recall) визначає, яку частку всіх реальних об'єктів модель успішно виявила. Метрика розраховується за наступною формулою:

$$Recall = \frac{TP}{TP+FN}, \quad (1)$$

де TP (True Positives) – кількість правильно виявлених об'єктів, FN (False Negatives) – кількість реальних об'єктів, які модель не змогла виявити.

Повнота вимірює здатність моделі детекції знаходити всі об'єкти на зображенні. Висока повнота означає, що модель виявляє майже всі шукані об'єкти, які містяться на зображенні. Низька повнота означає, що модель не виявляє значну частину об'єктів. Метрика повноти є критичною у випадках, коли пропуск важливого об'єкта є більш проблемним, ніж хибна детекція (наприклад, виявлення пішоходів у системах керування автономних автомобілів).

Метрика точності (Precision) вимірює частку правильно виявлених об'єктів серед усіх об'єктів, які модель детектує на зображенні, згідно з формулою:

$$Precision = \frac{TP}{TP+FP}, \quad (2)$$

де TP (True Positives) – кількість правильно виявлених об'єктів, FP (False Positives) – кількість хибно виявлених об'єктів (детектовані моделлю об'єкти, які насправді відсутні на зображенні).

Точність відображає, наскільки точні передбачення моделі щодо виявлених об'єктів. Висока точність означає, що модель майже не робить помилок, приймаючи фонові об'єкти за цільові. Низька точність свідчить про велику кількість хибно виявлених об'єктів.

Точність (Precision) тісно пов'язана з іншою метрикою – повнотою (Recall). Для досягнення балансу між цими двома метриками часто використовують F1-міру:

$$F1 = 2 \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (3)$$

Метрика IoU (Intersection over Union – Перетин над Об'єднанням) описує міру збігу передбаченої моделлю та реальної прямокутної рамки (bounding box) для виявленого об'єкту:

$$IoU = \frac{|A \cap B|}{|A \cup B|}, \quad (4)$$

де $|A \cap B|$ – площа перетину двох рамок, $|A \cup B|$ – площа їхнього об'єднання.

Чим вище значення IoU , тим точніше модель визначила положення об'єкта на зображенні.

Метрика Mean Average Precision $IoU = 0.5$ (mAP50) є одною з основних, що використовуються для оцінки якості детекції об'єктів у моделях глибокого навчання (зокрема, в YOLO). Дана метрика обчислює середнє значення точності AP (Average Precision) для всіх класів об'єктів при фіксованому порозі $IoU = 0.5$. Тобто, об'єкт вважається правильно детектованим (True Positive, TP), якщо його передбачена рамка перекривається з реальною хоча б на 50%. Значення середньої точності AP для одного класу обчислюється за формулою:

$$AP = \int_0^1 P(R) dR, \quad (5)$$

де $P(R)$ – значення метрики Precision як функція від Recall, інтегрування здійснюється за всіма значеннями Recall від 0 до 1.

Значення метрики mAP50 визначаються за формулою:

$$mAP50 = \frac{1}{N} \sum_{i=1}^N AP_i, \quad (6)$$

де N – кількість класів об'єктів, AP_i – середня точність для i -го класу.

Однак високе значення mAP50 не завжди означає точну локалізацію об'єктів. Тому також використовують метрику mAP50-95, яка враховує більше рівнів IoU (від 0.5 до 0.95 з кроком 0.05).

2. Створення набору даних для донавчання нейронної мережі

У даному дослідженні створено варіативний набір даних (датасет) з високим рівнем генералізації, що дозволяє донавчати й адаптувати сучасні моделі YOLOv8m та YOLOv11m до задачі детекції автомобілів у складних дорожніх сценах. Датасет сформовано шляхом вибору та анотування 1542 зображень, серед яких 1002 містили автомобілі, а 540 – не містили. Сформований датасет зображень поділено на навчальний набір (1018 зображень), контрольний набір (420 зображень) та тестовий набір (104 зображення). Значна увага приділена різноманіттю сценаріїв для зображень дорожнього руху, що охоплюють зміни розмірів зображень, освітлення, погодних умов, дистанцій, напрямків руху автомобілів та інших факторів.

Виділення положення об'єктів на зображеннях (анотація зображень) виконувалося в ручному режимі за допомогою інструменту Roboflow [12], що дозволило точно встановити координати обмежувальних прямокутних рамок для кожного автомобіля. З метою забезпечення повноти представленості різноманітних ситуацій встановлено такі вимоги до характеристик зображень датасету:

- Відсоток заповнення кадру транспортним засобом: від повнокадрових зображень автомобіля до кадрів, де транспортний засіб займає мінімальну частину сцени.
- Якість зображення: містить як високоякісні чіткі зображення, так і кадри з низькою роздільною здатністю або високим рівнем шуму [13].
- Інтенсивність трафіку: сцени з низькою, середньою та високою щільністю дорожнього руху.
- Час доби: денні, вранішні, вечірні та нічні умови освітлення для моделювання різних сценаріїв експлуатації.
- Погодні умови: сонячна погода, дощ, сніг, а також інші атмосферні ефекти, які можуть впливати на видимість.
- Дистанція до камери та деталізація: зображення з добре видимими деталями автомобіля та такі, на яких деталі розпізнаються погано.
- Основний напрямок руху транспортного засобу: рух у напрямку камери, від камери або під кутом до оптичної осі камери.
- Кут нахилу камери: зйомка зверху вниз, паралельно до земної поверхні або під кутом.

Доповнюючи вищенаведений перелік, також враховано додаткові параметри для покращення варіативності датасету:

- Оклюзії: часткове перекриття транспортного засобу іншими об'єктами сцени.
- Фон сцени: урбаністичне середовище, замські дороги, парковки, тунелі тощо.
- Артефакти зображення: відблиски від сонця чи фар, тіні, засвітки.
- Стан транспортного засобу: чистий або забруднений автомобіль, наявність пошкоджень кузова.
- Швидкість руху транспортного засобу: нерухомий, повільно рухається або рухається на високій швидкості, що може впливати на чіткість зображення автомобіля.

Такий підхід до формування датасету дозволяє забезпечити максимальну варіативність

навчальних даних, що, у свою чергу, підвищує здатність моделі до генералізації та забезпечує її ефективність у різноманітних умовах реального середовища.

Формування власного датасету виконано на основі відео з датасету «Highway Traffic Videos Dataset» [14] Розроблено скрипт, який зчитував і зберігав окремі кадри автомобільного трафіку з відео (рис. 1).



Highway Traffic Videos Dataset

Рис. 1. Приклад різної інтенсивності автомобільного трафіку (низька, середня, висока) на відео датасету «Highway Traffic Videos Dataset»

Датасет було аугментовано за допомогою інструменту Roboflow: додано зображення з розмиттям, поворотом, мозаїкою, різним рівнем яскравості та насичення. У результаті аугментації кількість зображень датасету збільшено до 3578. Аугментації дозволяє створити додаткову варіативність даних та зменшити ймовірність перенавчання нейромережі [15].

3. Програмна реалізація детекції об'єктів та донавчання нейронної мережі YOLO

Для підвищення ефективності моделей YOLO у реальних умовах було реалізовано їх донавчання (fine-tuning) на основі сформованого датасету. Виконано трансферне навчання нейромережі, тобто попередньо навчені ваги моделей були уточнені на розробленому наборі зображень. Це дозволило значно прискорити конвергенцію моделей та досягти високої точності детекції.

Програму для навчання нейромережі та детекції об'єктів розроблено на мові Python у хмарному сервісі Google Colab за допомогою записника Jupyter Notebook. В програмі використано ряд бібліотек, зокрема opencv, ultralytics, numpy, matplotlib та ін. Реалізацію моделей YOLO виконано з використанням бібліотеки Ultralytics [16], яка забезпечує широкий вибір параметрів та гіперпараметрів навчання нейромережі. У програмі використано моделі YOLO середнього розміру (medium), а саме моделі YOLOv8m та YOLOv11m.

На початковому етапі донавчання моделей YOLO виконувався на CPU (тривалість 1 епохи навчання – 1 година 9 хвилин). Після цього навчання виконувалося на графічному процесорі NVIDIA Tesla T4 [17], що дозволило скоротити тривалість 1 епохи навчання до 1 хвилини 37 секунд. Використання GPU забезпечило майже 40-кратне прискорення навчання, що пояснюється розпаралелюванням обчислень.

4. Оцінка точності та швидкодії детектування зображень об'єктів за допомогою згорткової нейронної мережі YOLO

На основі створеного програмного забезпечення було проведено донавчання моделей YOLOv8m та YOLOv11m на розробленому датасеті, поділеному на навчальний, контрольний (валідаційний) та тестовий набори даних. У даній роботі початкові (оригінальні, без донавчання) моделі YOLO позначено як Pre-trained, а моделі YOLO після донавчання позначено Fine-tuned.

За часом навчання модель YOLOv8m порівняно з YOLOv11m показала вищу швидкодію. У середньому одна епоха в YOLOv8m тривала 1 хвилину 14 секунд, а в YOLOv11m одна епоха тривала 1 хвилину 18 секунд. Загалом донавчання моделі YOLOv11m тривало 1 годину 21 хвилин, а донавчання YOLOv8m тривало 1 годину 10 хвилин. Після донавчання моделей YOLOv8m та YOLOv11m проведено їх тестування на незалежній тестовій вибірці. Для об'єктивного порівняння також було протестовано початкові (оригінальні, не дотреновані) версії цих моделей. Оцінювання здійснювалося за основними метриками: Precision, Recall, mAP50 та mAP50-95 (рис. 2, табл. 1).

Дотреновані моделі показали суттєво вищі значення всіх метрик (вищу точність детекції) завдяки адаптації до розробленого датасету. Підвищення значень метрики Recall після дотренування означає, що модель стала краще виявляти всі об'єкти (автомобілі) на зображеннях. Найбільше покращення отримано для метрики mAP50-95 (майже вдвічі), що підтверджує значне покращення точності локалізації об'єктів навіть різних розмірів.

Після завершення процесу дотренування моделі YOLOv8m (Fine-tuned) та YOLOv11m (Fine-tuned) збережено у спеціальні файли формату .pt, які містять параметри навченої нейромережі. Розмір цих файлів служить важливим індикатором складності та ефективності моделі. Розміри дотренованих моделей такі:

- YOLOv11m (Fine-tuned) – 38,7 МБ.
- YOLOv8m (Fine-tuned) – 49,6 МБ.

Менший розмір файлу дотренованої моделі YOLOv11m (Fine-tuned) свідчить про її оптимізовану архітектуру, яка використовує меншу кількість параметрів або більш ефективні структури шарів. Це є наслідком використання покращених методів операцій згортки, стиснення або квантування ваг. Компактніші моделі потребують менше пам'яті та володіють вищою швидкістю на пристроях із обмеженими ресурсами (наприклад, у вбудованих системах). Висока швидкість є особливо важливою для детекції зображень в режимі реального часу.

Модель YOLOv11m (Fine-tuned) в середньому обробляла зображення швидше, ніж модель YOLOv8m (Fine-tuned). Для YOLOv11m (Fine-tuned) середній час детекції 1 кадру рівний 0.047 с, а для YOLOv8m (Fine-tuned) рівний 0.050 с. Підвищення швидкості детекції пов'язано з розміром моделі, оскільки дотренована модель YOLOv11m (Fine-tuned) потребує майже на 10 МБ менше пам'яті, ніж YOLOv8m (Fine-tuned).

Значення метрик точності для моделей YOLOv8m (Fine-tuned) та YOLOv11m (Fine-tuned) після донавчання відрізняються незначно, що в значній мірі зумовлено особливістю використаного датасету (який характеризується високою різноманітністю використаних зображень). Вибір між YOLOv8m (Fine-tuned) та YOLOv11m (Fine-tuned) залежить від конкретного випадку використання. Якщо критичною є висока точність, то доцільно використовувати YOLOv8m. Якщо ж пріоритетом є швидкість та оптимальне використання ресурсів, то має перевагу модель YOLOv11m.

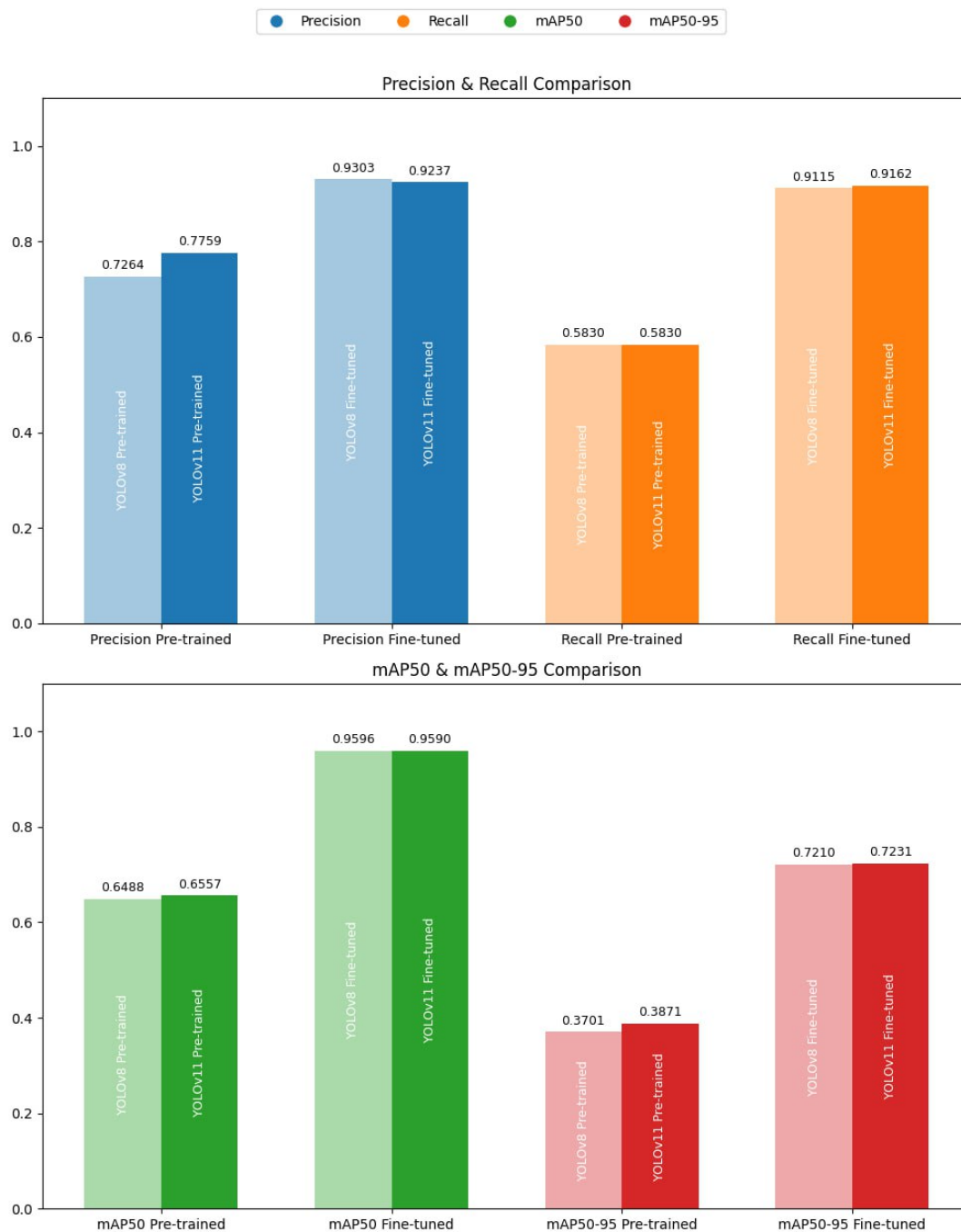


Рис. 2. Значення метрик Precision, Recall, mAP50 та mAP50-95 для початкових (Pre-trained) та дотренованих (Fine-tuned) моделей YOLOv8m та YOLOv11m на тестовому наборі

Таблиця 1

Значення метрик Precision, Recall, mAP50 та mAP50-95 для початкових (Pre-trained) та дотренованих (Fine-tuned) моделей YOLO8m та YOLO11m на тестовому наборі

Метрика	YOLOv8m (Pre-trained)	YOLOv11m (Pre-trained)		YOLOv8m (Fine-tuned)	YOLOv11m (Fine-tuned)
Precision	0.7264	0.7759		0.9303	0.9237
Recall	0.5830	0.5830		0.9115	0.9162
mAP50	0.6488	0.6557		0.9596	0.9590
mAP50-95	0.3701	0.3871		0.7210	0.7231

Для оцінки ефективності розпізнавання об'єктів у реальних умовах було протестовано початкові та дотреновані моделі YOLOv8m та YOLOv11m на відеозаписах із автомобілями. Крім виявлення автомобілів, також аналізувалася якість трекінгу їхнього руху за допомогою бібліотеки OpenCV. Дотреновані моделі чітко відокремлюють автомобілі навіть у складних сценаріях, коли автомобілі знаходяться поруч (рис. 3). Точність детекції автомобілів на відео дотренованими моделями YOLOv8m (Fine-tuned) та YOLOv11m (Fine-tuned) за використаними метриками (зокрема, за метрикою mAP50-95) практично співпадає.



Рис. 3. Приклад кадру відео з детекцією та трекінгом автомобілів за допомогою дотренованої моделі YOLOv8m (Fine-tuned)

На основі обробки відео отримано, що дотреновані моделі YOLOv8m (Fine-tuned) та YOLOv11m (Fine-tuned) демонструють плавну та стабільну побудову траєкторій руху автомобілів. Завдяки покращеній детекції, трекінг менше схильний до стрибків та втрати об'єктів у складних умовах (наприклад, коли автомобілі змінюють напрямок або тимчасово перекривають один одного). Таким чином, завдяки донавчанню моделей YOLO вдалося підвищити точність виявлення та відстеження автомобілів у порівнянні з початковими моделями.

Висновки з даного дослідження

і перспективи подальших розвідок у даному напрямі

У ході проведеного дослідження сформовано високоякісний датасет із 1542 зображень, з яких 1002 зображень містять автомобілі, а 540 зображень їх не містять. Зображення датасету характеризуються різною інтенсивністю трафіку, погодними умовами, часом доби, варіаціями фону, перекриванням транспортних засобів, їх швидкістю та напрямом руху. Особлива увага приділена ручній анотації зображень, що забезпечило високу точність локалізації об'єктів. Сформований датасет зображень поділено на навчальний (1018 зображень), контрольний (420 зображень) та тестовий набори (104 зображення). Контрольний набір використано для запобігання перенавчання нейронних мереж. За рахунок аугментації даних розмір датасету збільшено до 3578 зображень. На сформованому датасеті навчено моделі YOLOv8m та YOLOv11m, що дало змогу суттєво покращити точність детекції.

Аналіз метрик точності (Recall, Precision, mAP50, mAP50-95) на тестовому наборі даних продемонстрував суттєве зростання їх значень після донавчання у порівнянні з початковими моделями. Зокрема, модель YOLOv8m (Fine-tuned) показала кращі результати у Recall та mAP50-95, що свідчить про її здатність виявляти більшу кількість об'єктів у складних умовах. Модель YOLOv11m (Fine-tuned) відзначилась вищою точністю, демонструючи менше хибнопозитивних спрацювань. У загальному після донавчання обидві моделі показали близькі значення за метриками точності детекції. Результати

дослідження підтверджують, що навчання моделей YOLO на сформованому датасеті суттєво підвищує їхню точність в умовах, максимально наближених до реальних. Показано, що дотреновані моделі YOLO можуть успішно застосовуватися не тільки для детектування автомобілів на зображеннях, але й для трекінгу автомобілів на відео.

У подальшому перспективним є удосконалення та розширення датасету. Включення до датасету зображень з новими сценаріями, зокрема, зображень аварійних ситуацій, нетипових транспортних засобів та пішоходів дозволить підвищити його універсальність. Наступним кроком може бути інтеграція донавичених моделей у системи відеоспостереження або в інтелектуальні транспортні системи з метою детекції автомобілів та учасників дорожнього руху в реальному часі. Точність детекції зображень об'єктів можна підвищити за рахунок попередньої обробки зображень, отриманих з відеокамер [18]. Глибокий аналіз хибнопозитивних та хибнонегативних спрацювань, отриманих при детекції об'єктів, допоможе вдосконалити підходи до аугментації даних та обрати ефективні стратегії подальшого навчання нейромереж.

Література

1. Stelmakh O.P. Information Technology of Video Data Processing for Traffic Intensity Monitoring / O.P. Stelmakh, I.V. Stetsenko, D.V. Velyhotskiy // *Control systems & computers*. – 2020. – № 3. – С. 50-59. doi: [10.15407/usim.2020.03.050](https://doi.org/10.15407/usim.2020.03.050).
2. Mohammed Abduljabbar Zaid Al Bayati. Real-Time Vehicle Detection for Surveillance of River Dredging Areas Using Convolutional Neural Networks / Mohammed Abduljabbar Zaid Al Bayati, Muhammet Çakmak // *International Journal of Image, Graphics and Signal Processing (IJIGSP)*. – 2023. – Vol.15, No.5. – P. 17-28. doi:10.5815/ijigsp.2023.05.02.
3. Li Z. A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects / Z. Li, F. Liu, W. Yang, S. Peng, J. Zhou // *IEEE Transactions on Neural Networks and Learning Systems*. – 2022. – 33 (12). – P. 6999-7019. doi:10.1109/TNNLS.2021.3084827.
4. Lin T.-Y. Microsoft COCO: Common Objects in Context / T.-Y. Lin, M. Maire, S. Belongie // *European Conference on Computer Vision (ECCV)*. – 2014. – <https://arxiv.org/abs/1405.0312>.
5. Farid A. A Fast and Accurate Real-Time Vehicle Detection Method Using Deep Learning for Unconstrained Environments / A. Farid, F. Hussain, K. Khan, M. Shahzad, U. Khan, Z. Mahmood // *Applied Sciences*. – 2023. – Vol. 13, No. 5. – Article No. 3059. <https://www.mdpi.com/2076-3417/13/5/3059>.
6. Liu Z. Research on vehicle detection based on improved YOLOX_S / Z. Liu, W. Han, H. Xu, K. Gong, Q. Zeng, X. Zhao // *Scientific Reports*. – 2023. – Vol. 13. – Article No. 23081. <https://pmc.ncbi.nlm.nih.gov/articles/PMC10754853>.
7. Alahdal N.M. Real-time Object Detection in Autonomous Vehicles with YOLO / N.M. Alahdal, F. Abukhodair, L.H. Meftah, A. Cherif // *Procedia Computer Science*. – 2024. – Vol. 246. – P. 2792–2801. <https://www.sciencedirect.com/science/article/pii/S1877050924024293>.
8. Li X. Vehicle–Pedestrian Detection Method Based on Improved YOLOv8 / X. Li, Y. Wang, H. Zhang, J. Liu // *Electronics*. – 2024. – Vol. 13, No. 11. – Article No. 2149. <https://www.mdpi.com/2079-9292/13/11/2149>.
9. Tomas J.P.Q. YOLOv8-Based Vehicle Type Detection: An In-Depth Exploration of Deep Learning Techniques for Robust Analysis of Dashcam Footage / J.P.Q. Tomas, A. Aquino, G.E.M. David, R.D. Del Ayre, J.K.L. Alminiana // *Proceedings of the 2024 9th International Conference on Information and Communication Technology*. – 2024. – P. 91–96. <https://dl.acm.org/doi/10.1145/3670085.3670098>.
10. Balovsyak S. Car Image Recognition using Convolutional Neural Network with EfficientNet Architecture / S. Balovsyak, O. Kroitor, Kh. Odaiska, Abdel-Badeeh M. Salem, S. Stets // *IntelliTIS 2024: 5th International Workshop on Intelligent Information Technologies & Systems of Information Security with CEUR-WS, March 28, 2024. – Khmelnytskyi, Ukraine. – CEUR Workshop Proceedings / Edited by T. Hovorushchenko, O. Savenko, P. T. Popov, S. Lysenko. – 2024. – P. 182-195.*
11. Redmon J. YOLOv3: An Incremental Improvement / J. Redmon, A. Farhadi // *arXiv preprint*. – 2018. – arXiv:1804.02767. <https://arxiv.org/abs/1804.02767>.
12. Roboflow. Roboflow Annotation Tool. URL: <https://roboflow.com> (date of access: 10.03.2025).
13. Balovsyak S.V. Automatic Highly Accurate Estimation of Gaussian Noise Level in Digital Images Using Filtration and Edges Detection Methods / S.V. Balovsyak, Kh. S. Odaiska // *International Journal of Image, Graphics and Signal Processing (IJIGSP)*. – 2017. – Vol. 9, No.12. – P. 1-11. doi: 10.5815/ijigsp.2017.12.01.
14. Seattle DOT. Highway Traffic Video Dataset. URL: <https://data.seattle.gov> (date of access: 10.03.2025).
15. Bochkovskiy A. YOLOv4: Optimal Speed and Accuracy of Object Detection / A. Bochkovskiy, C.-Y. Wang, H.-Y. M. Liao // *arXiv preprint*. – 2020. – arXiv:2004. <https://arxiv.org/abs/2004.10934>.
16. Ultralytics. YOLOv8 Documentation. URL: <https://docs.ultralytics.com> (date of access: 11.03.2025).
17. NVIDIA. NVIDIA Tesla T4 GPU Architecture Overview. – 2023. <https://www.nvidia.com/en->

us/data-center/tesla-t4/

18. Balovsyak S. Adjusting the Brightness and Contrast parameters of digital video cameras using artificial neural networks / S. Balovsyak, Kh. Odaiska, O. Yakovenko, I. Iakovlieva // Proc. SPIE: Sixteenth International Conference on Correlation Optics. – 2024. – Vol. 12938. – P. 129380I-1 – 129380I-4.

References

1. Stelmakh O.P. Information Technology of Video Data Processing for Traffic Intensity Monitoring / O.P. Stelmakh, I.V. Stetsenko, D.V. Velyhotskiy // Control systems & computers. – 2020. – № 3. – C. 50-59. doi: 10.15407/usim.2020.03.050.
2. Mohammed Abduljabbar Zaid Al Bayati. Real-Time Vehicle Detection for Surveillance of River Dredging Areas Using Convolutional Neural Networks / Mohammed Abduljabbar Zaid Al Bayati, Muhammet Çakmak // International Journal of Image, Graphics and Signal Processing (IJIGSP). – 2023. – Vol.15, No.5. – P. 17-28. doi:10.5815/ijigsp.2023.05.02.
3. Li Z. A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects / Z. Li, F. Liu, W. Yang, S. Peng, J. Zhou // IEEE Transactions on Neural Networks and Learning Systems. – 2022. – 33 (12). – P. 6999-7019. doi:10.1109/TNNLS.2021.3084827.
4. Lin T.-Y. Microsoft COCO: Common Objects in Context / T.-Y. Lin, M. Maire, S. Belongie // European Conference on Computer Vision (ECCV). – 2014. – https://arxiv.org/abs/1405.0312.
5. Farid A. A Fast and Accurate Real-Time Vehicle Detection Method Using Deep Learning for Unconstrained Environments / A. Farid, F. Hussain, K. Khan, M. Shahzad, U. Khan, Z. Mahmood // Applied Sciences. – 2023. – Vol. 13, No. 5. – Article No. 3059. <https://www.mdpi.com/2076-3417/13/5/3059>.
6. Liu Z. Research on vehicle detection based on improved YOLOX_S / Z. Liu, W. Han, H. Xu, K. Gong, Q. Zeng, X. Zhao // Scientific Reports. – 2023. – Vol. 13. – Article No. 23081. <https://pmc.ncbi.nlm.nih.gov/articles/PMC10754853>.
7. Alahdal N.M. Real-time Object Detection in Autonomous Vehicles with YOLO / N.M. Alahdal, F. Abukhodair, L.H. Meftah, A. Cherif // Procedia Computer Science. – 2024. – Vol. 246. – P. 2792–2801. <https://www.sciencedirect.com/science/article/pii/S1877050924024293>.
8. Li X. Vehicle–Pedestrian Detection Method Based on Improved YOLOv8 / X. Li, Y. Wang, H. Zhang, J. Liu // Electronics. – 2024. – Vol. 13, No. 11. – Article No. 2149. <https://www.mdpi.com/2079-9292/13/11/2149>.
9. Tomas J.P.Q. YOLOv8-Based Vehicle Type Detection: An In-Depth Exploration of Deep Learning Techniques for Robust Analysis of Dashcam Footage / J.P.Q. Tomas, A. Aquino, G.E.M. David, R.D. Del Ayre, J.K.L. Alminiana // Proceedings of the 2024 9th International Conference on Information and Communication Technology. – 2024. – P. 91–96. <https://dl.acm.org/doi/10.1145/3670085.3670098>.
10. Balovsyak S. Car Image Recognition using Convolutional Neural Network with EfficientNet Architecture / S. Balovsyak, O. Kroitor, Kh. Odaiska, Abdel-Badeeh M. Salem, S. Stets // IntellITSIS 2024: 5th International Workshop on Intelligent Information Technologies & Systems of Information Security with CEUR-WS, March 28, 2024. – Khmelnytskyi, Ukraine. – CEUR Workshop Proceedings / Edited by T. Hovorushchenko, O. Savenko, P. T. Popov, S. Lysenko. – 2024. – P. 182-195.
11. Redmon J. YOLOv3: An Incremental Improvement / J. Redmon, A. Farhadi // arXiv preprint. – 2018. – arXiv:1804.02767. <https://arxiv.org/abs/1804.02767>.
12. Roboflow. Roboflow Annotation Tool. URL: <https://roboflow.com> (date of access: 10.03.2025).
13. Balovsyak S.V. Automatic Highly Accurate Estimation of Gaussian Noise Level in Digital Images Using Filtration and Edges Detection Methods / S.V. Balovsyak, Kh. S. Odaiska // International Journal of Image, Graphics and Signal Processing (IJIGSP). – 2017. – Vol. 9, No.12. – P. 1-11. doi: 10.5815/ijigsp.2017.12.01.
14. Seattle DOT. Highway Traffic Video Dataset. URL: <https://data.seattle.gov> (date of access: 10.03.2025).
15. Bochkovskiy A. YOLOv4: Optimal Speed and Accuracy of Object Detection / A. Bochkovskiy, C.-Y. Wang, H.-Y. M. Liao // arXiv preprint. – 2020. – arXiv:2004. <https://arxiv.org/abs/2004.10934>.
16. Ultralytics. YOLOv8 Documentation. URL: <https://docs.ultralytics.com> (date of access: 11.03.2025).
17. NVIDIA. NVIDIA Tesla T4 GPU Architecture Overview. – 2023. <https://www.nvidia.com/en-us/data-center/tesla-t4/>
18. Balovsyak S. Adjusting the Brightness and Contrast parameters of digital video cameras using artificial neural networks / S. Balovsyak, Kh. Odaiska, O. Yakovenko, I. Iakovlieva // Proc. SPIE: Sixteenth International Conference on Correlation Optics. – 2024. – Vol. 12938. – P. 129380I-1 – 129380I-4.