



УДК 004.932.2:004.896:004.94

ВПЛИВ РОЗДІЛЬНОЇ ЗДАТНОСТІ ВХІДНИХ ЗОБРАЖЕНЬ НА ПАРАМЕТРИ МОДЕЛЕЙ YOLO ПРИ ДЕТЕКТУВАННІ ОБ'ЄКТІВ

IMPACT OF INPUT IMAGE RESOLUTION ON YOLO MODEL PARAMETERS FOR OBJECT DETECTION

Щебель Ю. Р.**Shchebel Y. R.**

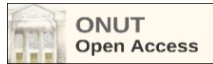
Приватний вищий навчальний заклад «Європейський університет», Київ, Україна

Private Higher Education Establishment “European University”, Kyiv, Ukraine

ORCID: <https://orcid.org/0009-0005-1889-4422>E-mail: yu.schebel@e-u.edu.ua

Copyright © 2025 by author and the journal “Automation of technological and business – processes”.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0>

DOI: 10.15673/atbp.v17i2.3115

Анотація. У статті представлено результати дослідження впливу роздільної здатності вхідних зображень на ключові параметри моделей глибокого навчання YOLOv5 і YOLOv8 при виконанні завдань детектування об'єктів. З урахуванням широкого розповсюдження моделей YOLO у сфері комп'ютерного бачення та автоматизації, особливо в системах реального часу, актуальним є питання визначення оптимальних параметрів їх застосування для забезпечення балансу між точністю виявлення об'єктів та продуктивністю систем. У дослідженні було проведено серію експериментів із використанням набору даних COCO128 та вхідних зображень з роздільною здатністю в межах від 256x256 до 1280x1280 пікселів. У процесі дослідження аналізувалися такі метрики, як точність (Precision), повнота (Recall), середня точність при порозі IoU=0.5 (mAP@0.5), середня точність при порогах IoU від 0.5 до 0.95 (mAP@0.5:0.95), час інференції (Inference Time) та частота обробки кадрів (FPS). За допомогою апроксимаційного моделювання було побудовано залежності між зазначеними параметрами та роздільною здатністю вхідних даних. Результати дослідження показали, що збільшення роздільної здатності позитивно впливає на якість виявлення об'єктів, однак суттєво знижує швидкість моделей, що є критичним фактором для систем реального часу. Найбільш оптимальні параметри для використання моделей YOLOv5 досягаються при роздільній здатності вхідних зображень не вище 512 пікселів, що забезпечує прийнятний баланс між точністю та продуктивністю. Для моделі YOLOv8 рекомендовано використовувати роздільну здатність в діапазоні 640–800 пікселів, що дозволяє досягти високих показників точності при прийнятній продуктивності. Отримані результати можуть бути використані при проектуванні систем відеоспостереження, автономних роботизованих систем, інтелектуальних транспортних засобів та інших прикладних рішень, де важливо забезпечити оптимальне поєднання точності розпізнавання та обчислювальної ефективності.

Abstract. The article presents the results of a study on the impact of input image resolution on key parameters of deep learning models YOLOv5 and YOLOv8 in the context of object detection tasks. Given the widespread use of YOLO models in the field of computer vision and automation, particularly in real-time systems, the issue of determining optimal deployment parameters to balance object detection accuracy and system performance is highly relevant. The study involved a series of experiments using the COCO128 dataset and input images with resolutions ranging from 256x256 to 1280x1280 pixels. During the research, such metrics as Precision, Recall, mean Average Precision at an IoU threshold of 0.5 (mAP@0.5), mean Average Precision averaged over IoU thresholds from 0.5 to 0.95 (mAP@0.5:0.95), Inference Time, and Frames Per Second (FPS) were analyzed. Using approximation modeling, dependencies between these parameters and the input image resolution were established. The results demonstrated that increasing resolution positively affects object detection quality but significantly reduces model inference speed, which is a critical factor for real-time systems. The optimal parameters for using YOLOv5 models are achieved at input resolutions not exceeding 512 pixels, ensuring an acceptable balance between accuracy and performance. For YOLOv8, it is recommended to use resolutions in the range of 640–800 pixels, which allows achieving high detection accuracy while maintaining acceptable computational efficiency. The obtained results can be applied in the design of video surveillance systems, autonomous



robotic platforms, intelligent transportation systems, and other practical solutions where achieving an optimal combination of recognition accuracy and computational efficiency is essential.

Ключові слова: розпізнавання об'єктів, глибоке навчання, YOLOv5, YOLOv8, продуктивність моделей, комп'ютерне бачення.

Keywords: object detection, deep learning, YOLOv5, YOLOv8, model performance, computer vision.

Вступ

Автоматичне розпізнавання та детектування об'єктів на зображеннях є однією з ключових задач у сфері комп'ютерного бачення та штучного інтелекту. Її актуальність зумовлена широким спектром практичних застосувань, включаючи системи відеоспостереження, автономні транспортні засоби, медичну діагностику, промислову автоматизацію та засоби безпеки. В умовах стрімкого розвитку цифрових технологій та зростання обсягів візуальних даних потреба у високотехнологічних рішеннях для ефективної обробки зображень лише посилюється. При цьому важливо забезпечити не лише високу точність виявлення об'єктів, а й достатню швидкодію моделей для використання в реальних системах, що мають обмежену продуктивність.

Одним із важливих факторів, що визначає ефективність систем розпізнавання, є роздільна здатність вхідних зображень. Зменшення роздільної здатності дозволяє суттєво знизити обчислювальне навантаження та збільшити швидкість обробки даних, однак це може призвести до втрати важливих візуальних ознак і зниження якості виявлення об'єктів. З іншого боку, надмірно висока роздільна здатність покращує деталізацію, але значно збільшує час обробки та ресурсні витрати, що є критичним для систем, орієнтованих на роботу в реальному часі.

Таким чином, актуальним є дослідження впливу роздільної здатності вхідних зображень на показники точності та продуктивності сучасних моделей детектування об'єктів, що дозволить сформулювати практичні рекомендації щодо вибору оптимальних параметрів для конкретних завдань автоматизованого аналізу зображень.

Аналіз літературних даних і постановка проблеми

Одним із основних досліджень у цій сфері є огляд сучасних методів детекції об'єктів, у якому було систематизовано основні підходи та архітектури, зокрема R-CNN, Fast R-CNN, Faster R-CNN, SSD та моделі сімейства YOLO (You Only Look Once) [1]. Особливе місце в сучасних дослідженнях займають моделі YOLO, які демонструють високі показники точності при збереженні здатності працювати в реальному часі. Доволі добре досліджені особливості третьої версії YOLO в роботі YOLOv3: An Incremental Improvement [2]. Добре вивченою є також YOLOv4, де було впроваджено численні техніки оптимізації для досягнення балансу між продуктивністю та точністю навіть при обмежених апаратних ресурсах: YOLOv4: Optimal Speed and Accuracy [3]. На жаль, дані версії є застарілими, а наступна стабільна і добре вивчена – YOLOv5 все ще активно використовується завдяки своїй гнучкості, оптимізації обчислень і відкритому вихідному коду [4]. Новішою та актуальною є версія YOLOv8, що має вдосконалений механізм обробки зображень, покращені механізми об'єднання ознак і більш високу точність завдяки оновленій архітектурі YOLOv8 [5].

Деякі дослідники розглядали вплив роздільної здатності вхідних зображень на якість класифікації та детектування об'єктів. Наприклад, у роботі EfficientNet: Rethinking Model Scaling [6] було запропоновано методику однорідного масштабування розмірів вхідних даних разом із глибиною та шириною нейромереж, що дозволяє підвищити точність класифікації. Однак це дослідження зосереджене переважно на задачах класифікації, а не детектуванні. В іншій роботі MobileNets: Efficient Convolutional Neural Networks [7] було розглянуто вплив розміру зображень на продуктивність мобільних моделей, але без поглибленого аналізу точності при різних роздільних здатностях у задачах виявлення об'єктів. Попри наявність окремих досліджень у цьому напрямку, питання комплексного порівняння моделей YOLOv5 та YOLOv8 щодо впливу роздільної здатності вхідних зображень на точність виявлення об'єктів і продуктивність моделей залишається недостатньо вивченим. Зокрема, відсутні систематизовані експериментальні дослідження, які дозволяють визначити оптимальні параметри роздільної здатності для реальних практичних застосувань, таких як системи відеоспостереження, автономні роботизовані системи та засоби моніторингу. Таким чином, залишається відкритою проблема аналізу компромісу між точністю розпізнавання об'єктів і швидкодією моделей в залежності від роздільної здатності вхідних зображень, що є критично важливим для побудови ефективних систем обробки візуальної інформації в режимі реального часу.

Мета і завдання дослідження

Метою дослідження є визначення впливу роздільної здатності вхідних зображень на точність розпізнавання об'єктів і продуктивність моделей YOLOv5 та YOLOv8. Це дозволить сформулювати практичні рекомендації щодо вибору оптимальних параметрів роздільної здатності для застосування в реальних системах обробки візуальної інформації.

Для досягнення поставленої мети були визначені такі завдання дослідження:

- Виконати навчання моделей YOLOv5 та YOLOv8 на єдиному датасеті при різних значеннях роздільної здатності вхідних зображень.
- Визначити параметри моделей, що будуть досліджуватись.
- Провести експериментальну оцінку моделей за визначеними параметрами.
- Побудувати графічні залежності ключових параметрів від роздільної здатності.
- Здійснити апроксимацію отриманих результатів для виявлення оптимальних значень параметрів.



Методи і матеріали досліджень

Дослідження проводилося в середовищі Google Colab [8], яке забезпечує зручний хмарний доступ до сучасних обчислювальних ресурсів, включаючи графічні процесори (GPU). Вибір Google Colab обумовлений його безкоштовністю, доступністю високопродуктивного апаратного забезпечення та простотою інтеграції з популярними бібліотеками машинного навчання.

Особливу увагу було приділено обчислювальним ресурсам, адже продуктивність обраної апаратної платформи безпосередньо впливає на тривалість навчання моделей і час інференції. Для навчання та тестування моделей використовувався графічний процесор NVIDIA Tesla T4, який підтримується у середовищі Google Colab, оскільки:

- GPU Tesla T4 є збалансованим рішенням між високою продуктивністю та помірним енергоспоживанням.
- Потужність обчислень дозволяє ефективно навчати навіть складні моделі глибинного навчання.
- Наявність апаратного прискорення для операцій згортки (Convolution), які є базовими для моделей комп'ютерного бачення.
- Широка доступність цього GPU у Google Colab дозволяє уникнути додаткових витрат на платні сервіси.

Завдяки таким характеристикам Tesla T4 є ефективним вибором для досліджень, пов'язаних з обробкою зображень і навчанням моделей нейронних мереж.

Програмне середовище дослідження:

Мова програмування: Python 3.11 – через широку підтримку бібліотек для глибинного навчання.

Бібліотеки: Ultralytics, PyTorch, Pandas, Matplotlib, NumPy – для побудови, навчання моделей і обробки отриманих результатів.

У дослідженні було обрано дві актуальні архітектури для детектування об'єктів – YOLOv5 [4] та YOLOv8 [5]. Дослідження проводилося послідовно в два основних етапи: спочатку було виконано навчання моделей YOLOv5, далі – проведено аналогічні експерименти з моделями YOLOv8 для можливості коректного порівняння результатів. Вибір саме цих моделей обумовлений декількома факторами. По-перше, ці моделі представляють різні покоління розвитку алгоритмів сімейства YOLO, що дозволяє оцінити вплив архітектурних змін на продуктивність і точність. По-друге, вони мають відкритий вихідний код, активно підтримуються спільнотою розробників і мають ґрунтовну документацію.

Аргументація вибору YOLOv5:

- Стабільна версія з широким застосуванням у промисловості та наукових дослідженнях.
- Відкрите програмне забезпечення з численними прикладами використання.
- Гнучка настройка гіперпараметрів для дослідження різних конфігурацій моделей.

Аргументація вибору YOLOv8:

- Остання стабільна версія від компанії Ultralytics, що має покращені механізми об'єднання ознак та оптимізовану архітектуру.
- Підтримка нових функціональних можливостей, таких як оптимізовані стратегії навчання та більш висока точність на стандартних датасетах.
- Новіші версії YOLO (наприклад, експериментальні YOLOv9, v10, v11, v12) не були включені до дослідження через їхню нестабільність, недостатню підтримку або надмірні обчислювальні вимоги [9].

Для дослідження було обрано компактний датасет COCO128, який є підмножиною відомого датасету COCO (Common Objects in Context) [10]. Основна перевага COCO128 полягає у зменшеному обсязі даних, що значно пришвидшує процес навчання моделей та дозволяє оперативно проводити серію експериментів [11]. Незважаючи на обмежений обсяг, цей датасет містить зображення з високою якістю розмітки, що дозволяє отримати коректні результати для порівняльного аналізу моделей.

З метою дослідження впливу роздільної здатності вхідних зображень на точність та швидкість моделей було обрано такі значення: 256x256, 320x320, 512x512, 640x640, 800x800, 1280x1280 пікселів. Такий вибір дозволяє охопити широкий діапазон сценаріїв застосування: від мобільних пристроїв із обмеженими ресурсами до потужних систем відеоспостереження та автономних роботизованих платформ. Вибрані значення базуються на рекомендаціях, наведених у роботах [6], [7], де відзначено важливість адаптації роздільної здатності під конкретні задачі для досягнення оптимального співвідношення між точністю та продуктивністю моделей.

У ході експериментів було обрано ключові метрики для оцінки продуктивності моделей. Ці параметри є загальноновизнаними в науковій спільноті та використовуються для оцінки якості моделей в задачах комп'ютерного бачення:

- Resolution (RS) – Роздільна здатність вхідних зображень. Цей параметр безпосередньо впливає на кількість інформації, яку отримує модель. Чим вища роздільна здатність, тим більше дрібних деталей присутні на зображенні, що може покращити точність розпізнавання. Проте, збільшення роздільної здатності призводить до зростання обчислювальних витрат і часу інференції.

- Precision (PS) – Точність розпізнавання. Обчислюється за формулою:

$$PS = \frac{TP}{TP + FP}$$

де TP – кількість істинно позитивних передбачень, FP – кількість хибно позитивних.

Високе значення Precision означає, що модель майже не робить помилкових детекцій.

- Recall (RC) – Повнота розпізнавання. Розраховується за формулою:

<http://www.atbp.ontu.edu.ua/>

$$RC = \frac{TP}{TP + FN}$$

де FN – кількість хибно негативних передбачень.

Recall показує, яку частку об'єктів модель виявила з усіх наявних на зображеннях.

– Mean Average Precision at IoU threshold 0.5 (mAP@0.5) – Середня точність при пороговому значенні Intersection over Union (IoU) = 0.5. IoU – це показник, що оцінює якість локалізації передбачених об'єктів. Формально IoU визначається як відношення площі перетину передбаченої рамки та еталонної до площі їх об'єднання. Параметр mAP@0.5 узагальнює точність локалізації об'єктів при достатньому перекритті передбаченої рамки з еталонною (щонайменше 50% співпадіння). Використовується для початкової оцінки якості моделей. Важливість метрики mAP як основного критерію оцінки ефективності моделей детекції була підтверджена в численних дослідженнях, зокрема і також при розробці архітектур ResNet, що використовуються як основи багатьох сучасних моделей [12].

– Mean Average Precision averaged over IoU thresholds from 0.5 to 0.95 with a step of 0.05 (mAP@0.5:0.95) – Середня точність, усереднена за порогами IoU від 0.5 до 0.95. Це суворіша метрика, рекомендована для комплексної оцінки моделей [11]. Вона дозволяє оцінити якість локалізації об'єктів на різних рівнях перекриття, що є важливим у випадках складних сцен та різноманітних об'єктів.

– Inference Time (IT) – Час інференції. Цей параметр визначає, скільки часу потрібно моделі для обробки одного зображення. Менше значення IT є критично важливим для застосувань у реальному часі, наприклад, у системах безпеки та автономного транспорту.

– Frames per Second (FPS) – Частота обробки кадрів за секунду. Обчислюється за формулою:

$$FPS = \frac{1000}{IT}$$

де IT виражено в мілісекундах.

Чим вища частота FPS, тим швидше модель може працювати в режимі реального часу.

Після підготовки середовища і вибору моделей було проведено етап дослідження, який включав навчання моделей, їхню валідацію та збір результатів для подальшого аналізу. Дослідження будувалося за принципами відтворюваності експериментів, що передбачає точне дотримання однакових умов під час роботи з різними архітектурами моделей і роздільними здатностями вхідних зображень.

Для роботи з моделлю YOLOv5 здійснили завантаження офіційного репозиторію YOLOv5, виконавши команду:

```
!git clone https://github.com/ultralytics/yolov5
```

Ця команда завантажує найновішу версію вихідного коду з GitHub [4], що забезпечує актуальність дослідження.

Перехід до робочої директорії:

```
%cd yolov5
```

Цей крок необхідний для коректної роботи зі скриптами, які знаходяться у відповідній директорії.

Встановлення всіх необхідних бібліотек:

```
!pip install -r requirements.txt
```

Ця команда виконує інсталяцію всіх залежностей, необхідних для коректного запуску моделі та її навчання, включаючи бібліотеки PyTorch, NumPy, Pandas та інші.

Завантаження датасету COCO128:

```
!bash data/scripts/get_coco128.sh
```

Ця команда дозволяє автоматично завантажити та розпакувати датасет COCO128, який застосовувався для навчання та тестування моделей [10].

Приклад команди для навчання моделі на роздільній здатності 256x256 пікселів:

```
!python train.py --img 256 --batch 16 --epochs 30 --data coco128.yaml --weights yolov5s.pt --name yolo_coco128_256
```

В цій команді використано наступні параметри:

- --img – роздільна здатність вхідних зображень (RS);
- --batch – розмір пакету, який був адаптований відповідно до доступної пам'яті GPU;
- --epochs – кількість епох (обрано 30 як компроміс між точністю і швидкістю навчання);
- --data – файл конфігурації, який містить інформацію про дата сет;
- --weights – попередньо навчені ваги моделі для прискорення навчання (використано yolov5s.pt як базову полегшену модель);
- --name – ім'я експерименту для зручної організації збережених результатів.

Команди з аналогічними параметрами використовувались для інших роздільних здатностей (320, 512, 640, 800, 1280 пікселів).

Для запуску та проведення валідації використовувався код, що представлений нижче:

```
!python val.py --weights runs/train/yolo_coco128_256/weights/best.pt --data coco128.yaml --img 256
```

Даний код використовує файл best.pt, що містить збережену модель, яка показала найкращі результати на валідаційних даних під час процесу навчання. Цей файл створюється автоматично і містить оптимальні значення ваг моделі, зафіксовані на епісі з найвищим значенням параметрів mAP@0.5 та mAP@0.5:0.95. Результати



валідації зберігалися для подальшого аналізу. Команди з аналогічними параметрами використовувались для інших роздільних здатностей (320, 512, 640, 800, 1280 пікселів).

З метою забезпечення коректного порівняння моделей було проведено аналогічні експерименти з використанням архітектури YOLOv8. Для роботи з моделлю YOLOv8 здійснили встановлення бібліотеки Ultralytics:

```
!pip install ultralytics
```

Бібліотека ultralytics забезпечує зручний інтерфейс для роботи з моделями YOLOv8 і включає всі необхідні інструменти для навчання, валідації та тестування моделей.

Завантаження датасету COCO128:

```
!curl -L https://github.com/ultralytics/yolov5/releases/download/v1.0/coco128.zip -o coco128.zip
```

```
!unzip -q coco128.zip -d ./datasets/
```

Хоча ці команди належать до офіційного репозиторію YOLOv5, структура датасету COCO128 є універсальною і повністю сумісною з YOLOv8.

Приклад команди для навчання моделі на роздільній здатності 256x256 пікселів:

```
from ultralytics import YOLO
```

```
model = YOLO('yolov8n.pt')
```

```
model.train(data='coco128.yaml', epochs=30, imgsz=256, batch=16, project='runs_v8', name='yolo_coco128_256')
```

В цій команді використано наступні параметри:

- data: Шлях до файлу з конфігурацією дата сету;
 - epochs: Кількість епох навчання (30 епох, аналогічно як і для YOLOv5);
 - imgsz: Роздільна здатність вхідних зображень;
 - batch: Розмір пакету обирається залежно від доступної пам'яті GPU;
 - project та name: Використовуються для організації результатів експериментів за окремими каталогами.
- Команди з аналогічними параметрами використовувались для інших роздільних здатностей (320, 512, 640, 800, 1280 пікселів).

Для запуску та проведення валідації використовувався код, що представлений нижче:

```
model = YOLO('runs_v8/yolo_coco128_256/weights/best.pt')
```

```
model.val(data='coco128.yaml', imgsz=256, batch=32)
```

Аналогічно як і у випадку для YOLOv5 був попередньо сформований та використаний файл best.pt. Команди з аналогічними параметрами використовувались для інших роздільних здатностей (320, 512, 640, 800, 1280 пікселів).

Результати дослідження було зібрано для всіх тестованих роздільних здатностей та моделей. Отримані дані включають значення PS, RC, mAP@0.5, mAP@0.5:0.95, IT та FPS, розрахований на основі IT.

Результати досліджень

Зведені результати експериментів для моделей YOLOv5 та YOLOv8 наведені в табл. 1 та табл. 2.

Таблиця 1 – Результати моделі YOLOv5

Роздільна здатність (RS)	Точність (PS)	Повнота (RC)	mAP@0.5	mAP@0.5:0.95	Час інференції (IT), мс	Частота кадрів (FPS)
256x256	0.768	0.628	0.693	0.46	5.2	192
320x320	0.823	0.722	0.774	0.531	4.4	227
512x512	0.895	0.782	0.868	0.648	6.3	159
640x640	0.874	0.868	0.903	0.68	11	91
800x800	0.858	0.861	0.918	0.671	17.3	58
1280x1280	0.87	0.744	0.856	0.577	30.3	33

Таблиця 2 – Результати моделі YOLOv8

Роздільна здатність (RS)	Точність (PS)	Повнота (RC)	mAP@0.5	mAP@0.5:0.95	Час інференції (IT), мс	Частота кадрів (FPS)
256x256	0.76	0.473	0.577	0.418	8.05	124
320x320	0.814	0.561	0.67	0.515	5.41	185
512x512	0.849	0.673	0.783	0.62	7.7	130
640x640	0.815	0.773	0.84	0.673	5.72	175
800x800	0.856	0.784	0.863	0.689	6.3	159
1280x1280	0.739	0.814	0.848	0.589	26.94	37

З метою візуального аналізу результатів та подальшого встановлення залежностей досліджуваних параметрів від роздільної здатності RS здійснена побудова графіків та апроксимація деяких з них.

Залежність параметрів PS, RC, mAP@0.5 та mAP@0.5:0.95 від RS дозволяє оцінити, як збільшення роздільної здатності впливає на точність моделей. Залежність часу IT від RS необхідна для аналізу продуктивності моделей



і визначення придатності для використання в режимі реального часу. Залежність FPS від RS дозволяє ілюструвати, наскільки збільшується обчислювальне навантаження при підвищенні роздільної здатності вхідних зображень. Ці залежності є стандартними в аналізі ефективності моделей комп'ютерного бачення, що підтверджено роботами [13], [14].

Для виявлення загальних закономірностей та побудови моделей прогнозування планується апроксимація графіків залежності $mAP@0.5$, $mAP@0.5:0.95$ та IT від RS. Параметри PS та RC апроксимувати недоцільно, оскільки їх зміни є менш плавними і часто мають нелінійний та нестабільний характер через вплив конкретних об'єктів у тестовому наборі, що підтверджується дослідженнями [15]. Параметр FPS також недоцільно апроксимувати, оскільки він повністю залежить від IT та є обернено пропорційним до нього.

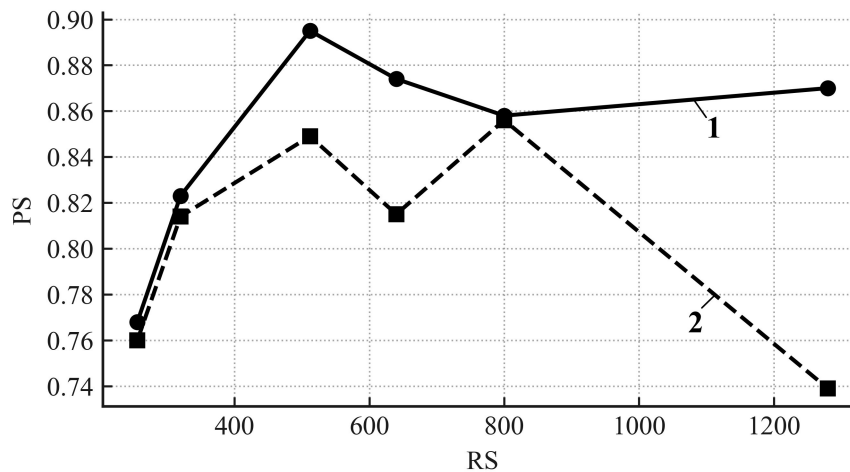
Аналіз наукової літератури показує, що для подібних досліджень найбільш ефективними є поліноміальна апроксимація третього порядку, що добре підходить для моделювання змін, характерних для $mAP(RS)$ [16]. Поліноміальна апроксимація вищого порядку виявилася також найбільш придатною для апроксимації часу інференції IT, оскільки ця характеристика демонструє складну нелінійну залежність від роздільної здатності вхідних зображень (RS). Така залежність обумовлена поєднанням факторів, серед яких архітектурні особливості нейронних мереж, оптимізації обчислювальних операцій та особливості використання апаратних ресурсів. Відповідно до рекомендацій, викладених у роботі [17], для опису подібних нелінійних процесів доцільно використовувати поліноми третього порядку.

Формально апроксимації $mAP(RS)$ та $IT(RS)$ описуються наступними формулами:

$$mAP(RS) = a \cdot RS^3 + b \cdot RS^2 + c \cdot RS + d$$

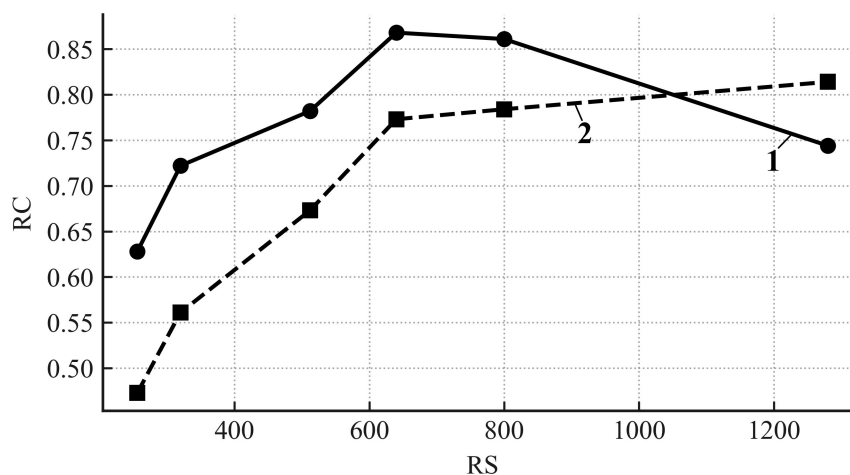
$$IT(RS) = a \cdot RS^3 + b \cdot RS^2 + c \cdot RS + d$$

де a, b, c, d – коефіцієнти апроксимації, визначені методом найменших квадратів.



1 – YOLOv5; 2 – YOLOv8

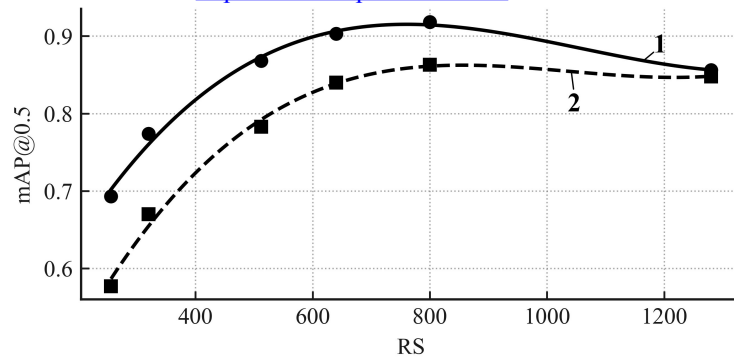
Рис. 1 – Графік залежності PS від RS
Fig. 1 – Graph of PS dependency on RS



1 – YOLOv5; 2 – YOLOv8

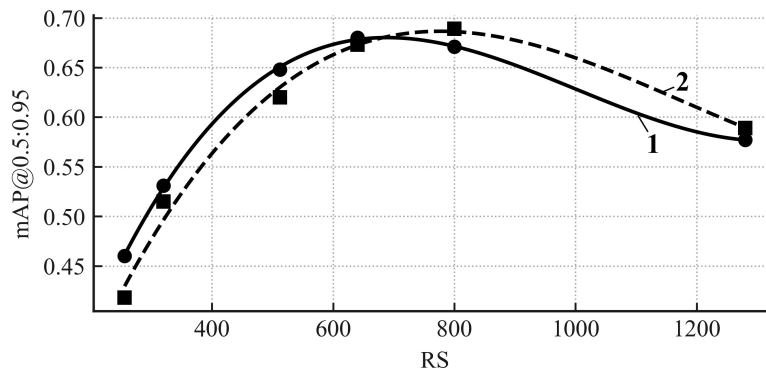
Рис. 2 – Графік залежності RC від RS
Fig. 2 – Graph of RC dependency on RS

Зазначені методи дозволяють отримати точні моделі прогнозування, що широко застосовується в сучасних дослідженнях комп'ютерного бачення.



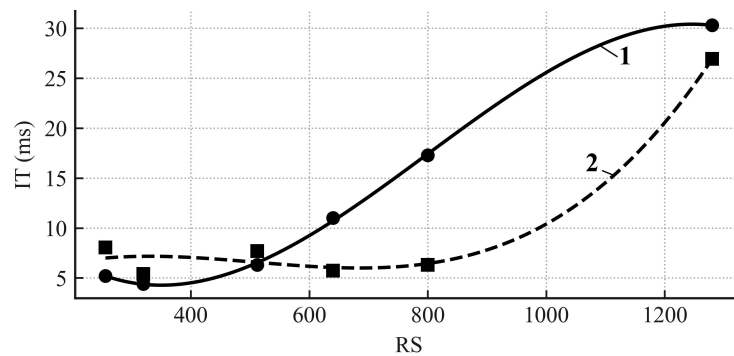
1 – YOLOv5 апрокс. (approx.); 2 – YOLOv8 апрокс. (approx.)

Рис. 3 – Графік залежності mAP@0.5 від RS
Fig. 3 – Graph of mAP@0.5 dependency on RS



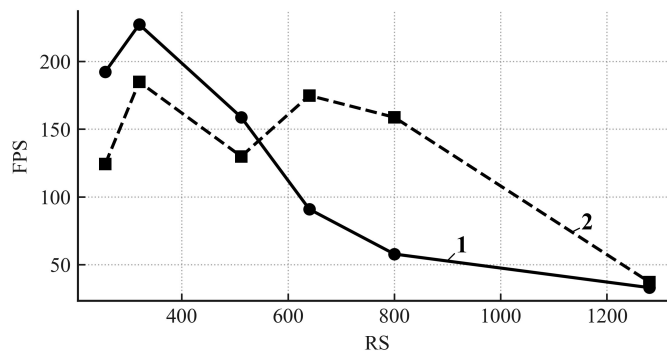
1 – YOLOv5 апрокс. (approx.); 2 – YOLOv8 апрокс. (approx.)

Рис. 4 – Графік залежності mAP@0.5:0.95 від RS
Fig. 4 – Graph of mAP@0.5:0.95 dependency on RS



1 – YOLOv5 апрокс. (approx.); 2 – YOLOv8 апрокс. (approx.)

Рис. 5 – Графік залежності IT від RS
Fig. 5 – Graph of IT dependency on RS



1 – YOLOv5; 2 – YOLOv8

Рис. 6 – Графік залежності FPS від RS
Fig. 6 – Graph of FPS dependency on RS



Обговорення результатів

У процесі дослідження було проаналізовано залежність ключових метрик якості детекції та продуктивності моделей YOLOv5 і YOLOv8 від роздільної здатності вхідних зображень. Отримані результати дозволяють виявити чіткі закономірності та особливості поведінки обох моделей в контексті різних практичних застосувань.

З аналізу графіка залежності точності (PS) від роздільної здатності (RS) можна зробити висновок, що для обох моделей спостерігається тенденція покращення точності зі збільшенням роздільної здатності вхідних даних. Проте характер цього зростання має певні особливості. Для YOLOv5 точність зростає більш плавно і стабільно, досягаючи свого піку при $RS = 512$ пікселів, після чого спостерігається незначне зниження або стабілізація результатів. Для YOLOv8 точність також демонструє покращення з підвищенням RS, але має більш виражені локальні коливання, що може бути пов'язано з особливостями її архітектури та обробки вхідних даних. Це свідчить про те, що при виборі YOLOv8 слід уважніше підходити до підбору оптимальної роздільної здатності, оскільки надмірне її збільшення не завжди гарантує покращення результатів.

Залежність повноти (RC) від RS показує аналогічну картину. Зростання повноти спостерігається для обох моделей, однак YOLOv8 демонструє більш стрімке зростання цього показника при високих значеннях RS. Це вказує на її кращу здатність виявляти об'єкти при високій деталізації зображення. Для YOLOv5 же спостерігається досягнення плато при RS понад 512 пікселів, що робить цю модель оптимальною для застосувань, де важливо забезпечити прийнятну повноту при помірних ресурсних витратах.

Аналіз показників середньої точності ($mAP@0.5$ та $mAP@0.5:0.95$) підтверджує виявлені раніше закономірності. Значення $mAP@0.5$ для YOLOv5 стабілізується на високому рівні вже при $RS = 512$, тоді як для YOLOv8 найвищі результати досягаються при $RS = 800$ пікселів. При цьому, графіки апроксимації показують, що подальше збільшення RS не призводить до значного покращення метрик, що дозволяє стверджувати про існування оптимальної межі роздільної здатності для кожної моделі. Це особливо важливо в контексті систем реального часу, де ресурси обмежені.

Час інференції (IT) показав найбільш виражену залежність від RS. Для YOLOv5 спостерігається стрімке зростання IT зі збільшенням RS, що повністю відповідає очікуванням і підтверджує обмеження цієї моделі в задачах з високими вимогами до швидкодії. YOLOv8, хоча й демонструє загальну тенденцію до зростання часу інференції, має менш виражену залежність, що свідчить про ефективніше використання апаратних ресурсів і більш глибоку оптимізацію обчислювальних процесів. Побудовані апроксимаційні криві дозволяють прогнозувати час інференції для будь-якого значення RS в межах досліджуваного діапазону, що є надзвичайно корисним для інженерних розрахунків під час проєктування систем.

Важливо також зазначити, що аналіз частоти обробки кадрів (FPS) вказує на швидке падіння продуктивності при збільшенні роздільної здатності. Для YOLOv5 це падіння є більш різким, що обмежує її використання в системах відеоспостереження та автономних роботизованих платформах, де важлива висока частота оновлення кадрів. Натомість YOLOv8 продемонструвала кращу продуктивність при високих роздільних здатностях, що робить її більш привабливою для застосувань у сфері інтелектуальних транспортних систем і медіа-аналітики, де необхідна висока точність виявлення об'єктів при збереженні прийнятної швидкодії.

Таким чином, отримані результати дослідження дозволяють обґрунтовано рекомендувати оптимальні параметри використання моделей YOLOv5 і YOLOv8 залежно від специфіки задач. Для систем з обмеженими обчислювальними ресурсами або вимогами до високої продуктивності доцільно застосовувати YOLOv5 при RS не вище 512 пікселів. Для більш вимогливих до точності систем, де важлива повнота виявлення об'єктів і допускається зменшення FPS, оптимальним вибором буде YOLOv8 із роздільною здатністю в межах 640–800 пікселів. Проведене апроксимаційне моделювання дозволяє прогнозувати ефективність моделей при зміні роздільної здатності, що значно спрощує процес прийняття рішень щодо вибору відповідної конфігурації систем комп'ютерного бачення.

Висновки

У межах проведеного дослідження було реалізовано комплексний аналіз впливу роздільної здатності вхідних зображень на якість виявлення об'єктів і продуктивність сучасних моделей глибокого навчання YOLOv5 та YOLOv8. Експериментальне порівняння засвідчило наявність виразної залежності між розміром вхідних зображень і основними метриками точності, повноти, середньої точності та швидкодії моделей.

Дослідження підтвердило, що збільшення роздільної здатності вхідних даних здатне покращити показники якості виявлення об'єктів, однак це супроводжується значним збільшенням часу інференції та зменшенням частоти обробки кадрів. Це підкреслює важливість пошуку компромісу між точністю та швидкістю моделей у практичних застосуваннях. Проведене апроксимаційне моделювання дозволило отримати аналітичні залежності, які можна використовувати для прогнозування продуктивності систем комп'ютерного бачення в залежності від обраної роздільної здатності.

Отримані результати мають прикладне значення для проєктування ефективних систем виявлення об'єктів у реальному часі, зокрема в галузях відеоспостереження, автономних роботизованих систем і інтелектуальних транспортних технологій. Подальші дослідження плануються зосередити на вивченні енергоспоживання моделей та їхньої адаптації до роботи на пристроях із обмеженими обчислювальними ресурсами.

**Список використаних джерел**

1. Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, "Object Detection with Deep Learning: A Review," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3212–3232, 2019.
2. J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *arXiv preprint arXiv:1804.02767*, 2018.
3. A. Bochkovskiy, C. Wang, and H. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," *arXiv preprint arXiv:2004.10934*, 2020.
4. G. Jocher et al., "YOLOv5 by Ultralytics," *GitHub Repository*, 2020. [Online]. Available: <https://github.com/ultralytics/yolov5>
5. Ultralytics, "YOLOv8 by Ultralytics," *GitHub Repository*, 2023. [Online]. Available: <https://github.com/ultralytics/ultralytics>
6. M. Tan and Q. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019, pp. 6105–6114.
7. A. Howard et al., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," *arXiv preprint arXiv:1704.04861*, 2017.
8. Google Colab, "Colaboratory by Google," [Online]. Available: <https://colab.research.google.com/>
9. Ultralytics, "YOLOv9 and Beyond: Experimental Developments," *Ultralytics Blog*, 2024.
10. COCO Dataset, "Common Objects in Context (COCO)," [Online]. Available: <https://cocodataset.org/>
11. T.-Y. Lin et al., "Microsoft COCO: Common Objects in Context," in *Proc. European Conf. on Computer Vision (ECCV)*, 2014, pp. 740–755.
12. K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.
13. H. Wang et al., "Benchmarking Deep Learning Models for Object Detection," in *IEEE Access*, vol. 8, pp. 110748–110760, 2020.
14. J. Huang et al., "Speed/Accuracy Trade-Offs for Modern Convolutional Object Detectors," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 7310–7311.
15. D. Lin et al., "Focal Loss for Dense Object Detection," in *Proc. IEEE ICCV*, 2017, pp. 2980–2988.
16. P. J. Rousseeuw and A. M. Leroy, *Robust Regression and Outlier Detection*, John Wiley & Sons, 2005.
17. H. H. Huang and Q. He, "Nonlinear Regression Analysis," *International Encyclopedia of Education*, 4th Edition, Elsevier, 2024.

References

1. Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, "Object Detection with Deep Learning: A Review," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3212–3232, 2019.
2. J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *arXiv preprint arXiv:1804.02767*, 2018.
3. A. Bochkovskiy, C. Wang, and H. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," *arXiv preprint arXiv:2004.10934*, 2020.
4. G. Jocher et al., "YOLOv5 by Ultralytics," *GitHub Repository*, 2020. [Online]. Available: <https://github.com/ultralytics/yolov5>
5. Ultralytics, "YOLOv8 by Ultralytics," *GitHub Repository*, 2023. [Online]. Available: <https://github.com/ultralytics/ultralytics>
6. M. Tan and Q. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019, pp. 6105–6114.
7. A. Howard et al., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," *arXiv preprint arXiv:1704.04861*, 2017.
8. Google Colab, "Colaboratory by Google," [Online]. Available: <https://colab.research.google.com/>
9. Ultralytics, "YOLOv9 and Beyond: Experimental Developments," *Ultralytics Blog*, 2024.
10. COCO Dataset, "Common Objects in Context (COCO)," [Online]. Available: <https://cocodataset.org/>
11. T.-Y. Lin et al., "Microsoft COCO: Common Objects in Context," in *Proc. European Conf. on Computer Vision (ECCV)*, 2014, pp. 740–755.
12. K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.
13. H. Wang et al., "Benchmarking Deep Learning Models for Object Detection," in *IEEE Access*, vol. 8, pp. 110748–110760, 2020.
14. J. Huang et al., "Speed/Accuracy Trade-Offs for Modern Convolutional Object Detectors," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 7310–7311.
15. D. Lin et al., "Focal Loss for Dense Object Detection," in *Proc. IEEE ICCV*, 2017, pp. 2980–2988.
16. P. J. Rousseeuw and A. M. Leroy, *Robust Regression and Outlier Detection*, John Wiley & Sons, 2005.
17. H. H. Huang and Q. He, "Nonlinear Regression Analysis," *International Encyclopedia of Education*, 4th Edition, Elsevier, 2024.

Отримана в редакції 20.05.2025. Прийнята до друку 10.06.2025. Received 21 May 2025. Approved 10 June 2025. Available in Internet 30 June 2025