

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему Метод виявлення та розпізнавання номерів автомобілів
нейромережевими засобами

Галузь знань 12 – Інформаційні технології
Шифр і назва галузі знань
Спеціальність 122 – Комп'ютерні науки
Шифр і назва спеціальності
Освітня програма Комп'ютерні науки
Назва освітньої програми

Виконав: студент 2 курсу, група КНМ-23-2
Курс, група виконавця
Керівник: д.т.н., професор кафедри КН
Науковий ступінь, посада
Нормоконтроль: к.т.н., доцент кафедри КН
Науковий ступінь, посада

Костянтин ЯВОРСЬКИЙ
Ініціали, прізвище

Едуард МАНЗЮК
Ініціали, прізвище

Руслан БАГРІЙ
Ініціали, прізвище

До захисту допускаю:


Зав. кафедри КН, д.т.н., професор

Олександр БАРМАК
Ініціали, прізвище

10 грудня 2024 р.

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
Факультет інформаційних технологій
Кафедра комп'ютерних наук
Освітній ступінь магістр
Галузь знань 12 – Інформаційні технології
Спеціальність 122 – Комп'ютерні науки

ЗАТВЕРДЖУЮ
Завідувач кафедри комп'ютерних наук


(підпис)
д.т.н., професор Олександр БАРМАК
« 02 » вересня 2024 року

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ МАГІСТРА

1. Тема кваліфікаційної роботи магістра: «Метод виявлення та розпізнавання номерів автомобілів нейромережевими засобами»

2. Завдання видано студенту Яворському Костянтину Андрійовичу
(прізвище, ім'я, по батькові)

3. Керівник роботи д.т.н., професор кафедри КН Едуард МАНЗЮК
(прізвище, ім'я, по батькові)

4. Затверджені наказом університету від « 26 » 08 2024 р. № 60.

5. Зміст пояснювальної записки (перелік задач) та вихідні дані:

Метод виявлення та розпізнавання номерів автомобілів нейромережевими засобами. Метод ґрунтується на зображеннях отриманих з камери, а ідентифікація об'єктів здійснюється за допомогою штучних нейронних мереж та оптичного розпізнавання символів.

Необхідно виконати такі завдання дослідження:

– розробити метод автоматизованого виявлення та розпізнавання номерних знаків на основі модифікованої архітектури YOLOv11 з інтеграцією механізмів просторової уваги та оптичного розпізнавання символів;

– розробити інформаційну систему для виявлення та розпізнавання номерних знаків автомобілів з використанням розробленого методу;

– провести експериментальне дослідження ефективності розробленого методу шляхом оцінки точності виявлення та розпізнавання номерних знаків за різних умов роботи системи.

Реферат

Кваліфікаційна робота магістра присвячена дослідженню науково-технічної задачі автоматизованого знаходження та розпізнавання номерів автомобілів за допомогою нейронних мереж. Результатом роботи є метод, який дозволяє автоматично виявляти та розпізнавати номери автомобілів на зображеннях. Вхідні дані у вигляді зображень транспортних засобів обробляються навченою нейромережею, яка здатна точно визначати положення номерних знаків та розпізнавати їх символи. Вихідна інформація представлена у вигляді прямокутників навколо знайдених номерних знаків, з виведенням розпізнаних символів у консоль.

Актуальність теми. На сьогоднішній день спостерігається зростання кількості автомобілів на дорогах, що збільшує потребу в ефективних системах контролю та моніторингу транспортних потоків. З поглибленням цифрової трансформації у світі, зростає важливість автоматизованого знаходження та розпізнавання номерних знаків автомобілів для підвищення ефективності контролю дорожнього руху, безпеки та знаходженню транспортних засобів.

Мета і задачі роботи. Мета дослідження полягає у підвищенні ефективності автоматизованого виявлення та розпізнавання номерних знаків автомобілів шляхом адаптивного налаштування гіперпараметрів нейронних мереж.

Необхідно виконати такі завдання дослідження:

- провести аналіз існуючих методів та підходів до автоматизованого виявлення і розпізнавання номерних знаків автомобілів, визначити їх переваги та недоліки для обґрунтування напрямів удосконалення;
- розробити метод автоматизованого виявлення та розпізнавання номерних знаків на основі модифікованої архітектури YOLOv11 з інтеграцією механізмів просторової уваги та оптичного розпізнавання символів;
- розробити інформаційну систему для виявлення та розпізнавання номерних знаків автомобілів з використанням розробленого методу;
- провести експериментальне дослідження ефективності розробленого методу шляхом оцінки точності виявлення та розпізнавання номерних знаків за різних умов роботи системи.

Об'єкт дослідження – процеси виявлення та розпізнавання номерних знаків автомобілів на зображеннях та відеопослідовностях.

Предмет дослідження – моделі, методи та алгоритми глибокого навчання для виявлення та розпізнавання номерних знаків автомобілів із застосуванням згорткових нейронних мереж.

Методи дослідження – які застосовані для вирішення поставлених завдань є такими: методи аналізу зображень, згорткова нейронна мережа.

Наукова новизна одержаних результатів. Удосконалено метод автоматизованого виявлення та розпізнавання номерних знаків автомобілів, який відрізняється від існуючих застосувань модифікованої нейромережевої архітектури YOLOv11 з інтегрованими механізмами просторової уваги та двоетапним підходом до обробки зображень, що в поєднанні з оптимізованим процесом навчання на основі адаптивного налаштування гіперпараметрів дозволило забезпечити стабільну роботу системи при різних умовах освітлення та якості вхідних зображень.

Апробація результатів кваліфікаційної роботи магістра та публікації.

За темою кваліфікаційної роботи магістра, автором виконано 1 наукову публікацію – Яворський К.А., Манзюк Е.А., Скрипник Т.К. Пасічник О. А. Визначення обсягу даних для ефективної класифікації номерів автомобілів. Прийнято до друку редакцією журналу “Вісник” Хмельницького національного університету.

Структура та обсяг роботи. Кваліфікаційна робота магістра складається з наступного: реферату, завдання, змісту, переліку скорочень, вступу, 4 розділів, висновків, переліку посилань з 40-ка найменувань. Обсяг основного тексту кваліфікаційної роботи магістра становить 88 сторінок. В роботі наведено 36 світлин і 3 таблиць.

Ключові слова: розпізнавання номерних знаків, нейронні мережі, комп'ютерний зір, автоматизація, CNN, обробка зображень.

Зміст

Перелік скорочень	4
Вступ.....	5
РОЗДІЛ 1 Аналіз сучасного стану досліджень виявлення та розпізнавання номерів автомобілей	7
1.1 Аналіз проблеми знаходження та розпізнавання номерів	7
1.1.1 Актуальність дослідження	7
1.1.2 Модель “ідеальної” системи розпізнавання номерів, приклади її використання	8
1.2 Аналіз рішень для знаходження та розпізнавання номерів	10
1.2.1 Застосування нейромережі для знаходження номерів	10
1.2.2 Особливості застосування згорткової нейронної мережі для розпізнавання об’єктів	11
1.2.3 Сучасні засоби для розпізнавання номерів.....	14
1.3 Постановка мети та задачі дослідження	19
РОЗДІЛ 2 Метод знаходження та розпізнавання автомобільних номерів	20
2.1 Структура методу знаходження та розпізнавання номерів	20
2.2 Етап навчання нейромережі	29
2.3 Знаходження автомобільних номерів	31
2.4 Розпізнавання символів на автомобільних номерах.....	33
Висновки до розділу 2	37
РОЗДІЛ 3 Інформаційна система розпізнавання та класифікації зони розташування номерів та символів.....	39
3.1 Апаратне забезпечення методу.....	39
3.2 Формування датасету.....	42
3.3 Фільтрування та робота з зображенням.....	45
3.4 Середовище для створення та навчання нейромереж	46
3.5 Засіб для навчання нейромережі	47
3.6 Бібліотка для розпізнання символів	48

3.7 Візуалізація результатів навчання.....	49
3.8 Структура інформаційного забезпечення методу.....	52
Висновки до розділу 3	57
РОЗДІЛ 4 Експериментальне дослідження ефективності застосування методу виявлення та розпізнавання номерів	59
4.1 Підготовка датасетів для навчання.....	59
4.2 Метрики тестування нейромережі.....	68
4.3 Налаштування параметрів нейромережі.....	70
4.4 Порівняння результатів навчання.....	71
Висновки до розділу 4	80
Загальні висновки.....	82
Перелік посилань.....	84
Додатки	

Перелік скорочень

Скорочення, термін, позначення	Пояснення
YOLO	Ти дивишся лише раз
HOG	Гістограма направлених градієнтів
DPM	Модель деформованих частин
SiLU	Сигмоїдний лінійний елемент
CNN	Згорткові нейронні мережі
IoU	Перетин на об'єднання
ONNX	Відкритий формат обміну нейронними мережами
GPU	Графічний процесор
OCR	Оптичне розпізнавання символів
R-CNN	Регіональні згорткові нейронні мережі

Вступ

Актуальність теми. На сьогоднішній день спостерігається зростання кількості автомобілів на дорогах, що збільшує потребу в ефективних системах контролю та моніторингу транспортних потоків. З поглибленням цифрової трансформації у світі, зростає важливість автоматизованого знаходження та розпізнавання номерних знаків автомобілів для підвищення ефективності контролю дорожнього руху, безпеки та знаходженню транспортних засобів.

Мета і задачі роботи. Мета дослідження полягає у підвищенні ефективності автоматизованого виявлення та розпізнавання номерних знаків автомобілів шляхом адаптивного налаштування гіперпараметрів нейронних мереж.

Необхідно виконати такі завдання дослідження:

- провести аналіз існуючих методів та підходів до автоматизованого виявлення і розпізнавання номерних знаків автомобілів, визначити їх переваги та недоліки для обґрунтування напрямів удосконалення;

- розробити метод автоматизованого виявлення та розпізнавання номерних знаків на основі модифікованої архітектури YOLOv11 з інтеграцією механізмів просторової уваги та оптичного розпізнавання символів;

- розробити інформаційну систему для виявлення та розпізнавання номерних знаків автомобілів з використанням розробленого методу;

- провести експериментальне дослідження ефективності розробленого методу шляхом оцінки точності виявлення та розпізнавання номерних знаків за різних умов роботи системи.

Об'єкт дослідження – процеси виявлення та розпізнавання номерних знаків автомобілів на зображеннях та відеопослідовностях.

Предмет дослідження – моделі, методи та алгоритми глибокого навчання для виявлення та розпізнавання номерних знаків автомобілів із застосуванням згорткових нейронних мереж.

Методи дослідження – які застосовані для вирішення поставлених завдань є такими: методи аналізу зображень, згортова нейронна мережа.

Наукова новизна одержаних результатів. Удосконалено метод автоматизованого виявлення та розпізнавання номерних знаків автомобілів, який відрізняється від існуючих застосуванням модифікованої нейромережевої архітектури YOLOv11 з інтегрованими механізмами просторової уваги та двоетапним підходом до обробки зображень, що в поєднанні з оптимізованим процесом навчання на основі адаптивного налаштування гіперпараметрів дозволило забезпечити стабільну роботу системи при різних умовах освітлення та якості вхідних зображень.

Апробація результатів кваліфікаційної роботи магістра та публікації.

За темою кваліфікаційної роботи магістра, автором виконано 1 наукову публікацію – Яворський К.А., Манзюк Е.А., Скрипник Т.К. Пасічник О. А. Визначення обсягу даних для ефективною класифікації номерів автомобілів. Прийнято до друку редакцією журналу “Вісник” Хмельницького національного університету.

Структура та обсяг роботи. Кваліфікаційна робота магістра складається з наступного: реферату, завдання, змісту, переліку скорочень, вступу, 4 розділів, висновків, переліку посилань з 40-ка найменувань. Обсяг основного тексту кваліфікаційної роботи магістра становить 88 сторінок. В роботі наведено 36 світлин і 3 таблиць.

РОЗДІЛ 1 Аналіз сучасного стану досліджень виявлення та розпізнавання номерів автомобілей

1.1 Аналіз проблеми знаходження та розпізнавання номерів

1.1.1 Актуальність дослідження

З кожним роком зростає важливість досліджень у сфері знаходження та розпізнавання автомобільних номерів за допомогою сучасних методів. Світ перебуває на етапі стрімкої урбанізації, площа міст зростає і кількість автомобілів на дорогах України, Європи, США збільшується. Згідно зі статистичними даними, щорічно кількість – транспортних засобів зростає, що створює нові виклики для транспортних систем, безпеки та управління дорожнім рухом. Автоматизовані системи розпізнавання номерних знаків відіграють важливу роль у підтриманні порядку на дорогах, автоматизації штрафних санкцій за порушення правил, спрощенні паркування, контролі транспортних засобів та підвищенні загального рівня безпеки. Традиційні методи розпізнавання номерних знаків зазвичай потребували значної участі людини або були обмеженими в своїй точності та швидкості. З розвитком нейромережевих технологій з'явилися нові підходи, які забезпечують вищий рівень автоматизації, точності та масштабованості [1, 2, 12, 19].

Комп'ютерний зір та нейромережі є одними з найактуальніших напрямів дослідження в сучасній науці. Це широкі підгалузі, що охоплюють велику кількість методів і підходів, спрямованих на аналіз, інтерпретацію та розуміння візуальної інформації за допомогою комп'ютерних систем. Основною метою цієї галузі є надання машинам здатності "бачити" та обробляти візуальні дані так, як це робить людина. Ці методи включають аналіз зображень, відео та інших типів візуальних даних для автоматичного розпізнавання об'єктів, класифікації, сегментації та інтерпретації змісту [5, 21].

Нейромережі можуть адаптуватися до різних умов, таких як погода, освітлення та кути огляду, що дозволяє покращити ефективність розпізнавання навіть в складних ситуаціях. Це робить нейромережеві технології критично

важливими для вирішення проблеми обробки великого обсягу даних, що генеруються транспортними системами [8, 29]. Таким чином, інтеграція нейромереж у процес розпізнавання автомобільних номерів дозволяє не лише оптимізувати існуючі системи, але й створювати нові, більш гнучкі та ефективні рішення для сучасних умов.

1.1.2 Модель “ідеальної” системи розпізнавання номерів, приклади її використання

В дослідженні даної теми та розробці нового, кращого методу, варто розглянути очікування від ідеальної системи знаходження та розпізнавання номерів. Наприклад, від автомобіля ми очікуємо велику швидкість руху, маневреність та зручність експлуатації.

Перш за все, система для розпізнавання номерів повинна мати здатність працювати в реальному часі з високою точністю незалежно від зовнішніх умов. Це означає, що вона буде функціонувати ефективно за будь-яких погодних умов (дощ, сніг, туман) і в різних світлових сценаріях (денне світло, сутінки, ніч). Вона має враховувати різні кути огляду камери та швидкість руху транспортних засобів, що вимагає застосування глибокого навчання для адаптації до змінних параметрів і умов роботи.

Наступний важливий компонент ідеальної системи – це інтеграція з іншими міськими і транспортними системами, такими як інтелектуальні камери, системи управління дорожнім рухом, смарт-паркування, бази даних поліції та інших правоохоронних органів.

Усі ці функції є ключовими складовими ефективною системи розпізнавання номерних знаків. Від ідеальної системи очікується, що вона зможе працювати в реальному часі за будь-яких умов. Важливо, щоб вона була інтегрована з іншими міськими та транспортними системами для автоматизації процесів контролю, безпеки та управління трафіком.

Система розпізнавання номерних знаків має широкий спектр застосувань у різних сферах. По-перше, вона необхідна правоохоронним органам для автоматичного виявлення автомобілів, що перебувають у розшуку, фіксації порушень правил дорожнього руху та відстеження транспортних засобів у режимі реального часу. Це значно підвищує ефективність поліції та служб безпеки, дозволяючи швидко реагувати на інциденти та контролювати ситуацію на дорогах [6, 20, 15].

Друге важливе застосування – у міській інфраструктурі та управлінні дорожнім рухом. У великих містах, де потік транспорту є інтенсивним і постійно зростає, система допомагає оптимізувати рух, зменшувати затори, автоматизувати процес паркування та фіксувати порушення. Це полегшує роботу дорожніх служб і дозволяє містам бути більш організованими [4, 7, 33].

Приватні компанії та закриті об'єкти, як-от торговельні центри, аеропорти або бізнес-парки, можуть використовувати цю систему для контролю доступу до своїх територій, забезпечуючи швидкий і безпечний в'їзд або виїзд транспортних засобів. Це покращує ефективність керування паркуванням та підвищує рівень безпеки на об'єктах.

Транспортні компанії, служби доставки та логістичні підприємства також можуть отримати значну вигоду від використання цієї технології. Система дозволяє контролювати маршрути вантажів, відстежувати пересування транспортних засобів і автоматично фіксувати їхню роботу, що оптимізує управління автопарком.

Окремо варто зазначити, що життя у великому місті нерідко супроводжується постійними хвилюваннями про безпеку своїх речей, оскільки вірогідність їх втрати є високою, а надія на їх повернення мінімальна. Втративши автомобіль або інші цінні речі, їх пошук може стати практично неможливим через масовість і інтенсивність міського життя. Однак, система розпізнавання номерних знаків здатна забезпечити людям додаткове відчуття безпеки. Вона дозволяє швидко і точно ідентифікувати транспортні засоби, що дає більше шансів на їх знаходження

та повернення власнику, що суттєво покращує загальну безпеку та спокій в умовах міста.

1.2 Аналіз рішень для знаходження та розпізнавання номерів

1.2.1 Застосування нейромережі для знаходження номерів

Існує багато методів для розпізнавання об'єктів на зображеннях, які використовують різні підходи до обробки інформації. Кожен метод має свої особливості та області застосування, від простих алгоритмів до складних систем, що дозволяють досягти високої точності та ефективності. Успіх таких методів залежить від здатності обробляти великі обсяги даних, розпізнавати складні патерни та адаптуватися до змінних умов.

До 2012 року існувало чимало ефективних методів для детекції об'єктів, серед яких був детектор Віюлі-Джонса, що використовував метод “пересувного вікна”. Цей алгоритм переглядав усі можливі позиції та масштаби зображення, щоб знайти людські обличчя в кадрі. Іншим популярним методом була гістограма направлених градієнтів (HOG), який поліпшував інваріантне виділення ознак та контурів об'єктів. Подальший розвиток цього підходу призвів до появи детектора DPM, який ефективніше розпізнавав об'єкти, враховуючи їхні деформації та складніші контексти [9, 10, 26].

Однак з 2012 року, відбулось швидке відродження згорткових нейронних мереж, які серйозно обігнали конкурентів за швидкістю та ефективністю. Це стало можливим завдяки декільком ключовим факторам, серед яких – покращення апаратних засобів, зокрема графічних процесорів, а також поява великих наборів даних для навчання та вдосконалення архітектур мереж. Важливим етапом стало впровадження глибинних згорткових мереж, таких як AlexNet, які продемонстрували неабиякі результати на змаганнях ImageNet у 2012 році. Вони перевершили традиційні методи класифікації зображень, досягнувши значно вищої точності [31, 39, 40].

Згорткові нейронні мережі стали популярними завдяки своїй здатності автоматично виділяти важливі ознаки зображень на різних рівнях: від простих геометричних форм, таких як краї та кути, до більш складних структур, які відповідають за розпізнавання об'єктів, облич або інших елементів на зображенні. На відміну від класичних методів комп'ютерного зору, де необхідно було вручну визначати ознаки або використовувати алгоритми для їх виділення, CNN самостійно навчаються знаходити ці ознаки на основі великих наборів даних, що значно підвищує їхню ефективність. Серйозне підвищення швидкості та ефективності пояснюється кількома факторами. По-перше, вони здатні одночасно обробляти великі обсяги даних завдяки паралельній обробці на графічних процесорах. По-друге, архітектура CNN дозволяє зменшити кількість параметрів у мережі за рахунок використання згорткових шарів, що дає змогу тренувати моделі на великих наборах зображень без необхідності надмірної обчислювальної потужності. Тому вони виявилися набагато швидшими та точнішими, ніж попередні методи, зокрема ті, що базувалися на традиційному виділенні ознак і класифікації [13. 24].

У підсумку, згорткові нейронні мережі виявилися перспективною та ефективною розробкою для задач детекції об'єктів. Вони забезпечують високу точність завдяки здатності автоматично навчатися на великих обсягах даних, що дає їм перевагу перед традиційними методами. Прогрес у розвитку CNN дозволив створити потужні моделі для розпізнавання та класифікації об'єктів, зокрема автомобільних номерів, що робить їх незамінними у сучасних системах комп'ютерного зору.

1.2.2 Особливості застосування згорткової нейронної мережі для розпізнавання об'єктів

Згорткові нейронні мережі (Convolutional Neural Networks) є одним з основних напрямів у галузі машинного навчання, спеціально розробленим для обробки та аналізу зображень. Ці мережі є частиною більш широкого класу

глибоких нейронних мереж прямого поширення і значно покращують ефективність візуальної обробки даних.

Особливістю згорткових нейронних мереж є використання багатопшарових перцептронів, сконструйованих таким чином, щоб мінімізувати необхідність попередньої обробки вхідних даних. Вони здатні до обробки інформації зображень з мінімальними змінами, що робить їх інваріантними до зсуву або просторового переміщення. Технічно ці мережі часто описуються як просторово інваріантні, що відображає їх здатність до ефективного розпізнавання зображень, не залежно від того, де на зображенні розташовані об'єкти.

Згорткові нейронні мережі були натхненні біологічними процесами в зоровій корі тварин. Зокрема, нейрони візуальної кори реагують на візуальні стимули лише в межах певних частин зорового поля, званих рецептивними полями. Рецептивні поля різних нейронів частково перекриваються, що дозволяє забезпечити повне охоплення зорового поля для ефективного сприймання всіх частин зображення.

Згорткові нейронні мережі базуються на трьох ключових архітектурних принципах, які забезпечують їхню інваріантність до змін масштабу, повороту та зсуву зображення. Завдяки цій структурі CNN забезпечують: локальні рецептивні поля, тобто локальне двовимірне з'єднання між нейронами, що дає можливість зосереджуватися на окремих деталях зображення, спільні синаптичні ваги – що дозволяють виявляти однакові ознаки в різних частинах зображення, значно зменшуючи кількість параметрів, які необхідно оптимізувати, ієрархічну організацію просторових підмножин – що сприяє виділенню важливих ознак на різних рівнях складності [17, 34, 38]. Операція згортки зображена на рисунку 1.1.

Алгоритм роботи операції згортки можна описати так: на вхідне зображення A накладається фільтр із ядром певного розміру B , який переміщується із заданим кроком. На кожному кроці значення елементів зображення і ядра поелементно перемножуються та підсумовуються, утворюючи новий елемент у вихідній матриці C .

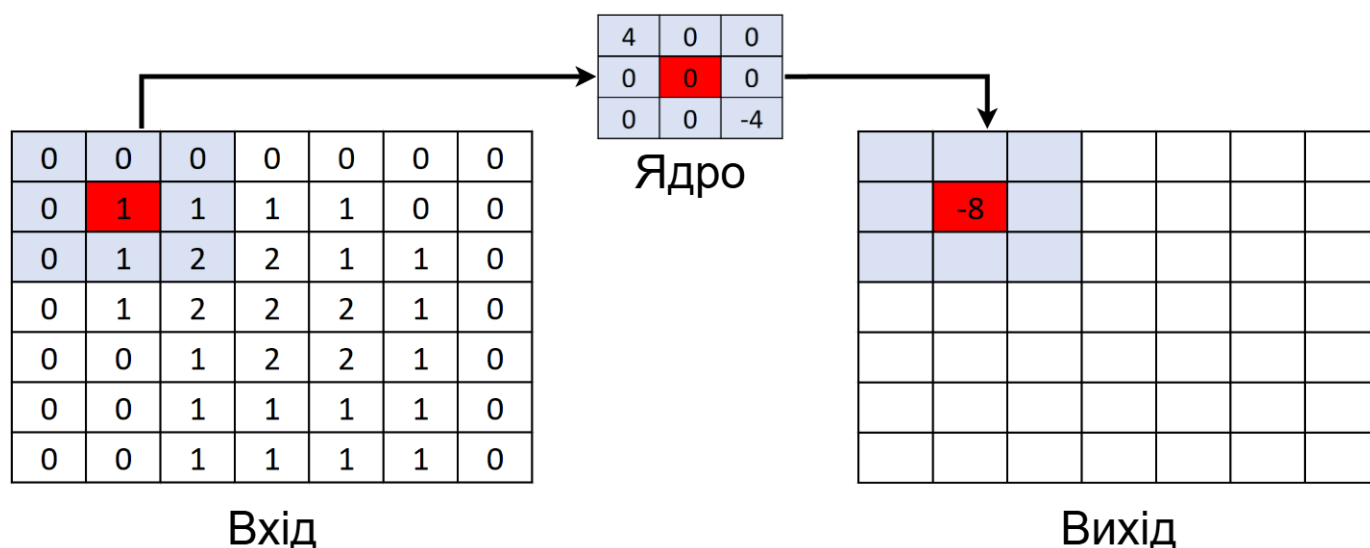


Рисунок 1.1 – Операція згортки, отримання згорткових карт

У структурі згорткової нейронної мережі основними видами прошарків являються: згорткові прошарки (convolutional layer), пулінгові прошарки (pooling layer), повнозв'язані прошарки (fully-connected layer). Загальна архітектура CNN зображена на рисунку 1.2.

У згортковому прошарку нейронної мережі застосовується операція згортки до виходів з попереднього шару, де вагові коефіцієнти ядра (пула) згортки являються навчальними параметрами. Додатково використовується ваговий коефіцієнт, як константний bias-зсув.

Пулінговий шар призначений для зменшення розмірності зображення, що дозволяє спростити обчислення та зменшити обсяг даних, зберігаючи при цьому важливі ознаки. Для цього вихідне зображення розбивається на блоки розміром $w \times h$, і в кожному блоці обчислюється значення за певною функцією. Найпоширенішими є функція максимуму (max pooling), яка зберігає найбільше значення в блоці, або функція середнього (average pooling), що обчислює середнє значення (іноді з урахуванням вагів) для елементів у блоці.

Повнозв'язаний прошарок (Inception module) у нейронній мережі працює таким чином, що кожному елементу попереднього шару відповідає певна область у вихідному зображенні. Кожна нова згортка поступово розширює зону охоплення зображення, поки елементи останніх шарів не почнуть відповідати всьому вхідному

зображенню. Така структура дозволяє Inception-модулю зберігати багат шарову інформацію та забезпечувати оптимальне охоплення ознак [37].

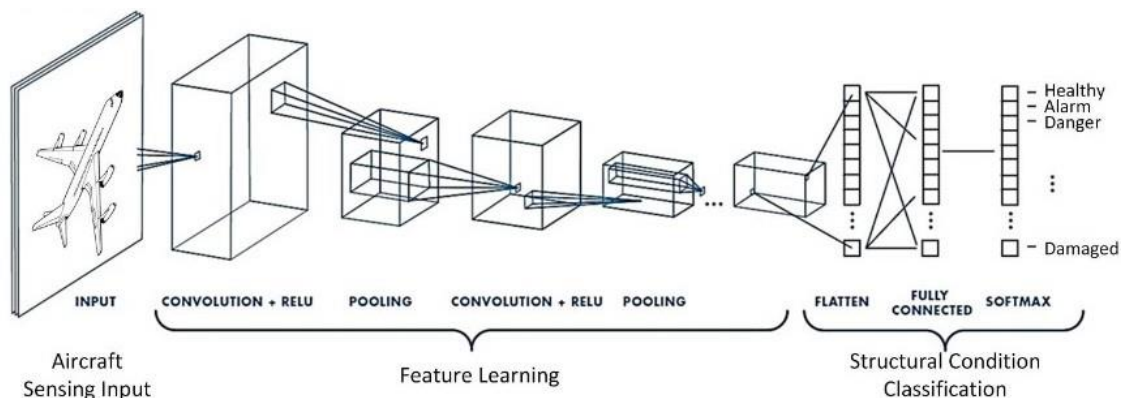


Рисунок 1.2 – Загальна архітектура CNN [9]

1.2.3 Сучасні засоби для розпізнавання номерів

Завдяки швидкому розвитку нейронних мереж та галузі детекції об'єктів в цілому, сьогодні є швидкі та результативні засоби, які можуть забезпечити ефективне знаходження та розпізнавання номерів. Однією з кращих та найефективніших технологій у цій сфері є YOLO, розроблена компанією Ultralytics. Ця система детекції об'єктів використовує CNN та методи глибокого навчання для швидкої та точної обробки зображень, що дозволяє виявляти й розпізнавати об'єкти в реальному часі.

Завдяки своїй архітектурі, YOLO є швидким, здатним здійснювати одночасне виявлення і класифікацію об'єктів в реальному часі, що робить її ідеальним вибором для завдань, де необхідна висока продуктивність. Вона працює за принципом, де зображення поділяється на сітку, і кожен елемент цієї сітки відповідає за детекцію об'єкта, що потрапляє в його область. Це дозволяє YOLO ефективно знаходити об'єкти та розпізнавати їхні категорії, зокрема номери на автомобілях, за лічені мілісекунди [30].

Кожна нова версія YOLO вдосконалювала точність, здатність розпізнавати об'єкти різних розмірів і швидкість навчання. YOLOv11 змінив підхід до

розпізнавання об'єктів, впровадивши єдину архітектуру нейронної мережі, яка одночасно вирішує завдання регресії для обмежувальних прямокутників та класифікації об'єктів. Така інтегрована методологія стала значним відходом від традиційних двоступеневих процесів виявлення, надаючи можливість навчання «від початку до кінця» завдяки своїй повній диференційованості [25, 32, 36].

Модель складається з кількох ключових частин. Каркас (backbone) у YOLO є ключовим елементом, який відповідає за витягування ознак із вхідного зображення на різних рівнях масштабування. Для цього використовуються згорткові шари та спеціалізовані блоки, що дозволяють отримувати карти ознак із різними роздільними здатностями. Архітектура YOLOv11 зберігає схожість із попередніми версіями, використовуючи початкові згорткові шари для поетапного зменшення розмірів зображення. Ці шари створюють основу для витягування ознак, поступово зменшуючи розміри зображення та збільшуючи кількість каналів.

Компонент шия (neck) об'єднує ознаки з різних рівнів масштабування та передає їх до головного блоку для прогнозування. Цей процес включає масштабування вгору, конкатенацію карт ознак і збереження багатомасштабної інформації для подальшої обробки.

Головний компонент – голова (head) YOLOv11 виконує фінальну обробку, генеруючи прогноз для виявлення об'єктів і їх класифікації. Він отримує карти ознак від шиї і формує кінцеві результати: координати обмежувальних рамок і класи об'єктів. У головному компоненті використовуються кілька блоків C_3k_2 для ефективної обробки карт ознак. Цей блок забезпечує швидшу обробку за рахунок менших згорток і оптимізованого використання параметрів, що робить його ефективним і гнучким рішенням для витягування більш детальних ознак.

CBS блоки додаються після C_3k_2 для подальшого вдосконалення ознак. Вони забезпечують виділення релевантних ознак, стабілізацію через нормалізацію та підвищення нелінійності завдяки функції активації SiLU. На кожній гілці детекції завершальними є згорткові шари Conv2D, які трансформують карти ознак у прогнози, що містять координати рамок для локалізації об'єктів, оцінки присутності

об'єктів та класифікаційні оцінки для визначення типу об'єкта. Останній шар детекції об'єднує всі прогнози в єдиний вихідний результат, забезпечуючи точне виявлення та класифікацію об'єктів. Схема алгоритму YOLOv11 зображена на рисунку 1.3.

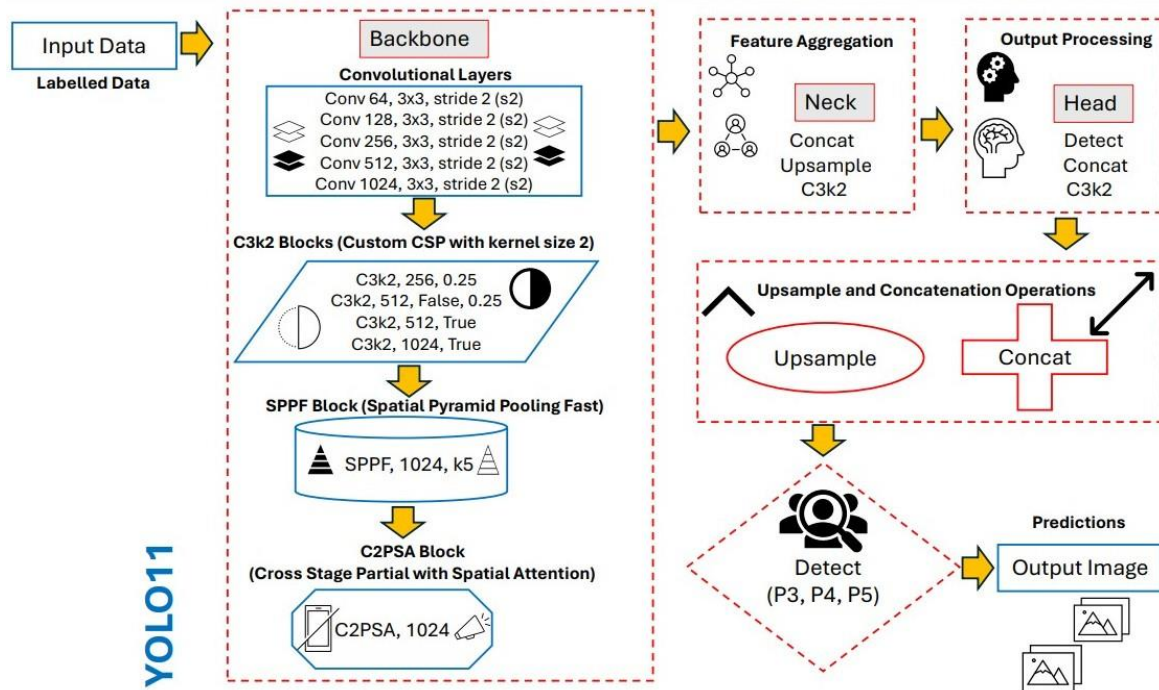


Рисунок 1.3 – Схема алгоритму YOLOv11 [25]

Будучи розширенням і вдосконаленням архітектури YOLOv8, YOLOv11 вводить архітектурні інновації та оптимізації параметрів для досягнення кращої продуктивності виявлення. Одним із головних нововведень стало впровадження блоку $C3k2$, який замінив $C2f$, що використовувався раніше. $C3k2$ оптимізує обчислення за рахунок використання двох малих згорток замість однієї великої, як у YOLOv8, що підвищує швидкість обробки без втрати якості. Рівняння що описує цей блок:

$$C3k2(X) = Conv(Split(X)) + Conv(Merge(Split(X))), \quad (1.1)$$

де Conv – це операція згортки, Split – операція розділення карти ознак на 2 частини, Merge – операція об’єднання вихідних результатів.

У YOLOv11 також зберігається використання блоку просторової піраміди (SPPF), але додано новий блок C₂PSA. Цей блок дозволяє моделі краще фокусуватися на важливих ділянках зображення, покращуючи точність виявлення об’єктів різного розміру та положення завдяки механізмам просторової уваги. Рівняння що описує цей блок:

$$C_2PSA(X) = Attention(Concat(X_{path1}, X_{path2})), \quad (1.2)$$

де Attention – це механізм, що дозволяє фокусуватися на важливих частинах, Concat – операція, яка з’єднує карти ознак.

Рівняння, що описує блок SPPF:

$$SPPF(X) = Concat(MaxPool(X, 5), MaxPool(X, 3), MaxPool(X, 1)), \quad (1.3)$$

де MaxPool(X) – це операція “максимального пулінгу” [3, 16, 35].

Ultralytics пропонує багато інструментів для інтеграції, що робить її однією з найбільш зручних платформ для застосування в реальних проектах, що зображено на рисунку 1.4.

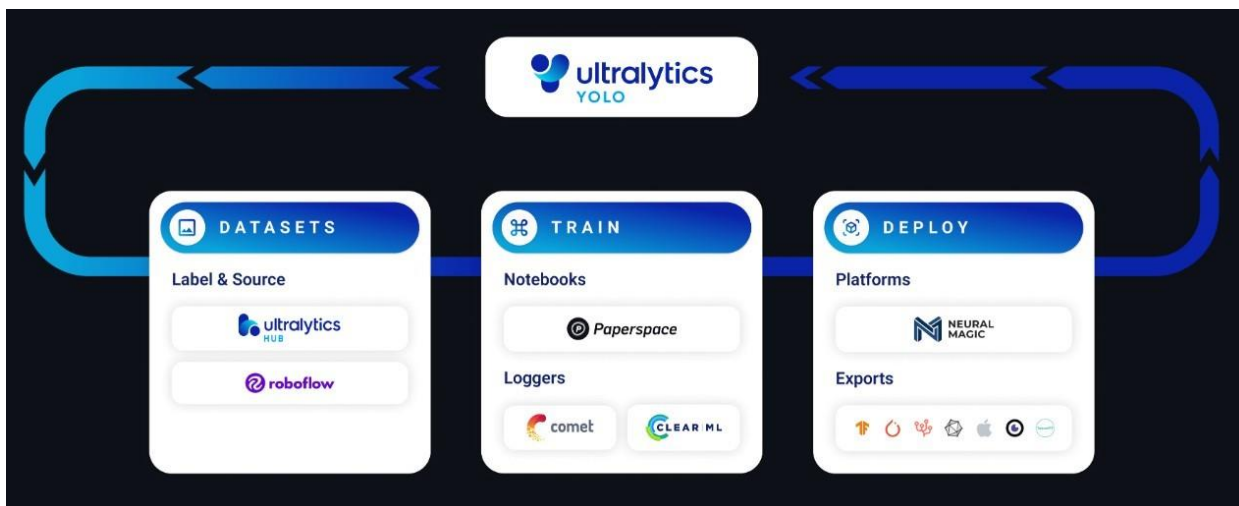


Рисунок 1.4 – Набір сумісних інструментів для навчання [11]

Для розпізнавання тексту, найпопулярнішим інструментом є Tesseract. Це потужна платформа на базі оптичного розпізнавання символів [10]. Його архітектура, що зображена на рисунку 1.5, складається з кількох етапів, починаючи з введення зображення, яке може бути сірим або кольоровим.

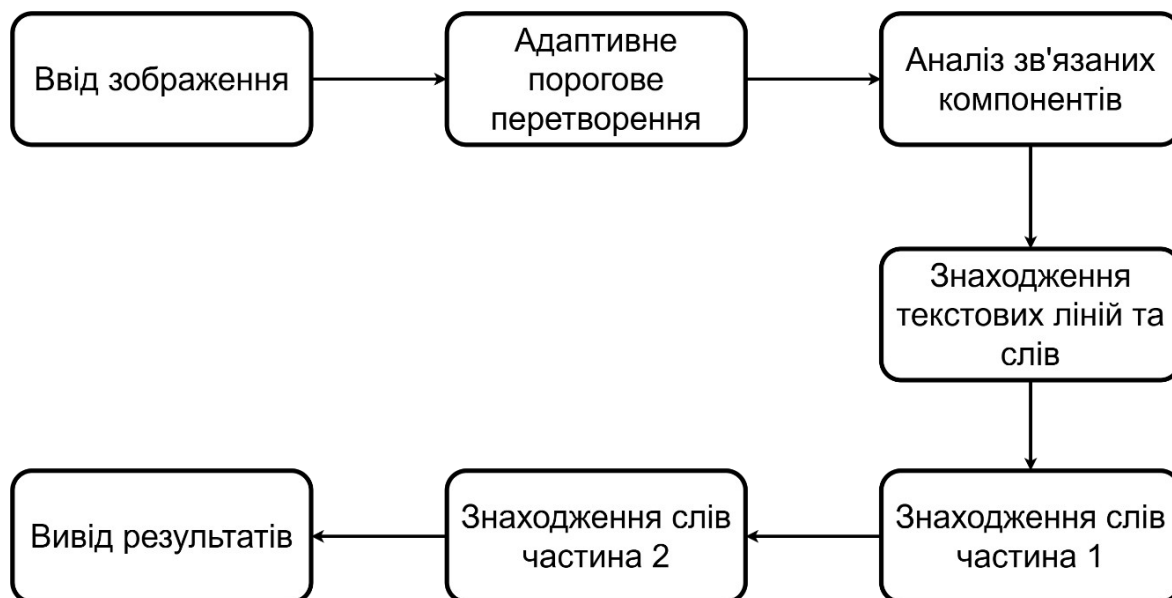


Рисунок 1.5 – Схема архітектури роботи Tesseract [18]

Першим кроком є адаптивне порогове перетворення, яке змінює вхідне зображення на бінарне. Потім відбувається аналіз зв'язаних компонентів, що дозволяє виявити контури символів на зображенні. Після цього Tesseract знаходить

текстові рядки та слова, організовуючи контури символів у відповідні слова. Далі відбувається два проходи для розпізнавання слів. Перший прохід виконується для первинного розпізнавання, а другий прохід уточнює розпізнавання. В результаті цих етапів отримується розпізнаний текст, який є кінцевим результатом роботи системи [22, 27, 28].

1.3 Постановка мети та задачі дослідження

Мета дослідження полягає у підвищенні ефективності автоматизованого виявлення та розпізнавання номерних знаків автомобілів шляхом адаптивного налаштування гіперпараметрів нейронних мереж.

Необхідно виконати такі завдання дослідження:

- провести аналіз існуючих методів та підходів до автоматизованого виявлення і розпізнавання номерних знаків автомобілів, визначити їх переваги та недоліки для обґрунтування напрямів удосконалення;
- розробити метод автоматизованого виявлення та розпізнавання номерних знаків на основі модифікованої архітектури YOLOv11 з інтеграцією механізмів просторової уваги та оптичного розпізнавання символів;
- розробити інформаційну систему для виявлення та розпізнавання номерних знаків автомобілів з використанням розробленого методу;
- провести експериментальне дослідження ефективності розробленого методу шляхом оцінки точності виявлення та розпізнавання номерних знаків за різних умов роботи системи.

РОЗДІЛ 2 Метод знаходження та розпізнавання автомобільних номерів

2.1 Структура методу знаходження та розпізнавання номерів

Знаходження та розпізнавання номерів автомобілей за допомогою нейромережі складається з багатьох кроків, розділених на чотири основні етапи: підготовки датасету зображень, навчання нейромережі, знаходження та розпізнавання номерів, тестування результатів. В цьому розділі представлений послідовний метод вирішення задачі.

Метод знаходження та розпізнавання автомобільних номерів за допомогою нейромережі зображений на рисунку 2.1.

Створення ефективного датасету зображень є необхідним елементом для навчання нейронних мереж. Саме від нього залежить, наскільки добре модель буде здатна розпізнавати та класифікувати об'єкти у реальних умовах. Особливо це актуально для задачі розпізнавання автомобільних номерів, де обставини зйомки можуть варіюватися: від різних погодних умов до різного часу доби.

Велика кількість зображень дозволяє моделі краще вивчити структуру номерних знаків і адаптуватися до різних умов. Як правило, для якісного навчання потрібно декілька тисяч зображень. Різноманітність даних дозволяє нейромережі навчитися працювати з реальними умовами, які можуть значно відрізнятися від лабораторних. Створення датасету складається зі збору даних, їх обробки та розподілу на навчальні, валідаційні та тестові набори.

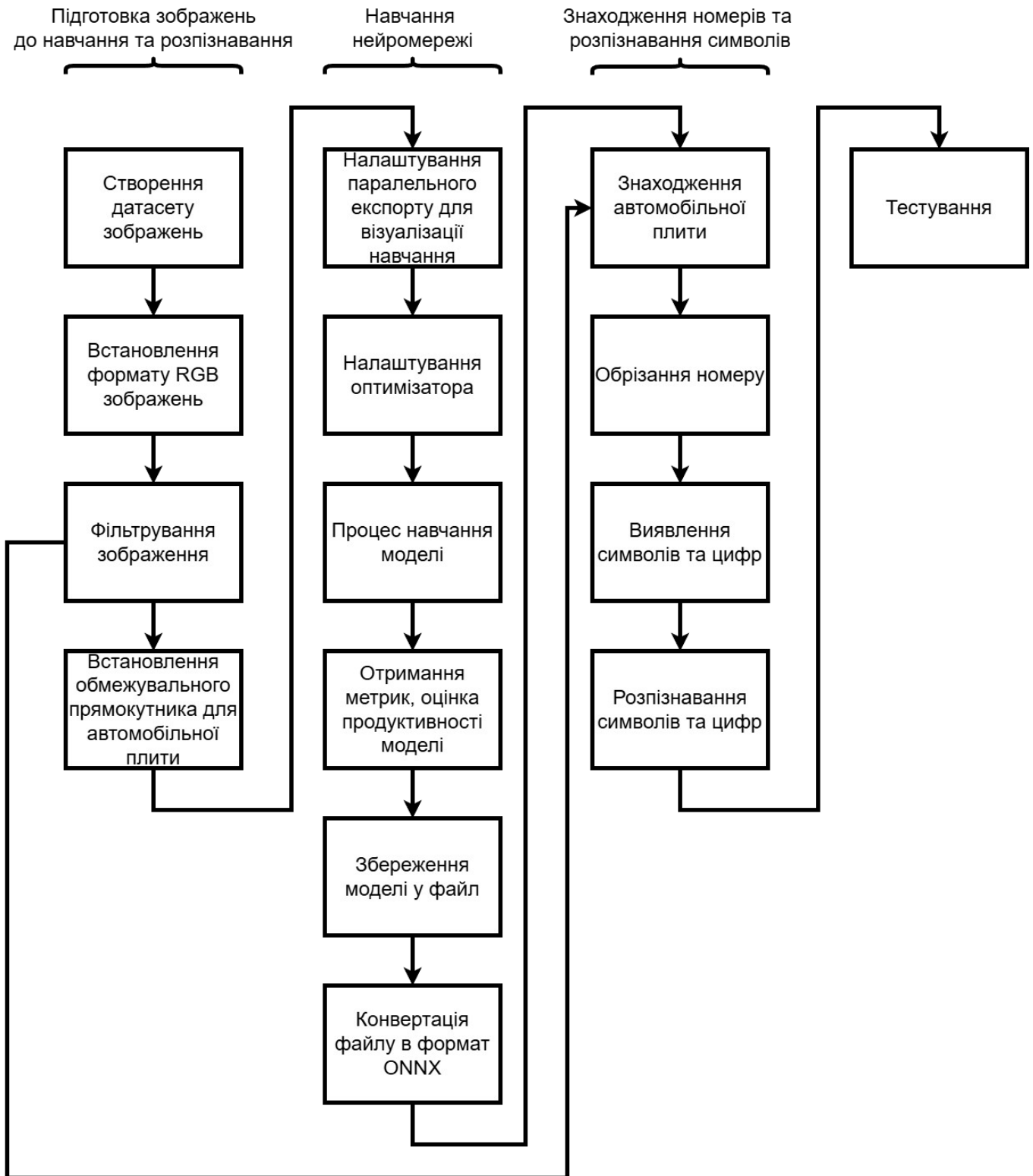


Рисунок 2.1 – Метод знаходження та розпізнавання автомобільних номерів за допомогою нейромережі

Наступним кроком, для того щоб нейромережа могла коректно обробляти зображення, необхідно переконатися, що всі вхідні дані представлені у

стандартному формату – зазвичай це RGB. Цей формат є одним з найпоширеніших у комп'ютерному зорі, оскільки він дозволяє зберігати та передавати кольорову інформацію у трьох каналах (Red – червоний, Green – зелений, Blue – синій). Це забезпечує точне відображення кольорів і дає нейромережі можливість отримувати важливу інформацію про колірні особливості об'єктів на зображенні, зокрема номерних знаків.

Зображення у форматі RGB мають глибину кольору 24 біти, що означає можливість відображення понад 16 мільйонів кольорів, які формуються комбінацією червоного, зеленого та синього каналів. Така велика кількість комбінацій робить RGB найбільш універсальним форматом, який можна використовувати для різноманітних задач, зокрема розпізнавання символів. Якщо зображення надано в іншому форматі, це може створити проблеми для нейромережі, яка навчалася на RGB-даних і очікує подібну структуру вхідних даних.

Окрім конверсії формату, важливо також переконатися, що всі зображення мають однакові параметри роздільної здатності, оскільки нейромережі часто чутливі до розміру вхідного зображення. Наприклад, YOLO та інші об'єктно-орієнтовані мережі потребують зображення однакового розміру, і перетворення формату повинно супроводжуватися відповідним масштабуванням. Це забезпечить ще більшу узгодженість вхідних даних, що є ключовим для побудови точної моделі.

Фільтрація зображень є важливим кроком підготовки датасету, оскільки якість вхідних зображень значно впливає на ефективність навчання нейромережі. Завдання цього процесу – видалити всі непотрібні, неякісні або проблемні зображення, які можуть негативно вплинути на навчання моделі. Наприклад, якщо в датасеті присутні зображення з розмитими або нечіткими номерами, модель може неправильно навчитися розпізнавати такі номери або навіть сприймати їх як допустимі варіанти, що призведе до зниження точності під час тестування.

Розмитість зображень може бути спричинена поганою якістю камери, рухом транспортного засобу під час зйомки, низькою роздільною здатністю або поганими умовами освітлення. Такі фактори роблять деякі зразки малоприслужними для

навчання і вимагають додаткової обробки. Перевірити зображення на розмитість можна автоматично, наприклад, за допомогою аналізу гістограми різкості або функції перетворення Фур'є, яка дозволяє ідентифікувати низькі частоти на зображенні, що свідчить про його розмитість. Інші методи, як використання значення варіації яскравості на пікселях, також можуть допомогти визначити, чи є зображення достатньо чітким для навчання.

Під час підготовки датасету до тренування обов'язковим кроком є встановлення обмежувальних квадратів (bounding boxes). Вони потрібні для того, щоб неймережа розуміла, що на зображенні є об'єктом навчання. Обмежувальний квадрат описується за допомогою координат, які задають прямокутну зону навколо об'єкта. Зазвичай ці координати включають верхній лівий і нижній правий кути прямокутника, або координати центру разом із шириною та висотою області. Важливо, щоб ці координати були максимально точними, оскільки від їхньої правильності залежить, наскільки ефективно модель навчиться виділяти саме ті області зображення, які містять номери. Неправильно встановлені обмежувальні квадрати можуть призвести до того, що модель почне «звертати увагу» на непотрібні деталі фону, що знижує точність і може призвести до помилкових спрацьовувань.

Файл анотацій для кожного зображення містить координати обмежувальних квадратів і служить путівником для моделі під час навчання. Обмежувальні квадрати також служать для того, щоб навчити модель фокусуватися лише на області, де знаходиться номерний знак, ігноруючи непотрібні деталі на фоні, такі як корпус автомобіля, дороги, інші транспортні засоби або люди. Цей підхід дозволяє значно зменшити обчислювальне навантаження, оскільки модель не витрачає ресурси на обробку зайвих частин зображення. Це особливо важливо під час роботи з великими датасетами або при обмежених ресурсах, оскільки правильне виділення ключових областей пришвидшує процес навчання і покращує результати моделі.

Ще одним важливим аспектом є те, що обмежувальні квадрати дозволяють моделі адаптуватися до різних умов освітлення, кута огляду та розташування

номерного знака на транспортному засобі. Якщо датасет містить зображення з різноманітними умовами (наприклад, різними типами автомобілів, змінними погодними умовами, різними кутами зйомки), нейромережа навчиться з більшою точністю знаходити і розпізнавати номери незалежно від обставин. Таким чином, bounding boxes не лише спрощують і пришвидшують процес навчання, але й роблять модель більш універсальною та стійкою до різних варіантів розміщення номерних знаків.

Наступним, є етап навчання нейронної мережі, для задачі розпізнавання номерів автомобілів є важливим і багатоступеневим процесом, що включає кілька ключових кроків.

Перший крок – налаштування паралельного експорту для візуалізації навчання. Це важливий етап, який значно полегшує моніторинг процесу навчання нейромережі. Візуалізація навчання в режимі реального часу дає змогу не просто відстежувати прогрес моделі, але й виявляти будь-які проблеми, які можуть негативно вплинути на кінцеві результати. Зокрема, завдяки візуалізації можна побачити тенденції, аномалії чи нетипові зміни в метриках, які можуть залишитися непоміченими при простому перегляді числових даних.

Для налаштування процесу експорту, перш за все, важливо інтегрувати обраний інструмент в код навчання. Далі необхідно налаштувати автоматичний збір метрик і параметрів для кожної епохи. Це забезпечить оновлення показників втрат, точності та інших метрик у реальному часі, що дозволяє відстежувати ефективність моделі на кожному кроці навчання.

Наступним кроком є налаштування оптимізатора, адже саме він у навчанні нейромереж є ключовим компонентом, який відповідає за поступову адаптацію параметрів моделі з метою мінімізації помилок. Його основне завдання – керувати процесом коригування вагових коефіцієнтів у мережі так, щоб результати навчання наближалися до цільових значень. За кожної ітерації навчання оптимізатор аналізує різницю між передбаченнями моделі та реальними мітками, а також визначає, у якому напрямку та наскільки слід змінити параметри, щоб зменшити цю різницю.

Під час роботи оптимізатор може використовувати різні методи для обчислення і зміни вагових коефіцієнтів. Ці методи враховують як поточні значення помилок, так і їхні тенденції змін. Наприклад, оптимізатор може відслідковувати попередні зміни в параметрах і враховувати їх при прийнятті нових рішень, що дозволяє уникати "застою" або "перепадів" у навчанні, коли модель застрягає у неефективному стані або навпаки – надмірно реагує на кожен новий приклад. Така адаптивність забезпечує плавність навчання та стабільність у досягненні мінімальних значень функції втрат.

Завдяки оптимізатору нейромережа отримує здатність поступово покращуватися, підлаштовуючись під структуру вхідних даних забезпечуючи узгодженість результатів. Це дозволяє моделі більш ефективно розпізнавати патерни, що важливо для точності і надійності в задачах, пов'язаних з обробкою зображень, наприклад, при розпізнаванні номерних знаків автомобілів.

Після налаштування оптимізатора запускається процес навчання моделі, що є центральним етапом у створенні нейромережі для розпізнавання об'єктів. На цьому етапі модель поступово "вчиться" знаходити патерни у вхідних даних, проходячи кілька кроків, кожен з яких спрямований на зниження похибки та покращення якості передбачень.

Наступним кроком є, безпосередньо, процес навчання моделі, який включає серію операцій, що повторюються протягом кількох циклів, які називаються епохами. Кожна епоха починається з того, що зображення з датасету проходять через шари згорткової нейронної мережі, після чого модель генерує прогноз, чи є автомобільні номери на зображенні. Потім обчислюється похибка для кожного прогнозу, яка є різницею між передбаченими значеннями і реальними мітками з датасету. Ця похибка вказує на точність прогнозу та служить сигналом для коригування вагових коефіцієнтів моделі. Навчання зазвичай потребує значного часу, особливо якщо використовуються великі датасети з тисячами зображень, що може тривати години, дні або навіть тижні залежно від складності мережі та обчислювальних ресурсів.

Слідуючий крок – отримання метрик. В кожній епосі модель оцінюється за допомогою вибраних метрик, що дозволяє відстежувати її продуктивність і вчасно коригувати процес навчання. Оцінка моделі на кожній епосі є важливою для того, щоб розуміти, наскільки добре вона здатна виконувати завдання, такі як розпізнавання номерних знаків та їх локалізація на зображеннях. Процес оцінки включає обчислення різних показників, які дозволяють детально вивчити, як модель працює на різних етапах навчання. Зазвичай ці метрики включають точність (accuracy), середній показник (IoU), precision, recall, а також F1-міру, які дають загальне уявлення про роботу моделі.

Важливою частиною навчання є впевненість у тому, що метрики коректно отримані на кожній епосі. Виведення значень метрик на екран або в систему моніторингу є важливим кроком, адже це дозволяє стежити за прогресом моделі в реальному часі. Оскільки навчання може тривати досить довго, мати візуалізацію показників продуктивності дає змогу оперативно реагувати на будь-які аномалії в процесі, наприклад, на перенавчання чи недонавчання моделі.

Моніторинг метрик під час валідації є важливим для розуміння того, як модель поводить себе на нових, невідомих даних. Валідаційний набір дозволяє оцінити, наскільки добре модель здатна узагальнювати свої прогнози і працювати з реальними ситуаціями, що можуть бути схожими на ті, які зустрічатимуться в реальному світі. На основі метрик, отриманих на валідаційному наборі, можна зрозуміти, чи модель готова до подальшого використання, чи потребує вона додаткового навчання або корекцій. Це також дає змогу визначити, чи є потреба в оптимізації параметрів навчання, наприклад, зміни архітектури або налаштування гіперпараметрів для поліпшення результатів.

Коли навчання нейромережі завершується, необхідним кроком є збереження моделі у файл для подальшого використання. Це дозволяє зберегти результати навчання, включаючи структуру мережі та її ваги, щоб можна було завантажити модель пізніше і застосувати її без потреби повторно проходити весь процес навчання. Зазвичай моделі зберігаються у форматах, які забезпечують збереження

не лише параметрів, але й архітектури моделі, що дозволяє відновити її у будь-який момент. Для таких моделей, як нейронні мережі в PyTorch, популярним форматом є .pt або .pth, який зберігає як структуру моделі, так і її ваги. Ці файли містять усі необхідні дані для відновлення моделі в її попередньому стані, включаючи параметри, які були оптимізовані під час навчання.

Збереження моделі дає кілька важливих переваг. По-перше, це дозволяє уникнути повторного навчання на тих самих даних, що може бути ресурсоємним і займати багато часу, особливо для великих датасетів або складних архітектур. Модель, збережена у файлі, може бути завантажена і використана для прогнозування на нових, невідомих даних, що є необхідним для реального застосування. Наприклад, якщо модель була натренована на великих даних для задачі розпізнавання номерів автомобілів, її можна зберегти і використовувати на реальних зображеннях для автоматичного розпізнавання номерних знаків на дорогах. Збережена модель також є важливою для інтеграції в системи розумного міста. Після завершення навчання, модель може бути переведена в середовище, де вона буде виконувати прогнозування у режимі реального часу або на великих обсягах даних. В такому випадку важливо мати можливість швидко завантажити модель без потреби повторного тренування, що зазвичай є значно повільнішим і дорожчим процесом. Окрім того, збереження моделі в файл дозволяє проводити тестування у різних середовищах, наприклад, на різних серверах або пристроях. Завдяки збереженому файлу, можна переносити модель між різними системами, зберігаючи її функціональність та ефективність.

Останнім кроком після навчання та збереження моделі є конвертація файлу моделі у формат ONNX [14]. Це відкритий стандарт, який забезпечує високу переносимість моделей між різними фреймворками та платформами. Однією з основних переваг конвертації моделі в формат ONNX є її сумісність з різними платформами і пристроями. Особливо важливим є використання ONNX для роботи з OpenCV що дозволяє легко інтегрувати нейронні мережі для подальшої обробки та аналізу зображень. Це відкриває можливості для використання потужних моделей у

реальних застосунках комп'ютерного зору, таких як розпізнавання об'єктів, аналіз відео, трекінг об'єктів, автоматичне розпізнавання номерних знаків і так далі, без потреби переписувати код або модифікувати модель під OpenCV.

Конвертація моделі в ONNX дозволяє зберегти оптимізовані ваги та архітектуру нейромережі, а також робить можливим використання її на різних платформах, таких як сервери, персональні комп'ютери, мобільні телефони або вбудовані пристрої. Завдяки підтримці ONNX, можна використовувати різні інструменти та фреймворки для тренування, а потім інтегрувати модель в систему для передбачення або аналізу в реальному часі.

Таким чином, етап навчання нейромережі для розпізнавання номерних знаків є комплексним і складається з налаштування параметрів для візуалізації, оптимізації моделі, навчання з регулярною оцінкою метрик, а також збереження та конвертації моделі для широкого використання. Кожен крок є важливим для забезпечення точності та надійності моделі в реальних умовах, що дозволяє досягти високих результатів у задачах розпізнавання та локалізації номерних знаків.

Наступний етап після тренування – знаходження та розпізнавання номерних знаків на зображеннях. Цей процес складається з кількох ключових кроків: знаходження автомобільних номерів, обрізання номерів, виявлення розташування символів і розпізнавання самих символів.

Перший крок полягає у виявленні області на зображенні, що містить номерний знак, за допомогою вже натренованої моделі. Нейромережа, яка була навчена для локалізації об'єктів, ефективно і точно визначає прямокутну область, де знаходиться номерний знак, надаючи точні координати його розташування.

Наступним кроком є обрізання цієї області, щоб ізолювати номерний знак від фону та інших елементів. Обрізання здійснюється за допомогою координат, отриманих на попередньому кроці. Це дозволяє зменшити обсяг даних для подальшої обробки та зробити задачу більш спеціалізованою, зосередившись лише на номерах, а не на всьому зображенні. Обрізані зображення часто піддаються

попередній обробці, такій як зміна розміру, нормалізація яскравості та контрасту, щоб покращити точність подальших етапів розпізнавання.

Третім кроком є виявлення окремих символів на обрізаних зображеннях номерного знака. Цей етап є надзвичайно важливим і складним, оскільки розпізнавання символів на номері може бути ускладнене різноманітними факторами. Наприклад, номерні знаки можуть містити різні шрифти, що варіюються від країни до країни, або навіть залежно від типу транспортного засобу. Крім того, відстані між символами можуть бути неоднаковими, що додає складності виявленню меж кожного символу. Існують також випадки, коли символи частково перекриваються, або одна частина символу може бути затінена чи погано видно, що ще більше ускладнює завдання. Для того, щоб успішно розділити номерний знак на окремі символи, варто застосувати методи сегментації зображень що передбачають розбиття зображення на окремі регіони, кожен з яких містить один символ.

Останній крок – це безпосереднє розпізнавання символів, які були виділені на попередньому кроці. Для цього використовуються методи оптичного розпізнавання символів. Після виявлення кожного символу на зображенні система розпізнавання порівнює отримані образи з шаблонами або використовує свої ваги, щоб з'ясувати, який саме символ відповідає кожній з частин номерного знака.

2.2 Етап навчання нейромережі

Етап навчання нейромережі є важливим, оскільки саме він має серйозний вплив на ефективність розпізнавання об'єктів. Як вже було сказано, під час навчання штучний інтелект коригує свої внутрішні параметри, щоб мінімізувати помилки в прогнозах. Це дозволяє їй з часом ставати більш точною та ефективною в розпізнаванні шаблонів, таких як об'єкти на зображеннях чи текст на документах. Без правильно налаштованої моделі нейромережа не зможе виконувати завдання на достатньо високому рівні.

Розроблений алгоритм навчання нейромережі зображений на рисунку 2.2.



Рисунок 2.2 – Схема алгоритму навчання нейромережі

На початку відбувається перевірка наявності датасета зображень, без якого безкорисний весь подальший процес. Далі відбувається фільтрація картинок, що включає встановлення правильного фону, розмиття, формату та роздільної здатності. Третім кроком є процес навчання, який відбувається протягом попередньо встановлених епох. Крок обчислення метрик відбувається після кожної епохи та при завершенні тренування. Слідом виводяться отримані результати та відбувається збереження моделі. Для ефективного подальшого її використання відбувається конвертація в формат ONNX.

2.3 Знаходження автомобільних номерів

Основний процес розпізнавання автомобільних номерів, зображений на рисунку 2.3, загалом, складається з трьох кроків: виділення ознак, класифікації та оптимізації. Ці етапи є критично важливими для досягнення точності та надійності розпізнавання символів.

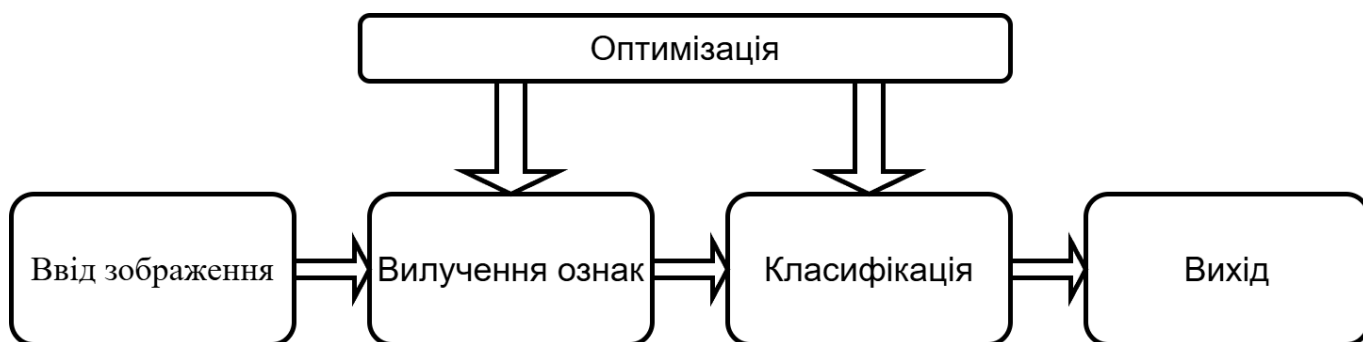


Рисунок 2.3 – Процес розпізнавання автомобільних номерів

Термін “ознака” можна визначити як спосіб репрезентації об’єкта на зображенні. Це характеристики, які дозволяють розрізнити об’єкти різних категорій. Вибір якісного набору ознак дає змогу системі розпізнавання ефективно відрізнити об’єкти різних класів, зменшувати надлишкову інформацію та забезпечувати стійкість до шумів і викривлень. Саме вибір ознак відіграє вирішальну роль у точності класифікації. Важливо, щоб ознаки були легкими для обчислення, стійкими до викривлень і не залежали від змін у зображеннях.

Загалом, ознаки поділяють на фізичні та трансформаційні. Фізичні ознаки зазвичай бувають геометричні, наприклад, точки, лінії, площу, периметр, центр тяжіння, довжину, ширину, співвідношення сторін, прямокутність, округлість, штрихи тощо або статистичні, наприклад, середнє значення, дисперсія, середнє відхилення і тому подібне. Відображувальні ознаки не мають фізичного зображення. Наприклад, до таких належать коефіцієнти Фур’є або Вейвлет-перетворення.

Серед переваг фізичних ознак виділяють відсутність зайвих даних для класифікації, оскільки зібраний набір є оптимальним. Водночас відображувальні ознаки полегшують класифікацію завдяки утворенню чітких меж між класами об'єктів, однак їх використання підвищує обчислювальну складність.

Після вибору оптимального підмножини ознак можна створити класифікатор, який здатний ефективно розрізняти різні класи об'єктів. Процес розпізнавання починається з того, що зображення розбивається на квадратні сітки, де кожна клітинка сітки відповідає за виявлення об'єктів у своїй області. Кожна клітинка прогнозує координати об'єкта, який може знаходитися в її межах, зокрема центр об'єкта, його ширину і висоту, а також ймовірність того, що об'єкт дійсно знаходиться в цій клітинці. Окрім цього, також прогнозується ймовірність того, до якого класу належить об'єкт, наприклад, автомобіль чи пішохід. Після того як мережа виявляє всі об'єкти, застосовується алгоритм який дозволяє уникнути дублювання результатів, відкидаючи прогнози з низькою ймовірністю і залишаючи найкращі. Таким чином, після виконання алгоритму система отримує найбільш точні координати об'єктів на зображенні. У результаті цього процесу мережа класифікує кожен об'єкт на зображенні, визначаючи його клас, що дозволяє здійснити точне розпізнавання об'єктів.

Оптимізація полягає в досягненні найкращих результатів відповідно до встановлених критеріїв для розпізнавання символів. Найчастіше використовується критерій точності, який визначає, наскільки правильно алгоритм виконує класифікацію. Іншим важливим критерієм є стабільність: алгоритм вважається стабільним, якщо він завжди дає схожі результати для різних наборів вхідних даних. Додатково враховуються такі критерії, як вартість (наприклад, витрати на отримання даних або помилки класифікації) та складність (зокрема, обчислювальна складність алгоритму або складність створених правил).

Оптимізація не є окремим етапом у процесі обробки зображень, а радше комплексом допоміжних методів, які підтримують усі етапи цього процесу. На етапі попередньої обробки оптимізація забезпечує високу якість вхідних даних. Під час

виділення ознак вона дозволяє отримати найоптимальніший набір ознак за допомогою методів ранжування, кореляційного аналізу чи оцінки релевантності. На етапі класифікації, після розпізнавання всіх символів, використовуються методи перевірки синтаксису для відповідності отриманих результатів заздалегідь відомим правилам, наприклад, щодо кількості літер і цифр, їхнього порядку тощо. Результати, які не відповідають цим вимогам, проходять повторну обробку. Такий підхід значно знижує ймовірність помилок класифікації.

2.4 Розпізнавання символів на автомобільних номерах

Успішне розпізнавання символів на номерному знаку полягає у правильному асоціюванні кожного символу з літерою (А – Я) або цифрою (0 – 9). Ми будемо використовувати метод оптичного розпізнавання символів (OCR) – заснований на порівнянні форми кожного символу з шаблонами літер та цифр. Алгоритм розпізнавання аналізує візуальні характеристики символу, такі як його контури, відстані між елементами та інші особливості, і порівнює ці характеристики з шаблонами, що містяться в базі даних. Якщо символ є літерою, алгоритм порівнює його з шаблонами літер кирилиці, а для цифр – з шаблонами чисел. Після цього система асоціює кожен символ з відповідною літерою або цифрою, визначаючи таким чином правильне значення для кожного символу на номерному знаку. Ця технологія дозволяє комп'ютеру автоматично визначати текст, що є схожим на процес взаємодії ока та мозку людини: око сприймає текст на зображеннях, а мозок обробляє та інтерпретує отриману інформацію. Під час розробки OCR-систем можуть виникнути певні труднощі. По-перше, інколи важко розрізнити схожі символи, такі як цифри та літери, наприклад, між "0" і "o". По-друге, можуть виникнути труднощі з розпізнаванням тексту на зображеннях з темним фоном або де текст перекривається іншими елементами, такими як графіка чи інші слова.

Важливо зазначити, що система повинна працювати навіть у випадку, коли символи мають дефекти або коли умови освітлення недостатньо хороші для чіткого

розпізнавання. Для цього застосовуються додаткові алгоритми для перевірки результатів. Наприклад, система може враховувати контекст символів, перевіряючи, чи відповідає комбінація літер і цифр формату номерного знака, що характерний для певної країни або регіону. Також можуть бути використані додаткові механізми для виявлення помилок у розпізнаванні, якщо алгоритм не може однозначно визначити символ. Схема розробленої OCR-системи зображена на рисунку 2.4.

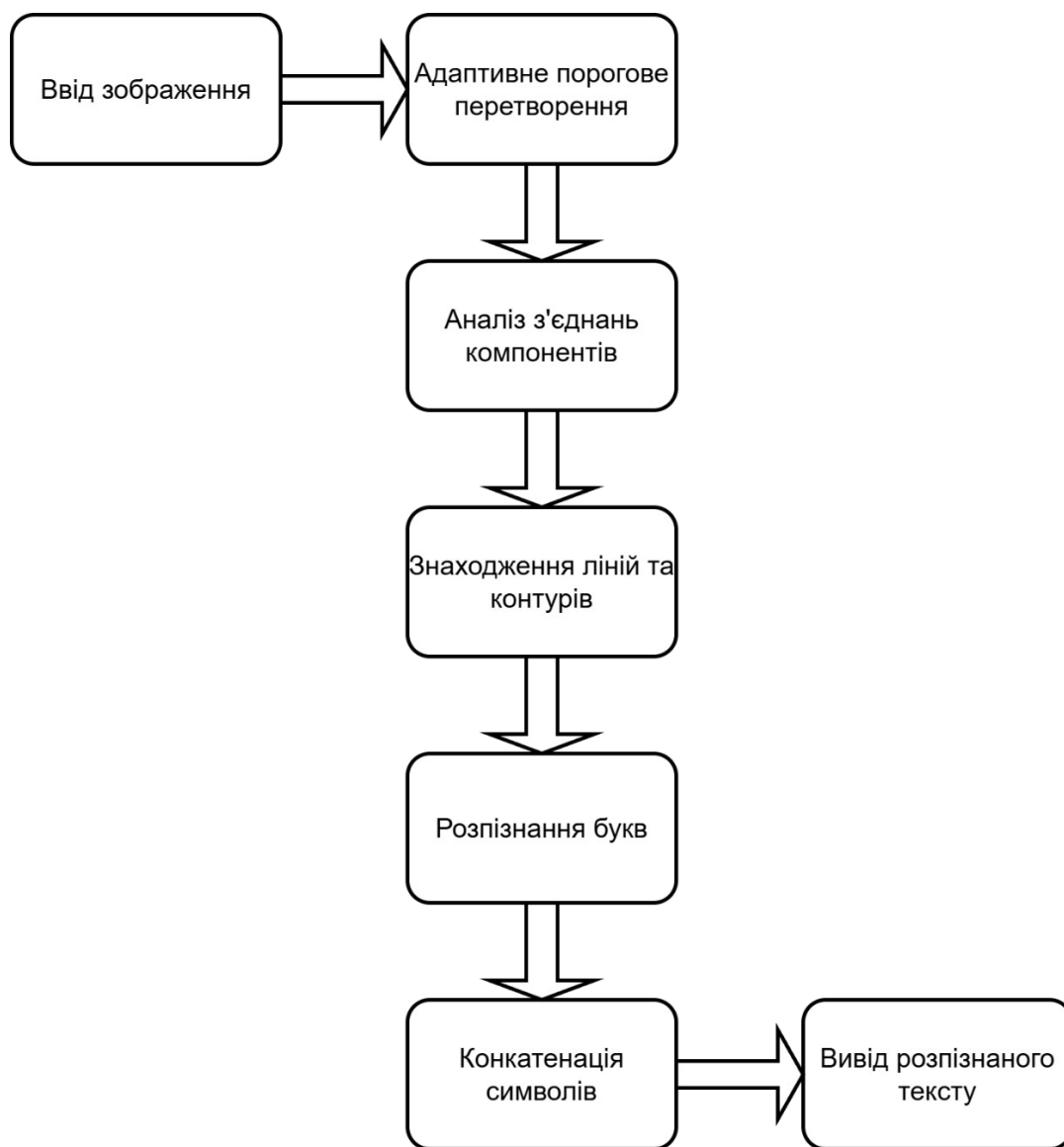


Рисунок 2.4 – Схема розробленої OCR системи

Процес розпізнавання тексту починається з адаптивного порогового перетворення, яке біналізує зображення. Це означає, що кожен піксель зображення

буде віднесений або до одного з двох значень: чорний або білий. Порогове перетворення важливе, оскільки воно дозволяє враховувати різні рівні освітленості в зображенні, що особливо корисно для зображень з різним контрастом. Таким чином, цей етап дозволяє зробити текст чітким, а фон – однорідним, полегшуючи подальше розпізнавання.

Далі проводиться аналіз з'єднаних компонентів, що включає ідентифікацію груп пікселів, які утворюють неперервні контури. Це дозволяє виділити окремі символи або частини тексту. На цьому етапі система може визначити кожний контур, який є потенційним символом, навіть якщо зображення має порушення чи спотворення. Завдяки цьому методу можна ефективно працювати з зображеннями, де текст знаходиться на контрастному фоні (наприклад, білий текст на чорному фоні).

Після того, як контури символів виявлені, вони перетворюються на з'єднані області пікселів, відомі як “blobs”. Потім вони організовуються в текстові рядки, що дозволяє аналізувати текстові структури. Під час цього етапу можна визначити, як символи повинні бути згруповані в рядки, оцінюючи їх відстань між собою та інші ознаки, такі як вирівнювання по горизонталі чи вертикалі.

Далі здійснюється аналіз цих рядків за фіксованими розмірами або еквівалентними розмірами тексту, щоб визначити межі між символами і словами. Використовуються алгоритми, що шукають проміжки між символами, які відповідають або чітким, або нечітким межам. Нечіткі проміжки враховують можливі варіації в межах тексту через різні шрифти або стилі письма.

Наступним кроком є розпізнавання символів. Кожен символ виявлений в рядку намагається бути співвіднесеним з відповідним елементом із бази даних символів. Це дає можливість точно визначити, який символ зображено, навіть якщо його форма або розмір злегка змінюються.

В останньому кроці система конкатенує розпізнані символи в одне ціле слово або текст. Це робиться шляхом послідовного поєднання правильно розпізнаних символів у вірному порядку, що дозволяє відновити весь текст. Текст виводиться у

вигляді результату, який може бути використаний в подальших етапах обробки або аналізу.

Цей процес забезпечує високоякісне розпізнавання символів зображення, використовуючи комбінацію різних методів і алгоритмів для точної та ефективної обробки зображень з текстом.

Алгоритм етапу знаходження та розпізнавання автомобільних номерів зображений на рисунку 2.5.

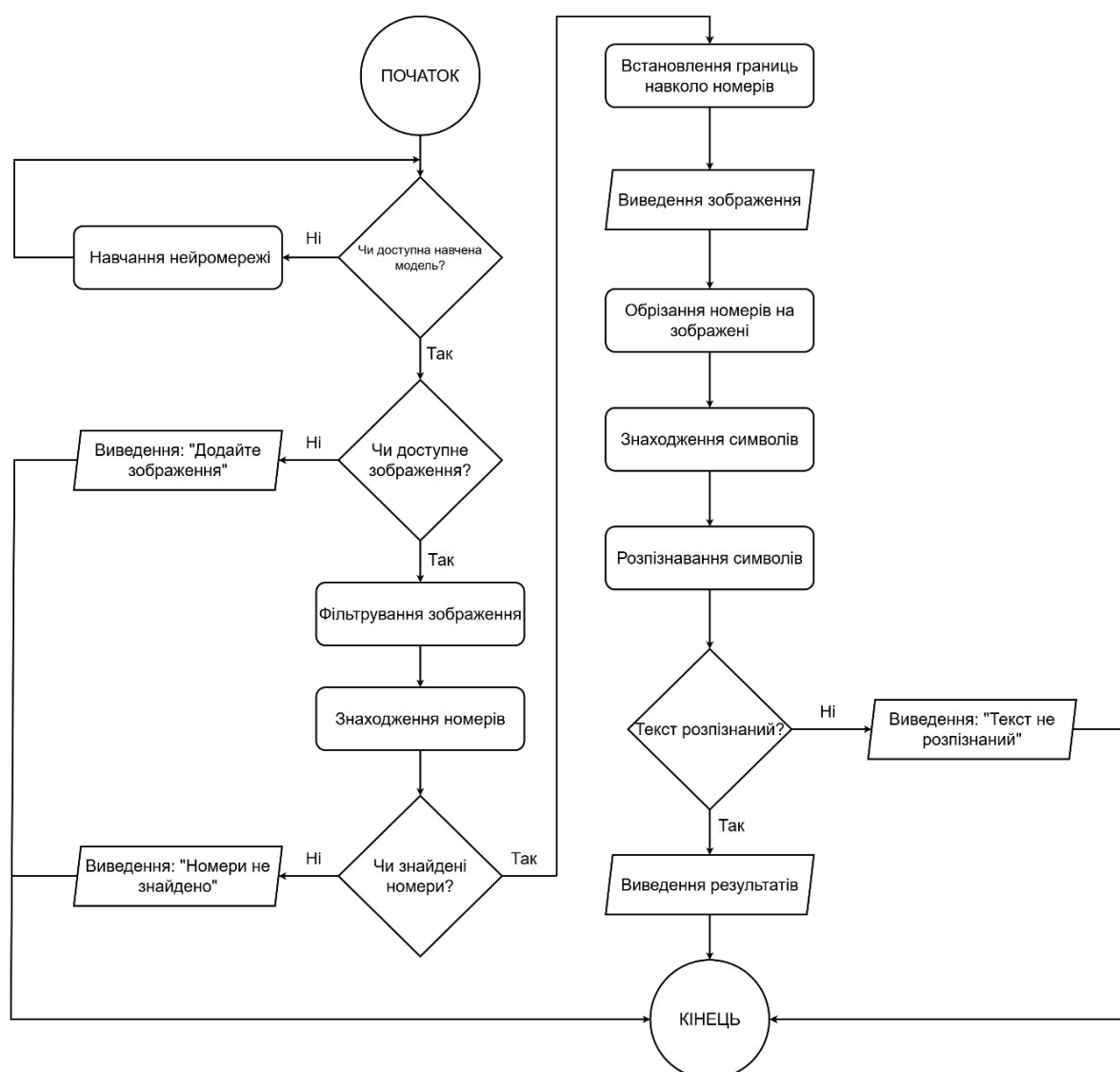


Рисунок 2.5 – Схема алгоритму знаходження та розпізнавання номерів автомобілей

Схема роботи алгоритму розпізнавання номерних знаків починається з перевірки наявності навченої моделі в форматі ONNX, якщо відсутня – відбувається

навчання нейромережі. Після цього виконується виявлення номерних знаків на зображенні, встановлюються границі навколо номерів, що дозволяє чітко виділити область інтересу. Далі відбувається обрізання фону, що зменшує обсяг оброблюваних даних, зберігаючи лише зону з номерним знаком. Після цього алгоритм виявляє окремі символи на номерному знаку, і кожен з них передається на наступний крок –розпізнавання. яке визначає літери та цифри на основі порівняння з шаблонами, результатом чого є отримання тексту номерного знака.

Висновки до розділу 2

В розділі розроблений та описаний метод знаходження та розпізнавання автомобільних номерів нейромережею. Також, наведенні схеми процесу навчання нейромережі, розпізнавання символів.

Розроблений метод складається з чотирьох етапів роботи: підготовки датасету, навчання нейромережі з архітектурою CNN, знаходження номерів за допомогою вилученню ознак, класифікації та оптимізації, знаходження та розпізнавання символів за допомогою методу оптичного розпізнавання символів, тестування результатів за допомогою різних метрик.

На першому етапі створюється датасет зображень, що використовуються для навчання нейромережі. Кожній світлинці присвоюється формат RGB, який дозволяє мережі працювати з ними. Для покращення ефективності навчання зображення фільтруються.

На другому етапі спочатку налаштовується паралельний експорт для зручної та реальної візуалізації результатів навчання. Потім конфігурується оптимізатор для досягнення максимальної ефективності. Після цього проводиться навчання моделі для розпізнавання номерів автомобілів. Далі отримуються метрики для оцінки продуктивності. Наприкінці модель зберігається у файл і конвертується у формат ONNX для універсальності.

Першим кроком на третьому етапі є детекція автомобільних номерів за допомогою попередньо навченої моделі. Знайдені об'єкти обрізаються, після чого в два етапи відбувається виявлення та розпізнавання символів і цифр. Останнім етапом є тестування ефективності мережі

РОЗДІЛ 3 Інформаційна система розпізнавання та класифікації зони розташування номерів та символів

3.1 Апаратне забезпечення методу

Для реалізації методу знаходження та розпізнавання автомобільних номерів необхідне відповідне апаратне забезпечення, яке гарантує ефективне функціонування системи. Застосунок планується використовувати для детекції транспортних засобів у певній зоні міста, що дозволить автоматизувати процес ідентифікації автомобілів.

Щоб забезпечити розпізнавання об'єктів у реальному часі необхідно мати доступ до запису відео, що надходить від камери. Відеопотік передається на стаціонарний комп'ютер для обробки, а також дублюється на веб-сервер для зберігання та додаткової обробки даних. Основна обробка відбувається на стаціонарному комп'ютері, що дозволяє забезпечити високу швидкість і продуктивність системи.

Після передачі відеоданих відбувається процес розкадровки, тобто розбиття відеопотоку на окремі кадри, які будуть аналізуватися індивідуально. На кожному кадрі виконується детекція номерних знаків за допомогою нейромережевої моделі, натренованої у форматі ONNX. Ця модель визначає положення номерних знаків на зображенні та вирізає відповідні області для подальшого розпізнавання символів. Після цього буде застосовуватись застосунок що розпізнає символи на знаку. Для підвищення точності результатів буде застосована додаткова обробка зображень, така як фільтрація шумів, підвищення контрастності або корекція геометрії.

Результати розпізнавання зберігаються. Останнім кроком дані на веб-сервері та стаціонарному комп'ютері синхронізуються для забезпечення резервного зберігання та можливості доступу до результатів у будь-який момент. Ця архітектура дозволяє забезпечити стабільну роботу системи та гарантує, що дані не

будуть втрачені навіть у випадку збою одного з компонентів. Схема алгоритму роботи методу на основі апаратного забезпечення зображена на рисунку 3.1.

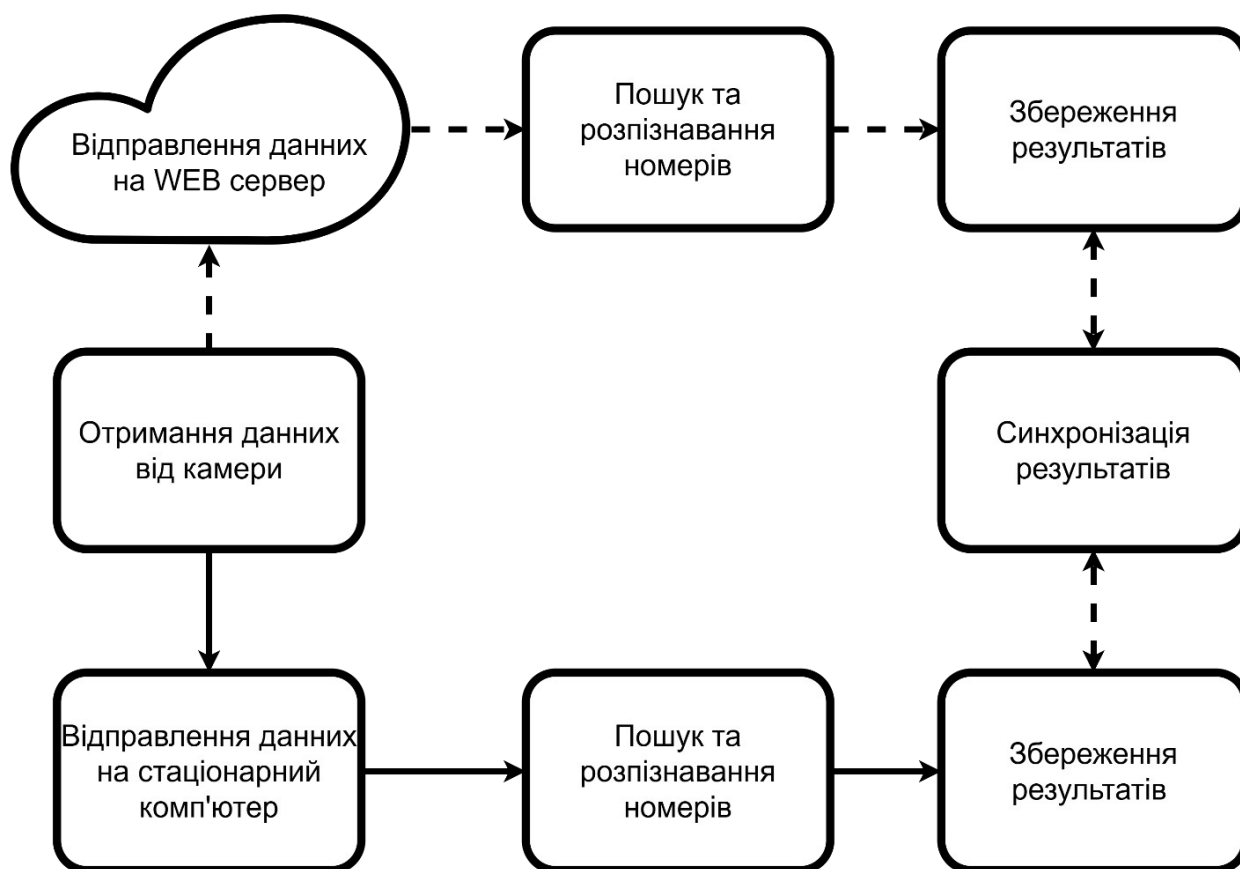


Рисунок 3.1 – Схема алгоритму роботи методу на основі апаратного забезпечення

Процедура застосування методу знаходження та розпізнавання автомобільних номерів неймережею включає комплексне поєднання апаратних засобів (тепловізора) та інформаційної системи, що показано на рисунку 3.2.

Для реалізації апаратного забезпечення потрібні будуть кілька важливих пристроїв. По-перше, необхідні камери, які здатні записувати відео певної території в реальному часі, забезпечуючи отримання якісних зображень номерних знаків транспортних засобів для подальшої обробки. На кожну територію потрібно поставити декілька пристроїв, для збільшення результативності в реальних умовах. По-друге, потрібно використовувати комп'ютер, який виконуватиме обробку отриманих відеопотоків за допомогою неймережі, а також зберігатиме результати розпізнавання для подальшого аналізу або використання.

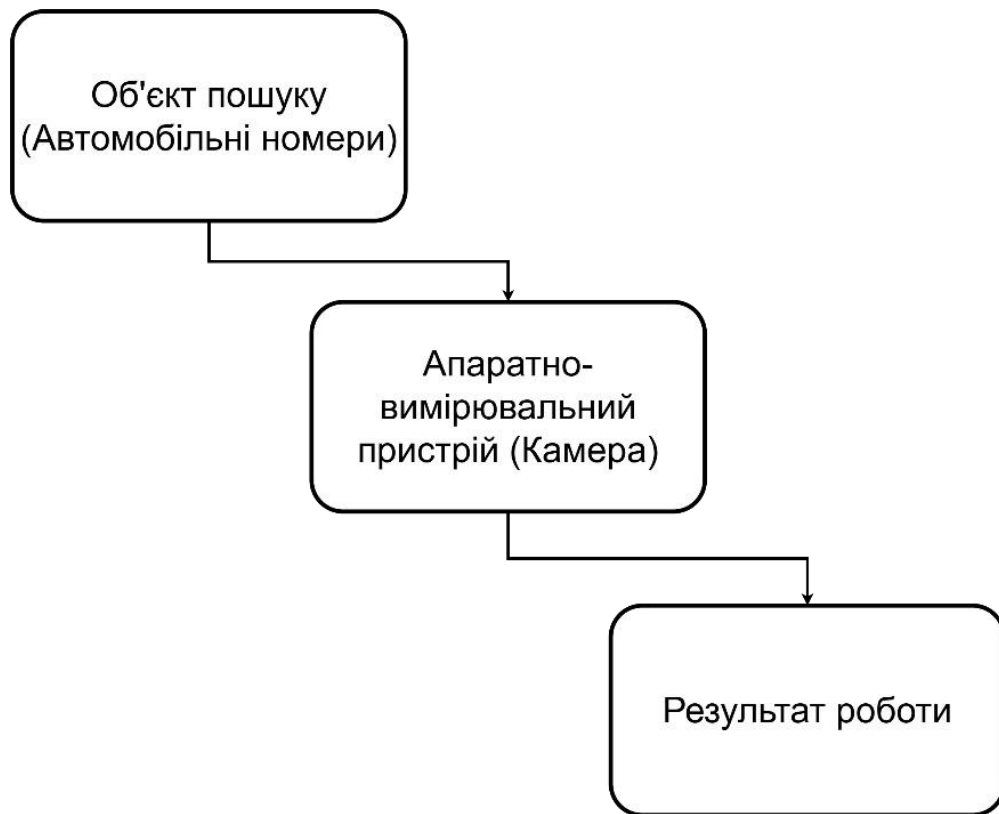


Рисунок 3.2 – Схема процедури застосування методу

Для підключення і координації роботи всіх цих пристроїв знадобиться комутатор, який дозволить об'єднати кілька камер та інших елементів системи в єдину мережу, забезпечуючи стабільну передачу даних. Крім того, важливу роль відіграє роутер, який буде ключовим елементом для передачі інформації між пристроями та забезпечення доступу до мережі для збереження результатів розпізнавання або віддаленого моніторингу. Схема апаратного забезпечення методу зображена на рисунку 3.3.

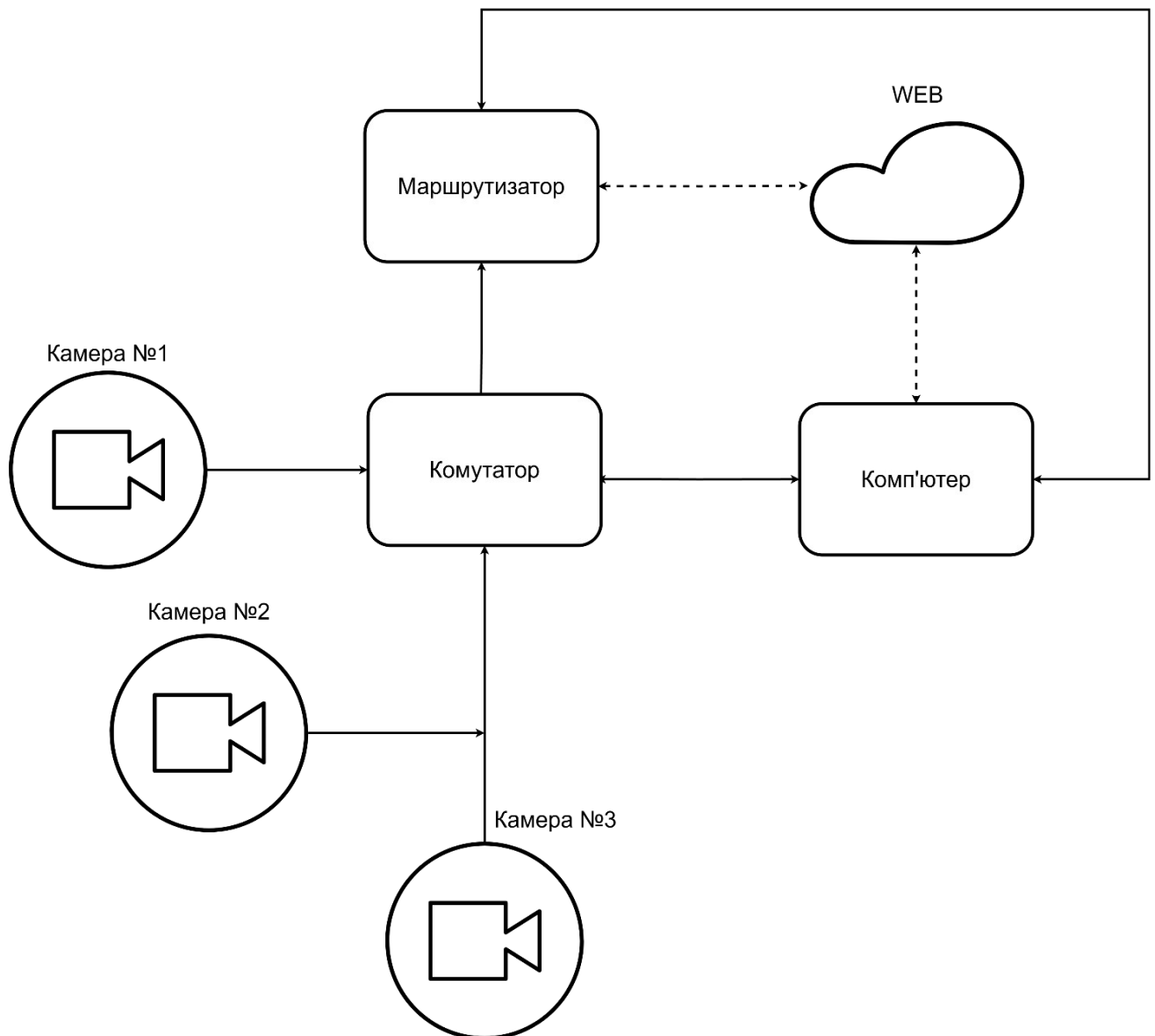


Рисунок 3.3 – Схема апаратного забезпечення методу

Завдяки правильно підбраному апаратному забезпеченню, система зможе працювати ефективно і точно, забезпечуючи надійне розпізнавання автомобільних номерів у міських умовах.

3.2 Формування датасету

Для зручного та ефективного збору, обробки даних та встановлення обмежувальних квадратів, а також для фільтрації зображень варто використати спеціалізовані інструменти. Одним із таких інструментів є Roboflow.

Однією з основних можливостей Roboflow є анотація зображень, що зображено на рисунку 3.4. Користувачі можуть вручну або автоматично позначати об'єкти на зображеннях, створюючи відповідні мітки, які після цього будуть використовуватися для тренування моделей. Це особливо корисно для задач, де важливо точно визначити межі об'єктів на зображеннях, як у випадку з розпізнаванням номерних знаків або індексами.

Застосунок дозволяє автоматично генерувати обмежувальні квадрати для об'єктів на зображеннях. Це суттєво прискорює процес підготовки даних, оскільки замість ручного малювання квадратів, можна скористатися вже готовими алгоритмами для автоматичної детекції та анотації об'єктів. Такі обмежувальні квадрати допомагають нейромережі краще вивчати місце розташування об'єктів, що є важливим для задач розпізнавання та локалізації. Незважаючи на це, найефективніше виставляти ці квадрати вручну, оскільки автоматична система може припустити помилки в ряді ситуацій. Наприклад, у складних випадках, коли об'єкти на зображенні перекриваються, знаходяться під кутом або мають складну текстуру, автоматичне розпізнавання меж об'єкта може бути неточним. Роботизовані алгоритми можуть помилятися при визначенні точних кордонів номерного знака, якщо він має неправильну орієнтацію, специфічний шрифт або перебуває в умовах з низьким освітленням. Це особливо важливо при роботі з номерними знаками, де необхідно точно визначити кожен символ на зображенні. Тому, незважаючи на зручність автоматичних інструментів, вручну виставлені обмежувальні квадрати гарантують вищу точність і дозволяють уникнути помилок, що можуть виникнути під час автоматичного процесу. Це дає більше контролю над якістю анотацій та підвищує ефективність подальших етапів розпізнавання.

Оскільки він працює в онлайн-середовищі, всі операції, включаючи пошук, скрапінг та імпорт зображень, здійснюються безпосередньо в інтернеті. Це забезпечує зручність у використанні, оскільки користувачі можуть швидко завантажувати зображення з будь-яких джерел без необхідності вручну переміщати файли на локальний комп'ютер. Крім того, весь процес зберігається онлайн, що

дозволяє без проблем отримати доступ до даних та результатів з будь-якої точки, а також інтегруватися з іншими хмарними сервісами. Завдяки цьому, RoboFlow пропонує високу зручність та ефективність для користувачів, які працюють над великими проектами, особливо коли потрібно працювати з великими обсягами даних або командою.

Платформа підтримує широкий набір інструментів для попередньої обробки, таких як зміна яскравості, контрасту, додавання шуму, обертання, обрізка, масштабування та багато іншого. Ці інструменти дозволяють створювати більш різноманітний і якісний набір даних, який покращує результати навчання моделей. Крім того, RoboFlow дозволяє автоматизувати процес підготовки зображень до навчання шляхом попередньої зміни розмірів або перетворення форматів файлів.

Контроль версій датасетів – ще одна важлива функція RoboFlow. Платформа автоматично створює версії кожного внесеного змінення в датасет, дозволяючи розробникам повертатися до попередніх версій або порівнювати результати моделей, навчання на різних етапах. Ця функція дуже корисна під час експериментів, оскільки дозволяє легко аналізувати вплив змін у даних на продуктивність моделі.



Рисунок 3.4 – Встановлення обмежувальних рамок у Roboflow

Таким чином, Roboflow значно спрощує процес підготовки датасетів і роботи з ними, дозволяючи користувачам зосередитися на розробці та оптимізації моделей, а не на складних етапах попередньої обробки даних. Це робить платформу надзвичайно корисною для проектів у сфері комп'ютерного зору та штучного інтелекту.

3.3 Фільтрування та робота з зображенням

Для фільтрування та обробки зображень у багатьох сучасних проєктах рекомендується застосовувати окремі програмні рішення, які забезпечують широкий спектр функцій для роботи із зображеннями. Одним із найбільш популярних і ефективних рішень у цій галузі є OpenCV. Це відкрита бібліотека, яка вже багато років залишається стандартом для розробників, науковців і ентузіастів, що працюють у сфері комп'ютерного зору. OpenCV надає інструменти для виконання численних завдань, включаючи фільтрацію, покращення якості, розпізнавання об'єктів, аналіз руху, побудову 3D-сцен та багато іншого.

Основний функціонал охоплює широкий набір можливостей для роботи із зображеннями. Серед них – обробка та фільтрація, включаючи корекцію кольорів, зміну розміру, обертання, обрізання та накладення фільтрів. OpenCV також дозволяє виконувати більш складні операції, такі як детекція країв, видалення шумів, сегментація об'єктів і вирівнювання зображень. Для задач комп'ютерного зору бібліотека пропонує інструменти для розпізнавання об'єктів, відстеження руху, аналізу потокового відео та побудови 3D-моделей. Функціонал бібліотеки постійно розширюється завдяки внеску величезної спільноти розробників, що забезпечує актуальність і ефективність для вирішення сучасних задач [23].

Завдяки доступності вихідного коду бібліотека активно розвивається, а користувачі можуть адаптувати її під свої специфічні потреби. Спільнота OpenCV

включає тисячі професіоналів і ентузіастів, які діляться своїми знаннями через форуми, документацію, навчальні курси та готові приклади коду.

Таким чином, OpenCV залишається одним із найпотужніших та найпопулярніших інструментів для фільтрування та роботи із зображеннями. Її багатий функціонал, підтримка спільноти та гнучкість використання роблять цю бібліотеку ідеальним вибором для розробників, які прагнуть реалізувати складні проєкти у сфері комп'ютерного зору.

3.4 Середовище для створення та навчання нейромереж

Для роботи з нейронними мережами, зокрема для задач обробки зображень, потрібно використовувати спеціалізовані інструменти. В проєкті застосований PyTorch – одна з провідних бібліотек для роботи з нейронними мережами, яка дозволяє створювати, навчати та тестувати моделі машинного навчання різного рівня складності. Бібліотека є потужним рішенням для роботи з обчислювальними задачами, які вимагають гнучкості, високої продуктивності та простоти у використанні. Вона забезпечує інструменти для роботи з нейронними мережами та необхідна для виконання складних задач штучного інтелекту, таких як обробка зображень, розпізнавання мовлення, обробка текстів та багато інших.

Основний функціонал PyTorch включає численні можливості, що дозволяють ефективно працювати з моделями глибокого навчання. Однією з ключових особливостей є динамічний обчислювальний граф, який дозволяє змінювати архітектуру нейронної мережі під час виконання програми. Це робить PyTorch особливо зручним для дослідників, які створюють і тестують нові моделі. Бібліотека підтримує автоматичне обчислення градієнтів, що спрощує оптимізацію параметрів і зменшує потребу у складних математичних розрахунках вручну.

PyTorch також має вбудовані інструменти для роботи з великими наборами даних. TorchVision, одна з основних бібліотек, пропонує доступ до популярних датасетів, інструменти для попередньої обробки та аугментації зображень. Це

значно полегшує підготовку даних для тренування моделей. Крім того, PyTorch забезпечує зручний інтерфейс для побудови шарів нейронних мереж і створення власних архітектур. Завдяки підтримці GPU, бібліотека забезпечує високу швидкість тренування навіть для великих моделей.

Ще однією важливою особливістю PyTorch є можливість експорту моделей у формат ONNX, що робить їх сумісними з іншими платформами та пристроями, такими як мобільні телефони або хмарні сервери. Бібліотека також підтримує розподілене навчання на кількох GPU або навіть у кластері, що дозволяє масштабувати навчання для вирішення складних задач.

Популярність PyTorch пояснюється її простотою, ефективністю та універсальністю. Вона має інтуїтивно зрозумілий інтерфейс, що робить її доступною для новачків, і водночас надає всі необхідні інструменти для професійних розробників. Бібліотека підтримується великою спільнотою розробників і активно використовується у дослідницьких проектах та індустрії. PyTorch інтегрується з популярними інструментами, такими як TensorBoard і Comet, що дозволяє легко відстежувати процес тренування моделей.

PyTorch заслужено вважається одним із найбільш важливих інструментів для роботи з нейронними мережами, завдяки чому він залишається вибором номер один для багатьох фахівців у галузі штучного інтелекту.

3.5 Засоби для навчання нейромережі

Для налаштування та навчання нейромережі варто використати спеціально створений для цього інструмент. Одним із найбільш популярних та ефективних засобів у цій галузі є Ultralytics – програмна платформа, що пропонує сучасні рішення для роботи з моделями серії YOLO. Ultralytics відома своєю простотою у використанні, високою продуктивністю та широким спектром функцій, що дозволяє розробникам швидко створювати, тренувати, тестувати та розгортати моделі комп'ютерного зору. Платформа пропонує інтуїтивний інструментарій для повного

циклу роботи з моделями нейронних мереж. Її основні функції включають тренування моделей YOLO, які можна швидко навчити на власних датасетах.

У проєкті було використано YOLOv11. Це остання ітерація серії YOLO, яка базується на основах, закладених у YOLOv1. Вперше представлений на конференції YOLO Vision 2024, YOLOv11 є значним кроком уперед у технології реального часу для виявлення об'єктів. Ця нова версія містить суттєві покращення як в архітектурі, так і в методах навчання, що дозволяє досягти нових висот у точності, швидкості та ефективності.

Інноваційний дизайн YOLOv11 включає передові методи витягнення ознак, що дозволяють захоплювати більш тонкі деталі, зберігаючи при цьому невелику кількість параметрів. Це призводить до покращення точності в різноманітних завданнях комп'ютерного зору, від розпізнавання об'єктів до класифікації. Крім того, YOLOv11 демонструє значні покращення в швидкості обробки, значно підвищуючи можливості реального часу. Ефективність порівняно з попередниками зображена на рисунку 3.5.

3.6 Бібліотка для розпізнавання символів

Для розпізнавання знайдених символів потрібно використовувати потужне рішення, що забезпечить ефективне виконання даної задачі. Для цього був застосований Tesseract. Це один з популярних відкритих движків для оптичного розпізнавання символів, який активно використовується для перетворення тексту на зображеннях у редагований текстовий формат. Tesseract написаний в основному на мові програмування C++, що забезпечує швидку роботу застосунка.

Tesseract орієнтується на зменшення кількості помилок у процесі розпізнавання, що дозволяє отримувати точні результати навіть на складних зображеннях. Однак Tesseract не зосереджується на досягненні абсолютної точності, тому на його вихід може впливати різноманітність вхідних зображень, які можуть містити складні шрифти, фони чи знижену якість.

Tesseract підтримує велику кількість мов, для кожної існують окремі файли, які містять відповідну інформацію для ефективного розпізнавання символів. Наприклад, для англійської мови користувач може завантажити лише англійський мовний файл, що дозволяє працювати з текстами на цій мові.

Залежно від типу зображення, застосунок може забезпечити різну точність розпізнавання. На простих зображеннях з чітким контрастом, таких як чорно-білі або сірі, система зазвичай досягає високої точності, навіть до 100%. Проте для більш складних зображень, таких як кольорові фотографії чи зображення з різними шрифтами і складними фонами, точність може знижуватись. Дослідження показали, що для досягнення кращих результатів розпізнавання тексту зображення краще перетворювати на градації сірого, оскільки це зменшує складність для алгоритмів і допомагає уникнути певних помилок, які можуть виникати при роботі з кольоровими зображеннями.

Робота Tesseract не обмежується лише стандартними зображеннями – він може застосовуватися до великого спектру різних вхідних даних. Завдяки відкритому коду, Tesseract можна інтегрувати в різноманітні програми, розширюючи його функціональність для специфічних потреб, наприклад, для аналізу документів, сканування архівів або автоматизації робочих процесів.

Попри те, що Tesseract є потужним інструментом для OCR, його ефективність значною мірою залежить від якості вхідного зображення. Всі ці фактори – тип зображення, якість тексту та мовний файл – мають велике значення для досягнення максимальної точності в процесі розпізнавання. Це робить Tesseract універсальним інструментом для багатьох застосувань, хоча й вимагає певної підготовки та налаштувань для досягнення найкращих результатів.

3.7 Візуалізація результатів навчання

Для візуалізації метрик та результатів навчання, був обраний інструмент Comet ML – це потужна платформа для моніторингу, відстеження та управління

експериментами в сфері глибокого навчання, що забезпечує вченим та дослідникам можливість більш ефективно управляти їхніми моделями та відслідковувати прогрес на всіх етапах розробки. Вона надає інструменти для збору, візуалізації та аналізу метрик та результатів експериментів, що дозволяє порівнювати різні налаштування моделі та вибирати найкращі підходи для оптимізації.

Однією з головних особливостей Comet є відстеження метрик та параметрів моделі в реальному часі що зображено на рисунку 3.5. Платформа дозволяє відстежувати такі метрики, як точність, втрати, а також інші параметри, які впливають на якість моделі, як швидкість навчання або гіперпараметри. Це дає змогу швидко коригувати стратегію навчання, надаючи можливість порівнювати ефективність різних моделей і знаходити оптимальні налаштування.

Інтерфейс Comet ML надає зручні інструменти для візуалізації результатів, що допомагає краще розуміти, як змінюються параметри і метрики під час тренування. Ці візуалізації включають графіки для аналізу прогресу навчання, зміни втрачених значень, точності та інших ключових показників. Також платформа дозволяє створювати інтерфейси для порівняння різних експериментів, що дозволяє відразу бачити, який з варіантів є найбільш ефективним.

Інтерфейс Comet ML надає зручні інструменти для візуалізації результатів, що допомагає краще розуміти, як змінюються параметри і метрики під час тренування. Ці візуалізації включають графіки для аналізу прогресу навчання, зміни втрачених значень, точності та інших ключових показників. Також платформа дозволяє створювати інтерфейси для порівняння різних експериментів, що дозволяє відразу бачити, який з варіантів є найбільш ефективним.

NAME	LAST	
	VALUE	STEP
loss	58.9532	6270
lr/pg0	0.00005375	6270
lr/pg1	0.00005375	6270
lr/pg2	0.00005375	6270
metrics/mAP50-95(B)	0.504705	6270
metrics/mAP50(B)	0.738522	6270
metrics/precision(B)	0.6973	6270
metrics/recall(B)	0.668709	6270
model/GFLOPs	6.444	209
model/parameters	2.59062e+6	209
model/spe...yTorch(ms)	217.937	209
train/box_loss	1.10878	6270

Рисунок 3.5 – Візуалізація отриманих результатів за допомогою Comet

Comet ML також надає можливість відслідковувати не лише моделі, але й весь процес розробки. Це включає контроль за версіями коду, параметрів даних, змін у гіперпараметрах та конфігураціях середовища. Це допомагає відтворювати результати та забезпечує надійність і стабільність під час розробки моделей машинного навчання.

Особливо важливим є аспект співпраці: Comet ML дозволяє команді працювати спільно, обмінюватися результатами експериментів, обговорювати зміни в моделі та відслідковувати, хто і коли вніс зміни. Це робить платформу ідеальним вибором для командної роботи над великими проектами машинного навчання.

Для того, щоб зберігати всі важливі дані та не обтяжувати локальні ресурси, Comet ML використовує хмарні технології, що дозволяє зберігати великі обсяги

даних без втрати продуктивності. Це особливо важливо для великих експериментів або коли є необхідність зберігати багато різних параметрів та результатів для подальшого аналізу.

Крім того, платформа забезпечує інтеграцію з іншими популярними інструментами для розробки та експериментування з нейронними мережами, такими як Weights & Biases, TensorBoard, MLflow. Це дозволяє користувачам поєднувати кілька платформ для досягнення максимальної гнучкості в роботі та забезпечити безперешкодний потік даних між різними інструментами.

Comet ML є потужним інструментом для дослідників, які займаються розробкою моделей глибокого навчання, оскільки вона дозволяє значно скоротити час, що витрачається на моніторинг, організацію та аналіз експериментів, а також покращити точність результатів завдяки здатності відстежувати і порівнювати великі обсяги даних.

У підсумку, Comet ML є незамінним інструментом для тих, хто працює в сфері машинного навчання, особливо для тих, хто займається розробкою та оптимізацією складних моделей, а також для команд, що потребують спільної роботи над проектами. Вона надає всі необхідні функції для безперебійного моніторингу та аналізу, дозволяючи значно покращити процес розробки і досягати вищих результатів.

3.8 Структура інформаційного забезпечення методу

Для реалізації методу знаходження та розпізнавання автомобільних номерів було обране відповідне програмне забезпечення. Як було розписано в другому розділі, метод складається з чотирьох етапів: підготовки датасету, навчання нейромережі, детекції та розпізнавання номерів автомобілів та тестування результатів.

На етапі підготовки датасету будуть використовуватись два інструменти для забезпечення якісного та точного навчання нейромережі. Roboflow стане основним

інструментом для збору та організації зображень. Ця платформа дозволяє завантажити зображення, створювати обмежувальні рамки, а також проводити попередню обробку даних. Попередня обробка включатиме такі дії, як зміна розміру зображень, їх нормалізація, а також аугментація для збільшення різноманітності датасету. OpenCV буде використовуватись для додаткового фільтрування, обробки та роботи з зображенням. Це дозволить підготувати зображення до навчання таким чином, щоб вони відповідали вимогам нейромережі та сприяли підвищенню точності моделі на етапі навчання.

На етапі навчання нейромережі буде використовуватись PyTorch, який надає широкі можливості для створення, налаштування та оптимізації моделей глибокого навчання. Для навчання нейромережі буде використана YOLOv11 – найновіша і найефективніша модель лінійки Yolo для задач детекції об'єктів.

Також на цьому етапі для моніторингу процесу навчання буде використовуватись Comet ML – потужний інструмент для відстеження експериментів і управління проектами машинного навчання.

На етапі детекції номерів та розпізнавання символів буде застосовуватись попередньо навчена модель YOLOv11, яка демонструє високу точність і швидкість у задачах виявлення об'єктів. Ця модель має ефективно знаходити автомобільні номери на зображеннях і відео. Після детекції номерного знака зображення передається до системи оптичного розпізнавання символів. У цьому процесі ключову роль відіграватиме застосунок Tesseract, який є одним із найпопулярніших і найефективніших інструментів для розпізнавання тексту на зображеннях. Схема інформаційної системи забезпечення методу зображена на рисунку 3.6.

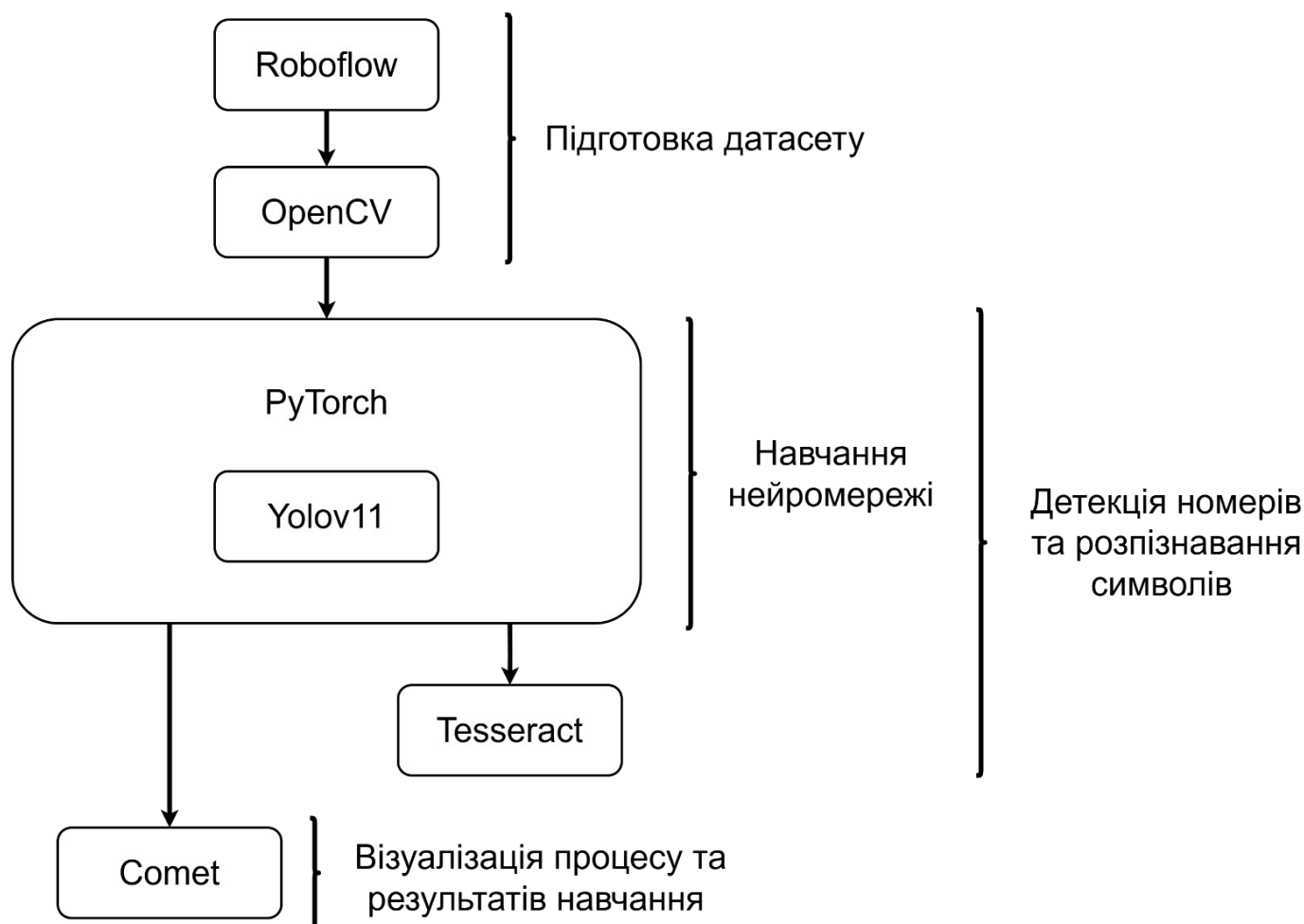


Рисунок 3.6 – Схема інформаційної системи забезпечення методу

Для забезпечення найкращої інтеграції з бібліотеками та технологіями, які використовуються в проєкті, буде вибраний Python 3.12.5. Це рішення базується на тому, що ця версія є офіційно підтримуваною для бібліотек та програм, які будуть використовуватись, згідно з рекомендаціями, наведеними на їхніх офіційних сторінках та в gitHub. Вибір цієї версії Python дозволить забезпечити сумісність з останніми оновленнями та функціоналом, що значно спростить інтеграцію та подальшу розробку проєкту.

Процес навчання нейромережі за допомогою YOLO, зображений на рисунку 3.7. Спочатку на вхід подаються фільтровані зображення, які вже пройшли попередню обробку. Ці зображення потрапляють до основних компонентів архітектури YOLO, які можна поділити на три головні частини: каркас, шия і голова.

Каркас відповідає за вилучення ознак із зображення – це глибока згорткова нейронна мережа, яка аналізує зображення і формує набір характеристик. Далі ці ознаки передаються до "шиї", яка об'єднує інформацію з різних рівнів ієрархії ознак і готує їх до остаточного передбачення. "Голова" моделі виконує фінальні операції детекції та ідентифікації об'єктів, формуючи відповідні бокси навколо об'єктів та визначаючи їх категорії.

На останніх етапах передбачення, отримані від "голови", проходять процеси загального об'єднання та вирівнювання, що забезпечує точність і стабільність виявлених об'єктів. Після цього модель зберігається у вигляді ваг і архітектури для подальшого використання, тестування чи розгортання.

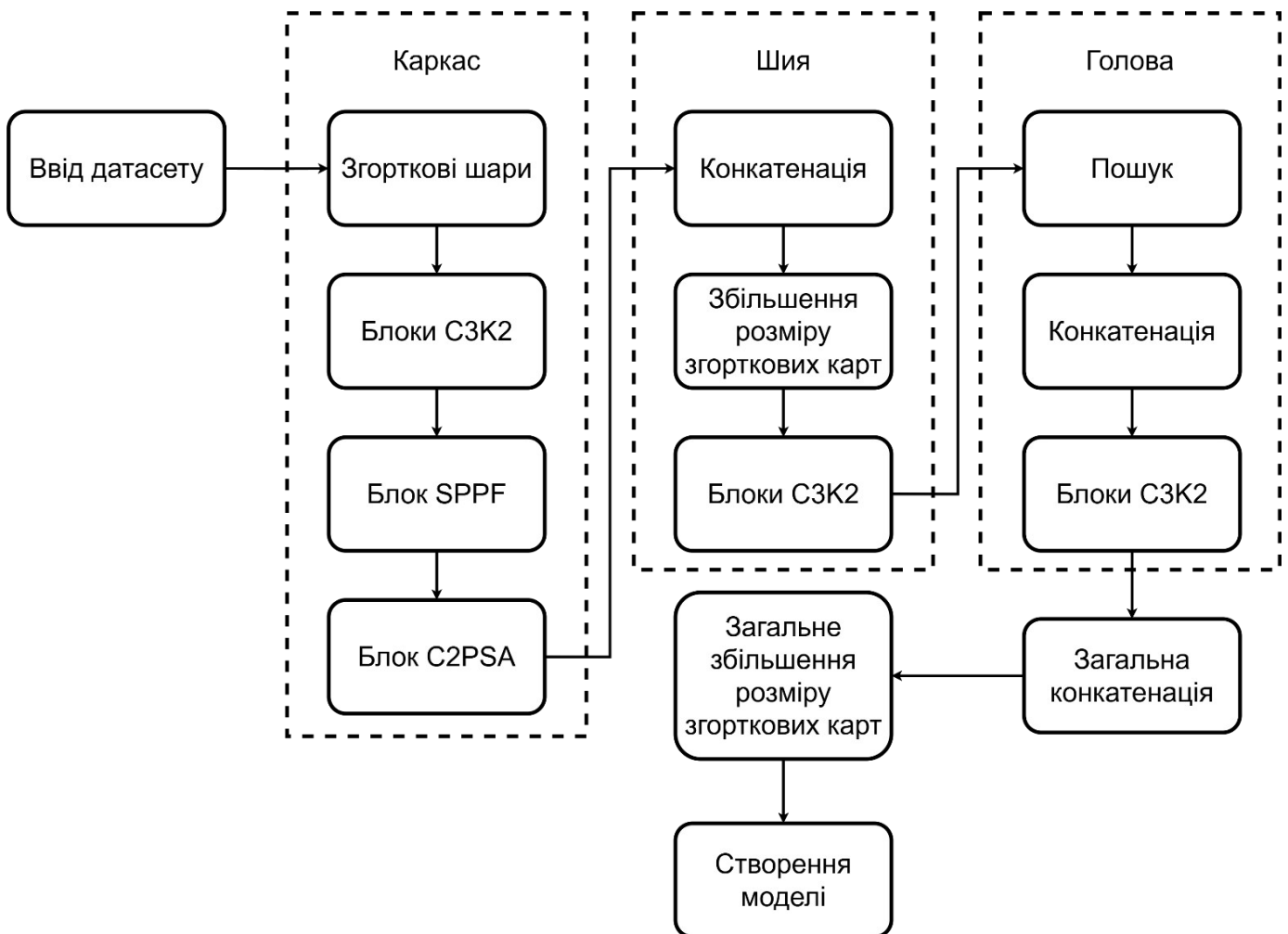


Рисунок 3.7 – Схема процесу навчання нейромережі

Процес знаходження та розпізнавання номерів автомобілей за допомогою попередньо навченої моделі YOLO та Tesseract, зображений на рисунку 3.8. Фільтроване зображення надходить на вхід попередньо навченої моделі YOLO, яка виконує задачу детекції об'єктів. YOLO аналізує зображення та визначає позиції номерних знаків, накладаючи обмежувальні рамки навколо знайдених номерів. Кожен знайдений номерний знак вирізається із зображення, створюючи окрему область інтересу. Це значно зменшує обсяг даних, що підлягають подальшому аналізу і прискорює роботу OCR-системи Tesseract.

На наступному кроці вирізана область проходить через процес бінаризації, тобто перетворення зображення в чорно-білий формат для кращого розпізнавання символів. Бінаризоване зображення піддається аналізу зв'язаних компонентів, що дозволяє визначити окремі об'єкти, які можуть бути символами. Далі відбувається детекція ліній і контурів, що допомагає локалізувати межі кожного символу на номерному знаку.

Після визначення позицій символів Tesseract виконує процес оптичного розпізнавання, ідентифікуючи окремі літери та цифри. Розпізнані символи об'єднуються у слова або послідовності символів, що відповідають номерному знаку автомобіля. Нарешті, отриманий результат обробки виводиться в консоль або передається для подальшої обробки, наприклад, для зберігання в базі даних або для використання в системах автоматичного контролю.

Цей підхід забезпечує ефективне та точне розпізнавання номерних знаків, оптимізуючи час обробки завдяки локалізації областей інтересу та використанню сучасних методів комп'ютерного зору та OCR.

Розглянутий підбір інструментів є оптимальним, оскільки забезпечує ефективне виконання всіх ключових етапів процесу: від підготовки даних і навчання нейромережі до детекції та розпізнавання номерних знаків. Такий підхід гарантує високу точність, продуктивність і стабільність роботи системи, що робить її надійною для використання в реальних умовах.

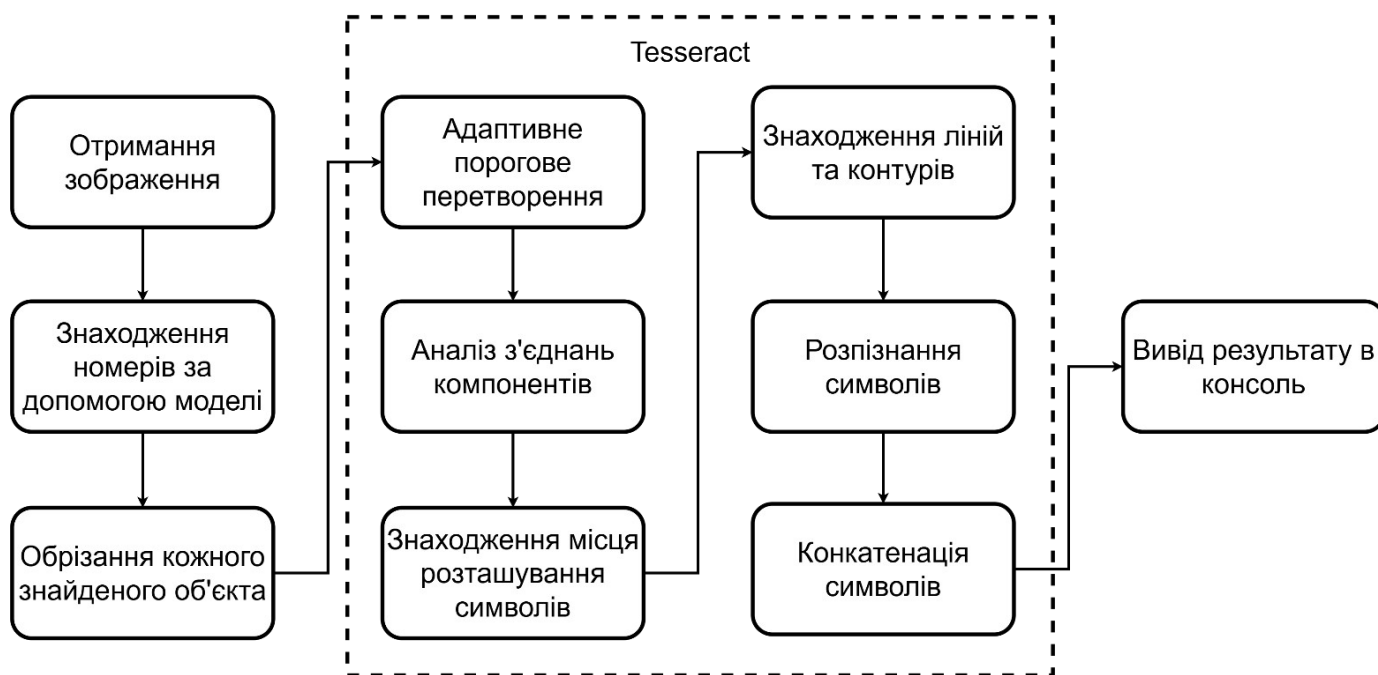


Рисунок 3.8 – Схема процесу знаходження та розпізнавання номерів автомобілей за допомогою попередньо навченої моделі YOLO та Tesseract

Висновки до розділу 3

В розділі була розроблена інформаційна система з інтеграцією методу знаходження та розпізнавання автомобільних номерів неймережею.

Процес навчання неймережі YOLO починається з отримання фільтрованих зображень, які проходять через три основні частини: каркас (для вилучення ознак), шию (для об'єднання інформації) та голову (для детекції об'єктів). Після цього відбувається об'єднання та вирівнювання передбачень, а модель зберігається для подальшого використання.

Процес виявлення та розпізнавання номерів починається з фільтрації зображення та подачі його на вхід YOLO для детекції номерних знаків. YOLO визначає їх позиції і накладає обмеження. Потім вирізані області передаються Tesseract для бінаризації, аналізу компонентів і детекції контурів. Tesseract розпізнає символи, формуючи номерні знаки для подальшого використання.

Для підготовки та попередньої обробки датасету використаний веб-додаток Roboflow. За допомогою бібліотеки OpenCV проводиться фільтрування та робота з

зображеннями. Щоб працювати з нейронними мережами була інтегрована та застосована бібліотека PyTorch. Для налаштування та навчання нейромережі застосовано YOLOv11. Щоб розпізнати символи, був використаний інструмент Tesseract OCR. Результат навчання нейромережі було візуалізовано за допомогою додатку Comet ML.

РОЗДІЛ 4 Експериментальне дослідження ефективності застосування методу виявлення та розпізнавання номерів

4.1 Підготовка датасетів для навчання

Для ефективного навчання нейромережі, що займається розпізнаванням номерних знаків автомобілів, критично важливим є використання великого і різноманітного датасету, що забезпечує максимальну універсальність моделі та її здатність працювати в різних умовах. Збір і підготовка таких даних включає кілька ключових аспектів, які суттєво впливають на точність і надійність моделі.

Зображення повинні бути високої якості, що дозволяє нейромережі точно виявляти деталі номерного знака, навіть якщо ці деталі мають дрібні дефекти або забруднення. Високоякісні зображення дозволяють зберігати чіткість контурів символів на номерних знаках, що є важливим для точного розпізнавання. Однак, лише використання зображень високої якості недостатньо для створення універсальної моделі. Зображення номерів у реальних умовах є необхідними для того, щоб нейромережа могла працювати в реальних ситуаціях. Важливо мати зображення, зроблені в різних погодних умовах і в різний час доби. Зображення вночі є особливо важливими, оскільки номерні знаки часто зображуються за умов поганого освітлення, а в таких ситуаціях нейромережа повинна правильно обробляти слабе освітлення та контраст. Зображення в дощ та туман додають ще один рівень складності, оскільки вода на склі чи розмиті об'єкти можуть сильно вплинути на чіткість зображення. Для цього нейромережа повинна вміти справлятися з такими природними факторами, як розмитість та часткові перешкоди на зображеннях. Підбір зображень для різноманітного датасету зображений на рисунку 4.1.

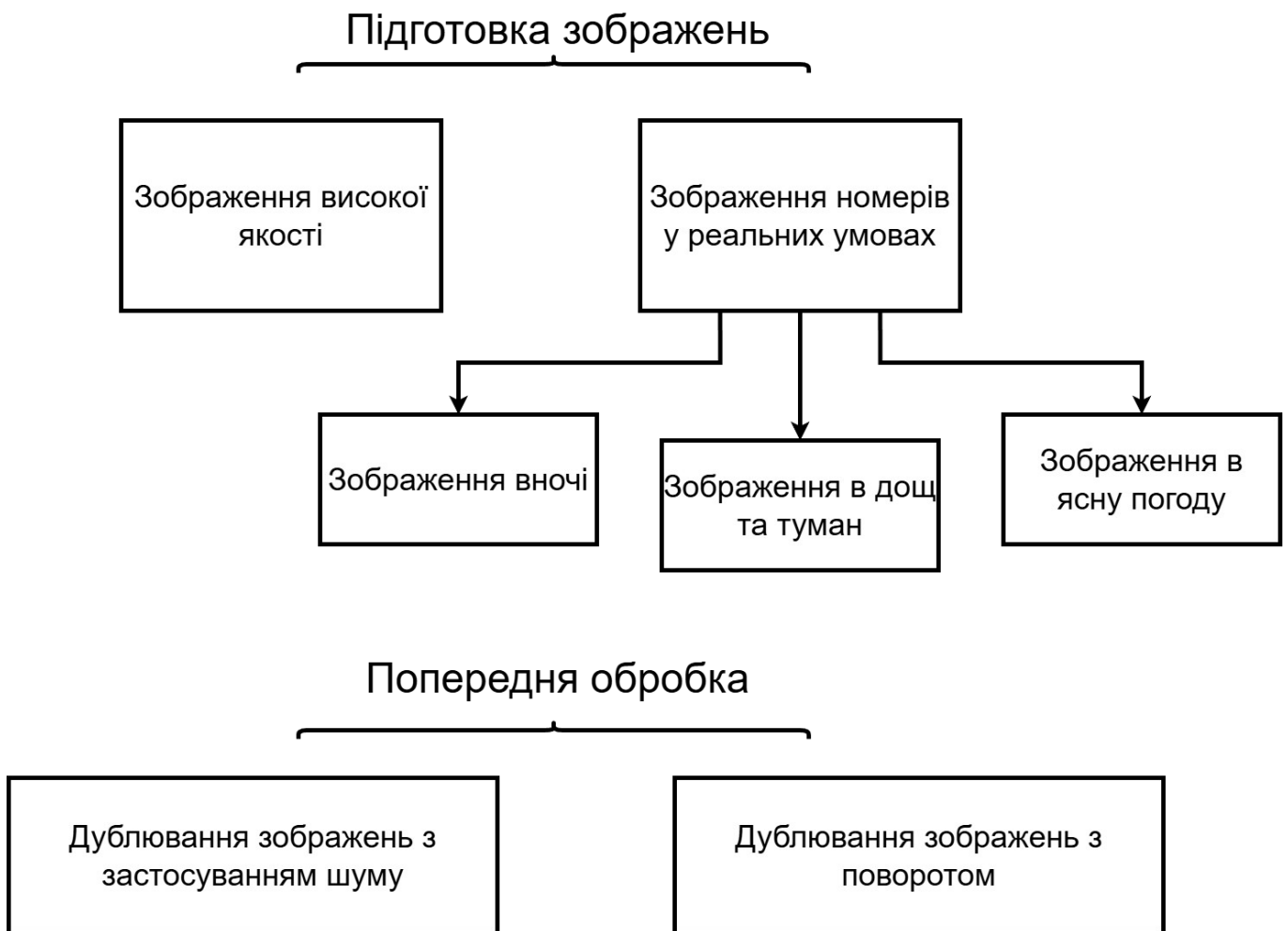


Рисунок 4.1 – Розбиття зображень датасету за основними критеріями

Крім того, необхідно забезпечити наявність зображень в ясну погоду, коли умови освітлення оптимальні, але в той же час можуть бути інші виклики, пов'язані з яскравим світлом і тінями, що можуть створювати певні спотворення на номерних знаках.

Попередня обробка зображень є важливою для підвищення ефективності навчання і розширення варіативності даних. Одним із методів, який допомагає моделі «навчитися» працювати з різними умовами, є дублювання зображень з застосуванням шуму. Додавання шуму дозволяє моделі навчитися обробляти зображення з дефектами або спотвореннями, які можуть виникати через різні фактори, як-то низька якість зображень або погані умови зйомки. Це важливо, оскільки номерні знаки можуть бути погано видно через пил або дефекти камери.

Крім того, дублювання зображень з поворотом дозволяє моделі навчитися обробляти зображення номерів з різними кутами нахилу. В реальних умовах номерні знаки можуть бути зняті з різних ракурсів, і без цього етапу підготовки модель може мати труднощі з розпізнаванням знаків, якщо камера була повернута або нахилена.

Різноманітний датасет, зображений на рисунку 4.1, дозволяє нейромережі навчатися в широкому діапазоні можливих ситуацій. Це підвищує її здатність до узагальнення, що в свою чергу забезпечує високу точність розпізнавання номерних знаків у реальних умовах, незалежно від того, чи знято зображення вночі або вдень, в дощ чи ясну погоду, з різними кутами нахилу камери або з додатковими спотвореннями.

В данній роботі буде проводитись тестування навченої нейромережі за допомогою середнього та малого датасету. У першому випадку зображення будуть поділені на чотири класи при одному тестуванні та об'єднанні в один клас при іншому тестуванні. У другому випадку невелика кількість зображень одного класу будуть навчатись протягом 15, 30 та 50 епох.

Для навчання нейромережі на середньому датасеті було підготовлено 1591 зображення з 2293 анотаціями, що зображено на рисунку 4.2. Середній розмір зображення 460 на 460, середній розмір зображення 0.17 мп.

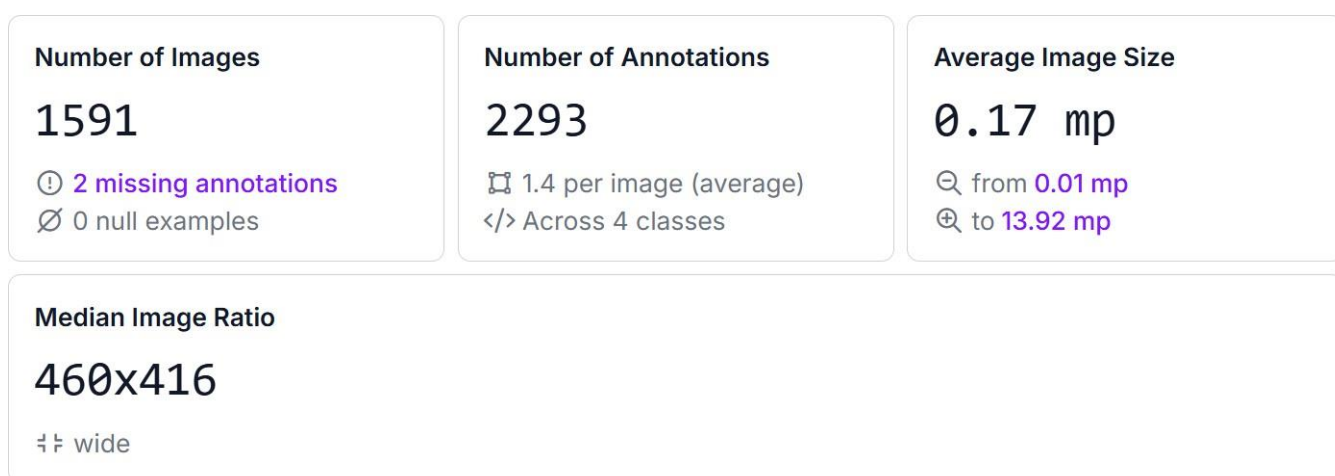


Рисунок 4.2 – Аналітика за кількістю зображень датасету

Зображення були розділені за наступними класами:

- номери машин (Car plates);
- номери машин України (Car plates Ukraine);
- номери машин ЄС (Car plates EU);
- номери машин США (Car plates USA).

Розбивка анотацій за вказаними класами зображена на рисунку 4.3.

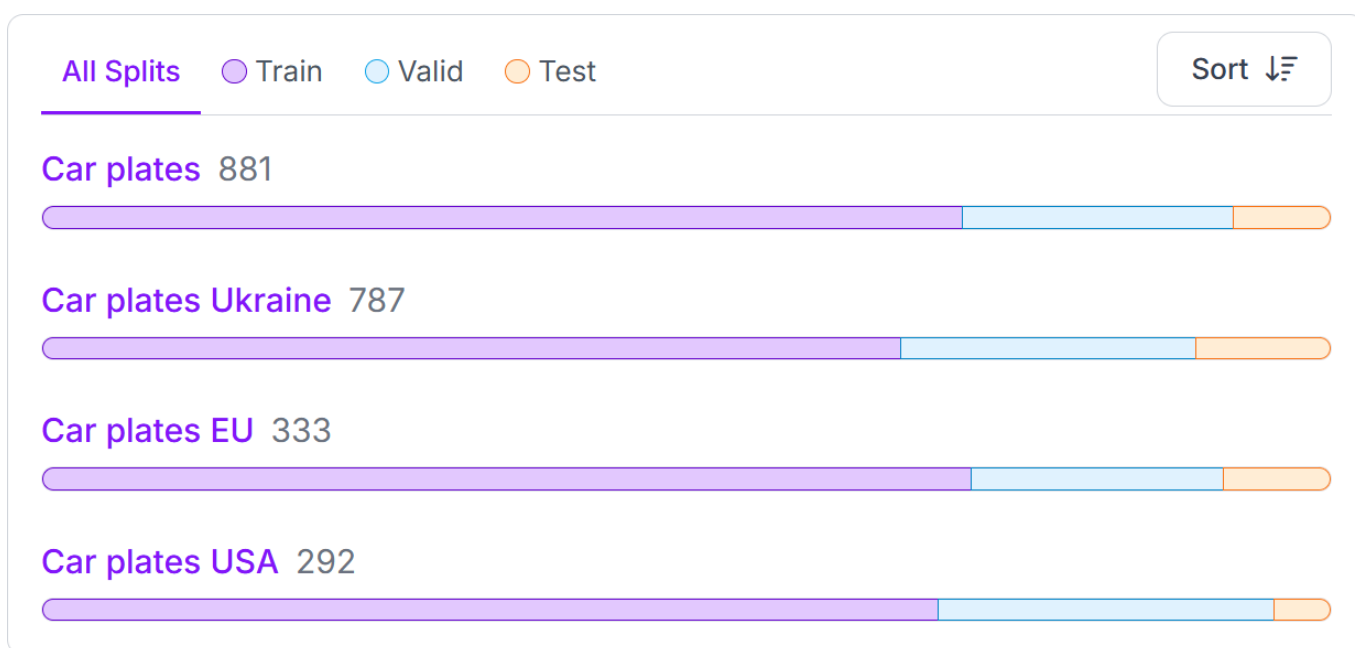


Рисунок 4.3 – Кількість анотацій кожного класу

При іншому випадку, зображення були об'єднані в один клас, що зображено на рисунку 4.4.

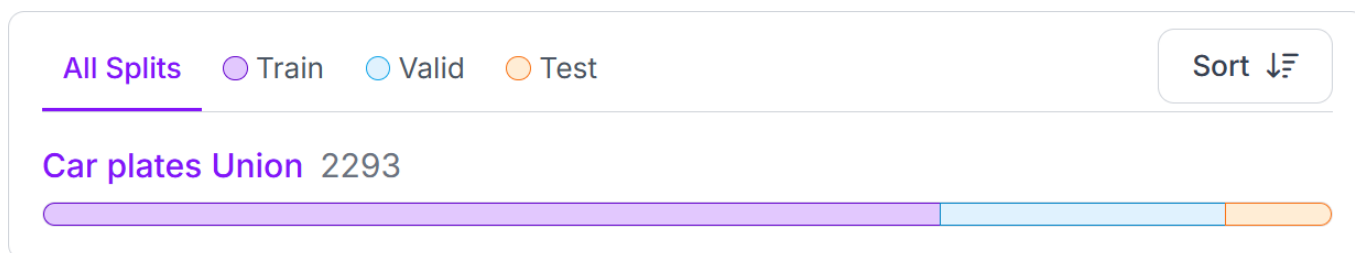


Рисунок 4.4 – Кількість анотацій об'єднаного класу

Статистика розміру зображень, представлена на рисунку 4.5, надає важливу інформацію щодо характеристик даних, з якими працює неймережа. З цієї інформації можна зробити висновок, що неймережа повинна бути оптимізована для обробки зображень, що мають різні розміри, зокрема великі та середні. Це означає, що під час навчання модель повинна мати здатність ефективно справлятися з варіаціями у розмірах об'єктів на зображеннях, а також правильно визначати важливі деталі як на великих зображеннях, так і на зображеннях середнього розміру.

Зазвичай, чим більший розмір зображення, тим більше інформації про об'єкти та їх контекст можна отримати, але це також збільшує вимоги до обчислювальних ресурсів і часу на тренування. У той же час, зображення середнього розміру можуть бути більш компактними, що дозволяє ефективніше використовувати ресурси, але при цьому важливо, щоб модель не втрачала здатність до детекції навіть малих або віддалених об'єктів на таких зображеннях.

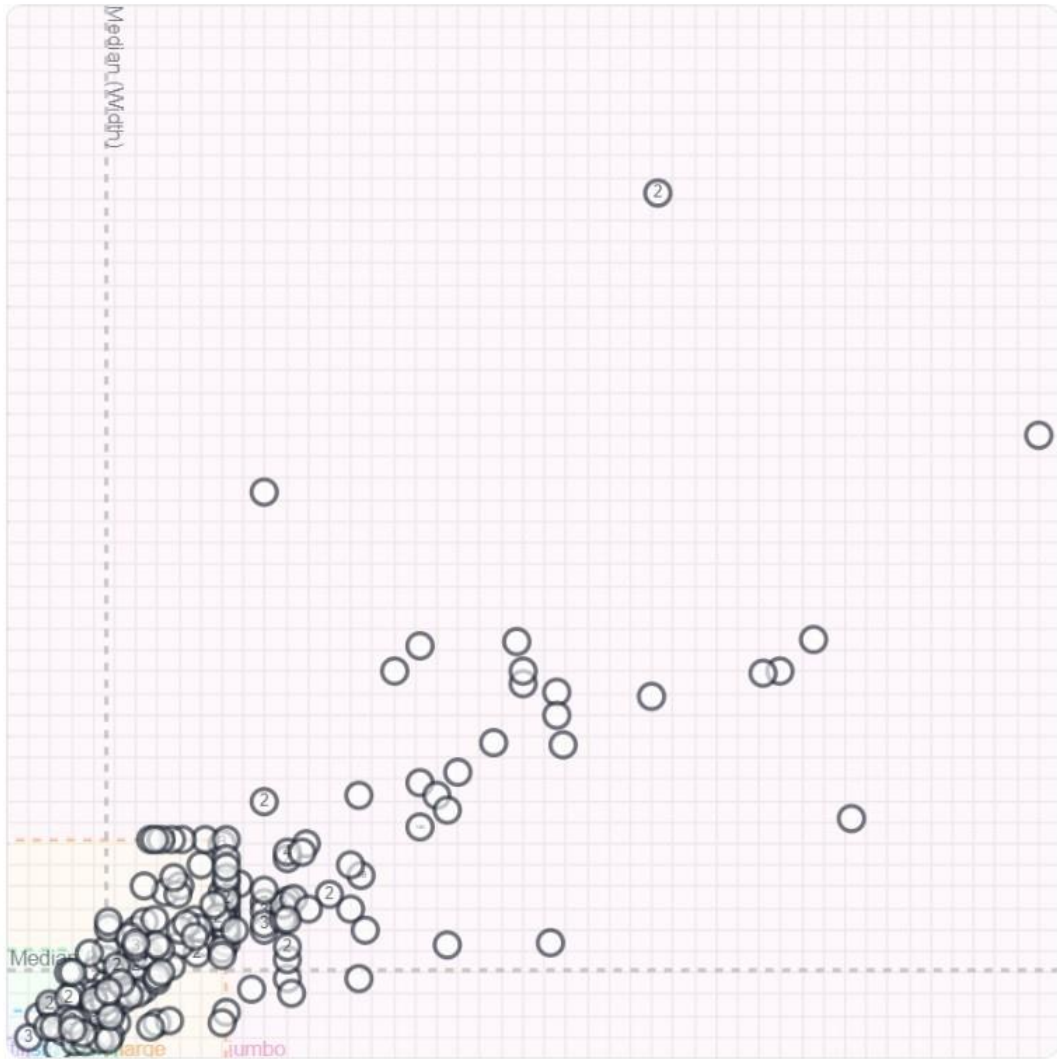


Рисунок 4.5 – Статистика розміру та пропорцій зображень

Зображення були попередньо оброблені. Як показано на зображенні 4.6, спочатку було зроблено налаштування контрастності, за допомогою використання адаптивного вирівнювання. Після цього деякі зображення були дубльовані та обернуті на двадцять градусів, та зсунуті по горизонталі та вертикалі на вісімнадцять градусів. Також були додані дубльовані зображення з шумом. Всього ми отримали 3829 зображень.

3829 Total Images

[View All Images →](#)



Dataset Split

TRAIN SET 88%

3357 Images

VALID SET 9%

356 Images

TEST SET 3%

116 Images

Preprocessing

Auto-Adjust Contrast: Using Adaptive Equalization

Augmentations

Outputs per training example: 3

Rotation: Between -20° and $+20^\circ$

Bounding Box: Shear: $\pm 18^\circ$ Horizontal, $\pm 18^\circ$ Vertical

Рисунок 4.6 – Отриманий датасет після попередньої обробки

Наступним кроком, був підібраний підготовлений датасет з малою кількістю зображень, що зображено на рисунку 4.7. Середній розмір зображення 395 на 395, середній розмір зображення 0.41 мп.

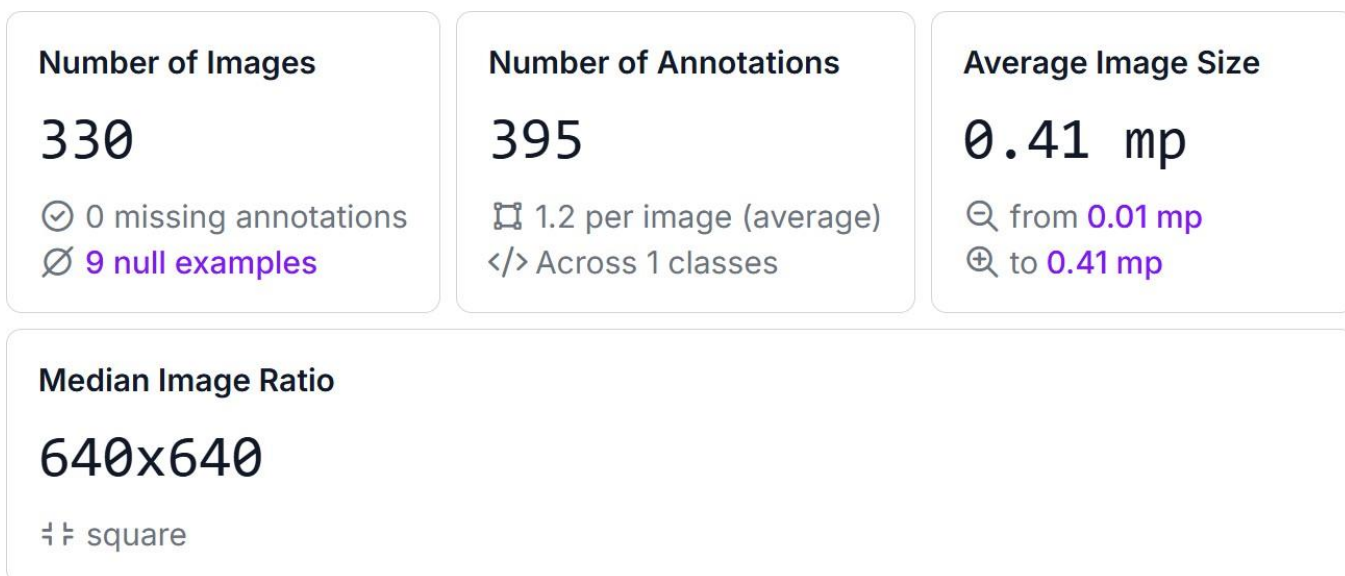


Рисунок 4.7 – Аналітика за кількістю зображень датасету

Розбивка за класами зображена на рисунку 4.8.

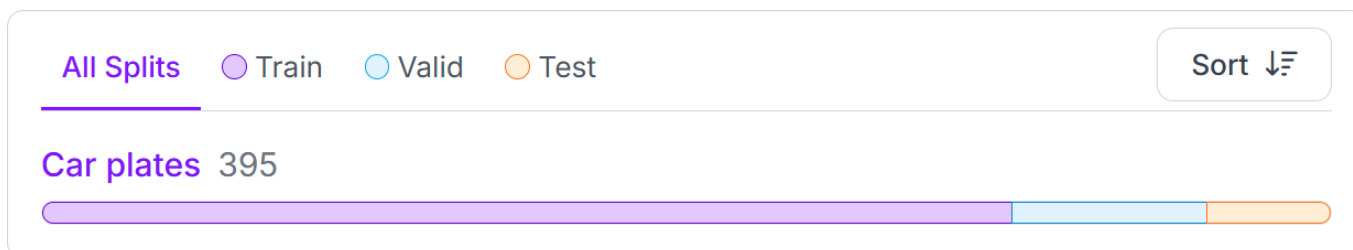


Рисунок 4.8 – Статистика розміру та пропорцій зображень

Статистика розміру зображень, представлена на рисунку 4.9, показує, що не представлені крихітні та великі зображення, тому пошук таких об'єктів, дасть гірший результат.



Рисунок 4.9 – Статистика розміру та пропорцій зображень

До зображень не застосовувалось дублювання з шумом та поворотом, тому збережена версія налічує 330 зображень, що зображено на рисунку 4.10.

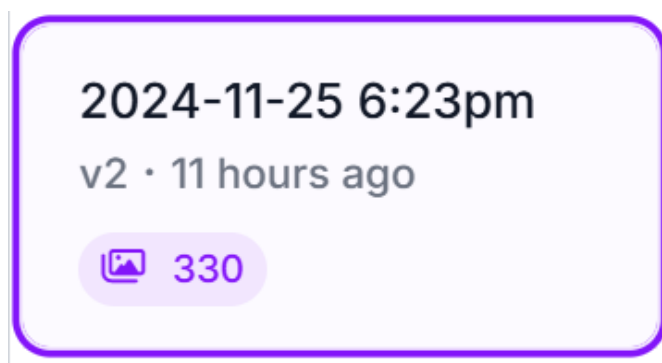


Рисунок 4.10 – Збережена версія малого датасета

4.2 Метрики тестування нейромережі

Метрики в тестуванні нейронних мереж є основним інструментом для оцінки їхньої ефективності та продуктивності. Вони дозволяють кількісно виміряти, наскільки добре модель виконує завдання, і надають важливу інформацію для подальшої оптимізації.

Вони дозволяють не лише оцінити загальну якість роботи моделі, а й виявити можливі проблеми, такі як перенавчання або недонавчання, а також визначити, чи добре модель справляється з задачами розпізнавання класів. Вони допомагають приймати обґрунтовані рішення щодо подальших кроків, таких як налаштування гіперпараметрів, вибір архітектури або модифікація навчальної вибірки. Важливість метрик полягає також у тому, що без них тестування нейронної мережі було б суб'єктивним і неефективним, а тільки за допомогою точних кількісних показників можна об'єктивно порівнювати різні моделі та методи.

Для тестування нейромережі будуть застосовуватись наступні основні метрики: прецензійність (precision), повнота (recall), F1-міра.

Прецензійність – це метрика, яка вимірює, скільки з передбачених класів є правильними й обчислюється наступним чином:

$$\text{Прецензійність} = \frac{TP}{TP + FP}, \quad (4.1)$$

де TP – правильно передбачений існуючий клас, FP – хибно передбачений існуючий клас

Повнота – це метрика, яка вимірює, скільки з усіх реальних існуючих класів було правильно виявлено моделлю. Це також називається чутливістю й обчислюється наступним чином:

$$\text{Повнота} = \frac{TP}{TP + FN}, \quad (4.2)$$

де FN – неправильно передбачений відсутній клас.

F1-міра – це метрика, яка використовується для оцінки ефективності моделей машинного навчання, де важливо враховувати баланс між точністю та повнотою й обчислюється наступним чином:

$$F1 = \frac{2 * \text{Прецензійність} * \text{Повнота}}{\text{Прецензійність} + \text{повнота}}, \quad (4.3)$$

Додатково буде використовуватись метрика втрат (loss) – яка є ключовою у Yolo, та дозволяє коригувати передбачення координат і класів, налаштовувати модель для правильного оцінювання ймовірності наявності об'єкта в кожній клітинці сітки.

Загальна формула втрат у Yolo обчислюється наступним чином:

$$X = X_{coord} + X_{obj} + X_{cls} + X_{noobj}, \quad (4.4)$$

де X – втрати, X_{coord} – втрати по координатах (реального прямокутника), X_{obj} – втрати за наявність об'єкта (objectness loss), X_{cls} – втрати за класифікацію (class loss), X_{noobj} – втрати “хибного передбачення” (no object loss)

Також буде використовуватись метрика mAP, яка розширює середню точність, обраховуючи середні значення по кількох класах об'єктів. Ця метрика дає уявлення про продуктивність моделі у виявленні об'єктів.

mAP50 – середня точність, розрахована при порозі IoU (що вимірює схожість між передбачуваними та реальними обмежувальними рамками), рівним 0,5. Це показник точності моделі, враховуючи тільки "легкі" виявлення.

mAP50-95 – значення середньої точності, розраховане при різних порогових значеннях IoU, що коливаються від 0,50 до 0,95. Воно дає повне представлення про якість моделей на різних рівнях складності виявлення.

Аналізуючи навчену модель за допомогою таких метрик, як влучність, повнота, F1-міра, втрати та mAP50, mAP50-95, можна отримати комплексне уявлення про її ефективність у задачах детекції об'єктів.

4.3 Налаштування параметрів нейромережі

Добре налаштування оптимізатора та параметрів нейромережі є важливою для ефективного навчання нейромережі, оскільки від нього залежить, як швидко і точно модель адаптується до вхідних даних.

Оптимізатор, для створення моделі, налаштований за допомогою AdamW – вдосконаленої версії Adam, який використовує адаптивні моменти для кожного параметра, що дозволяє ефективніше коригувати ваги в процесі навчання. Вибраний оптимізатор розв'язує одну з проблем класичного рішення – не зовсім коректне застосування вагового спаду (weight decay). У AdamW процес регуляризації та корекція параметрів за допомогою адаптивних моментів (перших і других моментів) відбуваються незалежно один від одного.

Цей оптимізатор добре підходить для навчання на середніх датасетах, зокрема для набору даних з 3900 зображень, як у нашому випадку. Завдяки своїй адаптивності та здатності коригувати швидкість навчання для кожного параметра окремо, AdamW знижує необхідність в ретельному налаштуванні гіперпараметрів, таких як швидкість навчання, і робить процес навчання більш стабільним. Це особливо корисно при роботі з відносно невеликими датасетами, де модель може швидше схильна до перенавчання, а регуляризація через ваговий спад допомагає покращити здатність моделі до узагальнення.

Для невеликих наборів зображень, AdamW забезпечує баланс між швидкістю навчання та стабільністю, дозволяючи моделі досягти хороших результатів навіть

при обмежених ресурсах. Це робить його популярним вибором для багатьох задач комп'ютерного зору, включаючи детекцію об'єктів та класифікацію.

Для навчання використовується початкова швидкість навчання (learning rate) 0.0001 – досить низьке значення, що дозволяє моделі поступово коригувати свої параметри, запобігаючи надмірним коливанням під час навчання. Параметр `weight_decay` встановлений на значення 0.0005, що додає до функції втрат штраф за великі значення ваг, сприяючи покращенню узагальнення моделі і зменшенню ймовірності перенавчання.

Модель навчається протягом 15, 30, 50 епох, що є достатнім для стабільного навчання на зображеннях датасету розміру 640 пікселів по довгій стороні. Використання цих налаштувань дозволяє збалансувати швидкість навчання і ефективність оптимізації, забезпечуючи стабільний процес навчання та мінімізацію втрат.

4.4 Порівняння результатів навчання

Під час тестування нейромережі, першим кроком відбулось навчання моделі на середньому датасеті з чотирма класами протягом 15, 30, 50 епох, другим кроком відбулось навчання, на середньому датасеті з одним класом на 30 епохах, третім кроком відбулось навчання моделі на малому датасеті протягом 15, 30, 50 епох. В кінці кожного кроку був проведений аналіз графіків та метрик. Четвертим кроком відбулось загальне порівняння моделей, та вибір оптимальної.

Почнімо з першого кроку. На графіку, зображеному на рисунку 4.11, можна побачити ефективність процесу навчання моделі на середньому датасеті, з чотирма класами протягом 15, 30, 50 епох.

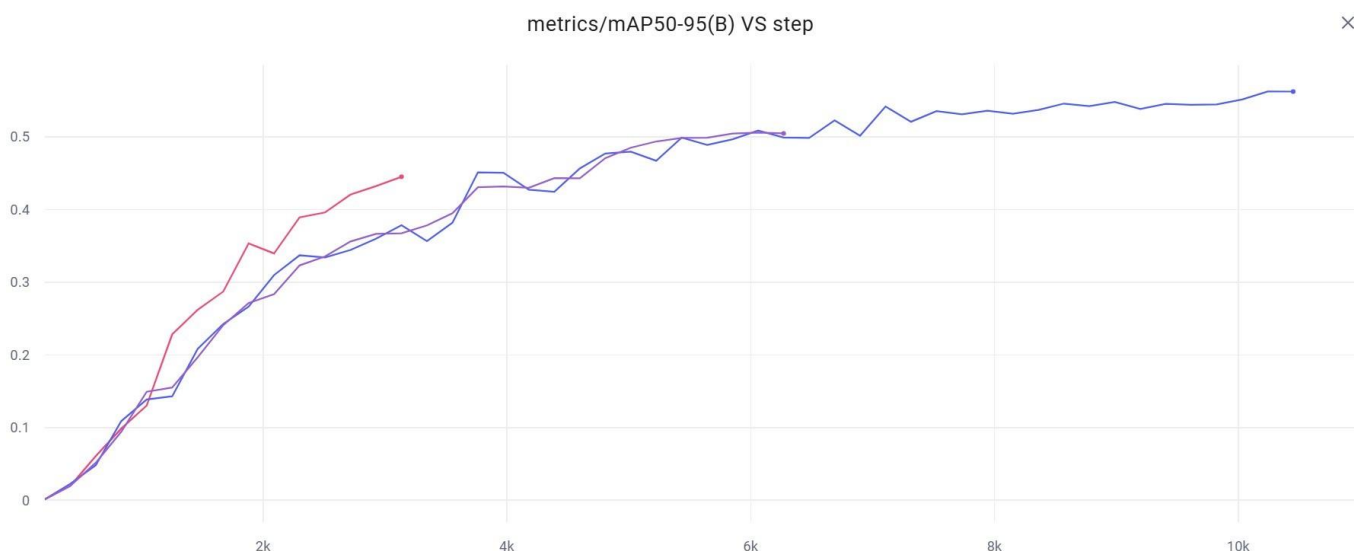


Рисунок 4.11 – Графік “mAP50-95 на крок” під час навчання нейромережі

Спостерігається швидке зростання mAP на початкових етапах (до 2000 кроків), що свідчить про ефективне навчання на початку. Після досягнення 4000-5000 кроків модель виходить на плато – приріст точності незначний, але стабільний. Після 6000 кроків mAP залишається майже незмінним, що може свідчити про досягнення межі продуктивності даної моделі на конкретному датасеті. Оптимальною моделлю на цьому кроці можна вважати ту, яка навчена на тридцяти епохах.

Другим кроком було проведено навчання на середньому датасеті протягом оптимальних 30 епох, без розділу на різні класи. Це зроблено для порівняння ефективності підходів до формування датасету.

На рисунку 4.12 зображено порівняння “mAP50-95” на крок двох моделей, навчених на середньому по величині датасету. В одному випадку датасет ділився на чотири класи, в іншому, всі зображення належали одному класу.

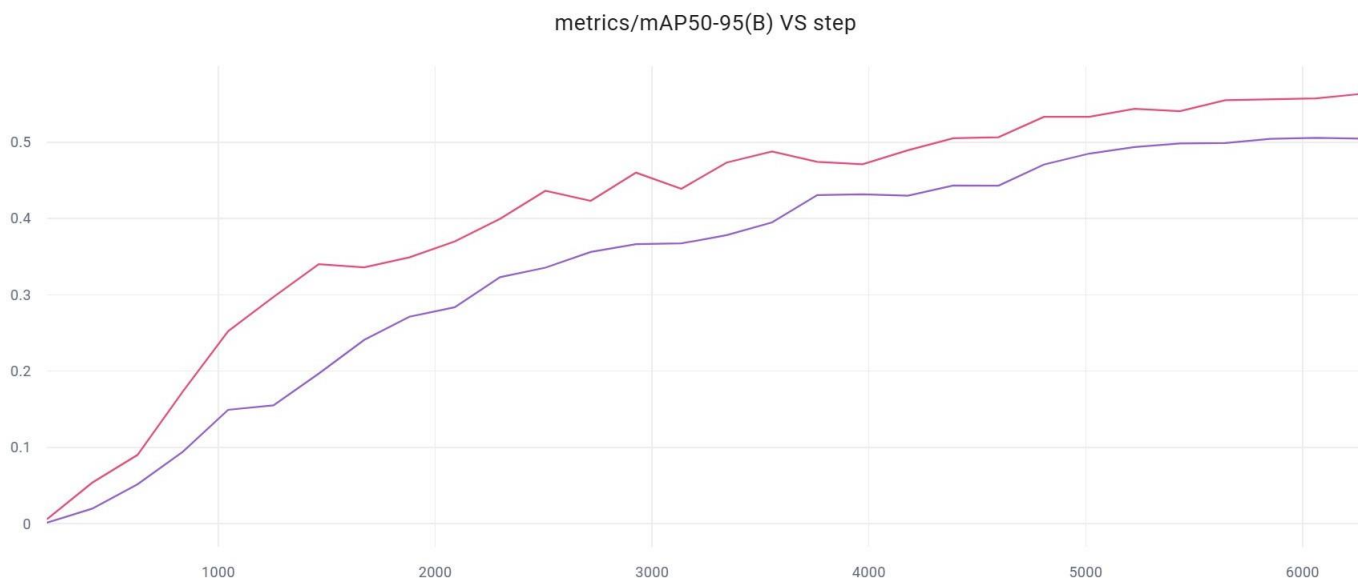


Рисунок 4.12 – Графік “mAP50-95 на крок

Обидві моделі показують схожий приріст продуктивності на початкових етапах. Однак модель, навчена на 4 класах, демонструє повільніший темп зростання mAP у порівнянні з моделлю, навченою на об'єднаному класі. На середніх кроках модель з об'єднаними класами має стабільно вищі показники точності порівняно з моделлю на чотирьох класах. Це може свідчити про те, що об'єднана модель краще справляється з узагальненням даних і швидше навчається розпізнавати основні характеристики об'єктів. На фінальних етапах різниця між моделями зберігається, причому об'єднана модель досягає вищого mAP на останніх етапах тренування.

Порівняння моделей, навчених на середньому датасеті за вибраними метриками представлено в таблиці 4.1.

Таблиця 4.1 – Метрики моделей навчених на середньому датасеті

	Сер. датасет, 4 кл			с.датасет 1 кл
Кількість епох	15	30	50	30
Прецензійність	0.642	0.6973	0.724	0.921
Повнота	0.598	0.6687	0.725	0.743
F1-міра	0.619	0.682	0.724	0.822
Втрати	58.953	58.953	49.524	48.267
mAP50	0.654	0.738	0.778	0.829
mAP50-95	0.445	0,504	0.562	0.563

Оптимальною, можна вважати середню модель, з одним класом, навчену на тридцяти епохах.

Після навчання моделей на малому датасеті проведемо їх порівняння. На графіку, зображеному на рисунку 4.13, можна побачити ефективність процесу навчання.

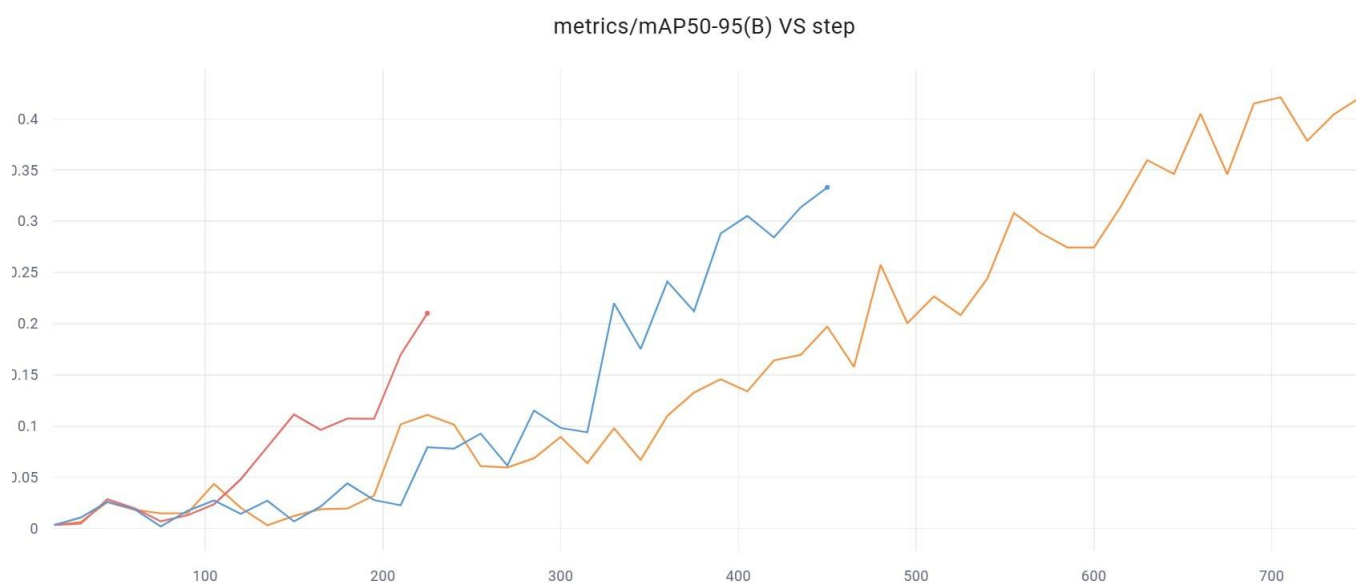


Рисунок 4.13 – Графік “mAP50-95 на крок” під час навчання нейромережі

На початкових етапах навчання всі моделі показують схожу результативність, проте згодом їх зростання стає більш стрімким. Модель, навчена на 50 епохах, демонструє найвищий темп поліпшення показників. Наприкінці навчання не спостерігається значного спаду темпу, що свідчить про доцільність продовження навчання на більшій кількості епох.

Проведемо порівняння моделей за вибраними метриками, які представлені в таблиці 4.2.

Таблиця 4.2 – Метрики моделей навчених на малому датасеті

	Малий датасет		
Кількість епох	15	30	50
Прецензійність	0.671	0.925	0.8
Повнота	0.3	0.483	0.6
F1-міра	0.41	0.634	0.68
Втрати	154.73 4	102.67 3	75.799
mAP50	0.368	0.613	0.672
mAP50-95	0.210	0.333	0.420

Останнім кроком проведемо загальне порівняння моделей. На рисунку 4.11 зображено порівняння за mAP 50-95 на крок.

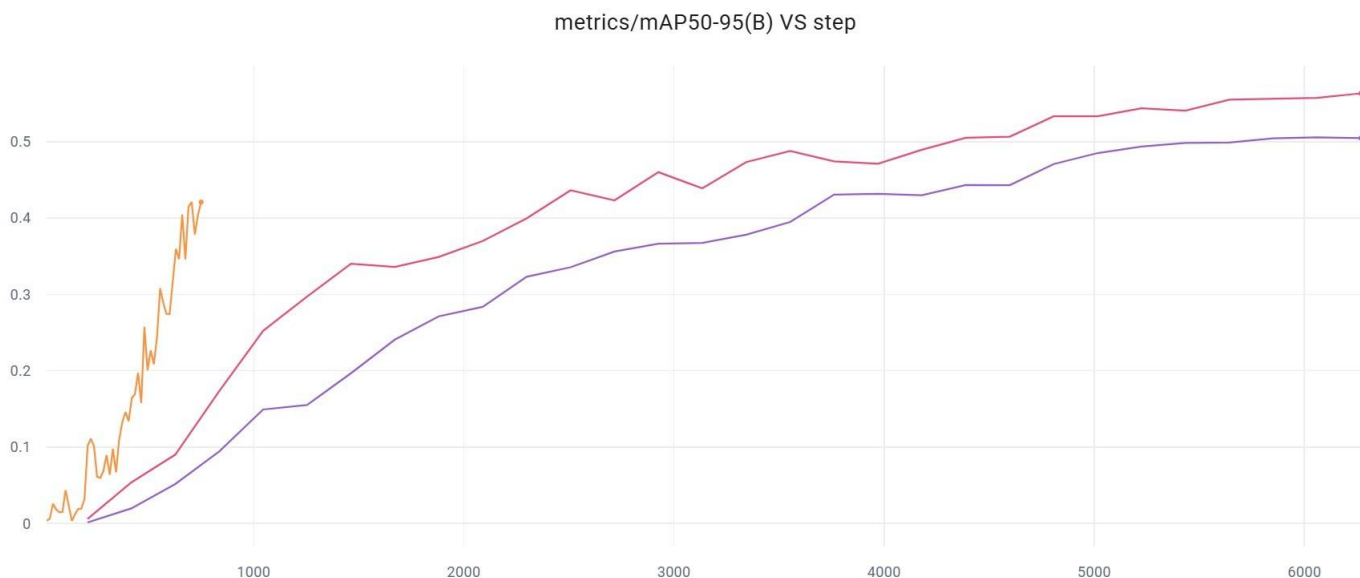


Рисунок 4.14 – Графік “mAP50-95 на крок” під час навчання нейромережі

На графіку спостерігається значна різниця між моделями, навченими на середньому та малому датасетах. У випадку малого датасету видно недостатню кількість кроків для досягнення стабільної роботи системи. Отже, оптимальною можна вважати модель, навчену на середньому датасеті. Схожі показники можна побачити на графіку mAP50 на крок.

Те наскільки добре моделі визначають координати рамки, яка обмежує об'єкт на зображенні. зображено на рисунку 4.15.

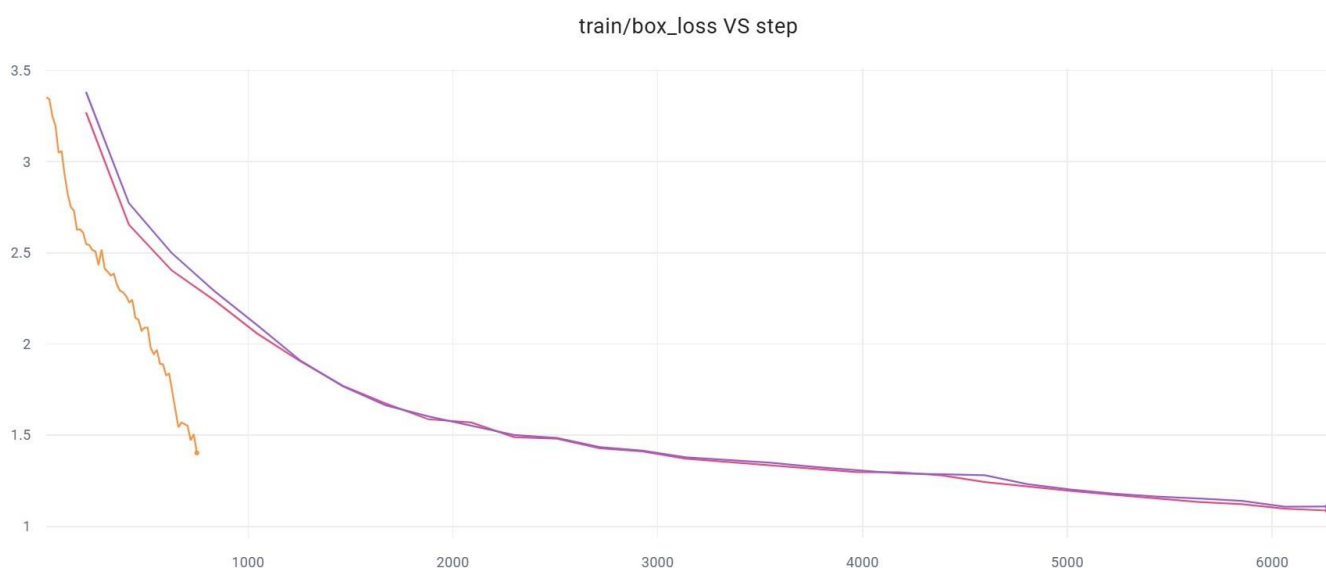


Рисунок 4.15 – Графік втрати “координат” границі на крок

Значне зниження показників означає, що моделі вже на ранніх етапах навчання почали добре визначати об'єкти. Після зниження до одиниці, втрати не збільшуються та залишаються стабільними, це свідчить про успішне навчання.

Данні представлені в таблиці 4.3.

Таблиця 4.3 – Метрики навчених моделей

	Сер. датасет, 4 кл			с.датасе т 1 кл	Малий датасет		
	15	30	50	30	15	30	50
Кількість епох	15	30	50	30	15	30	50
Прецензійність	0.642	0.6973	0.724	0.921	0.671	0.925	0.8
Повнота	0.598	0.6687	0.725	0.743	0.3	0.483	0.6
F1-міра	0.619	0.682	0.724	0.822	0.41	0.634	0.68
Втрати	58.95 3	58.953	49.524	48.267	154.73 4	102.67 3	75.799
mAP50	0.654	0.738	0.778	0.829	0.368	0.613	0.672
mAP50-95	0.445	0,504	0.562	0.563	0.210	0.333	0.420

Порівняння трьох різних датасетів – середнього з 4 класами, середнього з 1 класом та малого – виявляє відмінності в продуктивності моделей залежно від розміру і складності даних, а також кількості епох навчання.

Середній датасет з 4 класами демонструє стабільне покращення продуктивності зі збільшенням кількості епох. Прецензійність підвищується з 0.642 до 0.724, а повнота – з 0.598 до 0.725. F1-міра, що відображає баланс між цими метриками, також зростає до 0.724 після 50 епох. Втрати знижуються з 58.953 до 49.524, що свідчить про ефективне навчання. mAP50 значно покращується (з 0.654 до 0.778), тоді як mAP50-95 зростає помірно, досягаючи 0.562. Це вказує на те, що

модель ефективно справляється зі складними завданнями детекції, особливо при достатній кількості епох.

Середній датасет з 1 класом має найвищу прецензійність, що досягає 0.921 на 30 епохах, але після 50 епох вона знижується до 0.8, що може вказувати на перенавчання. Повнота досягає максимуму на 30 епохах (0.743), а F1-міра – 0.822. Втрати знижуються значно (від 154.734 до 48.267), що показує стабілізацію навчання. mAP50 досягає найвищого значення (0.829) на 30 епохах, після чого покращення стає мінімальним. Це свідчить про те, що модель швидко навчається на менш складних даних, але подальше навчання може призвести до деградації.

Малий датасет демонструє нестабільні результати через обмежену кількість даних. Прецензійність зростає до 0.925 на 30 епохах, але знижується до 0.8 на 50 епохах. Повнота починається з дуже низького значення (0.3), але поступово зростає до 0.6. F1-міра також зростає, досягаючи максимуму 0.68 на 50 епохах. Втрати залишаються високими порівняно із середніми датасетами, хоча і знижуються зі 154.734 до 75.799. mAP50 та mAP50-95 демонструють поступове покращення, досягаючи 0.672 та 0.420 відповідно. Цей датасет вимагає більше епох для досягнення прийнятних результатів, але через обмеженість даних перенавчання може настати швидше.

Таким чином, навчання на середніх датасетах забезпечує більш стабільні та високі результати, тоді як малий датасет потребує ретельного налаштування кількості епох для уникнення перенавчання і досягнення збалансованої продуктивності.

Як показано на рисунку 4.16, модель ефективно виконує розпізнавання автомобільних номерів різних типів і форматів. Вона точно визначає місцезнаходження номерних знаків на зображеннях, що підтверджує її здатність обробляти варіативні дані. Завдяки високій точності локалізації та розпізнавання, модель демонструє стабільну продуктивність, що є важливим показником її ефективності для практичного застосування у системах автоматичного контролю транспортних засобів.



Рисунок 4.16 – Результат розпізнавання номерів

Тестовий приклад зображення наведено на рисунку 4.17.



Рисунок 4.17 – Тестове зображення

Результат роботи розпізнавання символів зображено на рисунку 4.18.

```
PS D:\mkryava> & d:/mkryava/venv/Scripts/python.exe d:/mkryava/Sand1.py  
Розпізнаний текст на номері №2 'AA3349MM'  
Розпізнаний текст на номері №1 'KA0918EA'
```

Рисунок 4.18 – Результат розпізнання символів

Інформаційна система успішно виконала пошук та розпізнавання автомобільних номерів на зображенні. Спочатку були виявлені номерні знаки, після цього відбулось коректне розпізнавання символів.

Висновки до розділу 4

В четвертому розділі були підготовлені датасети на середній та малій кількості зображень, вибрані метрики тестування нейромережі, проведено налаштування оптимізатора, та експериментальне тестування ефективності нейромережі.

Для створення датасету були зібрані зображення номерів у високій якості та у реальних умовах: вночі, в дощ та туман, в ясну погоду. Відбулась попередня обробка, в результаті якої дублювались зображення з застосуванням шуму та поворотом за різними осями. Для середнього датасету ми отримали 3829 зображень, для малого – 330. Для тестування нейромережі будуть застосовуватись наступні основні метрики: прецензійність, повнота, F1-міра, втрати, mAP50, mAP50-95. Оптимізатор налаштований за допомогою AdamW. Для навчання використовується початкова швидкість навчання (learning rate) – 0.0001. Параметр weight_decay встановлений на значення 0.0005.

Під час тестування нейромережі, першим кроком відбулось навчання моделі на середньому датасеті з чотирьма класами протягом 15, 30, 50 епох, другим кроком

відбулось навчання, на середньому датасеті з одним класом на 30 епохах, третім кроком відбулось навчання моделі на малому датасеті протягом 15, 30, 50 епох. В кінці кожного кроку був проведений аналіз графіків та метрик. Четвертим кроком відбулось загальне порівняння моделей, та вибір оптимальної.

Середній датасет з 1 класом демонструє найвищу прецензійність, що досягає 0.921 на 30 епохах, але після 50 епох вона знижується до 0.8, що може свідчити про перенавчання. Повнота досягає максимуму на 30 епохах (0.743), а F1-міра – 0.822. Втрати знижуються значно (від 154.734 до 48.267), що вказує на стабілізацію навчання. mAP50 досягає найвищого значення (0.829) на 30 епохах, після чого покращення стає мінімальним, що свідчить про швидке навчання на менш складних даних, але можливу деградацію при подальшому навчанні. Навчання на середніх датасетах дає більш стабільні та високі результати, тоді як малий датасет вимагає ретельного налаштування кількості епох для уникнення перенавчання та досягнення оптимальної продуктивності. Модель, тренувана на одному класі середнього датасету, виявилася найкращою для використання в реальних умовах.

Загальні висновки

В результаті виконання кваліфікаційної роботи магістра створено метод виявлення та розпізнавання номерів автомобілів нейромережевими засобами

В процесі створення методу були виконані такі задачі дослідження:

– проведений аналіз існуючих методів та підходів до автоматизованого виявлення і розпізнавання номерних знаків автомобілів, були визначені їх переваги та недоліки для обґрунтування напрямів удосконалення;

– розроблений метод автоматизованого виявлення та розпізнавання номерних знаків на основі модифікованої архітектури YOLOv11 з інтеграцією механізмів просторової уваги та оптичного розпізнавання символів;

– реалізовано інформаційну систему для виявлення та розпізнавання номерних знаків автомобілів з використанням розробленого методу;

– проведено експериментальне дослідження ефективності розробленого методу шляхом оцінки точності виявлення та розпізнавання номерних знаків за різних умов роботи системи.

Оптимальною для навчання є модель, тренована на середньому датасеті з 1 класом. Вона досягає найвищої прецензійності (0.921 на 30 епохах) і високих результатів по інших метриках, зокрема F1-міра на рівні 0.822 і mAP50, який досягає 0.829. Однак після 50 епох спостерігається зниження прецензійності до 0.8, що може свідчити про перенавчання. Тому найкращі результати для цієї моделі досягаються саме на 30 епохах, що дозволяє забезпечити хорошу стабільність та високу ефективність.

Навчання на середньому датасеті з 4 класами показує більш стабільні результати на всіх етапах. Прецензійність і повнота збільшуються до 0.724 і 0.725 на 50 епохах відповідно, з високою F1-мірою на рівні 0.724. Модель стабільно працює і продовжує вдосконалюватися з кожною епохою, що свідчить про ефективне використання середніх даних для навчання. Високі значення mAP50 і mAP50-95 вказують на здатність моделі до точного виявлення та класифікації об'єктів.

Водночас навчання на малому датасеті виявляється менш стабільним. Прецензійність і F1-міра досягають максимальних значень лише на 30 епохах (0.925 та 0.634), після чого починають знижуватися, що вказує на ризик перенавчання. Повнота на малому датасеті значно нижча (0.6 на 50 епохах), а втрати значно вищі порівняно з середніми датасетами. Ці результати вказують на те, що для малих датасетів необхідно ретельно налаштовувати кількість епох, щоб уникнути перенавчання і досягти збалансованої продуктивності. Моделі, треновані на малих датасетах, мають проблеми з локалізацією і класифікацією об'єктів, що відображається в низьких значеннях mAP.

Удосконалено метод автоматизованого виявлення та розпізнавання номерних знаків автомобілів, який відрізняється від існуючих застосувань модифікованої нейромережевої архітектури YOLOv11 з інтегрованими механізмами просторової уваги та двоетапним підходом до обробки зображень, що в поєднанні з оптимізованим процесом навчання на основі адаптивного налаштування гіперпараметрів дозволило забезпечити стабільну роботу системи при різних умовах освітлення та якості вхідних зображень.

За темою кваліфікаційної роботи магістра, автором виконано 1 наукову публікацію – Яворський К.А., Манзюк Е.А., Скрипник Т.К. Метод виявлення та розпізнавання номерів автомобілів нейромережевими засобами. Збірник наукових праць за матеріалами XVI Всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук АПКН-2024». - Хмельницький, 2024. - С.572 – 575.

Перелік посилань

1. Автопарк України досяг 10 мільйонів одиниць: який середній вік авто?. *auto.24tv.ua*.
URL: https://auto.24tv.ua/avtopark_ukrainy_v_2021_rotsi_statystyka_n28094Zou.
2. Використання нейронних мереж – перспективна сфера науки і суспільства | наукові конференції. *Грудень 2013 / Наукові конференції*.
URL: <http://oldconf.neasmo.org.ua/node/139>. (дата звернення: 26.11.2024).
3. Alif M. Yolov11 for vehicle detection: advancements, performance, and applications in intelligent transportation systems. *https://arxiv.org/*.
URL: <https://arxiv.org/pdf/2410.22898>.
4. Aribas E., Daglarli E. Realtime object detection in IoT (Internet of Things) devices. *2017 25th Signal Processing and Communications Applications Conference (SIU), Antalya, Turkey, 15–18 May 2017. 2017*.
URL: <https://doi.org/10.1109/siu.2017.7960690> (date of access: 04.12.2024).
5. Artificial intelligence definitions. *Home / Stanford HAI*.
URL: <https://hai.stanford.edu/sites/default/files/2023-03/AI-Key-Terms-Glossary-Definition.pdf> (date of access: 26.11.2024).
6. Besl P. J., Jain R. C. Three-dimensional object recognition. *ACM computing surveys*. 1985. Vol. 17, no. 1. P. 75–145. URL: <https://doi.org/10.1145/4078.4081> (date of access: 26.11.2024).
7. Contributions and risks of artificial intelligence (ai) in building smarter cities: insights from a systematic review of the literature. *MDPI*.
URL: <https://doi.org/10.3390/en13061473> (date of access: 26.11.2024).
8. Deep learning-based object detection and scene perception under bad weather conditions / T. Sharma et al. *Electronics*. 2022. Vol. 11, no. 4. P. 563.
URL: <https://doi.org/10.3390/electronics11040563> (date of access: 26.11.2024).

9. Elhamraoui Z. Introduction to convolutional neural network. *Medium*. URL: <https://medium.com/analytics-vidhya/introduction-to-convolutional-neural-network-6942c189a723> (date of access: 26.11.2024).
10. GitHub - tesseract-ocr/tessdoc: tesseract documentation. *GitHub*. URL: <https://github.com/tesseract-ocr/tessdoc> (date of access: 04.12.2024).
11. GitHub - ultralytics/ultralytics: Ultralytics YOLO11. *GitHub*. URL: <https://github.com/ultralytics/ultralytics> (date of access: 04.12.2024).
12. Global ev outlook 2024 – analysis - iea. *IEA*. URL: <https://www.iea.org/reports/global-ev-outlook-2024> (date of access: 26.11.2024).
13. He K., Zhang X. Deep residual learning for image recognition. *arxiv.org*. URL: <https://arxiv.org/pdf/1512.03385>.
14. Jajal P., Jiang W. Analysis of failures and risks in deep learning model converters: A case study in the ONNX ecosystem. *https://arxiv.org/*. URL: <https://arxiv.org/abs/2303.17708> (date of access: 04.12.2024).
15. Jyothi M. A., Kalidas M. M. Real time smart object detection using machine learning. *International journal for research in applied science and engineering technology*. 2022. Vol. 10, no. 11. P. 212–217. URL: <https://doi.org/10.22214/ijraset.2022.47281> (date of access: 04.12.2024).
16. Khanam R., Hussain M. Yolov11: an overview of the key architectural enhancements. *https://arxiv.org/*. URL: <https://arxiv.org/pdf/2410.17725>.
17. Krizhevsky A., Sutskever I. ImageNet classification with deep convolutional neural networks. URL: <https://dl.acm.org/doi/10.1145/3065386>.
18. Mithe R., Indalkar S. Optical character recognition. *https://citeseerx.ist.psu.edu/*. URL: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=6a4b4f04d5ce3c3592832eb40c23cc8fc5a9131e>.
19. New car registrations: +4.3% in june 2024; battery electric 14.4% market share. *acea.auto*. URL: <https://www.acea.auto/pc-registrations/new-car-registrations-4-3-in-june-2024-battery-electric-14-4-market-share/> (дата звернення: 26.11.2024).

20. Parvin S., Rozario L. J., Islam M. E. Vehicle number plate detection and recognition techniques: a review. *Advances in science, technology and engineering systems journal*. 2021. Vol. 6, no. 2. P. 423–438. URL: <https://doi.org/10.25046/aj060249> (date of access: 04.12.2024).
21. Rahul, Bansal D. Object detection using machine learning and deep learning. *International journal for research in applied science and engineering technology*. 2023. Vol. 11, no. 2. P. 265–268. URL: <https://doi.org/10.22214/ijraset.2023.48958> (date of access: 04.12.2024).
22. Rakshit S., Basu S. Recognition of handwritten roman script using tesseract open source OCR engine. <https://arxiv.org/>. URL: <https://arxiv.org/pdf/1003.5891>.
23. Real-time computer vision with OpenCV / K. Pulli et al. *Communications of the ACM*. 2012. Vol. 55, no. 6. P. 61–69. URL: <https://doi.org/10.1145/2184319.2184337> (date of access: 04.12.2024).
24. Real-Time object detection using deep learning / B. Singh et al. *International journal for research in applied science and engineering technology*. 2022. Vol. 10, no. 5. P. 3159–3160. URL: <https://doi.org/10.22214/ijraset.2022.42820> (date of access: 04.12.2024).
25. Sapkota R., Karkee M. Comparing yolo11 and yolov8 for instance segmentation of occluded and non-occluded immature green fruits in complex orchard environment. <https://arxiv.org/>. URL: <https://arxiv.org/pdf/2410.19869>.
26. Shaikh S. H., Saeed K., Chaki N. Moving object detection approaches, challenges and object tracking. *Moving object detection using background subtraction*. Cham, 2014. P. 5–14. URL: https://doi.org/10.1007/978-3-319-07386-6_2 (date of access: 04.12.2024).
27. Shinde A. Text pre-processing and text segmentation for OCR. <https://ijcset.net/>. URL: <https://ijcset.net/docs/Volumes/volume2issue1/ijcset2012020111.pdf>.
28. Singh A., Bacchuwar K., Bhasin A. A survey of OCR applications. *International journal of machine learning and computing*. 2012. P. 314–318. URL: <https://doi.org/10.7763/ijmlc.2012.v2.137> (date of access: 04.12.2024).

29. Small object detection with multiscale features / G. X. Hu et al. *International journal of digital multimedia broadcasting*. 2018. Vol. 2018. P. 1–10. URL: <https://doi.org/10.1155/2018/4546896> (date of access: 04.12.2024).
30. Stancel M., Holic M. An introduction to image classification and object detection using YOLO detector. <https://www.semanticscholar.org/>. URL: <https://www.semanticscholar.org/paper/An-Introduction-to-Image-Classification-and-Object-Stancel-Holic/35aea156a4d0a196823044532e5844617930937e>.
31. Suetens P., Fua P., Hanson A. J. Computational strategies for object recognition. *ACM computing surveys*. 1992. Vol. 24, no. 1. P. 5–62. URL: <https://doi.org/10.1145/128762.128763> (date of access: 26.11.2024).
32. Ultralytics. *GitHub*. URL: <https://github.com/ultralytics> (date of access: 04.12.2024).
33. Urban ai: understanding the emerging role of artificial intelligence in smart cities / A. Luusua et al. *Ai & society*. 2022. URL: <https://doi.org/10.1007/s00146-022-01537-5> (date of access: 26.11.2024).
34. Very deep convolutional networks (VGG) essential guide - viso.ai. *viso.ai*. URL: <https://viso.ai/deep-learning/vgg-very-deep-convolutional-networks/> (date of access: 26.11.2024).
35. Wang C.-Y., Yeh I.-H. YOLOv9: learning what you want to learn using programmable gradient information. URL: <https://doi.org/10.48550/arXiv.2402.13616>.
36. Wen H., Dai F., Yuan Y. A study of YOLO algorithm for target detection. *Proceedings of international conference on artificial life and robotics*. 2021. Vol. 26. P. 622–625. URL: <https://doi.org/10.5954/icarob.2021.os13-9> (date of access: 04.12.2024).
37. Yang F., Choi W. Exploit all the layers: fast and accurate CNN object detector with scale dependent pooling and cascaded rejection classifiers. *openaccess.thecvf.com*. URL: https://openaccess.thecvf.com/content_cvpr_2016/papers/Yang_Exploit_All_the_CVPR_2016_paper.pdf.
38. Zhang Q., Wang W., Zhu S.-C. Examining CNN representations with respect to dataset bias. *Proceedings of the AAAI conference on artificial intelligence*. 2018.

Vol. 32, no. 1. URL: <https://doi.org/10.1609/aaai.v32i1.11833> (date of access: 26.11.2024).

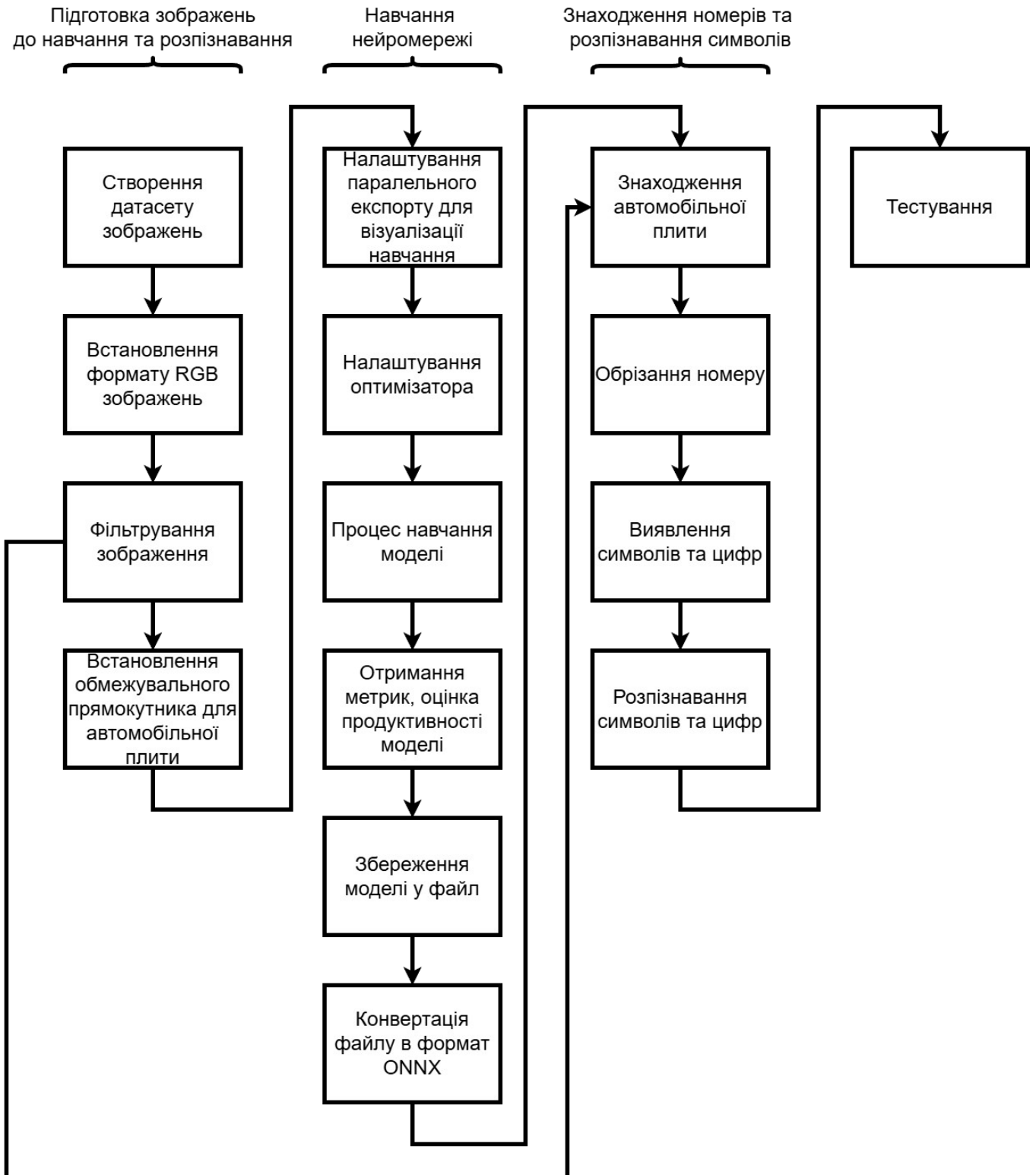
39. Zhao Z.-Q. Object detection with deep learning: a review. *arxiv.org*. URL: <http://arxiv.org/pdf/1807.05511>.

40. Zou Z., Chen K. Object detection in 20 years: a survey. URL: <https://doi.org/10.48550/arXiv.1905.05055>.

ДОДАТКИ

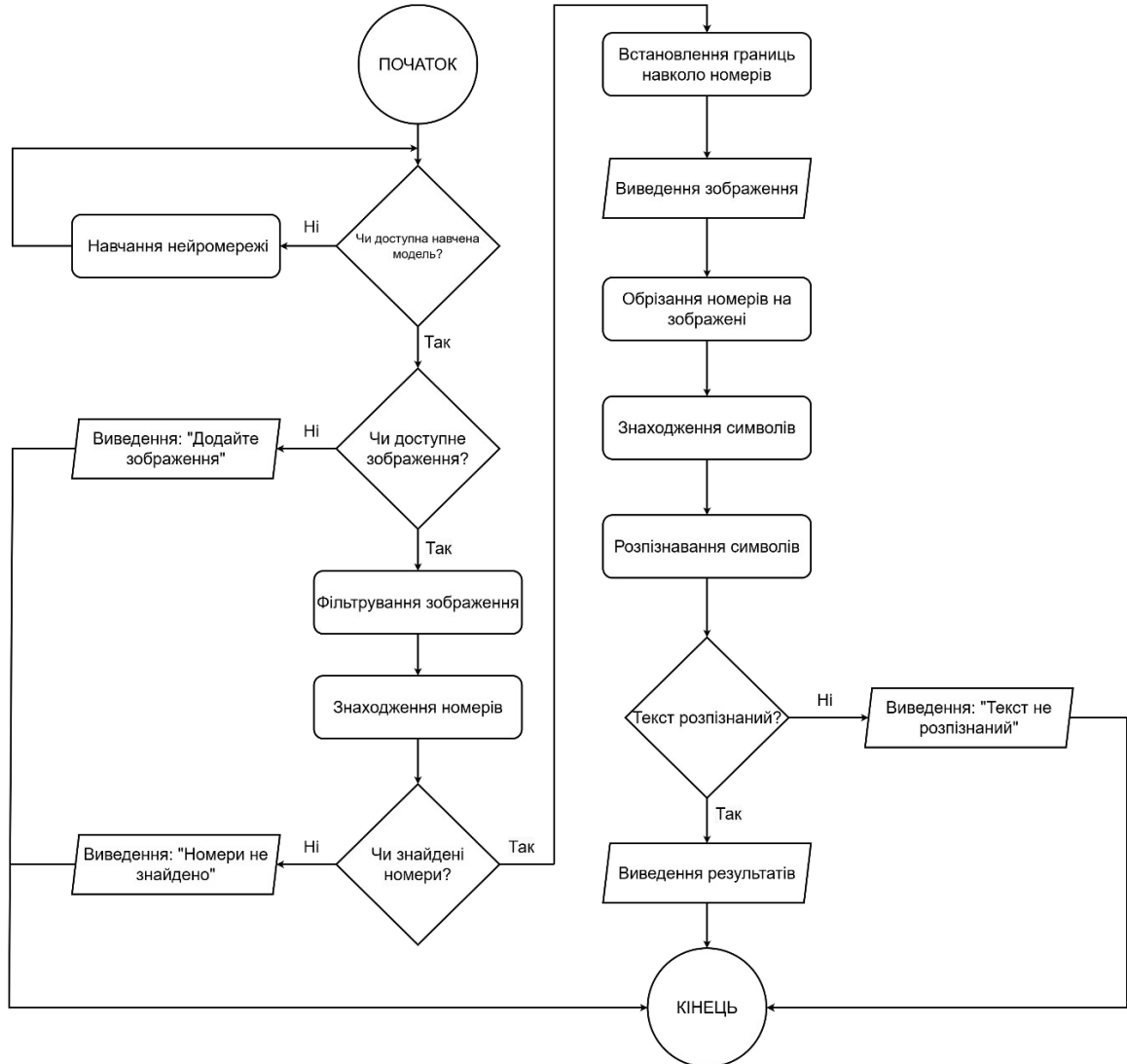
Додаток А

Метод знаходження та розпізнавання автомобільних номерів за допомогою неймережі



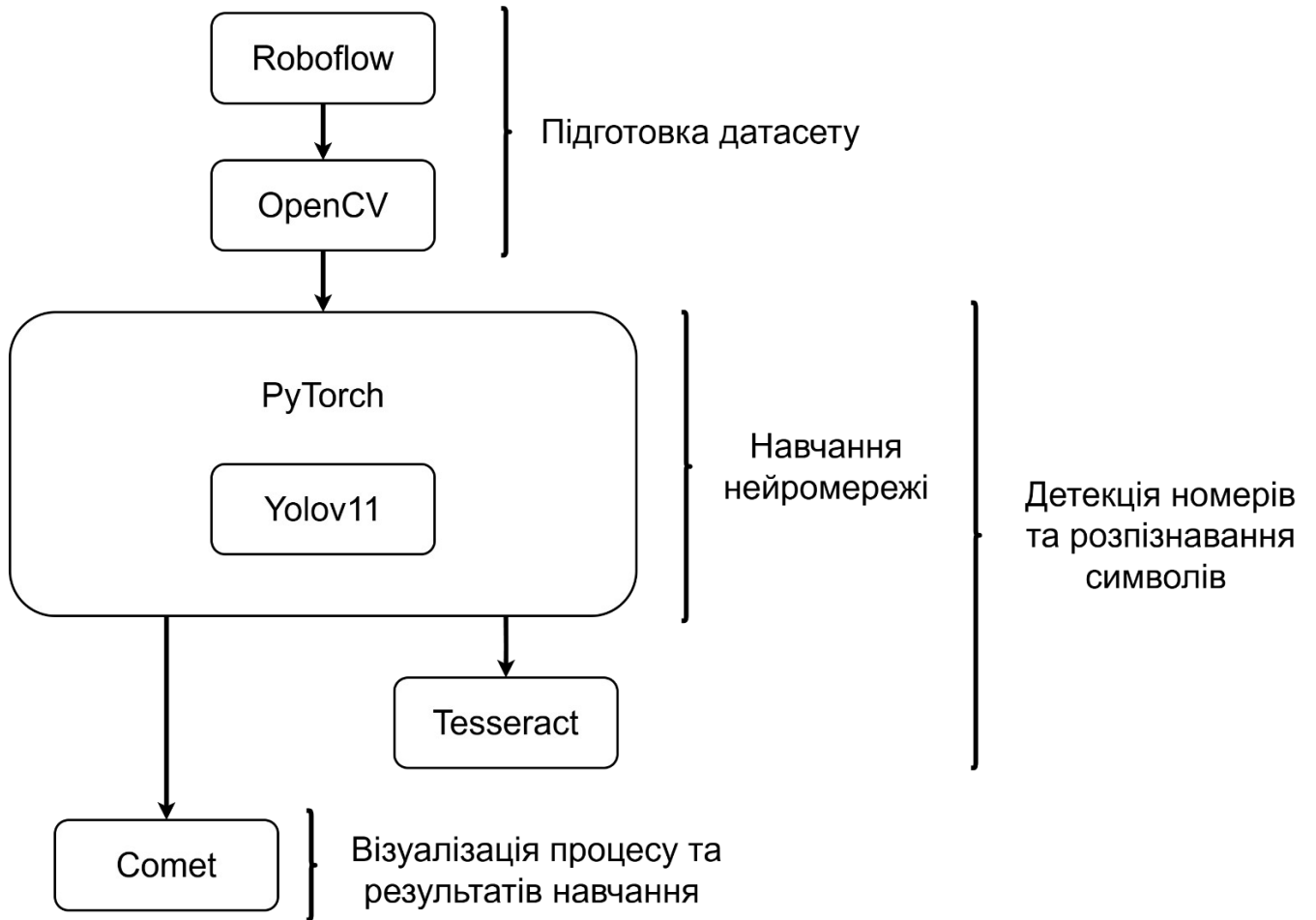
Додаток Б

Схема алгоритму знаходження та розпізнавання номерів автомобілей



Додаток В

Схема інформаційної системи забезпечення методу



Додаток Г
Презентаційний матеріал

Хмельницький національний
університет
Факультет інформаційних технологій
Кафедра комп'ютерних наук



МЕТОД ВИЯВЛЕННЯ ТА
РОЗПІЗНАВАННЯ НОМЕРІВ
АВТОМОБІЛІВ
НЕЙРОМЕРЕЖЕВИМИ ЗАСОБАМИ

Виконав: студент КНм-23-2
Яворський Костянтин Андрійович
Керівник: д.т.н., професор кафедри КН
Манзюк Едуард Андрійович

МЕТА ДОСЛІДЖЕННЯ

Мета дослідження полягає у підвищенні ефективності автоматизованого виявлення та розпізнавання номерних знаків автомобілів шляхом адаптивного налаштування гіперпараметрів нейронних мереж.

АКТУАЛЬНІСТЬ ДОСЛІДЖЕННЯ

На сьогоднішній день спостерігається зростання кількості автомобілів на дорогах, що збільшує потребу в ефективних системах контролю та моніторингу транспортних потоків. З поглибленням цифрової трансформації у світі, зростає важливість автоматизованого знаходження та розпізнавання номерних знаків автомобілів для підвищення ефективності контролю дорожнього руху, безпеки та знаходженню транспортних засобів.

Об'єкт дослідження – процеси виявлення та розпізнавання номерних знаків автомобілів на зображеннях та відеопослідовностях.

Предмет дослідження – моделі, методи та алгоритми глибокого навчання для виявлення та розпізнавання номерних знаків автомобілів із застосуванням згорткових нейронних мереж.

ЗАДАЧІ ДОСЛІДЖЕННЯ

Необхідно виконати такі завдання:

- провести аналіз існуючих методів та підходів до автоматизованого виявлення і розпізнавання номерних знаків автомобілів, визначити їх переваги та недоліки для обґрунтування напрямів удосконалення;
- розробити метод автоматизованого виявлення та розпізнавання номерних знаків на основі модифікованої архітектури YOLOv11 з інтеграцією механізмів просторової уваги та оптичного розпізнавання символів;
- розробити інформаційну систему для виявлення та розпізнавання номерних знаків автомобілів з використанням розробленого методу;
- провести експериментальне дослідження ефективності розробленого методу шляхом оцінки точності виявлення та розпізнавання номерних знаків за різних умов роботи системи.

ЕФЕКТИВНІ ІСНУЮЧІ МЕТОДИ ТА ПІДХОДИ ВИЯВЛЕННЯ ТА РОЗПІЗНАВАННЯ АВТОМОБІЛЬНИХ НОМЕРІВ

YOLO



Tesseract



Tesseract OCR

СХЕМА МЕТОДУ ЗНАХОДЖЕННЯ ТА РОЗПІЗНАВАННЯ АВТОМОБІЛЬНИХ НОМЕРІВ ЗА ДОПОМОГОЮ НЕЙРОМЕРЕЖІ

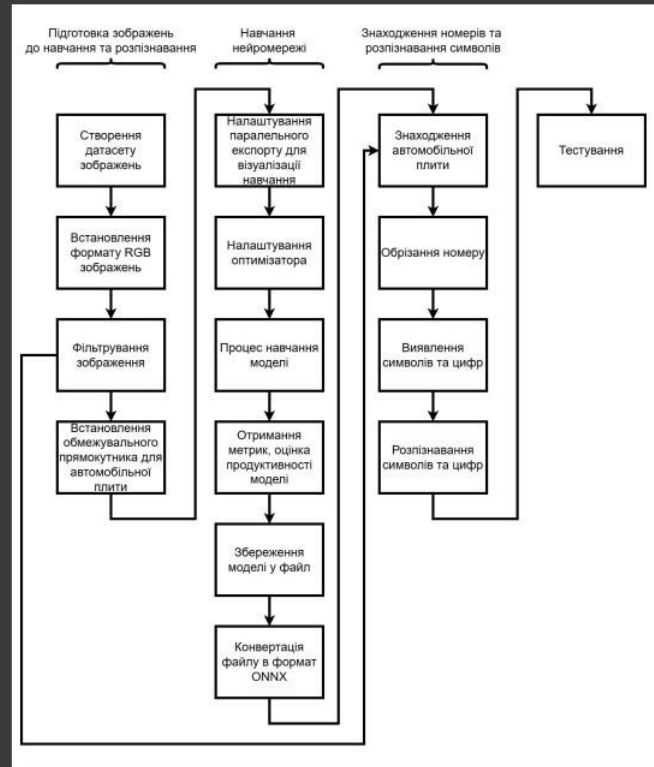


СХЕМА ІНФОРМАЦІЙНОЇ СИСТЕМИ ВІЯВЛЕННЯ ТА РОЗПІЗНАВАННЯ НОМЕРНИХ ЗНАКІВ АВТОМОБІЛІВ

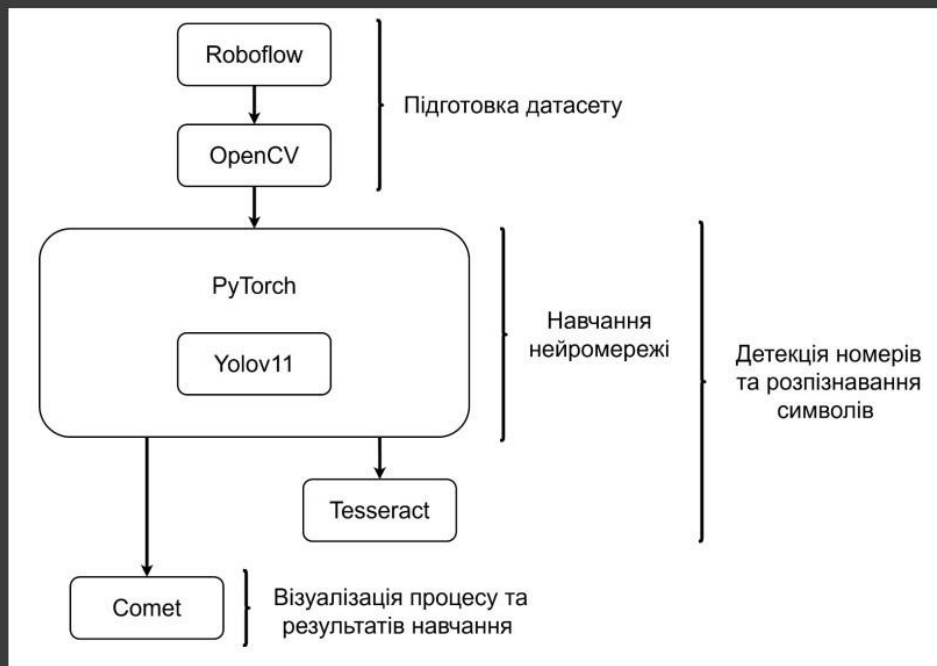


СХЕМА ПРОЦЕСУ НАВЧАННЯ НЕЙРОМЕРЕЖІ

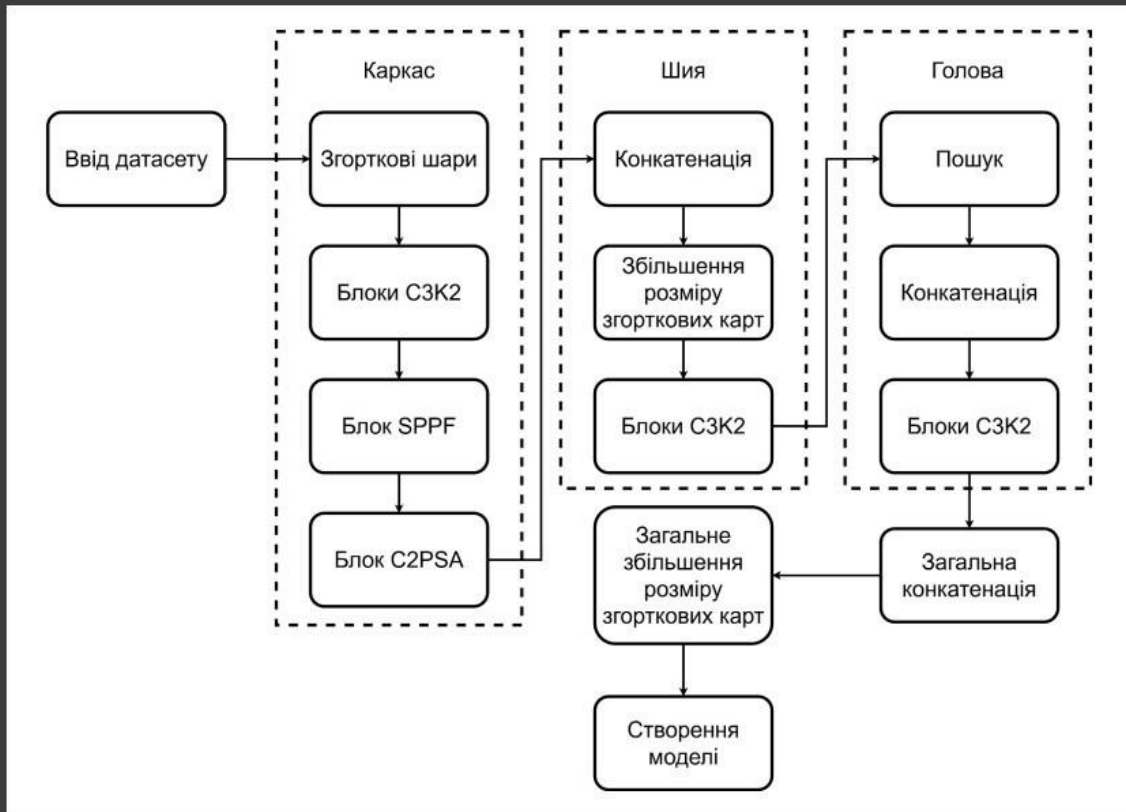


СХЕМА ПРОЦЕСУ ЗНАХОДЖЕННЯ ТА РОЗПІЗНАВАННЯ НОМЕРІВ АВТОМОБІЛЕЙ ЗА ДОПОМОГОЮ ПОПЕРЕДНЬО НАВЧЕНОЇ МОДЕЛІ YOLO ТА TESSERACT



ПРОЦЕС ТЕСТУВАННЯ НЕЙРОМЕРЕЖІ

- Крок №1. Навчання моделі на середньому датасеті з чотирьма класами протягом 15, 30, 50 епох. Вибір оптимальної моделі.
- Крок №2. Навчання моделі на середньому датасеті з одним класом на оптимальній (вибраній на минулому кроці) кількості епох.
- Крок №3. Навчання моделі на малому датасеті з одним класом протягом 15, 30, 50 епох. Вибір оптимальної моделі.
- Крок №4. – Загальне порівняння моделей. Вибір оптимальної.

Кроки	Величина датасету	Кількість класів	Кількість епох
Крок 1	Середній датасет	4	15, 30, 50
Крок 2	Середній датасет	1	30
Крок 3	Малий датасет	1	15, 30, 50

Модель	YOLOv11n
Оптимізатор	AdamW
Learning rate	0.0001
weight_decay	0.0005

В таблицях представлені основні параметри нейромережі, інформація про розміри датасету та параметри навчання

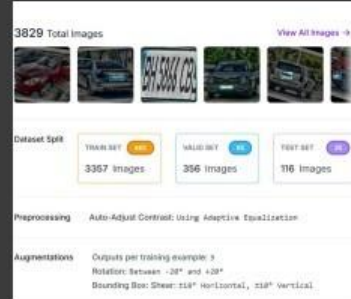
РОЗБИТТЯ ЗОБРАЖЕНЬ ДАТАСЕТУ ЗА ОСНОВНИМИ КРИТЕРІЯМИ



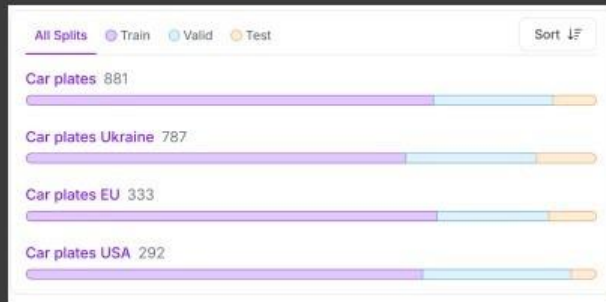
ПІДГОТОВЛЕНІ ДАТАСЕТИ ДЛЯ НАВЧАННЯ НЕЙРОМЕРЕЖІ

Малий за кількістю зображень датасет

Середній за кількістю зображень датасет



Середній за кількістю зображень датасет, розділений на чотири класи



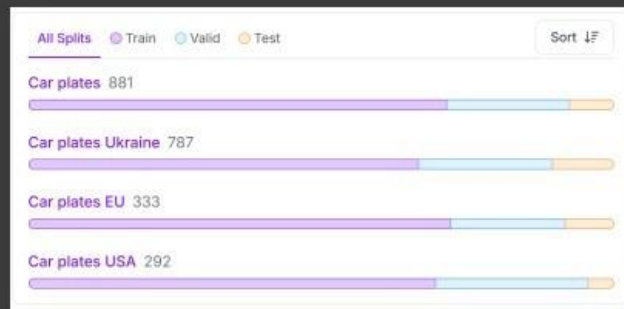
ПІДГОТОВЛЕНІ ДАТАСЕТИ ДЛЯ НАВЧАННЯ НЕЙРОМЕРЕЖІ

Несхожі автомобільні номери

Схожі автомобільні номери



Середній за кількістю зображень датасет, розділений на чотири класи



КРОК №1

Таблиця метрик та графік ефективності процесу навчання моделі на середньому датасеті, з чотирма класами протягом 15, 30, 50 епох.

	Сер. датасет, 4 кл		
Кількість епох	15	30	50
Прецензійність	0.642	0.6973	0.724
Повнота	0.598	0.6687	0.725
F1-міра	0.619	0.682	0.724
Втрати	58.953	58.953	49.524
mAP50	0.654	0.738	0.778
mAP50-95	0.445	0,504	0.562



КРОК №2

Таблиця метрик та графік ефективності процесу навчання моделей, навчених на середньому датасеті, на основі класів

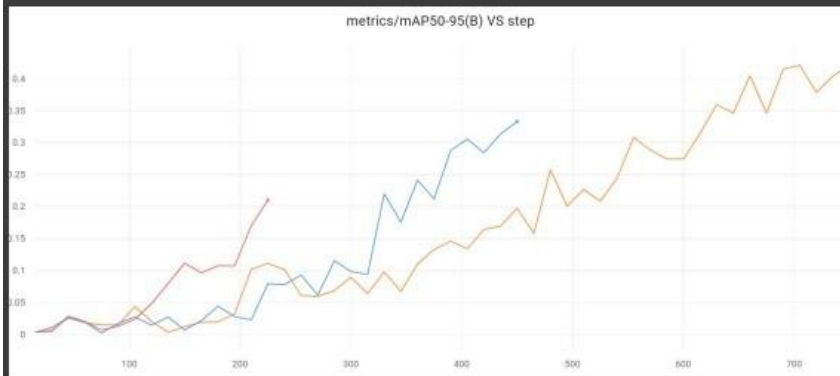
	Сер. датасет, 4 кл			с.датасет 1 кл
Кількість епох	15	30	50	30
Прецензійність	0.642	0.6973	0.724	0.921
Повнота	0.598	0.6687	0.725	0.743
F1-міра	0.619	0.682	0.724	0.822
Втрати	58.953	58.953	49.524	48.267
mAP50	0.654	0.738	0.778	0.829
mAP50-95	0.445	0,504	0.562	0.563



КРОК №3

Таблиця метрик та графік ефективності процесу навчання моделі на малому датасеті, з одним класом протягом 15, 30, 50 епох.

	Малий датасет		
	15	30	50
Кількість епох	15	30	50
Прецизійність	0.671	0.925	0.8
Повнота	0.3	0.483	0.6
F1-міра	0.41	0.634	0.68
Втрати	154.734	102.673	75.799
mAP50	0.368	0.613	0.672



КРОК №4

Графік ефективності процесу навчання, на основі моделей, представлених в таблиці:

Величина датасету	Кількість класів	Кількість епох
Середній датасет	4	30
Середній датасет	1	30
Малий датасет	1	50



КРОК №4

Загальне порівняння моделей

	Сер. датасет, 4 кл			с.датасет 1 кл	Малий датасет		
	15	30	50	30	15	30	50
Кількість епох	15	30	50	30	15	30	50
Прецизійність	0.642	0.6973	0.724	0.921	0.671	0.925	0.8
Повнота	0.598	0.6687	0.725	0.743	0.3	0.483	0.6
F1-міра	0.619	0.682	0.724	0.822	0.41	0.634	0.68
Втрати	58.953	58.953	49.524	48.267	154.734	102.673	75.799
mAP50	0.654	0.738	0.778	0.829	0.368	0.613	0.672
mAP50-95	0.445	0.504	0.562	0.563	0.210	0.333	0.420

НАУКОВА НОВИЗНА ОДЕРЖАНИХ РЕЗУЛЬТАТІВ.

Удосконалено метод автоматизованого виявлення та розпізнавання номерних знаків автомобілів, який відрізняється від існуючих застосуванням модифікованої нейромережевої архітектури YOLOv11 з інтегрованими механізмами просторової уваги та двоетапним підходом до обробки зображень, що в поєднанні з оптимізованим процесом навчання на основі адаптивного налаштування гіперпараметрів дозволило забезпечити стабільну роботу системи при різних умовах освітлення та якості вхідних зображень.

ЗАГАЛЬНІ ВИСНОВКИ

В результаті виконання кваліфікаційної роботи магістра створено метод виявлення та розпізнавання номерів автомобілів нейромережевими засобами.

В процесі створення методу були виконані такі завдання дослідження:

- проведений аналіз існуючих методів та підходів до автоматизованого виявлення і розпізнавання номерних знаків автомобілів, були визначені їх переваги та недоліки для обґрунтування напрямів удосконалення;

- розроблений метод автоматизованого виявлення та розпізнавання номерних знаків на основі модифікованої архітектури YOLOv11 з інтеграцією механізмів просторової уваги та оптичного розпізнавання символів;

- реалізовано інформаційну систему для виявлення та розпізнавання номерних знаків автомобілів з використанням розробленого методу;

- проведено експериментальне дослідження ефективності розробленого методу шляхом оцінки точності виявлення та розпізнавання номерних знаків за різних умов роботи системи.

ДЯКУЮ ЗА УВАГУ

Додаток Д

Публікація

УДК 004.8

ЯВОРСЬКИЙ К. А.,
Хмельницький національний університет
<https://orcid.org/0009-0000-7139-9723>
e-mail: yavakostya@gmail.com

МАНЗЮК Е. А.,
Хмельницький національний університет
<https://orcid.org/0000-0002-7310-2126>
e-mail: eduard.em.km@gmail.com

СКРИПНИК Т. К.
Хмельницький національний університет
<https://orcid.org/0000-0002-8531-5348>
e-mail: tskripnik1970@gmail.com

ПАСІЧНИК О. А.
Хмельницький національний університет
<https://orcid.org/0000-0002-8760-4688>
e-mail: o.a.pasichnyk@gmail.com

ВИЗНАЧЕННЯ ОБСЯГУ ДАНИХ ДЛЯ ЕФЕКТИВНОЇ КЛАСИФІКАЦІЇ НОМЕРІВ АВТОМОБІЛІВ

В роботі наведено результати дослідження впливу обсягу даних, зокрема за розміром датасету та за кількістю класів, на ефективність класифікації номерів автомобілів за допомогою нейронної мережі. Тестування проводиться на нейромережі YOLOv11.

Ключові слова: розпізнавання номерних знаків, нейронні мережі, комп'ютерний зір.

YAVORSKYI KOSTIANTYN, MANZIUK EDUARD, SKRYPNYK TETIANA, PASICHNYK OLEKSANDR
Khmelnitskyi National University

DETERMINATION OF THE DATA VOLUME FOR EFFECTIVE CLASSIFICATION OF VEHICLE LICENSE PLATES

This paper presents the results of a study investigating the impact of dataset size and class count on vehicle license plate classification using neural networks. The research was conducted using the YOLOv11 model, a state-of-the-art architecture for object detection tasks. The study comprises four main steps, each designed to assess how different configurations of training data affect the model's performance.

The first step involved training the model on a medium-sized dataset consisting of four classes over 15, 30, and 50 epochs. The second step focused on training the model using the same medium-sized dataset but limited to a single class for 30 epochs. The third step evaluated the model's performance on a smaller dataset over the same set of epochs (15, 30, 50). After each step, detailed analysis of training graphs and performance metrics was conducted to assess how the model's accuracy and loss evolved with different data sizes and epoch counts.

In the final step, a comprehensive comparison of all models was performed, taking into account the overall performance across the various dataset sizes and class configurations. This comparison aimed to identify the optimal training configuration, with a focus on maximizing accuracy and minimizing overfitting, to determine the best approach for vehicle license plate classification.

Through this study, the research highlights the influence of dataset volume and class distribution on model performance, providing insights into how to effectively train neural networks for classification tasks with varying data complexities. The findings underscore the importance of selecting an appropriate dataset size and class count to balance computational efficiency and classification accuracy in real-world applications.

Keywords: license plate recognition, neural networks, computer vision.

Вступ та постановка проблеми

Кількість автомобілів на дорогах України, Європи, США збільшується, що створює потребу в вдосконаленні методів детекції номерів. Ефективне розпізнавання об'єктів, таких як автомобільні номери, значною мірою залежить від якості та структури навчального набору даних. Грамотно сформований датасет може значно підвищити точність та продуктивність моделі. Автомобільні номери мають різноманітні характеристики, що їх відрізняють. Зокрема, існують унікальні, індивідуалізовані номери, які створені за спеціальними замовленнями та мають нестандартний дизайн. Водночас більшість номерів є стандартизованими, що дозволяє їх класифікувати за країною походження [1, 2].

Поділ датасету на класи може відбуватись на основі цих особливостей. Один клас може включати номери з унікальним дизайном, а інші – стандартизовані номери, розподілені за географічною ознакою. Такий підхід до структуризації даних сприяє покращенню адаптації моделі до різних типів номерів та підвищує її здатність до розпізнавання у реальних умовах.

З наукової точки зору, дослідження ефективності класифікації номерів допомагає покращити розуміння впливу різних параметрів навчальних даних на продуктивність моделей. Аналіз структурованих датасетів із різними класами (наприклад, унікальні чи стандартизовані номери) сприяє розробці більш універсальних алгоритмів глибокого навчання, здатних адаптуватися до складних і неоднорідних умов. Це також відкриває нові перспективи для вивчення адаптивності моделей до багатокласових задач, що є актуальним напрямом у машинному навчанні.

З практичної точки зору, розв'язання цієї проблеми має безпосередній вплив на реальні сценарії застосування, такі як автоматичне розпізнавання номерних знаків у транспортних системах, безпекових рішеннях, автоматизації процесів паркування та митного контролю. Використання грамотно підібраного датасету дозволяє підвищити точність роботи систем розпізнавання, знизити кількість помилкових визначень та покращити продуктивність навіть у складних умовах, таких як різні погодні умови чи змішання стилів номерних знаків.

Метою роботи є: дослідження впливу обсягу даних на ефективність класифікації за допомогою нейронних мереж, зокрема вивчення взаємозв'язку між розміром датасету та кількістю класів. Це дослідження включає аналіз того, як збільшення обсягу даних та варіації в класах можуть вплинути на точність класифікації в задачах детекції, зокрема для класифікації номерів автомобілів. Одним з ключових аспектів є визначення оптимального розміру датасету та кількості класів, що дозволяє досягти найкращих результатів при тренуванні моделей нейронних мереж, таких як YOLO, для завдань, що вимагають високої точності та ефективності. У рамках роботи також вивчаються питання адаптації моделей до змін в обсязі даних та оптимізація навчальних процесів для забезпечення кращої узагальнюючої здатності при роботі з реальними умовами.

Дослідження

Для проведення дослідження була використана нейромережа YOLOv11, яка є однією з найбільш ефективних архітектур для задач детекції об'єктів. Процес навчання моделі за допомогою YOLOv11, зображений на рисунку 1. Зображення датасету потрапляють до основних компонентів архітектури YOLO, які можна поділити на три головні частини: каркас, шия і голова.

Каркас відповідає за вилучення ознак із зображення – це глибока згортокова нейронна мережа, яка аналізує зображення і формує набір характеристик. Далі ці ознаки передаються до "ший", яка об'єднує інформацію з різних рівнів ієрархії ознак і готує їх до остаточного передбачення. "Голова" моделі виконує фінальні операції детекції та ідентифікації об'єктів, формуючи відповідні бокси навколо об'єктів та визначаючи їх категорії.

На останніх етапах передбачення, отримані від "голови", проходять процеси загального об'єднання та вирівнювання, що забезпечує точність і стабільність виявлених об'єктів. Після цього модель зберігається у вигляді ваг і архітектури для подальшого використання, тестування чи розгортання [3].

YOLOv11 включає нові архітектурні інновації та оптимізації в порівнянні з минулими версіями, зокрема заміну блоку C2f на C3k2 для покращення продуктивності детекції. C3k2 оптимізує обчислення за рахунок використання двох малих згорток замість однієї великої, як у YOLOv8, що підвищує швидкість обробки без втрати якості. Рівняння що описує цей блок:

$$C3k2(X) = Conv(Split(X)) + Conv(Merge(Split(X))), \quad (1)$$

де Conv – це операція згортки, Split – операція розділення карти ознак на 2 частини, Merge – операція об'єднання вихідних результатів.

У YOLOv11 також зберігається використання блоку просторової піраміди (SPPF), але додано новий блок C2PSA. Цей блок дозволяє моделі краще фокусуватися на важливих ділянках зображення, покращуючи точність виявлення об'єктів різного розміру та положення завдяки механізмам просторової уваги. Рівняння що описує цей блок:

$$C2PSA(X) = Attention(Concat(X_{path1}, X_{path2})), \quad (2)$$

де Attention – це механізм, що дозволяє фокусуватися на важливих частинах, Concat – операція, яка з'єднує карти ознак.

Рівняння, що описує блок SPPF:

$$SPPF(X) = Concat(MaxPool(X, 5), MaxPool(X, 3), MaxPool(X, 1)), \quad (3)$$

де MaxPool(X) – це операція “максимального пулінгу” [4].

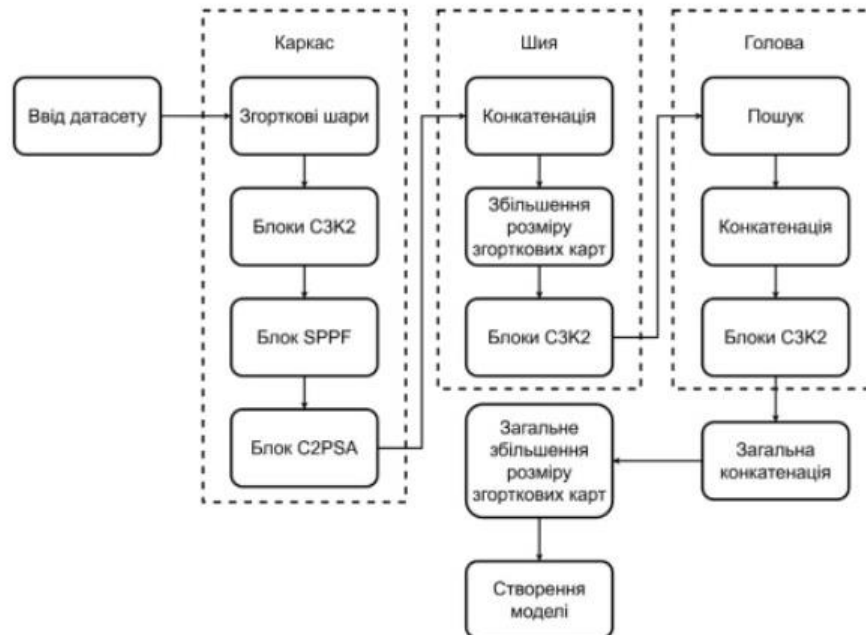


Рис. 1. Схема процесу навчання нейромережі YOLOv11

Для проведення дослідження були підготовлені датасети з малою та середньою кількістю зображень для навчання. Утворено дві версії середнього, за кількістю зображень, датасету. В першій, зображення розділились за чотирма наступними класами

– номери машин (Car plates);

- номери машин України (Car plates Ukraine);
- номери машин ЄС (Car plates EU);
- номери машин США (Car plates USA).

В другій версії, зображення належали одному об'єднаному класу. Розділення датасета на класи зображено на рисунку 2.

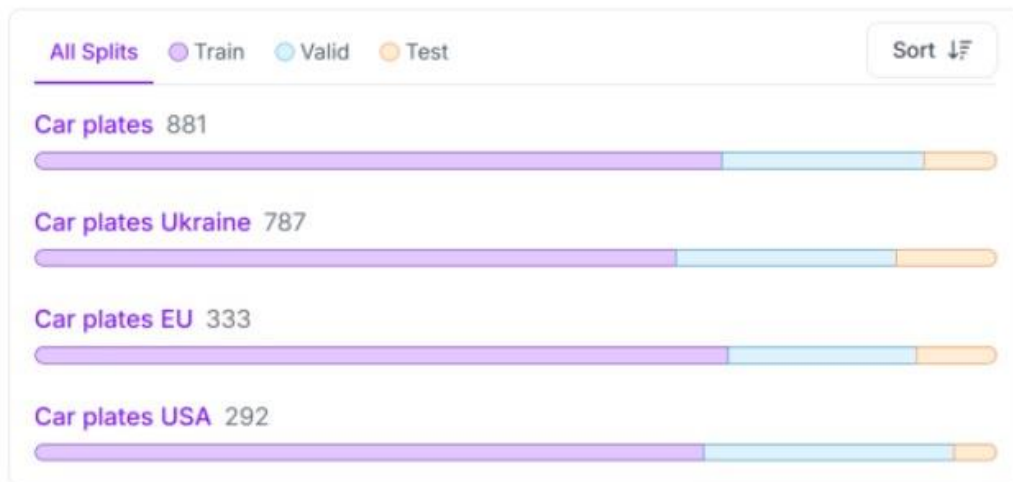


Рис. 2. Кількість анотацій кожного класу

Малий датасет складається з 330 зображень, тоді як середній датасет, після попередньої обробки, що включала дублювання зображень із шумом та їх поворот за осями, налічує 3829 зображень.

Оптимізатор, для створення моделі, налаштований за допомогою AdamW –вдосконаленої версії Adam, який використовує адаптивні моменти для кожного параметра, що дозволяє ефективніше коригувати ваги в процесі навчання. Вибраний оптимізатор розв'язує одну з проблем класичного рішення – не зовсім коректне застосування вагового спаду (weight decay). У AdamW процес регуляризації та корекція параметрів за допомогою адаптивних моментів (перших і других моментів) відбуваються незалежно один від одного.

Для навчання використовується початкова швидкість навчання (learning rate) 0.0001 – досить низьке значення, що дозволяє моделі поступово коригувати свої параметри, запобігаючи надмірним коливанням під час навчання. Параметр weight_decay встановлений на значення 0.0005, що додає до функції втрат штраф за великі значення ваг, сприяючи покращенню узагальнення моделі і зменшенню ймовірності перенавчання.

Використання цих налаштувань дозволяє збалансувати швидкість навчання і ефективність оптимізації, забезпечуючи стабільний процес навчання та мінімізацію втрат.

Дослідження складається з чотирьох кроків: першим кроком відбулось навчання моделі на середньому датасеті з чотирма класами протягом 15, 30, 50 епох, другим кроком відбулось навчання, на середньому датасеті з одним класом на 30 епохах, третім кроком відбулось навчання моделі на малому датасеті протягом 15, 30, 50 епох. В кінці кожного кроку був проведений аналіз графіків та метрик. Четвертим кроком відбулось загальне порівняння моделей, та вибір оптимальної.

Під час навчання на першому кроці спостерігається швидке зростання mAP на початкових етапах (до 2000 кроків), що свідчить про ефективне тренування на початку. Після досягнення 4000-5000 кроків модель виходить на плато – приріст точності незначний, але стабільний. Після 6000 кроків mAP залишається майже незмінним, що може свідчити про досягнення межі продуктивності даної моделі на конкретному датасеті. Оптимальною моделлю на цьому кроці можна вважати ту, яка навчена на тридцяти епохах. Ефективність навчання моделі на середньому датасеті протягом 15, 30, 50 епох зображена на рисунку 3.



Рис. 3. Графік "mAP50-95 на крок" під час навчання нейромережі

Другим кроком було проведене навчання на середньому датасеті протягом оптимальних 30 епох, без розділу на різні класи. На рисунку 4 зображено порівняння "mAP50-95" на крок" двох моделей, навчених на середньому по величині датасеті. Обидві моделі показують схожий приріст продуктивності на початкових етапах. Однак модель, навчена на 4 класах, демонструє повільніший темп зростання mAP у порівнянні з моделлю, навченою на об'єднаному класі. На середніх кроках модель з об'єднаними класами має стабільно вищі показники точності порівняно з моделлю на чотирьох класах. Це може свідчити про те, що об'єднана модель краще справляється з узагальненням даних і швидше навчається розпізнавати основні характеристики об'єктів. На фінальних етапах різниця між моделями зберігається, причому об'єднана модель досягає вищого mAP на останніх етапах тренування.



Рис. 4. Графік "mAP50-95 на крок" під час навчання нейромережі

На третьому кроці, після навчання моделей на малому датасеті проведемо їх порівняння. На графіку, зображеному на рисунку 5, можна побачити ефективність процесу навчання. На початкових етапах навчання всі моделі показують схожу результативність, проте згодом їх зростання стає більш стрімким. Модель, навчена на 50 епохах, демонструє найвищий темп поліпшення показників. Наприкінці навчання не спостерігається значного спаду темпу, що свідчить про доцільність продовження навчання на більшій кількості епох.



Рис. 5. Графік "mAP50-95 на крок" під час навчання нейромережі

Останнім кроком проведемо загальне порівняння моделей. На рисунку 6 зображено порівняння оптимальних моделей, обраних на минулих етапах, за mAP 50-95 на крок.



Рис. 6. Графік "mAP50-95 на крок" під час навчання нейромережі

На графіку спостерігається значна різниця між моделями, навченими на середньому та малому датасетах. У випадку малого датасету видно недостатню кількість кроків для досягнення стабільної роботи системи у порівнянні з іншими двома. Отже, оптимальною можна вважати модель, навчену на середньому датасеті з одним класом, протягом тридцяти епох, так як вона показала найкращі результати.

Отриманні під час навчання метрики представлені в таблиці 4.3.

Таблиця 1

Метрики навчених моделей

	Сер. датасет, 4 кл			с.датасет 1 кл	Малий датасет		
	15	30	50	30	15	30	50
Прецизійність	0.642	0.6973	0.724	0.921	0.671	0.925	0.8
Повнота	0.598	0.6687	0.725	0.743	0.3	0.483	0.6
F1-міра	0.619	0.682	0.724	0.822	0.41	0.634	0.68
Втрати	58.953	58.953	49.524	48.267	154.734	102.673	75.799
mAP50	0.654	0.738	0.778	0.829	0.368	0.613	0.672
mAP50-95	0.445	0.504	0.562	0.563	0.210	0.333	0.420

Порівняння трьох різних датасетів – середнього з 4 класами, середнього з 1 класом та малого – виявляє відмінності в продуктивності моделей залежно від розміру і складності даних, а також кількості епох навчання.

Середній датасет з 4 класами демонструє стабільне покращення продуктивності зі збільшенням кількості епох. Прецензійність підвищується з 0.642 до 0.724, а повнота – з 0.598 до 0.725. F1-міра, що відображає баланс між цими метриками, також зростає до 0.724 після 50 епох. Втрати знижуються з 58.953 до 49.524, що свідчить про ефективне навчання. mAP50 значно покращується (з 0.654 до 0.778), тоді як mAP50-95 зростає помірно, досягаючи 0.562. Це вказує на те, що модель ефективно справляється зі складними завданнями детекції, особливо при достатній кількості епох.

Середній датасет з 1 класом має найвищу прецензійність, що досягає 0.921 на 30 епохах, але після 50 епох вона знижується до 0.8, що може вказувати на перенавчання. Повнота досягає максимуму на 30 епохах (0.743), а F1-міра – 0.822. Втрати знижуються значно (від 154.734 до 48.267), що показує стабілізацію навчання. mAP50 досягає найвищого значення (0.829) на 30 епохах, після чого покращення стає мінімальним. Це свідчить про те, що модель швидко навчається на менш складних даних, але подальше навчання може призвести до деградації.

Малий датасет демонструє нестабільні результати через обмежену кількість даних. Прецензійність зростає до 0.925 на 30 епохах, але знижується до 0.8 на 50 епохах. Повнота починається з дуже низького значення (0.3), але поступово зростає до 0.6. F1-міра також зростає, досягаючи максимуму 0.68 на 50 епохах. Втрати залишаються високими порівняно із середніми датасетами, хоча і знижуються зі 154.734 до 75.799. mAP50 та mAP50-95 демонструють поступове покращення, досягаючи 0.672 та 0.420 відповідно. Цей датасет вимагає більше епох для досягнення прийнятних результатів, але через обмеженість даних перенавчання може настати швидше.

Таким чином, навчання на середніх датасетах забезпечує більш стабільні та високі результати, тоді як малий датасет потребує ретельного налаштування кількості епох для уникнення перенавчання і досягнення збалансованої продуктивності.

Висновки

Експериментальне дослідження впливу обсягу даних на ефективність класифікації зокрема за розміром датасету та за кількістю класів номерів автомобілів за допомогою моделі YOLOv11 виявило кілька важливих висновків.

По-перше, результати дослідження підтверджують, що розмір датасету та кількість класів суттєво впливають на точність класифікації. Великі датасети як правило, дають кращі результати, хоча це супроводжується збільшенням обчислювальних витрат. Навчання на малому датасеті часто призводить до зниження точності, що підкреслює важливість різноманіття даних та їх обсягу.

По-друге, навчання на середньому датасеті з більшою кількістю класів виявилось менш ефективним, ніж навчання моделі на датасеті, що містить лише один клас, для задачі детекції автомобільних номерів.

По-третє, кількість епох також має критичне значення в процесі навчання моделі. Аналіз показав, що більша кількість епох не завжди призводить до лінійного покращення результатів. Оптимальна кількість епох залежить від розміру та складності датасету. Менші датасети показують краще збереження результату при навчанні на п'ятдесяти епохах та більше. З іншої сторони, збільшується ризик перенавчання моделі.

Перспективи подальшого розвитку полягають в експериментах із більшими за об'ємом датасетами. Розширення обсягів даних дає можливість покращити та розширити отримані результати. Також варто зрівняти данні результати з моделями навченими на старіших версіях YOLO, та на інших нейромережах.

Література

1. Global EV outlook 2024 – analysis - IEA [Electronic resource] // IEA. – Mode of access: <https://www.iea.org/reports/global-ev-outlook-2024> (date of access: 01.11.2024). – Title from screen.

2. New car registrations: +4.3% in June 2024; battery electric 14.4% market share [Electronic resource] // *acea*. – Mode of access: <https://www.acea.auto/pc-registrations/new-car-registrations-4-3-in-june-2024-battery-electric-14-4-market-share/> (date of access: 01.11.2024). – Title from screen.
3. Sapkota R. Comparing yolo11 and yolov8 for instance segmentation of occluded and non-occluded immature green fruits in complex orchard environment [Electronic resource] / Ranjan Sapkota, Manoj Karkee // <https://arxiv.org/>. – Mode of access: <https://arxiv.org/pdf/2410.19869> (date of access: 30.10.2024). – Title from screen.
4. Alif M. Yolov11 for vehicle detection: advancements, performance, and applications in intelligent transportation systems [Electronic resource] / Mujadded Alif // https://arxiv.org. – Mode of access: <https://arxiv.org/pdf/2410.22898> (date of access: 01.11.2024). – Title from screen.

Додаток Ж Сертифікат

Довідка: ВХНУ ТН 18/11/24

Видання: Вісник Хмельницького національного університету. Технічні науки

Категорія фаховості видання: фахове видання України, у якому можуть публікуватися результати дисертаційних робіт на здобуття наукових ступенів доктора наук, кандидата наук та ступеня доктора філософії, категорії «Б» філософії, категорії «Б» (наказ МОН №1643 від 28.12.2019, наказ МОН №409 від 17.03.2020).

Напрямок – технічні науки за спеціальностями – 101, 121, 122, 123, 124, 125, 141, 151, 161, 172, 181, 182 (28.12.2019), спеціальності – 131, 132, 133 (17.03.2020)

Назва статті: Визначення обсягу даних для ефективної класифікації номерів автомобілів.

Автори: Яворський К.А., Манзюк Е.А., Скрипник Т.К. (Хмельницький національний університет))

Номер, у який прийнято статтю: №6 до друку орієнтовно буде рекомендовано до 30 грудня 2024 року.

18.11.2024

Начальник відділу
інтелектуальної власності та трансферу технологій Ю.В.Кравчик



Додаток 3
Сертифікат

Сертифікат № 2024-058-1



Міністерство освіти і науки України
Хмельницький національний університет

СЕРТИФІКАТ



Яворський Костянтин Андрійович

учасник XVI Всеукраїнської науково-практичної конференції
«Актуальні проблеми комп'ютерних наук АПКН-2024»

24 години участі (0,8 ECTS credits)

Голова оргкомітету АПКН-2024

Олег СИНЮК

проректор Хмельницького національного
університету з наукової роботи,
доктор технічних наук, професор

м. Хмельницький
15-16 листопада 2024

E-mail: apkt.khnu@gmail.com

07.12.2024, 22:30

result_5329175639295374796.html

Sat Dec 07 20:16:08 EET 2024, Петровський Сергій Степанович, Хмельницький національний університет, ХНУ

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 0.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилوک в документах: 10%**

ID: 155971 Назва: КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА на тему Метод виявлення та розпізнавання номерів автомобілів нейромережевими засобами Додано в БД: 2024-12-07 Автора: Костянтин ЯВОРСЬКИЙ Керівники: Едуард МАНЗЮК Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	98740	1450	1549 (2%)	22 (2%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

Протокол аналізу звіту подібності науковим керівником

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Костянтин ЯВОРСЬКИЙ

Співавтор:

Назва: Метод виявлення та розпізнавання номерів автомобілів нейромережевими засобами

Науковий керівник: Едуард МАНЗЮК, д.т.н., доцент

Підрозділ: Кафедра комп'ютерних наук

Коефіцієнт подібності 1: 1.6%

Коефіцієнт подібності 2: 0.6%

Мікропробіли: 2

Заміна букв: 13

Інтервали: 0

Білі знаки: 1

Дата створення звіту: 2024-12-07 19:52:04.0

Після аналізу Звіту подібності констатую наступне:

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

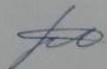
Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедур. Таким чином робота не приймається.

Обґрунтування:

Дата 07.12.24

експерт

 (Бачин Р.О.)

**РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ КАФЕДРИ КОМП'ЮТЕРНИХ НАУК
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ**

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Метод виявлення та розпізнавання номерів автомобілів нейромережевими засобами

Автор: студент гр. КНМ-23-2 Костянтин ЯВОРСЬКИЙ

Спеціальність: 122 – Комп'ютерні науки

Освітня програма: освітньо-професійна

Науковий керівник: д.т.н., професор Манзюк Е.А.

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданій поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданій поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укряття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі не є плагіатом, оскільки запозичення розміщені в розділі огляду існуючих підходів, не описують безпосередньо авторську роботу і не стосуються її результатів; усі запозичення фрагментарні; серед запозичень знаходяться загальновідомі терміни, скорочення.

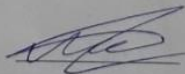
Обсяг запозичень, визначений системами виявлення збігів/ідентичності/схожості, складає:

- за системою Anti-Plagiarism: 0%;

- за системою StrikePlagiarism: КП1 1,6%, КП2 0,6%.

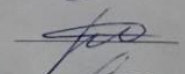
Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості є допустимим.

Керівник роботи



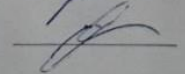
Едуард МАНЗЮК

Гарант ОП



Руслан БАГРІЙ

Завідувач кафедри КН



Олександр БАРМАК



ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МОН УКРАЇНИ

Кафедра комп'ютерних наук



ВІДГУК НАУКОВОГО КЕРІВНИКА на кваліфікаційну роботу магістра

студента гр. КНМ-23-2 Яворського Костянтина Андрійовича

за темою Метод виявлення та розпізнавання номерів автомобілів нейронмережевими засобами

1. Актуальність теми

У сучасному світі спостерігається стрімке збільшення кількості транспортних засобів, що створює загальну потребу в розробці та впровадженні сучасних систем для відстеження та управління рухом на дорогах. В епоху активної цифровізації особливого значення набуває розвиток автоматичних систем виявлення та ідентифікації автомобільних номерних знаків, що сприяє покращенню організації дорожнього руху, підвищенню рівня безпеки та оптимізації пошуку транспортних засобів.

2. Відповідність роботи предметній області Стандарту спеціальності 122 Комп'ютерні науки

Відповідно до встановлених стандартів у галузі, основними об'єктами дослідження виступають різноманітні моделі - математичні, інформаційні та імітаційні, які відображають реальні явища, об'єкти, системи та процеси. Мета дослідження полягає у підвищенні ефективності автоматизованого виявлення та розпізнавання номерних знаків автомобілів шляхом адаптивного налаштування гіперпараметрів нейронних мереж. В процесі розробки були застосовані математичне моделювання, спеціалізовані методи та алгоритми для розв'язання як теоретичних, так і практичних задач у сфері машинного навчання. Проведена робота повністю відповідає вимогам, що висуваються до магістерських досліджень за спеціальністю "Комп'ютерні науки".

3. Професійні та особистісні якості

В процесі підготовки магістерської кваліфікаційної роботи здобувач освіти Яворський К.А. показав значні досягнення у засвоєнні професійних компетентностей та продемонстрував відмінну здатність до вирішення поставлених завдань у встановлені терміни. При написанні пояснювальної записки та розробці методології проявив ґрунтовні знання та досягнув вагомих результатів у своїй роботі. Здобувач продемонстрував високий рівень оволодіння фаховими навичками у сфері комп'ютерних наук.

4. Ступінь самостійності під час виконання кваліфікаційної роботи

Одержані в роботі результати є наслідком особистої діяльності студента, який самостійно виконував усі поставлені задачі.

5. Наукова новизна та оригінальність запропонованих підходів

Магістерська кваліфікаційна робота містить елементи наукової новизни, що виражається в удосконаленні методу автоматичної детекції та ідентифікації автомобільних номерних знаків. Особливістю розробленого підходу є використання вдосконаленої архітектури нейронної мережі YOLOv11, доповненої механізмами просторової уваги та впровадженням двоетапної методики обробки зображень. Завдяки оптимізованому процесу навчання з адаптивним налаштуванням параметрів, система демонструє надійну роботу за різноманітних умов освітлення та при варіативній якості вхідних даних. Основні результати дослідження були представлені науковій спільноті на науково-практичній конференції.

6. Ступінь оволодіння методами дослідження

При реалізації кваліфікаційної роботи продемонстровано належний рівень компетентностей та володіння необхідними методами, методиками та технологіями предметної області комп'ютерних наук.

7. Повнота та якість розкриття теми роботи

Тема роботи повною мірою обґрунтована та розкрита належним чином. Проведено аналіз відомих досліджень відповідно до обраної теми. Поставлені завдання реалізовані та розроблено програмне забезпечення для реалізації запропонованого метода.

8. Логічність, послідовність, аргументованість, літературна грамотність викладення матеріалу

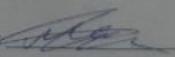
Структура роботи та послідовність викладення логічні та відповідають поставленій меті. Викладення матеріалу послідовне, аргументоване, літературно грамотне.

9. Можливість практичного застосування кваліфікаційної роботи, окремих її частин

Розроблений у роботі метод може бути використаний в автоматизованих системах виявлення автомобільних номерів.

10. Висновок про можливість допуску кваліфікаційної роботи до захисту, на яку оцінку заслуговує робота

Враховуючи належний рівень виконання та забезпечення усіх необхідних вимог, робота може бути допущена до захисту. Рекомендована оцінка «відмінно».

Керівник:  д.т.н., професор каф. КН Едуард МАНЗУК



ВІДГУК ОПОНЕНТА

на кваліфікаційну роботу магістра

ср. КНМ-23-2 Костянтин ЯВОРСЬКИЙ за темою: Метод виявлення та розпізнавання номерів автомобілів нейромережевими засобами

1. Актуальність обраної теми

Обрана магістрантом тема є достатньо актуальною на даний час, особливо з урахуванням тенденцій щодо розвитку штучного інтелекту. Актуальність обраної теми достатньо обгрунтована в роботі.

2. Відповідність роботи предметній області спеціальності 122 Комп'ютерні науки та загальним вимогам до наукових робіт

Тема кваліфікаційної роботи студента повністю відповідає предметній області спеціальності 122 Комп'ютерні науки та загальним вимогам до кваліфікаційної роботи магістра. Робота вирішує науково-прикладну задачу у сфері комп'ютерних наук (а саме застосування інтелектуальних методів обробки інформації (моделей глибокого навчання) до знаходження та розпізнавання автомобільних номерів). У роботі присутнє дослідження, яке розвиває існуючі підходи, та отримані нові результати, які за певними показниками – вищі, ніж в аналогічних підходах.

3. Повнота розкриття мети та завдань дослідження

Часткові завдання дослідження сформульовані коректно та повністю відповідають меті роботи.

4. Наявність наукової новизни

Магістерська робота характеризується наявністю наукової новизни. Положення наукової новизни сформульовані коректно та відображають їх сутність.

Матеріали магістерської роботи апробовані на науковій конференції «Актуальні проблеми комп'ютерних наук».

5. Зміст кожного розділу роботи

Робота складається з чотирьох розділів.

У першому розділі – «Аналіз сучасного стану у використанні методів глибокого навчання для знаходження автомобільних номерів за зображеннями» – автором проведено детальний аналіз предметної області, огляд існуючих публікацій, присвячених детекції об'єктів загалом та автомобільних номерів в частності, також розглянуто моделі глибокого навчання, що застосовуються для виявлення об'єктів малого та середнього розміру на зображеннях. На основі цього аналізу сформульовано задачу дослідження.

У другому розділі – «Вдосконалення методу виявлення та розпізнавання номерів автомобілів нейронними мережами» – магістрантом описано обрану модель глибокого навчання, запропоновано вдосконалення її архітектури, виконано доналаштування моделі, контроль за перенавчанням та оптимізацію гіперпараметрів. Крім того, визначено метрики для оцінки якості навченої моделі.

У третьому розділі – «Основні кроки вдосконаленого методу для виявлення та розпізнавання номерів автомобілів нейронними мережами» – автором представлено покроковий опис процесу отримання вдосконаленої моделі та процесу класифікації з використанням навченої моделі.

У четвертому розділі – «Експериментальні результати» – магістрантом проведено оцінку ефективності запропонованого методу. Як результати розглянуто порівняння між вдосконалим методом і існуючими моделями. Отримані результати підтвердили переваги запропонованого методу.

6. Ступінь розкриття теми роботи

Тема роботи розкрита достатньою мірою та мета досягнута.

7. Якість оформлення кваліфікаційної роботи

Робота написана літературною мовою та оформлена згідно відповідних вимог достатньо якісно.

8. Недоліки кваліфікаційної роботи

Разом з тим, у роботі є і певні недоліки. Так, у роботі відсутні відповіді на питання щодо ефективності авторського удосконалення на інших наборах даних, крім тих, що досліджені в магістерській роботі.

Проте зазначений недолік загалом не впливає на комплексну позитивну оцінку роботи.

9. Загальний висновок (допускається чи не допускається до захисту), якої оцінки заслуговує кваліфікаційна робота.

Рекомендую допустити роботу до захисту.

Кваліфікаційна робота заслуговує на оцінку «Відмінно».

Опонент (прізвище, ім'я, по батькові, посада, місце роботи)

Говорухинська Т.О., ректор ФІТ ХНУ

«11» 12 2024 р.

Підпис