КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет комп'ютерних наук та кібернетики
Кафедра математичної інформатики

Кваліфікаційна робота

на здобуття ступеня бакалавра
за спеціальністю 122 Комп'ютерні науки
на тему:

ВИКОРИСТАННЯ ЗМАГАЛЬНОГО МАШИННОГО НАВЧАННЯ В СИСТЕМАХ СПРИЙНЯТТЯ АВТОНОМНОГО КЕРУВАННЯ АВТОМОБІЛЕМ ЗА ДОПОМОГОЮ 3D ГЕНЕРАТИВНОЇ МОДЕЛІ

Виконала студентка 4-го курсу
Коваленко Вікторія Андріївна _____
(підпис)

Науковий керівник:
асистент
Бобиль Богдан Володимирович
_____
(підпис)

Консультант:
доцент кафедри математичної інформатики,
кандидат фізико-математичних наук
Деревянченко Олександр Валерійович _____
(підпис)

Засвідчую, що в цій роботі немає запозичень з праць інших авторів без відповідних посилань.

Студент _____
(підпис)

Роботу розглянуто й допущено до захисту на засіданні кафедри математичної інформатики
«_____»_____2023 р.,
протокол № _

Завідувач кафедри М.В. Терещенко _____
(підпис)

Київ - 2023

РЕФЕРАТ

Обсяг роботи 48 сторінок, 14 ілюстрацій, 2 таблиці, 48 джерел посилання.

АВТОНОМНЕ КЕРУВАННЯ АВТОМОБІЛЕМ, ГЛИБОКЕ МАШИННЕ НАВЧАННЯ, МОДЕЛІ ВИЯВЛЕННЯ ОБ'ЄКТІВ, ЗГОРТКОВІ НЕЙРОННІ МЕРЕЖІ, ЗМАГАЛЬНЕ МАШИННЕ НАВЧАННЯ, КОМП'ЮТЕРНИЙ ЗІР, МАШИННЕ НАВЧАННЯ, МОДЕЛЬ СПРИЙНЯТТЯ, 3D ТЕКСТУРОВАНІ ФОРМИ.

Об'єктом роботи є 3D-генеративні моделі для синтезу текстурованих 3D-сіток, а також моделі для виявлення об'єктів на зображенні. Предметом роботи є створення складних тренувальних прикладів для моделі сприйняття автономного автомобіля та аналіз моделей виявлення об'єктів для згенерованого набору даних.

Метою роботи є реалізація конвеєру генерації змагальних даних, створення змагальних RGB-зображень автомобілів, згенерованих за допомогою конвеєра, та проведення експериментів з виявлення об'єктів на моделях, натренованих на синтезованих незмагальних і змагальних зображеннях.

Методи розроблення: налаштування програмного середовища та бібліотек для виявлення об'єктів на зображенні. Інструменти розроблення: Python3, PyCharm IDE, Microsoft VSCode, Google Colab, Google Drive, PyTorch.

Результати роботи: проаналізовано останні досягнення в галузі 3D-генеративних моделей та змагального навчання, розроблено конвеєр для генерації змагальних даних, показано якісні та кількісні результати виявлення

об’єктів на зображеннях.

Дана робота пропонує потенційні вдосконалення для систем сприйняття автономних транспортних засобів, зокрема, зосередження на підвищенні надійності та стійкості систем виявлення об'єктів у реальних умовах. Результати, проведені в дослідженні, можуть бути підґрунтям для покращення якості згенерованих синтетичних зображень, а також їх більшої різноманітності.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ

AV – Autonomous Vehicle, Автономний транспортний засіб;

CNNs – Convolutional Neural Networks, Згорткові нейронні мережі;

DNNs – Deep Neural Networks, Глибинні нейронні мережі;

DMTet – Deep Marching Tetrahedra, Генеративна модель для тривимірних зображень;

GANs – Generative Adversarial Network, Генеративні змагальні мережі;

LiDAR – Light Detection and Ranging, Виявлення світла та визначення дальності, Лідар, далекомір оптичного діапазону;

LSTM – Long Short Term Memory, довга короткочасна пам’ять;

MSE – Mean Squared Error, середньоквадратична похибка;

NLP – Natural Language Processing, Обробка природної мови;

RNNs – Recurrent Neural Networks, Рекурентні нейронні мережі;

RoI – Region of Interest, Область інтересу;

RPN – Region Proposal Network, Мережа регіональних пропозицій;

SDF – Signed Distance Field, Підписане поле відстані;

YOLO – You Only Look Once, Нейронна мережа для виявлення об’єктів на зображенні.

ВСТУП

Оцінка сучасного стану об’єкта розробки. Автономні транспортні засоби (AVs) все частіше застосовуються і продовжують використовуватися у дуже складних навколишніх середовищах. Моделі сприйняття відіграють важливу роль у системах керування автономних транспортних засобів. Зокрема таке програмне забезпечення використовується для виявлення та сегментації об’єктів на фото та відео, як-от автомобілі, пішоходи, велосипедисти, дорожні знаки та ін. Використовуючи дані зображень, взятих з камери та хмар 3D точок з лідар пристроїв, такі моделі також можуть визначити місцерозташування та передбачення майбутніх рухів таких об’єктів. Зазвичай, стеки автономних автомобілів працюють лінійно, з декількома субмодулями: сприйняття, прогнозування, планування та керування [1]. Завдяки прямому зв'язку між модулями стека будь-які помилки в попередніх модулях, таких, як сприйняття, можуть мати значний вплив на наступні. Такі помилки можуть мати фатальні наслідки, спричиняючи дорожньо-транспортні пригоди на дорогах.

У складних середовищах модулі сприйняття стикаються з багатьма проблемами, такими, як часткова спостережуваність, оклюзія та дані поза розподілом; граничні випадки, які важко повністю врахувати в реальних наборах даних через небезпеку, яку вони становлять для водіїв. Таким чином, дуже важливо розглянути альтернативні підходи до навчання моделей і куруванню наборів даних.

На даний момент більшість методів вирішують вищезазначені проблеми, намагаючись вводити в оману звичайні AV-датчики (LiDAR, RGB камери тощо), як правило, шляхом підміни невидимих перешкод і/або створення навмисно складних текстур та візерунків.

Актуальність роботи та підстави для її виконання. Перші прототипи безпілотних автомобілів з’явилися у 80-х роках минулого сторіччя. Значний

розвиток та інтерес науковців безпілотні автомобілі отримали в XXI столітті з розвитком машинного навчання та комп'ютерного зору. Технологія Delphi, розроблена в 2015 році, була одним з найперших визначних досягнень у сфері безпілотного керування автомобілями. Автомобіль, що був протестований на даній системі, зміг проїхати більше 5000 кілометрів, залишаючись в режимі автономного водіння 99% часу [2]. Проект компанії Google, Waymo також привернув увагу аудиторії до галузі безпілотного керування транспортних засобів, оскільки автомобіль на даній технології зміг проїхати близько 8 млн. км [3]. Комерційне застосування систем автономного керування було використано у компаніях Honda, Tesla, BMW, Mercedez-Benz та ін.

Нейронні мережі, зокрема згорткові нейронні мережі, знайшли широке застосування в комп'ютерному зорі і, відповідно, в системах сприйняття для систем безпілотного керування, значно покращивши точність навігації. Завдяки збору та тренуванню великого набору даних, такі моделі здатні виявляти складні патерни та залежності на зображеннях, роблячи відповідні висновки. Таке програмне забезпечення може автоматично дотреновуватися на нових наборах даних та ще краще покращувати точність моделей розпізнавання об'єктів.

Попри значні здобутки у сфері навігації автономних транспортних систем, комерційний випуск таких безпілотних автомобілей все ще залишається складною задачею для розробників та інженерів. На даний момент лише компанія Mercedes є сертифікованою компанією, яка досягла третього рівня з п'яти автономності керування, а четвертий рівень лише є у планах розробки на наступне десятиліття в більшості автомобільних компаній. Тому покращення моделей сприйняття в автомобільних системах залишається актуальною проблемою для дослідників.

Мета й завдання роботи. Метою дипломної роботи є огляд 3D-генеративних моделей та використання змагального машинного навчання для

створення складних навчальних прикладів, розроблених для підвищення загальної стійкості моделі сприйняття автономного автомобіля та проведення експериментів для доопрацюваних моделей машинного навчання за допомогою згенерованих прикладів.

Для досягнення мети поставлено такі завдання:

- Розглянути сучасні 3D генеративні моделі;
- Дослідити й описати моделі розпізнавання об’єктів;
- Спроектувати та натренувати конвеєр для генерації змагальних навчальних прикладів;
- Проаналізувати результати роботи моделей розпізнавання об’єктів на синтезованих даних та даних з реального світу.

Об’єкт, методи й засоби розроблення. Об’єктом розроблення є конвеєр генерації змагальних зображень на основі 3D генеративних моделей. Для даної роботи використовувалися аналіз та пошук необхідної літератури, дослідження сучасного стану предметної області, розробка проекту та аналіз результатів. Під час розробки використовувалися такі інструментальні засоби, як інтегровані середовища розробки PyCharm, Microsoft VSCode; інтерактивне хмарне середовище для роботи з кодом Google Colab; сервіс зберігання, редагування та синхронізації файлів Google Drive; відкрита бібліотека машинного навчання PyTorch.

Можливі сфери застосування. Дана робота пропонує потенційні вдосконалення для систем сприйняття автономних транспортних засобів, які можуть бути використані у подальшому у комерційних проектах автомобільних компаній, що займаються розробкою безпілотних автомобілів.

Розділ 1. Теоретичні основи машинного навчання та комп’ютерного зору

1.1 Машинне навчання та глибинне навчання

Штучний інтелект (ШІ) має область під назвою "машинне навчання", яка зосереджена на створенні алгоритмів і моделей, що дозволяють комп'ютерам навчатися на основі даних без необхідності явного програмування. Це передбачає створення і розробку математичних і статистичних моделей, здатних автоматично навчатися на основі даних або досвіду чи вдосконалювати їх.

У машинному навчанні комп'ютери навчаються на наборі даних, які називаються навчальними даними, щоб вивчати закономірності, взаємозв'язки та основні структури в цих даних. Мета полягає в тому, щоб дозволити комп'ютеру узагальнити своє навчання і зробити точні прогнози або рішення на основі нових, небачених даних.

У навчанні з учителем навчанні вхідні дані (ознаки) та пов'язані з ними вихідні мітки складають навчальну вибірку. Алгоритм встановлює відповідність між вхідними та вихідними даними, що дозволяє йому передбачати або класифікувати раніше недосліджені дані.

Навчання без вчителя має справу з наборами даних, в яких немає явних вихідних міток для вхідних даних. Не маючи жодних попередніх знань, алгоритми намагаються знайти закономірності, структури або зв'язки в даних. Часто використовуються такі методи неконтрольованого навчання, як кластеризація та зменшення розмірності.

Частково контрольоване навчання включає в себе аспекти навчання як контрольованого, так і неконтрольованого. Щоб підвищити точність навчання і розширити застосовність прогнозів моделі, вона використовує меншу кількість маркованих даних разом з більшою кількістю немаркованих даних.

При навчанні з підкріпленням агент розвиває здатність робити правильний вибір, взаємодіючи з навколишнім середовищем. Агент вивчає найкращі тактики або процедури для виконання певного завдання, отримуючи зворотний зв'язок або винагороду на основі своєї поведінки.

Для зміни параметрів моделі та підвищення продуктивності системи машинного навчання використовують методи математичної оптимізації, такі, як градієнтний спуск і стохастичний градієнтний спуск. Ще одним важливим етапом підготовки даних для алгоритмів машинного навчання є інженерія ознак, яка передбачає вибір або маніпулювання відповідними вхідними ознаками.

Комп'ютерний зір, обробка природної мови, розпізнавання мови, рекомендаційні системи, виявлення шахрайства, охорона здоров'я, фінанси та багато інших сфер – це лише кілька прикладів численних галузей, де використовується машинне навчання. Методи глибокого навчання та нейронні мережі розширюють межі того, чого можуть навчитися і що можуть робити машини, оскільки вони продовжують вдосконалюватися і розвиватися.

1.2 Комп’ютерний зір

Комп'ютерний зір – це галузь комп'ютерних наук і штучного інтелекту (ШІ), яка має на меті дати комп'ютерам глибоке розуміння візуальних даних, таких як зображення і фільми. Для того, щоб витягувати значущу інформацію з візуальних даних, оцінювати, інтерпретувати і приймати рішення на основі цієї інформації, необхідно розробити алгоритми і процедури.

Основна мета комп'ютерного зору – зробити так, щоб комп'ютери могли розуміти та інтерпретувати візуальну інформацію подібно до того, як це робить людина. Це включає в себе такі операції, як розуміння сцени, відстеження, оцінка пози, ідентифікація та розпізнавання об'єктів, сегментація зображень, розуміння зображень і відео, а також категоризація об'єктів.

Для більш точного аналізу використовуються методи попередньої обробки, які покращують якість зображень або відео, усувають шум і змінюють яскравість або контрастність. Наступним етапом іде видобування ознак з зображення: пошук значущих шаблонів, країв, форм, текстур або інших візуальних якостей у вхідних даних. Для вилучення ознак можна використовувати такі методи, як градієнтні методи, масштабно-інваріантне перетворення ознак (SIFT) [32] або методи, засновані на глибокому навчанні. Витягнуті ознаки перетворюються у відповідне представлення, яке алгоритми машинного навчання можуть швидко обробити. Типовими представленнями є гістограми, діаграми мішків слів та вектори ознак. Після вилучення ознак алгоритми комп'ютерного зору оцінюють візуальні дані і роблять висновки або прогнози. Це може включати категоризацію, локалізацію об'єктів, семантичну сегментацію або інші відповідні завдання.

Існує кілька реальних застосувань комп'ютерного зору в багатьох галузях і секторах. Цей список включає в себе такі галузі, як робототехніка, автомобільна індустрія, біометричні системи безпеки, медична галузь, віртуальна реальність та доповнена реальність, безпеки та охоронна діяльність.

Згорткові нейронні мережі (CNNs) та їхня здатність витягувати ієрархічні ознаки, зокрема, досягли значних успіхів у глибокому навчанні, що призвело до підвищення точності та продуктивності в багатьох додатках.

1.3 Згорткові нейронні мережі

Мережі глибокого навчання, орієнтовані на обробку та оцінку візуальних даних, таких як зображення та відео, відомі як згорткові нейронні мережі (CNN). Використовуючи згорткові шари, вони спеціально створені для автоматичного навчання ієрархічних представлень візуальних характеристик. У CNN присутні один або кілька згорткових шарів, які застосовують операції

згортки до вхідних даних. Застосування набору фільтрів, що навчаються, які часто називають ядрами, до вхідних даних під час згортки створює карти ознак, які підкреслюють певні візуальні патерни або характеристики. Оскільки CNN використовують локальні рецептивні поля, кожен нейрон у згортковому шарі лише частково пов'язаний з шаром над ним, а не повністю. Така конфігурація дозволяє мережі ефективно реєструвати локальні патерни і просторові зв'язки. CNN часто містять шари об'єднання, такі як максимальне об'єднання або середнє об'єднання, після кожного шару згортки. Просторові розміри карт об'єктів зменшуються, а важливі дані зберігаються завдяки об'єднанню. Це зменшує обчислювальну складність, одночасно допомагаючи зробити мережу більш стійкою до змін у вхідних даних. Архітектура CNN може мати один або кілька повністю пов'язаних шарів на самому кінці. На основі вивчених ознак ці шари відповідають за класифікацію або прогнозування. Для визначення ймовірностей класів вихідні дані повністю зв'язаних шарів часто подаються у функцію активації softmax. Ідея спільного використання параметрів використовується в CNN, де однаковий набір ваг (фільтрів) застосовується до вхідних даних у різних просторових точках. Це робить CNN більш ефективною в обчислювальному плані, дозволяючи мережі вивчати спільні закономірності та зменшуючи кількість параметрів, що підлягають навчанню. Як і інші нейронні мережі, CNN навчаються за допомогою алгоритму зворотного поширення. На основі розрахованих градієнтів функції втрат по відношенню до параметрів мережі коригуються ваги та зсуви мережі. За допомогою цього ітеративного методу мережа оптимізується таким чином, щоб вона могла правильно передбачати навчальні дані.

Згорткові нейронні мережі довели свою неймовірну ефективність у низці застосувань комп'ютерного зору, включаючи класифікацію зображень, виявлення об'єктів, сегментацію зображень і розпізнавання облич. Вони стали

важливим інструментом для численних реальних застосувань і значно просунули галузь комп'ютерного зору.

1.4 Моделі сприйняття

Моделі штучного інтелекту і машинного навчання, які імітують людське сприйняття і розуміють різні компоненти сенсорного введення, такі, як візуальна інформація, слухові сигнали або текстові дані, називаються моделями сприйняття. Щоб зрозуміти сенс даних і зафіксувати основні структури та закономірності, що містяться в них, ці моделі намагаються імітувати те, як люди бачать та інтерпретують навколишнє середовище.

Моделі сприйняття комп'ютерного зору роблять акцент на розшифровці та інтерпретації візуальних даних. Ці моделі можуть виконувати такі завдання, як класифікація зображень, виявлення об'єктів, сегментація зображень, розпізнавання облич та інтерпретація сцен, оскільки вони були навчені на великих масивах даних. Моделі комп'ютерного зору часто використовують згорткові нейронні мережі (CNN) та їхні відгалуження, зокрема ResNet [33], Inception [34] та VGG [35]. Моделі для розпізнавання мовлення аналізують аудіоінформацію та перетворюють усне мовлення на текст. Вони використовують такі методи, як приховані марковські моделі, рекурентні нейронні мережі (RNN) та згорткові нейронні мережі (CNN), які навчаються на аудіоданих, щоб ідентифікувати фонетичні особливості та лінгвістичні патерни в мовних сигналах. Для того, щоб зрозуміти та інтерпретувати природну мову, моделі сприйняття NLP працюють з текстовим введенням. Аналіз настрою, розпізнавання іменованих об'єктів, категоризація тексту, машинний переклад, відповіді на запитання та синтез мови – це лише деякі з багатьох завдань, які вони виконують. NLP досягло значного прогресу завдяки таким моделям, як рекурентні нейронні мережі (RNN), довга короткочасна пам'ять (LSTM), трансформери [36] та BERT [37]. Щоб глибше зрозуміти дані, мультимодальні моделі сприйняття поєднують інформацію з

різних сенсорних модальностей, включаючи візуальну, аудіальну та мовну. Такі моделі можуть обробляти та інтерпретувати кілька джерел вхідної інформації одночасно, що дозволяє виконувати такі функції, як аудіовізуальне розпізнавання мови, візуальні відповіді на запитання та субтитрування відео. Моделі сприйняття також мають застосування в генеративних моделей. Основна мета генеративних моделей сприйняття полягає в тому, щоб навчатися на навчальних даних і створювати нові зразки даних, які точно відображають їхні закономірності та властивості. Прикладами генеративних моделей є генеративні змагальні мережі (GANs) [8] та варіаційні автокодери (VAE) [38], які можуть створювати точні зображення, аудіо та текст.

У багатьох сферах, таких, як автономне водіння, охорона здоров'я, робототехніка, рекомендаційні системи та віртуальні асистенти, моделі сприйняття мають важливе значення. Ці моделі можуть допомогти роботам спостерігати і розуміти навколишнє середовище, дозволяючи їм діяти розумно, приймаючи рішення, взаємодіючи з людьми і виконуючи складні завдання.

Розділ 2. Використані технології

2.1 Змагальне навчання та атаки зловмисників в автономному керуванні

Метод змагального машинного навчання передбачає навчання моделі шляхом зіставлення її з опонентом. Завдяки спеціально модифікованим вхідним даним модель може підвищити точність і надійність класифікації. Цей метод широко використовується в галузі комп'ютерного зору, зокрема для класифікації зображень, оскільки він дозволяє показати обмеження класифікації моделі та покращити її узагальнюваність.

Противник у цьому методі намагається згенерувати невидимі для людини модифікації вхідних зображень, які можуть призвести до того, що модель машинного навчання неправильно класифікує зображення. Найпопулярніші методи генерації прикладів включають градієнтні методи, алгоритми оптимізації, генетичні алгоритми та методи математичного перетворення вхідних зразків (додавання шуму, розмиття, зміни розміру або деформації вхідних даних).

Останнім часом для генерації змагальних прикладів використовують моделі GAN [8]. Ця нейронна мережа спрямована на розробку моделі генератора, яка може вловити основний розподіл даних і створити реалістичні зразки, такі як фотографії. У той час як дискримінатор намагається відрізнити реальні зразки з навчальних даних від фальшивих зразків, створених генератором, генератор використовує випадковий шум як вхідні дані для створення синтетичних зразків. Генератор намагається обдурити дискримінатор, а дискримінатор намагається правильно передбачити результати, оскільки обидві мережі навчаються одночасно.

Фундаментальним моментом в GAN є те, що обидві мережі стають кращими в результаті конкуренції між генератором і дискримінатором. Дискримінатору стає все важче відрідняти справжні зразки від фальшивих, оскільки генератор створює все більш реалістичні приклади. Ця ітеративна процедура триває доти, доки генератор не зможе створювати зразки, які дискримінатор не зможе відрізнити від реальних даних. GANs стали поштовхом до значних досліджень у галузі змагального навчання та генеративного моделювання. Прикладне застосування GANs отримали в синтез зображень, перенесення стилю, перетворення зображення в текст, перетворення тексту в зображення та ін.

AdvGAN [9] є специфічним варіантом генеративних змагальних мереж (GAN), який призначений для генерації змагальних прикладів. У звичайному GAN мережа дискримінатора має на меті розрізняти реальні та створені зразки, тоді як генераторна мережа вчиться зіставляти випадковий шум з реалістичними зразками. На противагу цьому, генераторна мережа змагальної GAN прагне створити зразки, які можуть обдурити як кодувальну мережу, так і мережу дискримінатора. Найголовнішими перевагами AdvGAN є більша обчислювальна ефективність та можливість виконувати атаки “напівбілого ящика” і “білого ящика”, на відміну від інших оптимізаційних методів.

Розглянемо схему архітектури AdvGAN на рис. 2.1. Основними елементами змагальної GAN є генератор, кодувальник та дискримінатор. Генератор створює синтетичні зразки з випадкового шуму на вході, кодувальник перетворює вхідні дані (створені або реальні) у латентний простір меншої розмірності, а дискримінатор намагається розрізнити справжні та штучні зразки. У тренуванні AdvGAN є два основні етапи: змагальне тренування та тренування з реконструкції.

На етапі змагального тренування одночасно тренуються мережі кодера і генератора. Створюючи зразки і кодування, які важко відрізнити від справжніх даних, генератор і кодувальник працюють разом, щоб обдурити дискримінатор.

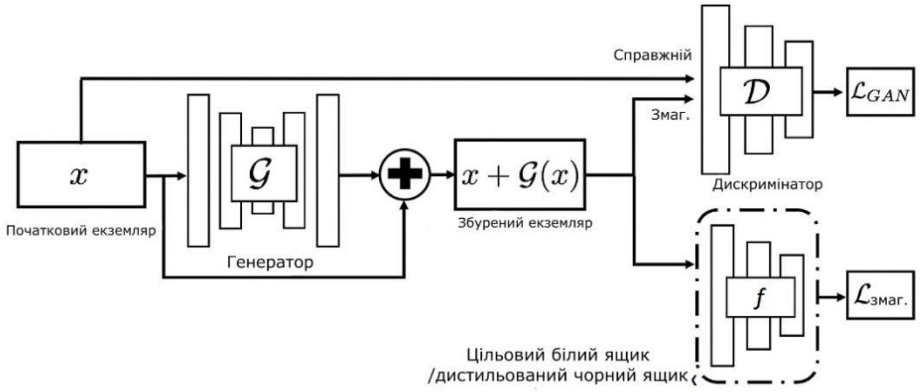

Рисунок 2.1 – Архітектура мережі AdvGAN

Після змагального тренування кодер фіксується, а генератор видаляється під час реконструктивного тренування. Декодувальник навчається декодувати закодовані зразки і відновлювати початковий вхідний сигнал. Цей процес підштовхує декодувальника до збору релевантного матеріалу з латентного простору і генерування точного представлення даних. Метою AdvGAN є розробка генеративної моделі, яка може створювати точні вибірки і точно відображати розподіл даних. Кодувальник також прагне зіставити згенеровані та реальні зразки в латентний простір, де дискримінатор не зможе їх розрізнити.

Атаки зловмисників мають на меті ввести в оману модель машинного навчання. Через недостатню інтерпретованість моделей

машинного навчання і занадто сильну чутливість до незначних змін вхідних даних, зловмисники можуть використати вразливості моделі машинного навчання для того, щоб привести її до неправильно класифікації.

Зловмисні атаки можна розділити на дві основні категорії: атака “білого ящика” і атака “чорного ящика”. В атаках "білого ящика" зловмисник має повне знання про цільову модель машинного навчання, включаючи її архітектуру, параметри та навчальні дані. Ця інформація дозволяє зловмиснику генерувати приклади, безпосередньо обчислюючи градієнти моделі та оптимізуючи збурення, щоб максимізувати вразливість моделі. В атаці “чорного ящика” зловмисник має обмежені знання про модель або взагалі не має інформації про неї. Попри те, що він може не мати доступу до внутрішніх деталей моделі, він може робити запити до моделі за допомогою певних вхідних даних і спостерігати за вихідними даними. Цю інформацію зловмисник може використовувати для генерації прикладів, застосовуючи різні методи оптимізації, наприклад, перенесення [10], коли приклади, згенеровані для однієї моделі, використовуються для атаки на іншу модель зі схожою архітектурою.

Такі атаки можуть мати серйозні наслідки в критично важливих системах важливих для безпеки програмних забезпеченнях для автономного керування автомобілем або медичної діагностики. На даний момент дослідники намагаються розробити більш надійні і стійкі моделі, попри те, що нові механізми атак продовжують з’являтися.

Два основні ресурси для виявлення об’єктів для систем автономного керування є LiDAR сенсори та сенсори з камери. Використовуючи вразливості цих сенсорів, зловмисники можуть вводити в оману процес розпізнавання об’єктів. Моделі розпізнавання об'єктів на основі LiDAR для AV-систем були

досліджені в [4], де запропонований алгоритм генерує змагальні 3D хмари точок, які можуть вводити підроблені перешкоди на близьких відстанях. LiDAR-Adv, представлений в [5], є підходом для генерування змагальних об'єктів у реальному світі, які не можуть бути виявлені системами детекції об’єктів на основі LiDAR. Інший метод атаки для генерації прикладів для хмар точок LiDAR було представлено в [6], який базується на вразливості архітектур сприйняття на основі LiDAR, може призвести до виявлення фальшивого транспортного засобу, що рухається переднім ходом. Као та ін. [7] побудували нову змагальну атаку на атаку шляхом модифікації різних форм 3D-об'єкта, що може призвести як до зміни положення точок у хмарі точок LiDAR, так і до зміни пікселів. Такі атаки можуть становити серйозну загрозу для життя пасажирів та водія транспортного засобу, оскільки на даний момент не було опубліковано методів протидій таких атак.

2.2 3D генеративні моделі

Останніми роками багато дослідників вивчають генеративні 3D-моделі для створення контенту у вигляді 3D візуалізації. Попередні роботи можна класифікувати відповідно до представлення вихідних даних для 3D генеративних моделей. Ранні підходи були спрямовані на безпосередню генерацію 3D воксельних сіток [11], що обмежує їх можливості у генеруванні складних 3D-форм через високий обсяг пам'яті та обчислювальній складності 3D-згорток у високій роздільній здатності.

Інші роботи досліджували хмару точок [12, 13] або неявну функцію [14, 15], однак, більшість з них зосереджені лише на геометрії та нехтують

зовнішній вигляд 3D-форм, обмежуючи їх застосування у змагальному навчанні для моделей сприйняття, які потенційно включають рендеринг зображень. Зовсім недавно, з успіхом нейронного об'ємного рендерингу [16] та неявних представлень [14, 15], деякі роботи також досліджували проблему синтезу зображень з урахуванням 3D синтезу зображень з урахуванням 3D-інформації [17, 18], однак, для отримання точної 3D-геометрії з цих робіт потрібно алгоритм недиференційованих маршових кубів [19], і, таким чином, перешкоджає застосуванню змагального навчання для сприйняття зображень на основі LiDAR.

Модель GET3D [39] (див. рис. 2.2) долає недоліки сучасних методів створення 3D-об'єктів з фотографій. Основними досягненнями цього алгоритму є подолання проблем попередніх моделей з досягненням реалістичних варіацій форми, текстур високої роздільної здатності та дрібних деталей. Для створення тривимірних форм GET3D використовує комбінацію 2D-графіки та тривимірних воксельних сіток. Для того, щоб відобразити геометричну структуру фігури, генератор геометрії створює її 3D воксельне представлення, а генератор текстур створює текстурну інформацію для покращення візуальної точності фігури. В процесі наскрізного тренування використовуються змагальні втрати визначені на наборі даних 2D-зображень. У моделі GET3D також використовуються два 2D дискримінатори для встановлення правдивості зображення.

Дана робота спирається на модель GET3D [39], яка дає змогу безпосередньо виводити текстуровані 3D-сітки. GET3D [39] – це генеративна 3D модель на основі GAN [8]. Зокрема, модель спочатку вибирає два латентні вектори з попередніх розподілів, один латентний вектор представляє геометрію, а інший – текстуру. Потім модель прогнозує тривимірне

представлення для 3D-форм, з якого модель декодує SDF. Після цього GET3D використовує DMTet [40] для вилучення сіток з нульовим перетином з SDF. GET3D представляє текстуру як текстурне поле, визначене на поверхні 3D-сіті. За допомогою передбачення з GET3D можна легко виокремлювати з нього хмари точок вибірки або рендерити зображення. Зокрема, вибірка хмар точок передбачає білінійну інтерполяцію вершин з сітки, а білінійна інтерполяція є диференційованою. Візуалізація зображень може бути досягнута за допомогою диференційованого рендерингу [22]. Поєднання диференційованого ізоповерхневого рендерингу [21], диференційованого рендерингу [22], а також генеративного моделювання [23] дозволяє застосовувати змагальне навчання як для моделей сприйняття на основі хмари точок, так і для моделей на основі зображень.

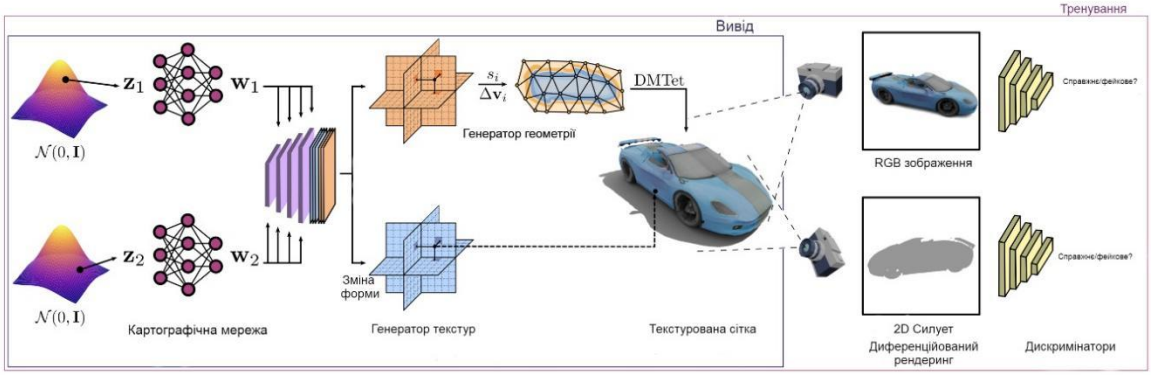

Рисунок 2.2 – Архітектура мережі GET3D

2.3 Розпізнавання 2D і 3D об'єктів для автономного водіння

У системах автономного водіння розпізнавання 2D і 3D об'єктів відіграє вирішальну роль у сприйнятті навколишнього середовища та ідентифікації цільових об'єктів. Виявлення об'єктів можна розділити на двовимірне (2D) та тривимірне (3D) виявлення об'єктів.

Процес розпізнавання 2D-об'єктів передбачає ідентифікацію та категоризацію об'єктів на парі фотографій або колекції зображень, які зазвичай були зроблені камерами, встановленими на автономному транспортному засобі. При виявленні 2D-об'єктів на вхід подаються зображення з камери, і для різних об'єктів у сцені генеруються 2D-обмежувальні коробки [24]. Для того, щоб виділити ознаки на фотографіях і класифікувати об'єкти за певними категоріями (наприклад, пішоходи, автомобілі та дорожні знаки), ця процедура часто використовує методи комп'ютерного зору, такі, як згорткові нейронні мережі (CNNs). Останнім часом для точної та ефективної локалізації об'єктів у 2D просторі використовуються такі моделі, як Faster R-CNN [26], YOLO [28] та SSD [29].

3D розпізнавання також прогнозує тривимірне положення, розмір та орієнтацію об'єктів у сцені. Розуміння просторових взаємозв'язків і динаміки в навколишньому середовищі вимагає врахування цієї інформації. Для цього використовується декілька сенсорних методів, включаючи LiDAR та глибинні камери. Датчики LiDAR генерують хмару точок, що представляють сцену, випромінюючи лазерні промені і реєструючи їхні відбиття для оцінки відстаней. Ці дані хмари точок використовуються алгоритмами розпізнавання 3D-об'єктів для сегментації об'єктів і обчислення їхніх 3D-меж, що дозволяє точно локалізувати і відстежувати об'єкти в 3D-просторі. Згадані вище типи даних подаються як вхідні дані до моделі машинного навчання. На виході з моделей виводяться детектори 3D-об'єктів разом з класом об'єктів та відсотком впевненості приналежності до певного класу. У межах 3D детекції об'єктів застосовуються різні методи залежно від того, які модальності датчиків використовуються, а також від того, як вони об'єднуються разом, якщо їх введено декілька.

Для побудови надійних моделей сприйняття систем автономного водіння часто використовують комбінацію 2D- і 3D-методів розпізнавання об'єктів. Хоча 2D-розпізнавання об'єктів надає детальну візуальну інформацію і є корисним у деяких ситуаціях, воно може мати проблеми з оклюзіями, змінами в освітленні та плутаниною об'єктів. Однак розпізнавання 3D-об'єктів на основі даних LiDAR має проблеми з прозорими або світловідбиваючими поверхнями, хоча й перевершує 2D-ропізнавання у точному визначенні положення та форми об'єкта. Автономні транспортні засоби можуть краще орієнтуватися в навколишньому середовищі і виконувати розпізнавання та відстеження об'єктів, поєднуючи дані двох сенсорів. Поєднання технології розпізнавання 2D і 3D об'єктів покращує сприйняття систем автономного водіння, дозволяючи їм розпізнавати і контролювати об'єкти, прогнозувати їхню поведінку і приймати обґрунтовані рішення для забезпечення безпечної та ефективної дорожньої навігації.

Оскільки обрана генеративна модель для даної роботи дозволяє витягувати як 2D-зображення з камер, так і хмари точок, вибір 2D чи 3D розпізнавання не є обмеженим. Для простоти реалізації та налаштування було використовувано YOLOv5 [25] та Faster R-CNN [26] 2D модель виявлення об'єктів з бібліотеки detectron2 [27].

2.4Модель YOLO

Модель YOLO [28] вперше була випущена в 2015 році, і з тих пір такі версії, як YOLOv2 [41], YOLOv3 [42] і YOLOv4 [43], розширили її ідеї та покращили можливості виявлення об'єктів. Основною ідеєю дослідження YOLO є уніфікована ідентифікація, яка забезпечує єдину архітектуру нейронної мережі, що здійснює категоризацію об'єктів і регресію

обмежувального поля за один прохід. YOLO зробила революцію в ідентифікації об'єктів, оскільки автори запропонували більш швидкий та простий підхід з пропозицією розпізнавального регіону у порівнянні з попередніми техніками.

YOLO розбиває вхідне зображення на сітку, яка потім прогнозує обмежувальні коробки та ймовірності класів для кожної комірки сітки. Незалежно від того, скільки об'єктів присутні на зображенні, кожна комірка сітки повинна прогнозувати певну кількість обмежувальних коробок. Використовуючи метод на основі сітки, виявлення об'єктів можна зробити швидко і ефективно. Кожне передбачення обмежувальної коробки складається з чотирьох координат, що відображають положення обмежувальної коробки. Крім того, для виявлених об'єктів кожній коробці присвоюється ймовірність віднесення до того чи іншого класу та клас об’єкта.

У функції втрат YOLO об'єднані помилки класифікації та локалізації. Функція втрат заохочує точні та з високою впевненістю прогнози, а неточні – штрафує, залежно від точності та об'єктивності локалізації. YOLO досягає виявлення об'єктів у реальному часі, максимізуючи проектні рішення мережі та використовуючи ефективність однопрохідної архітектури. Дана модель може виявляти об'єкти швидко та ефективно, але має проблеми з точною локалізацією малих об'єктів або об'єктів з великим співвідношенням сторін. Іншим недоліком методу на основі сітки може є ускладнення пошуку об'єктів, розташованих близько один до одного.

З часу першої публікації статті про YOLO [28] в таких оновленнях, як YOLOv2 [41], YOLOv3 [42] і YOLOv4 [43], з'явилися такі вдосконалення, як вища точність, швидша швидкість обробки та інші функції, як-от якірні прямокутники, пірамідальні мережі ознак і складні мережеві структури. Ці

розробки підвищили ефективність виявлення об'єктів у реальному часі, водночас усунувши деякі недоліки.

2.5 Модель Faster R-CNN

Faster R-CNN [26] – це регіональна згорткова нейронна мережа та популярний метод глибокого навчання для виявлення об'єктів. Faster R-CNN представляє RPN, окрему мережу, яка створює регіональні пропозиції або регіони-кандидати на створення об'єктів. Використовуючи вхідне зображення, RPN може генерувати список пропозицій об'єктів та їхні координати. Щоб розрізняти регіони переднього плану (об'єкти) і фону та покращувати пропозиції, RPN використовується для тренування. Також дана модель для виявлення об'єктів використовує два етапи. На першому етапі на основі наданого зображення RPN генерує пропозиції регіонів. На другому етапі ці пропозиції обробляються за допомогою шару об'єднання областей інтересу, а потім надсилаються через класифікатор для встановлення мітки класу і точного налаштування координат обмежувальної коробки. Faster R-CNN використовує об'єднання областей інтересу для отримання карт об'єктів фіксованого розміру для кожної запропонованої області. Для того, щоб створити карти ознак фіксованого розміру, цей процес ділить кожну запропоновану область на сітку підобластей і виконує максимальне об'єднання всередині кожної підобласті. На другому етапі ці карти особливостей використовуються для класифікації та регресії за методом обмежувальних рамок. За основу Faster R-CNN [26] часто використовують як основу згорткові нейронні мережі, наприклад, VGG16 [35] або ResNet [33]. RPN, етап класифікації об'єктів та етап регресії з використанням обмежувальних рамок –

всі вони використовують високорівневі представлення ознак, які витягуються з вхідного зображення базовою мережею. Для навчання Faster R-CNN використовується багатоступенева методика. Спочатку RPN навчається створювати відмінні пропозиції. Потім мережа навчається наскрізно за допомогою зворотного поширення, включаючи RPN і наступні етапи виявлення об'єктів. Етапи процесу навчання включає оптимізація класифікаційних оцінок, координат обмежувальних рамок і створення пропозицій RPN.

Широкий спектр завдань виявлення об'єктів, включаючи ідентифікацію пішоходів, розпізнавання облич і загальне розпізнавання об'єктів, ефективно вирішується за допомогою швидшого R-CNN. Завдяки своїй точності і здатності обробляти складні сценарії, він став кращим варіантом для багатьох застосувань комп'ютерного зору.

2.6 Модель Mask R-CNN

Прогресивним підходом глибокого навчання для задач виявлення об'єктів та сегментації екземплярів є Mask R-CNN [20]. Фреймворк Faster R-CNN [26], який складається з двох етапів класифікації регіонів та генерації пропозицій регіонів, слугує основою для виявлення об'єктів Mask R-CNN [20]. На першому етапі за допомогою мережі пропозицій регіонів створюються регіони-кандидати (пропозиції) об'єктів. На другому етапі ці пропозиції об'єднуються в класи об'єктів і вдосконалюються для більш точного прогнозування граничних областей. Mask R-CNN [20] розширює фреймворк Faster R-CNN [26] для сегментації екземплярів на додаток до виявлення об'єктів. Для кожної пропозиції об'єкта створюється паралельна гілка, яка прогнозує маски сегментації на рівні пікселів. Це дає змогу точно окреслити

межі об'єктів і розрізнити екземпляри, що перетинаються. Гілка маски Mask R-CNN [20] використовує повністю згорткову мережу для прогнозування бінарної маски для кожної пропозиції за допомогою карти ознак фіксованого розміру. За допомогою просторово вирівняного об'єднання RoI і послідовності згорткових шарів для побудови попіксельної маски сегментації здійснюється передбачення маски. Mask R-CNN [20] часто використовує мережу глибокої згортки як основу, наприклад, ResNet [33] або ResNeXt [44]. Ця опорна мережа витягує з вхідного зображення багаті та складні ознаки і використовує їх для розробки пропозицій областей, класифікації та прогнозування маски. Поєднуючи втрату класифікації, втрату регресії граничного поля та втрату сегментації маски, Mask R-CNN навчається від початку до кінця. Розбіжність між прогнозованою та істинною масками штрафується втратою маски, яка обчислюється за допомогою базових масок під час навчання. Таке одночасне покращення підвищує як точність ідентифікації об'єктів, так і якість сегментації екземплярів. Серед застосувань Mask R-CNN є виявлення об'єктів, сегментація екземплярів і виявлення ключових точок. Mask R-CNN часто використовується для цих завдань у комп'ютерному зорі прикладному до таких галузей, як робототехніка, автономне водіння, медична візуалізація та аналіз відео. Модель Mask R-CNN зарекомендувала себе як провідний фреймворк для вирішення завдань розпізнавання об'єктів і сегментації екземплярів, які вимагають точності на рівні пікселів і глибокого розуміння об'єктів, демонструючи передову продуктивність на еталонних наборах даних, таких, як COCO [45].

Розділ 3. Програмна реалізація системи

3.1 Загальний план реалізації системи

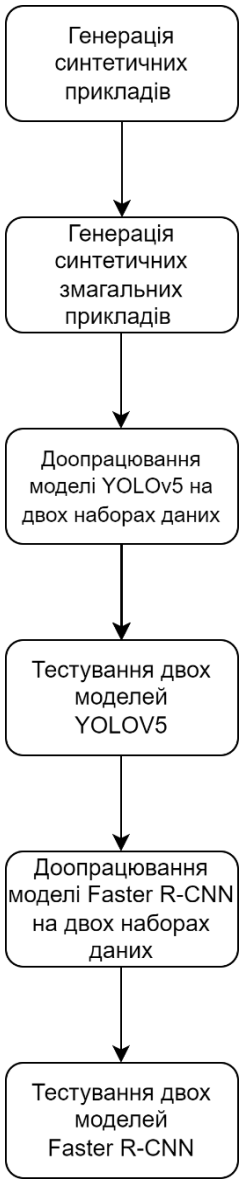

Рисунок 3.1 – Схема реалізації системи

На рис. 3.1 зображена схема реалізації системи, який складається з шести етапів. На першому етапі відбувається генерація синтетичних зображень 3D генеративної моделі GET3D класу “машина” за допомогою інтерфейсу генерації зображень та попередньо натреновані й моделі для класу “машина”.

Другий етап складається з генерації синтетичних змагальних прикладів, що були згенеровані за допомогою алгоритму генерації 2D змагальних прикладів. Два згенеровані набори даних будуть використані для доопрацювання моделі YOLOv5. Наступним етапом йде тестування двох видів моделі YOLOv5 на синтетичному наборі даних. На п’ятому етапі модель Faster R-CNN доопрацьовується на двох попередньо згенерованих наборах даних. На фінальному етапі дві типи моделей Faster R-CNN будуть протестовані на наборі даних з реального світу.

3.2 Генерація змагальних прикладів

Розглянемо алгоритм генерації 2D змагальних прикладів на рис. 3.2. Спочатку обираються два латентні вектори з нормального розподілу, один латентний вектор представляє геометрію, а інший — текстуру. Потім ініціаліазується оптимізатор з латентними векторами. Наступним кроком генеруються зображення та хмари точок з тривимірної генеративної моделі на основі GAN GET3D [39]. Далі слідують ітеративно повторювані кроки до збіжності. Модель генерує 3D-форми з латентних векторів та за допомогою диференційованого рендерингу відтворює 2D зображення. Отримані 2D зображення подаються на вхід для попередньо навченої мережі виявлення та класифікації 2D-об'єктів. На виході модель розпізнавання об’єктів буде видавати обмежувальні коробки, в яких знаходиться об’єкт. Вони будуть використані для порівняння того, наскільки добре змагальний приклад зміг "обдурити" модель сприйняття. Змагальна функція втрат спонукає моделі сприйняття передбачати неправильні обмежувальні коробки, в яких знаходиться об’єкт, обчислюючи різницю між справжніми обмежувальними коробками та змагальними. Ця функція втрат може бути зворотньо поширена на початкові латентні вектори. Оскільки спосіб, у який виділяється хмара точок для рендерингу зображень зображень, є диференційованим, як і весь 3D-

генератор і модель сприйняття, весь алгоритм можна навчити наскрізно генерувати змагальні приклади. Як тільки ці приклади будуть створені, їх можна буде використати для доопрацювання моделей сприйняття.

```
latent code = randn() // випадково вибрати латентний код з нормального розподілу
optimizer = Adam(latent code, lr) // оптимізатор для латентного коду
while not converged do
    three_d shape = GET3D(latent code) // згенерувати 3D форму з латентного коду
    two_d image = Render(three_d shape) // диференційовано прорендерити 3D-форму у 2D зображення
    Predict bbox = MaskRCNN(two_d image) // передбачити 2D-обмежувальну коробку з 2D зображення
    loss = ||predicted bbox - wrong bbox|| // обчислити змагальну втрату
    loss.backward() // зворотнє поширення
    optimizer.step() // градієнтний спуск на латентному коді
```

Рисунок 3.2 – Алгоритм генерації 2D змагальних прикладів

У ролі моделі розпізнавання об’єктів використовується попередньо навчену модель Mask R-CNN від Detectron [27] як модуль 2D сприйняття. 512-вимірні латентні коди для геометричного і текстурного генератора були згенеровані випадковим чином в GET3D [39], а офіційний випуск чекпоінта GET3D [39] використовується як 3D генеративна модель. Згенерована 3D-форма була візуалізована у фіксованій точці огляду камери з роздільною здатністю зображення 1024.

Для обчислення змагальної втрати була використана неправильна граничну область як [512, 512, 532, 532], і обчислена функцію втрат MSE між прогнозованою та неправильною обмежувальною коробкою об’єкта. Оптимізація відбувається з темпом навчання 1e-2 та оптимізатором Адама [46].

Приклади згенерованих зображень показані на рис. 3.3 та рис. 3.4. У першому рядку рис. 3.3 показано передбачення рамок для виявлених об’єктів для незмагальних згенерованих 3D-форм, у другому рядку та на рис. 3.4 – згенеровані змагальні приклади з використанням вищеописаного алгоритму.

Рисунок 3.3 – Приклади зображень, згенерованих з GET3D

Рисунок 3.4 – Приклади зображень, згенерованих з GET3D

Як показано на рис. 3.3, модель Mask R-CNN має тенденцію передбачувати дві обмежувальні коробки для визначення об’єкту в змагальних прикладах, замість однієї. Це може бути проблематичним для визначення об’єкта, оскільки це призводять до плутанини і неточної інтерпретації. Використання алгоритму може допомогти згенерувати змагальні приклади

для моделі виявлення об'єктів, що дозволить доопрацювати модель 2D сприйняття за допомогою 2D зображень, згенерованих 3D-генеративною моделлю.

Використовуючи запропонований конвеєр генерації прикладів, було сформовано набір даних з 10 000 зображень автомобілів, який було використано для доопрацювання нашої моделі виявлення об'єктів, починаючи з її попередньо навчених ваг. 8000 з цих зображень було використано в навчанні, а 2000 — як тестовий набір для оцінки моделі.

3.3 Порівняння неналаштованої та доопрацьованої моделі YOLOv5

Два типи моделей YOLOv5 [25]: неналаштована та доопрацьована були протестовані на двох наборах даних: синтезовані звичайні (незмагальні) приклади, який генеруються моделлю Get3D і синтезовані змагальні приклади, які були згенеровані запропонованим конвеєром генерації змагальних даних. Готовий детектор об'єктів YOLOv5 був використаний як базова модель, а доопрацьована модель YOLOv5 на змагальних прикладах була використана як індивідуальна модель. Валідаційний набір даних з виявленням лише класу “машина” та “вантажівка” був використаний для порівняння моделі.

У таблиці 3.5 зображено середні довірчі оцінки до класу між двома синтетичними (незмагальними та змагальними) наборами даних. Як видно з таблиці, спостерігається покращення достовірності класу виявлення об'єктів на обох наборах даних з доопрацьованою моделлю YOLOv5 порівняно з неналаштованою моделлю. Неналаштована модель YOLOv5 показала гірші результати довіри до об’єкту класу “машина” або “вантажівка” проте змогла виявити об'єкт на всіх зображеннях у наборі даних, на відміну від доопрацьованої моделі YOLO v5.

	Незмагальні приклади	Змагальні приклади
Неналаштована модель YOLOv5	54.34	51.22
Доопрацьована модель YOLOv5	85.97	86.54

Таблиця 3.5 – Середні довірчі оцінки до класу між двома синтетичними (незмагальними та змагальними) наборами даних.

3.4 Доопрацювання моделі розпізнавання Faster R-CNN та порівняння двох моделей

Використовуючи згенеровані змагальні приклади, отримані вище, була доналаштувана модель Faster R-CNN з темпом навчання 0.0125, максимальною кількістю ітерацій 2500, розміром порції 4 та 1024 як кількість пропозицій, що береться з RPN при обчисленні втрат.

На рис. 3.6 зображений графік точності класифікації моделі Faster R-CNN при доопрацюванні моделі набором даних із згенерованих незмагальних зображень в залежності від кількості ітерацій. Як видно з рисунка, на ітерації, що дорівнює 2000, точність класифікації практично не зростає.

На рис. 3.7 зображений графік функції втрат при доопрацюванні моделі Faster R-CNN набором даних із згенерованих зображень в залежності від кількості ітерацій. Як видно з рисунка, значне спадання значення функції втрат спостерігається при перших 500-ї операції. Починаючи з 1500-ї операції, значення функції втрат залишається практично незмінним.

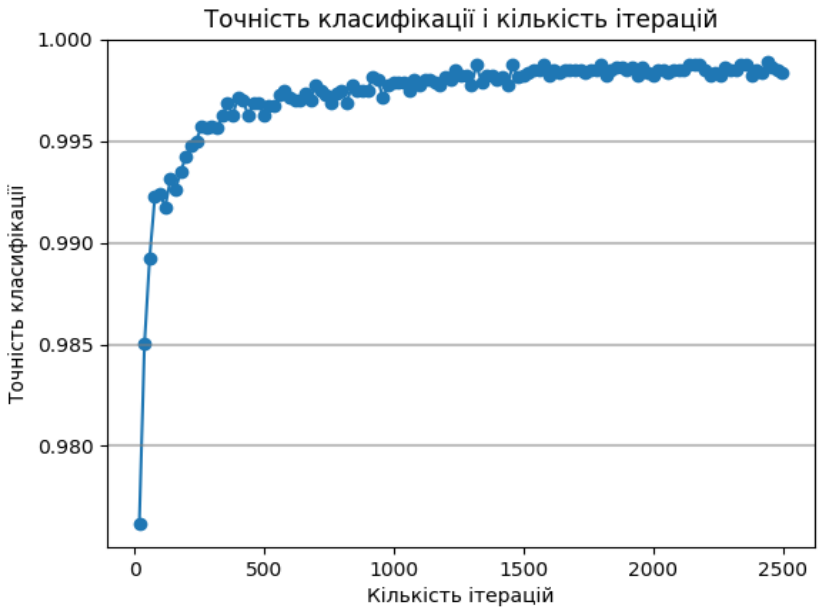

Рисунок 3.6 – Графік точності класифікації моделі Faster R-CNN (незмагальний набір даних)

На рис. 3.8 та рис. 3.9 зображено графіки точності класифікації моделі Faster R-CNN та графік функції втрат, відповідно, при доопрацюванні моделі Faster R-CNN набором даних із згенерованих змагальних зображень. Як видно з графіків, точність моделі, натренованої на змагальних прикладах, є дещо нижчою у порівнянні з точністю моделі, натренованої на згенерованих звичайних прикладах на такій самій кількості ітерацій. Значення функції втрат моделі, натренованої на змагальних прикладах, є вищою у порівнянні з точністю моделі, натренованої на згенерованих незмагальних прикладах.

Рисунок 3.7 – Графік функції втрат при доопрацюванні моделі Faster R-CNN (незмагальний набір даних)

На рис. 3.10, 3.11, 3.12 та 3.13 представлено приклади розпізнавання автомобілів моделі Faster R-CNN, що була доопрацьована на штучно створеному незмагальному та штучно створеному змагальному наборі даних. Кінцеві результати були отримані шляхом виконання тестування моделі на невеликій вибірці з набору даних nuScenes [47]. Цей набір даних містить показання датчиків під час руху транспортних засобів на різних дорогах, в різних умовах та в різних середовищах. Зокрема, для тестування було використано RGB зображення з передньої камери автомобіля.

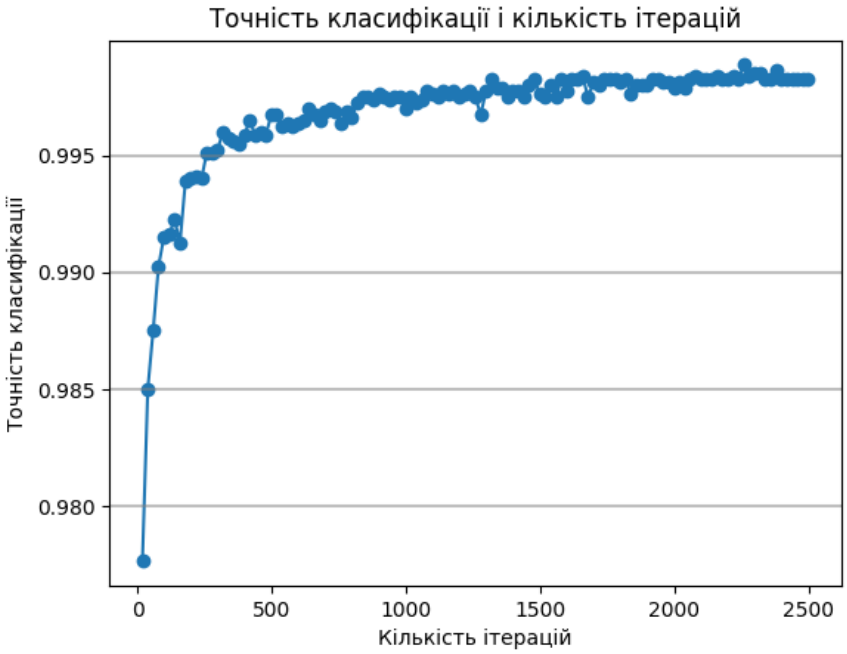

Рисунок 3.8 – Графік точності класифікації моделі Faster R-CNN (змагальний набір даних)

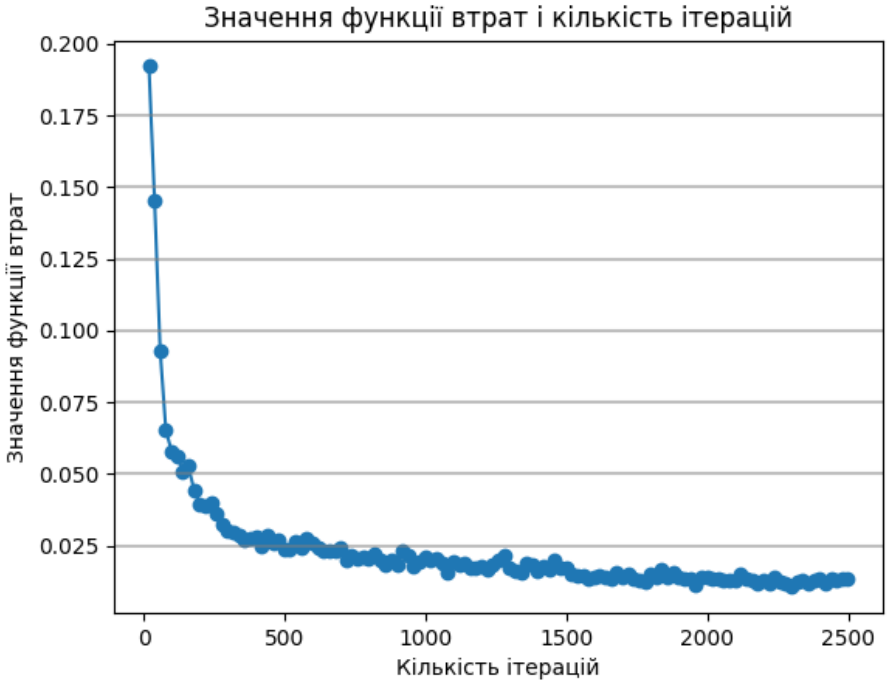

Рисунок 3.9 – Графік функції втрат при доопрацюванні моделі Faster R-CNN (змагальний набір даних)

Як видно з зображень, обидві моделі показують достатньо точні результати розпізнавання автомобіля на відео. Однак модель, доопрацьована на штучно створеному змагальному наборі даних класифікує додаткові неіснуючі автомобілі (рис. 3.10 та рис. 3.12) або класифікує автомобіль з меншою точністю (рис. 3.11 та рис. 3.13).

Рисунок 3.10 – Приклади розпізнавання автомобілів за допомогою моделі Faster R-CNN, доопрацьованої на двох наборах даних

Рисунок 3.11 – Приклади розпізнавання автомобілів за допомогою моделі Faster R-CNN, доопрацьованої на двох наборах даних

Рисунок 3.12 – Приклади розпізнавання автомобілів за допомогою моделі Faster R-CNN, доопрацьованої на двох наборах даних

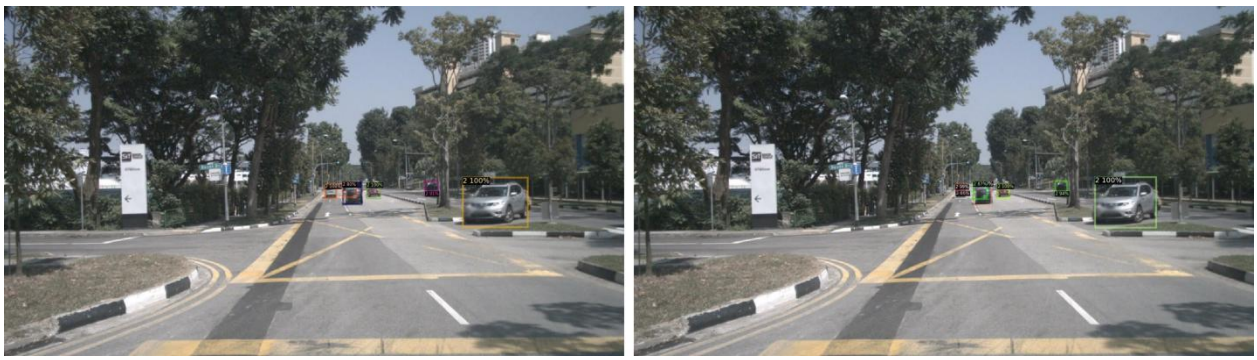

Рисунок 3.13 – Приклади розпізнавання автомобілів за допомогою моделі Faster R-CNN, доопрацьованої на двох наборах даних

У таблиці 3.14 зображено метрики середньої точності та середньої повноти для моделі Faster R-CNN, доопрацьованої на двох наборах даних. Тестувальний набір даних з 90 зображень з розмітками класу “машина” був взяти з тренувального набору даних BDD100K [48], що містив 100 000 зображень. Як видно з таблиці, попри незначне підвищення середньої точності моделі доопрацьованої моделі на змагальному наборі даних з 57 до 59 відсотків, спостерігається значний спад в середній повноті з 68 до 61 відсотків. Це свідчить про те, що модель, дотренована на змагальному наборі даних, не враховує багато існуючих об’єктів для класифікації, що призводить до того, що більша кількість об’єктів буде помилково класифікована як неіснуючі. Дані

проблеми класифікації можуть завдати багато аварій для автономних транспортних засобів.

	Звичайний набір даних	Змагальний набір даних
Середня точність (Mean precision)	57.00	59.51
Середня повнота (Mean recall)	68.97	61.17

Таблиця 3.14 – Метрики для розпізнавання автомобілів за допомогою моделі Faster R-CNN, доопрацьованої на двох наборах даних

ВИСНОВКИ

У цій роботі було представлено конвеєр генерації даних, який використовує продуктивність моделі сприйняття як зворотний зв'язок для безперервної генерації дедалі складніших прикладів. За допомогою цього конвеєра було згенеровано синтетичний змагальний набір даних. Цей набір даних був використаний для доопрацювання детектора об'єктів YOLOv5, який потім був порівняний неналаштованою моделлю YOLOv5. Була проведена оцінка довіри до класу між двома синтетичними (незмагальними та змагальними) наборами даних. Також було показано кількісні (метрики точності розпізнавання автомобілів) та якісні (розпізнавання автомобілів на зображеннях) результати до доопрацьованої моделі Faster R-CNN на синтетичному незмагальному та синтетичному змагальному наборах даних та порівняно результати розпізнавання двох моделей.

Подальшими покращеннями даної роботи може бути подолання розриву між синтетичними зображеннями та зображеннями з реального світу: додавання фону до кожного з зображень автомобілів для більш реалістичного порівняння; включення ширшого спектру класів та створення змагальних прикладів, які міститимуть кілька класів з більш реалістичним тлом.

На даний момент набуває розвитку галузь, яка називається органічним моделюванням на основі даних, що використовує показники датчиків з автономних транспортних засобів для створення реалістичних симуляцій в автономному режимі. Прикладом може слугувати [30], де автори доповнюють реальні сцени 3D-активами для створення складних, але реалістичних прикладів, або [31], де використовується симулятор з відкритим вихідним кодом, що здатний синтезувати високоточні сенсорні вимірювання, достатні для навчання та оцінки політики навчання.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Peter Karkus, Boris Ivanovic, Shie Mannor, and Marco Pavone. “DiffStack: A Differentiable and Modular Control Stack for Autonomous Vehicles”. 6-а щорічна конференція з навчання роботів, 2022.

2. Delphi's autonomous car [Електронний ресурс] – Режим доступу до ресурсу: https://www.wired.com/2015/04/delphi-autonomous-car-cross-country/.

3. On the road - Waymo [Електронний ресурс] – Режим доступу до ресурсу: https://web.archive.org/web/20180323062918/https://waymo.com/ontheroad/.

4. Yulong Cao, Chaowei Xiao, Benjamin Cyr, Yimeng Zhou, Won Park, Sara Rampazzi, Qi Alfred Chen, Kevin Fu, and Z Morley Mao. Adversarial sensor attack on lidar-based perception in autonomous driving. У матеріалах конференції ACM SIGSAC 2019 з комп'ютерної та комунікаційної безпеки, с. 2267–2281, 2019.

5. Yulong Cao, Chaowei Xiao, Dawei Yang, Jing Fang, Ruigang Yang, Mingyan Liu, and Bo Li. Adversarial objects against lidar-based autonomous driving systems. Препринт arXiv arXiv:1907.05418, 2019.

6. Jiachen Sun Sun, Yulong Cao Cao, Qi Alfred Chen, and Z Morley Mao. Towards robust lidar-based perception in autonomous driving: General black-box adversarial sensor attack and countermeasures. На симпозіумі USENIX Security Symposium (Usenix Security'20), 2020.

7. Yulong Cao, Ningfei Wang, Chaowei Xiao, Dawei Yang, Jin Fang, Ruigang Yang, Qi Alfred Chen, Mingyan Liu, and Bo Li. Invisible for both camera and lidar: Security of multi-sensor fusion based perception in autonomous driving under physical-world attacks. У 2021 році IEEE Symposium on Security and Privacy (SP), с. 176–194. IEEE, 2021.

8. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y. Generative adversarial nets. У: Досягнення нейронних систем обробки інформації. 2014: с. 2672–2680.

9. Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. 2018. Generating adversarial examples with adversarial networks. arXiv:1801.02610

10. Florian Tram`er, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. The space of transferable adversarial examples. Препринт arXiv , 2017.

11. Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. Advances in neural information processing systems, 29, 2016.

12. Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. На Міжнародній конференції з машинного навчання, с. 40–49. PMLR, 2018.

13. Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan. Point- flow: 3d point cloud generation with continuous normalizing flows. У матеріалах

Міжнародної конференції з комп'ютерного зору IEEE/CVF, с. 4541–4550, 2019.

14. Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. У матеріалах конференції IEEE з комп'ютерного зору та розпізнавання образів, с. 4460–4470, 2019.

15. Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. Матеріали конференції IEEE з комп'ютерного зору та розпізнавання образів (CVPR), 2019.

16. Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. Конференція ECCV, 2020.

17. Eric Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. У матеріалах конференції CVPR, 2021.

18. Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. У матеріалах конференції IEEE/CVF з комп'ютерного зору та розпізнавання образів, с. 16123–16133, 2022.

19. William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. Комп'ютерна графіка ACM signgraph, 21(4): с. 163–169, 1987.

20. Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. Конференція ICCV, 2017.

21. Tianchang Shen, Jun Gao, Kangxue Yin, Ming-Yu Liu, and Sanja Fidler. Deep marching tetrahedra: a hybrid representation for high-resolution 3d shape synthesis. Досягнення в галузі нейронних систем обробки інформації (NeurIPS), 2021.

22. Samuli Laine, Janne Hellsten, Tero Karras, Yeongho Seol, Jaakko Lehtinen, and Timo Aila. Modular primitives for high-performance differentiable rendering. ACM Transactions on Graphics, 39(6), 2020.

23. Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. У матеріалах конференції CVPR, 2020.

24. Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. У матеріалах конференції IEEE/CVF з комп'ютерного зору та розпізнавання образів, с. 2446–2454, 2020.

25. Glenn Jocher, Ayush Chaurasia, Alex Stoken, Jirka Borovec, NanoCode012, Yonghye Kwon, Kalen Michael, TaoXie, Jiacong Fang, imyhxy, Lorna, Zeng Yifu, Colin Wong, Abhiram V, Diego Montes, Zhiqiang Wang, Cristi Fati, Jebastin Nadar, Laughing, UnglvKitDe, Victor Sonck, tkianai, yxNONG, Piotr Skalski, Adam Hogan, Dhruv Nair, Max Strobel, and Mrinal Jain. ultralytics/yolov5: v7.0 — YOLOv5 SOTA Realtime Instance Segmentation, November 2022.

26. Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. Досягнення в галузі нейронних систем обробки інформації (NeurIPS), 2015.

27. Detectron2 [Електронний ресурс] / [Y. Wu, A. Kirillov, F. Massa та ін.]. – 2019. – Режим доступу до ресурсу: https://github.com/facebookresearch/detectron.

28. Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. У матеріалах конференції IEEE з комп'ютерного зору та розпізнавання образів, с. 779–788, 2016.

29. Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. Європейська конференція з комп'ютерного зору, с. 21–37. Springer, 2016.

30. Yun Chen, Frieda Rong, Shivam Duggal, Shenlong Wang, Xinchen Yan, Sivabalan Manivasagam, Shangjie Xue, Ersin Yumer, and Raquel Urtasun. Geosim: Realistic video simulation via geometry-aware com position for self-driving. У матеріалах конференції IEEE/CVF з комп'ютерного зору та розпізнавання образів (CVPR), с. 7230-7240, червень 2021 р..

31. A. Amini, T. H. Wang, I. Gilitschenski, W. Schwarting, Z. J. Liu, S. Han, S. Karaman, and D. Rus, “ VISTA 2.0: An open, data-driven simulator for multimodal sensing and policy learning for autonomous vehicles,” У матеріалах міжнародної конференції з робототехніки та автоматизації (ICRA), 2022, с. 2419–2426.

32. Lowe DG. 2004 SIFT: scale invariant feature transform. Журнал Comput. Vision, с. 91–110.

33. He, K. M.; Zhang, X. Y.; Ren, S. Q.; Sun, J. Deep residual learning for image recognition. У матеріалах конференції IEEE з комп'ютерного зору та розпізнавання образів, с. 770–778, 2016.

34. Szegedy, C.; Liu, W.; Jia, Y. Q.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. У матеріалах конференції IEEE з комп'ютерного зору та розпізнавання образів, с.1-9, 2015.

35. Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. Препринт arXiv:1409.1556, 2014.

36. Vaswani, A.; Shazeer, N. M.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; Polosukhin, I. Attention is all you need. У матеріалах 31-ї Міжнародної конференції з нейронних систем обробки інформації, с. 6000–6010, 2017.

37. Devlin, J.; Chang, M. W.; Lee, K.; Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. Препринт arXiv arXiv:1810.04805, 2018.

38. Diederik P. Kingma and Max Welling. 2013. Auto-encoding variational bayes. arXiv:1312.6114.

39. Jun Gao, Tianchang Shen, Zian Wang, Wenzheng Chen, Kangxue Yin, Daiqing Li, Or Litany, Zan Gojcic, and Sanja Fidler. Get3d: A generative model of high quality 3d textured shapes learned from images. Досягнення в галузі нейронних систем обробки інформації, 2022.

40. Tianchang Shen, Jun Gao, Kangxue Yin, Ming-Yu Liu, and Sanja Fidler. Deep marching tetrahedra: a hybrid representation for high-resolution 3d shape synthesis. Досягнення в галузі нейронних систем обробки інформації (NeurIPS), 2021.

41. Redmon J, Farhadi A. Yolo9000: better, faster, stronger. В кн.: Матеріали конференції IEEE з комп'ютерного зору та розпізнавання образів. 2017. с. 6517–25

42. Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. Препринт arXiv arXiv:1804.02767, 2018.

43. Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. Препринт arXiv arXiv:2004.10934, 2020.

44. Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. Конференція CVPR, 2017.

45. Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. Конференція ECCV. 2014.

46. D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, arXiv:1412.6980v9

47. H. Caesar et al., “nuScenes: A multimodal dataset for autonomous driving,” У матеріалах IEEE конференції з комп’ютерного зору та розпізнавання патернів, 2020, с. 11621–11631.

48. F. Yu et al., “BDD100K: A diverse driving dataset for heterogeneous multitask learning,” 2018, arXiv:1805.04687.