Plag

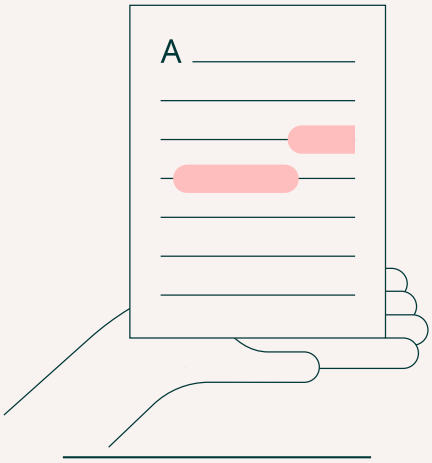
Звіт про оригінальність

● Оцінка схожості 26%

● Ризик плагіату НАЙВИЩИЙ

Ігор Кагало 2025-06-14 10:14

Посилання на звіт: 10bhT / Посилання користувача: qAHy



Ось вона –
Ваша звіт про оригінальність!

Ми раді повідомити, що перевірка вашого документа завершена, і результати вже готові! Наші алгоритми старанно працювали, щоб знайти збіги в наших базах даних.

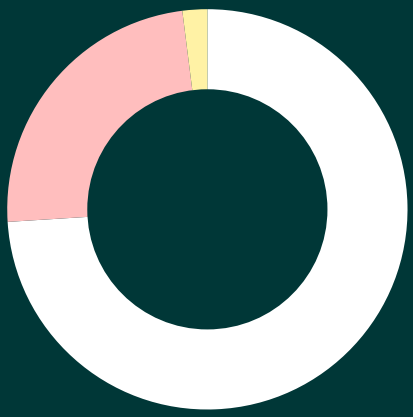
На наступних сторінках ви знайдете результати перевірки:

Бали
Збіги
Посилання

Ваш документ було перевірено за такими джерелами:

- ✓ База даних інтернет-джерел
- ✓ База даних наукових статей
- ○ Глибока перевірка (наш вдосконалений алгоритм)

Бали



Збіги тексту	24%
Перефразування	2%
Цитований текст	0%
Неправильне цитування	0%
Збігів не знайдено	74%

Ризик плагіату НАЙВИЩИЙ

Ризик плагіату вказує, як збіги тексту розподілені по документу. Вищий ризик виникає, коли збіги з'являються близько один до одного, наприклад, у тому самому абзаці або розділі.

Оцінка схожості 26%

Оцінка схожості показує, скільки слів або символів у вашому документі збігаються з текстами інших документів, включаючи перефразовані тексти або неправильні цитати.

Збіги

1 РОЗПІЗНАВАННЯ АВТОМОБІЛЬНИХ ЗНАКІВ ЯК КЛЮЧ ДО БЕЗПЕКИ ДОРОЖНЬОГО РУХУ

1.1 Ключові терміни та визначення розпізнавання автомобільних знаків

У сучасному світі, де рух транспорту невпинно зростає, важко уявити собі ефективне керування ним без технології розпізнавання номерних знаків. Ідентифікація автомобілів за їх реєстраційними номерами відіграє важливу роль у забезпеченні безпеки на дорогах.

Одним зі способів розпізнавання літер та цифр на автомобільних номерах є використання згорткових нейронних мереж. Процедура розпізнавання номера включає п'ять ключових етапів:

Пошук номерної пластини на зображенні,

Відокремлення символів номера,

Розділення номера на окремі символи,

Розпізнавання кожного з цих символів та оцінка ймовірності кожного розпізнаного символу.

Алгоритми розпізнавання повинні враховувати спотворення, що виникають через швидкість руху авто та кут огляду камери.

На рисунку 1.1 можна побачити 1 зразки українських номерних знаків.

1 Рисунок 1 1.1 – 1 Зразки українських номерних знаків

1 Одним з найважливіших аспектів розпізнавання є те, що алгоритми, які розпізнають номерні знаки транспортних засобів, повинні 1 бути стійкими до деформацій зображень номерів, що виникають через швидкість руху автомобіля, кут камери та забруднення номерних знаків.

Крім того, виклик у розпізнаванні автомобільних номерів криється у їхній

різноманітності, яка проявляється у відмінностях форми символів, фону та розмірах номерних знаків. Тому, дані для машинного навчання мають 1 включати велику кількість зображень номерів різних типів. Завдяки проєкту Nomeroff.net, ця проблема більше не є актуальною. Для навчання цієї згорткової мережі було використано їхню базу даних, що містить тисячі 1 реальних та згенерованих автомобільних знаків.

1 Згорткова нейронна мережа, ЗНМ, 13 CNN – це ключовий 1 інструмент для класифікації та розпізнавання об'єктів, облич на фотографіях та розпізнавання мови. Існує багато варіантів використання CNN, зокрема 1 Deep Convolutional Neural Network (DCNN), RegionCNN (RCNN), 1 Fully_Convolutional_Neural_Networks 1 (FCN N), Mask R-CNN та інші.

1 Згорткова нейронна мережа, застосовуючи спеціальну операцію – згортку – дозволяє одночасно зменшити обсяг 1 інформації, що зберігається в пам'яті, що підвищує її ефективність при роботі з зображеннями 1 високої роздільної здатності, а також виділити ключові характеристики зображення, наприклад, краї, 1 контури або грані. На наступному етапі обробки з цих країв 1 та граней можливо 1 розпізнати повторювані фрагменти текстур, які згодом можуть скласти фрагменти зображення. Завдяки системам 1 технічного зору та нейромережі, як-от YOLOv8, можна швидко та точно виявляти об'єкти або автомобілі. YOLOv8 — це один із найпопулярніших алгоритмів для виявлення об'єктів у реальному часі, який використовує глибокі нейронні мережі для класифікації та локалізації об'єктів. YOLOv8 здатний 1 розпізнавати 1 об'єкти на зображеннях та відео з високою швидкістю та точністю. Ця модель може працювати на різних апаратних платформах, включаючи мобільні пристрої та комп'ютери. Якщо системи технічного зору оснащені нейромережами YOLOv8, вони можуть швидко реагувати на автомобілі, що рухаються по 1 дорозі. Алгоритм YOLOV8 побудований 1 на архітектурі згорткових нейронних мереж і використовує методи навчання з учителем. Ця модель приймає зображення як вхідні дані та видає ймовірність наявності певного об'єкта на зображенні в реальному часі. 1 Для цього YOLOV8 застосовує методи визначення областей інтересу (ROI), що дозволяє визначити ділянки зображення, де можуть бути об'єкти. Загальновживаними 1 математичними методами, які використовуються в YOLOV8, є методи згорткової нейронної мережі, навчання з учителем та оптимізації функції втрат. Вони 1 базуються 1 на алгоритмах глибокого навчання, таких як стохастичний градієнтний спуск та зворотне поширення помилки.

1 2 АНАЛІЗ СИТУАЦІЇ ТА ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

У сучасних умовах розвитку інтелектуальних транспортних систем (ІТС) значну роль відіграє автоматичне розпізнавання номерних знаків (АРНЗ). Це дає змогу ефективно керувати транспортними потоками, автоматизувати паркування, впроваджувати безконтактну оплату проїзду та підвищувати загальний рівень безпеки на дорогах.

АРНЗ є прикладною задачею комп'ютерного зору, основна мета якої - виявити та розпізнати символи державного номерного знака транспортного засобу на зображенні або відео [1], [6].

Зазвичай, процес АРНЗ включає в себе наступні етапи:

•визначення розташування транспортного засобу;

•визначення місцезнаходження номерного знака;

•обробка області, що містить номерний знак;

•поділ символів на окремі елементи;

•розпізнавання окремих символів;

•аналіз та коригування результатів розпізнавання.

Загальна структура стандартної системи автоматичного розпізнавання номерних знаків включає декілька етапів: виявлення транспортного засобу, локалізація номерного знака, сегментація символів та оптичне розпізнавання (OCR).

На рисунку 2.1 представлено загальну схему послідовності обробки даних в таких системах.

Рисунок 2.1 – Етапи розпізнавання номерного знаку в системі ALPR

Для впровадження таких систем застосовуються як традиційні методи комп'ютерного зору [2], так і сучасні підходи, що спираються на нейронні мережі [3], [5].

2.1 Аналіз існуючих технологій та інструментів

Серед найпоширеніших знарядь, які залучаються у створенні систем розпізнавання номерів, можна визначити:

• OpenCV — бібліотека комп'ютерного зору, що дозволяє реалізовувати базову обробку зображень, у тому числі фільтрування, морфологічні операції та інші [4].

• Tesseract OCR — оптичне розпізнавання тексту з відкритим вихідним кодом (OCR), здатне до навчання на власних шрифтах та мовах [8].

• YOLO — високошвидкісний алгоритм глибинного навчання, призначений для виявлення об'єктів на зображеннях [3].

• SSD — ще один популярний алгоритм для одночасної детекції багатьох об'єктів, що

базується на CNN [5].

• OpenALPR — програмне забезпечення (відкрите або комерційне) для розпізнавання автомобільних номерів, яке надається з готовими модулями для інтеграції [6].

У таблиці 2.1 наведено порівняльну характеристику популярних інструментів для АНЗ.

Таблиця 2.1. – Порівняння популярних інструментів для АНЗ

Назва

Тип

Детекція знаку

Розпізна-вання знаку

Можливість кастомізації

Відкритий код

OpenCV

Бібліотека CV

Так

Частково (з OCR)

Висока

Так

Tesseract OCR

OCR-бібліотека

Ні

Так

Так

Так

YOLO

Нейромережа

Так

Ні

Висока

Так

OpenALPR

Готове рішення

Так

Так

Обмежено

Частково

EasyOCR

OCR-модуль

Ні

Так

Так

Так

2.2 Огляд готових систем на ринку

Найкращим безкоштовним варіантом для України та Європи є Nomeroff.net.

Nomeroff.net – це дітище розробників, які стояли біля витоків популярного ресурсу авторіа. Вони надихнулись відкриттям МВС України доступу до державних реєстрів номерів. Це дозволило, знаючи номер автомобіля, отримати більше даних про нього, розміщених на сайті. Аналізуючи величезні обсяги інформації впродовж багатьох років, 1 проект Nomeroff.net досяг 97% точності 1 у розпізнаванні 1 українських номерів. Серед переваг варто відзначити безкоштовність проекту, а також наявність версії для пристроїв з невеликими можливостями відеокарти.

Програма SecurOS™ Auto License Plate Recognition (LPR/ANPR) пропонує користувачам безліч унікальних переваг, зокрема здатність з високою точністю фіксувати інформацію про номерні знаки на швидкості до 155 миль/год (250 км/год) та за будь-яких погодних умов: туман, дощ, сніг.

SecurOS™ Auto використовує передові алгоритми глибинного навчання та методи, засновані на шаблонах, для досягнення високої точності, розрізняючи літери від цифр, щоб, наприклад, «8» не було помилково прийнято за «B», що є критичною перевагою, особливо в ситуаціях, коли на це є лише мілісекунди.

Також ця програма підтримує номери з більшості країн світу, має модифікації для низької та високої швидкості, обробляє декілька смуг з однієї камери та багато інших функцій. SecurOS™ Auto легко інтегрується з сторонніми системами управління паркуванням або розумними дорожніми системами, а також з застарілим обладнанням безпеки та зовнішніми базами даних. Відчуйте всю міць розпізнавання номерних знаків (LPR/ANPR) з SecurOS™ Auto, вашим надійним вибором у технологіях розпізнавання номерних знаків.

Функціональність системи:

• Розпізнавання номерних знаків з понад 60 країн світу;

• Розпізнавання номерів з окремих кадрів без відео;

• Швидке налаштування оновлень для розпізнавання номерів нових стандартів;

• Формування бази даних розпізнаних номерів зі збереженням супутньої інформації про дату, час і місце виявлення автомобіля, швидкість та напрямок його руху, а також посилання на відеофрагмент;

• Організація пошуку розпізнаних раніше номерів за заданими параметрами;

• Налаштування необхідних реакцій системи на розпізнавання номера, на результати пошуку;

• Оперативне інформування оператора (підтримується, зокрема, і голосове сповіщення) та/або надсилання повідомлення (e-mail, SMS тощо) зовнішнім службам про результати розпізнавання номера та/або про результати зіставлення даних про автомобіль зі списками номерів;

• Автоматичне генерування звітів різних видів на основі результатів розпізнавання, пошуку по базах даних.

Можливості інтеграції та спільного використання:

• Стаціонарне та мобільне обладнання спостереження та прилади контролю й обліку: бар'єри та шлагбауми, автоматичні ворота, радари, телекомунікаційні мережі, термінали оплати тощо;

• Сторонні програмні комплекси (автоматизовані системи розрахунку оплати, СУБД та ін.);

• Інтеграція з «SecurOS Traffic Scanner» (системою автоматичного визначення фактів порушення правил дорожнього руху) – можливість побудувати потужне комплексне рішення для контролю та управління ситуацією на дорогах.

TesseractOCR – це програмне забезпечення має відкритий код, і його мета – автоматичне розпізнавання символів та цілісного тексту. Серед переваг Tesseract варто відзначити його кросплатформність, простоту в опануванні та передбачуваність функціонування. Водночас недоліком є труднощі з обробкою тексту, що містить розриви, забруднення чи спотворення. Приблизний відсоток успішно розпізнаних автомобільних номерів коливається в діапазоні 20-30%.

На рисунку 2.2 наведено логотип TesseractOCR.

Рисунок 2.2 – Tesseract OCR

k-nearest – цей метод розпізнавання, незважаючи на простоту, часто демонструє добрі показники.

Принцип роботи:

Заздалегідь було підготовлено й відібрано певну кількість зображень.

Встановлено вимірювання відстані між символами (якщо зображення бінарне, то операція XOR є найкращим варіантом).

При спробі розпізнати літери, дистанції обчислюються почергово. Між символом, що розпізнається, і всіма основними символами. Серед найближчих сусідів можуть бути представники різних класів. Символ відноситься до того самого класу, що й у переважної більшості найближчих символів.

Недоліком методу є потреба у швидкості обчислень. Необхідно обчислити відстань між зображеннями, конвертувати їх у двійковий формат та застосувати XOR. Бінаризація модифікує символи доволі непередбачувано. Зашумлені або стерті номери можуть викликати труднощі. Проте, за наявності великої бази даних з прикладами символів, зібраними за різних обставин, k-nearest – це теоретично найоптимальніше рішення.

1 Штучна нейронна мережа (ШНМ, нейромережа) – це сукупність нейронів, які взаємодіють між собою. Основна задача нейронної мережі – трансформація інформації: вона перетворює вхідні вектори на вихідні.

Практично 1 будь-яке завдання можна представити таким чином, щоб його розв'язала нейронна мережа. Для старту роботи мережу потрібно підготувати. Спочатку в нейронну мережу заносять певний зразок, потім подають вхідний сигнал. Відбувається 1 порівняння з базовою моделлю, 1 і розбіжність між зразком 1 та вхідними даними корегує зразок. Чим більше використано навчальних даних, тим ефективніше відбувається навчання нейронної мережі.

Переваги ШНМ:

• При правильних параметрах та належному навчанні, мережа працює результативніше, ніж інші методи.

• За наявності великої кількості навчальних даних підвищується стійкість до неточностей у символах.

Недоліки ШНМ:

• Складність процесу навчання;

• Відсутність можливості виявлення відхилень в роботі багатошарових мереж.

Рис. 2.3. Приблизна схема роботи нейронної мережі.

З ЗАСОБИ РОЗРОБКИ

3.1 1 Огляд середовища розробки

1 Програмування в PyCharm має беззаперечні переваги, і ось кілька реальних ситуацій, 1 чому варто обрати саме 1 цей інструмент, особливо для студентів, враховуючи безкоштовну ліцензію від JetBrains, з якою кодити — суцільне задоволення.

У PyCharm 1 ви можете легко створювати та управляти 1 проєктами. Зручний інтерфейс дозволяє оперативно 1 додавати файли, бачити структуру проєкту та без 1 зайвих зусиль 1 перемикатися між різними частинами коду.

1 PyCharm 1 автоматично виявляє помилки безпосередньо 1 під час написання коду. Скажімо, 1 якщо ви використаєте 1 неправильний синтаксис або оголосите 1 змінну, якої 1 не існує, PyCharm підсвітить 1 це та запропонує способи виправлення.

1 Під час написання коду PyCharm пропонує 1 автодоповнення, що значно збільшує

1 продуктивність. Наприклад, ви можете швидко обрати доступні методи або властивості об'єкта, просто 1 ввівши 1 першу літеру.

1 Інтеграція Git в PyCharm спрощує відстеження змін коду, фіксацію, розгалуження та вирішення конфліктів за допомогою зрозумілого інтерфейсу.

Ви маєте змогу легко налаштовувати 1 віртуальні середовища Python та керувати ними безпосередньо з PyCharm. Це дає 1 змогу ефективно управляти 1 залежностями вашого проєкту.

1 PyCharm допомагає проаналізувати код та визначити потенційні покращення. Наприклад, ви можете скористатися 1 функціями 1 рефакторингу для автоматичного перейменування змінних, методів чи 1 класів.

1 Якщо ви розробляєте веб-проєкт, PyCharm може надати інструменти для роботи з HTML, CSS та JavaScript. Крім того, ви можете легко налаштувати інші інструменти веб-розробки та фреймворки.

1 Ці приклади демонструють, як PyCharm полегшує роботу розробників та надає широкий спектр функцій для підвищення продуктивності.

1 На рисунку 3.1 продемонстровано приклад програми в середовищі розробки IDE PyCharm.

10 Рисунок 3.1 – Приклад програми в середовищі розробки IDE PyCharm.

3.2 1 Мови програмування, фреймворки, розмітка 1 та стилі

1 Python – це 1 потужна мова програмування, що 10 здобула популярність завдяки своїй легкості в опануванні. Вона відкриває простір для ефективного використання складних структур даних та надає простий, але водночас потужний 1 підхід до об'єктно-орієнтованого програмування. Гармонійне поєднання елегантного синтаксису з 1 динамічною типізацією 1 інтерпретованої мови робить Python оптимальним рішенням для швидкого створення сценаріїв та розробки додатків, які охоплюють широкий спектр галузей і платформ.

Інтерпретатор Python та стандартна бібліотека є безкоштовними і доступні як у вихідному коді, так і у вигляді бінарних файлів для всіх ключових 1 платформ на офіційному сайті Python, з правом вільного розповсюдження. На цьому ж веб-сайті можна знайти 1 дистрибутиви та посилання на численні сторонні модулі для Python, різноманітні програми та інструменти, а також розширену документацію. Інтерпретатор Python має можливість легкого розширення завдяки інтеграції нових функцій і типів даних, розроблених 1 на C/C++ (або інших мовах, сумісних з C). Крім

того, Python успішно використовується як мова розширення для спеціалізованого програмного забезпечення.

SORT – це базове втілення візуальної системи для відслідковування багатьох об'єктів, що спирається на ключові підходи асоціації даних та оцінки стану. Створена для застосувань онлайн-стеження, де доступні лише минулі та поточні кадри, методика миттєво видає ідентифікатори об'єктів. Хоча цей спрощений трекер не керує оклюзіями чи повторним появом об'єктів, його завдання – служити вихідною точкою та середовищем для експериментів при розробці майбутніх трекерів.

Захоплююча бібліотека, яка є одним з ключових складників цієї програми, дозволяє стежити за кількома об'єктами одночасно, наприклад, за машиною та її номерним знаком, що унеможливлює спотворення даних, коли номерний знак реєструється без автомобіля.

YOLO (You Only Look Once) – популярна модель для розпізнавання об'єктів та поділу зображень, розроблена Джозефом Редмоном та Алі Фархаді з Університету Вашингтона. Випущена у 2015 році, YOLO миттєво здобула славу через свою швидкість та влучність.

YOLOv8 – найновіша варіація YOLO від Ultralytics. Як сучасна модель (SOTA), YOLOv8 базується на успіху попередніх версій, демонструючи нові функції та покращення для збільшення продуктивності, гнучкості та ефективності. YOLOv8 підтримує різноманітні завдання штучного інтелекту для зору, включаючи виявлення, сегментацію, визначення пози, відстеження та класифікацію. Ця універсальність дозволяє користувачам застосовувати 1 можливості YOLOv8 у різноманітних додатках та сферах.

На рисунку 3.2. можна побачити архітектуру Yolov8.

Рисунок 3.2 – Архітектура Yolov8

Звичайно, складно заперечувати, що Django — це могутній монстр у розробці, але й Flask не відстає, впевнено тримаючись на плаву попри будь-які зміни трендів. 1 Обидва фреймворки мають власні 1 переваги, і оптимальний 1 вибір залежить від особливостей завдання. Ось декілька причин віддати перевагу Flask 1 для невеликих проектів:

1 • Легкість у вивченні: Flask — мікрофреймворк, який надає лише найнеобхідніший функціонал для веб-розробки. Якщо проект невеликий та не потребує всього розмаїття можливостей 1 Django, Flask може виявитись значно простішим для вивчення та застосування.

• Гнучкість: Flask забезпечує широке поле для вибору інструментів та бібліотек. На

відміну від Django, який має строго регламентовану структуру, Flask дозволяє формувати компоненти згідно з потребами конкретного проекту.

• Прозорість: Flask має більш зрозумілу структуру, що оцінять ті, хто надає перевагу контролювати організацію коду та управління програмою.

• Швидкість розробки: В окремих ситуаціях 1 Flask може забезпечити вищу швидкість розробки завдяки своїй простоті та мінімалізму. Не пропонуючи настільки широкого функціоналу, він виглядає привабливим рішенням для маленьких проектів.

На рисунку 3.3 продемонстровано архітектуру Flask.

Рисунок 3.3 – Архітектура Flask

3.3 1 Опис програмної реалізації

1 Опис алгоритму:

1 1. Захоплення та попередня обробка відео: Система стартує з отримання вхідного відеопотоку. На цій стадії може відбуватися попередня обробка кадрів, така як прибирання шумів, корекція освітлення чи стабілізація зображення. Це потрібно для поліпшення якості вхідних даних і оптимізації наступних етапів обробки.

2. Виявлення об'єктів інтересу та їх сегментація: Кожен оброблений кадр аналізується з метою знаходження зон з автомобілем та потенційним номером. Для цього використовуються передові методи комп'ютерного зору, наприклад, глибоке навчання (deep learning) або класичні алгоритми розпізнавання об'єктів. Після знаходження, області з номером та авто точно виокремлюються для подальшої роботи.

3. Розпізнавання символів номерного знаку: Кожна визначена область з номером піддається процесу оптичного розпізнавання символів (OCR). Цей етап включає в себе підготовку зображення номерного знаку (скажімо, бінаризація, вирівнювання), розподіл окремих символів та їх класифікацію за допомогою навчених моделей.

4. Агрегація 1 та аналіз даних з відеопотоку: Після розпізнавання номеру з окремих кадрів, отримані дані об'єднуються та досліджуються в контексті усього відеопотоку. Це може передбачати відсіювання невірних розпізнавань, відслідковування номера на протязі серії кадрів, а також виявлення тенденцій чи аномалій.

5. Просторова та часова інтерполяція результатів: Для збільшення надійності та точності кінцевих даних може використовуватися інтерполяція. Просторова інтерполяція здатна заповнювати пропущені чи нечіткі частини номерного знаку, а часова інтерполяція дозволяє згладжувати результати розпізнавання між сусідніми

кадрами, зменшуючи вплив випадкових похибок.

6. Візуалізація результатів 1 на поточному кадрі: Розпізнаний номерний знак наноситься на відповідну область автомобіля у поточному кадрі. Це може включати показ розпізнаного тексту, обведення ділянки номера рамкою чи інші візуальні маркери для наочної демонстрації результатів.

7. Створення підсумкового відео: На завершальному етапі послідовність оброблених кадрів, де відображено розпізнані номерні знаки, збирається в результуючий відеофайл. Цей файл є результатом роботи системи, демонструючи процес розпізнавання номерних знаків в динаміці.

На рисунку 3.4 зображено 1 опис програмної реалізації.

1 Обробка вхідних відео

1 та отримання набору кадрів

1 Відображення

1 номерного знаку на поточному кадрі

1 Створення

1 результуючого відео з опрацьованих кадрів

1 Сегментація кадрів,

1 виділення областей з номерним знаком та автомобілем

1 Інтерполяція значень

1 для покращеня результатів виводу даних

1 Розпізнавання

1 номерного знаку з кожної виділеної області

1 Опрацювання та

1 аналіз даних з кожного відеопотоку

1 Обробка вхідних відео

1 та отримання набору кадрів

1 Відображення

1 номерного знаку на поточному кадрі

1 Створення

1 результуючого відео з опрацьованих кадрів

1 Сегментація кадрів,

1 виділення областей з номерним знаком та автомобілем

1 Інтерполяція значень

1 для покращеня результатів виводу даних

1 Розпізнавання

1 номерного знаку з кожної виділеної області

1 Опрацювання та

1 аналіз даних з кожного відеопотоку

1 Рисунок 1 3.4 – 1 Опис програмної реалізації

1 YOLO (You 8 Only Look Once) є передовим алгоритмом 7 виявлення об'єктів у реальному часі, що відзначається високою продуктивністю та значним розмаїттям застосувань у сфері комп'ютерного бачення. Розроблений на основі новаторських ідей наукової спільноти, YOLO став одним з найбільш вживаних підходів до розв'язання класичної задачі детекції 7 об'єктів на зображеннях.

7 На відміну від завдання класифікації, яке лише визначає наявність конкретних об'єктів на зображенні, детекція об'єктів є значно складнішим завданням, адже передбачає не тільки ідентифікацію об'єктів, а й точне визначення їхнього розташування за допомогою обмежувальних рамок. Класифікаційні моделі також не можуть ефективно обробляти зображення, що містять кілька об'єктів різних класів.

Однією з головних переваг YOLO є його здатність досягати високої точності детекції при високій швидкості обробки. Принцип роботи алгоритму "дивись лише один раз" полягає в тому, що для здійснення прогнозу йому необхідний 1 лише один прохід через нейронну мережу. Після 1 цього 1 розпізнані об'єкти відображаються на зображенні разом з відповідними обмежувальними рамками.

В його основі лежить використання згорткової нейронної мережі (CNN), яка

8 одночасно прогнозує множинні обмежувальні рамки та ймовірності належності кожної рамки до певного класу. Цей підхід 11 має низку суттєвих переваг порівняно з іншими методами детекції об'єктів:

• Висока швидкість обробки: Завдяки однократному проходженню через мережу, YOLO демонструє значно вищу швидкість обробки в порівнянні з методами, які потребують багаторазового аналізу зображення.

• Глобальне сприйняття контексту: 5 Під час навчання та тестування YOLO аналізує зображення цілісно, що дозволяє йому неявно кодувати контекстуальну 5 інформацію про об'єкти та їхні взаємозв'язки.

• Узагальнення ознак об'єктів: YOLO здатний вивчати загальні риси об'єктів, що робить його більш стійким до нових або штучно згенерованих даних в порівнянні з іншими методами детекції.

Принцип роботи YOLO базується на поділі вхідного 4 зображення на сітку розмірністю SxS. 5 Якщо центр об'єкта потрапляє в певну 5 комірку цієї сітки, то саме 5 ця комірка 4 відповідає за виявлення даного об'єкта. Для кожної комірки сітки YOLO одночасно виконує завдання класифікації (визначення класу об'єкта) та локалізації (визначення положення обмежувальної рамки).

Архітектура YOLO на основі сітки SxS також має певні обмеження:

1. Обмежена кількість об'єктів на комірку: Оскільки 4 кожна комірка сітки відповідає лише за виявлення одного об'єкта, максимальна кількість об'єктів, яку може виявити модель у межах однієї комірки, становить одиницю.

2. Проблеми з перекриттям об'єктів: Якщо в одній комірці сітки знаходяться центри кількох об'єктів, YOLO може мати труднощі з їхнім окремим виявленням.

3. Повторне виявлення одного об'єкта: Об'єкт, який займає значну площу на зображенні, може потрапляти в кілька сусідніх комірок сітки, що може призвести до його багаторазового виявлення.

Незважаючи на ці обмеження, YOLO залишається одним з найбільш ефективних алгоритмів виявлення об'єктів завдяки тому, що всі комірки сітки SxS аналізуються одночасно. Для кожної передбачуваної обмежувальної рамки модель також прогнозує оцінку вірогідності (confidence score), яка вказує на ймовірність того, що в даній рамці справді знаходиться об'єкт. Ця оцінка допомагає відфільтрувати помилкові виявлення фонових областей. Якщо в комірці немає жодного об'єкта, то оцінка вірогідності наближається до 0.

Крім оцінки вірогідності, модель також генерує чотири числа ((x_1, y_1), x_2, y_2) для визначення положення та розміру запропонованої обмежувальної рамки. 4 Координати (x_1, y_1) представляють центр обмежувальної рамки відносно верхнього лівого кута відповідної комірки сітки. Ширина та висота рамки (x_2, y_2) нормалізовані відносно розмірів усього зображення, тому їхні значення знаходяться в діапазоні 0 < (x_1, y_1, x_2, y_2) < 1.

3.4 Модель згорткової мережі

На представлених графіках (рисунок 3.5; рисунок 3.6) відображено динаміку навчання моделі протягом 200 епох.

Рисунок 3.5 – Графік точності

Рисунок 3.6 – Графік втрат

Аналіз отриманих результатів вказує на те, що для тестової вибірки було досягнуто точність у 91% (0.91). Це значення свідчить про здатність навченої моделі правильно класифікувати нові дані, які раніше не бачила. Функція втрат для тестової вибірки дорівнює 0.32. Це відображає ступінь розбіжності між прогнозами моделі та фактичними значеннями на тестових даних. Більше значення функції втрат 3 на тестовій вибірці порівняно з навчальною може вказувати на можливе перенавчання моделі або на складнішу структуру тестових даних.

Щодо тренувальної вибірки, точність моделі сягнула 90% (0.9), а значення функції втрат становить лише 0.035. Низьке значення функції втрат на тренувальній вибірці свідчить про добру адаптацію моделі до даних навчання. Однак, дещо вища функція втрат 3 на тестовій вибірці при трохи вищій точності може натякати на необхідність подальшої оптимізації моделі задля кращого узагальнення на нових даних.

Додаткові спостереження та можливі інтерпретації:

• Для глибшого розуміння процесу навчання було б корисним проаналізувати графіки зміни точності та функції втрат протягом усіх 200 епох. Це дозволило б оцінити, чи модель вийшла на стабільний стан, чи продовжує вдосконалюватися, або ж спостерігаються ознаки перенавчання (зростання функції втрат 3 на тестовій вибірці при подальшому зростанні точності на тренувальній).

• Наявний певний розрив між показниками точності та 3 функції втрат на навчальній та тестовій вибірках. Хоча різниця в точності незначна (1%), суттєво вища функція втрат 3 на тестовій вибірці може свідчити про те, що модель краще "запам'ятала" навчальні дані, ніж навчилася узагальнювати закономірності.

Подальші кроки: Для поліпшення продуктивності моделі можна розглянути наступні кроки:

Регуляризація: Застосування методів регуляризації (наприклад, dropout, weight decay) для зменшення перенавчання.

Збільшення обсягу навчальних даних: Забезпечення моделі більшою кількістю різноманітних навчальних прикладів може підвищити її здатність до узагальнення.

Налаштування гіперпараметрів: Експерименти з різними значеннями гіперпараметрів (наприклад, швидкість навчання, розмір батчу) можуть сприяти кращій збіжності моделі.

Оцінка інших метрик: Розгляд інших метрик оцінювання якості моделі (наприклад, precision, recall, F1-score) може надати повніше уявлення про її продуктивність, особливо у випадку незбалансованих класів.

Таким чином, представлені результати навчання є досить обнадійливими, демонструючи високу точність моделі. Проте, подальший аналіз динаміки навчання та вжиття заходів для зменшення розриву між навчальною та тестовою вибірками можуть призвести до ще кращих результатів узагальнення на нових даних.

Основними будівельними блоками архітектури роздробленої моделі є такі типи шарів:

• Згорткові шари (Conv2D) є ключовими компонентами для обробки зображень. Вони застосовують набір фільтрів (ядер згортки) до вхідного зображення, виконуючи операцію згортки. Це дозволяє моделі автоматично виявляти локальні патерни, як-от краї, кути, текстури та інші візуальні особливості на різних рівнях абстракції. Кількість фільтрів, розмір ядра згортки, крок згортки (stride) та функція активації є важливими гіперпараметрами, які визначають поведінку згорткових шарів.

• Шари максполінгу використовуються для зменшення просторової розмірності вихідних карт ознак (feature maps) зі згорткових шарів. Це допомагає знизити обчислювальну складність наступних шарів, а також робить модель 6 більш стійкою до незначних зсувів та спотворень вхідного зображення. Операція максполінгу вибирає 6 максимальне значення в межах певного вікна (pooling window) на карті ознак. Розмір цього вікна та крок його переміщення є основними гіперпараметрами шарів максполінгу.

• Шари Dropout є методом регуляризації, який випадковим чином вимикає певну частку нейронів 6 під час навчання. Це запобігає надмірній адаптації моделі до навчальних даних (перенавчанню) та сприяє кращому узагальненню на нових,

невідомих даних. Імовірність виключення нейронів (dropout rate) є гіперпараметром цього шару.

• Шар Flatten перетворює багатовимірні вихідні дані попередніх шарів (наприклад, тривимірні карти ознак після згорткових шарів) на одновимірний вектор. Це необхідно для подачі даних до повнозв'язних шарів, які очікують на вхід векторної форми.

• Повнозв'язні шари (Dense), також відомі як лінійні або з'єднані шари, де 9 кожен нейрон у шарі з'єднаний з кожним нейроном у попередньому шарі. Ці шари використовуються для виконання складних нелінійних перетворень над ознаками, отриманими на попередніх етапах, і врешті-решт приймають рішення щодо класифікації або регресії. Кількість нейронів у кожному повнозв'язному шарі та функція активації є важливими гіперпараметрами.

На рисунку 3.7 детально показано архітектуру розробленої моделі, яка складається з послідовності різноманітних шарів нейронної мережі, кожен з яких виконує певну функцію в процесі обробки та аналізу вхідних даних.

Рисунок 3.7 – Архітектура моделі

4 РОЗРОБКА ІНТЕРФЕЙСУ ТА ФУНКЦІОНАЛУ ВЕБ-ДОДАТКУ

4.1 Робота з веб-додатком

Щоб розпочати роботу з функціоналом програми, користувачеві 12 слід створити новий обліковий запис, пройшовши реєстрацію, або ж увійти до наявного. Зовнішній вигляд екрану, який відповідає за реєстрацію та вхід користувача, показано на рисунку 4.1.

Рисунок 4.1 – Вхід до додатку

Після запуску процесу реєстрації через натискання відповідної клавіші, перед користувачем з'явиться інтерфейс для введення реєстраційних даних, чітко представлений на рисунку 4.2. Якщо користувач вже має акаунт в системі, йому потрібно скористатися функцією входу.

Рисунок 4.2 – Форма реєстрації

Натискаючи кнопку "Логін", відкриється вікно діалогу, яке ілюструє рисунок 4.3, де користувачеві потрібно ввести свої дані для входу – логін та пароль – для подальшої авторизації та отримання доступу до можливостей застосунку.

Рисунок 4.3 – Форма авторизації

Користувачі мають альтернативний спосіб миттєвої реєстрації та входу в додаток, скориставшись інтеграцією зі своїм акаунтом у соціальній мережі GitHub. Щоб скористатися цим варіантом, потрібно перейти на екран 1 входу або реєстрації та вибрати іконку GitHub, що проілюстровано на рисунку 4.4.

Рисунок 4.4 – Меню 1 вибору реєстрації за допомогою GitHub

1 Далі користувачу запропонують підтвердити дозвіл додатку на доступ до основних даних профілю GitHub для завершення процедури реєстрації або входу.

Після вибору авторизації через GitHub, користувач автоматично потрапить на сторінку авторизації цієї соцмережі. На рисунку 4.5. наочно показано, як відбувається надання дозволу застосунку на доступ до акаунта користувача в GitHub. Це потрібно для успішної реєстрації або входу в систему.

Рисунок 4.5 1 — Реєстрація за допомогою платформи GitHub

1 Для початку роботи із системою користувачам потрібно буде завантажити один чи декілька відеофайлів, які підтримуються 1 у форматі ".MOV" або ".MP4". Інтерфейс завантаження файлів показано на рисунку 4.6.

Рисунок 4.6 – 1 Вікно завантаження файлів

1 Після успішного завантаження користувачі зможуть переглянути детальну інформацію про кожен файл. Ця інформація містить назву та формат, як показано на рисунку 4.7.

Рис. 4.7. 1 Назва та тип завантажених файлів

1 Якщо буде виявлено помилково завантажені файли, користувачі матимуть можливість 1 видалити всі завантажені відео одним натисканням, як зображено на рисунках 4.8 та 4.9.

Рисунок 4.8. 1 — Кнопка видалення всіх завантажених відео-файлів

1 Рисунок 4.9. 1 — Кнопка початку опрацювання всіх завантажених відео-файлів

1 4.2. 1 Процес опрацювання відео згортковою нейронною мережею

На первинному етапі обробки завантаженого відео, згорткова нейронка проводить покадровий розбір. Завдання цього етапу полягає в розпізнаванні на кожному окремому кадрі зображень автомобілів та їх номерних знаків. Після вдалого розпізнавання інформація про кожен автомобіль та його номер фіксується у форматі .csv. Кожному об'єкту присвоюється унікальний ідентифікатор (ID) для наступного

відстеження та аналізу. Візуалізація процесу покадрової обробки і приклад збереження даних у файл .csv зображені на рисунку 4.10 та рисунку 4.11 відповідно.

Рисунок 4.10 – 1 Файл з результатами розпізнавання

1 Рисунок 1 4.11 – 1 Файл з результатами розпізнавання

1 На наступному етапі, після впорядкування ідентифікованих машин за унікальними номерами (ID), стає видимою одна з особливостей функціонування нейронної мережі – пропускання об'єктів на кадрах, що йдуть один за одним. Транспортний засіб може час від часу не розпізнаватися на певних фреймах відеоряду. Цей факт, хоча й не критичний для точності розпізнавання номерів та фінального результату, може дещо впливати на візуальне відчуття безперервності відстеження об'єкта. Ці "провали" в ідентифікації є характерним наслідком особливостей роботи алгоритмів обробки зображень та можуть бути зумовлені різкою зміною освітлення, частковим перекриттям об'єкта іншими компонентами сцени або іншими умовами, що ускладнюють його розпізнавання на окремих кадрах.

На рисунках 4.12 та 4.13 можна побачити файли з відсортованими результатами розпізнавання.

Рисунок 4.12 – Файл з відсортованими результатами розпізнавання

Рисунок 4.13 – Файл з відсортованими результатами розпізнавання

Для мінімізації візуальних недоліків, які виникають через пропуски при визначенні об'єктів на окремих кадрах, а також для забезпечення більш плавного відображення результатів, в системі використовується інтерполяція даних. Цей метод передбачає усереднення інформації про положення та розміри виявлених об'єктів між тими кадрами, де їх вдалося успішно розпізнати. Використання інтерполяції дозволяє сформувати більш стабільну та візуально комфортну відеопослідовність з накладеними результатами розпізнавання. Наочний приклад ефекту від застосування інтерполяції до вхідних даних можна спостерігати на рисунку 4.14. та рисунку 4.15., де помітно згладжування траєкторії руху об'єкта і усунення різких перепадів, які могли б з'явитися через пропущені детекти.

Рисунок 4.14 – Файл з результатами розпізнавання після інтерполяції.

Рисунок 4.15 – Файл з результатами розпізнавання після інтерполяції

Внаслідок обробки відео, що було подане на вхід, користувач отримує на виході відеофайл. Тривалість цього файлу повністю відповідає тривалості вихідного відео. На цьому відео візуально демонструється процес розпізнавання номерних знаків всіх

транспортних засобів, які було виявлено протягом усього запису.

Окрім кінцевого відео, користувачеві також надаються два додаткових файли з даними. Перший містить інформацію до застосування інтерполяції, а другий – після. Ці файли містять вичерпну інформацію про кожен розпізнаний номерний знак, включно з ймовірністю коректного розпізнавання, а також рівнем впевненості нейронної мережі в цьому розпізнаванні.

Приклади візуалізації вихідного відео та фрагментів файлів даних до та після інтерполяції представлені на рисунках 4.16, 4.17 та 4.18 відповідно.

Рисунок 4.16 – Результати візуалізації

Рисунок 4.17 – Результати візуалізації.

Рисунок 4.18 – Результати візуалізації.

Посилання

Це джерела виділених збігів у вашому документі. Кожен збіг позначено темно-зеленим числом, яке відповідає вказаному тут джерелу. Джерела впорядковані за схожістю — чим вищий бал, тим сильніше збіг.

#	Джерело	%
1	dp.knute.edu.ua	17.8%
2	ela.kpi.ua	3.0%
3	krs.chmnu.edu.ua	0.5%
4	dspace.znu.edu.ua	0.4%
5	iq.vntu.edu.ua	0.3%
6	ela.kpi.ua	0.3%
7	ds.knu.edu.ua	0.3%
8	dnu.dp.ua	0.2%
9	essuir.sumdu.edu.ua	0.2%
10	ela.kpi.ua	0.2%
11	dspace.msu.edu.ua	0.2%
12	secunda.com.ua	0.1%
13	er.chdtu.edu.ua	0.0%

Plag

Дякуємо, що перевірили
свій документ за допомогою
Plag!