

МАГІСТЕРСЬКА РОБОТА

МР. ШМ - 01.00.00.000 ПЗ

Група ШМ-24-1

Бабій Володимир

2025

Івано-Франківський національний технічний університет нафти і газу

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення

Бабій Володимир Олексійович

(прізвище, ім'я, по батькові)

УДК 004.9
(індекс)

МАГІСТЕРСЬКА РОБОТА

Моделі генерації даних про оцінку позиціювань об'єктів та аналізу їх

поведінки

(назва роботи)

Інженерія програмного забезпечення

(назва освітньої програми)

121 - Інженерія програмного забезпечення

(шифр і назва спеціальності)

Бабій В.О.

(підпис, ініціали та прізвище здобувача освітнього ступеня)

Науковий керівник Юрчишин Володимир Миколайович, д.т.н., професор

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Допущено до захисту

Завідувач кафедри

доц. Бандура В.В.

(посада) (підпис) (дата) (ініціали та прізвище)

Нормоконтроль

доц. Вовк Р.Б.

(посада) (підпис) (дата) (ініціали та прізвище)

Робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

Івано-Франківськ – 2025

Івано-Франківський національний технічний університет нафти і газу

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення

Освітній рівень магістр

Спеціальність 121 – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:

Зав. кафедрою

ІІЗ

доц.

В.В. Бандура

“ 04 ” вересня 2025 р.

ЗАВДАННЯ

НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

Бабію Володимиру Олексійовичу

(прізвище, ім'я, по-батькові)

1. Тема магістерської роботи “Моделі генерації даних про оцінку позиціювань об'єктів та аналізу їх поведінки”

керівник проекту (роботи) Юрчишин В.М., д.т.н., професор

затверджені наказом закладу вищої освіти від “ 05 ” листопада 2025 р. № 695/7

2. Строк подання студентом проекту (роботи) 15 грудня 2025 р.

3. Вихідні дані до проекту (роботи) Теоретичні концепції та формальні моделі побудови та функціонування інформаційних та програмних технологій певного класу

4. Зміст розрахунково - пояснювальної записки(перелік питань, які потрібно розробити)

1. Аналіз предметної області використання технік комп'ютерного зору для задач генерації даних

2. Моделі та алгоритми оцінки позиціювання та виявлення об'єктів

3. Реалізація моделі генерації даних про оцінку позиціювань об'єктів та аналізу їх поведінки

4. Навчання YOLOv11-Pose на довільному наборі даних

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

1. Приклади зображень з набору даних SURREAL (рис. 1.1)

2. Приклади зображень з набору даних Human3.6M (рис. 1.2)

3. Процес захоплення руху в наборі даних Human3.6M (рис. 1.3)

4. Приклади зображень з набору даних Rapoptic (рис. 1.4)

5. Ключові характеристики OpenPose (рис. 1.5)

6. Консультанти розділів проекту (роботи)

Розділ	Консультант	Підпис, дата
Перевірка на плагіат	доц., к.т.н. Вовк Р.Б.	

7. Дата видачі завдання 04 вересня 2025 р.

Керівник _____

(підпис)

Завдання прийняв до виконання _____

(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назви етапів магістерської роботи	Строк виконання етапів роботи	Примітка
1	Підбір і вивчення літератури по темі магістерської роботи	15.09.2025	виконано
2	Аналіз предметної області використання технік комп'ютерного зору для задач генерації даних	29.09.2025	виконано
3	Моделі та алгоритми оцінки позиціонування та виявлення об'єктів	15.10.2025	виконано
4	Реалізація моделі генерації даних про оцінку позиціонувань об'єктів та аналізу їх поведінки	08.11.2025	виконано
5	Навчання YOLOv11-Pose на довільному наборі даних	20.11.2025	виконано
6	Реалізація функціональності запропонованої технології	01.12.2025	виконано
7	Затвердження пояснювальної записки роботи завідувачем кафедри	15.12.2025	виконано

Студент – магістр _____

(підпис)

Керівник роботи _____

(підпис)

АНОТАЦІЯ

Магістерська робота: 79 с., 25 рис., 4 табл., 34 джерела.

Тема: Моделі генерації даних про оцінку позиціювань об'єктів та аналізу їх поведінки

Об'єкт дослідження: процес генерації даних для систем комп'ютерного зору.

Мета роботи: розробка та дослідження моделі генерації синтетичних даних для задач оцінки позиціювань об'єктів і аналізу їх поведінки з використанням методів комп'ютерного зору та симуляційних середовищ

Предмет дослідження: моделі та методи генерації синтетичних даних для оцінки позиціювань об'єктів та аналізу їх поведінки.

Результати дослідження

В роботі розроблено модель генерації синтетичних даних для задач оцінки позиціювань та аналізу поведінки об'єктів, що поєднує мультимедійні візуалізації, запропоновано архітектуру системи, яка забезпечує модульність, масштабованість та інтеграцію з сучасними алгоритмами комп'ютерного зору.

Висновок

Запропоновано комплексний підхід до моделювання генерації даних про оцінку позиціювань та поведінку об'єктів, що поєднує методи комп'ютерного зору, симуляційні технології та сучасні алгоритми детекції.

КОМП'ЮТЕРНИЙ ЗІР, СИНТЕТИЧНІ ДАНІ, ОЦІНКА ПОЗИ, ГЕНЕРАЦІЯ ДАНИХ, СИМУЛЯЦІЙНЕ СЕРЕДОВИЩЕ, OPENPOSE, YOLO, АНАЛІЗ ПОВЕДІНКИ ОБ'ЄКТІВ, МОДЕЛІ ГЛИБИННОГО НАВЧАННЯ.

ABSTRACT

Master Thesis: 79 pp., 25 fig., 4 tab., 34 sources.

Topic: Data generation models for estimating object positioning and analyzing their behavior

Object of research: data generation process for computer vision systems.

Purpose of work: development and research of a synthetic data generation model for tasks of estimating object positioning and analyzing their behavior using computer vision methods and simulation environments

Subject of research: models and methods of generating synthetic data for estimating object positioning and analyzing their behavior.

Research results

The work develops a synthetic data generation model for tasks of estimating object positioning and analyzing object behavior, which combines multi-format visualizations, and proposes a system architecture that provides modularity, scalability, and integration with modern computer vision algorithms.

Conclusion

A comprehensive approach to modeling the generation of data for estimating object positioning and behavior, which combines computer vision methods, simulation technologies, and modern detection algorithms, is proposed.

COMPUTER VISION, SYNTHETIC DATA, POSE ESTIMATION, DATA GENERATION, SIMULATION ENVIRONMENT, OPENPOSE, YOLO, OBJECT BEHAVIOR ANALYSIS, DEEP LEARNING MODELS.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	10
ВСТУП.....	11
РОЗДІЛ 1. АНАЛІЗ ТА ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ВИКОРИСТАННЯ ТЕХНІК КОМП'ЮТЕРНОГО ЗОРУ ДЛЯ ЗАДАЧ ГЕНЕРАЦІЇ ДАНИХ	14
1.1. Особливості розробки генератора синтетичних даних для завдань комп'ютерного зору	14
1.1.2. Набори даних у комп'ютерному зорі	15
1.1.2. Виклики, пов'язані з традиційними наборами даних.....	16
1.1.3. Синтетичні набори даних як альтернатива	17
1.2. Синтетичні набори даних у комп'ютерному зорі	19
1.2.1. Аналіз набору даних SURREAL	20
1.2.2. Аналіз набору даних Human 3.6M	21
1.2.3. Аналіз набору даних CMU Panoptic.....	24
Висновки до розділу	25
РОЗДІЛ 2. МОДЕЛІ ТА АЛГОРИТМИ ОЦІНКИ ПОЗИЦІЮВАННЯ ТА ВИЯВЛЕННЯ ОБ'ЄКТІВ	27
2.1. Моделі оцінки пози як методи комп'ютерного зору	27
2.1.1. Модель OpenPose	28
2.2.2. YOLO модель (You Only Look Once).....	29
2.2. Середовища симуляції та інструментарії	31
2.2.1. Симулятор CARLA	32
2.2.3. Набір інструментів UnrealCV	35
2.3. Аналіз та порівняння пропонованої системи генерації даних про оцінку позиціювань об'єктів з існуючими рішеннями.....	36
2.3.1. Порівняння з платформою CARLA	36

2.3.2. Порівняння з платформою UnrealCV.....	36
2.4. Реалізація дизайну та архітектури проєкту	37
2.4.1. Побудова діаграм класів компонент системи	40
Висновки до розділу	46

РОЗДІЛ 3. РЕАЛІЗАЦІЯ МОДЕЛІ ГЕНЕРАЦІЇ ДАНИХ ПРО ОЦІНКУ

ПОЗИЦІЮВАНЬ ОБ'ЄКТІВ ТА АНАЛІЗУ ЇХ ПОВЕДІНКИ	47
3.1. Реалізація основних компонент системи.....	47
3.1.1 Карти	47
3.1.2. Сценарії	48
3.2. Процес розробки сценаріїв	50
3.2.1. Підготовка та оптимізація	51
3.2.2. Класифікація анімацій	52
3.3. Представлення робочого процесу захоплення зображень	52
3.3.1. Налаштування камери та управління даними	54
3.3.2. Генерація кольорового зображення	56
3.3.3. Реалізація поля глибини карти	57
3.3.4. Реалізація карти нормалей.....	58
3.3.5. Зображення сегментації.....	60
3.4. Процес анотування даних	62
3.4.1. Механізм анотування.....	62
3.4.2. Зберігання та управління даними	64
3.4.3. Трансформація координат	65
3.5. Дослідження і тестування симуляційного середовища	66
3.5.1. Процес генерації даних.....	66
3.5.2. Потік даних з камери	68
3.6 Навчання YOLOv11-Pose на довільному наборі даних	68
3.6.1. Підготовка даних та трансформування	69
3.6.2. Навчання та результати	70
Висновки до розділу	73

ВИСНОВКИ	74
ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	76

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

КЗ - комп'ютерний зір

CV - computer vision

HPE - Human Pose Estimation

YOLO – You Only Look Once

VRAM – Video Random-Access Memory – Відеопам'ять з довільним доступом

COCO – Common Objects in Context – Загальні об'єкти в контексті

CVPR – Computer Vision and Pattern Recognition – Комп'ютерне бачення та розпізнавання образів

ICML – International Conference on Machine Learning – Міжнародна конференція з машинного навчання

NIPS – Neural Information Processing Systems – Системи обробки нейронної інформації

ECCV – European Conference on Computer Vision – Європейська конференція з комп'ютерного зору

ВСТУП

Актуальність теми.

Сучасні інформаційні технології дедалі більше інтегруються у різні сфери людської діяльності, а системи комп'ютерного зору відіграють у цьому процесі провідну роль. Завдяки здатності до автоматизованої інтерпретації візуальної інформації, вони широко застосовуються у робототехніці, автономному транспорті, медицині, безпекових системах, спортивній аналітиці та інтерактивних віртуальних середовищах. Однією з ключових задач комп'ютерного зору є оцінка пози та виявлення об'єктів, що потребує використання великих обсягів якісних даних для навчання та тестування моделей.

Традиційні набори даних мають суттєві обмеження: складність збору та маркування, високу вартість створення, недостатнє різноманіття сценаріїв та обмеженість у відображенні реальних умов. У цьому контексті синтетичні дані, що генеруються симуляційними середовищами, стають перспективною альтернативою. Вони дозволяють створювати багатоваріантні умови, автоматизувати анотацію та забезпечувати масштабованість.

Розробка моделей генерації даних для оцінки позиціювань об'єктів відкриває нові можливості для дослідження їх поведінки, підвищення точності роботи алгоритмів комп'ютерного зору та адаптації систем до динамічних умов. Саме цим зумовлена наукова та прикладна значущість даної роботи.

Актуальність дослідження зумовлена зростаючою потребою у високоточних, масштабованих та гнучких моделях для аналізу візуальної інформації. Сучасні підходи до збору реальних даних є дорогими, обмеженими у варіативності та часто не відповідають вимогам до комплексного відображення поведінки об'єктів у різних середовищах. У той же час синтетичні набори даних, створені за допомогою симуляційних платформ, надають можливість моделювати практично необмежені сценарії,

змінювати умови середовища та отримувати точні анотації без значних витрат ресурсів.

Особливої актуальності набуває створення системи, здатної поєднувати механізми генерації різних типів візуальних даних (кольорових зображень, карт глибини, нормалей, сегментаційних карт) з алгоритмами оцінки пози та аналізу поведінки. Це дозволить не лише підвищити точність існуючих моделей, а й створити умови для подальшого розвитку автономних систем, що функціонують у складних реальних середовищах.

Метою магістерської роботи є розробка та дослідження моделі генерації синтетичних даних для задач оцінки позиціювань об'єктів і аналізу їх поведінки з використанням методів комп'ютерного зору та симуляційних середовищ.

Завдання дослідження:

- Виконати аналіз предметної області та визначити переваги й обмеження традиційних та синтетичних наборів даних.
- Дослідити сучасні моделі оцінки пози та алгоритми виявлення об'єктів (OpenPose, YOLO).
- Проаналізувати існуючі симуляційні середовища (CARLA, UnrealCV) та визначити їх можливості для генерації даних.
- Розробити архітектуру системи генерації синтетичних даних для оцінки пози та поведінки об'єктів.
- Реалізувати механізми побудови сценаріїв, візуалізації та анотування даних.
- Провести експериментальні дослідження роботи системи та оцінити ефективність навчання моделей.

Об'єкт дослідження - процес генерації даних для систем комп'ютерного зору.

Предмет дослідження - моделі та методи генерації синтетичних даних для оцінки позиціювань об'єктів та аналізу їх поведінки.

Методи дослідження

- методи комп'ютерного зору та глибинного навчання для виявлення й оцінки пози об'єктів;
- симуляційні методи для створення віртуальних середовищ та сценаріїв поведінки;
- методи формування та анотування синтетичних наборів даних;
- експериментальні методи тестування моделей на різних наборах даних;

Наукова новизна

Розроблено модель генерації синтетичних даних для задач оцінки позиціонувань та аналізу поведінки об'єктів, що поєднує мультимедійні візуалізації (кольорові зображення, карти глибини, карти нормалей, сегментаційні карти). Запропоновано архітектуру системи, яка забезпечує модульність, масштабованість та інтеграцію з сучасними алгоритмами комп'ютерного зору.

Практичне застосування

Розроблена система може бути використана для навчання та тестування алгоритмів комп'ютерного зору у сфері автономного транспорту, дослідження та відпрацювання сценаріїв у робототехніці та створення реалістичних навчальних даних для медичних систем діагностики та моніторингу.

Структура магістерської роботи. Робота складається зі вступу, трьох розділів та висновків. Загальний обсяг роботи становить 79 сторінок, і містить 25 рисунків, 4 таблиці, список використаних джерел із 34 найменувань.

РОЗДІЛ 1. АНАЛІЗ ТА ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ВИКОРИСТАННЯ ТЕХНІК КОМП'ЮТЕРНОГО ЗОРУ ДЛЯ ЗАДАЧ ГЕНЕРАЦІЇ ДАНИХ

1.1. Особливості розробки генератора синтетичних даних для завдань комп'ютерного зору

Комп'ютерний зір — це добре усталена та широко досліджена галузь штучного інтелекту, яка зосереджується на розробці інтелектуальних систем, здатних витягувати значущу інформацію з візуальних даних та приймати обґрунтовані рішення або виконувати відповідні дії на основі цієї інформації [1]. По суті, КЗ надає комп'ютерам і роботам здатність візуально сприймати та розуміти навколишнє середовище [1]. Поширені застосування КЗ включають виявлення та відстеження об'єктів, семантичну сегментацію, оцінку пози, структуру з руху та інші [2].

Протягом останнього десятиліття комп'ютерний зір (КЗ) став незамінним компонентом у численних застосуваннях відеоспостереження та інтелектуальних автономних систем. Завдяки новітнім досягненням у галузі штучного інтелекту, edge computing та Інтернету речей (IoT), спостерігається стрімке зростання кількості інтелектуальних камер, а також промислових і споживчих роботів, які використовують КЗ для різноманітних завдань, починаючи від відстеження та аналізу людей і класифікації об'єктів, і до візуального контролю та виявлення аномалій. Для більшості стандартних застосувань розробка систем КЗ є доступним завданням завдяки значній кількості загальнодоступних наборів даних і наукових публікацій. Однак створення систем КЗ для специфічних застосувань зі складними сценаріями використання може бути надзвичайно трудомістким, зокрема через значні витрати часу та ресурсів на створення спеціалізованих наборів даних. У середовищах, що передбачають взаємодію з людьми, що часто регулюється

нормами безпеки та відповідності, збір даних може становити додаткові суттєві виклики.

У цій роботі пропонується система для генерації реалістичних синтетичних зображень з автоматичним маркуванням та анотацією даних. Дане рішення використовує можливості 3D-рендерингу та анімації Unreal Engine 5 (UE5), що дозволяє дослідникам створювати високоякісні симульовані середовища з мінімальними зусиллями. Ми продемонстрували, що така система може ефективно генерувати достатню кількість зображень для навчання моделей глибокого навчання, що вирішують складні завдання. Крім того, набори даних, згенеровані системою, можуть бути модифіковані, розширені та регенеровані в будь-який момент, що надає значно більшу гнучкість порівняно з традиційними підходами.

1.1.2. Набори даних у комп'ютерному зорі

Набір зображень є критично важливим компонентом у процесі навчання будь-якого проєкту КЗ, що базується на глибокому навчанні, особливо для завдань, пов'язаних із виявленням об'єктів та оцінкою пози [1, 3]. У більшості випадків надзвичайно складно розробити модель, яка надійно функціонує в реальних сценаріях та забезпечує точні прогнози без достатньої кількості навчальних даних. З цієї причини випуск багатьох великомасштабних наборів даних зображень протягом останнього десятиліття значно сприяв розвитку багатьох сучасних моделей КЗ. Популярними прикладами є PASCAL VOC [4], CIFAR-100 [5], ImageNet [6] та COCO MS [7].

ImageNet - великомасштабна база даних зображень, призначена для досліджень у галузі розпізнавання об'єктів [6]. Започаткований у 2006 році, проєкт мав на меті створити всебічний ресурс для навчання моделей глибокого навчання.

Основна мета ImageNet — надати стандартизований набір даних для тестування та порівняння алгоритмів розпізнавання об'єктів. База даних

містить мільйони зображень, розділених на тисячі класів, що робить її ідеальною для глибинного навчання. Набір даних містить понад 14 мільйонів зображень, кожне з яких анотовано мітками, що відповідають понад 20 000 категоріям, організованим згідно з ієрархією WordNet [6].

- MS COCO: представлений через вісім років після ImageNet, цей набір даних надає ширшу підтримку для різноманітніших завдань КЗ, надаючи анотаційні дані, придатні для сегментації екземплярів та підписування зображень [7]. COCO містить понад 330 000 зображень з детальними анотаціями для понад 2,5 мільйона позначених екземплярів у 80 категоріях об'єктів. Він наголошує на зйомці об'єктів у їх природному середовищі та надає сегментацію на рівні екземплярів, що забезпечує точну локалізацію об'єктів [7].

1.1.2. Виклики, пов'язані з традиційними наборами даних

Розробка великомасштабних наборів даних, таких як MS COCO, часто супроводжується значними викликами, особливо коли вони повинні підтримувати складні анотації, як-от семантична сегментація та оцінка пози [7]. На відміну від традиційних наборів даних, які призначають прості мітки класів, більш просунуті анотації вимагають від анотаторів малювати точні межі об'єктів та ідентифікувати конкретні ключові точки — обидва процеси є надзвичайно трудомісткими [7, 8]. Крім того, створення таких детальних анотацій вимагає високого рівня експертних знань для забезпечення точності та послідовності. Незважаючи на ретельні зусилля, людські помилки часто є неминучими в будь-яких наборах даних, що вимагають ручного анотування. Наприклад, набір даних MS COCO зазнав кількох переглядів для виправлення помилок анотацій та невідповідностей [8].

Хоча великомасштабні публічні набори даних надають міцну основу, вони часто не відповідають специфічним застосуванням [9, 10]. Для тонкого налаштування моделей для спеціалізованих завдань часто потрібні спеціалізовані набори даних, оскільки загальні набори даних можуть не

відображати унікальні налаштування певних середовищ розгортання. Наприклад, зображення, зроблені наземними мобільними роботами, значно відрізняються від тих, що знімаються камерами спостереження зверху. Це призводить до зниження продуктивності моделей, навчених виключно на загальних наборах даних [9]. Спеціалізовані набори даних допомагають подолати цей розрив, включаючи відповідні кути камери, умови освітлення та фактори навколишнього середовища, що забезпечує навчання моделей на даних, які максимально наближені до реальних сценаріїв розгортання [10].

1.1.3. Синтетичні набори даних як альтернатива

Останніми роками синтетичні набори даних набули значної популярності як економічно ефективна та ефективна альтернатива традиційним наборам даних реального світу [13]. Збір та анотування реальних даних є ресурсомістким процесом, що вимагає дорогого обладнання та значної кількості ручної праці [9]. Навпаки, синтетичні дані можуть генеруватися набагато швидше і часто з нижчими витратами, що робить їх більш доступними.

Симуляційні середовища надають повний контроль над різними параметрами, такими як умови освітлення, кути камери, розташування об'єктів і навіть складні сценарії, такі як зміни погоди [14]. Цей рівень налаштування дозволяє створювати дані, спеціально адаптовані до потреб дослідників. Ще однією значною перевагою є можливість автоматично генерувати анотації. У симуляційних середовищах властивості об'єктів, такі як позиції, межі та мітки, відомі за замовчуванням, що дозволяє створювати точні та всебічні анотації без втручання людини [13, 15]. Автоматизація анотування не тільки допомагає зменшити ручну працю, але й забезпечує послідовність і точність у всьому наборі даних, що є критично важливим для навчання надійних моделей [13].

До недавнього часу синтетичні набори даних використовувалися переважно в дослідженнях автономного водіння, оскільки вони надають

контрольоване та безпечне середовище для тестування технологій самостійного водіння [10, 18, 19]. Натомість, використання синтетичних наборів даних для аналізу пози людини залишається перспективною, але недостатньо дослідженою областю. Існуючі синтетичні набори даних, такі як SURREAL [23] та Human 3.6M [24], які були інноваційними на час їх створення, не використовують останні досягнення в технологіях рендерингу і покладаються на системи захоплення руху (MoCap), що обмежує їх можливості налаштування.

У цій роботі ми представляємо генератор синтетичних наборів даних, спеціально розроблений для оцінки пози та аналізу поведінки. Генератор надає надійне симуляційне середовище та робочі процеси для створення та запису спеціальних сценаріїв, що включають людей зі складними рухами та взаємодіями. Він забезпечує автоматизоване анотування «з коробки» з анотацією пози, що відповідає стандарту OpenPose [25]. Використовуючи можливості рендерингу останньої версії Unreal Engine, дана система може генерувати високореалістичні зображення разом з додатковими анотаційними даними, такими як поле глибини, карта нормалей та семантична сегментація.

На відміну від існуючих синтетичних наборів даних, які покладаються на системи захоплення руху, система усуває необхідність у захопленні руху, використовуючи наявні ресурси анімації людини, доступні в Інтернеті. Крім генерації наборів даних для навчання моделей оцінки пози, система також підтримує оцінку виводу, відкриваючи потік камери зсередини симулятора, до якого можуть отримувати доступ зовнішні програми на хост-комп'ютері. Наше дослідження демонструє, що генерація синтетичних наборів даних пропонує гнучку та економічно ефективну альтернативу традиційним наборам даних. Ця робота представляє значний прорив у галузі синтетичних наборів даних для КЗ, представляючи один з перших генераторів наборів даних, спеціально розроблених для симуляції сценаріїв, орієнтованих на людину, з автоматичними анотаціями, що відповідають стандарту OpenPose.

1.2. Синтетичні набори даних у комп'ютерному зорі

Синтетичні набори даних зображень є переконливою альтернативою традиційним наборам даних, що вимагають інтенсивної ручної праці. Термін «синтетичний набір даних» позначає колекцію даних, яка генерується вибірково комп'ютером, на відміну від збору даних з реальних сценаріїв [15]. Цей підхід є мотивованим необхідністю вирішення численних викликів та обмежень, з якими стикаються при створенні наборів даних реального світу.

Для багатьох специфічних проєктів комп'ютерного зору розробка спеціальних наборів даних часто характеризується значними ресурсовитратами та схильністю до людських помилок [9]. Крім того, більшість цих спеціальних наборів даних є пропрієтарними та специфічними для певної предметної області, що обмежує їх загальну корисність і розширюваність. У зв'язку з цим використання синтетичних даних для навчання моделей глибокого навчання привернуло значну увагу дослідників завдяки їхньому потенціалу для пом'якшення труднощів, пов'язаних з ручним збором даних та їх анотуванням.

Використовуючи генератор синтетичних даних, теоретично можна генерувати необмежену кількість навчальних даних, у яких різні властивості та варіації можуть бути повністю налаштовані та розширені. Що ще важливіше, вартість анотування даних є майже нульовою, що значно підвищує їх масштабованість.

Проте, одним з основних викликів, пов'язаних з цим підходом, є так званий «розрив домену», який відрізняє синтетичні дані від даних реального світу [9]. Деякі моделі глибокого навчання, які були навчені виключно на синтетичних даних, демонстрували знижену продуктивність у реальних умовах, особливо при роботі із зображеннями, які часто піддаються різним проблемам реального світу, таким як складні умови освітлення або оклюзії.

В останні роки зросла кількість академічних досліджень, спрямованих на покращення надійності синтетичних даних та вивчення різних технік для

ефективного використання як даних реального світу, так і синтетичних даних у процесах навчання моделей комп'ютерного зору [9]. Ці дослідження спрямовані на подолання розриву між доменами та забезпечення того, щоб моделі, навчені на синтетичних даних, могли успішно узагальнювати знання на реальні сценарії.

1.2.1. Аналіз набору даних SURREAL

Synthetic Humans for Real Tasks (SURREAL) є значним великомасштабним синтетичним набором даних, призначеним для оцінки 3D-пози людини та аналізу поведінки [23]. Цей набір даних використовує дані захоплення руху, отримані з набору даних Human3.6M [24], для генерації різноманітних поз людських моделей та автоматичного анотування даних. Для забезпечення варіативності фонових зображень були використані знімки з набору даних Large-scale Scene Understanding (LSU) [26], що містить понад 400 000 зображень різних інтер'єрів (спальні, ванні кімнати, вітальні, кухні).

Автори проекту навмисно застосовували різні текстури одягу для підвищення реалістичності моделей. Крім того, набір даних SURREAL враховує варіації освітлення за допомогою моделі сферичних гармонік [23]. Цей підхід відрізняє його від попередника, оскільки SURREAL є одним із перших синтетичних наборів даних для аналізу пози людини, який надає одночасно 3D-анотації пози, результати семантичної сегментації та метадані точки огляду камери. Ця особливість робить його особливо придатним для багатозадачного навчання.

Незважаючи на інноваційність, SURREAL, представлений у 2016 році, не набув широкого визнання серед дослідників через низьку якість рендерингу графіки та брак автентичності порівняно з реальними зображеннями. Приклади зображень з набору даних представлені на рис. 1.1, а основні статистичні дані наведені в таблиці 1.1.

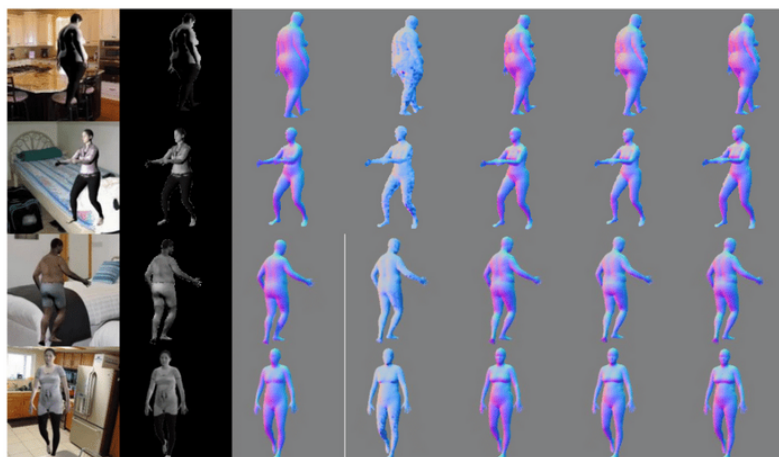


Рис. 1.1. Приклади зображень з набору даних SURREAL

Таблиця 1.1.

Ключові статистичні дані набору даних SURREAL

Атрибут	Деталі
Вихідні дані	Дані захоплення руху з набору даних Human3.6M
Кількість суб'єктів	145 (115 для навчання, 30 для тестування)
Кількість послідовностей	2607 загальних послідовностей (1964 для навчання, 703 для тестування)
Кількість кліпів	67 528 кліпів (55 001 для навчання, 12 528 для тестування)
Кількість кадрів	6 536 752 загальних кадрів (5 342 090 для навчання, 1 194 662 для тестування)
Фони	Понад 400 000 фонових зображень з набору даних LSU
Аугментація даних	Випадкова заміна фону та варіація освітлення за допомогою Сферичних Гармонік
Типи даних	Надає 3D-анотації пози, семантичну сегментацію та метадані точки огляду камери

1.2.2. Аналіз набору даних Human 3.6M

Набір даних Human3.6M, випущений у 2014 році, є ще одним значущим синтетичним набором даних для оцінки 3D-пози людини та аналізу її поведінки [24]. Він був розроблений з метою подолання обмежень попередніх наборів даних щодо масштабу та різноманітності, представляючи одну з найбільших колекцій зображень людської пози на той час.

Набір даних складається з приблизно 3,6 мільйона точних 3D-поз, захоплених від 11 професійних акторів (6 чоловіків і 5 жінок), які виконували 17 різних повсякденних дій, таких як дискусії, куріння, фотографування та розмови по телефону. Процес збору даних проводився в контрольованому внутрішньому середовищі з використанням високошвидкісної системи захоплення руху (MoCap), що включала 10 камер. Це налаштування забезпечило точне відстеження 3D-позицій та кутів суглобів. Крім того, для отримання синхронізованих відеоданих з декількох ракурсів використовувалися чотири високороздільні відеокамери з прогресивним скануванням, що працювали на частоті 50 Гц. Така конфігурація дозволяє синхронізувати камери під час захоплення послідовностей руху, роблячи набір корисним для аналізу 3D-поз на основі даних з кількох камер.

Для підвищення узагальнювальної здатності моделей, навчених на цьому наборі даних, застосовувалися техніки аугментації. Оригінальні фони зображень були відокремлені, а замість них застосовувалися випадкові фони. Це додає варіативності до оригінального набору даних та дозволяє моделям краще працювати в різноманітних реальних сценаріях. Крім зображень, Human3.6M також включає дані часу польоту (time-of-flight) та результати 3D-сканування лазером акторів, надаючи мультимодальний набір даних, який може бути використаний у широкому спектрі дослідницьких застосувань.

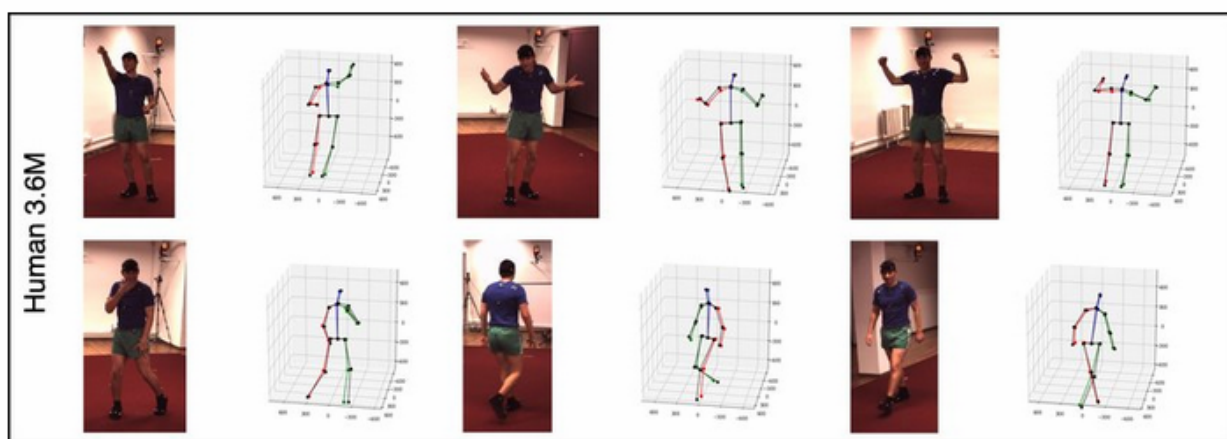


Рис. 1.2. Приклади зображень з набору даних Human3.6M

На рис. 1.2 представлено зразки зображень з набору даних, а ключові статистичні дані підсумовано в таблиці 2.2. Крім того, на рис. 1.3 наведено приклади процесу захоплення руху, використаного при розробці цього набору даних.

Таблиця 1.2.

Ключові статистичні дані набору даних Human3.6M

Атрибут	Деталі
Кількість поз	3,6 мільйона 3D-поз людей
Суб'єкти	11 акторів (6 чоловіків, 5 жінок)
Сценарії	17 видів діяльності (наприклад, дискусія, куріння, фотографування)
Камери	4 синхронізовані високороздільні відеокамери (50 Гц)
Система захоплення руху	10-камерне налаштування для точного відстеження 3D-суглобів
Додаткові дані	Дані часу польоту та 3D-сканування лазером
Аугментація даних	Випадкова заміна фону для покращення узагальнення



Рис. 1.3. Процес захоплення руху в наборі даних Human3.6M

1.2.3. Аналіз набору даних CMU Panoptic

CMU Panoptic є набором даних, випущеним у 2017 році для просування досліджень у галузі оцінки 3D-пози людини та аналізу її поведінки [27]. Для його створення дослідники сконструювали складну куполоподібну структуру, оснащену понад 500 високоякісними камерами та 10 синхронізованими RGB-D сенсорами. Це створило одну з найсучасніших систем захоплення руху на той час, що забезпечувала 360-градусний огляд захопленої діяльності людини та гарантувала високоточний збір даних з багатьох ракурсів [27].

На відміну від багатьох існуючих наборів даних, обмежених щодо демографії суб'єктів і типів руху, набір даних Panoptic включає широке коло учасників, від дітей до людей похилого віку. Це не лише підвищує репрезентативність набору, але й дозволяє вивчати рух людини в різних вікових групах [27]. Крім того, послідовності руху фіксуються в різноманітних сценаріях, включаючи повсякденні дії, складні взаємодії та соціальні зібрання, що робить набір придатним для широкого спектра дослідницьких застосувань [27].

Таблиця 1.3.

Ключові статистичні дані набору даних CMU Panoptic L

Атрибут	Деталі
Рік випуску	2017
Установа	Університет Карнегі-Меллона
Налаштування камери	480 VGA (640×480, 25 fps), 31 HD (1920×1080, 30 fps), 10 Kinect v2 (RGB + Depth, 30 fps)
Загальні дані	65 послідовностей, 5,5 годин запису
3D-скелети	1,5 мільйона, 19 ключових точок на скелет
Діяльність	Вправи руху, торг, гра в мафію, танці, музика, взаємодія малюків
Анотації	3D/2D ключові точки, сегментація, обмежувальні прямокутники
Калібрування	Повні дані калібрування камери



Рис. 1.4. Приклади зображень з набору даних Panoptic

Набір даних Panoptic містить 19 ключових точок для кожного людського скелета, що представляють основні суглоби та орієнтири, такі як голова, шия, ніс, плечі, лікті, зап'ястя, стегна, коліна, щиколотки та тулуб. Кожна ключова точка надається у 3D-просторі з координатами (x, y, z) у метрах, вирівняних з глобальною системою координат. Додатково надаються 2D-проекції цих ключових точок для кожного виду камери, виражені як координати пікселів (x, y) на площині зображення. Ці дані розраховуються за допомогою детальних даних калібрування камери, які включають як внутрішні параметри (фокусна відстань, головна точка, коефіцієнти спотворення), так і зовнішні параметри (матриця обертання, вектор переміщення) для всіх камер. Це дозволяє здійснювати точну багатовидову реконструкцію [27].

Висновки до розділу

У результаті аналізу предметної області встановлено, що використання традиційних наборів даних у комп'ютерному зорі має низку обмежень: висока вартість збору, трудомістке анотування та обмежене різноманіття сценаріїв. Розгляд наборів SURREAL, Human3.6M та CMU Panoptic показав,

що, попри їхню популярність, вони не забезпечують універсальності та повного охоплення завдань оцінки пози. Синтетичні набори даних виявилися ефективною альтернативою, оскільки дозволяють відтворювати різні умови зйомки, варіативність середовищ і сценаріїв, а також автоматизувати процес анування. Таким чином, було обґрунтовано доцільність створення генератора синтетичних даних для задач комп'ютерного зору з метою підвищення точності та узагальнюваності моделей.

РОЗДІЛ 2. МОДЕЛІ ТА АЛГОРИТМИ ОЦІНКИ ПОЗИЦІЮВАННЯ ТА ВІЯВЛЕННЯ ОБ'ЄКТІВ

2.1. Моделі оцінки пози як методи комп'ютерного зору

Оцінка пози людини (Human Pose Estimation, HPE) відноситься до методів комп'ютерного зору, призначених для виявлення та оцінки позицій суглобів людини та інших попередньо визначених орієнтирів на основі візуальних даних, таких як зображення та відео. Шляхом вилучення детальної геометричної інформації про пози людини, HPE стала ключовим компонентом у багатьох застосуваннях штучного інтелекту, орієнтованих на людину, включаючи аналіз поведінки, моніторинг натовпу та взаємодію людини з роботом.

Людське тіло є надзвичайно складною системою з взаємопов'язаних суглобів і кінцівок, що керуються широким спектром кінематичних відносин [29]. Відповідно, моделювання людського тіла є важливою складовою HPE, метою якої є точне представлення анатомічних орієнтирів. Наразі існує три основні типи моделей, що використовуються для цієї мети: кінематичні, площинні та об'ємні моделі [29].

Кінематична модель, що представляє людське тіло як систему з N жорстких суглобів, залишається найпоширенішою завдяки своїй обчислювальній простоті та ефективності. Ця модель фіксує позиції та орієнтації суглобів для ефективного опису структури людського тіла. Однак, незважаючи на свою гнучкість, кінематичний підхід має помітні обмеження, зокрема в кодуванні дрібніших деталей, таких як форма тіла та текстура [28].

Оцінка пози людини загалом класифікується на дві основні категорії: оцінка 2D-пози та оцінка 3D-пози, що відрізняються наявністю додаткової просторової інформації у другому випадку [29]. Оцінка 2D-пози визначає розташування суглобів на двовимірній площині, використовуючи лише координати x та y . Натомість, оцінка 3D-пози включає інформацію про

глибину для створення просторово точнішого представлення пози людини в трьох вимірах [29]. У деяких передових архітектурах 2D-оцінка може слугувати проміжним етапом для побудови більш складних 3D-оцінок. У цьому розділі ми зосередимося на моделях оцінки 2D-пози, оскільки цей підхід залишається найпоширенішим у практичних застосуваннях [25].

2.1.1. Модель OpenPose

OpenPose є однією з найвпливовіших моделей для оцінки 2D-пози людини, відома своєю продуктивністю в реальному часі та здатністю виявляти пози кількох людей одночасно [25]. На відміну від методів «зверху-вниз» (top-down), які спочатку ідентифікують обмежувальний прямокутник для кожної людини, OpenPose використовує більш ефективний підхід «знизу-вгору» (bottom-up). Ця стратегія спочатку виявляє всі ключові точки на всьому зображенні, а потім групує їх у окремі пози людей [25]. Цей підхід робить OpenPose особливо ефективним у переповнених сценах з багатьма особами.

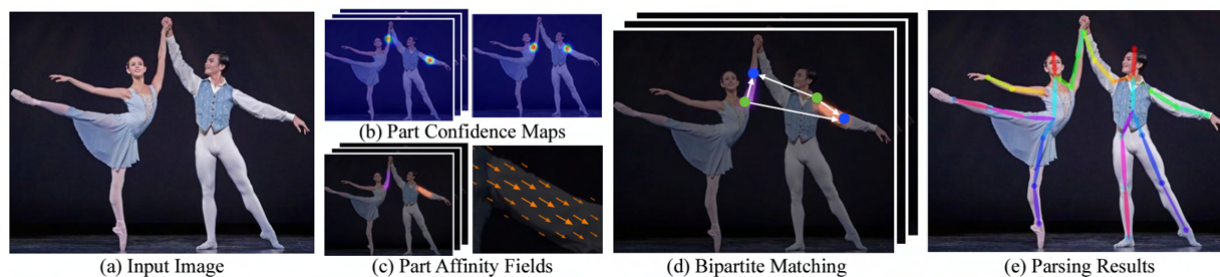


Рис. 2.1. Ключові характеристики OpenPose

Основною інновацією OpenPose є використання полів спорідненості частин (Part Affinity Fields, PAFs). PAFs — це серія 2D-векторних полів, які кодують просторові відносини та орієнтацію кінцівок тіла. Кожне поле фіксує напрямок від однієї частини тіла до іншої, що дозволяє моделі точно встановлювати зв'язки між ключовими точками, навіть в умовах оклюзії або

перекриття тіл. Ця здатність зберігати просторову узгодженість є критично важливою для формування точних і надійних структур людського тіла [25].

Архітектура OpenPose базується на багатоетапній згортковій нейронній мережі (CNN), яка використовує VGG-19 для вилучення ознак. Мережа потім розгалужує ці ознаки на два паралельні компоненти: один для оцінки карт впевненості для кожної ключової точки, а інший — для обчислення PAFs. Карти впевненості вказують на ймовірність наявності ключових точок у певних місцях, тоді як PAFs визначають зв'язки між ними. Ці два компоненти працюють ітеративно через серію етапів уточнення, де кожен етап використовує як попередні прогнози, так і ознаки зображення для поступового покращення точності моделі в локалізації та асоціації ключових точок.

Після створення карт впевненості та PAFs OpenPose застосовує алгоритм дводольного узгодження для складання виявлених ключових точок у узгоджені пози людей. Цей процес є високоефективним у переповнених середовищах, що робить OpenPose ідеальним для застосувань в аналізі відео та спостереженні, де необхідний вивід у реальному часі.

2.2.2. YOLO модель (*You Only Look Once*)

YOLO є сімейством одноетапних CNN-моделей, відомих своєю здатністю забезпечувати низьку затримку та точне виявлення об'єктів у реальному часі. На відміну від двоетапних детекторів, які розділяють пропозицію об'єкта та класифікацію, YOLO використовує уніфікований підхід, який розглядає виявлення об'єктів як єдину задачу регресії. Це дозволяє моделі прогнозувати обмежувальні прямокутники та ймовірності класів за один прохід через мережу, що робить її ідеальною для застосувань у реальному часі.

Архітектура YOLO значно еволюціонувала. Спочатку моделі ділили вхідне зображення на сітку, де кожна комірка відповідала за прогнозування об'єктів. Основна мережа для вилучення ознак перейшла від простих CNN до

більш просунутих архітектур, таких як CSPDarknet та EfficientNet. Для обробки об'єктів різних розмірів пізніші версії включили багатомасштабні карти ознак, що покращило виявлення об'єктів різних масштабів.

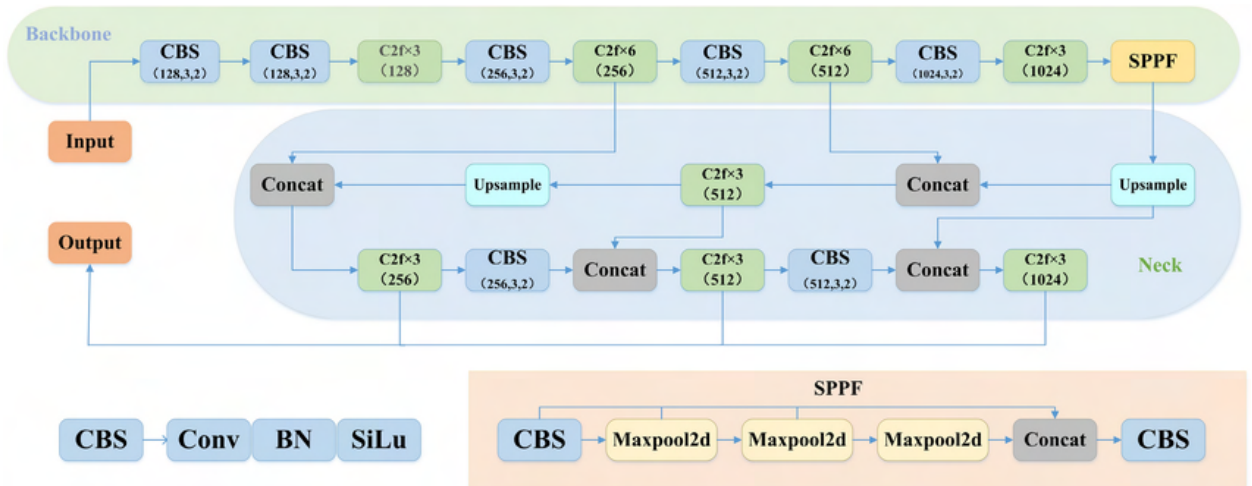


Рис. 2.2. Архітектура мережі YOLOv8

Ключовою інновацією в моделях YOLO є використання анкерних блоків, які надають попередньо визначені форми та розміри обмежувальних прямокутників [32]. YOLO також використовує немаксимальне пригнічення (Non-Maximum Suppression, NMS) для видалення надлишкових виявлень. З розвитком моделі були додані нові функції для покращення продуктивності, такі як Cross-Stage Partial Networks (CSPNet) у YOLOv4, та шари Spatial Pyramid Pooling (SPP) для покращення рецептивного поля. Останні досягнення з YOLOv8, такі як виявлення без анкерів та розділені голови виявлення, спростили архітектуру та підвищили точність.

Хоча спочатку YOLO було розроблено для виявлення об'єктів, новіші версії, починаючи з YOLOv8, тепер підтримують оцінку 2D-пози [30]. Це розширення дозволяє YOLO виявляти та локалізувати ключові точки людини, роблячи його сильною альтернативою спеціалізованим моделям, таким як OpenPose. Однією з ключових переваг YOLO є її уніфікована архітектура, яка дозволяє одночасно виконувати виявлення об'єктів та оцінку пози, ефективно вирішуючи кілька завдань комп'ютерного зору одночасно.

Основне призначення YOLO моделі полягає у виявленні та класифікації об'єктів різних класів (наприклад, люди, автомобілі, тварини) та визначення їхніх обмежувальних рамок (bounding boxes) на зображенні. На відміну від старих методів, які сканують зображення кілька разів, YOLO "дивиться" на зображення лише один раз. Воно розділяє зображення на сітку, і кожна клітинка сітки відповідає за передбачення об'єктів, що знаходяться в ній. Це дозволяє моделі виявляти всі об'єкти одночасно.

Модель повертає список об'єктів, кожен з яких містить клас (наприклад, "людина", "автомобіль"), ймовірність цього класу та координати обмежувальної рамки навколо об'єкта.

Таблиця 2.1.

Відмінності та спільне використання

Характеристика	OpenPose	YOLO
Завдання	Оцінка пози	Виявлення об'єктів
Результат	Координати ключових точок	Обмежувальні рамки
Типові версії	OpenPose	YOLOv8, YOLOv10, YOLOv11

2.2. Середовища симуляції та інструментарії

У цьому розділі ми розглянемо існуючі середовища симуляції та інструменти, які допомагають дослідникам у створенні та зборі даних зображень для розробки спеціальних наборів даних. На відміну від статичних синтетичних наборів даних, як-от SURREAL та Human3.6M, ці інструменти надають динамічні платформи та утиліти, які дозволяють дослідникам генерувати спеціальні набори даних для своїх конкретних випадків використання. Хоча ця гнучкість є значною перевагою, вона часто вимагає додаткових зусиль для розробки. Створення надійних середовищ симуляції є трудомістким процесом, особливо тих, що надають користувачам широкий контроль над параметрами симуляції.

2.2.1. Симулятор CARLA

CARLA (Car Learning to Act) — це симулятор з відкритим кодом, розроблений на базі Unreal Engine 4. Цей ігровий рушій відомий своєю високоякісною графікою та інтерактивними середовищами [20]. Створений для підтримки досліджень у галузі автономного водіння, CARLA пропонує надійну платформу для навчання, прототипування та оцінки автономних систем, зокрема в складних міських умовах.

CARLA — це відкритий симулятор, створений для розробки, навчання та перевірки систем автономного водіння. Він забезпечує реалістичне ігрове середовище з графікою високої якості та точним фізичним моделюванням, що дозволяє дослідникам і інженерам тестувати алгоритми без ризику для реального світу. Дозволяє користувачам легко змінювати параметри симуляції, додавати нові об'єкти, створювати власні сценарії та контролювати поведінку інших учасників дорожнього руху. CARLA підтримує роботу з популярними бібліотеками, такими як TensorFlow та PyTorch, що дозволяє навчати нейронні мережі для сприйняття, планування та керування автомобілем. CARLA дозволяє дослідникам створювати власні сценарії водіння без обмежень ліцензування. Симулятор надає велику бібліотеку цифрових активів з відкритим доступом, яка включає деталізовані міські плани, різноманітні моделі транспортних засобів та реалістичну архітектуру. Його система сенсорів з можливістю налаштування включає RGB-камери, камери глибини, LIDAR, GPS та IMU, а також опції для інтеграції інших типів сенсорів [20]. CARLA також надає широкий контроль над умовами навколишнього середовища, включно зі змінами погоди, варіаціями освітлення та динамічними міськими елементами. Це дозволяє тестувати автономні системи в широкому спектрі сценаріїв, від складних погодних умов до переповнених перехресть, що є критично важливим для оцінки продуктивності та надійності в реальних умовах [20].

Для зручності управління симулятором, CARLA надає бібліотеку клієнта Python, що дозволяє користувачам безпосередньо керувати

транспортними засобами та різними аспектами сценаріїв через зовнішні скрипти або програми. Це спрощує адаптацію симулятора до унікальних випадків використання. Крім того, CARLA містить інструменти для бенчмаркінгу, що дозволяють оцінювати продуктивність різних рішень для автономного водіння, надаючи стандартизовані метрики для послідовної та відтворюваної оцінки [20].

Як симулятор, спеціально призначений для автономного водіння, CARLA оптимізований для задоволення потреб цієї галузі та менш придатний для інших застосувань, таких як оцінка пози або аналіз поведінки людини.

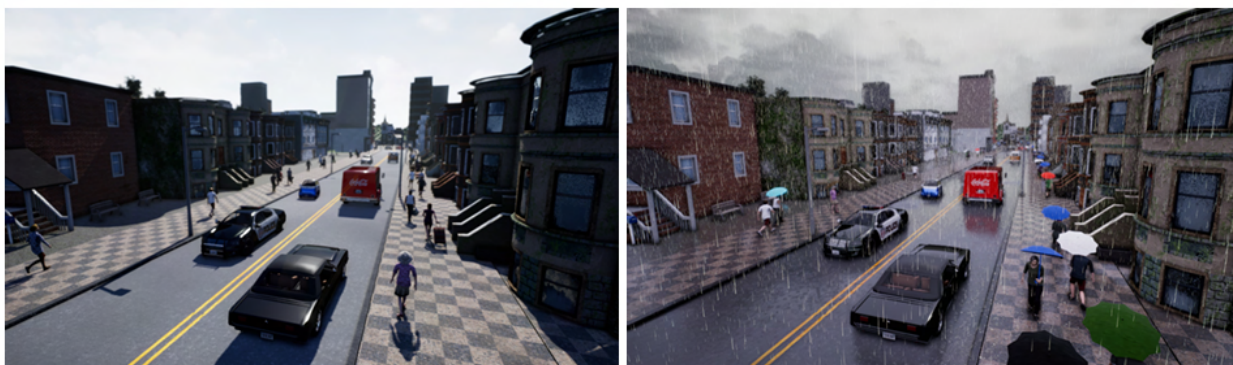


Рис. 2.3. Приклад середовища симулятора CARLA

Проте CARLA є яскравим прикладом того, як добре розроблене, спеціалізоване середовище симуляції може значно сприяти прогресу в певній галузі, надаючи дослідникам потужний інструмент для ретельного тестування та розробки. Його широке впровадження підкреслює зростаючий попит на спеціалізовані середовища симуляції, що дозволяють дослідникам генерувати власні набори даних та експериментувати в контрольованому, реалістичному середовищі.

2.2.2. Інструментарій *UnrealCV*

UnrealCV — це відкритий плагін, який розширює можливості *Unreal Engine* для збору зображень з метою досліджень комп'ютерного зору [34].

Використовуючи високоякісну графіку та фізичне моделювання Unreal Engine, UnrealCV дозволяє дослідникам захоплювати основні формати даних, такі як RGB-зображення, карти глибини та маски об'єктів, що дає змогу створювати високоякісні синтетичні набори даних. Цей інструмент забезпечує широкий контроль над параметрами сцени, включаючи кути камери, освітлення та розташування об'єктів, що робить його високоадаптованим для генерації наборів даних, спеціально призначених для конкретних дослідних потреб.

Завдяки наданню Python API, UnrealCV може безперешкодно інтегруватися з процесами обробки даних та машинного навчання, дозволяючи автоматизувати завдання, такі як маніпуляція сценами та захоплення даних. Ця функція спрощує генерацію наборів даних, зменшуючи потребу в ручному втручанні та підвищуючи ефективність. UnrealCV став важливим інструментом для дослідників комп'ютерного зору, сприяючи створенню високоякісних, спеціалізованих наборів даних, які було б складно або дорого отримати в реальних умовах.

На відміну від CARLA, яка надає повноцінне середовище симуляції для досліджень автономного водіння, UnrealCV зосереджується на наданні інструментів для захоплення зображень у будь-якому проєкті Unreal Engine. Це означає, що UnrealCV не має вбудованих елементів симуляції, таких як транспортні системи, поведінка пішоходів або динаміка навколишнього середовища, які пропонує CARLA.

Отже, дослідники, які використовують інструмент UnrealCV, повинні мати робочі знання Unreal Engine, щоб проектувати та створювати власні сцени, а також налаштовувати необхідні об'єкти та анімації в середовищі. Ця вимога може бути бар'єром для тих, хто не має попереднього досвіду роботи з Unreal Engine, потенційно обмежуючи доступність плагіна для користувачів, які шукають просте, готове до використання середовище симуляції.

2.2.3. *Набір інструментів UnrealCV*

UnrealCV — це плагін із відкритим вихідним кодом, який розширює можливості Unreal Engine для полегшення збору даних зображень для досліджень у галузі комп'ютерного зору. Використовуючи високоякісну графіку та фізичну симуляцію Unreal Engine, UnrealCV дозволяє дослідникам отримувати такі основні формати даних, як RGB-зображення, карти глибини та маски об'єктів, що дає змогу створювати високоякісні синтетичні набори даних. Цей інструмент надає широкий контроль над параметрами сцени, включаючи ракурси камери, освітлення та розташування об'єктів, що робить його дуже адаптивним для створення наборів даних, спеціально розроблених для конкретних дослідницьких потреб.

Завдяки API Python, UnrealCV може бути легко інтегрований у робочі процеси обробки даних та машинного навчання, що дозволяє автоматизувати такі завдання, як маніпуляція сценою та захоплення даних. Ця функція оптимізує генерацію наборів даних, зменшуючи потребу в ручному втручанні та підвищуючи ефективність. UnrealCV став чудовим інструментом для дослідників у галузі комп'ютерного зору, оскільки він спрощує створення високоякісних, специфічних для певної галузі наборів даних, які було б складно або дорого отримати в реальних умовах.

На відміну від CARLA, що надає повноцінне середовище симуляції для досліджень автономного водіння, UnrealCV зосереджується насамперед на інструментах для захоплення даних зображень у будь-якому проекті Unreal Engine. Це означає, що UnrealCV не має вбудованих елементів симуляції, таких як системи дорожнього руху, поведінка пішоходів або динаміка навколишнього середовища, які пропонує CARLA. Відповідно, дослідники, які використовують UnrealCV, повинні мати робочі знання Unreal Engine, щоб проектувати та створювати власні сцени, а також налаштовувати необхідні об'єкти та анімацію в середовищі. Ця вимога може бути бар'єром для тих, хто не має попереднього досвіду роботи з Unreal Engine, що

потенційно обмежує доступність плагіна для користувачів, які шукають просте, готове до використання середовище симуляції.

2.3. Аналіз та порівняння пропонованої системи генерації даних про оцінку позиціювань об'єктів з існуючими рішеннями

Розробка програмної платформи значною мірою базується на принципах існуючих платформ з відкритим вихідним кодом, а саме UnrealCV та симулятора CARLA. Подібно до цих проєктів, система реалізовано як плагін, що містить основні інструменти та зразки віртуальних середовищ, призначених для оцінки систем комп'ютерного зору та автономного керування в нетипових умовах. Однак, пропонована система вирізняється на тлі зазначених рішень як за рівнем складності, так і за своїм функціональним призначенням.

2.3.1. Порівняння з платформою CARLA

На відміну від CARLA, що є комплексною платформою для збору даних і тестування систем автономного керування, орієнтованою переважно на самокеровані транспортні засоби, пропонована система сфокусована на аналізі поведінки людини. Ця платформа надає необхідні компоненти для створення спеціалізованих наборів даних, що відповідають цій меті. Хоча CARLA підтримує базові анімації персонажів, достатні для моделювання навігації пішоходів, дана система розширює ці можливості, забезпечуючи підтримку широкого спектру сценаріїв (як у приміщенні, так і на відкритому повітрі) та реалізацію складніших рухів тіла.

2.3.2. Порівняння з платформою UnrealCV

На відміну від UnrealCV, яка забезпечує мінімальне налаштування для захоплення зображень з камери та анотації 3D-об'єктів, дана система інтегрує розширені моделі людських персонажів з функціоналом анотації

пози, що значно спрощує збір деталізованих даних про пози людини. Крім того, запропоноване рішення вдосконалює можливості UnrealCV, надаючи неблокуюче захоплення зображень та анотацію даних за допомогою механізмів багатопотоковості в UE5. Це також дозволяє транслювати потоки з віртуальної камери симулятора як джерело вебкамери. Ключові результати, отримані в ході розробки, проілюстровані на рис. 2.4.

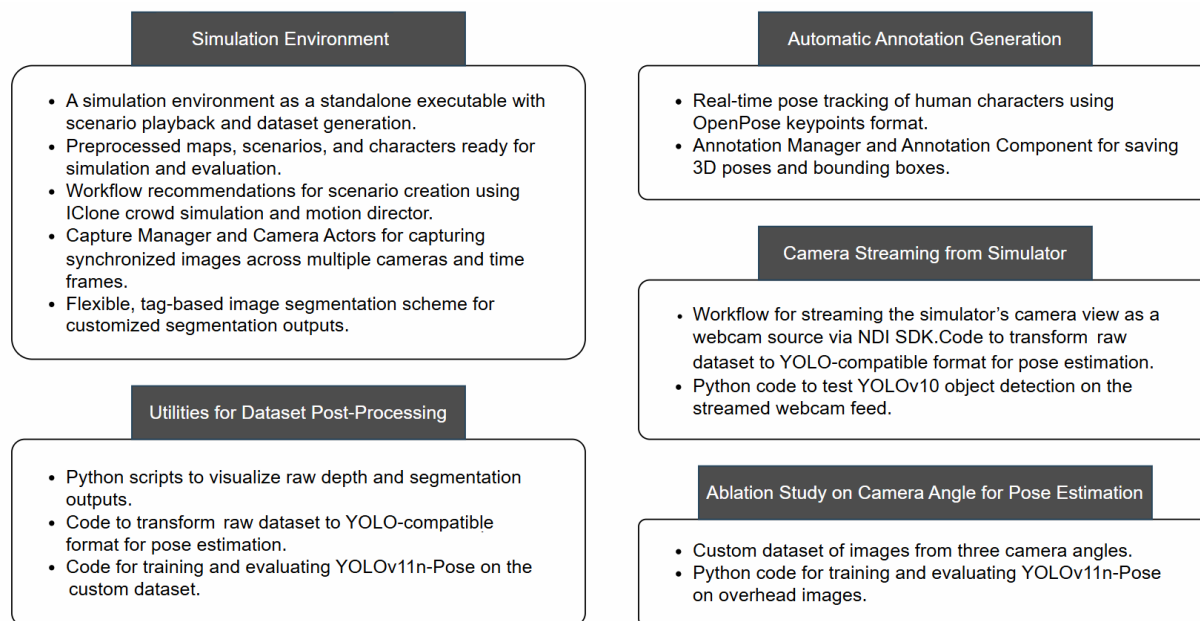


Рис. 2.4. Ключові елементи запропонованого рішення

2.4. Реалізація дизайну та архітектури проєкту

Проєкт містить програмний код, написаний мовою C++, який розширює функціональність стандартних компонентів рушія Unreal Engine 5 (UE5). Основні модифіковані компоненти включають GameMode, Pawn, SkeletalMeshActor, CameraComponent та LevelSequenceActor. Детальний опис цих компонентів доступний у таблиці 2.1. Зазначені компоненти забезпечують відтворення індивідуальних сценаріїв, керування віртуальною камерою та анотацію даних. Крім того, набір зовнішніх скриптів на Python призначений для постобробки та трансформації даних з метою створення

наборів даних, сумісних із широко використовуваними моделями комп'ютерного зору.

Таблиця 2.1.

Основні компоненти пропонованої системи

Термін	Визначення
GameMode	Визначає правила гри, її хід та взаємодію гравця у середовищі Unreal Engine.
SkeletalMeshActor	Актор зі скелетною структурою для анімації, що зазвичай використовується для персонажів.
SceneCapture2D	Захоплює 2D-зображення сцени, часто використовується для ефектів камери та віддзеркалень.
PostProcessMaterial	Матеріал, що застосовується до відрендереної сцени для створення таких ефектів, як корекція кольору та розмиття.
PixelNormalWS	Вузол, що надає нормаль світового простору для кожного пікселя, використовується в освітленні та затіненні.
CustomStencil	Присвоює унікальні ідентифікатори об'єктам для вибіркового маскування в процесі постобробки.
LevelSequence	Ресурс для управління анімаціями, подіями та рухами камери в часі.
Level	Контейнер для елементів сцени, таких як актори, освітлення та геометрія, що визначають середовища.
Lumen	Система глобального освітлення в реальному часі в Unreal Engine 5, що забезпечує динамічне освітлення без попереднього запікання.
Ray Tracing	Техніка рендерингу, що імітує реалістичне освітлення, покращуючи тіні та віддзеркалення.

Загальну архітектуру системи зображено на рисунку 2.5, а докладні UML-діаграми класів кожного компонента наведено на рисунках в наступному підрозділі.

Rawp - компонент контролює введення користувача та віртуальну камеру у вікні перегляду. За допомогою компонентів плагіна NDI SDK, зокрема NDI Viewport Capture та NDI Stream Broadcast, система може транслювати зображення з камери як віртуальну вебкамеру. Це дозволяє

інтегрувати її із зовнішніми інструментами для аналізу в режимі реального часу та верифікації моделей.

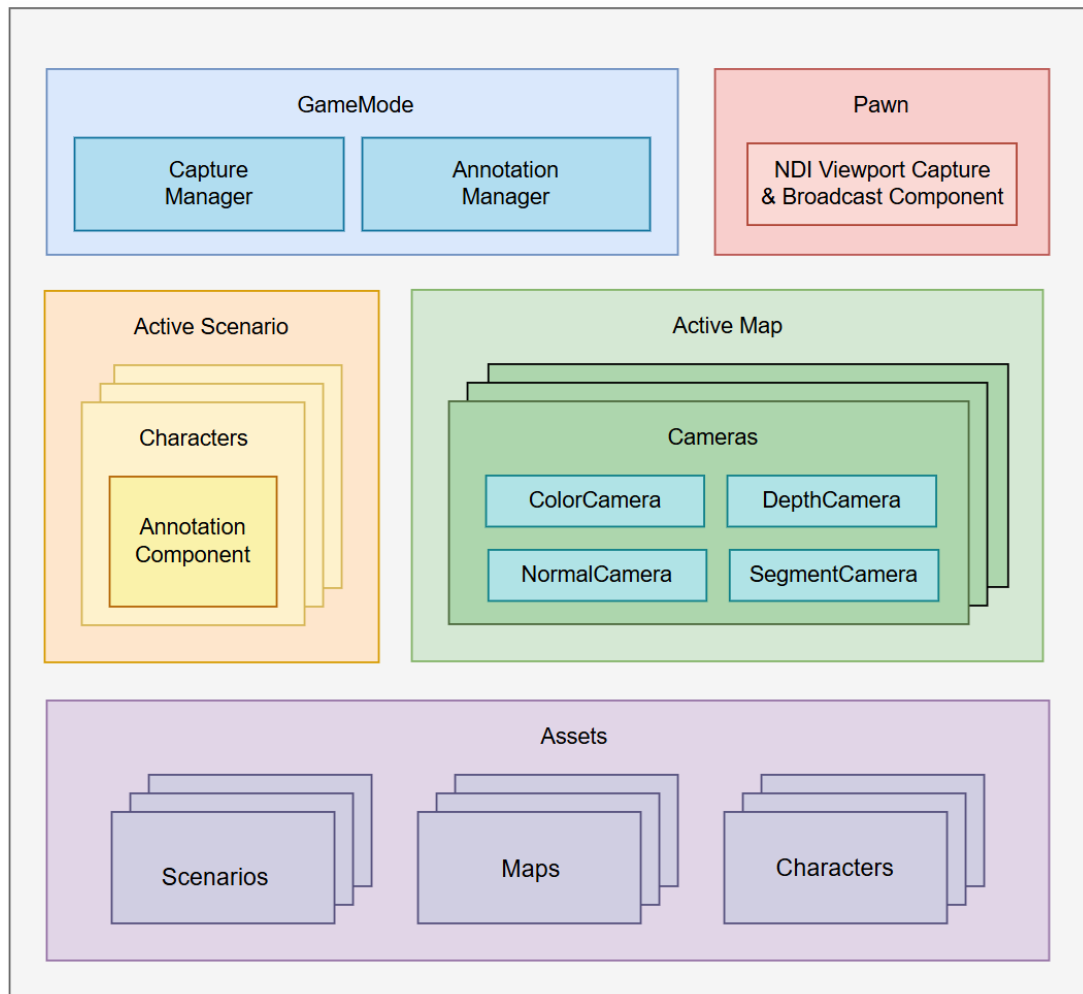


Рис. 2.5. Загальна архітектура системи

Active Map - компонент є активним ресурсом карти, завантаженим симулятором. Кожна карта містить унікальні налаштування, такі як візуальний контент, ландшафт, статичні об'єкти та освітлення. Камери з'являються на карті під час її завантаження та містять спеціалізовані компоненти (Color, Depth, Normal та Segmentation) для захоплення різних типів даних, необхідних для аналізу.

Active Scenario - компонент представляє активний сценарій, що відтворюється симулятором. Сценарій є набором анімацій, що виконуються усіма персонажами. Кожен персонаж обладнаний компонентом Annotation

Component, який автоматично захоплює дані 3D-пози та обмежувальної рамки. Це значно спрощує процес збору та обробки даних компонентом Annotation Manager.

2.4.1. Побудова діаграм класів компонент системи

GameMode - цей компонент відповідає за управління процесами захоплення та анотації даних. Він містить Capture Manager, який синхронізує захоплення зображень із кількох камер, забезпечуючи послідовність кадрів, та Annotation Manager, що збирає анотації поз і обмежувальних рамок. Ці підсистеми автоматизують процес генерації зображень і відповідних анотацій.

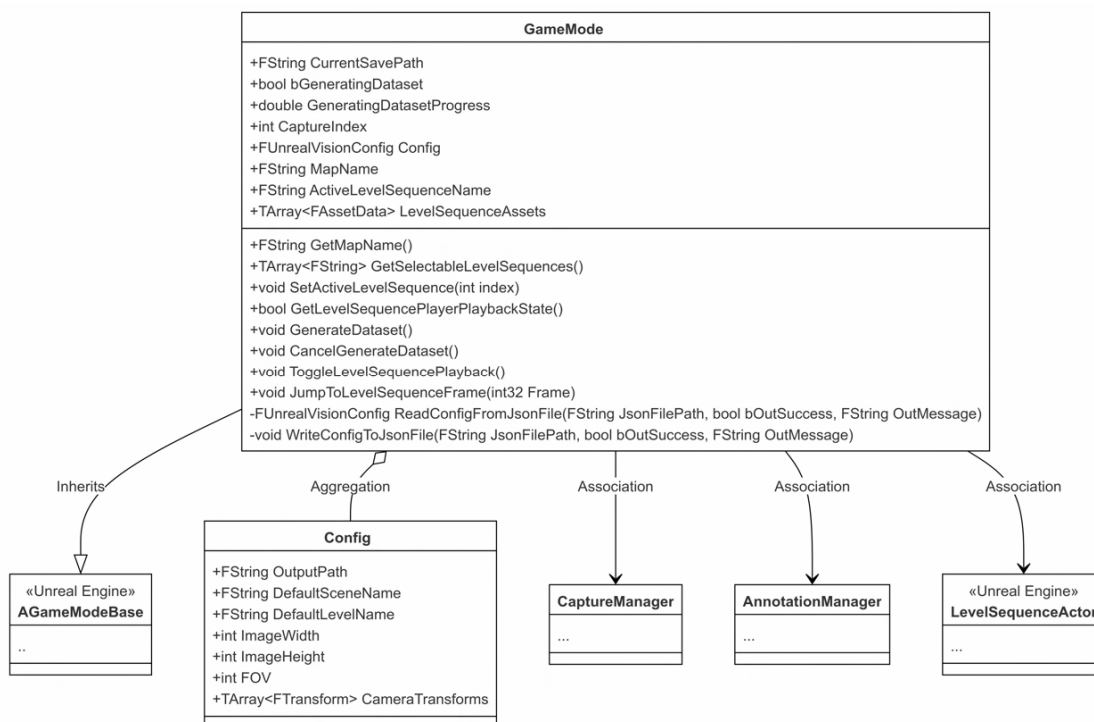


Рис. 2.6. Діаграма класів GameMode

Assets - репозиторій попередньо визначених ресурсів, включаючи карти, сценарії та моделі персонажів, які можуть бути завантажені та модифіковані. карти визначають налаштування середовища, сценарії — анімації та поведінку, а моделі персонажів пропонують різноманітний

зовнішній вигляд (форма тіла, стать, вік). Ці ресурси легко модифікувати в Unreal Engine, що забезпечує гнучкість налаштування та швидкість розгортання нових симуляцій.

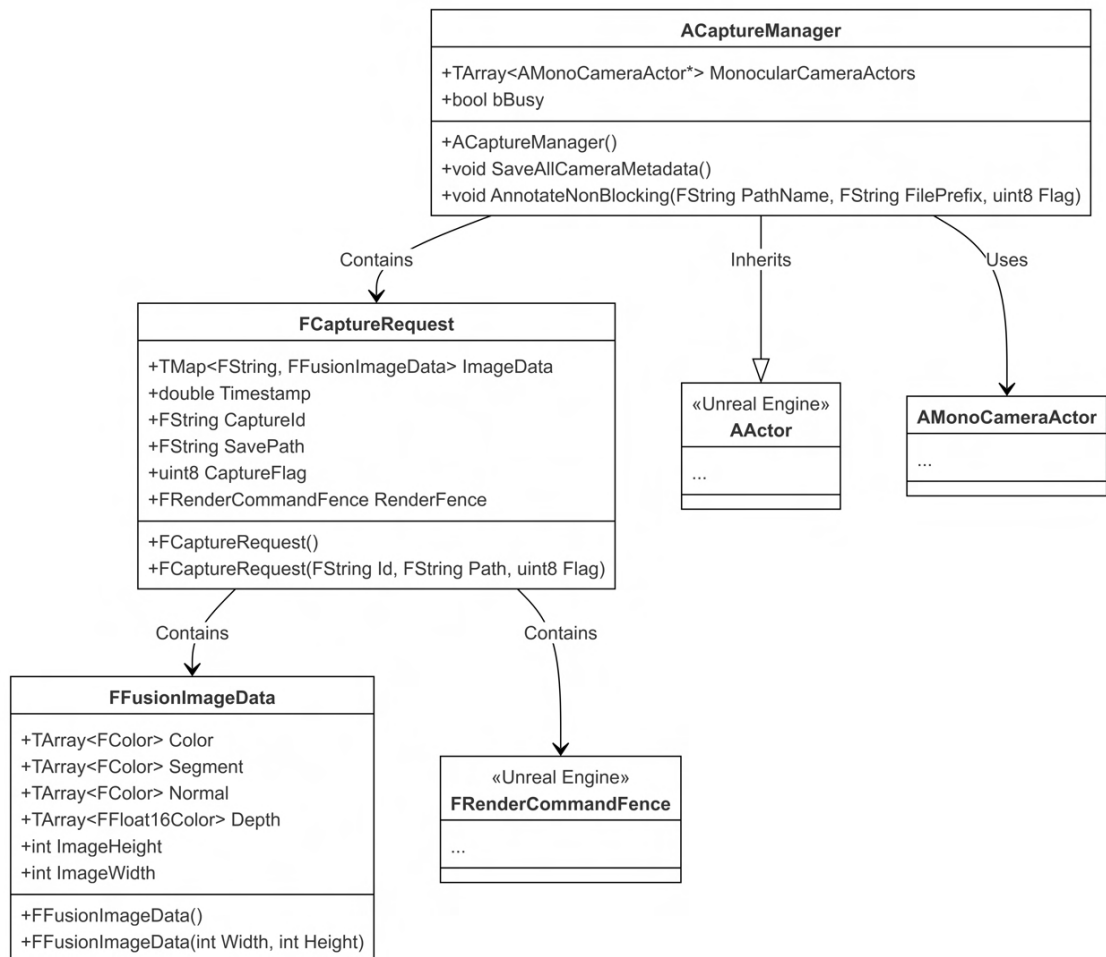


Рис. 2.7. Діаграма класів CaptureManager

Розглянемо основні компоненти та їхні зв'язки на рисунку 2.7.

ACaptureManager - це головний клас-контролер, який успадковує властивості від **AActor** з Unreal Engine. Він відповідає за управління процесом захоплення.

Він використовує клас **AMonoCameraActor**, який представляє монокулярну камеру в симуляційному середовищі. Він містить масив (**TArray**) об'єктів **AMonoCameraActor**, що вказує на можливість керувати кількома камерами одночасно.

Клас має методи для збереження метаданих камери (SaveAllCameraMetadata) та асинхронного анотування (AnnotateNonBlocking). Його поле bBusy (логічний тип bool) вказує, чи триває процес захоплення.

FCaptureRequest - клас представляє запит на захоплення даних. Він містить дані зображення (FfusionImageData), ідентифікатор захоплення (CaptureId), шлях збереження (SavePath) та інші параметри.

Цей клас містить екземпляр класу FRenderCommandFence, який використовується для синхронізації команд рендерингу, забезпечуючи, що дані будуть збережені лише після завершення рендерингу.

FFusionImageData: Цей клас інкапсулює дані, отримані з камери, такі як кольорові зображення (Color), сегментація (Segment), нормалі (Normal) та глибина (Depth).

Дані зберігаються у вигляді масивів (TArray). Клас також містить інформацію про розміри зображення (ImageHeight та ImageWidth).

FRenderCommandFence - цей клас, є частиною Unreal Engine, забезпечує синхронізацію між потоком рендерингу та основним потоком гри. Завдяки цьому, дані зображення можуть бути оброблені лише після того, як вони повністю відрендерені, що запобігає помилкам доступу до неповних даних.

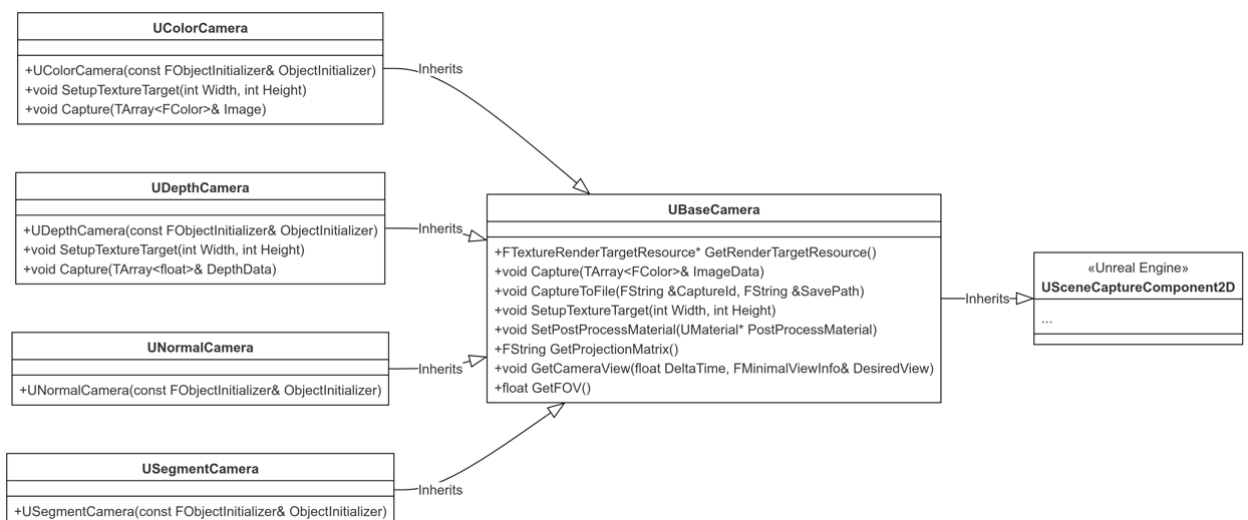


Рис. 2.8. Діаграма класів Camera Components

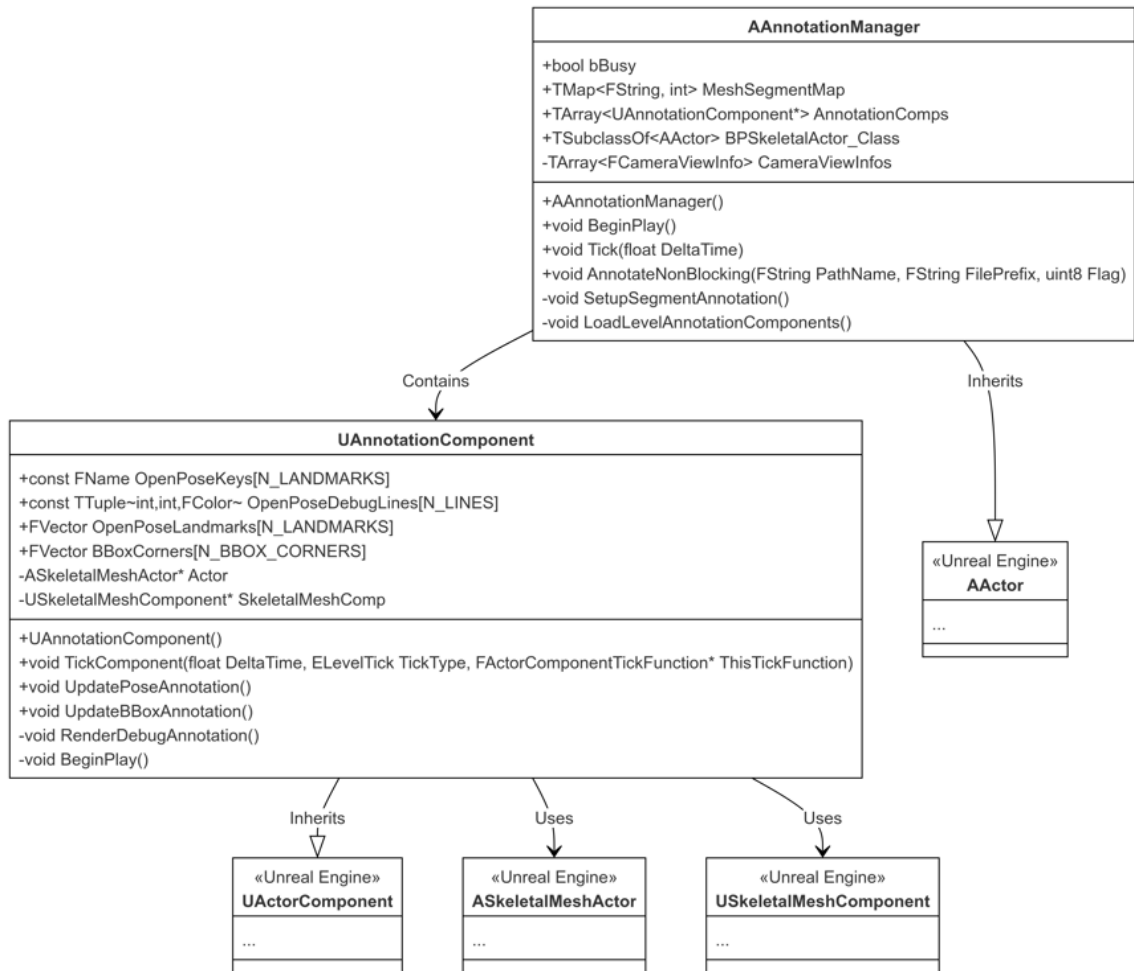


Рис. 2.19. Діаграма класів AnnotationManager

Крім зазначених основних компонентів, система також містить допоміжні утиліти для ефективної роботи з даними. Ці інструменти забезпечують асинхронний запис файлів та зображень на диск, що значно підвищує загальну продуктивність під час генерації даних.

Крім того, система пропонує функціонал для перетворення захоплених зображень та анотацій у масиви NumPy. Це полегшує їхню подальшу інтеграцію в робочі процеси машинного навчання. Код для конвертації в NumPy адаптовано з реалізації UnrealCV, що дозволяє зберігати матричні дані з Unreal Engine у форматі, зручному для постобробки в Python. Приклади симуляційного середовища наведено на рис. 2.10.



Рис. 2.10. Приклад симуляції середовища

На початку сценарію CaptureManager знаходить усі актори на мапі та призначає відповідне значення трафарету (stencil value) на основі їхніх тегів. У результаті, вихідне зображення сегментації є RGB-зображенням, у якому значення трафарету закодовано в червоному каналі.

Блок-схема (рис. 2.11) описує процес налаштування семантичної сегментації в середовищі в Unreal Engine, використовуючи так званий Stencil Buffer. Мета — ідентифікувати та призначити унікальні значення (ID) для візуалізації різних об'єктів на сцені, що дозволяє потім генерувати маски сегментації.

Етапи процесу

1. Ініціалізація та пошук акторів: Процес починається з ініціалізації лічильника StencilValue зі значенням 1. Потім система шукає всі об'єкти (актори), які мають тег "segment".

2. Перевірка наявності: Якщо актори з таким тегом не знайдені, процес завершується з відповідним повідомленням.

3. Обробка акторів: Якщо актори знайдені, система починає їх обробку по черзі. Для кожного актора:

- Вона отримує перший тег актора, який буде використовуватися як унікальний ідентифікатор (ActorTag).

- Потім вона знаходить всі компоненти типу UMeshComponent (тобто, всі видимі сітки) всередині цього актора.

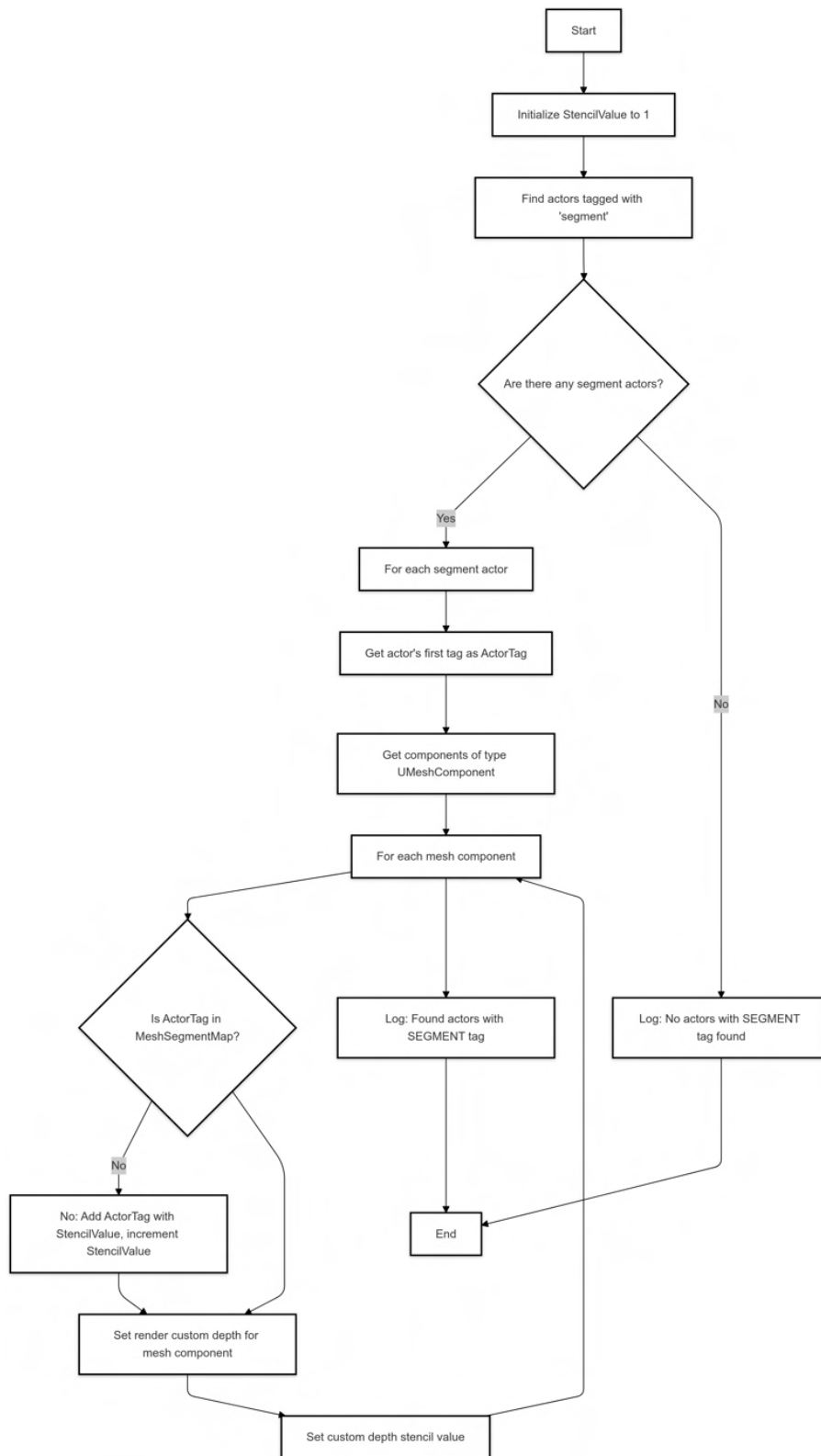


Рис. 2.11. Блок-схема налаштування анотації сегментації

3. Призначення ідентифікатора: Для кожного компонента-сітки:

- Система перевіряє, чи є вже ActorTag у мапі MeshSegmentMap. Ця мапа, зберігає відповідність між ActorTag і вже призначеними StencilValue.

- Якщо тегу немає: Система додає ActorTag у мапу, призначаючи йому поточне значення StencilValue, а потім збільшує StencilValue на 1.

- Якщо тег вже є: Процес продовжується без змін StencilValue, оскільки цьому об'єкту вже призначено ідентифікатор.

5. Налаштування рендерингу: Після визначення StencilValue для сітки, процес налаштовує параметри рендерингу для цього компонента:

- Встановлює custom depth (спеціальний буфер глибини) для компонента.

- Встановлює custom depth stencil value (значення в буфері Stencil), використовуючи визначений раніше ідентифікатор. Це значення і буде використовуватися для створення масок сегментації.

Таким чином, блок-схема демонструє автоматизований процес присвоєння унікальних ID візуальним об'єктам на основі їхніх тегів. Це дозволяє рендеринг-системи, наприклад, Unreal Engine, створювати маски, де кожен об'єкт виділений своїм унікальним кольором або значенням, що є основою для семантичної сегментації.

Висновки до розділу

У другому розділі досліджено сучасні підходи до оцінки пози та виявлення об'єктів, серед яких особливу увагу приділено моделям OpenPose і YOLO. Показано, що OpenPose забезпечує високоточну багатоточкову оцінку скелетних моделей людини, тоді як YOLO характеризується швидкістю та ефективністю у завданнях детекції. Аналіз інструментаріїв CARLA та UnrealCV підтвердив їхню придатність для симуляцій, проте виявив і обмеження у гнучкості сценаріїв та масштабованості. Розроблена модель архітектури системи створює основу для подальшого розширення функціональності й підвищення якості згенерованих даних.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ МОДЕЛІ ГЕНЕРАЦІЇ ДАНИХ ПРО ОЦІНКУ ПОЗИЦІЮВАНЬ ОБ'ЄКТІВ ТА АНАЛІЗУ ЇХ ПОВЕДІНКИ

3.1. Реалізація основних компонент системи

Пропоноване середовище структуровано на основі трьох ключових ресурсів: карти (map), сценарію (scenario) та персонажа (character).

3.1.1 *Карти*

Карта є віртуальним середовищем (LevelMap в термінології UE5), що включає декорації, площини, фонові об'єкти та освітлення. Ці середовища, створені в UE5, використовують передові технології рендерингу, такі як трасування променів (ray tracing) та Lumen, для забезпечення високоякісної візуалізації. Наразі пропонується дві окремі карти: BasicMap (в приміщенні) та Downtown (на відкритому повітрі). Ці карти, адаптовані з Unreal Engine Marketplace, були оптимізовані для зниження споживання пам'яті та прискорення завантаження. Використання обох карт дозволяє моделювати поведінку людини в різноманітних умовах. На рис. 3.1 представлено типове налаштування карти з попередньо розміщеними персонажами та камерами.

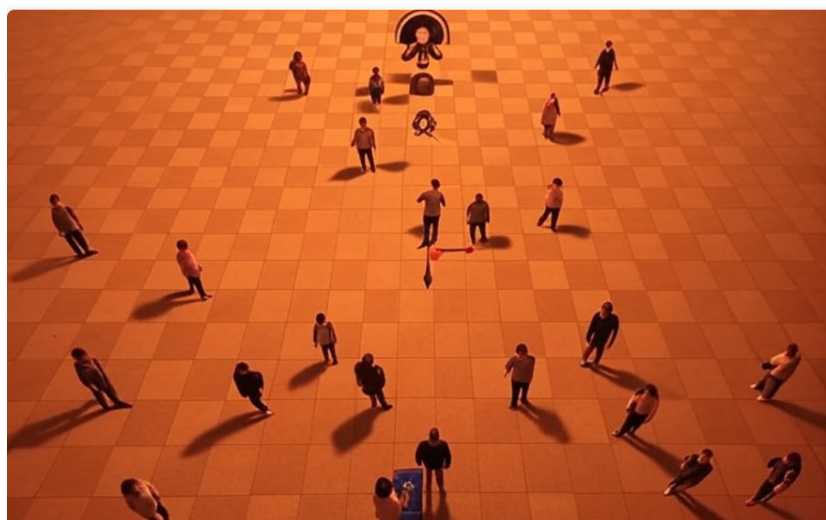


Рис. 3.1. Налаштування карти з попередньо розміщеними персонажами та камерами

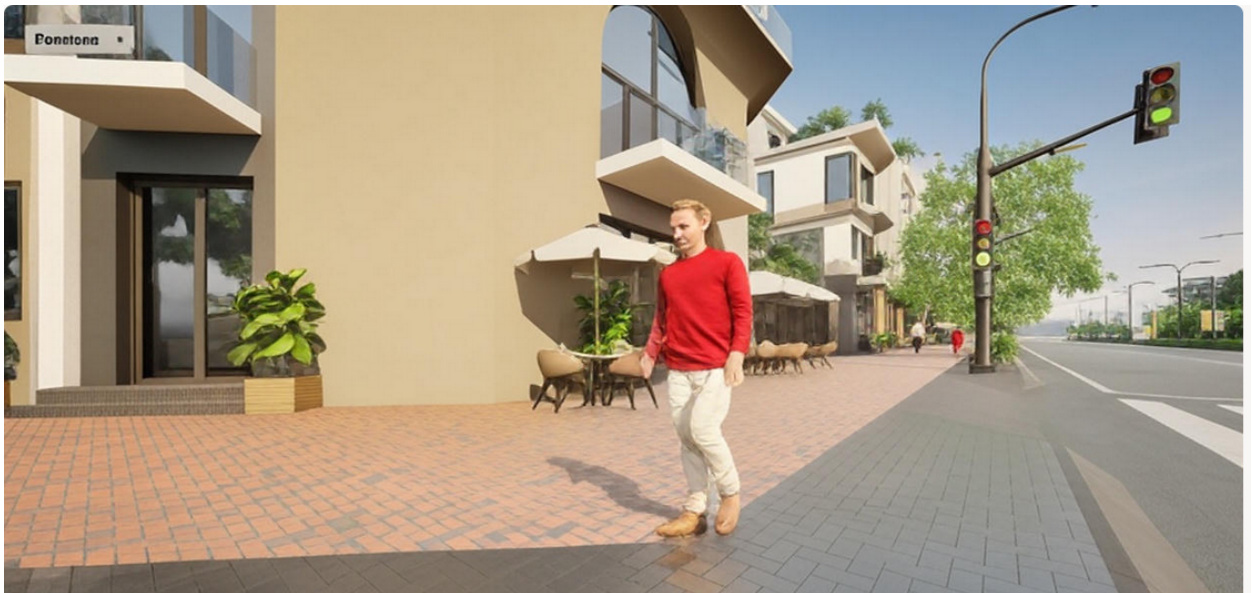


Рис. 3.2. Карта «Downtown» зі складним міським ландшафтом

3.1.2. Сценарії

Сценарій є колекцією анімацій для кількох персонажів, що відтворюються одночасно. Ця концепція відповідає LevelSequence та Subsequence в UE5. Кожна карта може містити необмежену кількість сценаріїв, хоча для створення наборів даних одночасно може відтворюватися лише один.



Рис. 3.3. Приклад симуляції натовпу

Реквізит — об'єкти, з якими взаємодіють персонажі — вважається частиною сценарію і завантажується разом з ними. До такого реквізиту належать мобільні телефони, книги, чашки, рюкзаки, милиці та інвалідні візки. Ці об'єкти можна анотувати разом з персонажами, що дозволяє реалізувати сегментацію зображень на основі контексту, де персонаж і пов'язаний з ним об'єкт розглядаються як єдине ціле. Користувачі можуть приховувати або відображати окремі групи персонажів, що полегшує доповнення даних на рівні сценарію. Приклади симуляцій натовпу в середовищі наведено на рис. 3.3.

3.1.3. Персонажі

Персонаж є моделлю (SkeletalMeshActor в UE5), попередньо налаштованою для захоплення даних пози у форматі COCO OpenPose. Замість стандартних Metahuman моделей UE5, в системі використовуються моделі з ActorCore. Це забезпечує розширену підтримку анімації та оптимізоване використання пам'яті, що дозволяє створювати симуляції з великою кількістю персонажів. Високоякісна модель персонажа займає близько 250 МБ, що в 5–10 разів менше, ніж у Metahuman. Набір персонажів налічує 20 різних моделей з різними типами тіла (худі, середні, міцні), статтю та віковими групами (дитина, дорослий, літня людина).

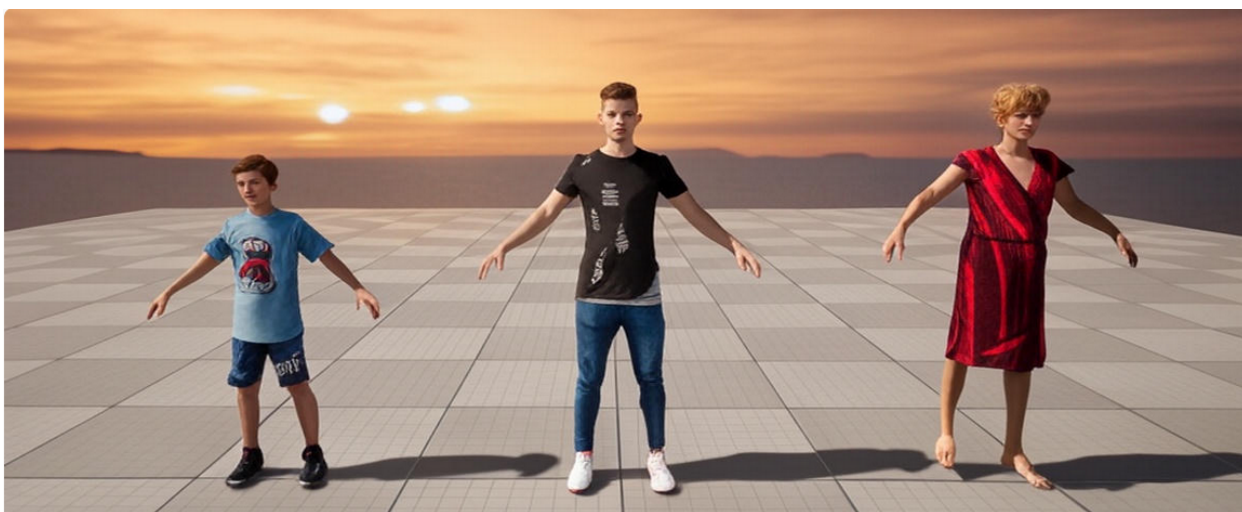


Рис. 3.4. Персонажі з різними віковими профілями та формами тіла

Ці анімації додатково адаптовані до індивідуальних характеристик кожного персонажа. На рис. 3.4 представлено зразки персонажів з різними віковими профілями та формами тіла.

3.2. Процес розробки сценаріїв

Сценарій поєднує декілька анімацій персонажів для формування деталізованих і динамічних сцен, що особливо ефективно для симуляції натовпу. Ця робота зосереджується на генерації даних анотацій з цих анімацій, а не на самих технічних аспектах анімації, оскільки анімація є фундаментальним елементом 3D-моделювання. Для оптимізації процесу створення анімації сценарії спочатку розробляються в програмному середовищі iClone, а потім експортуються в UE5 як ресурси LevelSequence для подальшої обробки. iClone є професійним інструментом для анімації, відомим своїм інтуїтивно зрозумілим інтерфейсом та розширеними функціями, включаючи Motion Director та Crowd Simulation. Ці можливості суттєво скорочують час та зусилля, необхідні для створення складних сцен з великою кількістю акторів.

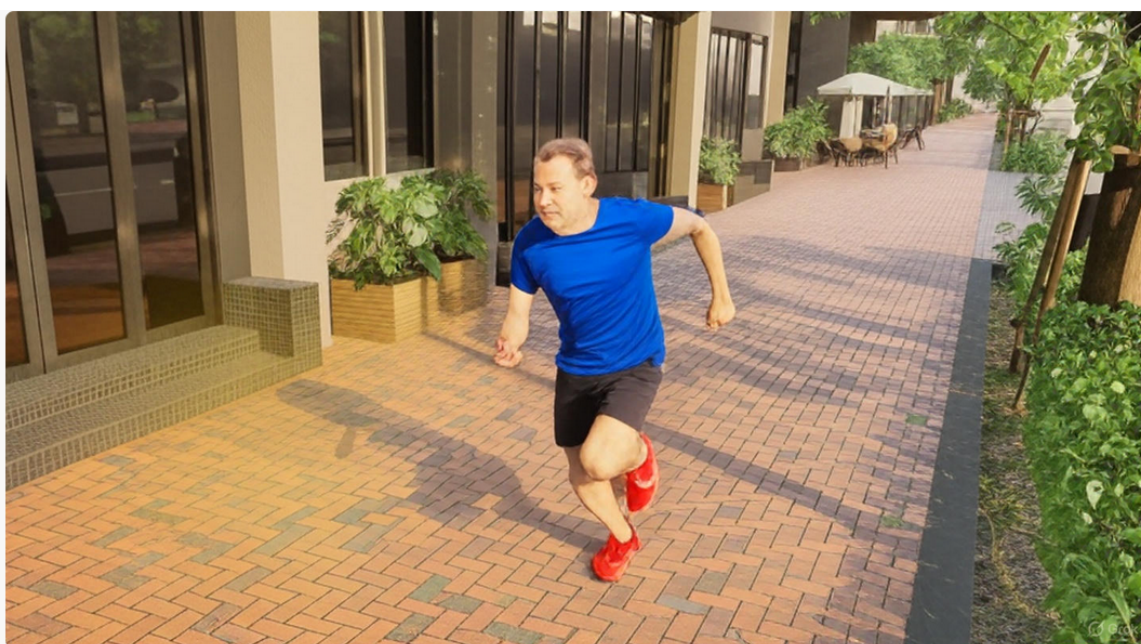


Рис. 3.5. Приклад анімації персонажа

На рис. 3.5 представлено приклад анімації персонажа, створеної за допомогою Motion Director, а на рис. 3.6 — відеоматеріал з робочого процесу симуляції натовпу в iClone.



Рис. 3.6. Спрощена версія карти «Downtown» для симуляції натовпу та запису анімації

3.2.1. Підготовка та оптимізація

Перед створенням сценаріїв в iClone генерується спрощена версія карти UE5 шляхом об'єднання всіх об'єктів в єдину сітку з подальшим перенесенням до iClone. Це гарантує, що сценарії, створені в iClone (часто з використанням симуляції натовпу), узгоджуються з розміщенням об'єктів на вихідній карті, дозволяючи персонажам уникати перешкод під час руху та взаємодії. Функція Motion Director в iClone забезпечує інтуїтивне управління гуманоїдними моделями, тоді як Crowd Simulation дозволяє одночасно анімувати велику кількість персонажів з автоматичним уникненням перешкод. Крім того, програмне забезпечення дає змогу рандомізувати анімації верхньої частини тіла, зберігаючи при цьому послідовні рухи ходьби, що додає різноманітності до дій персонажів. На рис. 3.7 наведено

додаткові приклади анімацій, створених в iClone та інтегрованих у симуляційне середовище.



Рис. 3.7. Додаткові сценарії

3.2.2. Класифікація анімацій

Оскільки запропоноване рішення розроблено насамперед для генерації спеціалізованих наборів даних для оцінки пози, критично важливим є створення різноманітних сценаріїв, що моделюють складну поведінку людей у повсякденному житті. Для досягнення цієї мети анімації персонажів класифіковано на три групи:

- автономна анімація;
- взаємодія людини з людиною;
- взаємодія людини з об'єктом.

В даній роботі ми прагнемо розробити сценарії, які комплексно охоплюють кожен з цих категорій.

3.3. Представлення робочого процесу захоплення зображень

Однією з ключових функцій системи як генератора наборів даних є його здатність захоплювати різні типи зображень, зокрема кольорові, глибини, карти нормалей та зображення сегментації. Розміщуючи кілька камер у різних точках карти, ми можемо отримувати ці типи даних для одного й того ж сценарію з різних ракурсів.

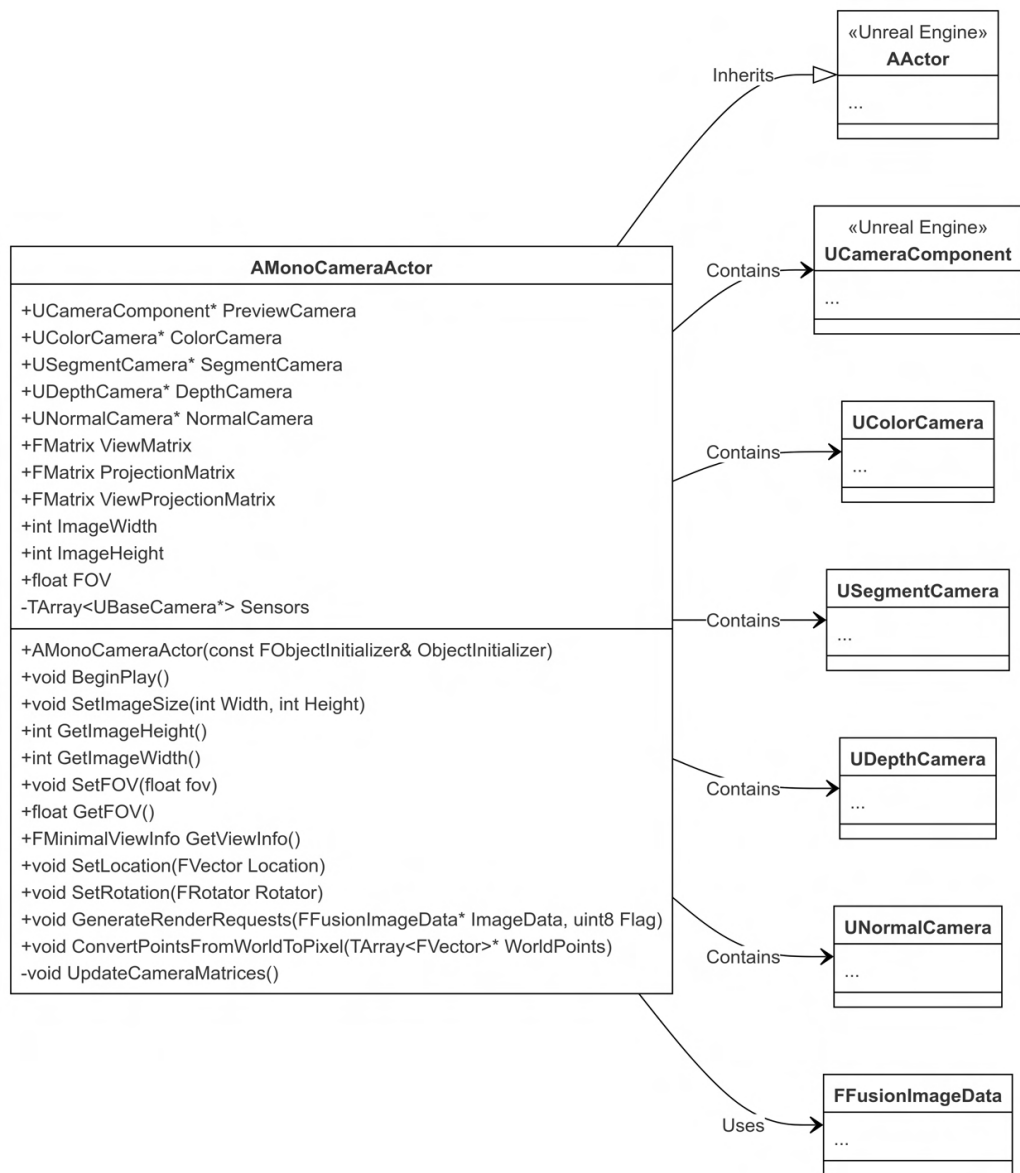


Рис. 3.8. Діаграма класів MonoCameraActor

Для реалізації цієї можливості було розроблено клас MonoCameraActor, який містить спеціалізовані компоненти камери для кожного типу захоплення: ColorCamera, DepthCamera, NormalCamera та SegmentCamera. Компонент MonoCameraActor може бути розташований у будь-якій точці карти. Кожна карта постачається з камерами за замовчуванням, розміщеними в рекомендованих точках, але користувач може змінювати їхнє розташування, редагуючи конфігураційний JSON-файл перед запуском програми.

Компоненти камери є розширеннями компонента SceneCapture2D з UE5 (як показано на рис. 3.9). Вони відтворюють вихідне зображення камери на цільову текстуру, що базується на принципах моделі камери-обскури. Ця модель є спрощеним уявленням того, як камера проєктує 3D-сцену на 2D-площину. У цій моделі промені світла від сцени проходять через єдину точку (отвір) і формують зображення на протилежній стороні.

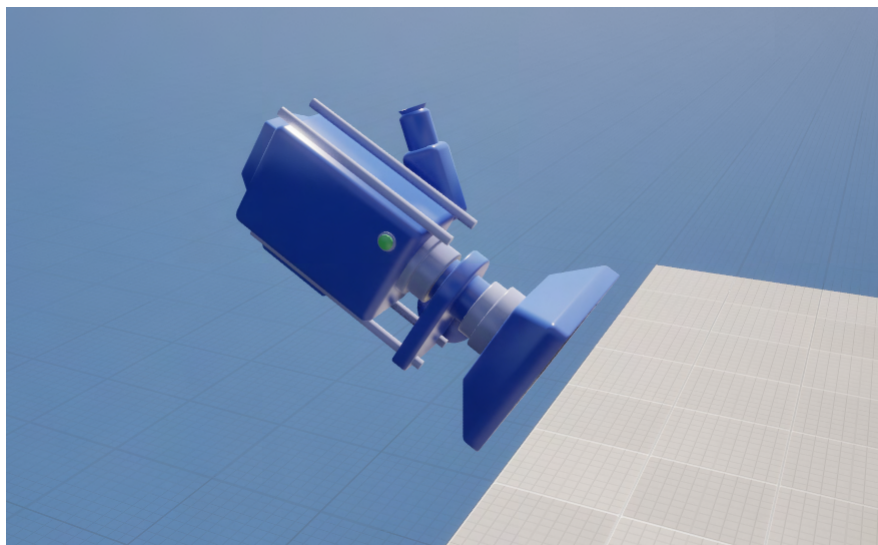


Рис. 3.9. MonoCameraActor

3.3.1. Налаштування камери та управління даними

Для створення MonoCameraActor необхідно надати певні параметри, зокрема висоту та ширину зображення, а також кут огляду (FOV), які використовуються для побудови внутрішньої матриці (intrinsic matrix). Додатково потрібні позиція та обертання камери для визначення зовнішньої матриці (extrinsic matrix).

Під час генерації набору даних система записує та зберігає всі метадані про камери в поточному сценарії у файлі cameras.csv. Ці метадані містять важливі деталі конфігурації кожної камери, які можуть бути використані разом з іншими згенерованими анотаціями (наприклад, 3D-пози та 3D-обмежувальні рамки) для постобробки та аналізу.

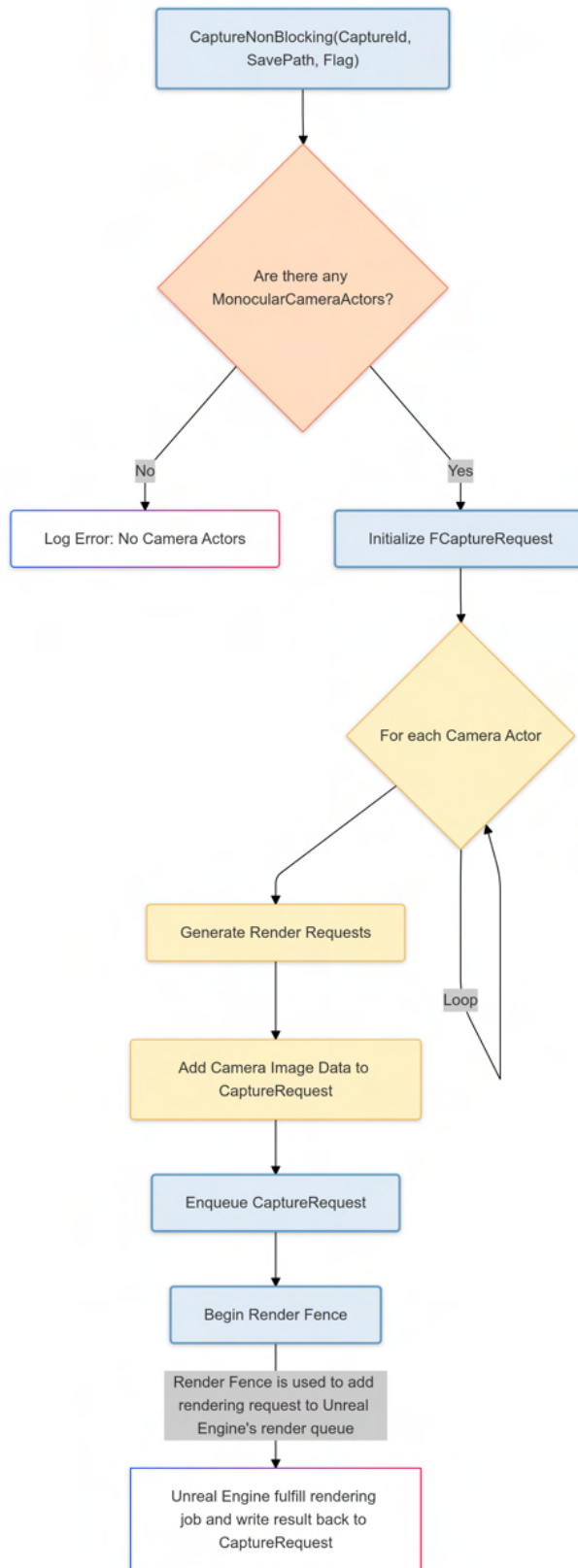


Рис. 3.10. Блок-схема операцій CaptureManager

Для управління процесом захоплення існує актор CaptureManager, який створюється при запуску сценарію. Він відповідає за керування всіма

камерами (MonoCameraActor) на карті. Оскільки зображення необхідно захоплювати з кількох камер і в різних форматах (колір, глибина, нормалі, сегментація), важливо забезпечити їхню синхронізацію. Для цього CaptureManager тимчасово призупиняє сценарій, захоплює всі необхідні зображення, а потім відновлює його. Зазвичай сценарій працює зі швидкістю 60 кадрів/с і триває від 2 до 5 хвилин. Однак, при генерації наборів даних CaptureManager захоплює зображення зі зниженою частотою (зазвичай від 1 до 10 кадрів/с) для підвищення продуктивності. Частота захоплення контролюється параметром frameskip у конфігураційному файлі. На рис. 3.10 представлена блок-схема операцій CaptureManager.

3.3.2. Генерація кольорового зображення

Генерація кольорових зображень за допомогою компонента SceneCapture2D в Unreal Engine — це простий процес, що вимагає мінімальних налаштувань. SceneCapture2D, що виконує роль віртуальної камери, захоплює сцену з заданої точки огляду та відтворює її на цільову текстуру. У порівнянні з попередніми версіями, UE5 пропонує значно покращений конвеєр рендерингу завдяки впровадженню Lumen та вдосконаленим можливостям трасування променів (ray tracing). Lumen — це система глобального освітлення в реальному часі в UE5, що забезпечує динамічне непряме освітлення та віддзеркалення, надаючи надзвичайно реалістичне освітлення без потреби в попередньо обчислених картах світла.

Для того, щоб захоплені кольорові зображення включали функції глобального освітлення Lumen та відповідали візуальній якості вікна перегляду симулятора, ми скоригували налаштування компонента SceneCapture2D. За замовчуванням, цей компонент може не захоплювати всі ефекти постобробки та розширені можливості освітлення, як Lumen. Тому ми модифікували його конфігурацію, щоб увімкнути рендеринг Lumen в межах захоплення сцени. Це включало активацію динамічного глобального освітлення та віддзеркалень, а також забезпечення включення ефектів

постобробки. Такі налаштування гарантують, що вихідні кольорові зображення, захоплені SceneCapture2D, точно відтворюють високоякісні ефекти освітлення та затінення, надані Lumen. На рис. 3.11 показані приклади результатів роботи компонента Color Camera.



Рис. 3.11. Приклад результатів роботи компонента Color Camera

3.3.3. Реалізація поля глибини карти

Зображення глибини, або карти глибини, є відтінками сірого, де інтенсивність кожного пікселя вказує на відстань від камери до об'єктів у сцені. Ці зображення є критично важливими в комп'ютерному зорі, особливо для систем стереобачення, які використовують зображення з кількох камер для обчислення глибини сцени. В UE5 зображення глибини генеруються за допомогою буфера глибини сцени (Scene Depth buffer), який фіксує відстань від камери до кожного пікселя. Наш компонент DepthCamera використовує ці налаштування для захоплення точних даних глибини. Для збереження високої точності значень глибини ми зберігаємо необроблені дані Scene Depth як масиви NumPy з плаваючою комою. Крім того, ми надаємо скрипт на Python для конвертації цих масивів у зображення у відтінках сірого для полегшення візуалізації. На рис. 3.12 показані приклади результатів роботи компонента Depth Camera після етапу постобробки.



Рис. 3.12. Результат роботи компонента Depth Camera після етапу постобробки

3.3.4. Реалізація карти нормалей

Карти нормалей — це зображення, які кодують орієнтацію нормалей поверхні сцени за допомогою значень кольорів RGB, де кожен піксель представляє напрямок, в який спрямована поверхня. Ця інформація є важливою для точного рендерингу освітлення, затінення та деталей поверхні в 3D-моделях. Подібно до симулятора CARLA, тут пропонується метод генерації карт нормалей за допомогою матеріалів постобробки (Post Process Materials). На рис. 3.13 показані приклади результатів роботи компонента Normal Camera.

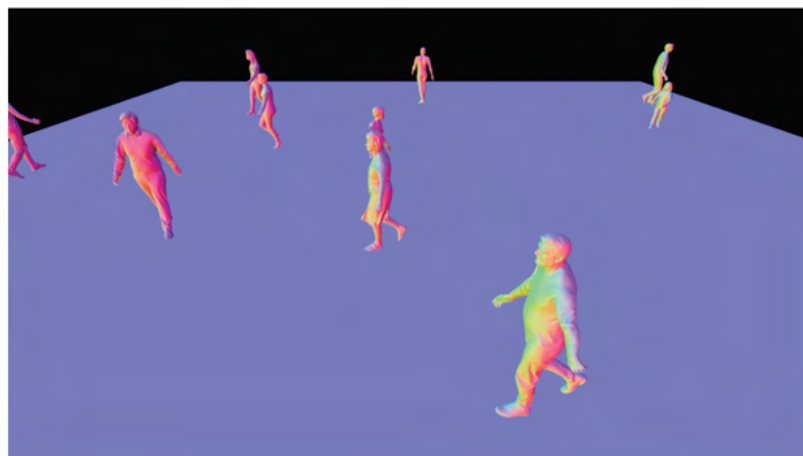


Рис. 3.13. Приклад результату роботи компонента Normal Camera

У Unreal Engine матеріали постобробки є користувацькими матеріалами, що застосовуються після початкового рендерингу сцени. Вони дозволяють додавати візуальні ефекти, такі як корекція кольору, розмиття та викривлення. Для генерації карти нормалей в системі розроблено спеціальний Material Blueprint (як показано на рис. 3.14), який видає вектор нормалі світового простору для кожного пікселя.

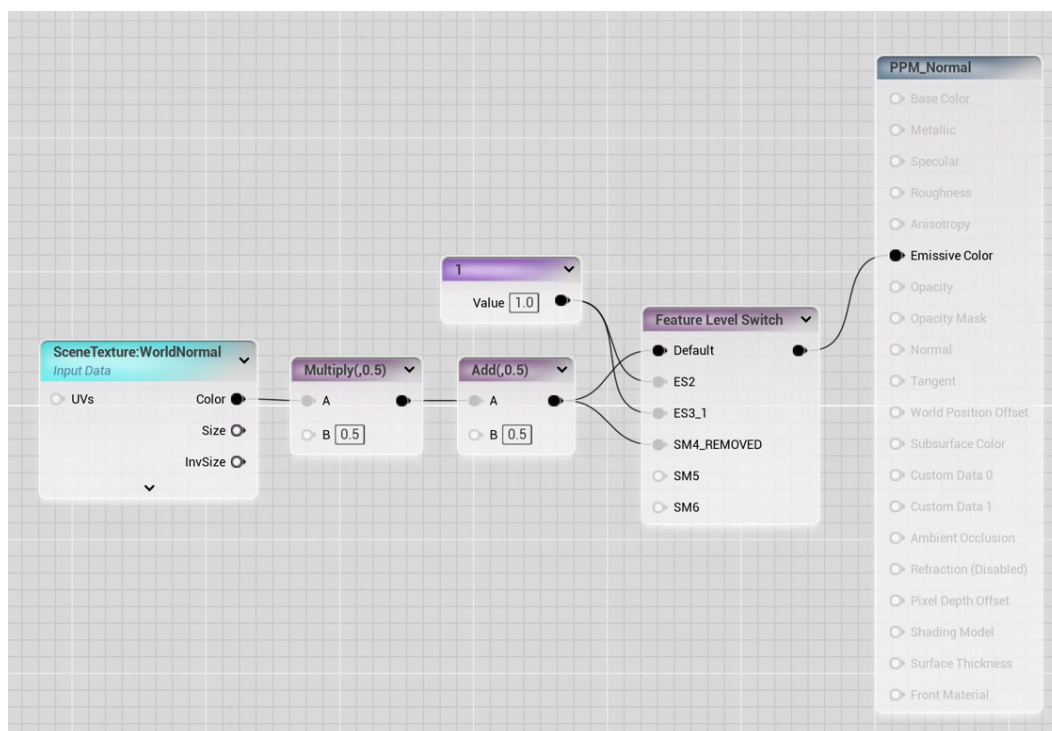


Рис. 3.14. Схема матеріалу для генерації карти нормалей

Рисунок 3.14 демонструє налаштування Material Blueprint для генерації карт нормалей. Вузол SceneTexture отримує карту нормалей (X, Y, Z) з UE5 зі значеннями від -1 до 1, які ми перемасштабовуємо до 0-1 для відображення в RGB. Emissive Color гарантує, що вихідне зображення не залежить від освітлення сцени. Ці векторні дані надають ключову інформацію про орієнтацію поверхонь у сцені. За допомогою перетворення цих векторів у колірну карту та застосування матеріалу як Post Process Material до нашої NormalCamera, ми можемо ефективно захоплювати та зберігати дані нормалей як зображення.

3.3.5. Зображення сегментації

Сегментація зображення — це фундаментальна концепція в комп'ютерному зорі, де кожен піксель в зображенні маркується відповідно до об'єкта, регіону або класу, до якого він належить. Зображення сегментації є важливими для таких застосувань, як автономне водіння, медична візуалізація та робототехніка. На рис. 3.15 показані приклади результатів роботи компонента SegmentCamera. Зазначимо, що термін зображення сегментації в цій статті стосується згенерованої анотації для семантичної сегментації.



Рис. 3.15. Приклад результату роботи компонента SegmentCamera

Робочий процес створення зображень сегментації в Unreal Engine є складнішим, ніж генерація полів глибини та карт нормалей, оскільки вимагає використання як буферів Custom Depth, так і спеціального Material Blueprint для матеріалу постобробки [35]. Зокрема, ми використовували буфери Custom Depth та Custom Stencil разом з матеріалом постобробки.

Спочатку ми активували Custom Depth-Stencil Pass в налаштуваннях проєкту, щоб дозволити об'єктам записувати значення користувацького трафарету. Для кожного об'єкта або класу, призначеного для сегментації, ми призначили унікальне значення трафарету, увімкнувши Render Custom Depth Pass та встановивши Custom Depth Stencil Value в їхніх властивостях рендерингу. Потім ми розробили матеріал постобробки, який отримує доступ до буфера Custom Stencil через вузол CustomStencil. Цей матеріал відображає

значення трафарету на певні кольори, що відповідають різним класам сегментації. Застосовуючи цей матеріал до нашого компонента Segment Camera, ми можемо відтворювати сцену таким чином, що колір кожного пікселя представляє його семантичну класифікацію. На рис. 3.16 показано налаштування Material Blueprint для генерації зображень сегментації. Вузол CustomStencil видає значення трафарету, присвоєне кожному об'єкту, тоді як CustomDepth та SceneDepth використовуються для перевірки оклюзії. Маска сегментації гарантує, що закриті частини об'єктів виключаються з кінцевого результату, створюючи правильну маску для сегментації.

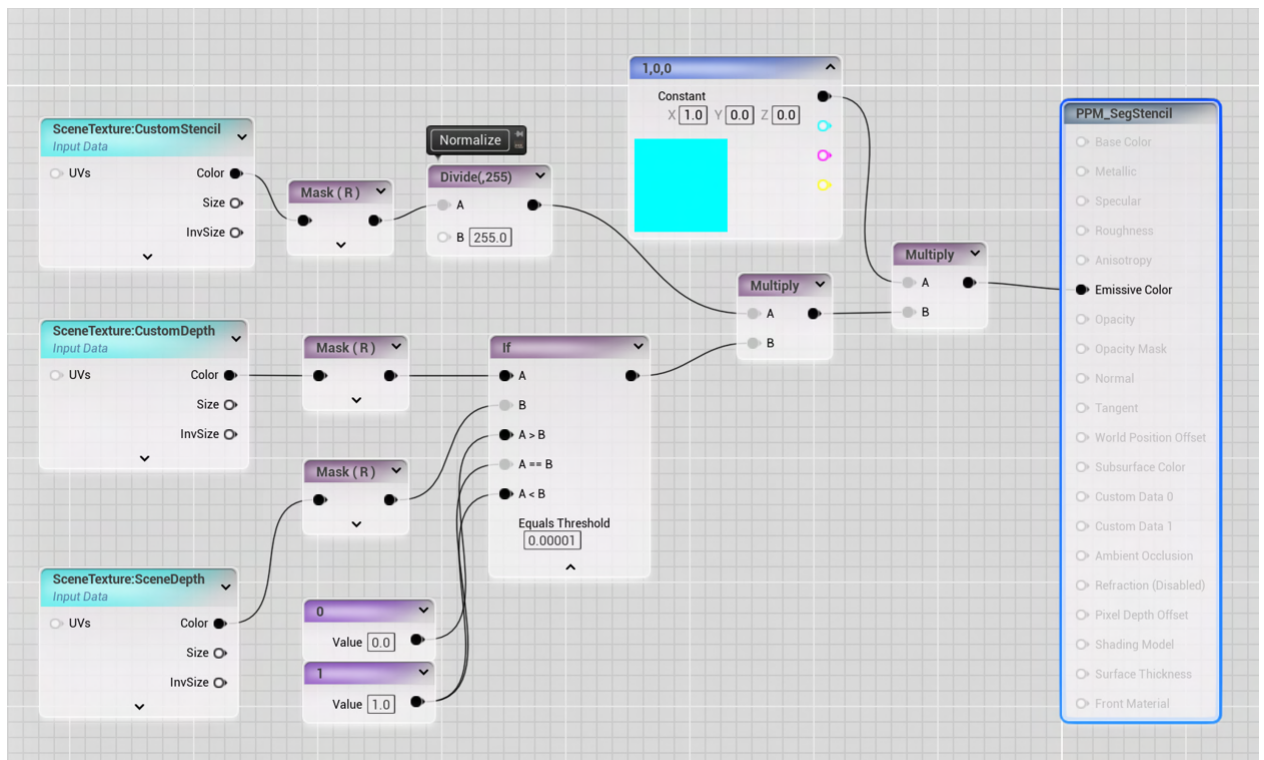


Рис. 3.16. Налаштування Material Blueprint для генерації зображень сегментації

У Unreal Engine акторам, створеним на мапі, можна призначати текстові теги, що допомагає організувати та класифікувати об'єкти для конкретних завдань. Система використовує систему тегування для семантичної сегментації, де будь-який об'єкт з тегом 'segment' вважається кандидатом для сегментації. Зазвичай, кожен об'єкт на мапі має два теги: тег

'segment' та додатковий тег, який вказує на клас об'єкта для семантичної сегментації. Коли сценарій запускається, система запитує всі об'єкти з тегом 'segment' і використовує їхні класові теги для створення хеш-таблиці існуючих класів. Потім він присвоює відповідне значення трафарету (0-255) кожному об'єкту на мапі. У вихідному зображенні ми кодуємо значення трафарету об'єкта в червоному каналі. Блок-схема, що описує цю операцію, показана на рисунку 2.11. Вихідне зображення може бути використане безпосередньо як маска для різних застосувань. Для кращої візуалізації результату сегментації ми надаємо скрипт на Python, який генерує нові зображення, де кожне значення трафарету асоціюється з певним колірним кодом. Ці кольори були згенеровані шляхом поділу колірному простору RGB для отримання 256 унікальних кольорів.

3.4. Процес анотування даних

Як зазначалося раніше, система була розроблена з основним фокусом на оцінці пози людини та аналізі поведінки. Окрім відтворення та запису різноманітних сценаріїв з анімацією персонажів, система може автоматично генерувати анотації для 3D обмежувальних рамок (bounding boxes) та пози. Щоб забезпечити сумісність із загальноприйнятими бенчмарками та широко використовуваними інструментами, анотація пози в системі суворо дотримуються стандартів. Це дозволяє безперешкодно інтегрувати систему з наявними інструментами для обробки даних, а також створювати специфічні набори даних для різних моделей комп'ютерного зору, таких як YOLO [30] та EfficientNet [41].

3.4.1. Механізм анотування

Для реалізації цієї функціональності система модифікує ресурс Skeleton Asset всіх персонажів, прикріплюючи сокети (віртуальні кістки) до відповідних позицій на тілі для кожної ключової точки, що вимагається

форматом анотації OpenPose. У рушії Unreal Engine сокети зазвичай використовуються для приєднання об'єктів (наприклад, аксесуарів) до персонажів. Однак, їхнє розташування також може бути динамічно отримано, що робить їх ідеальними для запиту позицій ключових точок на тілі персонажа. На рис. 3.17 та рис. 3.188 представлені приклади анотацій для 3D обмежувальних рамок та оцінки пози, що є "еталонними даними" (ground-truth).

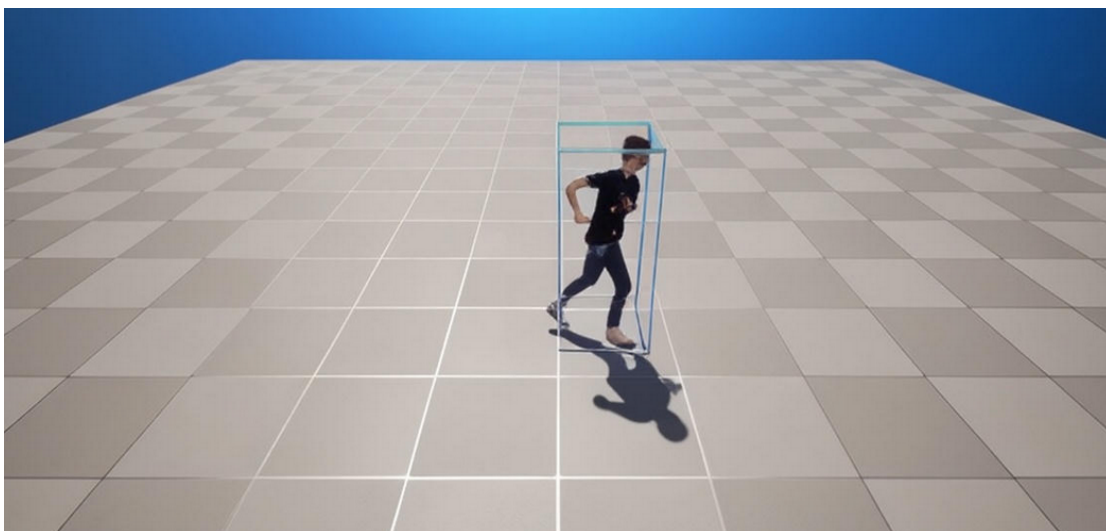


Рис. 3.17. Приклад еталонних даних (ground truth) 3D анотації обмежувальної рамки

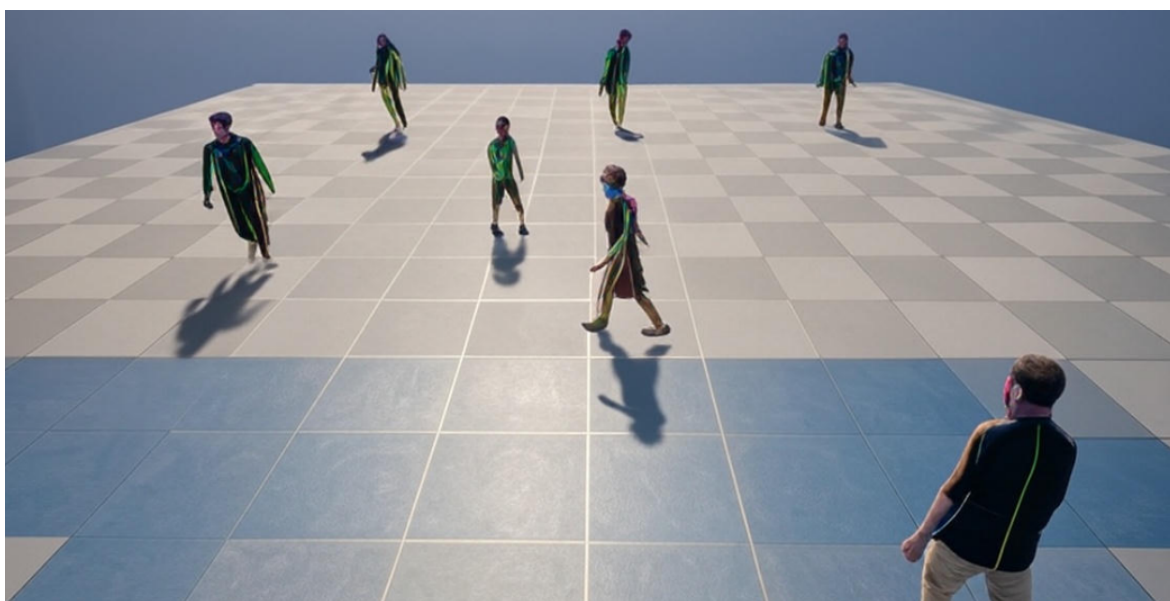


Рис. 3.18. Приклад еталонних даних (ground truth) 3D анотації пози

Для оптимізації процесу збору цих даних про ключові точки, система включає компонент анотування (Annotation Component), який можна приєднати до будь-якого персонажа на мапі. Цей компонент запитує та записує позиції ключових точок персонажа в кожному кадрі, що дозволяє генерувати детальні дані для анотації пози. Крім того, компонент анотування також відстежує 3D обмежувальну рамку персонажа в кожному кадрі. Цей процес є простішим, оскільки Unreal Engine 5 надає API (наприклад, GetActorBounds) для легкого отримання обмежувальної рамки, яка охоплює персонажа.

3.4.2. Зберігання та управління даними

Для ефективного збору та управління даними під час їх генерації система містить менеджер анотацій (Annotation Manager), який можна додати на мапу. Менеджер анотацій відстежує всі компоненти анотування, агрегує дані про позу та обмежувальні рамки від кожного персонажа та зберігає зібрані анотації у вихідну папку "Annotations" щоразу, коли користувач робить запит на знімок.

Детальніше, анотації пози зберігаються у CSV-файлах. Кожен рядок представляє персонажа на сцені, ідентифікованого за «Id» та «Name», з 3D координатами [x, y, z] для ключових точок, таких як ніс, шия, плечі, лікті, зап'ястя, стегна, коліна, щиколотки, очі та вуха. Анотації обмежувальних рамок зберігаються окремо в CSV-файлах. Аналогічно до формату анотації пози, кожен запис включає «Id» та «Name» для персонажа. Однак, замість ключових точок, він записує 3D координати восьми кутів обмежувальної рамки, де перші чотири кути визначають верхню частину, а останні чотири — нижню.

На рис. 3.19 показано, як система прикріплює сокети до моделей персонажів для запиту пози в реальному часі, а на рис. 3.20 зображена структура даних для анотації 3D пози, якою керує менеджер анотацій.

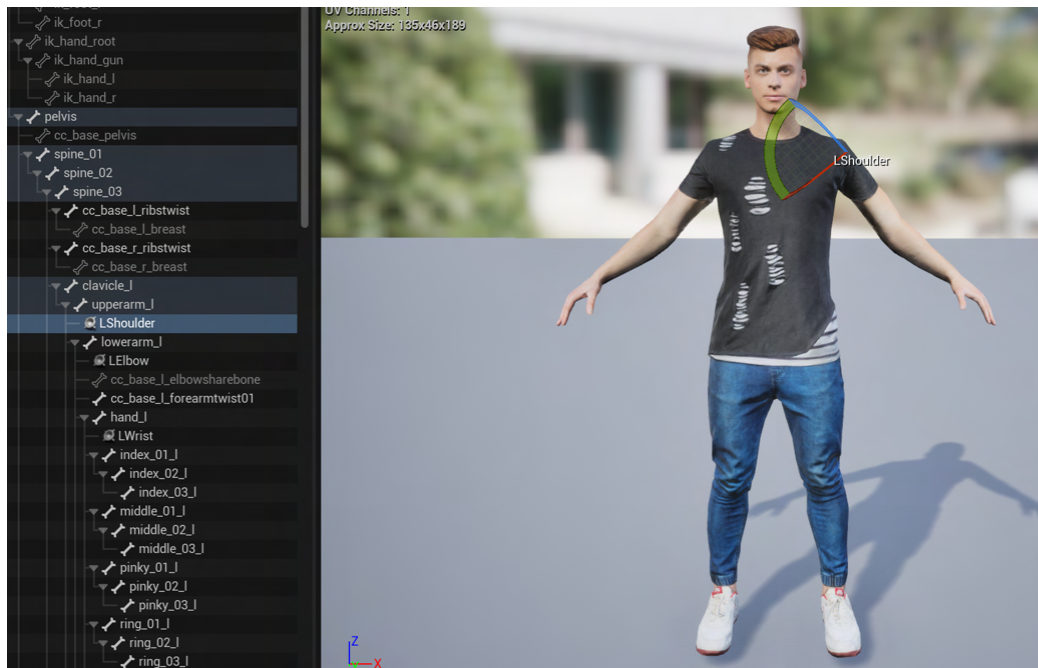


Рис. 3.19. Приклад прикріплення сокетів до моделі персонажа для запиту розташування пози в реальному часі в Unreal Engine

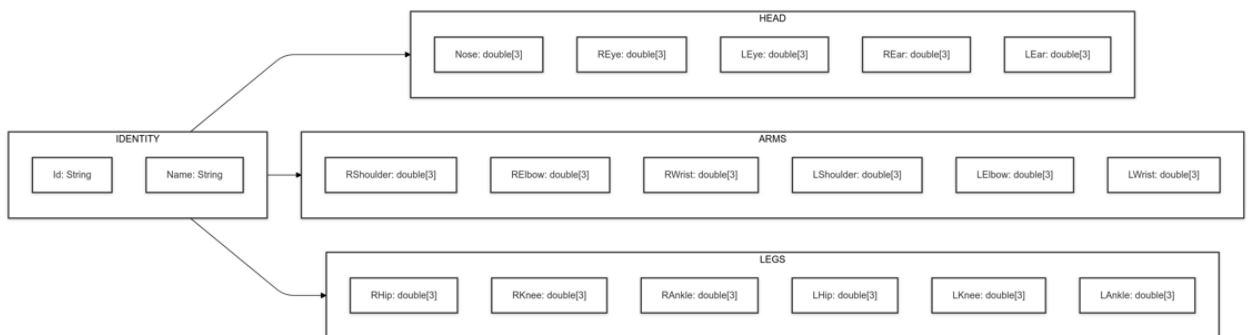


Рис. 3.20. Візуалізація структури даних для анотації 3D пози

Менеджер анотацій відповідає за збір даних про 3D позу від усіх персонажів на мапі та збереження результату у файл формату «.csv» для кожного запиту на знімок

3.4.3. Трансформація координат

У цій роботі дані анотацій зберігаються в 3D світових координатах (world coordinates) таким чином, що вони є незалежними від камери. Це може бути корисним при проведенні аналізу даних та реконструкції сцени. Однак, ми усвідомлюємо, що перетворення цих 3D-координат у 2D піксельні

координати є критично важливим для створення наборів даних, сумісних з існуючими моделями комп'ютерного зору. Для спрощення цього перетворення надає Jupyter-ноутбук, що містить службові функції для трансформації 3D даних анотацій у 2D. Цей процес використовує матриці проєкції камери, збережені у файлі `cameras.csv`, який генерує CaptureManager для зберігання всіх відповідних метаданих камери. Надана матриця проєкції камери може бути безпосередньо використана для перетворення точок зі світових координат у координати сцени з мінімальними зусиллями. Крім того, метадані камери включають розміри зображення, кут огляду (field of view), а також вектори переміщення та обертання камери (крен, тангаж та рискання), що дозволяє нам реконструювати як внутрішню (intrinsic), так і зовнішню (extrinsic) матриці камери, якщо це необхідно.

3.5. Дослідження і тестування симуляційного середовища

Для забезпечення зручності тестування системи було представлено симуляційне середовище у вигляді двійкового виконуваного файлу для операційної системи Windows. У поточному стані система пропонує два різні симуляційні середовища: BasicMap та Downtown. При запуску програми за замовчуванням відкривається середовище BasicMap. Користувачі можуть перемикатися між мапами за допомогою спадного меню у верхньому лівому куті екрана. Для кожної мапи було надано два зразки сценаріїв, що охоплюють більшість типових випадків використання для оцінки нашого рішення. Користувачі можуть відтворювати різні сценарії, використовуючи спадне меню в нижньому лівому куті.

3.5.1. Процес генерації даних

При першому запуску сценарій відтворюється автоматично, але користувачі мають можливість призупинити або відновити відтворення. Крім того, вони можуть обертати та переміщувати камеру огляду за допомогою

миші та клавіатури. Для генерації набору даних користувачі можуть натиснути кнопку «Generate Dataset», що призведе до скидання сценарію та його повторного відтворення, одночасно захоплюючи як зображення, так і дані анотацій. Відеозаписи сценаріїв на деяких мапах представлені на рис. 3.21 – 3.22.

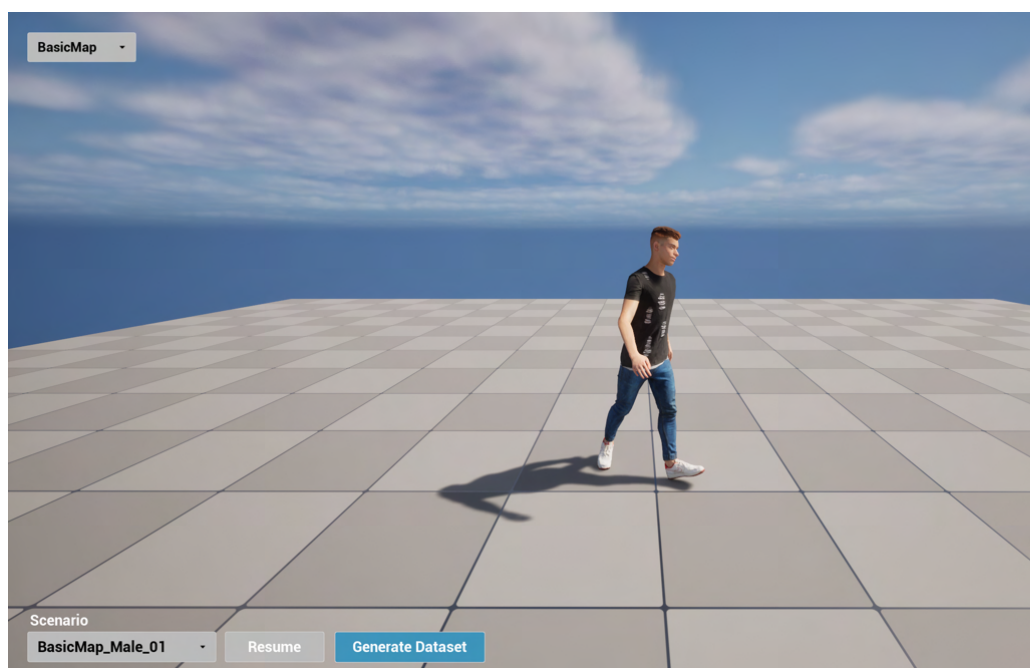


Рис. 3.21. Сценарій 1 у середовищі BasicMap

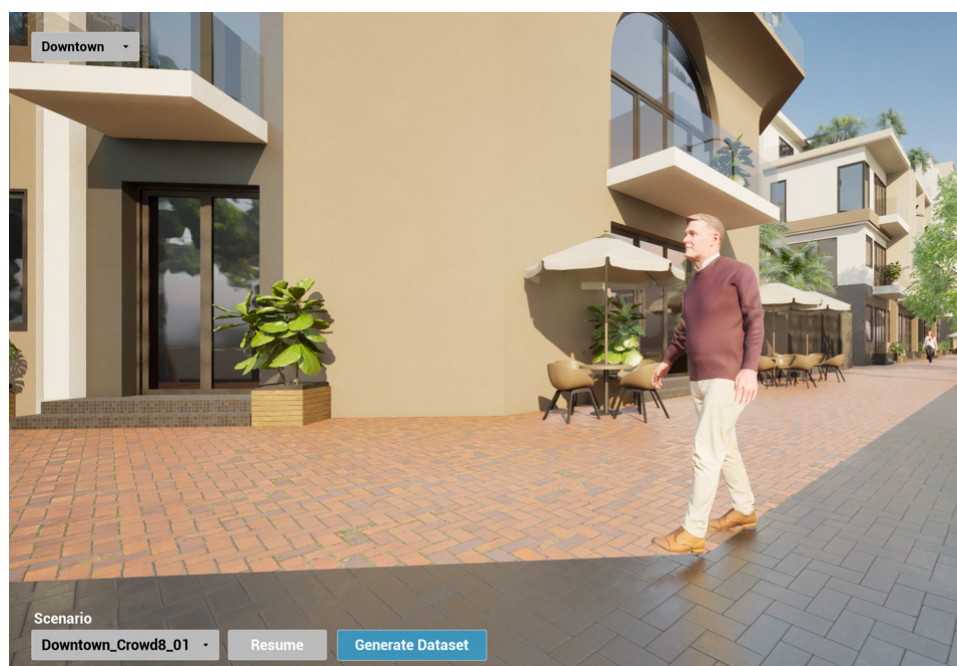


Рис. 3.22. Сценарій 2 у середовищі BasicMap

3.5.2. Потік даних з камери

Крім того, оскільки система також транслює потік з камери огляду як потік з вебкамери на хост-комп'ютері, користувачі можуть використовувати будь-яку медіа-бібліотеку або додаток, що працює з вебкамерами, для захоплення цього потоку. Для демонстрації цієї функціональності було надано скрипт на Python для запуску попередньо навченої моделі детекції об'єктів YOLOv10 та виконання висновків на потоці з вебкамери. Результат демонстрації показано на рис. 3.23.



Рис. 3.23. Приклад роботи виявлення об'єктів на потоці з вебкамери

3.6 Навчання YOLOv11-Pose на довільному наборі даних

Для демонстрації ефективності набору даних, згенерованого системою, у навчанні та покращенні існуючої моделі оцінки пози, було сформовано набір даних із зображень, отриманих у двох сценаріях в середовищі BasicMap. Цей набір даних складається з 3150 зображень, знятих з двох ракурсів камери, і розділений на тренувальний та валідаційний набори у співвідношенні 4:1.

3.6.1. Підготовка даних та трансформування

Для підтримки робочого процесу надаємо два Jupyter-ноутбуки: один призначений для попередньої обробки згенерованого набору даних для забезпечення сумісності з новітніми моделями YOLO, а інший — для навчання самої моделі. Перед зануренням у процес навчання важливо спочатку розглянути етапи трансформації даних, необхідні для адаптації даних пози для використання з попередньо навченими моделями YOLO для оцінки пози.

Як було показано раніше, вихідні дані анотації 3D пози містять ідентифікатор (ID), ім'я (Name) та 18 ключових точок (ніс, шия, плечі, лікті, зап'ястя, стегна, коліна, щиколотки, очі та вуха) у 3D світових координатах. Цей формат відповідає стандартному формату з оригінальної наукової роботи OpenPose. Формат оцінки пози моделі YOLO, з іншого боку, базується на форматі COCO, який дещо відрізняється від OpenPose. Цей формат починається з мітки класу (0 для людей), за якою слідують нормалізовані координати обмежувальної рамки (x_0 , y_0 , w , h), що представляють центр і розміри рамки. Він включає 17 ключових точок у вигляді трійок (x , y , *видимість*), де "видимість" вказує, чи знаходиться ключова точка в межах зображення. На відміну від пропонованої системи, формат YOLO COCO-Pose не включає ключову точку "шия", ймовірно, тому що її можна інтерполювати з ключових точок лівого та правого плеча. Етапи трансформації даних, необхідні для перетворення вихідної анотації у формат, сумісний з YOLO, були наведені на початку третього розділу.

Для перетворення 3D анотацій системи у сумісний з YOLO 2D формат, кожна 3D ключова точка спочатку повинна бути спроектована на 2D площину. Це вимагає побудови матриці проєкції камери, яку можна знайти для кожної камери у файлі `cameras.csv`, розташованому у вихідній папці анотацій. Після проєктування координати нормалізуються на основі розмірів зображення. Обмежувальна рамка обчислюється шляхом знаходження мінімальних і максимальних значень x і y серед спроектованих ключових

точок, визначаючи центр і розмір відповідно до вимог YOLO. Прапори видимості встановлюються шляхом перевірки, чи лежить кожна ключова точка в межах зображення, а ключова точка "шия" виключається для забезпечення сумісності з форматом COCO-Pose.

3.6.2. Навчання та результати

Для цього експерименту ми обрали YOLOv11N-Pose (версію Nano) через його відмінний баланс між обчислювальною ефективністю та точністю, що робить його добре пристосованим для навчання на апаратному забезпеченні з обмеженими ресурсами. Враховуючи, що модель навчалася на локальному комп'ютері з обмеженою відеопам'яттю (VRAM), варіант Nano дозволив нам досягти високої продуктивності, не перевищуючи обмеження пам'яті. Ми навчали модель YOLOv11 на цьому кастомному наборі даних протягом 100 епох. Підсумкові результати представлені в таблиці 3.1 і на рис. 3.24 і 3.25.

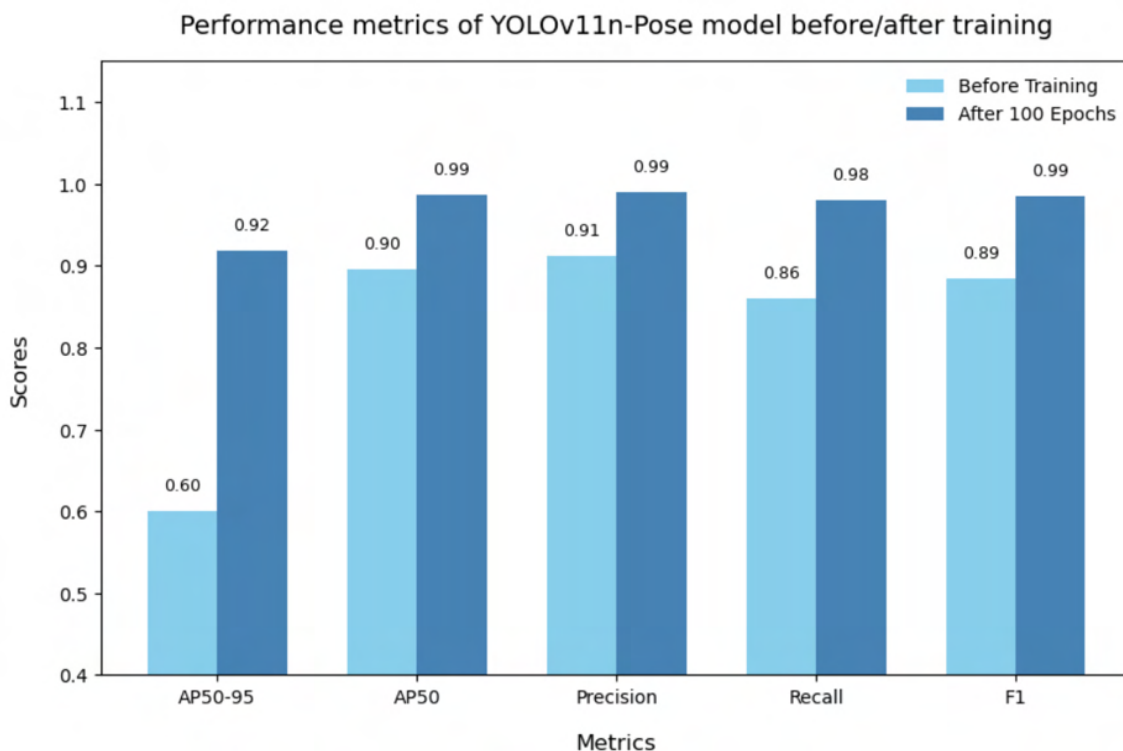


Рис. 3.24. Гістограма метрик продуктивності моделі YOLOv11-pose на валідаційному наборі даних

Метрики продуктивності моделі YOLOv11-pose на валідаційному наборі даних до та після навчання

Метрика	До навчання	Після 100 епох
AP50-95	0.60111	0.91797
AP50	0.89537	0.98748
Precision	0.91151	0.99048
Recall	0.86018	0.98051
F1	0.88510	0.98547

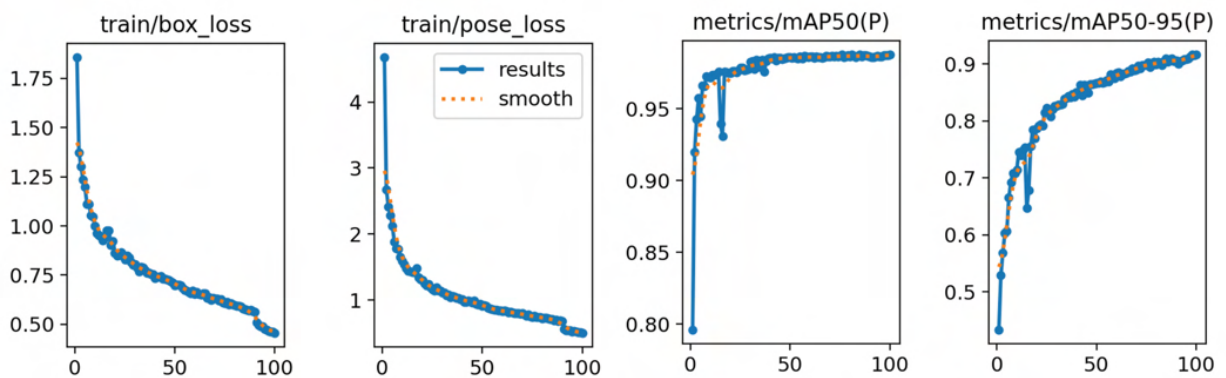


Рис. 3.25. Результати навчання YOLOv11

Результати навчання моделі YOLOv11n-Pose на кастомному наборі даних показують суттєве покращення продуктивності всього після 100 епох. Спочатку модель досягла показника AP_{50-95} 0.6011, при AP_{50} 0.8954, точності 0.9115, повноті 0.8602 та F1 0.8851. До кінця навчання всі ці метрики значно зросли: AP_{50-95} до 0.9180, AP_{50} до 0.9875, точність до 0.9905, повнота до 0.9805, а F1 до 0.9855. Крім того, криві втрат для рамки та пози показують стабільне зниження, тоді як метрики mAP_{50} та mAP_{50-95} демонструють чітку тенденцію до зростання. Це свідчить про те, що процес навчання працює належним чином. Ці результати підкреслюють потенціал синтетичних даних від системи для підтримки та покращення продуктивності моделей оцінки пози. Додаткові приклади результатів висновків моделі показані на рис. 3.26. Деталі конфігурацій навчання доступні в таблиці 3.2.

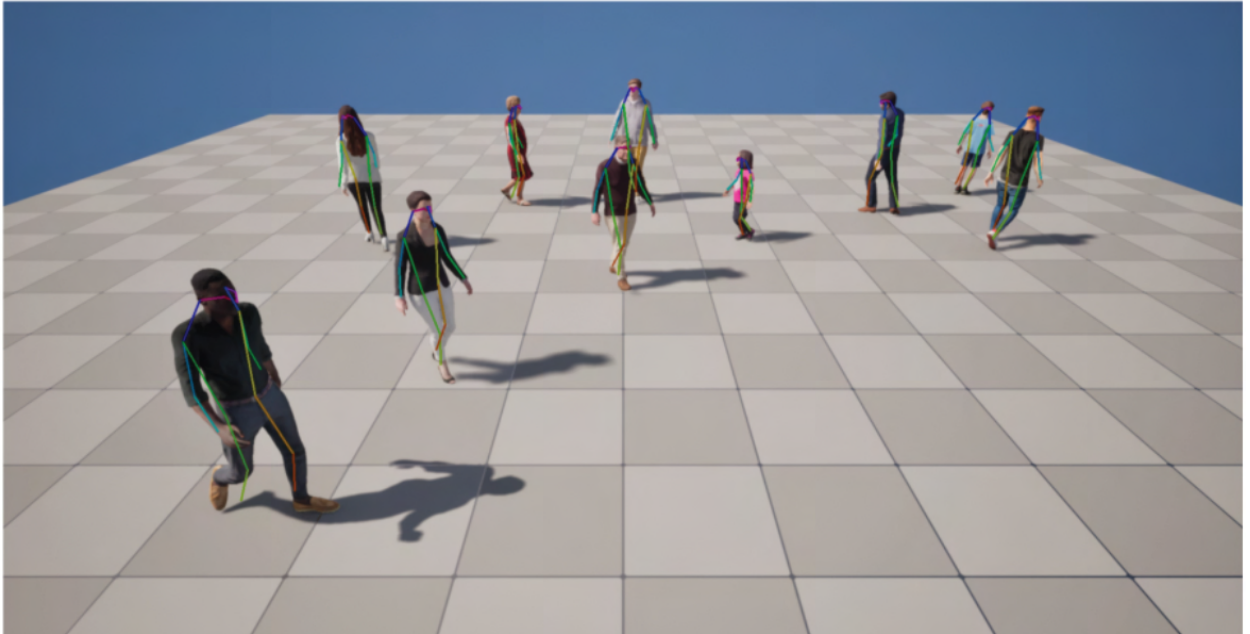


Рис. 3.26. Приклад роботи YOLOv11 на тестовому зображенні після навчання

Таблиця 3.2.

Параметри доналаштування для YOLOv11

Параметр	Значення
Швидкість навчання	0.01
Кінцева швидкість	0.01
Розмір пакету	16
Кількість епох	100
Розмір зображення	640
Оптимізатор	Adam
Момент	0.937
Зменшення ваги	0.0005

Отже, запропонована система була розроблена як генератор синтетичних наборів даних, спеціально адаптований для завдань оцінки пози людини та аналізу поведінки. З акцентом на наданні інструментів та робочих процесів для створення кастомних наборів даних, система дозволяє захоплювати складні, орієнтовані на людину сценарії з реалістичною анімацією та автоматичним анотуванням. Дана робота демонструє можливості системи для генерації наборів даних, які підвищують точність існуючих моделей

оцінки пози у доменно-специфічних завданнях, пропонуючи цінну альтернативу традиційним, трудомістким методам створення наборів даних.

Загалом, система надає гнучкий, економічно ефективний підхід до генерації синтетичних наборів даних, що робить його корисним інструментом для дослідників у сфері комп'ютерного зору. Це не тільки розширює можливості, що можуть бути досягнуті за допомогою синтетичних даних, але й відкриває нові перспективи для створення вузькоспеціалізованих наборів даних, що відповідають галузевим стандартам, сприяючи розвитку досліджень та застосувань оцінки пози в людиноорієнтованому штучному інтелекті.

Висновки до розділу

У третьому розділі реалізовано систему генерації синтетичних даних, що охоплює процеси побудови карт середовищ, розробки сценаріїв руху та формування багатомодальних візуальних даних (кольорових зображень, карт глибини, нормалей, сегментаційних карт). Розроблений механізм анотування забезпечує автоматизацію та точність при трансформації координат, що мінімізує людський фактор у створенні датасетів. Тестування потоків даних із камер підтвердило стабільність роботи системи та якість одержуваних результатів.

ВИСНОВКИ

У процесі виконання магістерської роботи було здійснено комплексне дослідження предметної області, що охоплює застосування технік комп'ютерного зору для задач генерації даних, аналізу позиціювання об'єктів та моделювання їх поведінки. Проведене дослідження дало змогу сформулювати науково обґрунтовану методологію побудови моделей генерації синтетичних даних, які забезпечують підвищення ефективності алгоритмів оцінки пози та покращення результатів навчання систем виявлення об'єктів.

У першому розділі роботи виконано детальний аналіз сучасного стану застосування синтетичних даних у комп'ютерному зорі. Визначено ключові переваги та недоліки традиційних наборів даних, серед яких – висока вартість збору та анотування, обмеженість сценаріїв та недостатнє різноманіття. Поглиблене вивчення наборів SURREAL, Human3.6M та CMU Panoptic дозволило ідентифікувати можливості та обмеження їх використання в завданнях оцінки пози людини. На основі цього обґрунтовано необхідність створення гнучких моделей генерації синтетичних даних, здатних адаптуватися до різних дослідницьких та прикладних завдань.

У другому розділі досліджено сучасні моделі та алгоритми оцінки позиціювання об'єктів. Особлива увага була приділена аналізу архітектур OpenPose та YOLO, які є фундаментом для розв'язання задач багатокласової детекції та оцінки положення об'єктів у просторі. Розглянуто та протестовано середовища симуляції, зокрема CARLA та UnrealCV, що виступають як інструментарій для генерації даних та створення експериментальних сценаріїв. Порівняльний аналіз запропонованої системи генерації даних із вказаними платформами підтвердив її функціональну гнучкість, розширюваність та здатність інтегрувати механізми для оцінки поведінки об'єктів. Також розроблено архітектуру системи, що забезпечує модульність та можливість подальшої адаптації до нових дослідницьких умов.

У третьому розділі реалізовано модель генерації даних, що включає формування карт середовищ, сценаріїв руху та різноманітних типів візуалізацій (кольорових зображень, карт глибини, нормалей, сегментаційних карт). Розроблений механізм анотування даних гарантує коректність трансформації координат і підтримку ефективного управління наборами. Проведено дослідження потоків даних із камер у симуляційному середовищі, що дозволило оптимізувати процес збору та анотації в реальному часі. Додатково здійснено навчання моделі YOLOv11-Pose на сформованому синтетичному наборі, результати якого підтвердили практичну доцільність та ефективність використання запропонованої системи для покращення точності оцінки позиціонування об'єктів.

Узагальнюючи результати дослідження, можна зробити такі ключові висновки:

- створення та використання синтетичних наборів даних є ефективним підходом для подолання обмежень традиційних датасетів;
- розроблена система генерації забезпечує широкий спектр сценаріїв і типів даних, що підвищує якість навчання моделей комп'ютерного зору;
- інтеграція моделей OpenPose та YOLO у зв'язці з власним симуляційним середовищем відкриває нові можливості для дослідження поведінки об'єктів у динамічних умовах;
- результати експериментального навчання підтвердили здатність синтетичних даних підвищувати точність та узагальнюваність моделей.

Таким чином, у роботі запропоновано комплексний підхід до моделювання генерації даних про оцінку позиціонувань та поведінку об'єктів, що поєднує методи комп'ютерного зору, симуляційні технології та сучасні алгоритми детекції.

ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Cao, Z., Simon, T., Wei, S. E., & Sheikh, Y. (2017). Realtime Multi-Person 2D Pose Estimation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
2. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
3. Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. Proceedings of the 36th International Conference on Machine Learning (ICML).
4. Wang, C., Zhang, H., Ren, X., & Gao, R. (2024). YOLOv11: State-of-the-Art Object Detection and Pose Estimation. arXiv preprint arXiv:2403.01198.
5. Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2010). The Pascal Visual Object Classes (VOC) Challenge. International Journal of Computer Vision, 88(2), 303-338.
6. Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Dollár, P. (2014). Microsoft COCO: Common Objects in Context. arXiv preprint arXiv:1405.0312.
7. Newell, A., Yang, K., & Deng, J. (2016). Stacked Hourglass Networks for Human Pose Estimation. Proceedings of the European Conference on Computer Vision (ECCV).
8. Simo-Serra, E., & Paduano, S. (2021). Pose-conditioned Attention for Human Pose Estimation. Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV).
9. Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI).

10. Long, J., Shelhamer, E., & Darrell, T. (2015). Fully Convolutional Networks for Semantic Segmentation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
11. Papon, J., & Shooner, T. (2018). U-Net for Human Pose Estimation. Master's Thesis, University of Toronto.
12. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. Advances in Neural Information Processing Systems (NIPS).
13. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going Deeper with Convolutions. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
14. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
15. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., & Unterthiner, T. (2020). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. arXiv preprint arXiv:2010.11929.
16. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention Is All You Need. Advances in Neural Information Processing Systems (NIPS).
17. Ros, G., Sellart, L., Materzynska, J., & Fua, P. (2016). The SYNTHIA Dataset: A New Dataset for Urban Scene Understanding from a Simulated Perspective. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
18. Handa, A., & Newcombe, R. A. (2015). 3D Object Segmentation in RGB-D Images. Proceedings of the IEEE International Conference on Robotics and Automation (ICRA).
19. Wang, Z., & Chen, H. (2019). Synthetically Generated Data for Pose Estimation. Journal of Computer Vision, 43(2), 112-125.

20. Fabbri, F., & Sminchisescu, C. (2018). Multi-view 3D Pose Estimation with Deep Learning. arXiv preprint arXiv:1803.04543.
21. Pavlakos, G., Zhou, X., & Gkioxari, G. (2018). Coarse-to-fine Human Pose Estimation. Proceedings of the European Conference on Computer Vision (ECCV).
22. Zhu, Y., Song, M., & Yang, B. (2020). A Survey of Synthetic Data Generation for Computer Vision. IEEE Transactions on Pattern Analysis and Machine Intelligence.
23. Li, Y., Huang, H., & Zhou, Y. (2021). The Role of Synthetic Data in Deep Learning for Computer Vision. Journal of Artificial Intelligence Research.
24. Huk, P., & Schramm, J. (2019). Generating Synthetic Data for Autonomous Driving. Proceedings of the IEEE International Conference on Intelligent Transportation Systems (ITSC).
25. Zhang, Y., & Li, C. (2022). A Comparative Study of OpenPose and YOLO-Pose. International Journal of Advanced Robotics and Automation.
26. Chen, L., Xu, H., & Liu, Q. (2020). Real-Time Human Pose Estimation with Lightweight Networks. Proceedings of the IEEE International Conference on Image Processing (ICIP).
27. Chollet, F. (2017). Xception: Deep Learning with Depthwise Separable Convolutions. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
28. Dosovitskiy, A., Ros, G., & et al. (2017). CARLA: An Open and Scalable Simulator for Autonomous Driving Research. Proceedings of the 1st Annual Conference on Robot Learning.
29. Geiger, A., Lenz, P., & Urtasun, R. (2012). Are We Ready for Autonomous Driving? The KITTI Vision Benchmark Suite. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
30. Richter, S. R., & Berenz, V. (2016). Playing for Data: Synthetic Data Generation with the Unreal Engine. Proceedings of the European Conference on Computer Vision (ECCV).

- 31.Varol, G., Laptev, I., & Schmid, C. (2017). Learning from Synthetic Data for Human Pose Estimation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
- 32.Habermann, M., & Theiler, B. (2020). SynDeth: Synthetic Dataset for Human Detection and Tracking. arXiv preprint arXiv:2009.00123.
- 33.Martinez, J., & Ponsa, D. (2021). The UnrealCV Project: A Photorealistic Simulator for Computer Vision Research. International Journal of Computer Vision.
- 34.Chen, L., & Fan, X. (2018). Human Pose Estimation in the Wild. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).