

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)Кафедра Інформатики
(повна назва)Рівень вищої освіти другий (магістерський)Спеціальність 122 Комп'ютерні науки
(код і повна назва)Тип програми освітньо-професійнаОсвітня програма Інформатика
(повна назва освітньої програми)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

« ____ » _____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУстудентові Гречишкіну Данилу Сергійовичу
(прізвище, ім'я, по батькові)1. Тема роботи Дослідження питання створення розмічених наборів даних за допомогою розробленого модуля для редактора Blender на прикладі вирішення задачі підрахунку натовпу

затверджена наказом по університету від 25 листопада 2024 року № 1246Ст

2. Термін подання студентом роботи до екзаменаційної комісії 23 грудня 2024 р.3. Вихідні дані до роботи математичні моделі нейронних мереж, теоретичні відомості про методи класифікації зображень, теоретичні відомості щодо написання розширень до програмного забезпечення Blender, заготовлені набори розмічених зображень, посібник Python, перелік використаних програмних засобів: огляд сучасних методів оцінки зображень натовпу.

4. Перелік питань, що потрібно опрацювати в роботі _____

1. Огляд новітніх методів підрахунку натовпу.

2. Математичні моделі нейронних мереж.

3. Огляд розмічених наборів даних для підрахунку натовпу.

4. Аналіз збалансованості розмічених наборів даних.

5. Розробка розширення для програмного забезпечення Blender.

6. Огляд можливості генерації штучних розмічених наборів даних.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) актуальність проблеми обробки зображень, постановка задачі, тестові зображення.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	25.11.2024	
2	Аналіз завдання, підбір літератури	26.11.24-29.11.24	
3	Аналіз літератури з досліджуваної проблеми	30.11.24-03.11.24	
4	Аналіз технічних засобів та наборів даних	04.12.24-06.11.24	
5	Розробка розширення	07.12.24-09.12.24	
6	Програмна реалізація	10.12.24-17.12.24	
7	Оформлення пояснювальної записки	18.12.24-20.12.24	
8	Перевірка на плагіат	21.12.2024	
9	Рецензування	22.12.2024	
10	Підготовка презентації та доповіді	23.12.2024	
11	Занесення роботи в електронний архів	01.01.2025	
12	Попередній захист кваліфікаційної роботи	02.01.2025	

Дата видачі завдання 25 листопада 2024 р.

Студент _____
(підпис)

Керівник роботи _____
(підпис)

_____ доц. Яковлева О.В.
(посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 94 с., 6 табл., 42 рис., 3 дод., 65 джерело.

ДАТАСЕТИ ДЛЯ ПІДРАХУНКУ НАТОВПУ, ГЕНЕРАЦІЯ СЦЕН, P2PNET, BLENDER, ГЕОМЕТРИЧНІ ВУЗЛИ, РОЗМІТКА ДАТАСЕТІВ.

Об'єктом роботи є створення розмічених наборів даних для підрахунку кількості людей у натовпі шляхом генерування синтетичних зображень. У роботі проаналізовано різноманітні набори даних, зокрема найбільш вживані.

Метою роботи є дослідження питання створення розмічених наборів даних за допомогою розробленого модуля для редактора Blender, який дозволяє автоматично створювати зображення натовпу та анотації для них. Особливу увагу приділено генерації високоякісних синтетичних зображень із детальним налаштуванням умов освітлення, щільності натовпу та перспективи, а також точності й зручності анотації об'єктів на зображеннях.

Дослідження базується на можливостях Blender для генерування сцен та створення анотацій, необхідних для навчання моделей глибинного навчання, таких як згорткові нейронні мережі для задач підрахунку людей у натовпі. Середовищем розробки обрано Blender, що дозволило використати його гнучкі інструменти моделювання для створення синтетичних датасетів, а також Python для програмування модуля.

DATSETS FOR CROWD COUNTING, GENERATION OF SCENES, P2PNET, BLENDER, GEOMETRY NODES, ANNOTATION OF DATASETS.

The object of the work is to create labelled datasets for counting the number of people in a crowd by generating synthetic images. The paper analyses various datasets, including the most commonly used ones.

The aim of this work is to study the issue of creating labeled datasets using a developed module for the Blender editor that allows you to automatically create crowd images and annotations for them. Particular attention is paid to the generation of high-quality synthetic images with detailed settings for lighting conditions, crowd density and perspective, as well as the accuracy and convenience of annotating objects in the images.

The research is based on the capabilities of Blender to generate scenes and create annotations necessary for training deep learning models, such as convolutional neural networks for crowd counting tasks. Blender was chosen as the development environment, which allowed us to use its flexible modelling tools to create synthetic datasets and Python to program the module.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	8
Вступ.....	9
1 Сучасний стан питання підрахунку людей у натовпі.....	12
1.1 Підходи для підрахунку людей у натовпі	12
1.2 Підрахунок на основі розв’язання задачі комп’ютерного зору	13
1.2.1 Підрахунок людей на основі CSRNet.....	15
1.2.2 Підрахунок людей на основі MCNN	16
1.2.3 Підхід на основі попередньої класифікації	17
1.2.4 Сучасні підходи та їхня точність.....	18
1.3 Огляд датасетів, що містять натовп	20
1.4 Можливості створення штучних датасетів	21
1.4.1 Формування датасетів за допомогою нейронних мереж	23
1.4.2 Програмне забезпечення для розмітки датасетів.....	23
1.4.3 Редактор Blender.....	24
1.5 Сучасне програмне забезпечення для вирішення завдань комп’ютерного зору	24
1.6 Постановка задачі дослідження.....	25
2 Аналіз методів і структур моделей підрахунку людей у натовпі та аналіз точності підрахунку для різного натовпу різної щільності	28
2.1 Метрики для вимірювання точності підрахунку людей у натовпі	28
2.2 Математична модель найбільш точних сучасних методів підрахунку людей.....	31
2.2.1 APGCC	32
2.2.2 CLIP-EBC	33
2.2.3 P2PNet.....	34
2.3 Аналіз особливостей найбільш популярних датасетів	36

2.3.1	Загальні характеристики датасетів ShanghaiTech, UCF-QNRF, JHU-CROWD++, UCF-CC-50.....	37
2.3.2	Оцінка збалансованості датасетів ShanghaiTech, UCF-QNRF, JHU-CROWD++, UCF-CC-50.....	39
2.4	Аналіз якості підрахунку людей сучасних методів на датасетах ShanghaiTech, UCF-QNRF, JHU-CROWD++, UCF-CC-50	41
2.4.1	Точність на всьому датасеті.....	41
2.4.2	Точність для зображень натовпу різної щільності	43
3	Розробка plugin для редактора Blender для створення штучного датасету натовпу з метою підвищення точності підрахунку натовпу різної щільності	48
3.1	Особливості створення plugin для редактора Blender.....	48
3.2	Проектування та розробка розширення.....	50
3.2.1	Структура розширення	50
3.2.2	Налаштування геометричних вузлів	52
3.2.3	Налаштування камери відносно перспективи сцени.....	55
3.2.4	Постобробка зображення	57
3.2.5	Сценарії використання	58
3.2.6	Версіонування коду	61
3.2.7	Безперервна інтеграція та безперервна доставка/розгортання.....	62
3.2.8	Автоматизація збірки розширення	63
3.3	Формування штучної вибірки за допомогою розширення	65
3.3.1	Формування штучного датасету.....	66
3.3.1	Візуальне порівняння отриманих результатів	67
3.3.2	Формування комбінованого датасету	70
3.4	Тестування точності підрахунку натовпу різної щільності.....	71
3.5	Рекомендації щодо створення штучних датасетів за допомогою редактора Blender	72
3.5.1	Загальні налаштування	72
3.5.2	Застосування текстур та матеріалів	74

3.5.3 Формування штучного датасету	75
Висновки	77
Перелік джерел посилання	79
Додаток А Зображення з класичних розмічених наборів	86
Додаток Б Зображення створені розширенням до Blender	89
Додаток В Інтерфейс налаштування розширення	93

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

КЗ – Комп’ютерний зір

CNN – Convolutional Neural Networks (згорткова нейронна мережа)

NLP – Natural Language Processing (обробки природної мови)

RNN – Recurrent Neural Network (рекурентні нейронні мережі)

SGD – Stochastic Gradient Descent (стохастичний градієнтний спуск)

MCNN – Multi-Scale Convolutional Neural Networks (багатомасштабні згорткові нейронні мережі)

CSRNet – Congested Scene Recognition Convolutional Neural Networks (згорткові нейронні мережі для розпізнавання перевантажених сцен)

GPU – Graphics Process Unit (графічний процесор)

CPU – Central Process Unit (центральний процесор)

IDE – Integrated Development Environment (інтегроване середовище розробки)

ВСТУП

Комп'ютерний зір (КЗ) – це відносно молодий напрямок, спрямований на аналіз та розпізнавання об'єктів і образів у візуальних даних [1-3]. На початку його розвитку у середині 20 століття та наступні десятиліття підходи до розпізнавання, сегментації, детекції зображень будувалися на основі евристичних підходів [4-6]. Методи були математичною інтерпретацією певних особливостей зображення, що дозволяло використовувати їх для таких задач, як розпізнавання об'єктів, відстеження руху та реконструкція 3D-сцен. Значні успіхи для аналізу зображень показували методи на основі різних дескрипторів характерних точок [7-9], методів морфології, задалегідь підібраних масок [10-13]. Такі методи можна віднести до класичних методів комп'ютерного зору. Хоча вже на початку існували нейронні мережі. Для їх використання для рішення реальних задач не вистачало обчислювальних потужностей.

На початку 2000 року та далі КЗ почав використовувати методи машинного навчання. Зокрема, виникли алгоритми, які використовували метод опорних векторів та інші класифікатори для аналізу ознак зображень. Екстракція особливостей зображень виконувалась все ще за допомогою евристики, але навчання дозволило значно підвищити точність.

Справжня революція в КЗ почалася з використання згорткових нейронних мереж у 2012 році, вони дозволили автоматично виділяти особливості з даних, що зменшило залежність від евристичного підходу [14]. Точність класифікації зросла до рівня, порівнянного з людським.

Також, суттєве зростання обчислювальних можливостей повернуло дослідників до продовження вивчення та використання класичних нейронних архітектур, таких як мережа Кохонена [15, 16].

Сучасні системи комбінують глибокі нейронні мережі з іншими технологіями та класичними підходами до аналізу зображень, на основі чого створюються безліч систем для вирішення задач в реальному житті [17-21].

Але значний прогрес останні роки не лише в області КЗ, схожа ситуація в області обробки природної мови (NLP). До появи нейронних мереж в NLP використовувалися також класичні підходи, засновані на правилах і статистиці, наприклад, моделі на основі частот слів (n-грамні моделі), алгоритми із жорсткими правилами, що використовували лексичні і синтаксичні правила, статистичний машинний переклад, де переклади ґрунтувалися на ймовірностях і великих корпусах паралельних текстів. Ці методи не враховували контексту та мали обмежену здатність до генерації тексту.

З середини 2010-х років почали застосовуватися для обробки тексту рекурентні нейронні мережі (RNN). Вони дозволили враховувати порядок слів і зберігати інформацію про попередні слова в послідовності та привели до суттєвих успіхів в області машинного перекладу та генерація тексту (зокрема, chatbot). У 2017 році відбувся реальний прорив, коли з'явилася модель Transformer, яка замінила RNN. Головною ідеєю стало використання механізму уваги (attention), який дозволив враховувати контекст кожного слова у всьому тексті. Моделей BERT, GPT та їх похідні привели до суттєвого покращення якості машинного перекладу, відповідей на питання, текстових пошукових систем. Сучасні моделі, такі як Gemini on Vertex AI від Google, GPTs від Open AI, Claude від Anthropic, виконують складні задачі, як спілкування, програмування, творчість та інше. На їх основі створюється програмні додатки в різних галузях [22-32].

Важливою частиною розвитку галузі КЗ та NLP є створення спеціалізованих наборів даних, що дозволяють тренувати та перевіряти моделі, здатні виконувати задачі комп'ютерного бачення.

Створення датасетів у КЗ набуло особливої популярності наприкінці 2010-х років, коли обчислювальні потужності та технології глибинного навчання почали дозволяти обробляти великі обсяги даних з високою точністю. Основними задачами у сфері КЗ, де потрібні датасети, це розпізнавання, сегментація, детекція, відстеження руху та інше. Для створення

великих датасетів існує проблема автоматизація розмітки, оскільки ручна розмітка є тривалим та затратним процесом. Автоматизація дозволяє суттєво скоротити цей час та мінімізувати похибки, створюючи можливість для більшого обсягу анотаційних даних при обмежених ресурсах. Використання інструментів автоматизації у візуальних редакторах, таких як Blender, є особливо ефективним підходом [33, 34].

Також, створення синтетичних датасетів стало важливою частиною галузі, дозволяючи генерувати високоякісні зображення та їх розмітку для навчання моделей без потреби у великих обсягах реальних даних. Такі набори можуть імітувати різні сценарії, включаючи освітлення, кути огляду, складні щільності натовпу тощо. Синтетичні дані надають дослідникам та розробникам нові можливості для тестування моделей у різноманітних умовах.

Задача створення розмічених наборів даних для підрахунку людей у натовпі є особливо актуальною в контексті комп'ютерного бачення. Підрахунок натовпу став важливим аспектом у багатьох сферах, включаючи міське планування, забезпечення громадського порядку та маркетинг. З огляду на складність та затратність збору реальних даних, використання синтетичних зображень для тренування моделей підрахунку людей стало популярним рішенням [34]. Пандемія COVID-19 посилила потребу у контролі за масовими скупченнями людей та моніторингу їх чисельності у громадських місцях. У таких умовах інструменти, що спрощують створення та анотацію синтетичних даних, зокрема для вирішення задач підрахунку людей, є цінними для дослідників та розробників.

1 СУЧАСНИЙ СТАН ПИТАННЯ ПІДРАХУНКУ ЛЮДЕЙ У НАТОВПІ

Підрахунок людей у натовпі дедалі більше стає популярною задачею у галузі комп'ютерного зору. З основних проблем, з якими стикаються фахівці у цій сфері – це різноманіття натовпу (щільний, неоднорідний, рухомий). Вагомий вплив на підрахунок також дають перекриття та кут огляду та рівень освітлення. Всі ці фактори впливають на кінцевий результат підрахунку, але вони не є єдиними. Кожний специфічний спосіб підрахунку має свої нюанси, переваги і недоліки. Задля ретельного висвітлення сучасного становища в цьому питанні, методи будуть розглянуті окремо.

1.1 Підходи для підрахунку людей у натовпі

Одним з найперших примітивних методів підрахунку людей вважається метод Герберт Джейкобса. У 60-х роках, в Каліфорнійському університеті в Берклі, запропонував свій метод підрахунку. Із вікна свого кабінету професор спостерігав за студентами, які збиралися на площі та протестували проти війни у В'єтнамі. Площу було поділено на квадрати, і Джейкобс підраховував кількість студентів у кількох з них, щоб обчислити середню кількість протестувальників на один квадрат, а потім множив це число на загальну кількість квадратів. Він також запропонував основне правило для визначення щільності натовпу: «рідким натовпом» вважається зібрання, де на 0,9 кв. м припадає одна людина, у «щільному натовпі» одна людина займає 0,4 кв. м, а в дуже скупченому натовпі – 0,2 кв. м на одну людину.

З технічних методів підрахунку натовпу варто згадати традиційні методи, які з хронологічної точки зору, є найстарішими. Ранній підхід засновувався на використанні детекторів осіб або частин тіла, таких як голова

чи плечі, але ці методи мали обмежений успіх через проблеми з перекриттям і складними умовами.

Сучасні рішення здебільшого базуються на нейронних мережах, зокрема на архітектурах Convolutional Neural Networks (CNN) та їхніх модифікаціях, таких як Fully Convolutional Networks (FCN) та інші.

Інший популярний підхід полягає у створенні карти щільності натовпу (density map), яка використовується для підрахунку кількості людей на основі локальних рівнів щільності. Цей метод добре працює в умовах високої щільності натовпу.

1.2 Підрахунок на основі розв'язання задачі комп'ютерного зору

Сучасні рішення цієї проблеми базуються на поєднанні класичних методів комп'ютерного зору та технологій глибокого навчання, які використовуються для аналізу відео- або зображень у реальному часі. Ранні підходи до підрахунку людей використовували детекцію об'єктів на основі класичних методів:

- Haar cascades (метод для виявлення облич на основі шаблонів);
- Histogram of Oriented Gradients (HOG) для виявлення людських фігур;
- Deformable Part Models (DPM), що визначають людей на основі розпізнавання окремих частин тіла.

Сучасні рішення для детекції людей використовують методи на основі глибокого навчання нейронних мереж. Найбільш відомі архітектури для детекції об'єктів:

- Faster R-CNN: методика регіональних згорткових нейронних мереж для точного виявлення об'єктів, включаючи людей;

- YOLO (You Only Look Once): ефективний метод, який забезпечує швидке і точне виявлення об'єктів на зображеннях;
- SSD (Single Shot MultiBox Detector): ще одна популярна модель для швидкого детектування.

Ці алгоритми використовують навчання на великих наборах даних для виявлення людських фігур у натовпі. Важливою перевагою є здатність до реального часу обробки, що дозволяє використовувати їх для відеоспостереження.

Оцінка щільності натовпу (Crowd Density Estimation) – це підхід, при якому не вимагається прямого розпізнавання кожної окремої людини, що є корисним для щільних натовпів, де об'єкти можуть перекривати один одного. Алгоритм генерує карту, яка показує рівень щільності людей на кожному пікселі зображення або відео. Основна ідея полягає в тому, щоб перетворити задачу детекції людей на задачу регресії – модель передбачає кількість людей у кожному фрагменті зображення. Моделі на основі CNN використовуються для генерації таких карт, серед них популярні:

- MCNN (Multi-column CNN): використовує кілька CNN, кожна з яких спеціалізується на різній щільності натовпу [35];
- CSRNet (Convolutional Neural Networks with Dilated Convolutions) [36, 37]: ефективно визначає щільність у натовпах різної щільності, використовуючи спеціальні згорткові шари для обробки великих масштабів зображення.

Окрім методів, які працюють переважно з статичним кадрами, існують методи, що аналізують потік зображень і націлені на відстеження об'єктів. Вони використовуються для підрахунку людей у відеопотоках. Відстеження дозволяє виявляти рухомі об'єкти на основі послідовних кадрів, що дозволяє точно підрахувати кількість унікальних об'єктів. Optical Flow, наприклад, використовує аналіз руху пікселів між послідовними кадрами для оцінки траєкторії руху об'єктів, таких як люди. Більш сучасні підходи використовують рекурентні нейронні мережі (RNN), наприклад LSTM (Long

Short-Term Memory), які можуть зберігати інформацію про рух об'єктів у часі, що дозволяє враховувати навіть складні траєкторії.

Свою нішу зайняли й комбіновані методи, які мають характеристика як відстеження, так і підрахунку, що дозволяє ефективно вирішувати задачі комп'ютерного зору за складних умов.

1.2.1 Підрахунок людей на основі CSRNet

CSRNet (Convolutional Neural Networks with Dilated Convolutions for Crowd Counting) — це один із найефективніших підходів до підрахунку людей у натовпі на основі карт щільності (density maps) [35-37]. Модель CSRNet була представлена у 2018 році і здобула популярність завдяки своїй здатності працювати з дуже щільними натовпами, де класичні методи та навіть інші нейронні мережі могли зазнавати труднощів.

CSRNet використовує згорткові шари для вилучення особливостей (features) із вхідного зображення. Ці шари допомагають моделі розпізнавати людей у натовпі та забезпечують початкову обробку зображення.

Ключовою особливістю CSRNet є використання розширених згортки (або dilated convolutions). Вони дозволяють моделі охоплювати більшу область зображення без збільшення кількості параметрів і зберігають високу роздільну здатність вихідних карт щільності.

Розширені згортки допомагають вирішити проблему, пов'язану зі зміною масштабу та щільністю натовпу: вони ефективно захоплюють як локальні, так і глобальні патерни в зображенні.

Замість прямого виявлення кожної людини, CSRNet генерує карти щільності, де кожен піксель відповідає кількості людей у відповідній частині зображення. Це дозволяє моделі адаптуватися до різних рівнів щільності: від рідкісних скупчень людей до дуже щільних натовпів. CSRNet складається з двох основних частин:

- фронтенд базується на першій частині добре відомої мережі VGG-16. Це вже натренована мережа, яка використовується для вилучення початкових ознак зображення. Вибір VGG-16 пояснюється її здатністю вилучати багаті особливості з вхідних даних, зберігаючи ефективність навчання;
- бекенд складається з декількох шарів розширених згорток, що дозволяють моделі обробляти зображення великого масштабу та точно оцінювати щільність натовпу.

Розширені згортки використовують коефіцієнт розширення [37-40], що дає змогу моделі бачити ширший контекст навколо кожної точки на зображенні без збільшення кількості параметрів. Це особливо важливо для підрахунку людей у натовпах з нерівномірною щільністю.

1.2.2 Підрахунок людей на основі MCNN

MCNN (Multi-Column Convolutional Neural Network) — це одна з перших архітектур глибокого навчання, розроблена для підрахунку людей у натовпі на основі карт щільності [35]. Вона була представлена у 2016 році і стала основою для подальших удосконалень у задачі підрахунку людей за допомогою комп'ютерного зору.

Основна мета MCNN — вирішити проблему великої варіації щільності натовпу в зображеннях. Ця варіація може змінюватися від рідкісних груп людей до дуже щільних скупчень, що робить задачу підрахунку надзвичайно складною. MCNN намагається вирішити цю проблему, використовуючи різні шари з різними розмірами ядер (фільтрів) для охоплення об'єктів на різних масштабах.

Головна особливість MCNN полягає в наявності кількох колон (гілок) нейронної мережі, кожна з яких відповідає за обробку інформації на різних масштабах. Ці колони дозволяють моделі працювати як з зображеннями, де

об'єкти розташовані далеко (тобто мають малий розмір на зображенні), так і з зображеннями, де об'єкти дуже близькі або перекриваються.

MCNN використовує підхід, заснований на картах щільності, де модель намагається передбачити карту, на якій кожен піксель представляє кількість людей у певній області зображення. Для тренування моделі використовуються спеціальні карти щільності, створені на основі анотацій координат людей.

Мережа складається з трьох основних колон, кожна з яких має власні згорткові шари з різними розмірами фільтрів:

- дрібні фільтри (7×7), які допомагають моделі захоплювати деталі з зображень, де люди або об'єкти з'являються на мікрорівні, тобто мають великий масштаб на зображенні;
- фільтри середнього розміру (9×9), які допомагають захоплювати інформацію зі зображень середнього масштабу;
- великі фільтри (13×13), що дозволяють обробляти інформацію з мало масштабних об'єктів, наприклад, людей, які перебувають на великій відстані та виглядають маленькими на зображенні.

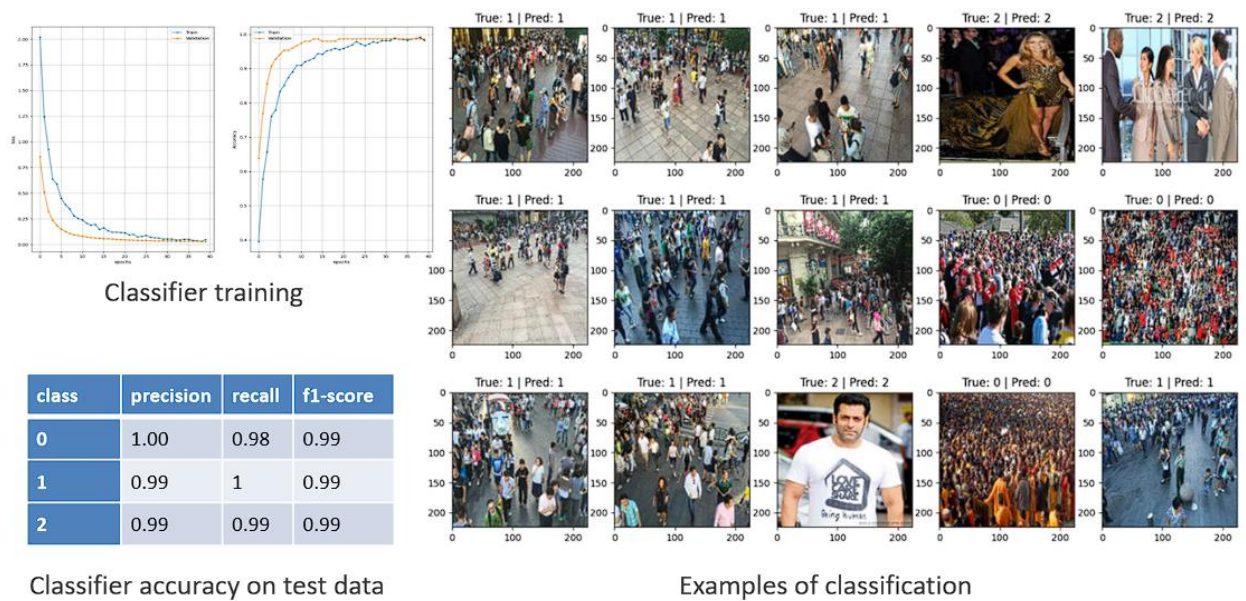
MCNN зробила значний внесок у розвиток методів підрахунку людей у натовпі. Її мультиколонна архітектура дозволяє моделі ефективно працювати з варіативністю масштабу та щільності людей у натовпі. Хоча більш нові моделі, такі як CSRNet, можуть перевершувати MCNN за точністю, MCNN все ще залишається важливим інструментом для розуміння основ підрахунку людей за допомогою комп'ютерного зору та використання карт щільності.

1.2.3 Підхід на основі попередньої класифікації

Можливим рішенням питання підрахунку людей для різних зображень можуть бути комбіновані методи, ті що поєднують декілька методів. Ускладнюючи систему можливо додати проміжну ланку, яка б вже перенаправляла відповідне зображення до методу, який найкраще його

обробить. Таким чином, вимальовується мережа-класифікатор, яка розрізняє класи зображення і перенаправляє їх на методи, які найкраще їх оброблять [42-45].

На рисунку 1.1 а) подано результати графіків навчання та метрики мережі класифікатора, а на рисунку 1.1 б) показано зображення, які були віднесені до певного класу.



а)

б)

Рисунок 1.1 – Навчання мережі класифікатора зображень:

- а) графіки навчання та значення метрик точності;
- б) приклади зображень, що були класифіковані

Така мережа потребує окремого навчання і вибірки власних наборів даних з чітким розподілом по класам [34].

1.2.4 Сучасні підходи та їхня точність

Методи підрахунку натовпу на основі згорткових нейронних мереж постійно вдосконалюються і перенавчаються. Висока конкуренція народжує

більш ефективні і точні мережі. APGCC, PSL-Net та FGENet – це мережі, представлені у 2024 році, однак кожна модель підходить до задачі підрахунку людей по-своєму [46-50].

APGCC використовує так зване «допоміжне точкове керування» (auxiliary point guidance), яке покращує точність підрахунку і локалізації людей у натовпі [51]. Основна ідея – визначення центрів індивідуальних людей за допомогою додаткової допоміжної інформації. Цей підхід знижує похибку в підрахунку і забезпечує високу деталізацію, особливо в випадках, коли особи в натовпі тісно розташовані одна до одної. Модель вважається дуже точною, що підтверджується найнижчим показником середньої абсолютної похибки MAE – 48.8, а також мінімальною середньоквадратичною помилкою MSE – 76.7.

APGCC відрізняється фокусом на точковій локалізації (point-based localization) через введення допоміжної інформації, що є ключовою інновацією порівняно з іншими моделями [51, 52].

PSL-Net працює на основі так званого «псевдоквадратного маркування» (pseudo square label), що дозволяє поєднати підрахунок людей із їх локалізацією. Ідея полягає в поданні місць присутності людей у натовпі у вигляді квадратів, а не точок, що надає певну гнучкість і адаптивність у визначенні областей для індивідуальних осіб. Це також спрощує обробку натовпу із змінною щільністю. Для підрахунку PSL-Net використовує архітектуру VGG для вилучення ознак, що підвищує стабільність та ефективність моделі. Вона показала MAE у 49.9 та RMSE у 77.6 [53].

PSL-Net відрізняється від інших моделей використанням квадратно-орієнтованого підходу до локалізації, що дозволяє спрощувати обробку щільних натовпів і розподіляти розташування людей ширшими областями.

FGENet (Fine-Grained Extraction Network) акцентує увагу на «тонкому вилученні ознак» (fine-grained feature extraction), що дозволяє точно працювати в умовах дуже високої щільності натовпу. Основна мета моделі — підвищити точність вилучення ознак окремих людей в дуже насичених сценах. Це

досягається завдяки більш детальному аналізу зображень і залученню багаторівневих ознак. FGNet показує MAE 51.66 та MSE 85.00, що є менш точним результатом порівняно з іншими моделями, але підходить для особливо насичених сцен [54].

FGNet має спеціальну архітектуру, що підходить для високощільних натовпів, і є більш адаптованою до обробки зображень із численними перешкодами.

1.3 Огляд датасетів, що містять натовп

У області розв'язання задачі підрахунку людей у натовпі широко відомі такі датасети ShanghaiTech [55], UCF-QNRF [56], JHU-CROWD++ [57], UCF-CC-50 [58]. Згідно з наведеними джерелами [59], де зібрані наукові статті та програмний код, щоб можна було повторити експерименти, а також порівнюються результати запропонованих методів для розв'язання конкретної задачі, найбільша кількість експериментів у сфері підрахунку натовпу спирається на датасет ShanghaiTech [55].

ShanghaiTech – це масштабний датасет для підрахунку людей у натовпі, який містить 1198 анотованих зображень натовпу. Датасет розділений на дві частини: Частина А, яка включає 482 зображення, та Частина В, яка включає 716 зображень. Частина А поділена на тренувальну та тестову підмножини з 300 та 182 зображень відповідно. Частина В також поділена на тренувальну та тестову підмножини з 400 та 316 зображень відповідно.

Кожна людина на зображеннях позначена точкою, яка знаходиться близько до центру голови. Загалом датасет включає 330,165 анотованих людей. Зображення з Частини А було зібрано з Інтернету, а зображення з Частини В – на живих вулицях Шанхаю.

Хоча цей датасет вважається класичним для питань навчання мереж на підрахунок натовпу. Але проаналізувавши зображення, можна дійти висновку

що він не є збалансованим. Таблиця 1.1 наочно демонструє, що деякі проміжки перенасичені зображеннями – проміжок з 50-99 має 109 зображень, в той час як заповнені зображення на 500-549 людей налічують лише 3 зображення (рис. 1.2).

Таблиця 1.1 – Кількість зображень відносно певних сегментів

Segment	0 - 49	50 - 99	100 - 149	150 - 199	200-249	250-299	300-349	350-399	400-449	450-499	500-549
Quantity	60	109	51	46	20	13	7	1	2	4	3

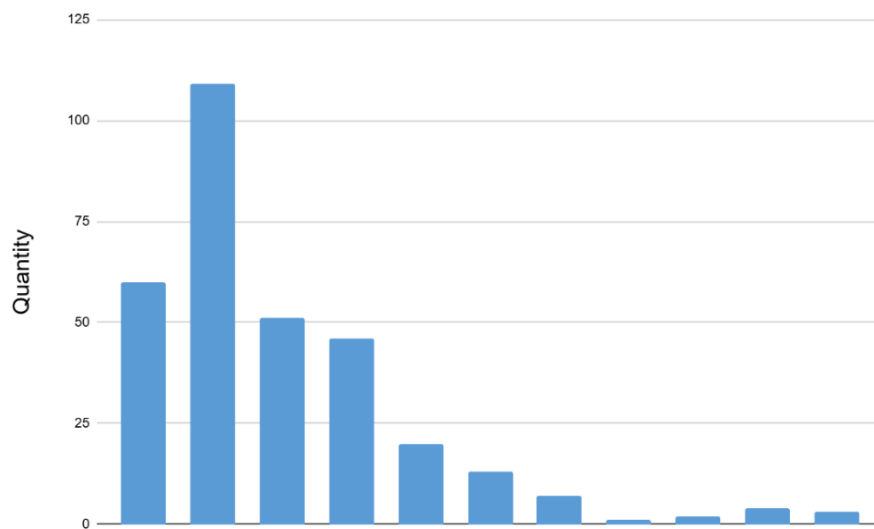


Рисунок 1.2 – Розподіл зображень по діапазонам

Як відомо, вибірка даних для навчання напряму впливає на роботу моделі після навчання. Тому вирішення питання незбалансованості потенційно може допомогти покращити вже існуючі методи.

1.4 Можливості створення штучних датасетів

Штучно створені датасети дозволяють точно контролювати розподіл даних, що є корисним для задач, які вимагають рівномірного представлення

всіх класів. Наприклад, у медичних дослідженнях можна імітувати рівномірний розподіл хворих із різними стадіями захворювання, що рідко трапляється у реальних даних.

При створенні штучного датасету можна точно задавати та контролювати анотацію, зокрема розташування, розмір, орієнтацію та інші властивості об'єктів. Це значно спрощує процес анотації та забезпечує високу точність маркування, що важливо для завдань комп'ютерного зору.

Поєднання синтетичних даних із реальними допомогло би поліпшити ефективність моделей, особливо для моделей, схильних до перенавчання. Синтетичні дані можуть слугувати «домішковими» для підвищення загальної кількості прикладів і покращення навчального процесу.

Штучні датасети можна створити за допомогою різних підходів, наприклад:

- генерування нових штучних зображень за допомогою нейронних мереж, наприклад, Generative Adversarial Networks (GANs), Variational Autoencoders (VAEs), Denoising Diffusion Models, Transformer-based Models, Neural Radiance Fields (NeRF);

- створення зображень за допомогою графічних редакторів, наприклад, Blender, та інструментів для розробки ігор, наприклад, Unity, Unreal Engine. Дані редактори та платформи дають змогу створювати реалістичні 3D-сцени і контролювати всі аспекти сцени, включно з освітленням, положенням об'єктів, кутами огляду та багато іншого;

- збагачення існуючих датасетів на основі аугментації. Наприклад, Python-бібліотеки, такі як Albumentations, imgaug, Augmentor і TensorFlow Image, можуть бути корисні для створення штучних датасетів через аугментацію. Ці бібліотеки застосовуються для модифікації зображень, щоб підвищити їхнє розмаїття і поліпшити узагальнюючу здатність моделі.

1.4.1 Формування датасетів за допомогою нейронних мереж

Генеративні моделі, такі як GAN (генеративно-змагальні мережі), VAE (автокодери з варіаційними властивостями) та Diffusion Models, є інструментами для створення реалістичних та деталізованих зображень. Вони дозволяють моделювати надзвичайно складні сцени та природні текстури, що робить штучні дані максимально наближеними до реальних. Вибір нейронної мережі для генерації зображень залежить від мети створення датасету, потрібного рівня деталізації, а також наявності конкретних умов, таких як задана кількість об'єктів, певні пози, освітлення тощо. GAN та Diffusion-моделі (наприклад, StyleGAN, BigGAN та Latent Diffusion Models) залишаються одними з найпопулярніших і найпотужніших інструментів для створення синтетичних зображень для завдань машинного навчання.

1.4.2 Програмне забезпечення для розмітки датасетів

Для якісної підготовки датасетів у задачах комп'ютерного зору використовуються спеціалізовані інструменти для розмітки, такі як RoboFlow, LabelImg та CVAT. RoboFlow – це веб-платформа, що забезпечує зручне середовище для розмітки, аугментації, експорту в різні формати та навіть попереднього навчання моделей. Завдяки функціям автоматичної анотації та підтримці хмарного зберігання, RoboFlow дозволяє значно оптимізувати процес підготовки даних. LabelImg – легкий та відкритий інструмент для розмітки зображень, що надає можливості для створення рамок навколо об'єктів і підтримує формати PASCAL VOC та YOLO. Він є особливо корисним для задач об'єктного детектування. CVAT (Computer Vision Annotation Tool) – також інструмент з відкритим кодом, орієнтований на розмітку як зображень, так і відео. CVAT забезпечує розмітку різних типів, таких як рамки, полігони, маски, а також підтримує трекінг об'єктів у відео,

що робить його зручним для роботи з великими обсягами даних у командних проєктах.

1.4.3 Редактор Blender

Редактор Blender є потужним інструментом для створення синтетичних датасетів у задачах комп'ютерного зору, зокрема для моделювання натовпу та автоматичної розмітки. Завдяки можливості генерації тривимірних сцен та скриптовій підтримці через Python API, Blender дозволяє створювати масштабовані сцени з натовпом, контролюючи положення, орієнтацію, щільність та інші характеристики об'єктів. За допомогою процедурного моделювання можна налаштувати різні умови освітлення, пози, різноманітність зовнішності людей та динамічні сценарії руху. Blender автоматизує процес розмітки, генеруючи координати, рамки, маски чи ключові точки для кожного об'єкта в сцені, що особливо цінно для задач детектування, сегментації та підрахунку людей у натовпі. Використання Blender у генерації синтетичних датасетів дозволяє отримати великий обсяг реалістичних даних із високою точністю анотацій, що знижує потребу в ручній розмітці.

1.5 Сучасне програмне забезпечення для вирішення завдань комп'ютерного зору

Сучасне ПЗ для комп'ютерного зору надає різноманітні можливості для обробки зображень та відео, від детектування об'єктів до високоточної сегментації та трекінгу. Бібліотеки OpenCV, PyTorch та TensorFlow підходять для розробки кастомних моделей, тоді як спеціалізовані інструменти, як-от

YOLO, Detectron2 та DeepStream SDK, оптимізовані для роботи з великими потоками даних та високої продуктивності в реальному часі.

1.6 Постановка задачі дослідження

Проблема підрахунку кількості людей у натовпі є важливою для різних сфер життя. Незважаючи на досягнуті успіхи у розв'язанні цієї задачі, існує потреба подальших досліджень у цій галузі, особливо актуальною є адаптація відомих методів і розробка нових для розв'язання задачі підрахунку в реальному часі та розробка реальних сервісів і застосунків.

Більшість сучасних датасетів (наприклад, ShanghaiTech, UCF-QNRF, JHU-CROWD) прагнуть до балансу за розміром зображень, варіативністю сцен, якістю, освітленням та щільністю натовпу. Проте в деяких датасетах є нерівномірний розподіл щільності натовпу або зображення з обмеженою різноманітністю умов освітлення. Важливо враховувати якість зображень і наявність сцен із нічним освітленням, оскільки такі варіанти можуть суттєво вплинути на точність моделей і потребують ретельного підбору методу розмітки та обробки.

Сучасні моделі, як-от CSRNet, CAN, і SANet, демонструють різну точність за метриками MAE (середня абсолютна похибка) і RMSE (середньоквадратична похибка) залежно від щільності натовпу. Високо щільні зони часто призводять до вищої похибки через перекривання об'єктів та складність у детектуванні меж між людьми.

В роботі пропонується для вирішення задачі підрахунку натовпу спробувати використати графічний редактор Blender для створення штучних датасетів або для збагачення існуючих реальних датасетів для подальшого навчання моделей з метою підвищення їх точності і використати. Для цього необхідно створити плагін для генерації сценаріїв із натовпом, який дозволить налаштовувати густоту, розміщення, освітлення об'єктів. Це допоможе

отримати контрольовану вибірку для навчання і тестування моделей, а також забезпечить автоматичну розмітку даних.

Для підтвердження можливості використання для навчання згенерованих зображень необхідно порівняти точність підрахунку вже навченими моделями на реальних і синтетичних зображеннях. Це також допоможе виділити особливості, що потребують вдосконалення в синтетичних даних.

Для формування штучного датасету варто розробити рекомендації щодо щільності об'єктів, варіації освітлення та текстур, враховуючи, як ці фактори впливають на точність розпізнавання моделей.

Збалансованість сцен та налаштування умов, подібних до реальних, таких як нічне освітлення чи різка зміна кута огляду, підвищать ефективність використання синтетичних даних у практичних застосунках.

Об'єктом роботи є створення розмічених наборів даних для підрахунку кількості людей у натовпі шляхом генерування синтетичних зображень.

Метою роботи є дослідження питання створення розмічених наборів даних за допомогою розробленого модуля для редактора Blender з метою вирішення задачі підрахунку натовпу. Особливу увагу приділено генерації високоякісних синтетичних зображень натовпу із детальним налаштуванням умов освітлення, щільності натовпу та перспективи, а також точності й зручності анотації об'єктів на зображеннях.

Задля досягнення цієї мети необхідно:

- проаналізувати особливості сучасних датасетів, що містять натовп (збалансованість, вміст зображень із поганою якістю, нічним освітленням);
- проаналізувати якість роботи сучасних методів для натовпу різної щільності;
- спроектувати, розробити та протестувати plugin для редактора Blender для створення штучного датасету;

- порівняти значення метрик точності підрахунку людей для натовпу різної щільності для датасетів із реальними та штучними зображеннями для сучасних моделей;
- розробити рекомендації щодо формування штучних датасетів на основі генерації зображень у Blender.

2 АНАЛІЗ МЕТОДІВ І СТРУКТУР МОДЕЛЕЙ ПІДРАХУНКУ ЛЮДЕЙ У НАТОВПІ ТА АНАЛІЗ ТОЧНОСТІ ПІДРАХУНКУ ДЛЯ РІЗНОГО НАТОВПУ РІЗНОЇ ЩІЛЬНОСТІ

2.1 Метрики для вимірювання точності підрахунку людей у натовпі

Для оцінки ефективності алгоритмів підрахунку людей у натовпі використовуються різноманітні метрики, які дозволяють детально проаналізувати їхню точність і продуктивність. Ці метрики відіграють ключову роль у визначенні того, наскільки добре модель здатна виконувати свої функції, особливо в умовах, коли сцени мають підвищену складність. Складні сцени часто включають високу щільність людей, перекриття між об'єктами, різні ракурси та варіативність освітлення, що створює додаткові виклики для моделей підрахунку.

Використання правильних метрик дозволяє розробникам не лише оцінити загальну точність алгоритму, але й зрозуміти його поведінку в різних умовах. Це важливо для виявлення слабких місць моделі, наприклад, у випадках з дуже щільними натовпами, де окремі об'єкти можуть бути важко розрізнити. Відповідні метрики допомагають ідентифікувати ці проблеми та надати орієнтири для подальшого вдосконалення алгоритму. Ось чому об'єктивне і ґрунтовне вимірювання точності підрахунку людей у натовпі стає критично важливим етапом у розробці та вдосконаленні таких систем.

Mean Absolute Error (MAE) – це метрика, яка обчислює середнє абсолютне значення похибки між реальною кількістю людей і прогнозованим значенням. Вона є однією з найбільш поширених метрик для оцінки точності моделей підрахунку, оскільки надає пряме уявлення про те, наскільки модель помиляється у своїх передбаченнях.

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - x_i|, \quad (2.1)$$

де n – кількість зображень;

y_i – кількість людей підрахована мережею;

x_i – фактична кількість людей.

Mean Squared Error (MSE) – це метрика, яка обчислює середнє значення квадратів різниці між реальним значенням (кількістю людей) та прогнозованим значенням моделі. Вона дозволяє оцінити точність алгоритму, враховуючи не лише середню похибку, а й величину відхилень, які суттєво впливають на результат.

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - x_i)^2. \quad (2.2)$$

У підрахунку людей у натовпі MSE використовується як додатковий інструмент разом із MAE, що дозволяє краще зрозуміти як середні, так і крайні значення похибок. Це дає змогу розробникам створювати більш точні та надійні алгоритми для складних сцен із високою щільністю чи перекриттям об'єктів.

Root Mean Squared Error (RMSE) – це метрика, яка є квадратним коренем з MSE. RMSE забезпечує інтуїтивно зрозумілу оцінку похибки, оскільки повертає її в тих самих одиницях вимірювання, що й реальні значення (наприклад, кількість людей). Це робить RMSE корисним для порівняння і більш практичним у багатьох випадках, особливо під час аналізу моделей підрахунку людей у натовпі.

Хоча RMSE надає інформацію про середнє відхилення, вона, як і MSE, може бути не ідеальною для розуміння помилок у реальних даних із нерівномірним розподілом, де маленькі та великі похибки мають різну важливість.

Precision та Recall – це ключові метрики, що застосовуються для оцінки ефективності алгоритмів виявлення, включно з задачами підрахунку людей у натовпі. Вони дозволяють виміряти різні аспекти роботи моделей, зокрема

їхню влучність і повноту, що є важливими для аналізу якості виявлення об'єктів у складних сценах.

Precision (точність) вимірює, наскільки правильно алгоритм визначає людей серед усіх виявлених об'єктів. Високе значення Precision означає, що алгоритм здебільшого правильно класифікує об'єкти як людей, мінімізуючи кількість помилкових спрацьовувань (false positives). Графічно їх можна показати, як на зображенні нижче (рис 2.1).

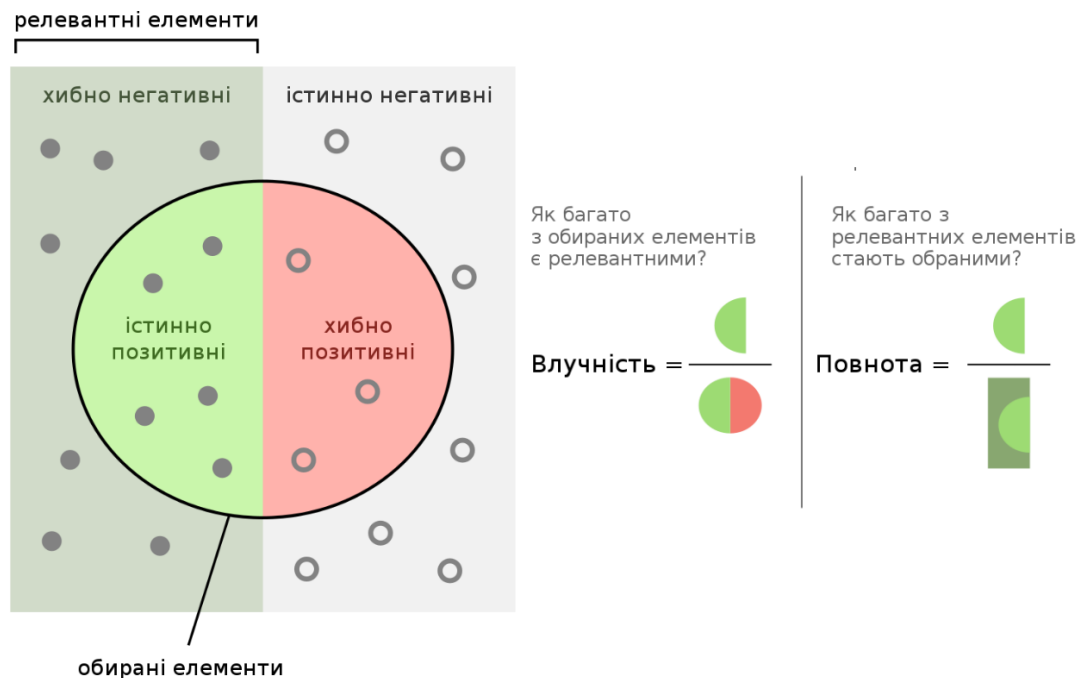


Рисунок 2.1 – Метрики Precision та Recall

F1-показник є важливою метрикою, яка об'єднує Precision (точність) і Recall (повноту) в одне число, щоб забезпечити збалансовану оцінку роботи алгоритму. Його особливістю є використання гармонійного середнього, яке зменшує вплив великих диспропорцій між точністю і повнотою. У задачах підрахунку людей це допомагає оцінити загальну якість алгоритму, зокрема у складних умовах із високою щільністю натовпу чи перекриттям об'єктів.

$$F_1 = \frac{2}{\text{Recall}^{-1} + \text{Precision}^{-1}}. \quad (2.3)$$

У багатьох задачах, включаючи підрахунок людей у натовпі, Precision і Recall мають різні пріоритети залежно від контексту. Наприклад:

- у критичних сценаріях, таких як безпека чи надзвичайні ситуації, важливо не пропустити жодного об'єкта (високий Recall);
- у задачах із великим обсягом шумових даних може бути важливіше мінімізувати хибні спрацювання (високий Precision);
- F1-показник дозволяє досягти компромісу між цими метриками, забезпечуючи загальну оцінку ефективності алгоритму.

Mean Absolute Percentage Error (MAPE) – це метрика, яка вимірює середнє абсолютне значення відсоткової похибки між прогнозованою кількістю людей і реальною. Вона виражається у відсотках, що робить її інтуїтивно зрозумілою для оцінки точності алгоритмів у задачах підрахунку людей. MAPE часто використовується для порівняння моделей, оскільки вона показує, наскільки велика відносна помилка в середньому

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^n \frac{|x_i - y_i|}{x_i}. \quad (2.4)$$

2.2 Математична модель найбільш точних сучасних методів підрахунку людей

У сфері аналізу натовпу існують дві основні методології: підходи на основі мап і підходи на основі локалізації. Методи на основі карт, що використовують ядра Гаусса для візуалізації карт щільності, ефективно надають моделям критично важливу інформацією для вивчення щільності натовпу.

Незважаючи на їхню здатність досягати локалізації за допомогою додаткових конструкцій, вони все ще стикаються з такими проблемами, як

перекриття карт у щільних ділянках зображення і необхідність багато масштабного представлення.

2.2.1 APGCC

Математична модель методу APGCC (Auxiliary Point Guidance Crowd Counting) фокусується на вирішенні проблеми нестабільності у відповідності точок (proposal-target matching) в методах підрахунку натовпу [52] (рис 2.2). Основна ідея полягає в покращенні стабільності через додаткове керування точками. Це включає алгоритм, який спрямовує пропозиції точок до цільових точок через оптимізацію функції втрат, що мінімізує різницю між очікуваними і фактичними координатами точок.

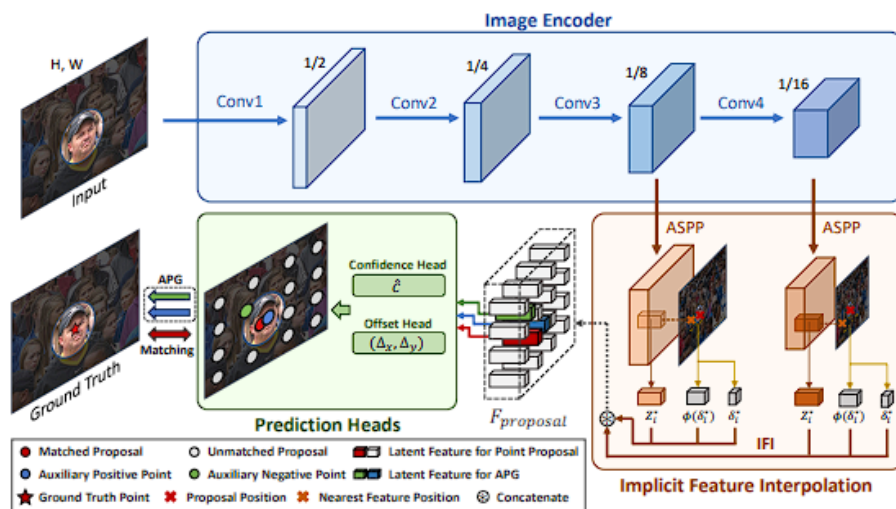


Рисунок 2.2 – Архітектура APGCC

У APGCC використовуються координати точок, які позначають місця, де знаходяться люди. Ці точки надають моделі додаткову інформацію про місцезнаходження об'єктів, що допомагає алгоритму краще зрозуміти розподіл людей у натовпі. Завдяки цьому метод стає менш залежним від розмитих або неоднозначних сигналів із зображення.

Використання точкових підказок допомагає моделі швидше та ефективніше адаптуватися до складних сцен, таких як високо щільні натовпи або області з поганою видимістю. Це дозволяє зменшити кількість хибних виявлень та пропусків.

Метод забезпечує більш точне формування щільнісних карт (density maps), які є основним інструментом у підрахунку людей. Завдяки допоміжним точкам, ці карти стають деталізованішими та краще відображають реальний розподіл людей.

APGCC може бути інтегрований у більшість існуючих моделей глибокого навчання для підрахунку людей. Це забезпечує гнучкість та адаптивність підходу до різних наборів даних і типів сцен.

2.2.2 CLIP-EBC

CLIP-EBC (Contrastive Language-Image Pre-training with Enhanced Blockwise Classification) — це метод для підвищення точності підрахунку натовпу за допомогою можливостей CLIP у розпізнаванні зображень, орієнтований на специфічні виклики задач щільності оцінки. Модель розроблена для ефективного розв'язання задач підрахунку, що базуються на інтервальних цілих числах, а не на безперервних значеннях, з якими зазвичай працює CLIP [53] (рис 2.3).

Enhanced Blockwise Classification (EBC): Замість традиційних методів, що використовують оцінки на рівні пікселів, EBC застосовує блокову класифікацію. Це дозволяє поділити зображення на блоки та призначити кожному блоку певне значення кількості людей. Таким чином, зменшуються помилки через шум міток і забезпечується точніше оцінювання, зосереджене на блоках, що є стабільнішими, ніж окремі пікселі.

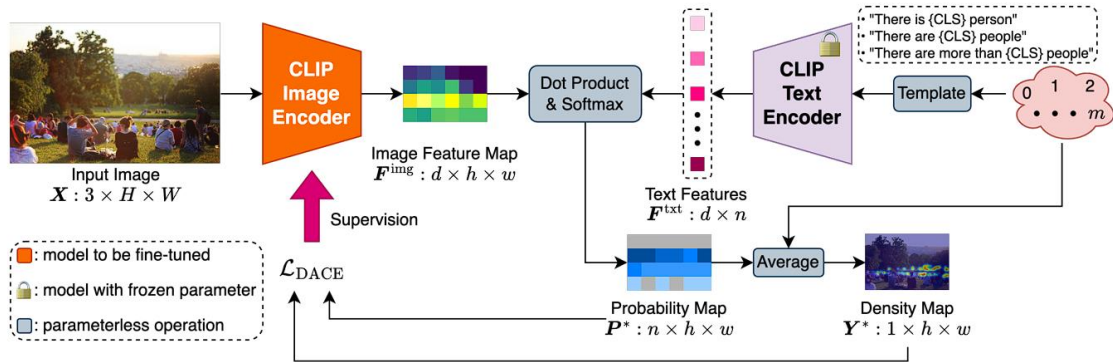


Рисунок 2.3 – Архітектура CLIP-EVC

Інтервальні бінарні значення (Integer-Valued Bins): EVC використовує цілі значення в межах інтервалів (бінів), що допомагає мінімізувати невизначеність на межах між ними. Це знижує ймовірність класифікаційних помилок, які виникають через розмитість та перекриття меж, зокрема в задачах із високою щільністю людей.

Втрата регресії на картах щільності (Regression Loss on Density Maps): Окрім блокової класифікації, CLIP-EVC додає функцію втрат, яка враховує карти щільності для більш точної оцінки розподілу натовпу на зображенні. Це робить модель більш відповідною до підрахунку натовпу, де важливими є не лише кількість, але й розподіл людей.

2.2.3 P2PNet

P2PNet (Point-to-Point Network) є методикою для підрахунку натовпу, що акцентує увагу на прямій регресії координат точок в натовпі, замість використання традиційних карт щільності. Досягається це конкретною архітектурою, яка представлена на рисунку 2.4 [54].

Цей підхід підходить для розв'язання проблеми точного локалізування та підрахунку людей у складних сценах, де натовп може мати різну щільність.

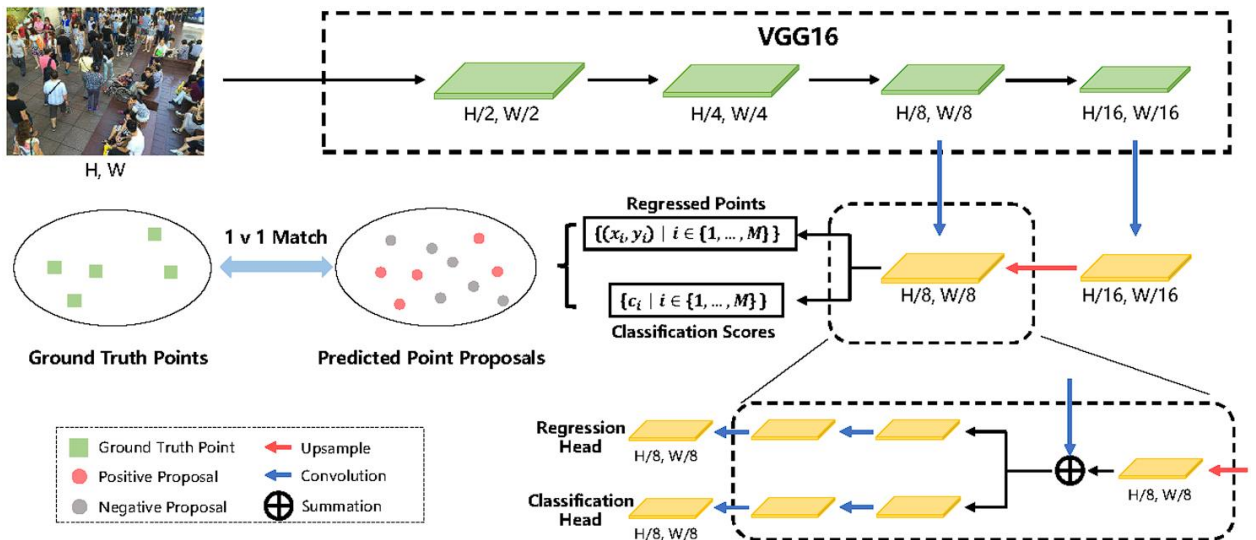


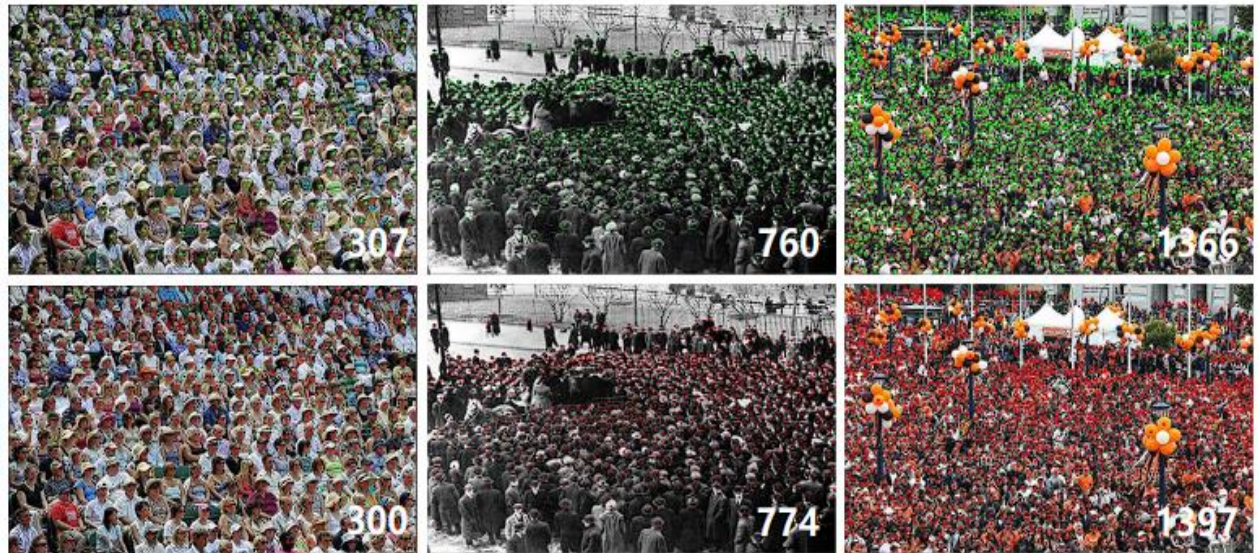
Рисунок 2.4 – Архітектура P2PNet

Пряма регресія координат: P2PNet здійснює регресію координат для кожної особи в натовпі, що дозволяє моделі напряму визначати позиції точок без проміжного етапу перетворення в карти щільності. Це допомагає зменшити вплив шуму, що може виникати в процесі згладжування щільності.

Модель P2PNet застосовує нейронні мережі для виявлення і обробки цих точкових пар (рис 2.5). Використовуючи архітектуру, яка орієнтується на визначення зв'язків між точками людей, вона може ефективно працювати навіть у дуже щільних сценах. Цей підхід дозволяє моделі визначати як індивідуальні об'єкти, так і їхні зв'язки в контексті загальної структури натовпу.

Зв'язок між точками: метод впроваджує мережеву архітектуру, яка обробляє входи як набори точок, і має здатність оцінювати зв'язки між ними, що підвищує точність прогнозування в умовах, коли об'єкти можуть перекриватися або бути частково закритими.

Використання інформації про сусідні точки: P2PNet використовує інформацію про сусідні точки для підвищення точності підрахунку. Залучення контексту з сусідніх координат дозволяє моделі краще розуміти, як людина може бути розташована відносно інших, зменшуючи ймовірність помилок у випадках, коли особи знаходяться близько одна до одної.



а)

б)

в)

Рисунок 2.5 – Приклади зображень, оброблених мережею
 а) мала наповненість; б) середня наповненість; в) велика наповненість

R2PNet є потужним методом для підрахунку людей у натовпі, оскільки зосереджується на точкових парах, що дозволяє точніше оцінювати структуру натовпу і знижувати вплив перекриття. Цей підхід є перспективним у застосуваннях, де важлива точність і здатність обробляти складні сцени з високою щільністю.

2.3 Аналіз особливостей найбільш популярних датасетів

Аналіз особливостей найбільш популярних датасетів для підрахунку натовпу є важливим етапом у розробці та вдосконаленні моделей комп'ютерного зору. Розглянемо кілька відомих датасетів, які використовуються для навчання моделей у цій області.

2.3.1 Загальні характеристики датасетів ShanghaiTech, UCF-QNRF, JHU-CROWD++, UCF-CC-50

Набір даних ShanghaiTech – це великомасштабний набір даних для підрахунку натовпу [55]. Він складається з 1198 анотованих зображень натовпу. Набір даних розділений на дві частини: частина А містить 482 зображення, а частина В – 716 зображень. Частина А розділена на навчальну та тестову підмножини, що складаються з 300 та 182 зображень відповідно. Частина-В розділена на підмножини поїздів і тестів, що складаються з 400 і 316 зображень. Кожна людина на зображенні натовпу анотована однією точкою, розташованою близько до центру голови. Загалом набір даних складається з 330 165 анотованих людей. Зображення з частини А були зібрані з Інтернету, тоді як зображення з частини Б були зібрані на жвавих вулицях Шанхаю.

Набір даних UCF-QNRF – це набір даних для підрахунку натовпу, і він містить велику різноманітність як сцен, так і типів фону [56]. Він складається з 1535 зображень з високою роздільною здатністю з Flickr, вебпошуку та відеоматеріалів про хадж. Кількість людей (тобто підрахунок) варіюється від 50 до 12 000 на кожному зображенні.

JHU-CROWD++ – це великомасштабний необмежений набір даних з підрахунку натовпу, що містить 4 372 зображення та 1,51 мільйона анотацій [57]. Цей набір даних зібрано за різних сценаріїв та умов навколишнього середовища. Крім того, набір даних містить порівняно багатий набір анотацій, таких як точки, приблизні межі, рівні розмиття тощо.

UCF-CC-50 – це набір даних для підрахунку натовпу, який складається із зображень надзвичайно щільного натовпу [58]. Загалом він містить 50 зображень з 63 974 анотаціями центрів голів. Кількість голів коливається від 94 до 4,543 на одне зображення. Невеликий розмір набору даних і велика дисперсія роблять його дуже складним для підрахунку.

В таблиці 2.1 зібрано загальну інформацію щодо найбільш популярних датасетів з натовпом, розглянутих вище.

Таблиця 2.1 – Опис наборів даних

Назва датасета, рік створення	Кількість зображень (навчальна вибірка, тестова вибірка)	Кількість людей на одному зображенні (від - до), кількість анотацій всього	Походження датасета	Особливості датасета
ShanghaiTech Part A, 2016	482 (300 навчальна, 182 тестова)	від 33 до ~3000, всього анотацій: 244,635	Інтернет (висока щільність)	Містить зображення з високою щільністю натовпу, здебільшого міські сцени.
ShanghaiTech Part B, 2016	716 (400 навчальна, 316 тестова)	від 9 до ~500, всього анотацій: 88,488	Спеціально сфотографовані	Менша щільність натовпу, сцени з менш переповнених вулиць міста Шанхай.
UCF-QNRF, 2018	1,535 (1,201 навчальна, 334 тестова)	від 49 до ~12,865, всього анотацій: 1,251,642	Інтернет	Дуже висока щільність, сцени з різних місць світу, включають розмитість, оклюзії та варіативність кутів зйомки.
JHU-CROWD++, 2020	4,372 (2,272 навчальна, 1,600 тестова)	від 0 до ~25,000, всього анотацій: 1,114,785	Інтернет та спеціально сфотографовані	Включає складні сцени: часткові оклюзії, різниця в погодних умовах, денні/нічні зображення, а також облік нечітких та прихованих людей.
UCF-CC-50, 2013	50 (40 навчальна, 10 тестова)	від 94 до ~4,543, всього анотацій: 63,974	Інтернет	Невеликий, але складний датасет з високою щільністю натовпу, включає різноманітні сцени зі світових подій.

2.3.2 Оцінка збалансованості датасетів ShanghaiTech, UCF-QNRF, JHU-CROWD++, UCF-CC-50

Хоча найрозповсюдженішим набором розмічених зображень в питаннях навчання нейронних мереж є Шанхайський набір, він є найбільш незбалансованим, що показано на рисунках 2.6 та 2.7 [60].

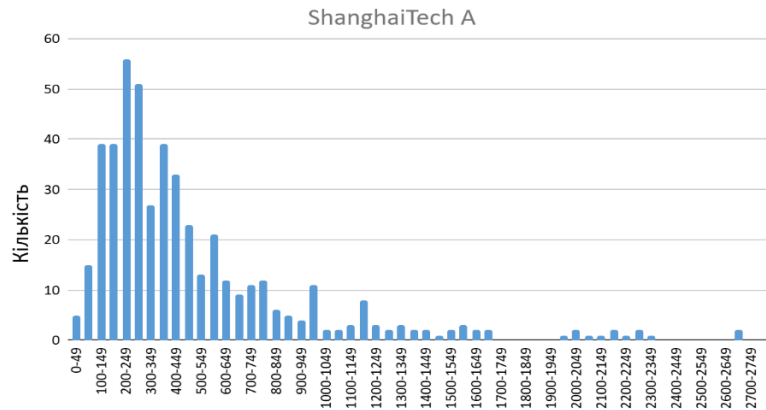


Рисунок 2.6 – Збалансованість у Шанхайському наборі даних, А

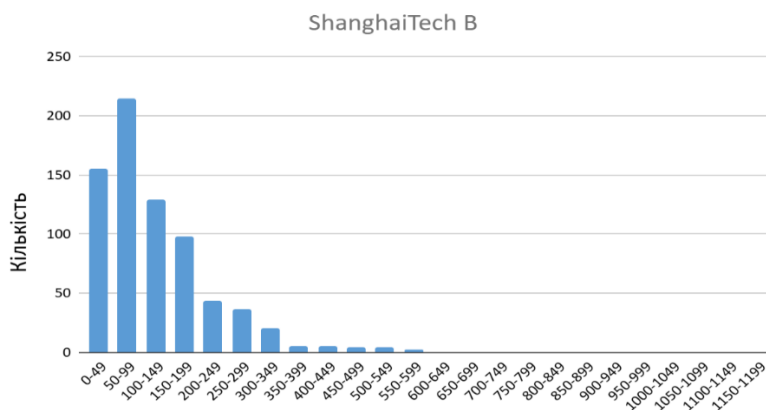


Рисунок 2.7 – Збалансованість у Шанхайському наборі даних, Б

Питання збалансованості означає співвідношення кількості зображень певної щільності до всієї кількості. На прикладі графіків видно, що Шанхайський датасет може покрити навчання на малі натовпи до 50-200 людей на зображення. Цілком логічно, що зображення більшої наповненості будуть мати більшу похибку в результаті.

Набір даних UCF-QNRF в свою чергу є більш збалансованим, але видно що зображень від 1200 менше 400 (рис 2.8). Натомість UCF-CC-50 має чудовий розподіл щодо відношень зображень різних класів, але складається всього з 50 зображень з великою роздільною здатністю. Такої кількості звичайно вкрай мало для серйозного навчання (рис 2.9).

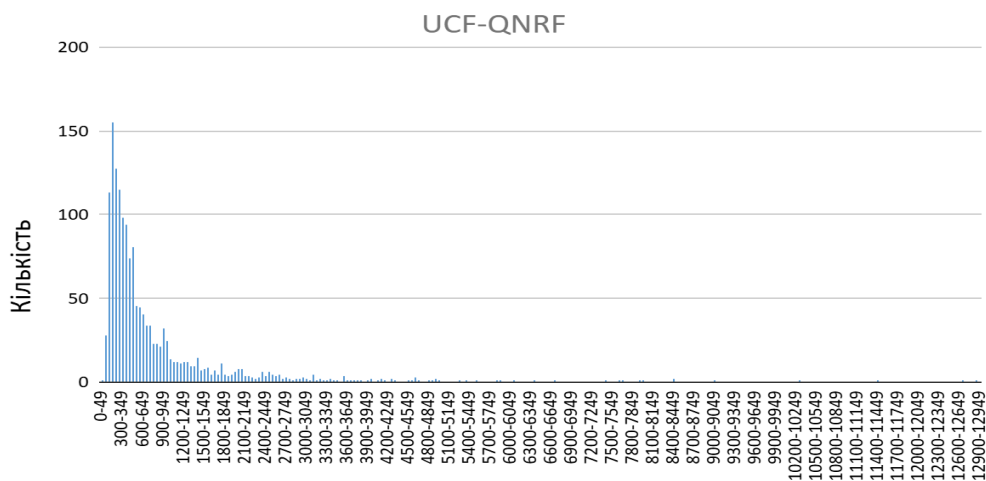


Рисунок 2.8 – Збалансованість у UCF-QNRF

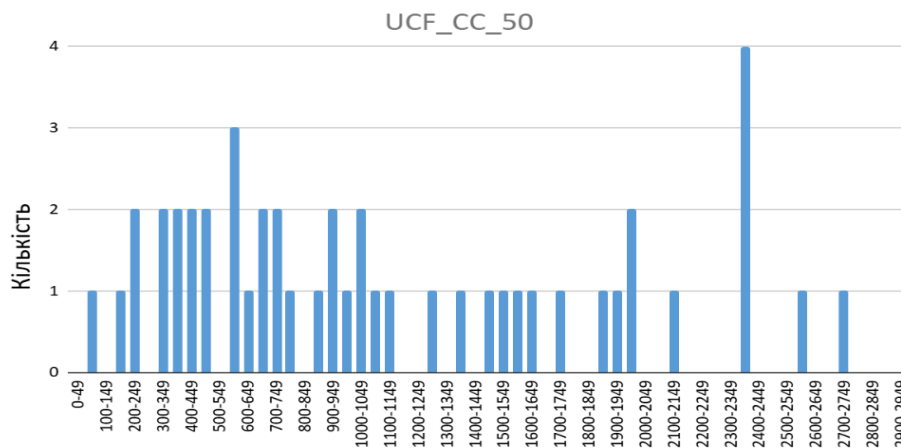


Рисунок 2.9 – Збалансованість у UCF-CC-50

JHU-CROWD++ – складається з великої вибірки образень різної щільності і різних погодних умов (рис 2.10).

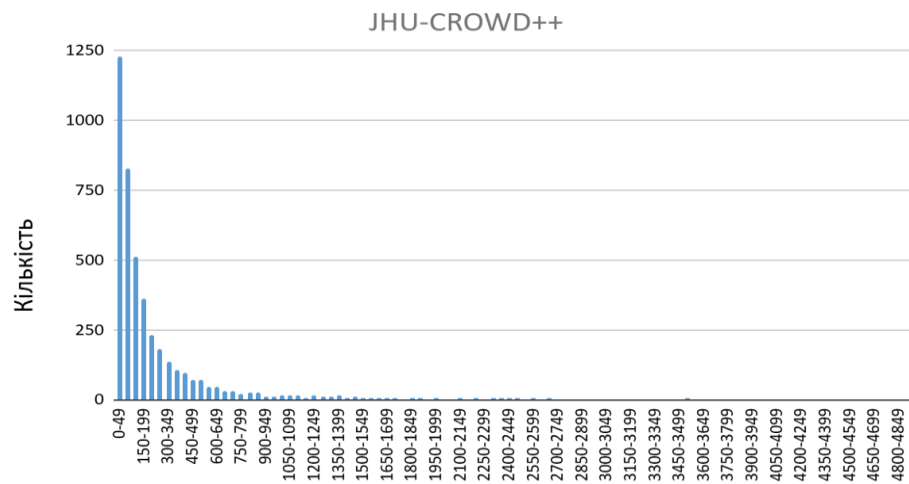


Рисунок 2.10 – Збалансованість у UCF-CC-50

Як бачимо цей датасет з усіх перелічених має найкращий баланс. Спільна кількість зображень для відповідних розмічених наборів відповідає даним з таблиці 2.2.

Таблиця 2.2 – Опис наборів даних

	ShanghaiTech A	ShanghaiTech B	UCF- QNRF	jhu_crowd_v2.0	UCF_CC_50
Кількість зображень	482	716	1535	4369	50

2.4 Аналіз якості підрахунку людей сучасних методів на датасетах ShanghaiTech, UCF-QNRF, JHU-CROWD++, UCF-CC-50

2.4.1 Точність на всьому датасеті

Методи, що з'явилися у 2024 році, показали наступні результати [52-54]:

Таблиця 2.3 – Похибки новітніх нейронних мереж на заданих датасетах

Мережа	Shanghai A	Shanghai B	UCF-QNRF	JHU-CROWD++	UCF-CC-50
P2PNet	52.74	6.25	85.32	-	172.72
CLIP-EBC	54.0	6	80.3	-	-
APGCC	48.8	5.6	80.1	54.3	154.8

Дійсно, зміни в математичних моделях різних методів різних років показують різницю, але не таку значну, як було у попередніх роках [61] (рис 2.11-2.13).

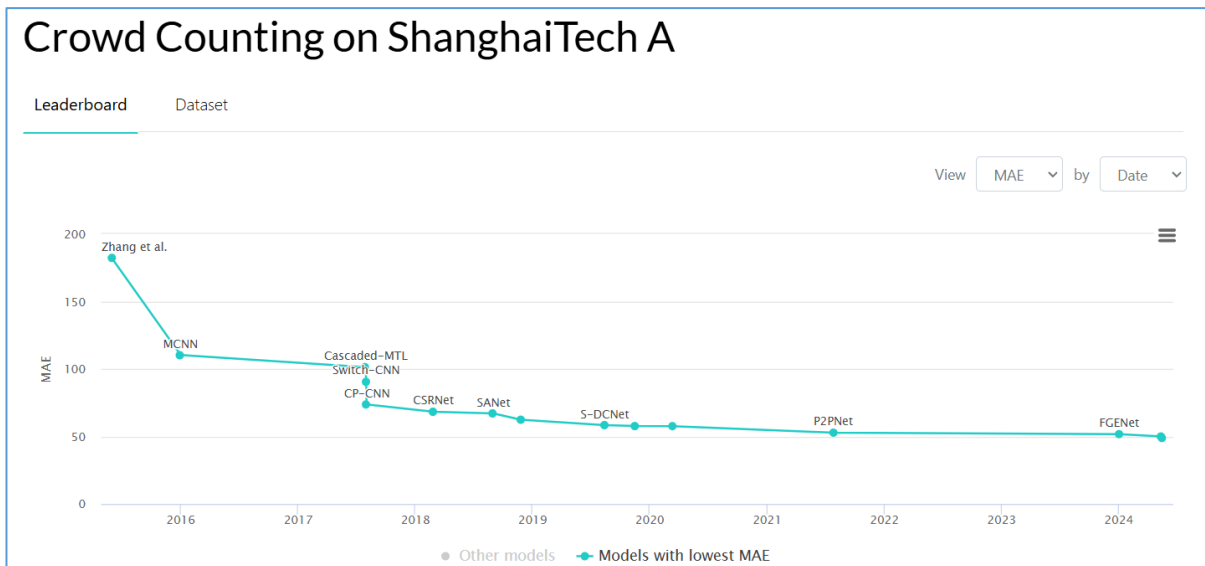


Рисунок 2.11 – Графік демонстрації мереж відносно різниці в похибці MAE

30	ACSCP	75.7	Crowd Counting via Adversarial Cross-Scale Consistency Pursuit	🔗	🔗	2018
31	Switch-CNN	90.4	Switching Convolutional Neural Network for Crowd Counting	🔗	🔗	2017
32	Cascaded-MTL	101.3 152.4	CNN-based Cascaded Multi-task Learning of High-level Prior and Density Estimation for Crowd Counting	🔗	🔗	2017
33	MCNN	110.2	Single-Image Crowd Counting via Multi-Column Convolutional Neural Network	🔗	🔗	2016
34	Zhang et al.	181.8	Cross-Scene Crowd Counting via Deep Convolutional Neural Networks	🔗	🔗	2015

Рисунок 2.12 – Список найперших рішень у цій сфері

Rank	Model	MAE ↓	MSE	RMSE	Paper	Code	Result	Year	Tags
1	APGCC	48.8	76.7		Improving Point-based Crowd Counting and Localization Based on Auxiliary Point Guidance			2024	
2	PSL-Net	49.9		77.6	Crowd Counting and Individual Localization Using Pseudo Square Label			2024	VGG
3	FGENet	51.66	85.00		FGENet: Fine-Grained Extraction Network for Congested Crowd Counting			2024	CNN
4	VMambaCC	51.87	81.3		VMambaCC: A Visual State Space Model for Crowd Counting			2024	Mamba
5	CLIP-EBC (ViT-B/16)	52.5		85.9	CLIP-EBC: CLIP Can Count Accurately through Enhanced Blockwise Classification			2024	ViT-B/16 CLIP EBC
6	P2PNet	52.74	85.06		Rethinking Counting and Localization in Crowds: A Purely Point-Based Framework			2021	
7	CLIP-EBC (ResNet50)	54.0		83.2	CLIP-EBC: CLIP Can Count Accurately through Enhanced Blockwise Classification			2024	EBC ResNet-50 CLIP
8	GauNet (ResNet-50)	54.8	89.1		Rethinking Spatial Invariance of Convolutional Networks for Object Counting			2022	

Рисунок 2.13 – Список найкращих мереж

2.4.2 Точність для зображень натовпу різної щільності

Значення похибки, яке демонструється у усіх моделях зазвичай є значенням середнім по всьому датасету, так значення звісно ж, не варто екстраполювати на щільні натовпи та навпроти на мало заповнені сцени. Задля того, щоб побачити реальну картину росту похибки відносно натовпу, варто провести експеримент власноруч.

На прикладі P2PNet видно, що відповідно до документації мережі її похибка MAE складає 53 по всій вибірці зображень. Запустивши локальну версію цієї мережі вдалося досягти похибки у 54 одиниці, що не є великою різницею. Цікавість представляє те, яка похибка була на різній щільності зображень. Користуючись логікою розбиття, згруповані відрізки складають таблицю 2.4:

Таблиця 2.4 – Тестування мережі P2PNet на Шангайському датасеті

Ліва межа	Права межа	Кіл-сть зображень	MAE	MSE
0	149	37	6,89	106,29
150	299	79	16,86	585,49
300	449	63	26,17	1010,30
450	599	34	28,44	1255,32
600	749	26	40,07	3361,84
750	899	17	59,11	4928,64
900	1049	12	99,91	14291,08
1050	1199	5	71	15223
1200	1349	3	55	3597,66
1350	1499	4	128,75	26439,75
1500	1649	5	302,8	140898
1650	1799	2	438	337005
1800	1949	0	-	-
1950	2099	4	236,5	84664,5
2100	2249	4	484,5	270274,5
2250	2399	2	520,5	399442,5
2400	2549	0	-	-
2550	2699	2	626	695477
2700	2849	0	-	-
2850	2999	0	-	-
3000	3149	1	317	100489

Таблиця містить результати аналізу точності передбачення для певної кількості зображень у різних інтервалах значень. Зазначені метрики: MAE – середня абсолютна похибка, MSE – середньоквадратична похибка.

В таблиці видно, що, дійсно, для зображень з низькою щільністю похибка значно менша, бо мережа краще знаходить ознаки окремих постатей. Окрім цього і кількість таких зображень значно більша, що свідчить про те, що такі зображення мережа буде опрацьовувати краще за зображення інших щільностей. Кількість зображень зменшується з підвищенням значень інтервалів.

Певні групи мають вкрай мало екземплярів що в результаті буде суттєво впливати на позитивні виявлення ознак. Деякі інтервали мають надзвичайно високі значення похибок:

- інтервал 1500–1649: MAE = 302.8, MSE = 140898;

- інтервал 2100–2249: MAE = 484.5, MSE = 270274.5.

Це може бути спричинено недостатньою кількістю зображень для цих діапазонів (наприклад, лише 5 або 4 зображення), що ускладнює навчання моделі.

Окрім цього, видно, що деякі групи щільності взагалі не представлені. Такий клас зображень як 2400-2549 або 2700-2849 взагалі не мають своїх власних представників у датасеті.

Для покращення результатів роботи моделі варто збільшити кількість зображень у рідкісних діапазонах, оскільки недостатня кількість даних у високих інтервалах значно впливає на точність. Додавання синтетичних даних для відсутніх інтервалів, таких як 1800–1949 чи 2400–2549, також може допомогти вирівняти розподіл вибірки.

Крім того, доцільно розглянути сегментовану оптимізацію моделі, адаптуючи її для кращої роботи у високих діапазонах значень. Застосування методів нормалізації або вагового регулювання допоможе зменшити вплив великих значень на загальні результати та покращити точність передбачень. Для покращення ситуації можна використати такі рекомендації:

- збільшити кількість зображень у рідкісних діапазонах. Це допоможе зменшити похибки у великих інтервалах, де модель показує низьку точність (наприклад, від 1500 і вище);
- розглянути сегментовану оптимізацію. Модель може бути переналаштована для кращої роботи у високих діапазонах;
- додати синтетичні дані. Створення даних для діапазонів із 0 зображень (1800–1949, 2400–2549 тощо) може покращити результати;
- застосувати методи нормалізації чи вагового регулювання. Це може знизити вплив великих значень на результати моделі.

Проблема нестабільної точності в моделюванні натовпу є однією з головних складнощів, з якими стикаються сучасні методи комп'ютерного зору та глибокого навчання. Вона виникає через низку факторів, серед яких варіації в щільності натовпу, умовах освітлення та наявності об'єктів, що можуть

ускладнювати точне підрахунку людей. Ці проблеми впливають на ефективність і стабільність результатів, знижуючи точність моделей, що особливо очевидно в умовах складних реальних сценаріїв.

Однією з основних причин нестабільної точності є варіації в щільності натовпу. Як показує практика, точність моделей змінюється в залежності від щільності натовпу. У випадках, коли натовп малий або помірний, моделі часто демонструють високу точність, однак при високій щільності людей, коли об'єкти частково перекривають один одного, точність значно знижується. Це відбувається через труднощі у виявленні окремих об'єктів, що можуть бути в тіні або частково сховані за іншими людьми, що ускладнює їх точний підрахунок.

Зовнішні умови, такі як освітлення, також є важливим фактором, що значно впливає на стабільність роботи моделей. Погані умови освітлення, тіні, відблиски або нерівномірне освітлення можуть серйозно порушити процес виявлення та ідентифікації людей. Тоді, коли світло або тіні змінюються в процесі зйомки, зображення стає менш чітким, а моделі глибокого навчання не завжди здатні правильно обробити ці зміни, що призводить до помилкових передбачень і підрахунків.

Іншим суттєвим чинником є зміни в контексті зображення, пов'язані з наявністю інших об'єктів або складних фонів. Якщо на зображенні є додаткові елементи, такі як будівлі, дерева чи інші предмети, вони можуть створювати фон, що ускладнює виявлення людей. Це особливо стосується сцен, де натовп перебуває на фоні насичених або складних пейзажів. У таких випадках моделі можуть неправильно інтерпретувати межі об'єктів, що призводить до зниження точності підрахунку.

Методологічні обмеження також відіграють важливу роль у формуванні нестабільних результатів. Багато існуючих методів моделювання натовпу базуються на регресії для оцінки щільності, однак такі методи можуть бути неефективними для натовпів з високою щільністю. Вони часто використовують бінінги або класи для зменшення варіацій в оцінках, що в

свою чергу може призвести до неточних результатів при великій кількості об'єктів, особливо коли ці об'єкти мають малий розмір або перекривають один одного.

В результаті модель може продемонструвати задовільну точність при низьких рівнях щільності, однак її продуктивність значно знижується в умовах високої щільності натовпу. Це може бути ознакою того, що дані для навчання моделі нерівномірно розподілені або існують певні недоліки в самому алгоритмі, які потребують оптимізації.

Отже, проблема нестабільної точності для моделювання натовпу є важливим і актуальним викликом для сучасних методів. Вона вимагає подальших досліджень та вдосконалення існуючих моделей. Інноваційні підходи можуть сприяти покращенню точності, але все ж нестабільність результатів залишатиметься значущою проблемою, яку необхідно вирішувати для досягнення високої надійності моделей в реальних умовах.

3 РОЗРОБКА PLUGIN ДЛЯ РЕДАКТОРА BLENDER ДЛЯ СТВОРЕННЯ ШТУЧНОГО ДАТАСЕТУ НАТОВПУ З МЕТОЮ ПІДВИЩЕННЯ ТОЧНОСТІ ПІДРАХУНКУ НАТОВПУ РІЗНОЇ ЩІЛЬНОСТІ

Розробка плагіну для Blender для створення штучного датасету натовпу – це потенційно ефективний підхід для покращення моделей підрахунку натовпу. Цей плагін може автоматизувати процес генерації сцен натовпів різної щільності й складу, різного освітлення і моменту часу; допомогти в отриманні точних анотацій для тренування моделей [62].

Важливим фактором є те, наскільки зображення сходяться з реальними екземплярами. Адаптація рендерів з програмного забезпечення також повинна бути максимально наближеною до кінцевого зображення, з яким застосунок буде стикатися на постійній основі.

3.1 Особливості створення plugin для редактора Blender

В програмному забезпеченні Blender запроваджено гнучку систему скриптового програмування. Завдяки ній можна маніпулювати поведінкою та виглядом окремих їх компонентів. Створення плагіна для Blender має свої особливості, зумовлені як специфікою самого програмного середовища, так і вимогами до автоматизації процесів у Blender. Blender підтримує розширення функціоналу через Python API, що дозволяє автоматизувати роботу в середовищі Blender і створювати плагіни. Це API забезпечує доступ до всіх ключових елементів і властивостей сцени, таких як об'єкти, матеріали, текстур, камери, джерела світла та інше. Особливості Python API в Blender:

- ієрархічна структура даних: доступ до всіх об'єктів сцени можна отримати через дерево даних (`bpy.data`, `bpy.context`);

- можливості інтеграції: API дозволяє створювати нові панелі інтерфейсу, кнопки та меню в Blender;
- подієве програмування: можна створювати функції для автоматизації рутинних завдань або створення складних об'єктів та сцен.

Плагін Blender має мати чітку структуру, яка складається з файлів Python. Основний файл зазвичай містить реєстрацію класів, що забезпечують функціонал плагіна. Основні елементи структури:

- клас плагіна: У Python-класі описуються всі функції та методи, необхідні для роботи плагіна;
- функції реєстрації: `register()` і `unregister()` функції потрібні для підключення та відключення плагіна;
- інтерфейс користувача: Можна додати UI-панелі, наприклад, у вкладку Tool Shelf (Т-панель) або N-панель, що робить плагін доступним у робочому просторі Blender.

Blender надає інструменти для створення елементів інтерфейсу, які можуть бути інтегровані в різні частини вікна редактора. Плагін для генерації натовпу може включати UI-панель з налаштуваннями для:

- вибору кількості та щільності об'єктів у сцені;
- зміни параметрів об'єктів (позиція, розміри, орієнтація);
- налаштування освітлення і фону.

Для створення інтерфейсу використовуються:

- класи `bpy.types.Panel` та `bpy.types.Operator`, що дозволяють створювати кнопки, повзунки, чекбокси та інші елементи;
- кастомні UI-елементи: Плагін може мати окремі вкладки, панелі, або інтеграцію у вже існуючі розділи Blender.

Для генерації великої кількості зображень натовпу різної щільності плагін повинен автоматично налаштовувати сцену і запускати рендеринг. Основні завдання:

- скриптування розташування об'єктів: використовуючи Python API, плагін може випадковим чином розташовувати людей у сцені, забезпечуючи різну щільність;
- налаштування камери та освітлення: автоматичне розміщення камери, налаштування фокусної відстані, перспективи та параметрів освітлення;
- запуск рендерингу та збереження зображень: плагін повинен автоматично зберігати результати рендерингу в певну папку, дотримуючись структури даних, необхідної для машинного навчання.

Розробка плагіна для Blender включає як створення інтуїтивного інтерфейсу, так і автоматизацію складних процесів. Такий плагін не тільки спрощує генерацію даних для підрахунку натовпу, але й забезпечує високий рівень кастомізації та продуктивності для ефективного створення якісних синтетичних датасетів.

3.2 Проектування та розробка розширення

Проектування та розробка плагіна для Blender передбачає структурований підхід, який охоплює планування функціональності, налаштування інтерфейсу користувача, використання Python API, та забезпечення продуктивності й зручності використання.

3.2.1 Структура розширення

Blender API потребує файл ініціалізації, який буде завантажувати і вивантажувати кастомізацію.

Контекст імплементації – це контейнер який зберігає проміжні змінні, потрібні для роботи плагіну.

Панель інтерфейсу запроваджує інтерфейс для користувача. Через неї можна маніпулювати певними змінними контексту (рис 3.1).

Оператор рендеру повинен викликатися через інтерфейс користувача та розширяти процес рендеру.

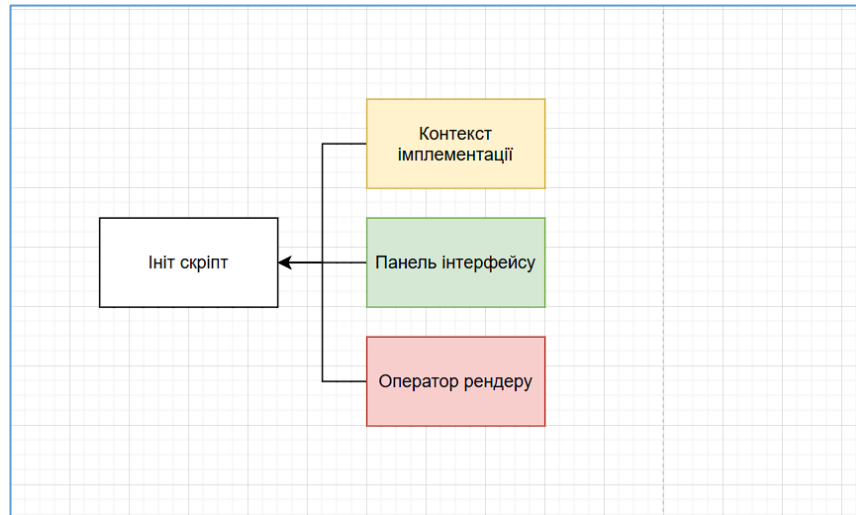


Рисунок 3.1 – Схема модулів застосунку

Код для плагіна Blender призначений для створення та рендерингу анотацій зображень, зокрема для використання у задачах підрахунку кількості людей на зображенні.

`ContextProperties` – клас визначає властивості, які зберігають налаштування рендерингу та параметри анотацій, зокрема шлях до папки для збереження зображень, кількість рендерів, фільтрацію оклюзій, а також об'єкт із налаштованими геометричними вузлами.

`RenderOperator` – клас реалізує операцію рендерингу з анотаціями. Він перевіряє правильність налаштувань і зберігає згенеровані зображення з координатами головних точок у кадрі. Функція `on_frame_changed` обчислює кількість об'єктів (людей), які потрапляють у поле зору камери на кожному кадрі.

`UiPanel` – відображає панель у інтерфейсі Blender, де користувач може налаштовувати параметри анотацій і рендерингу, зокрема вибирати головний

об'єкт, кількість кадрів, опцію фільтрації оклюзій тощо. Забезпечує зручний інтерфейс для запуску рендеру.

Helper Functions – містить функції для обробки координат, визначення видимих об'єктів, фільтрації за допомогою Z-буфера, обчислення абсолютних шляхів, і підрахунку об'єктів у кадрі, які потрапляють у поле зору камери. Функція `generate_meta` створює метадані для кожного рендеру, що можуть бути використані для анотацій. Ієрархія залежностей може бути відображена у UML синтаксі з рисунку 3.2.

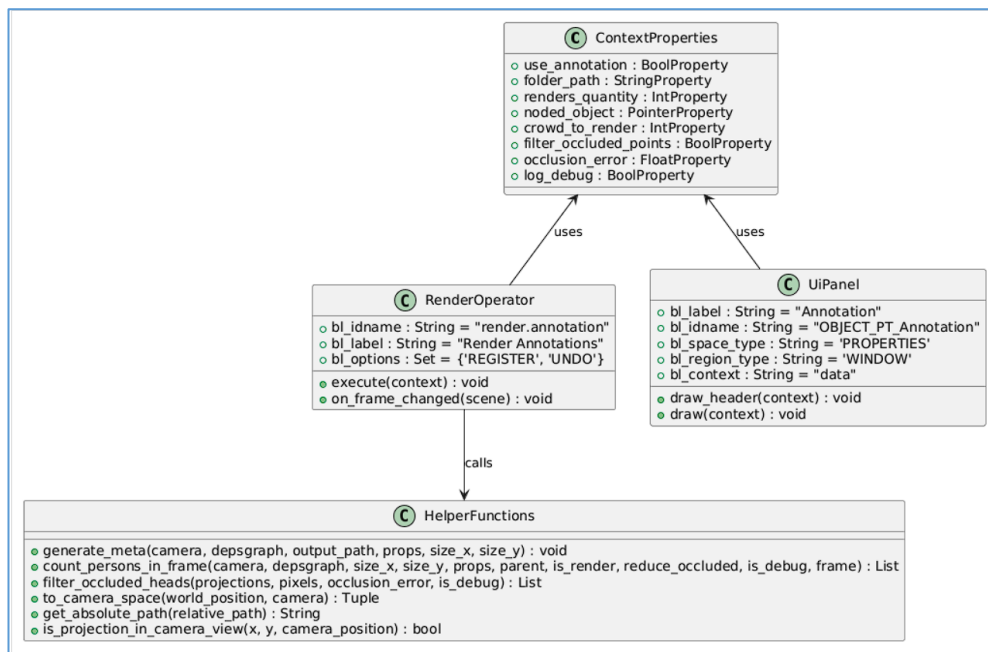


Рисунок 3.2 – UML діаграма класів розширення

3.2.2 Налаштування геометричних вузлів

Geometry Nodes у Blender – це система, яка дозволяє створювати та модифікувати геометрію об'єктів за допомогою візуального програмування. Це один з потужних інструментів Blender, який надає можливість гнучкого та параметричного моделювання через ноди (вузли), що забезпечує високий контроль над створенням та маніпуляцією об'єктів.

Налаштування геометричних вузлів буде зберігатися в об'єкті, який виступає поверхнею сцени, на якій довільно розміщені моделі людей. На початку в проєкт імпортовано заздалегідь створені моделі людей (рис 3.3).



Рисунок 3.3 – Моделі людей

На рисунку 3.4 видно, що окремі частини проєкту відведені під свої папки, їх відображення для рендеру налаштовано таким чином, аби відображалися лише моделі з поверхні.

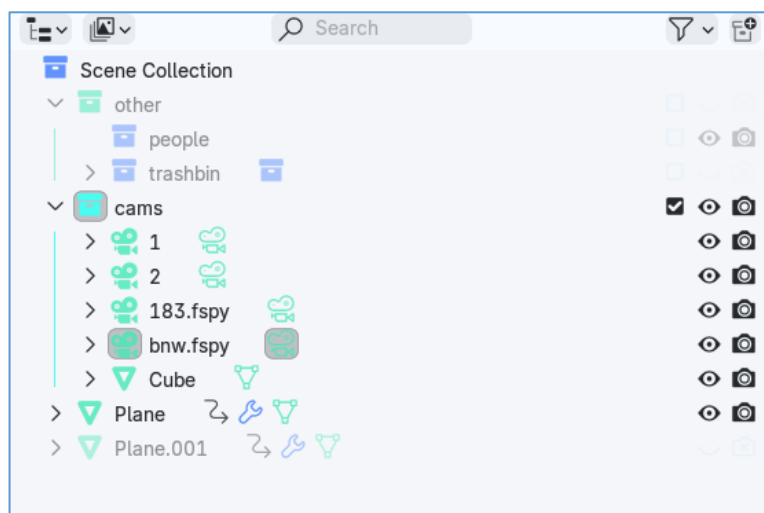


Рисунок 3.4 – Ієрархія сцени проєкту

Налаштування геометричних вузлів складається з декількох базових вузлів [63] (рис 3.5). На вхід передаємо певну кількість параметрів, на вихід –

геометрія яка складається після маніпуляцій. Окремим вузлом є імпорт колекції в якій підготовлені моделі людей для розміщення. Цікавими тут є вузли, які відповідають за фільтрацію людей поза межами кадру та вузол для розміщення людей.

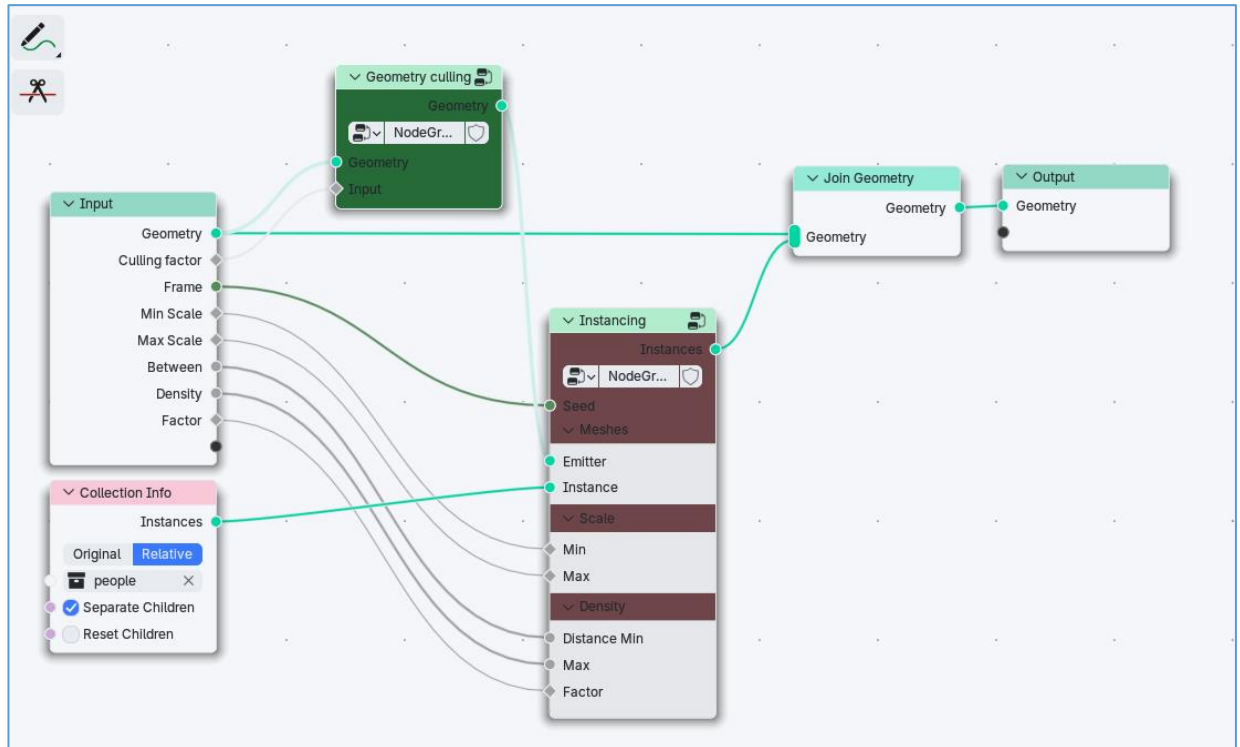


Рисунок 3.5 – Батьківські вузли проекту

Щоб економити ресурси при рендерінгу та при перегляді отриманої сцени в реальному часі, можна скористатися кулінгом камери, завдяки якому всі люди, розміщені поза межами камери, будуть фільтруватися і таким чином не будуть відображатися взагалі. Щоб зрозуміти чи належить людина до кадру камери використано налаштування з рисунку 3.6.

За умови, що кут об'єкту до камеру більше заданого значення, то ця геометрія видаляється (рис 3.7).

На зображенні вище видно що кількість геометрії значно зменшується. За значення кута в 0 градусів на сцені є дві з половиною тисячі об'єктів. При значенні у 0.5 – це вже тисяча сімсот, для 0.95 – шістсот.

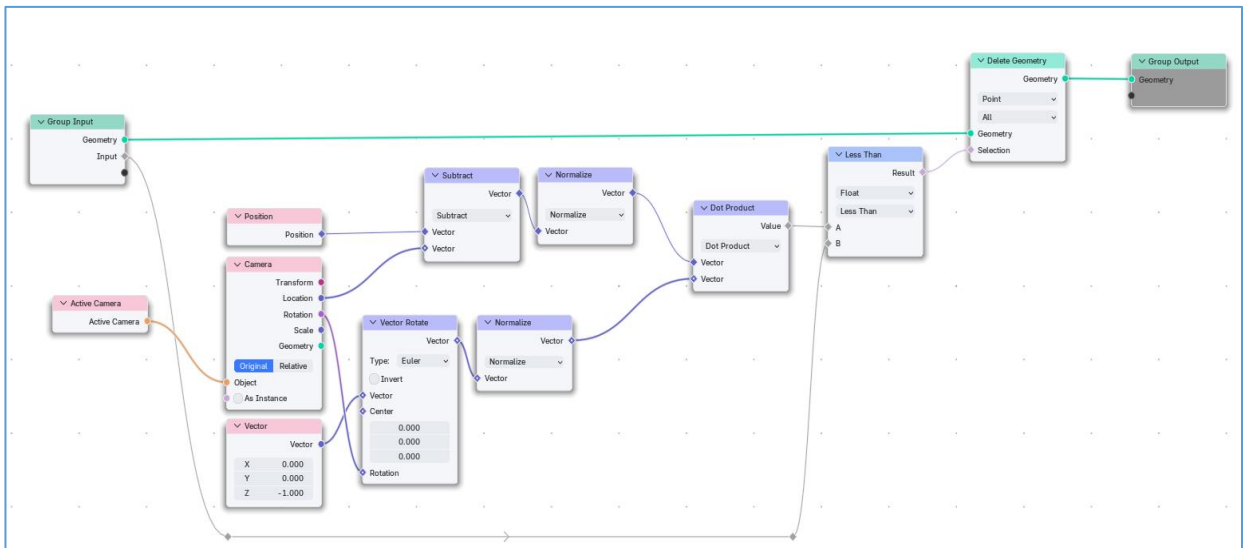


Рисунок 3.6 – Вузли відповідні за фільтрацію моделей поза кадром

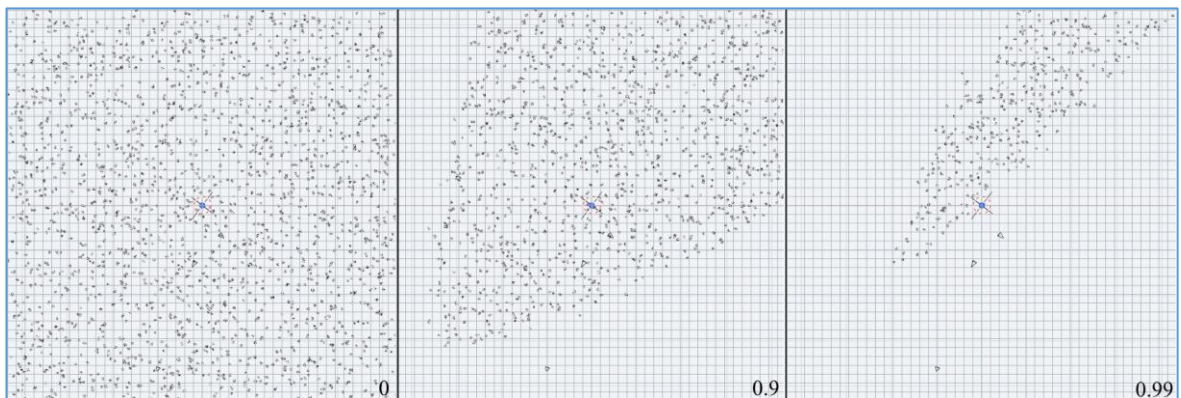


Рисунок 3.7 – Ефект від фільтрації моделей поза кадром

3.2.3 Налаштування камери відносно перспективи сцени

Правильне налаштування камери є важливим аспектом у створенні синтетичних наборів даних, оскільки від нього залежить достовірність сцени та якість отриманих зображень. Для задач підрахунку людей у натовпі, особливо у випадках із змінною щільністю натовпу та різними умовами освітлення, важливо налаштувати камеру таким чином, щоб вона відтворювала реалістичну перспективу сцени.

Налаштування фокусної відстані дозволяє контролювати кут огляду камери. Для імітації людського погляду зазвичай використовують фокусну відстань близько 35-50 мм, тоді як для сцени з великим охопленням натовпу можна обрати ширший кут, наприклад, 24 мм.

Визначення параметрів глибини різкості допомагає створити реалістичну сцену, де об'єкти ближчі до камери чіткі, а віддалені можуть виглядати розмитими. Це особливо важливо для імітації умов, де люди в натовпі знаходяться на різній відстані від камери.

Для відтворення сцени з натовпом камера зазвичай встановлюється на висоті, яка дозволяє охопити якомога більшу площу. Це може бути розташування камери зверху під кутом, щоб усі об'єкти були в полі зору. Кут від 45° до 60° може створити візуально привабливу сцену та захопити достатньо деталей.

Камера має бути орієнтована таким чином, щоб центр її поля зору потрапив на середину натовпу або місце з найбільшою щільністю людей. Це дозволить уникнути ефекту перспективного спотворення по краях зображення, що може негативно вплинути на точність анотації та підрахунку.

FSpy – це безкоштовний та відкритий інструмент, який допомагає створювати та налаштовувати камеру у Blender відповідно до перспективи на фотографії чи зображенні [64]. За допомогою FSpy можна легко відтворити положення та налаштування камери, щоб геометрія сцени у 3D відповідала перспективі оригінального зображення.

Це особливо корисно для таких завдань, як інтеграція 3D-об'єктів у реальні фотографії, архітектурна візуалізація та відтворення сцен на основі реальних зображень (рис 3.8).

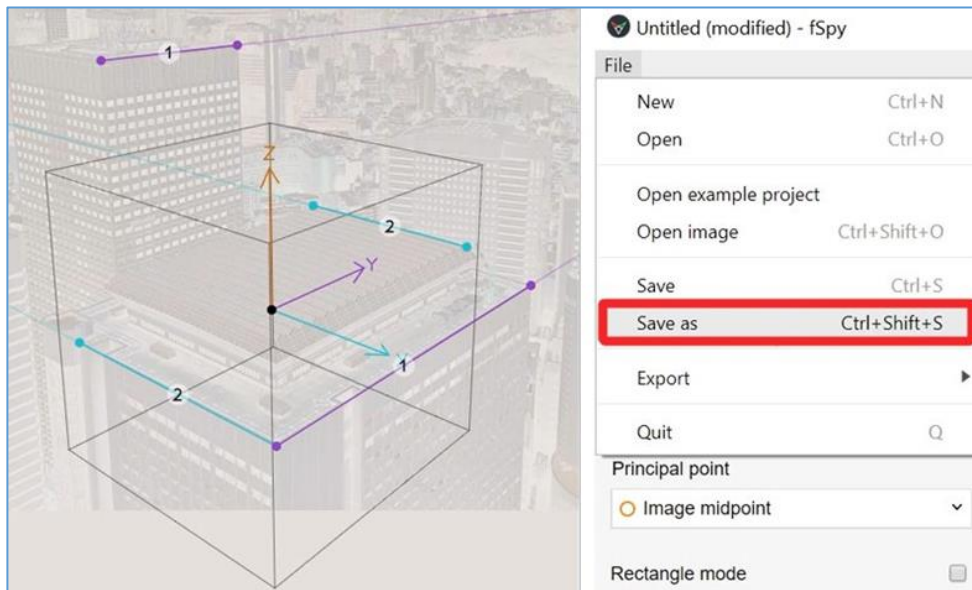


Рисунок 3.8 – Інтерфейс ПЗ FSpY

3.2.4 Постобробка зображення

Оклюзія голів на рендерах – це природне явище, коли частини одного об’єкта закривають інші об’єкти у сцені, як, наприклад, голови, що можуть частково або повністю закривати інші голови або частини тіла у натовпі. У задачах створення синтетичних датасетів для підрахунку людей у натовпі важливо правильно налаштувати оклюзію, оскільки вона наближає рендери до реальних умов та підвищує точність тренування моделей глибинного навчання.

Після рендерінгу для кожної моделі геометричних вузлів координати переводяться у проекцію камери, фінальний вектор має координати камери та відстань до неї. Щоб відфільтрувати голови, які не потрапляють у камеру достатньо перевірити щоб вони були у межах камери. В ситуації, коли одну голову перекриває інша, яка знаходиться ближче достатньо скористатися мапою глибини, яку програма вміє генерувати самостійно (рис 3.9).

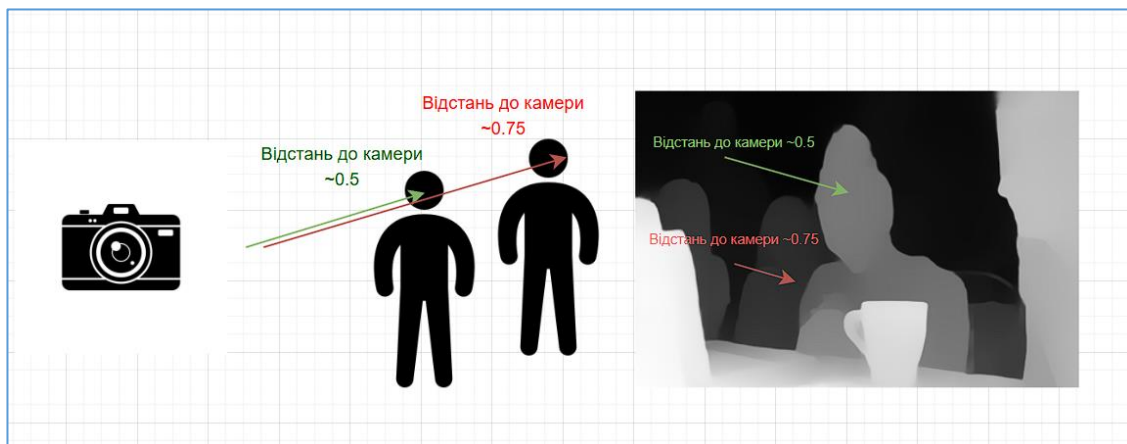


Рисунок 3.9 – Принцип фільтрації силуетів

Таким чином значення відстані конкретної голови порівнюється з малою глибини і якщо вони відрізняються, то цю голову не видно і її треба відфільтрувати.

3.2.5 Сценарії використання

Користувач визначає кількість людей у кадрі завдяки налаштуванням геометричних вузлів. Доречним є генерування декількох екземплярів рендерів для одного набору налаштування. Таким чином можливо відібрати найкращі екземпляри (рис 3.10).



Рисунок 3.10 – Зображення сцени з малою щільністю

Процедурний підхід при генерації сцен дозволяє перезбирати сцену змінюючи лише певні параметри. Таким чином зміна параметру щільності дозволяє переходити від сцени з малою наповненістю до середньої (рис 3.11).



Рисунок 3.11 – Зображення сцени з середньою щільністю

Користуючись параметрами сцени можна гнучко змінювати її стан. Продовжуючи тенденцію збільшення кількості людей у натовпі можна досягти такого стану сцени, який описує натовпи на концертах і масованих демонстраціях. Приклад такого зображення наведено на рисунку 3.12.



Рисунок 3.12 – Зображення сцени з високою щільністю

Окрім збільшення кількості людей на площині сцени, користувач має змогу контролювати відстань між людьми, що також суттєво впливає на фінальний результат. Змінивши відстань між людьми у попередній сцені отримано результат на зображенні 3.13.

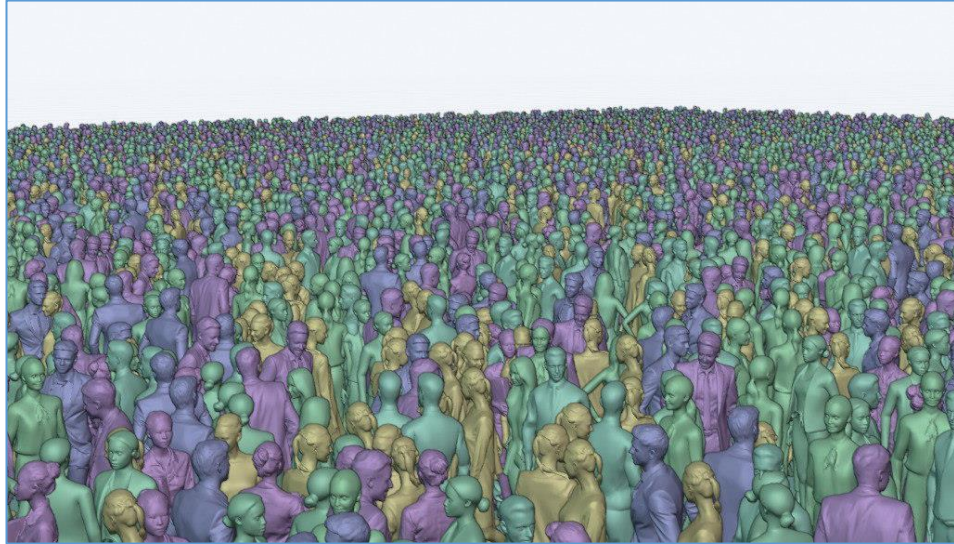
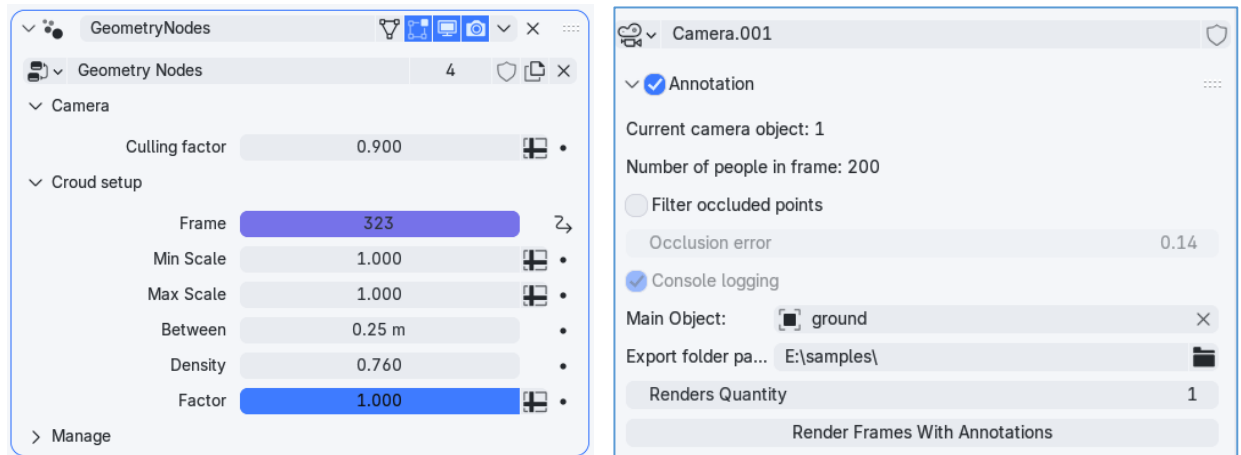


Рисунок 3.13 – Зображення сцени з високою щільністю та малою відстанню між людьми

Виконавши налаштування описаних вище параметрів, можна переходити до налаштування параметрів рендерінгу. Інтерфейси налаштування рендеру та самої сцени зазначено на рисунку 3.14.

Необхідно задати кількість кадрів, для яких налаштування будуть скопійовані, але натовп буде випадковим чином змінений. Серед налаштувань варто зазначити папку, де будуть збережені файли, об'єкт камери та платформи з людьми та фільтрацію перекритих голів.

Результатом буде згенеровано певну кількість зображень одної перспективи з різним положенням та розміщенням натовпу. Такі зображення можна додатково перевіряти на дефекти і зробити вибірку найкращих результатів.



а)

б)

Рисунок 3.14 – Інтерфейс налаштування параметрів рендерінгу та натовпу:

а) параметри натовпу; б) параметри рендерінгу

3.2.6 Версіонування коду

Версіонування є важливим аспектом розробки програмного забезпечення, оскільки дозволяє ефективно відстежувати зміни в кодовій базі. За допомогою системи контролю версій можна побачити, хто, коли і чому вносив певні зміни. Це полегшує аналіз помилок, повернення до попередньої стабільної версії та загальне розуміння розвитку проєкту.

Версіонування коду критично важливе для командної роботи. Коли декілька розробників працюють над одними і тими самими файлами, система версіонування запобігає конфліктам, дозволяючи об'єднувати зміни всіх учасників без втрати внесених правок. Крім того, вона підтримує паралельну розробку, наприклад, у різних гілках для додавання нових функцій або виправлення помилок.

Ще однією перевагою є резервування. Системи на кшталт Git створюють історію всіх змін, що дозволяє відновити видалені файли чи функції або повернутися до стабільної версії програми, якщо нові зміни викликають проблеми. Це забезпечує додатковий рівень захисту від втрати даних.

Версіонування також полегшує роботу з масштабними проєктами, де потрібно підтримувати різні версії програмного забезпечення одночасно. Наприклад, це корисно для виправлення помилок у старих релізах, додаючи нові функції до поточних версій без перешкод.

Сучасні підходи DevOps покладаються на системи контролю версій, дозволяючи автоматизувати тестування, збірку та доставку програмного забезпечення. Релізи продукту зв'язуються з конкретними тегами, що забезпечує чітку організацію роботи. Окрім того, версіонування створює можливості для історичного аналізу. Розробники можуть дослідити, як змінювався проєкт, і використовувати ці дані для прийняття кращих рішень у майбутньому.

Без версіонування зростають ризики втрати важливого коду, ускладнюється інтеграція командної роботи та підтримка старих версій продукту. У результаті, система контролю версій стає «страховкою», яка дозволяє ефективно управляти змінами та швидко вирішувати проблеми.

3.2.7 Безперервна інтеграція та безперервна доставка/розгортання

CI/CD (Continuous Integration/Continuous Delivery/Deployment) – це підхід у розробці програмного забезпечення, який забезпечує автоматизацію і покращення процесів інтеграції, тестування, доставки та розгортання додатків. Цей метод сприяє підвищенню якості продукту, зменшенню ризиків та прискоренню випуску нових функцій.

CI полягає в регулярному інтегруванні змін до основної гілки репозиторію. Це включає автоматичне тестування, що забезпечує швидку перевірку якості нового коду. Основні аспекти CI:

- автоматичне тестування: після внесення змін до коду система запускає набір тестів для перевірки його працездатності;

- часте інтегрування: розробники вносять свої зміни кілька разів на день, що дозволяє швидко виявляти та виправляти конфлікти;
- єдина база коду: усі розробники працюють із актуальною версією, мінімізуючи ризики «застарілого» коду.

Continuous Delivery є продовженням CI. Воно забезпечує, щоб код, який пройшов усі автоматичні перевірки, був готовий для розгортання в будь-який момент. Основна ідея CD полягає в тому, щоб підтримувати продукт у стані, придатному до випуску:

- автоматизація доставки: код автоматично збирається, тестується і готується до релізу;
- відсутність ручних процесів: мінімум ручних втручань, що знижує ймовірність помилок;
- можливість частого релізу: хоча реліз може бути не миттєвим, продукт готовий до випуску в будь-який час.

Цей етап є ще більш автоматизованим. У Continuous Deployment після проходження всіх перевірок код автоматично розгортається в продуктивне середовище без втручання людини:

- швидке розгортання: Код розгортається на продакшн одразу після проходження тестів;
- підвищення якості: Завдяки частим релізам команди можуть швидше реагувати на зворотний зв'язок від користувачів;
- вимоги до надійності тестів: Оскільки процеси повністю автоматизовані, важливо мати 100% точні тести, щоб уникнути розгортання багів.

3.2.8 Автоматизація збірки розширення

Будь який сервіс по створенню віддалених сховищ коду надає можливість збирати його версії та, наприклад зберігати їх. Обраний GitHub в

свою чергу, має інструменти для автоматичної збірки коду після злиття змін в окрему гілку. GitHub Actions – це скриптова мова програмування, в якій можна визначити процедуру збірки коду, що буде передумовою для цього. Кінцевий продукт в контексті розширення для Blender – це стиснений архів, в якому знаходиться код та мета інформація розширення. Таким чином процес збірки може складатися з кроків, як показано на рисунку 3.15.

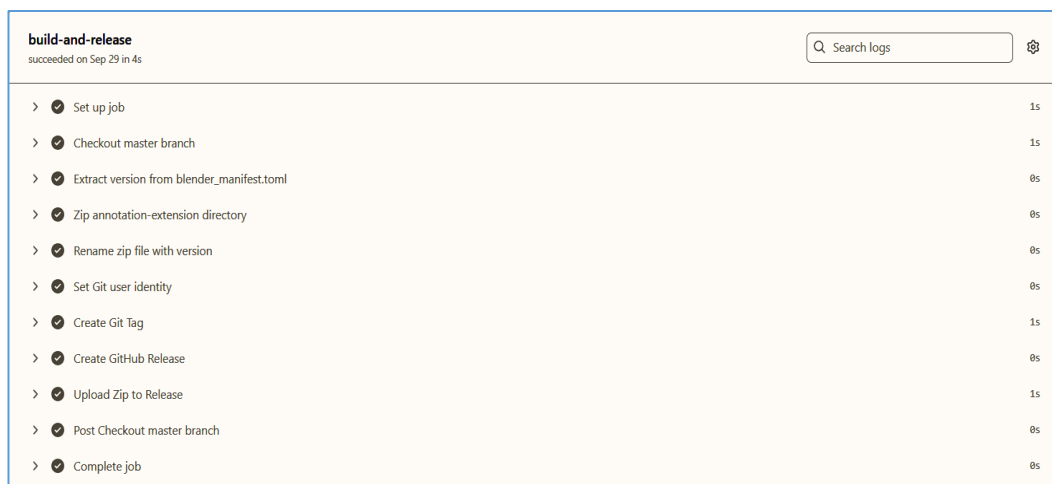


Рисунок 3.15 – Перелік етапів збірки релізного пакету

Відповідно до версій, гарним тоном вважається ведення відліку змін – changelog. Таким чином користувач може ознайомитися зі змінами і обрати ту версію розширення, яка підходить йому найбільше (рис 3.16-17). Такий підхід дозволяє розробляти програмне забезпечення прозоро та зрозуміло як для користувача, так і для інших розробників у рамках великої команди.

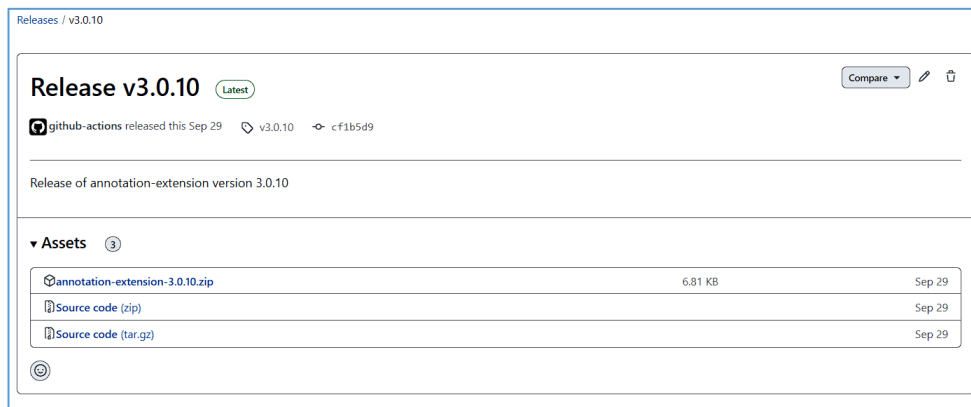


Рисунок 3.16 – Приклад сторінки релізу

Changelog
This file contains changelog history. Even though it's reflected in commits here you have short change log story.
3.0.0
This release was aimed to refactor depth map part as it was working incorrectly in several cases. Initially, I tried to get advice from community.
<ul style="list-style-type: none"> • ADDED Script that tests Mist and Z-buffer values (as teoretically they both could be used for depth mesurment) • ADDED UI option and functionality for detailed logging. Shows diffs for each point. • CHANGED Iterating over the depth map as it has anchor point in left-bottom and method calculation returns it left-top format.
3.0.8 CHANGED GitHub actions should be stored under release tags 3.0.7 ADDED GitHub actions to publishing artifact 3.0.6 ADDED Generation of the META of render batch 3.0.5 ADDED Depth Error could be configured with UI 3.0.4 CHANGED Adapted toml file 3.0.3 CHANGED Refactor Render Operator to separate file 3.0.2 CHANGED Refactored to separate files 3.0.1 ADDED Check for the Z-Buffer Composition set up
2.0.0
New revision of the addon was aimed to add filter of the occluded heads. The solution is based on the comparing depth values of particular head and Z-buffer value.
<ul style="list-style-type: none"> • ADDED Generating of the Z-buffer • ADDED UI option for enabling occluded heads filtering
2.0.1 REMOVED Generation of the annotation exclusively as it is now depth-map dependent
1.0.0

Рисунок 3.17 – Список змін в різних версіях релізів

3.3 Формування штучної вибірки за допомогою розширення

Розроблений плагін для Blender автоматизує процес створення штучного датасету для підрахунку натовпу. Завдяки плагіну користувач може генерувати зображення та відповідні анотації, зокрема координати розташування об'єктів у кадрі, що дозволяє зберігати інформацію про видимі об'єкти в форматі, придатному для тренування моделей підрахунку натовпу. Переваги використання плагіна для формування датасету:

- автоматизація: плагін значно прискорює процес створення датасетів, мінімізуючи ручну роботу;
- гнучкість налаштувань: можливість налаштувати кількість кадрів, параметри фільтрації та інші змінні дозволяє створювати датасети для різних задач і умов;
- підвищення точності: використання Z-буфера для фільтрації оклюзій та автоматичне створення анотацій підвищує точність датасету та робить його більш придатним для навчання моделей підрахунку.

Таким чином, розроблений плагін для Blender забезпечує зручний інструмент для створення якісних штучних датасетів, оптимізованих для задач аналізу натовпу.

3.3.1 Формування штучного датасету

Створення штучного датасету за допомогою розробленого плагіна в Blender виконується шляхом автоматизованої генерації зображень сцени з натовпом та відповідних анотацій для кожного кадру. Налаштування сцени та об'єктів для рендерингу:

- перший етап передбачає налаштування сцени з об'єктами, що представляють натовп;
- за допомогою геометричних вузлів та інших інструментів Blender налаштовується «натовп» із бажаною кількістю об'єктів (наприклад, 3D-моделей людей або спрощених об'єктів, які імітують людей);
- камера сцени встановлюється так, щоб охопити бажану область з об'єктами.

Встановлення параметрів рендерингу в плагіні. Через панель інтерфейсу плагіна в Blender користувач задає основні параметри, які впливатимуть на рендеринг і структуру майбутнього датасету:

- папка для збереження: шлях, куди зберігатимуться результати;
- кількість кадрів: задається кількість рендерів, які будуть створені для датасету;
- фільтрація оклюзій: опція, яка дозволяє виключати об'єкти, що частково або повністю закриті іншими об'єктами;
- параметри анотації: такі як активація логів, точність Z-буфера тощо.

Рендерінг з анотаціями:

- при запуску процесу рендерингу, плагін автоматично зберігає кожен кадр як окреме зображення. Рендеринг виконується на основі заданих параметрів, забезпечуючи стабільну якість зображень;

- разом із рендерингом зображень, плагін створює анотації — текстові файли, які містять координати об’єктів у полі зору камери та кількість об’єктів у кадрі;

- якщо фільтрація оклюзій активована, для кожного кадру застосовується Z-буфер для видалення точок, які знаходяться за іншими об’єктами.

Створені анотації зберігаються у вигляді текстових файлів, розташованих у відповідній папці. Формат анотацій містить:

- координати об’єктів у кадрі: позиції об’єктів (піксельні координати на зображенні), які можна використати для подальшого навчання моделей;

- додаткові метадані: такі як кількість об’єктів, які не були виключені фільтром, та помилки оклюзії;

- час створення: дата й час, коли кожен кадр був згенерований, що полегшує відстеження процесу.

Таким чином, розроблений плагін забезпечує швидке й ефективне створення високоякісного штучного датасету, оптимізованого для задач підрахунку натоппу та інших подібних комп’ютерних зору.

3.3.1 Візуальне порівняння отриманих результатів

Для кожного проміжку було згенеровано по тридцять зображень, які повинні покривати реальні зображення такого класу. Для порівняння розглянемо зображення та його перспективу перенесемо у камеру програмного забезпечення (рис 3.18).

Після рендеру кожної групи зображень отримуємо структуру папок, яка складається з «meta», «image» та «gt», де мета має інформацію щодо сгенерованих картинок, сама папка картинок та зняті анотації відповідно.

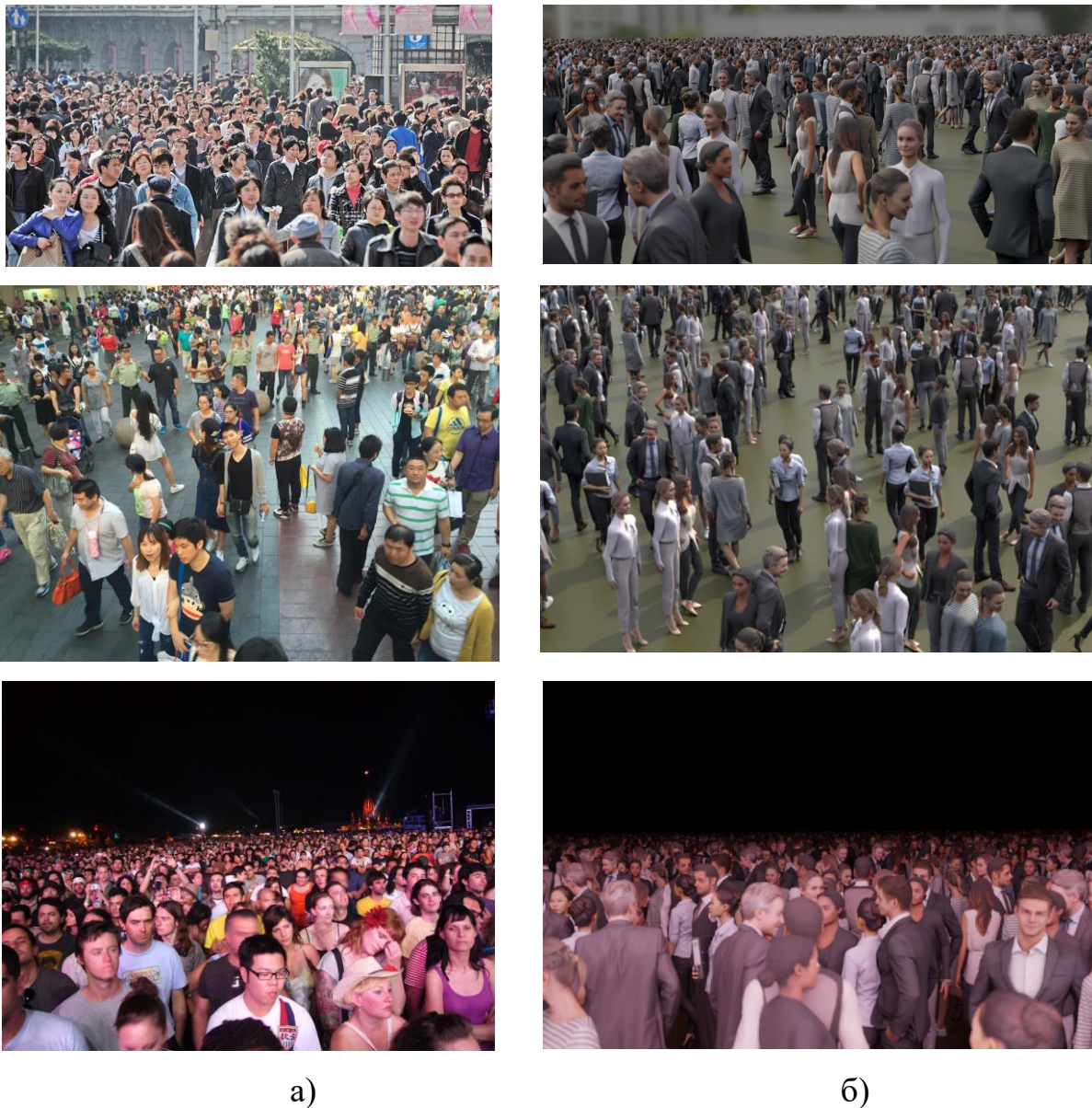


Рисунок 3.18 – Порівняння згенерованих та реальних зображень

а) зображення з датасету; б) згенеровані зображення

Таким чином після декількох запусків у корневій папці отримуємо зображення, які можна використовувати у навчанні або тестуванні нейронних мереж (рис 3.19).

Якщо скористатися будь-яким засобом, щоб перенести точки анотацій та зображення, то вийде зображення, на якому кожна голова, яка не перекрита, має анотацію (рис 3.20).

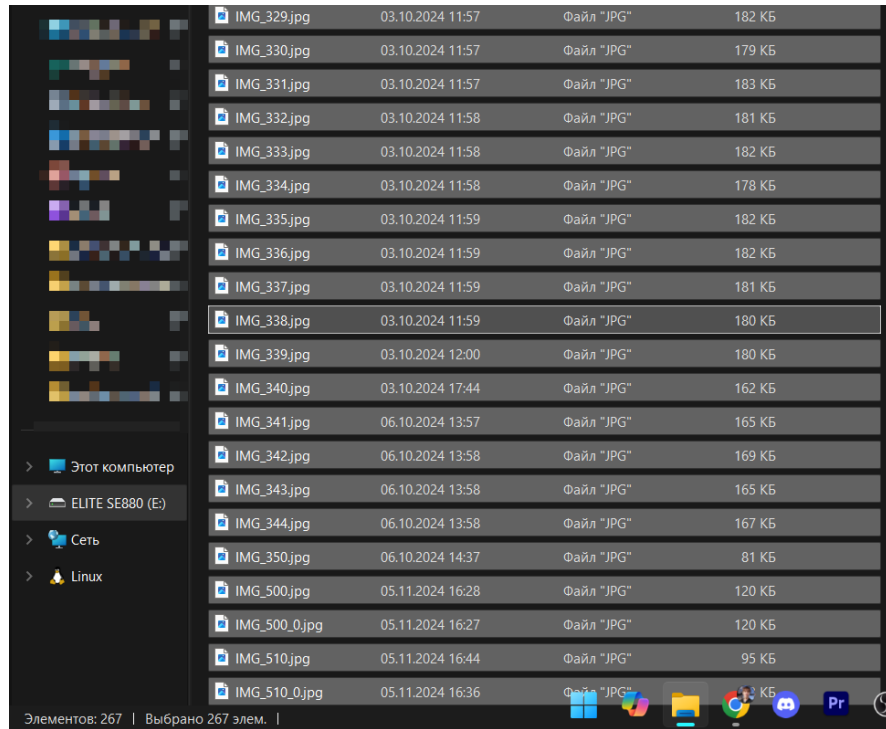


Рисунок 3.19 – Список згенерованих зображень



Рисунок 3.20 – Зображення з анотаціями

3.3.2 Формування комбінованого датасету

Комбінований датасет зазвичай об'єднує реальні та синтетичні дані для використання переваг обох типів. Основною метою є створення датасету, що поєднує різноманітність і розмір реальних даних з контролем і змінюваністю, які надає синтетичний контент. Це допомагає підвищити точність моделей і їхню стійкість, особливо в випадках, коли реальні дані обмежені або важко доступні (наприклад, сцени вночі або при екстремальних щільностях натовпу).

Для об'єднання реальних і синтетичних датасетів можуть використовуватись кілька методів:

- аугментація даних: реальні зображення можна доповнити синтетичними даними різними способами (наприклад, змінюючи умови освітлення, додаючи шум), щоб симулювати різні умови або щільність натовпу;

- злиття: синтетичні зображення можна вставляти в реальні сцени, або реальні натовпи можна доповнювати синтетичними людьми, створеними через Blender;

- вагове об'єднання: реальні та синтетичні дані можуть бути злиті з різним вагою при тренуванні, щоб уникнути перенавчання на синтетичних ознаках.

Переваги комбінованого датасету:

- різноманітність даних: поєднання реальних і синтетичних даних забезпечує ширший спектр сценаріїв для тренування, що підвищує стійкість моделі;

- покращена точність: завдяки подоланню обмежень як реальних, так і синтетичних даних комбінований датасет допомагає досягти кращого узагальнення;

- баланс реальних та синтетичних даних: забезпечення правильного балансу допомагає моделі навчатися реалістичним динамікам натовпу без перенавчання на штучних ознаках.

3.4 Тестування точності підрахунку натовпу різної щільності

Для того щоб проаналізувати те, як розпізнавання буде відбуватися для згенерованих картинок, в сумі було створено більше трьохсот зображень різного розміре, перспективи і заповнення. Для тестів створювалося по 30 зображень різних категорії. Кожна з яких має на 150 людей більше (рис 3.21).



Рисунок 3.21 – Приклад зображення

Таким чином, кожне зображення можна пропустити через P2PNet і порівняти з еталонною розміткою і порахувати метрики. Таким чином можна отримати інформацію відносно того, наскільки згенеровані кадри певної щільності добре оброблюються мережею, в порівнянні з реальними зображеннями (табл.3.1).

Таблиця 3.1 – Похибка на штучному датасеті

Проміжок	Кількість	MAE	MSE	MAE S	MSE S
0-149	37	6,89	106,29	4,73	30,13
150-299	79	16,86	585,49	14,2	215,46
300-449	63	26,17	1010,30	28,6	873,46
450-599	34	28,44	1255,32	45,76	2217,96
600-749	26	40,07	3361,84	64	9807,8
750-899	17	59,11	4928,64	-	-

При тестуванні використовувалась модель P2PNet, яка була згадана раніше. Як видно з результатів, після на третьому проміжку похибка у кількості людей починає збільшуватися швидше, аніж у випадку з реальними зображеннями.

3.5 Рекомендації щодо створення штучних датасетів за допомогою редактора Blender

Створення штучних датасетів у Blender дозволяє генерувати контрольовані та варіативні зображення, що використовуються для тренування алгоритмів комп'ютерного бачення. Для задачі підрахунку людей у натовпі або інших сценаріїв, де важлива точність та реалістичність синтетичних даних, рекомендується дотримуватися ряду правил і методів. Нижче наведені основні рекомендації для створення ефективних штучних датасетів у Blender.

Аналогічно можна вирішувати задачі, де потрібно порахувати авто. Наприклад, на паркінгах, вулицях, чи для моніторингу трафіку. У медичній сфері автоматизація генерації зображень клітин, пухлин і їх варіацій. У роздрібній торгівлі: моніторинг покупців, кількості товару, виявлення полицок з товаром. У сільському господарстві та питаннях екології, можливо моделювати зображення врожаю, кількості тварин у стадах, чи на фермах.

3.5.1 Загальні налаштування

По-перше, треба звернути увагу на налаштування сцени: кількість людей, положення об'єктів, щільність натовпу, освітлення, перспектива камери тощо.

По-друге, треба використовувати різноманітні пози та анімації для моделей, щоб створити максимально реалістичний набір даних із динамічними та статичними зображеннями.

По-третє, для кожного наступного кадру треба налаштувати рандомізацію, особливо для об'єктів, які представляють людей у натовпі. Це дозволяє отримати великий діапазон зображень з різними положеннями та взаємними перекриттями об'єктів.

Різні умови освітлення дозволяють створювати більш реалістичну картинку. Передачі зовнішніх умов: денне світло, сутінки, нічне освітлення, штучне світло – це забезпечить адаптивність моделі до різних ситуацій (рис. 3.22, 3.23).



Рисунок 3.22 – Зображення сцени з низькою кількістю світла



Рисунок 3.23 – Фронтальне зображення у низкій освітленості

3.5.2 Застосування текстур та матеріалів

Також має сенс використовувати джерела світла з випадковими налаштуваннями інтенсивності, напрямку та кольору для створення сцен з реалістичними тінями та освітленням. Відображення та відблиски, якщо сцена передбачає дзеркальні поверхні, наприклад, скло або воду додають реалістичності рендеру [61] (рис 3.24).



Рисунок 3.24 – Зображення матеріалів, які можуть бути використані у сцені

Різноманітні текстури для моделей одягу, шкіри, поверхонь, на яких розташований натовп – це додає реалістичності та ускладнює задачу моделі, покращуючи її здатність розпізнавати людей у різних контекстах. Деталізовані текстури для фону або оточуючих об'єктів, що відтворюють умови реального світу (тротуари, стіни, рослинність тощо).

3.5.3 Формування штучного датасету

Використовуйте моделі з помірною кількістю полігонів для прискорення рендерингу або використати remesh для моделі, особливо якщо необхідно згенерувати велику кількість зображень. У випадку потреби можна зменшити деталізацію далеких об'єктів або застосувати різні рівні деталізації. Для великих натовпів використовуйте методи інстанціювання, які дозволяють ефективно клонувати об'єкти без значного збільшення навантаження на систему.

Перед початком процесу генерування варто зробити перевірку налаштування. Для цього має сенс відрендерити один екземпляр зображення та перевірити зображення на виході, з списком розмічених голів. Розмічене зображення має відповідати поставленим вимогам точності. Голови, які перекриваються людьми переднього плану можуть відфільтруватися, але це залежить від перспективи зображення (рис. 3.25).



Рисунок 3.25 – Зображення з анотацію людей

ВИСНОВКИ

У рамках кваліфікаційної роботи було досліджено питання підрахунку наповну використанням нейронних згорткових мереж. Робота надає глибокий аналіз архітектур останніх актуальних рішень, аналізує їх переваги та відмінності. У результаті дослідження здійснено аналіз найпопулярніших розмічених наборів даних, який показав, що найширше вживані набори підпадають під проблему незбалансованості, і деякі класи не мають достатньої кількості зображення. Розроблено технічне розширення, яке дозволяє покрити незбалансованість реальних наборів даних за допомогою генерації штучних зображень, наближених до реальних. Розроблено технічне рішення для автоматичної анотації згенерованих зображень.

Експериментальні дослідження показали, що навчені на реальних зображеннях мережі здатні працювати зі штучними зображеннями та ідентифікувати їх. Найбільшу ефективність мережа продемонстрували при роботі з зображеннями малої та середньої наповненості. Також було проаналізовано розмічені анотації та анотації отримані від мережі. На вище зазначених класах анотації мережі співпадали з анотаціями розміченими програмно.

Проте результати для зображень класу високої щільності показали, що мережа починає втрачати точність швидше, аніж це було для зображень зроблених традиційним шляхом, взятих з класичних розмічених наборів. Така поведінка може бути зумовлена багатьма факторами:

- потрібно більш точно обробляти штучні зображення, додавати певні фільтри та робити корекцію кольору, корегувати різкість та розмиття; налаштування камери може відрізнитися від реальних зображень, що впливає на вихідний результат;

- сама сцена, з якої генерувалося зображення, потребує виправлень (більша кількість тестових моделей, зміна оточення навколо, зміна hdri-мапи та моделі освітлення);

– без дослідження лишилось те, як мережа навчена виключно на штучних зображеннях буде реагувати на реальні зображення.

Наукова новизна полягає у підході реорганізації наборів даних як таких, в той час як нові архітектури мереж надають незначні покращення у точності і націлені на мережу, розглянута робота фокусується на розмічених наборах даних та показує їх незбалансованість та пропонує спосіб в який цієї незбалансованості можна позбутися автоматично та прицільно відносно класу та наповненості людей у зображенні.

Таким чином, проведене дослідження показало проблему незбалансованості найбільш вживаних розмічених наборів даних. Було запропоновано технічне рішення для генерації розмічених наборів даних завдяки розширенню до програмного забезпечення Blender. Проведено дослідження на згенерованих зображеннях, яке показало потенціал цього напрямку. Покращення штучних зображень дозволить зняти обмеження, які існують при використанні класичних розмічених наборів даних.

Результати дослідження апробовано у вигляді тез доповідей під час XV Міжнародної науково-практичної конференції «Free and open source software» [33] та під час Міжнародного молодіжного форуму «Радіоелектроніка та молодь у XXI столітті», конференції «Сучасні методи обробки зображень» [34].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Кобилін, О.А., & Творошенко І.С. (2021). *Методи цифрової обробки зображень: навч. посібник*. Харків: ХНУРЕ.
2. Гороховатський, В.О., & Творошенко І.С. (2021). *Методи інтелектуального аналізу та оброблення даних: навч. посібник*.
3. Кобилін, О. А., & Творошенко І. С. (2021). *Методи цифрової обробки зображень*.
4. Gorokhovatskyi V., Tvoroshenko I., Kobylin O., and Vlasenko N. (2023) Search for visual objects by request in the form of a cluster representation for the structural image description, *Advances in Electrical and Electronic Engineering*, 21(1), pp. 19-27.
5. Kobylin, O., & Lyashenko, V. (2020). Time series clustering based on the k-means algorithm.
6. Y. Daradkeh, V. Gorokhovatskyi, I. Tvoroshenko, S. Gadetska, M. Al-Dhaifallah, (2021). Methods of classification of images on the basis of the values of statistical distributions for the composition of structural description components, *IEEE Access* 9 92964–92973.
7. Gorokhovatskyi V., Tvoroshenko I., and Yakovleva O. (2024) Transforming image descriptions as a set of descriptors to construct classification features, *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 33, no. 1, pp. 113-125. DOI: 10.11591/ijeecs.v33.i1.pp113-125.
8. Gorokhovatskyi , O., & Yakovleva , O. (2024). Medoids as a Packing Of ORB Image Descriptors. *Advanced Information Systems*, 8(2), pp. 5–11. <https://doi.org/10.20998/2522-9052.2024.2.01>
9. Yakovleva, O., & Nikolaieva K. (2020). Research of descriptor based image normalization and comparative analysis of SURF, SIFT, BRISK, ORB, KAZE, AKAZE descriptors. *Advanced Information Systems*, 4(4), 89-101. doi:10.20998/2522-9052.2020.4.13.

10. Yakovleva O., Nebeský L, Liakhov P. (2023). Research methods of texture image analysis to solve the texture search problem. Proceedings of the IV International Scientific and Practical Conference «The world of modern technologies and inventions». Vienna, Austria. 2023. pp. 252-261.

11. Яковлева, О. В., & Кускова, І. В. (2006). Дослідження результатів сегментації зображень методом матриць збігів. Вісник Національного технічного університету «ХПІ». №39 - С.164 -171.

12. Яковлева, О. В., & Панченко, І. А. (2007). Застосування енергетичних характеристик Лавса для сегментації зображень. Біоніка інтелекту : науково-технічний журнал. №2(67). - С.94-98.

13. Яковлева О.В., Нестерова О.П. (2009) Порівняльний аналіз методів характеристик Лавса і матриць збігів у задачах сегментації текстурних зображень. Прикладна радіо-електроніка: науч.-техн. журнал, Том 8, №2. - С.181 - 187.

14. Gorokhovatskyi, O., Peredrii, O., Gorokhovatskyi, V., & Vlasenko, N. (2023). Explanation of CNN image classifiers with hiding parts. In Explainable Deep Learning AI (pp. 125-146). Academic Press.

15. V. Gorokhovatskyi, I. Tvoroshenko (2020). Image Classification Based on the Kohonen Network and the Data Space Modification, in: CEUR Workshop Proceedings: Computer Modeling and Intelligent Systems (CMIS-2020), pp. 1013–1026.

16. Gorokhovatskyi V., Tvoroshenko I., Yakovleva O., Hudáková M., and Gorokhovatskyi O. (2024) Application a committee of Kohonen neural networks to training of image classifier based on description of descriptors set, IEEE Access, vol. 12, pp. 73376-73385. DOI: 10.1109/ACCESS.2024.3404371

17. Творошенко, І.С. (2021). Технології прийняття рішень в інформаційних системах: навч. посібник. Харків: ХНУРЕ.

18. Yakovleva, O., Kovtunenکو, A., Liubchenko, V., Honcharenko, V., & Kobylin, O. (2023). Face Detection for Video Surveillance-based Security System. CEUR Workshop Proceedings Vol. 3403. pp. 69-86. ISSN 1613-0073

19. Yakovleva, O., Kovač, M., Ardasov, V. & Yeremenko, I. (2023). Study on adding functionality to the Zoom online conference system for monitoring the participant activities. *Public Administration and Regional Development*, 19(1), pp. 161–186.

20. Yakovleva O., Matúšová S., Tvoroshenko I., Isaiev Y. (2024). Visitor counting based on video stream analysis from surveillance cameras. *Scientific Journal of Bratislava University of Economics and Management «Public Administration and Regional Development, Economics, Management and Marketing»*, vol. 20, no. 1, pp. 67–87.

21. Ковтуненко, А. Р., Яковлева, О. В., Любченко, В. А., & Янголенко, О. В. (2020). Дослідження сумісного використання математичної морфології та згорткових нейронних мереж для вирішення задачі розпізнавання цінників.

22. Cherednichenko, O., Kanishcheva, O., Yakovleva, O., & Arkatov, D. (2020). Collection and Processing of a Medical Corpus in Ukrainian. In *COLINS* (pp. 272-282).

23. Naumenko V., Shelest V., Yakovleva O. (2024). Combination of .Net technology and Angular framework to develop application for testing SQL language knowledge. *Proceedings of the XVth International Scientific and Practical Conference «Free And Open Source Software»*, Ukraine, Kharkiv, February 13-14, 2024. pp.63-66.

24. Cherednichenko, O., Vovk, M., Yanholenko, O., & Yakovleva, O. (2020). Towards the technology of employers' requirements collection development. In *Integrated Computer Technologies in Mechanical Engineering: Synergetic Engineering* (pp. 228-239). Cham: Springer International Publishing.

25. Shelest V., Yakovleva O. (2024) Research on selecting web application architecture based on the analysis of applied requirements. *Proceedings of the X International Scientific and Practical Conference «Computerintegrated technologies of automation of technological processes»*. Hamburg, Germany, pp. 46-54.

26. Cherednichenko, O., Yanholenko, O., Iakovleva, O., & Kustov, O. (2014). Models of research activity measurement: web-based monitoring implementation.

In Information Systems: Education, Applications, Research: 7th SIGSAND/PLAIS EuroSymposium 2014, Gdańsk, Poland, September 25, 2014. Proceedings 7 (pp. 75-87). Springer International Publishing.

27. Cherednichenko, O., Yanholenko, O., Liutenko, I., & Iakovleva, O. (2013). Monitoring and Evaluation Problems in Higher Education-Comprehensive Assessment Framework Development. In CSEDU (pp. 455-460).

28. Yanholenko, O., Cherednichenko, O., Yakovleva, O., & Arkatov, D. (2020). A Model for Estimating the Security Level of Mobile Applications: a Fuzzy Logic Approach. In IntelITSIS (pp. 252-266).

29. Klyuchka, Y. A., Cherednichenko, O. Y., Vasylenko, A. V., & Yakovleva, O. V. (2017). Forecasting the results of football matches on the internet based information. Bulletin of National Technical University "KhPI". Series: System Analysis, Control and Information Technologies, 0(55), 51–59.

30. Yakovleva O., Nebeský L., Kirichenko A. (2023). Using the GPT models for responses based on custom content to develop neural consultant for university applicants. Abstracts of V International Scientific and Practical Conference «The world of modern technologies and inventions» Madrid, Spain. Pp. 172-178.

31. Cherednichenko, Olga, Ivashchenko, Oksana, Cibák, Ľuboš and Lincenyi, Marcel. «Item Matching Model in E-Commerce: How Users Benefit» Economics and Culture, vol.20, no.1, 2023, pp.77-90.

32. Cherednichenko, O., Ivashchenko, O., Lincényi, M., Kováč, M. 2023. Information technology for intellectual analysis of item descriptions in e-commerce, Entrepreneurship and Sustainability Issues 11(1): 178-190.

33. Гречишкін Д.С., Яковлева О.В. (2024). Огляд можливостей застосування Blender для процедурного моделювання. Матеріали XV-ої міжнародної науково-практичної конференції «Free And Open Source Software», Україна, Харків, 13-14 лютого 2024. С.122-124.

34. Гречишкін Д.С. (2024). Метод детектування людей у натовпах різної щільності. Матеріали XXVIII Міжнародного молодіжного форуму

«Радіоелектроніка та молодь у XXI столітті», Конференція «Сучасні методи обробки зображень», Т.7. Харків: ХНУРЕ. (С. 33-35).

35. Гречишкін, Д. С. (2023). Огляд датасетів для вирішення задачі підрахунку людей. Матеріали XIV-ої Міжнародної науково-практичної конференції «Free and Open Source Software», Харків, 14-16 лютого 2023 р.– Харків: Харківський національний економічний університет імені Семена Кузнеця, 110 с., 53.

36. Гречишкін Д.С., Яковлева О.В. (2023), Аналіз у порівняльному аспекті моделей MCNN та CSRNet для вирішення задачі підрахунку людей у натовпі. 27-ий міжнародний молодіжний форум «Радіоелектроніка і молодь у XXI столітті», С. 48-49.

37. Zhang, Y., Zhou, D., Chen, S., Gao, S., & Ma, Y. (2016). Single-image crowd counting via multi-column convolutional neural network. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 589-597).

38. Li, Y., Zhang, X., & Chen, D. (2018). Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1091-1100).

39. P. Viola, M. Jones, (2001). Rapid Object Detection Using a Boosted Cascade of Simple Features, in: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001).

40. N. Dalal, B. Triggs, (2005). Histograms of Oriented Gradients for Human Detection, in: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05).

41. H. Li, Z. Lin, X. Shen, J. Brandt, G. Hua, (2015). A convolutional neural network Cascade for face detection, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5325–5334.

42. R. Girshick, J. Donahue, T. Darrell, J. Malik, (2014). Rich feature hierarchies for accurate object detection and semantic segmentation, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587.

43. S. Ren, K. He, R. Girshick, J. Sun, (2017). Faster R-CNN: Towards real-time object detection with region proposal networks, in: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1137–1149.
44. K. Zhang, Z. Zhang, Z. Li, Y. Qiao, (2016). Joint face detection and alignment using multitask cascaded convolutional networks, *IEEE Signal Processing Letters* 23(10) 1499–1503.
45. S. Zhang, X. Zhu, Z. Lei, H. Shi, X. Wang, S. Z. Li, (2017). FaceBoxes: A CPU real-time face detector with high accuracy, in: *2017 IEEE International Joint Conference on Biometrics (IJCB)*, pp. 1–9.
46. S. Ren, K. He, R. Girshick, J. Sun, (2016). Faster R-CNN: Towards real-time object detection with region proposal networks, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39(6) 1137–1149.
47. J. Deng, J. Guo, E. Ververas, I. Kotsia, S. Zafeiriou, (2020). Retinaface: Single-shot multi-level face localisation in the wild, in: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5203–5212.
48. E. Zhang, Y. Zhang, (2009). Average precision, *Encyclopedia of Database Systems* 192–193.
49. Pretrained Pytorch Face Detection (MTCNN) and facial recognition (InceptionResnet) models, 2019. URL: <https://github.com/timesler/facenet-pytorch>.
50. Star-Clouds/Centerface: Face detection, 2019. URL: <https://github.com/Star-Clouds/CenterFace>.
51. InsightFace Python Library, 2022. URL: <https://github.com/deepinsight/insightface/tree/master/python-package>.
52. Unique, worry-free model photos. Generated Photos, 2022. URL: <https://generated.photos>.
53. APGCC: Improving Point-based Crowd Counting and Localization Based on Auxiliary Point Guidance., 2024. URL: <https://apgcc.github.io>.
54. Ma, Y., Sanchez, V., & Guha, T. (2024). CLIP-EBC: CLIP Can Count Accurately through Enhanced Blockwise Classification. arXiv preprint arXiv:2403.09281.

55. TencentYoutuResearch/CrowdCounting-P2PNet: The official codes for the ICCV2021 Oral presentation «Rethinking Counting and Localization in Crowds: A Purely Point-Based Framework», 2024. URL: <https://github.com/TencentYoutuResearch/CrowdCounting-P2PNet>.

56. ShanghaiTech Dataset | Papers With Code, 2024. URL: <https://paperswithcode.com/dataset/shanghaitech>

57. UCF-QNRF Dataset | Papers With Code., 2024. URL: <https://paperswithcode.com/dataset/ucf-qnrf>.

58. JHU-CROWD++ | A large-scale unconstrained crowd counting dataset, 2024. URL: <http://www.crowd-counting.com>.

59. UCF-CC-50 Dataset | Papers With Code., 2024. URL: <https://paperswithcode.com/dataset/ucf-cc-50-1>.

60. The latest in Machine Learning | Papers With Code 2024. URL: <https://paperswithcode.com/>

61. Idrees, H. et al. (2018). Composition Loss for Counting, Density Map Estimation and Localization in Dense Crowds. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds) Computer Vision – ECCV 2018. ECCV 2018. Lecture Notes in Computer Science(), vol 11206. Springer, Cham. https://doi.org/10.1007/978-3-030-01216-8_33

62. ShanghaiTech A Benchmark (Crowd Counting) | Papers With Code. 2024. URL: <https://paperswithcode.com/sota/crowd-counting-on-shanghaitech-a>.

63. blender.org - Home of the Blender project - Free and Open 3D Creation Software 2024. URL: blender.org

64. Geometry Nodes - Blender 4.3 Manual. 2024. URL: https://docs.blender.org/manual/en/latest/modeling/geometry_nodes/index.html.

65. fSpy. 2024. URL: <https://fspy.io/>