

# **Тренування моделі YOLO для локалізації номерних знаків на зображеннях автомобілів**

## **Постановка задачі:**

- 1) Профільтрувати вибірку (можна використовуючи рішення з першого дз);
- 2) Проаугментувати;
- 3) Натренувати мережу для локалізації номерних знаків на зображенні;
- 4) За допомогою натренованої моделі YOLO (якщо результати хороші) + алгоритми опенсв виділити літери на номерних знаках та промаркувати використовуючи рішення з першого дз.

## Етап 1: Дослідження та фільтрування датасету

Датасет, взятий із Kaggle (<https://universe.roboflow.com/augmented-startups/vehicle-registration-plates-trudk/dataset/1>), містить 8823 зображення транспортних засобів із номерними знаками (рис. 1).

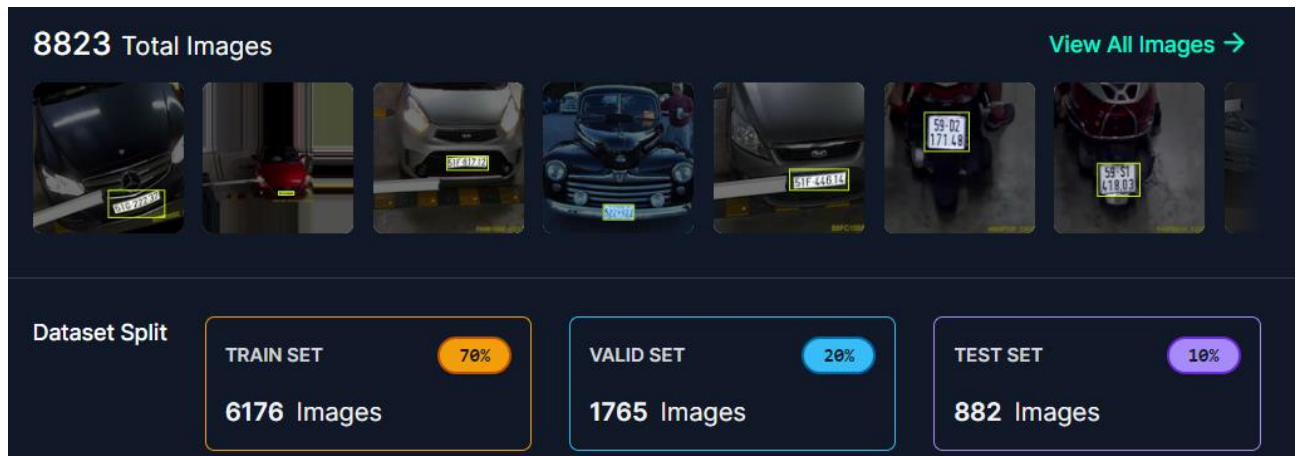


Рис. 1. Інформація про датасет Vehicle Registration Plates Image Dataset

У датасеті наведені зображення різних транспортних засобів (рис. 2), не лише автомобілів.



Рис. 2. Наявність різних транспортних засобів із номерними знаками в досліджуваному датасеті

Назви файлів із зображеннями автомобілів починаються з "CarLongPlate" і цю умову було використано для фільтрування досліджуваного датасету з метою отримати лише зображення автомобілів із номерними знаками (рис. 3), в результаті чого кількість зображень в тренувальному датасеті зменшилася від 6176 (рис. 1) до 3436 (рис. 3). Аналогічну операцію виконано для валідаційних (рис. 4) і тестових даних (рис. 5).

```
train_car_long_plate_images = [image_name for image_name in train_images if image_name.startswith('CarLongPlate')]
train_car_long_plate_labels = [label for label in train_labels if label.startswith('CarLongPlate')]

print(f"len(train_car_long_plate_images) = {len(train_car_long_plate_images)}")
print(f"len(train_car_long_plate_labels) = {len(train_car_long_plate_labels)}")

len(train_car_long_plate_images) = 3436
len(train_car_long_plate_labels) = 3436
```

Рис. 3. Результати відбору зображень із автомобілями тренувального датасету

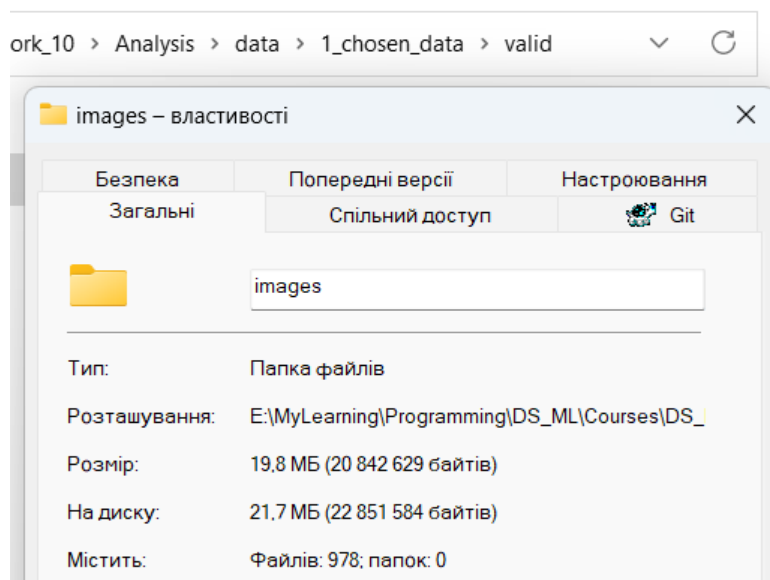


Рис. 4. Результати відбору зображень із автомобілями валідаційного датасету

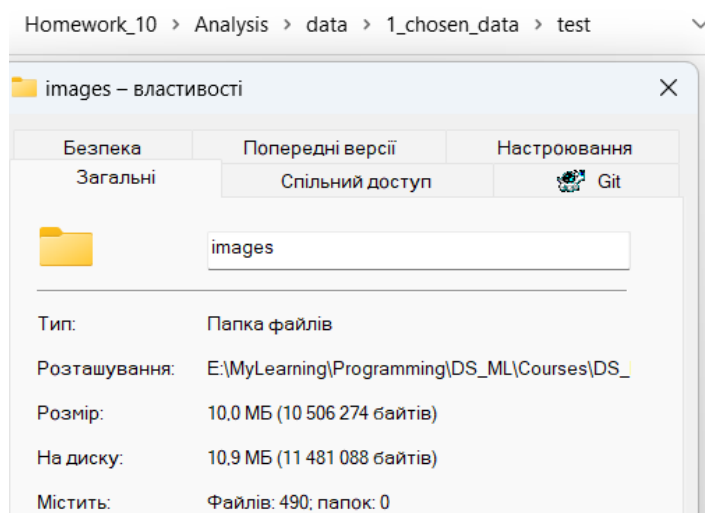


Рис. 5. Результати відбору зображень із автомобілями тестового датасету

Варто зазначити, що серед відібраних зображень наявні "биті" зображення (рис. 6). Такі зображення було вилучено вручну із тренувального, валідаційного і тестового датасетів, що призвело до зменшення їх обсягів (рис. 7), однак співвідношення 70/20/10 для тренувальної/валідаційної/тестової вибірок збереглося.



Рис. 6. "Биті" зображення у досліджуваному датасеті

Homework_10 > Analysis > data > 1_selected_data > train	Homework_10 > Analysis > data > 1_selected_data > valid	Homework_10 > Analysis > data > 1_selected_data > test
<div>images – властивості</div> <div>Безпека Загальні</div> <div>Попередні версії Спільний доступ</div> <div>Налаштування Git</div> <div>images</div> <div>Тип: Папка файлів</div> <div>Розташування: E:\MyLearning\Programming\DS_ML\Courses\DS_</div> <div>Розмір: 58.7 МБ (61 611 301 байтів)</div> <div>На диску: 64.0 МБ (67 129 344 байтів)</div> <div>Містить: Файлів: 2 715; папок: 0</div>	<div>images – властивості</div> <div>Безпека Загальні</div> <div>Попередні версії Спільний доступ</div> <div>Налаштування Git</div> <div>images</div> <div>Тип: Папка файлів</div> <div>Розташування: E:\MyLearning\Programming\DS_ML\Courses\DS_</div> <div>Розмір: 17.0 МБ (17 881 040 байтів)</div> <div>На диску: 18.6 МБ (19 521 536 байтів)</div> <div>Містить: Файлів: 796; папок: 0</div>	<div>images – властивості</div> <div>Безпека Загальні</div> <div>Попередні версії Спільний доступ</div> <div>Налаштування Git</div> <div>images</div> <div>Тип: Папка файлів</div> <div>Розташування: E:\MyLearning\Programming\DS_ML\Courses\DS_</div> <div>Розмір: 8.50 МБ (8 919 216 байтів)</div> <div>На диску: 9.25 МБ (9 707 520 байтів)</div> <div>Містить: Файлів: 394; папок: 0</div>

Рис. 7. Результати вилучення "битих" зображень із досліджуваного датасету

Для подальшого фільтрування зображень було використано алгоритм з попереднього домашнього завдання, який було вдосконалено в ході фільтрування:

- 1) Було додано перевірку на кількість розпізнаних за допомогою easyOCR символів номерного знаку: якщо кількість символів становила 3 і більше, то локалізована частина зображення автомобіля вважалася номерним знаком.
- 2) В алгоритмі, який використовує функції бібліотеки opencv було також додано перевірку співвідношення сторін локалізованого чотирикутника, що дозволило швидше (ще до моменту розпізнавання символів на локалізованій частині зображення) відкидати помилкові результати (вікна, колеса, двері і т.д.). Зокрема, на рис. 8 зображено результат роботи алгоритму з мінімальним відношенням сторін 1.5 (було виявлено лише частину номерного знака), а на рис. 9 – мінімальне співвідношення збільшено до 2. Можливість змінювати мінімальне та максимальне співвідношення між сторонами визначеного чотирикутника забезпечує більшу гнучкість та надійність алгоритму.



```
# OpenCV algorithm
location = opencv_license_plates_localization(file_path=image_path, min_aspect_ratio=1.5)
Image_path = data/example_data/images\CarLongPlateGen3256_jpg.rf.f7a39a9f20f8e685b7e1f100d9249c90.jpg
out.shape : (507, 6)
out.shape : (2028, 6)
out.shape : (8112, 6)
```

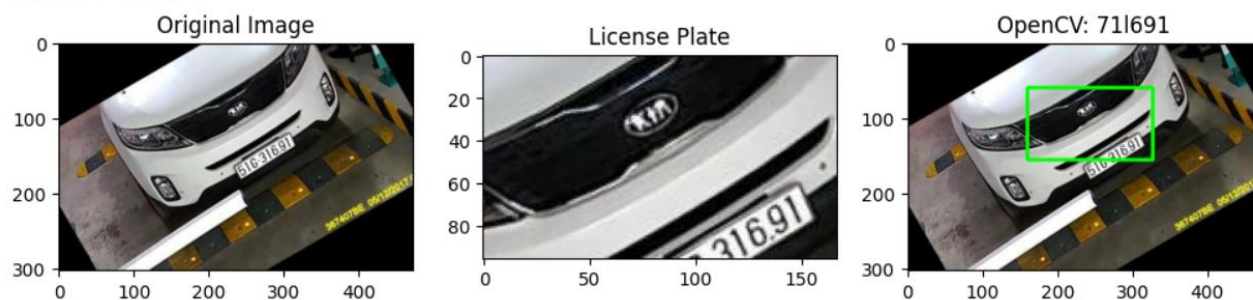


Рис. 8. Результати локалізації номерного знаку із  $\text{min\_aspect\_ratio}=1.5$

```
# OpenCV algorithm
location = opencv_license_plates_localization(file_path=image_path, min_aspect_ratio=2)
```

NO LICENSE PLATE DETECTED IN THIS IMAGE: CarLongPlateGen3256\_jpg.rf.f7a39a9f20f8e685b7e1f100d9249c90.jpg!!!

CarLongPlateGen3256\_jpg.rf.f7a39a9f20f8e685b7e1f100d9249c90.jpg



Рис. 9. Результати локалізації номерного знаку із  $\text{min\_aspect\_ratio}=2$

- 3) В ході тестування алгоритму було виявлено велику кількість зображень, розміщених під певним кутом до горизонталі (rotation-аугментація), тому для полегшення читабельності номерних знаків для easyOCR було використано афінні перетворення для вирівнювання локалізованого номерного знаку. Результати розпізнавання символів за допомогою easyOCR на оригінальних номерних знаках зображено на рис. 10, а на вирівняних – на рис. 11.

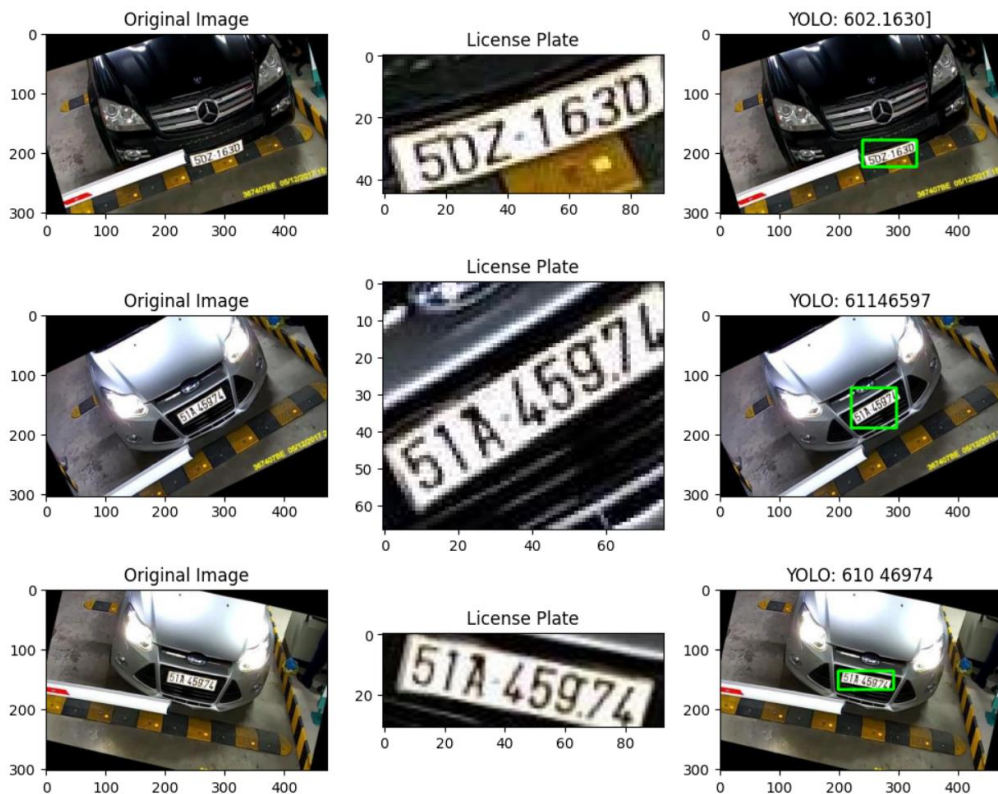


Рис. 10. Результати розпізнавання символів на локалізованих оригінальних номерних знаках за допомогою easyOCR

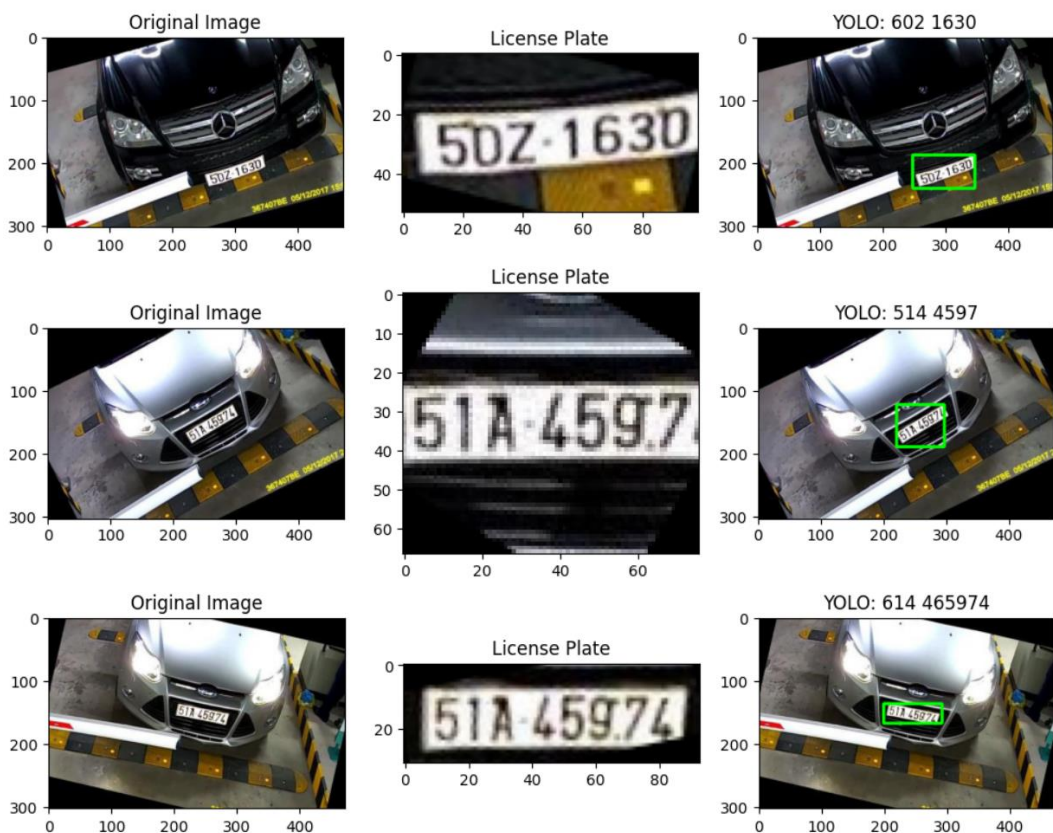


Рис. 11. Результати розпізнавання символів на локалізованих вирівняних номерних знаках за допомогою easyOCR

Отже, вдосконалений алгоритм з домашнього завдання №9 працює так:

*Крок 1)* Спочатку модель YOLO намагається виділити номерний знак на зображенні автомобіля. Якщо їй це не вдається → Крок 2, інакше -> Крок 3.

*Крок 2)* Для виділення номерного знаку використовується алгоритм з методами openCV опрацювання зображень. Цей алгоритм передбачає виконання таких кроків:

- 1) Перетворення зображення на градації сірого (grayscale);
- 2) Використання двостороннього фільтру (bilateralFilter) для зменшення шумів (Noise reduction);
- 3) Використання оператора Кенні для виділення контурів;
- 4) Знаходження контурів номерного знаку (у вигляді чотирикутника);
- 5) Перевірка відношення сторін виділеного чотирикутника.

Якщо номерний знак було виділено вдало → Крок 3, інакше -> Крок 5.

*Крок 3)* Використання афінних перетворень для вирівнювання зображення номерного знаку з метою полегшення розпізнавання символів моделлю easyOCR;

*Крок 4)* Для перевірки того, що виділена область і справді є номерним знаком, використовується модель easyOCR. Якщо easyOCR змогла розпізнати текст -> Крок 7, інакше -> Крок 6.

*Крок 5)* Виведення повідомлення про те, що не вдалося виділити номерний знак та відображення оригінального зображення автомобіля. Вихід із алгоритму.

*Крок 6)* Виведення повідомлення про те, що на виділеній області не вдалося розпізнати текст або довжина тексту є замалою та відображення оригінального зображення автомобіля. Вихід із алгоритму.

*Крок 7)* Зберігання даних про виділений номерний знак та відображення результатів виділення на оригінальному зображенні. Вихід із алгоритму.

Отриманий алгоритм було застосовано для фільтрування зображень досліджуваного датасету. Алгоритм показав доволі хороші результати, оскільки дозволив вилучити зображення:

- 1) На яких неможливо було повноцінно локалізувати та прочитати номерний знак (рис. 12)
- 2) Які мали погану якість (рис.13);
- 3) Номерний знак було видно не повністю (рис. 14);
- 4) "Самозванців" (зображення мотоциклів, промаркованих як автомобілі) (рис. 15).

NO LICENSE PLATE DETECTED IN THIS IMAGE: CarLongPlateGen2835\_jpg.rf.f547f7d9a0aba1b2f49c20fe94fe71ad.jpg!!!

CarLongPlateGen2835\_jpg.rf.f547f7d9a0aba1b2f49c20fe94fe71ad.jpg



NO LICENSE PLATE DETECTED IN THIS IMAGE: CarLongPlateGen2836\_jpg.rf.c1b05ee6dd1f01b37a41544235d10fd2.jpg!!!

CarLongPlateGen2836\_jpg.rf.c1b05ee6dd1f01b37a41544235d10fd2.jpg



Рис. 12. Відкинуті алгоритмом зображення низької якості, на яких номерні знаки зображено не повністю



NO LICENSE PLATE DETECTED IN THIS IMAGE: CarLongPlate103\_jpg.rf.344c8ec636c3fac0667537ff778ffa3a.jpg!!!



NO LICENSE PLATE DETECTED IN THIS IMAGE: CarLongPlate526\_jpg.rf.deeec64a2dec3f23dcdd3ad25a4a9340.jpg!!!



Рис. 13. Відкинуті алгоритмом зображення низької якості та зображення, на яких номерні знаки зображено не повністю

NO TEXT WAS DETECTED ON THE SELECTED PART OF THE IMAGE: CarLongPlateGen2718\_jpg.rf.4e63e1cb4064d68d59179e640ad77327.jpg!!!

CarLongPlateGen2718\_jpg.rf.4e63e1cb4064d68d59179e640ad77327.jpg



NO TEXT WAS DETECTED ON THE SELECTED PART OF THE IMAGE: CarLongPlateGen2870\_jpg.rf.010fa139fd87abea438de3077d792d3a.jpg!!!

CarLongPlateGen2870\_jpg.rf.010fa139fd87abea438de3077d792d3a.jpg



Рис. 14. Відкинуті алгоритмом зображення, на яких номерні знаки зображено не повністю

NO LICENSE PLATE DETECTED IN THIS IMAGE: CarLongPlate843\_jpg.rf.bfb7e9abc5a3f406ffa33de88f5d1e.jpg!!!

CarLongPlate843\_jpg.rf.bfb7e9abc5a3f406ffa33de88f5d1e.jpg



NO LICENSE PLATE DETECTED IN THIS IMAGE: CarLongPlate84\_jpg.rf.04be65e3e78229d1649639020b0cdeed.jpg!!!

CarLongPlate84\_jpg.rf.04be65e3e78229d1649639020b0cdeed.jpg



Рис. 15. Відкинуті алгоритмом "зображення-самозванці"

В результаті використання мого алгоритму для фільтрування зображень, було вилучено приблизно 20% зображень як в тренувальному дата сеті, так і в валідаційному та тестовому (рис. 16-18).

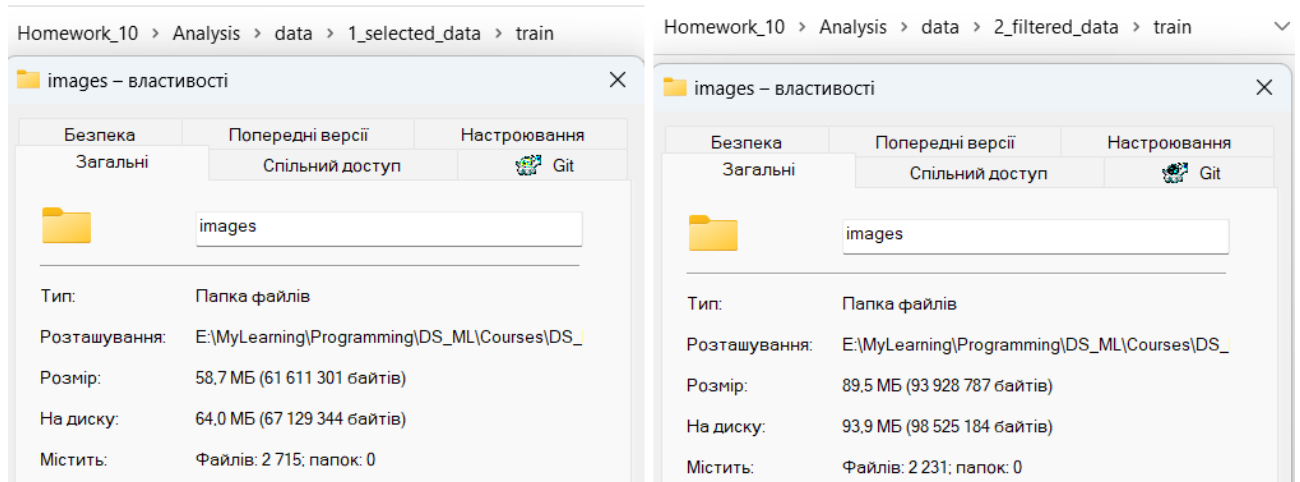


Рис. 16. Результати відсіювання зображень тренувального датасету за допомогою вдосконаленого алгоритму з дз 9 (зліва – кількість файлів до фільтрування, справа – після фільтрування)

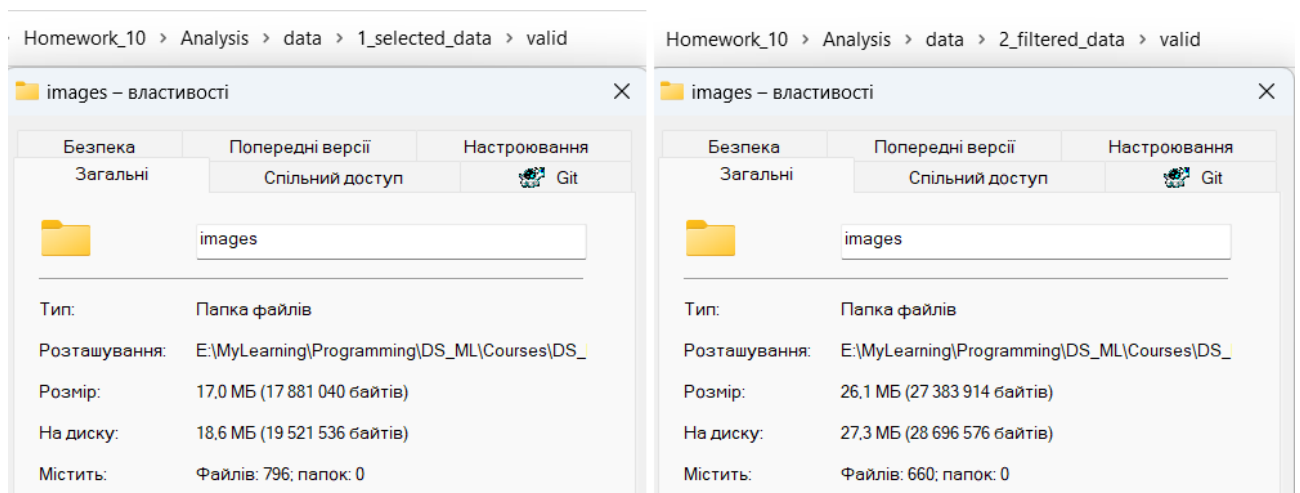


Рис. 17. Результати відсіювання зображень валідаційного датасету за допомогою вдосконаленого алгоритму з дз 9 (зліва – кількість файлів до фільтрування, справа – після фільтрування)

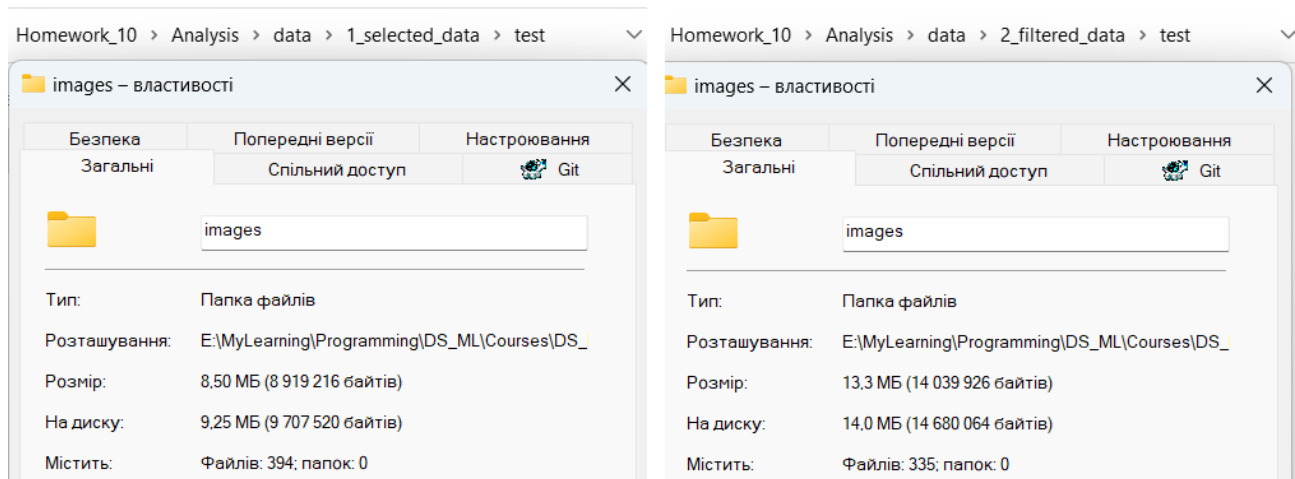


Рис. 18. Результати відсіювання зображень тестового датасету за допомогою вдосконаленого алгоритму з дз 9 (зліва – кількість файлів до фільтрування, справа – після фільтрування)

## Етап 2: Аугментація

Під час фільтрування датасету було виявлено, що він уже містить зображення, отримані внаслідок застосування до оригінальних зображень різних технік аугментації (такі файли містять суфікс "Gen" у своїй назві), зокрема:

- 1) Brightness and Contrast Adjustment (налаштування яскравості та контрастності) – рис. 19;
  - 2) Noise Addition (додавання шуму) – рис. 20;
  - 3) Data Augmentation with Backgrounds (доповнення даних за допомогою фону) – рис. 21;
  - 4) Rotation (повороти) – рис. 22;
- І т. д.



Рис. 19. Brightness and Contrast Adjustment Augmentation





Рис. 20. Noise Addition Augmentation



Рис. 21. Data Augmentation with Backgrounds



Рис. 22. Rotation Augmentation

Тому було прийнято рішення тренувати модель YOLO саме на отриманому датасеті, без використання власних технік аугментації.

### Етап 3: Тренування моделі YOLOv8

Навчання моделі проводилося в Google Colab з метою пришвидшення процесу тренування. Обсяг тренувального датасету становив 2231 зображення, валідаційного – 660, тестувального – 335 (рис. 23).

```
len(train_images) = 2231
len(train_labels) = 2231
len(valid_images) = 660
len(valid_labels) = 660
len(test_images) = 335
len(test_labels) = 335
```

Рис. 23. Дані для тренування моделі YOLOv8

Модел ь YOLOv8n тренувалася протягом 5 епох. Результати тренування наведено на рис. 24, а на рис. 25 зображено Precision-Recall Curve натренованої моделі.

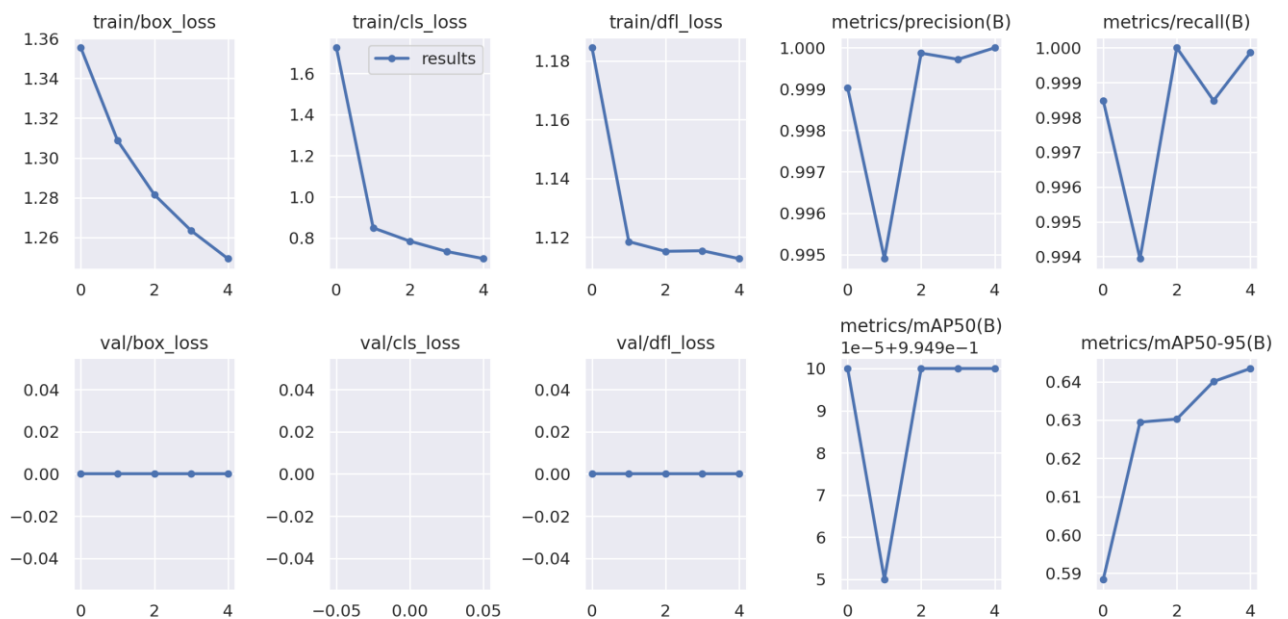


Рис. 24. Результати тренування моделі YOLOv8

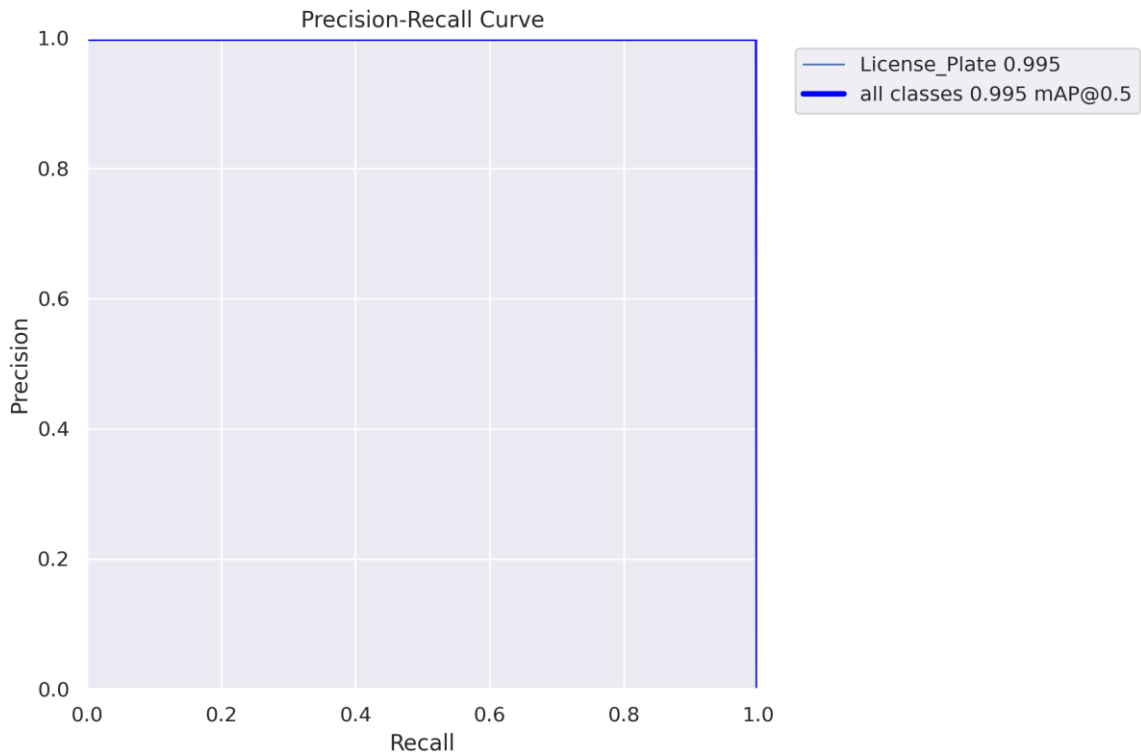


Рис. 25. Precision-Recall Curve моделі YOLOv8

Натреновану модель було протестовано на тренувальних зображеннях і вона показала хороші результати, правильно локалізуючи номерні знаки на всіх зображеннях (рис. 26).



Рис. 26. Використання натренованої моделі YOLOv8 для локалізації номерних знаків на тестових зображеннях

Натреновану модель було протестовано ще і на відеоролику з автомобілями, де вона також показала хороші результати.

Для порівняння з оригінальними labels (рис. 27), було отримано результати моделі у форматі xwhn (рис. 28) і наведено результати для перших 5 тестових зображень.



labels\_df

	image_name	x	y	width	height
0	CarLongPlate68_jpg.rf.1d7a7c63697a0a3cb182d1aa...	0.647246	0.782178	0.175847	0.066007
1	CarLongPlate23_jpg.rf.21d4b3887143f1de62137cc0...	0.622881	0.801980	0.237288	0.092409
2	CarLongPlate83_jpg.rf.241f3e468e783f0a5e6f1db2...	0.533898	0.773927	0.228814	0.089109
3	CarLongPlate24_jpg.rf.d7388a8d0f429ccfa474f94a...	0.384534	0.486799	0.239407	0.082508
4	CarLongPlate50_jpg.rf.33cfec44292ef250e66c8ce7...	0.406780	0.678218	0.262712	0.108911

Рис. 27. Оригінальні labels



predicted\_df

	image_name	x	y	width	height
0	CarLongPlate68_jpg.rf.1d7a7c63697a0a3cb182d1aa...	0.647637	0.790929	0.174770	0.081907
1	CarLongPlate23_jpg.rf.21d4b3887143f1de62137cc0...	0.381430	0.483228	0.240576	0.110060
2	CarLongPlate83_jpg.rf.241f3e468e783f0a5e6f1db2...	0.531956	0.772811	0.204393	0.076726
3	CarLongPlate24_jpg.rf.d7388a8d0f429ccfa474f94a...	0.407195	0.678014	0.250246	0.107725
4	CarLongPlate50_jpg.rf.33cfec44292ef250e66c8ce7...	0.621668	0.799889	0.227422	0.112242

Рис. 28. Predicted labels



## Висновки

Було вдосконалено алгоритм локалізації номерних знаків з домашнього завдання № 9 за рахунок додавання перевірки співвідношення сторін виділеного чотирикутника в алгоритмі openCV, здійснення афінних перетворень для вирівнювання локалізованого номерного знака. Згадані модернізації дозволили значно вдосконалити алгоритм: замість відсіювання 50% зображень (як це було у дз №9), цей показник було зведено до 20%.

На відфільтрованих зображеннях було натреновано модель YOLOv8n, яка показала доволі хороші результати на тестових зображеннях (як оригінальних, так і аугментованих) і навіть правильно локалізувала номерні знаки на тестовому відеоролику.