МАГІСТЕРСЬКА РОБОТА

МР. ІПм - 48.00.00.000 ПЗ

Група ІПм-23-2

Макаренко Володимир

2024

Івано-Франківський національний технічний університет нафти і газу

Інститут інформаційних технологій

Кафедра інженерії програмного забезпечення

Макаренко Володимир Миколайович

(прізвище, ім'я, по батькові)

УДК 004.942

(індекс)

МАГІСТЕРСЬКА РОБОТА

Моделі та методи використанням природної мови при обробці зображень

(назва роботи)

Інженерія програмного забезпечення

(назва освітньої програми)

121 - Інженерія програмного забезпечення

(шифр і назва спеціальності)

Макаренко В.М.

(підпис, ініціали та прізвище здобувача освітнього ступеня)

Науковий керівник Шекета Василь Іванович, д.т.н., професор

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

Допущено до захисту

Завідувач кафедри

доц. Бандура В.В.

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

Нормоконтроль

доц. Вовк Р.Б.

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

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

Івано-Франківськ – 2024

Івано-Франківський національний технічний університет нафти і газу

Інститут інформаційних технологій

Кафедра інженерії програмного забезпечення

Освітній рівень магістр

Спеціальність 121 – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:
Зав. кафедрою **ІПЗ**
доц. В.В. Бандура
“ 04 ” вересня 2024 р.

ЗАВДАННЯ

НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

Макаренку Володимиру Миколайовичу

(прізвище, ім’я, по-батькові)

1. Тема магістерської роботи “ Моделі та методи використанням природної мови при обробці зображень”

керівник проекту (роботи) Шекета Василь Іванович, д.т.н., професор

затверджені наказом закладу вищої освіти від “ 22 ” листопада 2024 р. № 781/7

2. Строк подання студентом проекту (роботи) 15 грудня 2024 р.

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

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

1. Дослідження області використання комп’ютерного зору та розпізнавання зображень
2. Моделі та алгоритми застосування нейронних мереж для процесів розпізнавання об’єктів
3. Підхід та алгоритми реалізації процесу розпізнавання об’єктів
4. Імплементація моделей та алгоритмів використання нейронних мереж при обробці зображень

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

1. Проекти Equal Entry щодо VR для людей з обмеженими можливостями (рис. 1.1)
2. Проект Microsoft Research, який досліджує інноваційні аудіотехнології (рис. 1.2)
3. Компоненти системи Range-IT (рис. 1.3)
4. Об’єкт а XYZ площині (рис. 1.4)
5. Завдання комп’ютерного зору з розпізнавання об’єктів (рис. 1.5)

6. Консультанти розділів проекту (роботи)

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

7. Дата видачі завдання 04 вересня 2024 р.

Керівник _____
(підпис)

Завдання прийняв до виконання_____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назви етапів магістерської роботи	Строк виконання етапів роботи	Примітка
1	Підбір і вивчення літератури по темі магістерської роботи	15.09.2024	виконано
2	Аналіз концепцій та алгоритмів предметної області	29.09.2024	виконано
3	Дослідження області використання комп'ютерного зору та розпізнавання зображень	15.10.2024	виконано
4	Моделі та алгоритми застосування нейронних мереж для процесів розпізнавання об'єктів	08.11.2024	виконано
5	Підхід та алгоритми реалізації процесу розпізнавання об'єктів	20.11.2024	виконано
6	Імплементація моделей та алгоритмів використання нейронних мереж при обробці зображень	01.12.2024	виконано
7	Затвердження пояснювальної записки роботи завідувачем кафедри	15.12.2024	виконано

Студент – магістр _____
(підпис)

Керівник роботи _____
(підпис)

АНОТАЦІЯ

Магістерська робота: 79 с., 49 рис., 1 табл., 51 джерело.

Тема: Моделі та методи використанням природної мови при обробці зображень

Об’єкт дослідження: процеси аналізу, розпізнавання та опису об'єктів у віртуальному середовищі.

Мета роботи: розробка та імплементація комплексного рішення для розпізнавання, аналізу та опису об’єктів у віртуальному середовищі, заснованого на застосуванні нейронних мереж та методів обробки природної мови.

Предмет дослідження: моделі, алгоритми та методи використання нейронних мереж і природної мови для розпізнавання та опису об'єктів у віртуальному середовищі.

Результати дослідження

В роботі проведено порівняльний аналіз ефективності об'єктно-орієнтованих і середовищно-орієнтованих алгоритмів у завданнях пошуку та узагальнення.

Висновок

Запропоноване рішення інтегрує кілька компонентів, зокрема алгоритми машинного навчання для виявлення об'єктів та X3DOM для візуалізації 3D-сцен у браузері

ГЛИБОКЕ НАВЧАННЯ, НЕЙРОННІ МЕРЕЖІ, ОБРОБКА ПРИРОДНОЇ МОВИ, РОЗПІЗНАВАННЯ ОБ’ЄКТІВ, ОПИС СЦЕН, ВІРТУАЛЬНЕ СЕРЕДОВИЩЕ, АЛГОРИТМИ ОПТИМІЗАЦІЇ, СТАТИСТИЧНИЙ АНАЛІЗ.

ABSTRACT

Master Thesis: 79 pp., 49 fig., 1 tab., 51 sources.

Thesis Subject: Models and methods using natural language in image processing

Object of research: processes of analysis, recognition and description of objects in a virtual environment.

The purpose of the work: development and implementation of a complex solution for recognition, analysis and description of objects in a virtual environment, based on the application of neural networks and natural language processing methods.

Research subject: models, algorithms and methods of using neural networks and natural language for recognizing and describing objects in a virtual environment.

Research results

In the work, a comparative analysis of the effectiveness of object-oriented and environment-oriented algorithms in search and generalization tasks is carried out.

Conclusion

The proposed solution integrates several components, including machine learning algorithms for object detection and X3DOM for rendering 3D scenes in the browser

DEEP LEARNING, NEURAL NETWORKS, NATURAL LANGUAGE PROCESSING, OBJECT RECOGNITION, SCENE DESCRIPTION, VIRTUAL ENVIRONMENT, OPTIMIZATION ALGORITHMS, STATISTICAL ANALYSIS.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

Algo - Algorithm

EC - Environment Centric

NLP - Natural Language Processing

OC - Object Centric

ANOVA - Analysis of Variance – дисперсійний аналіз

ВСТУП

Актуальність теми.

З розвитком інформаційних технологій, штучного інтелекту та віртуальних середовищ зростає потреба у розробці інноваційних рішень для аналізу, розпізнавання та опису візуальних даних. Важливість цієї теми зумовлена кількома чинниками.

По-перше, сучасні системи взаємодії з віртуальним середовищем активно застосовуються в різних сферах, таких як освіта, охорона здоров’я, промисловість, ігрові технології та підтримка людей із особливими потребами. Зокрема, для людей із вадами зору актуальною є можливість отримання текстових або голосових описів навколишніх об'єктів, що сприяє інтеграції цих користувачів у цифровий світ.

По-друге, досягнення у сфері глибокого навчання, включаючи алгоритми виявлення та розпізнавання об'єктів (наприклад, YOLO, R-CNN), дозволяють створювати системи, здатні працювати в реальному часі з високою точністю. Однак ці алгоритми потребують адаптації для інтеграції з іншими технологіями, такими як обробка природної мови, щоб забезпечити автоматизовану генерацію описів об’єктів.

По-третє, використання віртуальних середовищ у веб-браузерах, зокрема на основі X3DOM, відкриває нові можливості для створення інтерактивних 3D-сцен. Водночас такі середовища мають обмеження, пов’язані з продуктивністю, якістю візуалізації та зручністю користувацької взаємодії. Розробка рішень, які враховують ці обмеження, є актуальним завданням.

Крім того, дослідження взаємодії користувачів із такими системами дозволяє краще зрозуміти їхні потреби та вподобання. Це сприяє створенню більш адаптивних і персоналізованих інтерфейсів, що відповідають сучасним вимогам інклюзивності та доступності.

Таким чином, актуальність теми дослідження визначається необхідністю інтеграції алгоритмів глибокого навчання, обробки природної мови та візуалізації у віртуальних середовищах для створення ефективних систем розпізнавання й опису об’єктів, що знаходять застосування у багатьох суспільно важливих сферах.

Мета дослідження – розробка та імплементація комплексного рішення для розпізнавання, аналізу та опису об’єктів у віртуальному середовищі, заснованого на застосуванні нейронних мереж та методів обробки природної мови.

Об’єкт дослідження – процеси аналізу, розпізнавання та опису об'єктів у віртуальному середовищі.

Предмет дослідження – моделі, алгоритми та методи використання нейронних мереж і природної мови для розпізнавання та опису об'єктів у віртуальному середовищі.

Завдання дослідження:

- Провести аналіз сучасних моделей глибокого навчання для розпізнавання об'єктів, зокрема моделей сімейства R-CNN та YOLO.

- Розробити архітектуру інтегрованої системи для виявлення та опису об’єктів у віртуальному середовищі.

- Імплементувати алгоритми обробки візуальних даних та генерації описів з використанням природної мови.

- Оцінити ефективність розроблених алгоритмів у різних типах задач (узагальнення та пошук).

- Проаналізувати обмеження запропонованого рішення та розробити рекомендації щодо його вдосконалення.

Методи дослідження

В роботі використано теоретичний аналіз існуючих моделей та алгоритмів обробки зображень і природної мови, експериментальна реалізація алгоритмів розпізнавання об'єктів і опису сцен, методи

статистичного аналізу, зокрема двофакторний дисперсійний аналіз і непараметрична кореляція, для оцінки ефективності алгоритмів.

Наукова новизна отриманих результатів

Запропоновано архітектуру інтегрованої системи для опису віртуальних сцен, що базується на поєднанні алгоритмів глибокого навчання та методів обробки природної мови.

Практичне значення результатів

Розроблене рішення може бути використане у створенні адаптивних систем навігації та взаємодії для людей із вадами зору, а також у сфері віртуальної реальності, зокрема в освітніх, ігрових і промислових додатках. Запропоновані алгоритми та методи дозволяють підвищити точність і продуктивність систем розпізнавання об'єктів у реальному часі.

Структура магістерської роботи. Робота складається зі вступу, трьох розділів та висновків. Загальний обсяг роботи становить 79 сторінок, і містить 49 рисунків, 1 таблицю, список використаних джерел із 51 найменування.

РОЗДІЛ 1. ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ВИКОРИСТАННЯ КОМП'ЮТЕРНОГО ЗОРУ ТА РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ

1.1. Опис області дослідження розпізнавання зображень

Мільйони людей не можуть зрозуміти навколишнє через проблеми із зором. Незважаючи на те, що вони можуть досить добре адаптуватися до своїх повсякденних справ, у них виникають певні проблеми з навігацією через об'єкти на сцені, і їм важко візуалізувати або описати навколишнє у своєму розумі. Крім того, сліпим або іншим вадам зору може бути складно визначити типи, кількість або розміри об'єктів у кімнаті, зображенні чи віртуальному середовищі. Короткозорість і далекозорість є двома категоріями порушень зору [1]. Нескориговані дефекти рефракції, вікові проблеми з очима, глаукома, катаракта, діабетична ретинопатія, трахома, помутніння рогівки або нелікована пресбіопія — це лише деякі причини погіршення зору [3]. Близько 80% людей із вадами зору або сліпими мешкають у країнах із середнім і низьким рівнем доходу, де вони не можуть придбати дорогі допоміжні технології.

Проблема, описана вище, є причиною яка спонукала створити рішення на основі штучного інтелекту, щоб допомогти з доступністю до віртуального середовища. Щоб покращити доступність сцени, ми вирішили реалізувати алгоритми розпізнавання об'єктів, щоб створити опис сцени природною мовою. Основним компонентом рішення, разом із алгоритмом бачення, є інші чотири алгоритми, які ми розробили, щоб відповісти на важливі питання, пов'язані з навколишнім середовищем. Це кількість, видатність, зліва направо та знизу вгору.

Вони будуть пояснені далі в наступних розділах. Використовуючи ці алгоритми, ми створили систему оповідання природною мовою, яка озвучує

(оповідає за допомогою системного звуку) сцену навколо користувача в будь-якому середовищі.

Перш ніж приступити до розробки рішення, ми спробували зрозуміти проблеми, з якими стикаються незрячі люди, прочитавши певні інтерв'ю. На основі інтерв'ю [3] ми дійшли висновку, що віртуальна реальність не дуже доступна для людей зі слабким зором. Зробити віртуальну реальність доступною для сліпих користувачів і користувачів зі слабким зором пов'язані значні проблеми. Найважчим аспектом віртуальної реальності для користувачів із слабким зором є інтерфейс користувача, вбудований у гарнітуру. Якщо ми говоримо про комерційно доступні інструменти або програмне забезпечення для спеціальних можливостей, які складаються з екранних луп, висококонтрастних панелей віртуальної реальності, програм зчитування з екрану або ігрових інтерфейсів, то наразі таких речей немає. SeeingVR був дослідницьким проектом Microsoft, але вони не вийшли зі стадії дослідження.

Удосконалення VR для спеціальних можливостей починається з пропозиції різних варіантів розміру тексту, де користувач може переважно використовувати повзунок для керування розміром тексту. Розмір тексту може допомогти не лише людям із значною втратою зору, але й людям, які носять окуляри. Збільшення та оповідання меню є критичними функціями для користувачів із слабким зором і сліпих.

Ще одна важлива функція для покращення доступності для людей зі слабким зором — повні шість ступенів відстеження від початку до закриття програми для гарнітури. Відстеження в трьох градусах означає, що ви можете дивитися лише вгору, вниз, ліворуч і праворуч. Шість градусів означає, що можна рухатися або нахилятися в будь-якому напрямку. Це більше нагадує реальний сценарій, у якому користувач може нахилятися або рухатися в будь-якому напрямку, який забажає, або наближатися до будь-якого елемента інтерфейсу користувача, як це можна робити в реальному житті.

Компанія Equal Entry проводить дослідження технологій віртуальної реальності, намагаючись зробити їх доступними для всіх людей з обмеженими можливостями. Вони вирішили провести дослідження середовища в Інтернеті, щоб оцінити навігацію людей у віртуальному середовищі. Говорячи про своє дослідницьке середовище, вони побудували кімнату відпочинку, конференцію та магазин. Середовище можна відкрити в браузері гарнітури. Об’єкт у середовищі мав альтернативні описи зображення, як це було раніше на двовимірних зображеннях у вигляді альтернативних текстів.

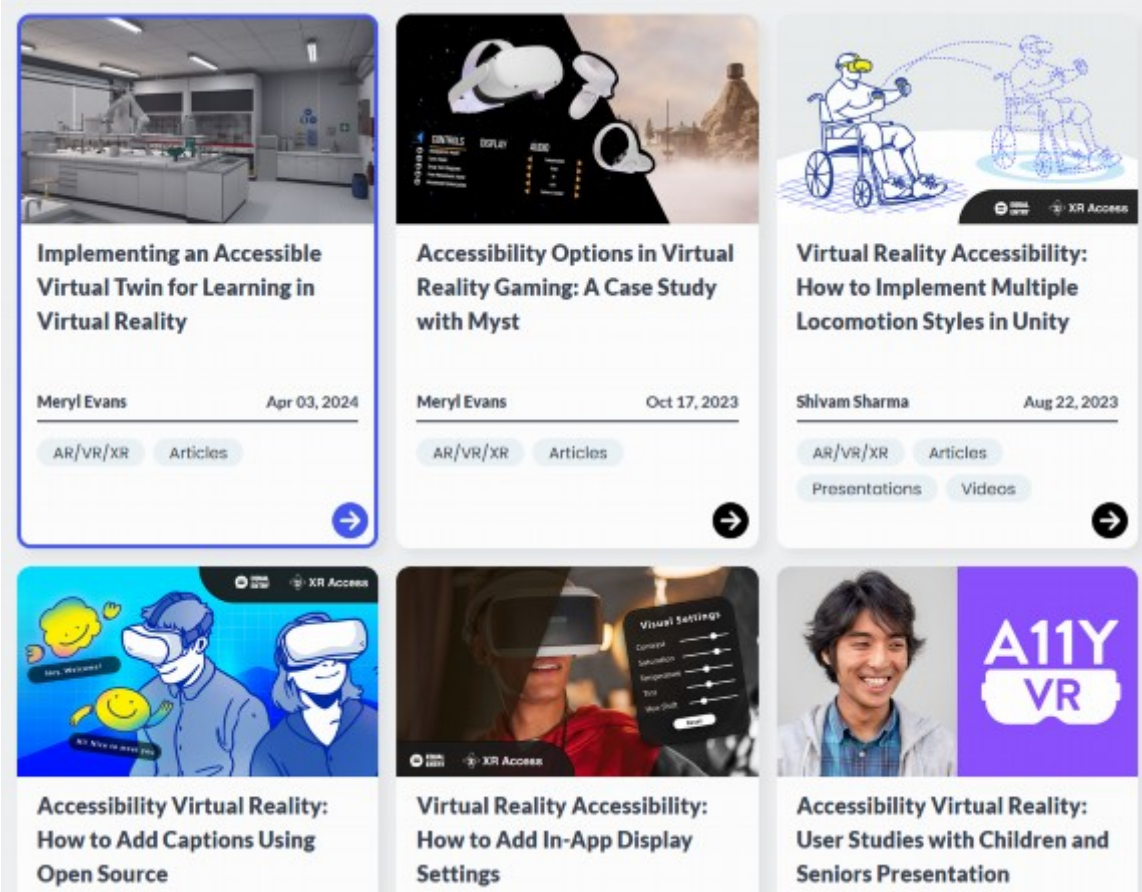

Рис. 1.1. Проекти Equal Entry щодо VR для людей з обмеженими можливостями

Вони намагалися знайти відповіді на два конкретних питання:

1. Чи може користувач перейти від входу до віртуального простору для проведення заходів і знайти різні кімнати та об’єкти?

2. Чи може користувач досліджувати та запитувати інформацію про набір 3D-об’єктів, які відображаються на трьох полицях?

Вони використовували гарнітуру Meta Quest 2 з 2 контролерами.

Опис 3D об'єктів відбувався нступним чином. Об’єкти сканувалися цифровим способом за допомогою процесу, який називається фотограмметрією. Вони досліджували різні метадані для кожного доступного об’єкта. У початковому проекті об’єкта вони передбачили назву, опис, ціну, розмір і вагу для кожного об’єкта.

Вони провели дослідження та виявили певні проблеми та рекомендували рішення на основі проблем.

1. Використовувати ручний регулятор для вибору невеликого об’єкта важко. Вказувати на певні об’єкти було непросто, оскільки, коли учасники вказували на об’єкт, вказівник рухався через природне тремтіння рук учасників. Тому іноді їм було незрозуміло, куди вони вказують. Вони знайшли рішення, коли вони збільшили область, на яку вказують, щоб вказівник не рухався легко.

2. Немає звукового сповіщення, коли об'єкт було захоплено. Користувач не знав і не отримував жодного відгуку у вигляді звукового сповіщення, коли він схопив певний предмет. Рішення полягає в тому, що захоплений звук повинен імітувати реальне захоплення фізичних об’єктів із подібним звуком.

3. Звуки ходьби не відрізняються за висотою і не є надійним вимірюванням відстані.

4. Відсутність звуку зіткнення зі стіною: у дослідженні не було жодного звуку, коли учасник зіткнувся зі стіною, учасник усе ще чув звук кроків, який був неправильним, оскільки шлях перешкоджав.

5. Забагато інформації оголошено для продуктів магазину: потрібно виконати певні кроки, щоб отримати відповідну інформацію або детальний опис об’єктів.

Підсумовуючи, пропонована робота спрямована на створення рішення з використанням моделі виявлення об’єктів (YOLOv3) у поєднанні з розробленим алгоритмом для підрахунку, помітності, зліва направо та знизу вгору. Згенерований опис навколишнього середовища буде передано користувачеві за допомогою природної мови.

1.2. Існуючі підходи розпізнавання об’єктів та оточення

У цьому розділі ми надаємо передумови та попередні роботи, які були виконані щодо цієї теми. Існують мобільні додатки та технології на основі розпізнавання об’єктів. Робота проводиться в просторі виявлення об'єктів; однак ці ідеї та продукти не зосереджені на розповіді сцени у віртуальному середовищі та не використовують жодних просторових алгоритмів і методів для покращення досвіду розуміння розповіді сцени.

Деякі роботи існують у широкій категорії сенсорної заміни. Наприклад, в дослідженні [8] автор, який повністю сліпий, досяг точної ехолокаційної здатності, використовуючи «клацання ротом» для незалежних завдань навігації, таких як їзда верхи та трекінг. Ніл Харбіссон, художник-дальтонік, створив гаджет, який перетворює інформацію про колір у звукові частоти. Голосова технологія [8] робить інтенсивну спробу перенести візуальне сприйняття на звук. Голосова система сканує кожен знімок камери зліва направо, що пов’язує висоту з кроком, а яскравість – з гучністю. Однак ці методи сенсорної заміни пов’язані зі складним процесом навчання. З іншого боку, ми використовуємо алгоритми візуального розпізнавання, які ведуть до більш прямих методів розуміння елементів у візуальній сцені.

В останніх роботах дослідники використовували методи озвучення, щоб полегшити доступ до візуальної інформації для людей із вадами зору. Наприклад, доступність графіків [9, 10], вивчення карт і графіки [11, 12], а також допомога в наданні обертових інструкцій для полегшення навігації в середовищах [13], надаючи докладну інформацію про функції та зміст

інтереси. Audemes є запропонованим рішенням яка базується на звукових піктограмах, щоб допомогти учням із вадами зору в навчальних усних письмових текстах [14].

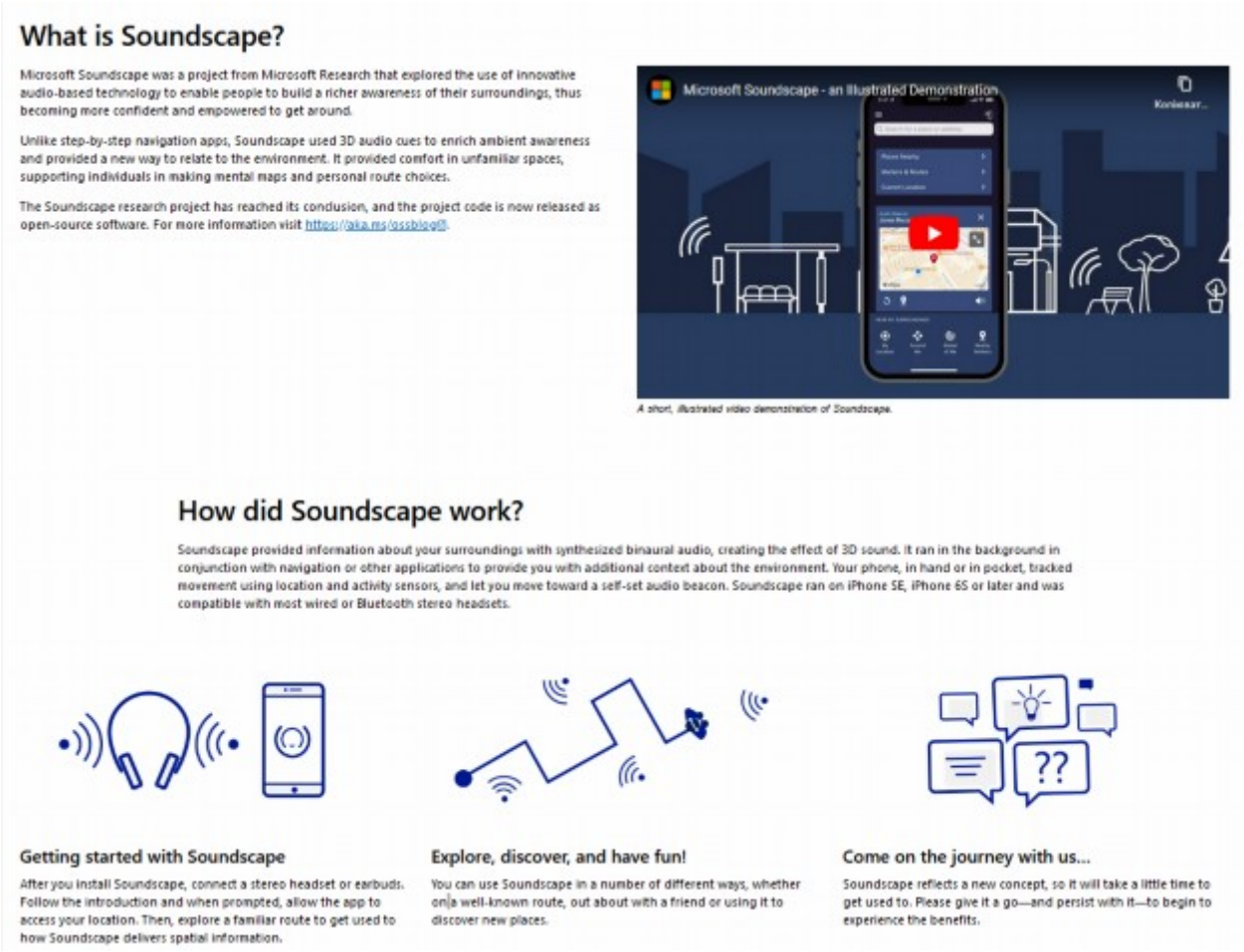

Рис. 1.2. Проект Microsoft Research, який досліджує використання інноваційної аудіотехнології

Microsoft Soundscape [15] — це продукт Microsoft Research, який допомагає людям впевненіше досліджувати довкілля на основі аудіотехнологій. Soundscape використовує 3D-аудіосигнали для підвищення обізнаності та нових способів спілкування та взаємодії з навколишнім середовищем. Soundscape може використовувати кожен у фоновому режимі, тому кожен може сміливо використовувати його разом з іншими програмами, такими як подкасти, аудіокниги, електронна пошта та GPS-навігація!

Те, що може зробити Soundscape:

• Встановіть аудіомаяк десь локально зі списку Nearby Places. Увімкніть спот і послухайте, як змінюється звук. Використання навушників змусить користувача відчути, що звук надходить із вибраного місця аудіомаяка.

• Збережіть маркер для місць, які ви часто відвідуєте, щоб було легше на майбутнє. Наприклад, місцеві автобусні зупинки, будинки тощо.

• Вирушайте на прогулянку та слухайте аудіо звуки навколо вас. Soundscape може викликати перехрестя та орієнтири, коли до них наближаєшся.

• Звук маяка може направити вас додому або в інші місця, які ви позначили.

В роботі [16] автор досліджував перетворення тексту в мову та використовував динаміки та звукові піктограми як стратегію озвучування оглядів звукових маршрутів. У статті зроблено висновок, що для отримання інформації про об’єкт інтересу підходять звукові значки.

Автори в [17] вивчали озвучення за допомогою різних засобів, таких як музика, копії та лірики, і намагався зрозуміти звуковий зв’язок, приписані значення та інтуїтивність. Але оцінка не проводилась для людей із вадами зору, тому ми не можемо багато сказати про результати, оскільки люди з проблемами зору та зрячі люди мають різні переваги щодо інтерфейсів користувача [18], а когнітивні вимоги також відрізняються під час їх використання. .

В дослідженні [19] надають описи які можна порівняти з нашими з точки зору озвучення. Було порівняно використання та міру вивчення навушників, звукових піктограм, списів і мови, яка використовується для зображення об’єктів. Однак у цьому творі немає обмежень за часом для тривалості звуків, що необхідно для представлення кількох речей за короткий проміжок часу.

Є кілька методів, які використовували техніку CV для виявлення внутрішніх і зовнішніх сцен і сповіщення користувача за допомогою голосу [20, 21] або вібротактильного зворотного зв’язку [22].

В [21] досліджується система Range-IT - допоміжний засіб для мобільності на основі камери, призначений для покращення можливостей користувачів білої тростини сприймати навколишні перешкоди/об'єкти під час пересування. Окрім 3D-камери глибини, система Range-IT має кілька компонентів (рис. 1.1).

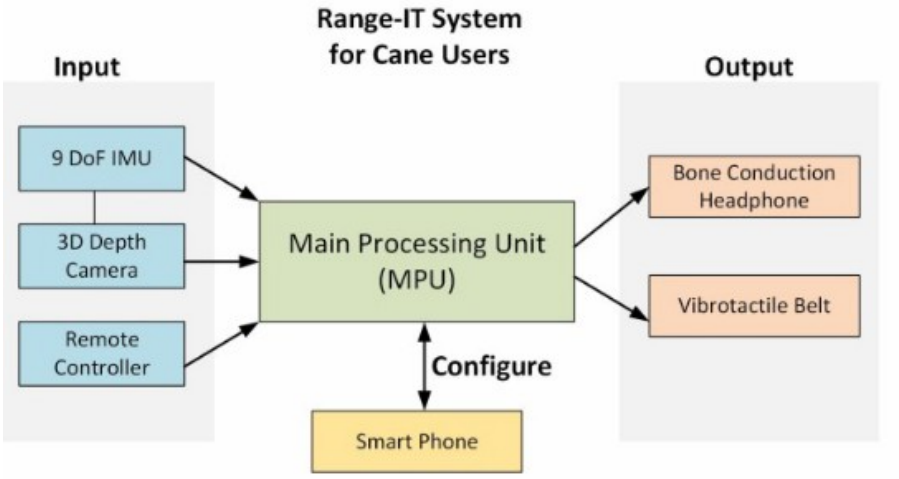

Рис. 1.3. Компоненти системи Range-IT

Система (рис. 1.3) складається з таких компонентів:

1. Вхідні дані.

1.1. 9 DoF IMU: Інерційний вимірювальний блок з 9 ступенями свободи. Він вимірює кутову швидкість та прискорення пристрою, що дозволяє визначити його орієнтацію та рух у просторі.

1.2. 3D Depth Camera: 3D-камера глибини. Вона створює тривимірну карту навколишнього середовища, вимірюючи відстань до об'єктів.

1.3. Remote Controller: Пульт дистанційного керування. Він дозволяє користувачеві керувати системою та налаштовувати її параметри.

2. Основний блок обробки.

2.1. Main Processing Unit (MPU): Головний обробний блок. Він отримує дані з датчиків, обробляє їх та генерує сигнали для вихідних пристроїв.

3. Вихідні дані.

3.1. Bone Conduction Headphone: Навушники з кістковою провідністю. Вони передають звук через кістки черепа, залишаючи вуха вільними для сприйняття звуків навколишнього середовища.

3.2. Vibrotactile Belt: Вібротактильний пояс. Він передає інформацію про перешкоди за допомогою вібраційних сигналів на різних ділянках пояса.

4. Конфігурація.

4.1. Smart Phone: Смартфон. Він використовується для налаштування системи та оновлення її програмного забезпечення.

5. Взаємодія компонентів

Вхідні дані з 9 DoF IMU, 3D камери глибини та пульта дистанційного керування надходять до головного обробного блоку (MPU). MPU обробляє ці дані та генерує сигнали для вихідних пристроїв - навушників з кістковою провідністю та вібротактильного пояса. Смартфон використовується для налаштування системи та зв'язку з MPU.

Крім того, поточні системи використовують розпізнавання мови та інші методи для розпізнавання сцен, такі як класифікація зображень для розпізнавання орієнтирів. Однак це робиться за допомогою камери телефону, що є недоліком цього підходу: сліпим людям буде важко триматися прямо, тримаючи камеру в домінуючій руці [23].

Інші дослідження намагалися включити системи комп’ютерного зору та немовні звуки, щоб допомогти людям у закритих приміщеннях [24]. Вони також використовували алгоритми розпізнавання облич, щоб ідентифікувати знайомих людей на задньому плані. Вони використовували методи озвучення, щоб допомогти користувачам знаходити ближчі об’єкти [25 - 29].

В роботах [30, 31] поділився досить актуальними методиками. Зокрема в [30] працювали над виявленням зовнішнього середовища за допомогою технологій комп’ютерного зору, які озвучуються за допомогою 3D-звуку.

Проблема полягає в тому, що техніка локалізує лише один клас елементів одночасно.

1.3. Технічний огляд 3D сцени

Тривимірність відноситься до чогось, що має ширину, висоту та глибину (довжину). Кожен день ми рухаємося в трьох вимірах у нашому фізичному світі.

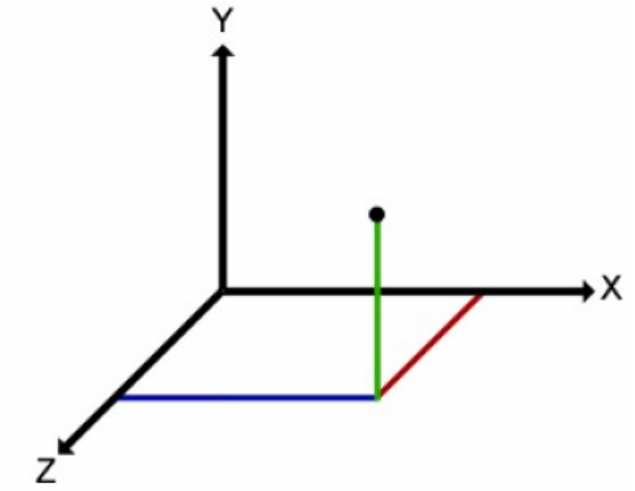

Рис. 1.4. Об’єкт а XYZ площині

У нас є тривимірне сприйняття, або відоме як сприйняття глибини, яке допомагає зрозуміти просторові відносини. Сітківка в кожному оці створює двовимірне представлення нашого середовища, дивлячись навколо.

Однак важливо підкреслити, що перегляд у 3D обома очима (стереоскопічне або бінокулярне бачення) — не єдиний варіант. Люди, які бачать лише одним оком (монокулярний зір), можуть відчувати навколишнє середовище в трьох вимірах і можуть не знати, що вони стереосліпі. У них відсутній один із інструментів, необхідних для перегляду в 3D; тому вони покладаються на інших, навіть не усвідомлюючи цього.

Люди використовують такі моноскопічні сигнали для сприйняття глибини:

• Паралакс : здається, що ближчі предмети рухаються швидше, ніж віддалені, коли ваша голова рухається з боку в бік.

• Знайомство з розмірами : якщо ви знаєте приблизний розмір об'єкта, ви можете використовувати його розмір, щоб оцінити, наскільки далеко він знаходиться. Подібним чином, якщо ви знаєте, що два предмети мають однаковий розмір, але один виглядає більш помітним, ніж інший, ви прийдете до висновку, що більший предмет знаходиться ближче.

• Повітряна перспектива : через випадкове розсіювання світла повітрям віддалені предмети здаються менш контрастними, ніж сусідні об’єкти. В результаті сторонні предмети виглядають менш насиченими за кольором і мають тонкий відтінок, який можна порівняти з фоном (зазвичай синім).

Рис. 1.5. Зір людини

Бажано імітувати якомога більше цих інструментів сприйняття, щоб відобразити 3D-оточення на плоскій (2D) поверхні, такій як екран дисплея. Хоча неможливо відтворити всі одночасно, у відео використовується комбінація. Відеокамера, наприклад, автоматично знімає перспективу з повітря та знайомий розмір. Зйомку з висоти пташиного польоту слід використовувати в CGI-сценах, щоб віддалені об’єкти виглядали менш

чіткими (це називається дистанційним туманом). Стереоскопічні сигнали, такі як бінокулярна диспарація та акомодація, можуть забезпечити суттєве покращення сприйняття глибини, особливо на малих відстанях.

1.4. Технологія розширюваної тривимірної (X3D) графіки

X3D Graphics — це безкоштовний відкритий стандарт для публікації, перегляду, друку та архівування інтерактивних 3D-моделей в Інтернеті. Консорціум Web3D [7] створює та підтримує стандарти X3D і HAnim. X3D — це набір стандартів ISO/IEC, що описує графік сцени, кодування файлів і API для 3D-сцен у реальному часі, включаючи моделі, вигляд, освітлення, анімацію та інтерактивність.

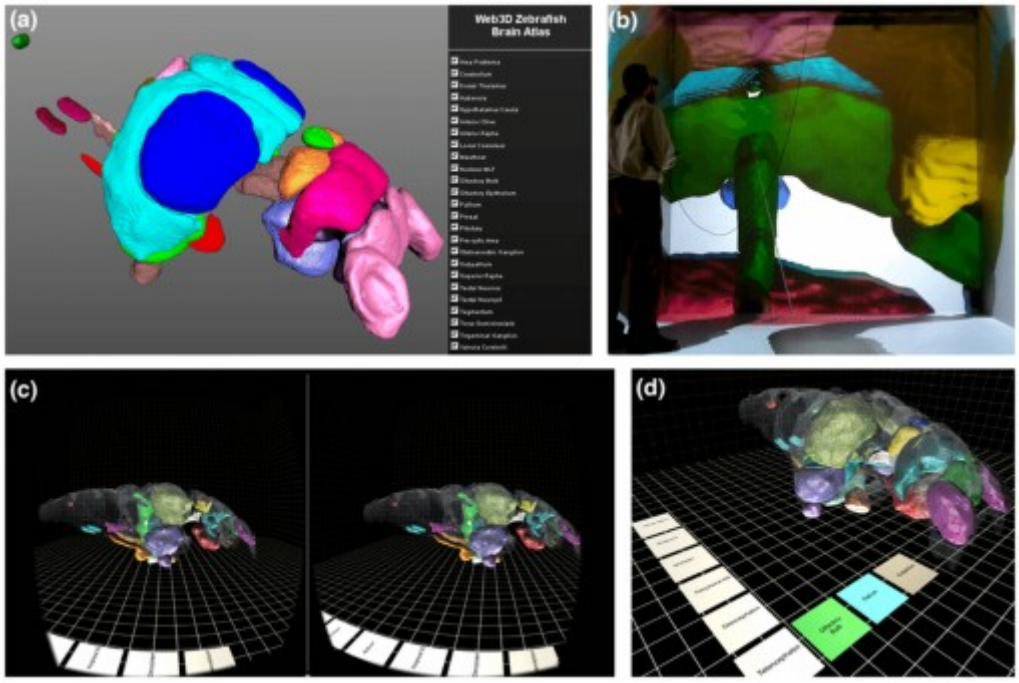

Рис. 1.6. Тривимірна візуалізація анатомії мозку рибки в X3D у веб-браузері, CAVE та дисплеї, встановленому на голові (HMD)

Ось деякі ключові особливості X3D:

- Розширюваність. X3D дозволяє розробникам додавати нові функції та можливості, не порушуючи сумісність з існуючими програмами та даними.

- Інтеграція з XML. X3D використовує XML для представлення даних, що полегшує інтеграцію з іншими веб-технологіями та програмами.

- Підтримка різних форматів. X3D підтримує різні формати даних, включаючи VRML, Open Inventor та бінарні формати.

- Широкий спектр застосувань. X3D використовується в різних галузях, включаючи віртуальну реальність, наукову візуалізацію, інженерне проектування, архітектуру та освіту.

X3D є потужною та універсальною технологією, яка має широкий спектр застосувань. Він є відкритим стандартом, який постійно розвивається та вдосконалюється.

Висновки до розділу

В даному розділі проведено дослідження предметної області використання комп'ютерного зору та розпізнавання зображень. Було проаналізовано основні аспекти комп'ютерного зору як міждисциплінарної галузі, яка поєднує методи машинного навчання, штучного інтелекту та обробки зображень. Основною метою є автоматизація процесів аналізу та інтерпретації візуальної інформації, що отримується за допомогою камер або інших сенсорів.

Розглянуто різні підходи до розпізнавання об'єктів, включаючи класичні методи та сучасні підходи. Також досліджено алгоритми для аналізу складних сцен, таких як тривимірне оточення. Описано особливості аналізу 3D сцен, включаючи методи створення тривимірних моделей, реконструкції об'єктів та їхнє позиціонування у віртуальному просторі. Розглянуто роль сенсорів, таких як LiDAR, стереокамери та глибокі камери, у створенні точних 3D моделей.

Досліджено можливості та перспективи використання формату X3D для представлення та інтеграції тривимірних сцен. X3D забезпечує сумісність із сучасними системами візуалізації, підтримує розширюваність і інтеграцію

з веб-технологіями, що робить його перспективним рішенням у сфері візуалізації та обробки 3D-даних.

Таким чином, даний розділ закладає теоретичну основу для розуміння сучасних можливостей комп’ютерного зору, розпізнавання об’єктів та аналізу 3D сцен, що є важливим для подальших досліджень і практичної реалізації відповідних алгоритмів.

РОЗДІЛ 2. МОДЕЛІ ТА АЛГОРИТМИ ЗАСТОСУВАННЯ НЕЙРОННИХ МЕРЕЖ ДЛЯ ПРОЦЕСІВ РОЗПІЗНАВАННЯ ОБ'ЄКТІВ ТА ЗОБРАЖЕНЬ

2.1. Розпізнавання об'єктів з використанням глибокого навчання

Щоб просто зрозуміти різницю між класифікацією зображення та локалізацією об'єкта, ми можемо уявити класифікацію зображення як позначення зображення відповідним ім'ям класу. Тоді як локалізація об'єктів — це створення обмежувальних рамок навколо різних об'єктів на зображенні. Справи стають складними, коли алгоритму виявлення об'єктів доводиться працювати з двома завданнями: намалювати обмежувальні прямокутники навколо цікавих об'єктів і позначити їх належним чином. У сукупності це називається розпізнаванням об'єктів.

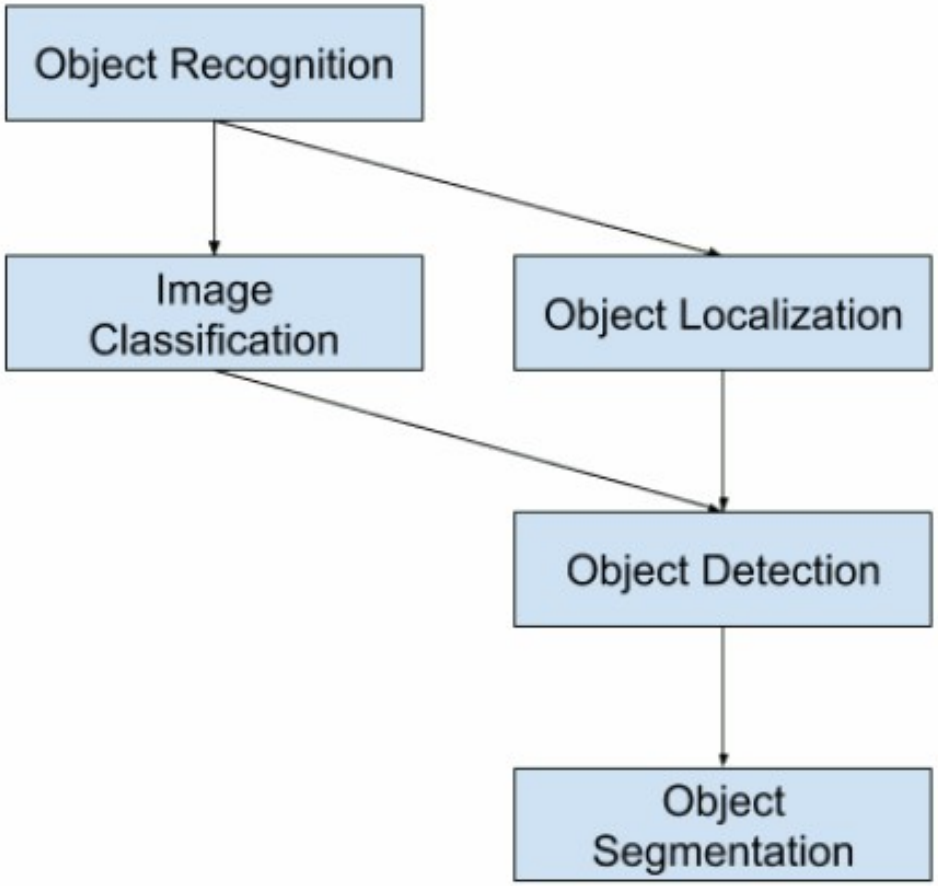

Рис. 2.1. Завдання комп'ютерного зору з розпізнавання об'єктів

2.1.1. Сімейство моделей Region-Based Convolutional Neural Network

R-CNN складається з наступних модулів [32]:

- Модуль 1: Генерація пропозицій регіонів. Створення та вилучення пропозицій регіонів, незалежних від категорії, таких як рамки-кандидати.

- Модуль 2: Вилучення ознак. Використання глибокої згорткової нейронної мережі для вилучення ознак з кожного регіону-кандидата.

- Модуль 3: Класифікатор. Класифікація ознак розпізнаного класу, наприклад, за допомогою лінійної моделі класифікатора SVM.

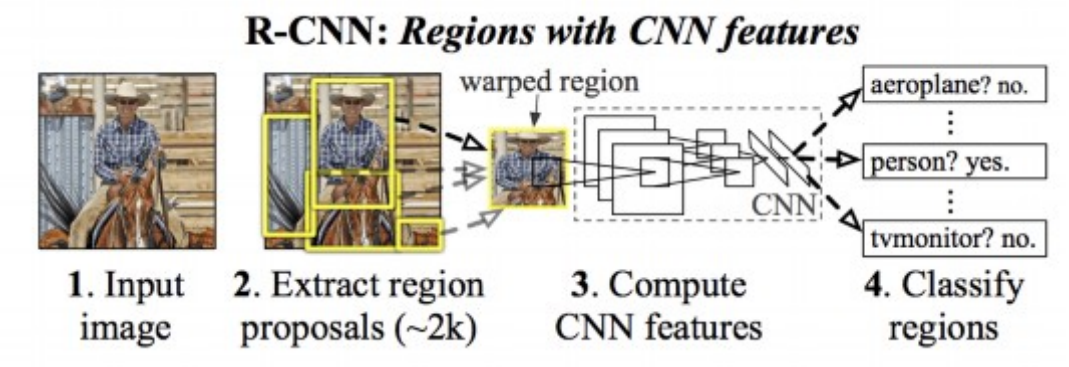

Рис. 2.2. Рисунок 2.5: Архітектура моделі R-CNN

Техніка комп'ютерного зору, заснована на "вибірковому пошуку", надає регіони-кандидати або рамки потенційних об'єктів на зображенні. Водночас, гнучкість дизайну дозволяє використовувати різні методи генерації пропозицій регіонів. Для вилучення ознак використовувалася глибока згорткова нейронна мережа AlexNet, яка перемогла в конкурсі ILSVRC-2012 з класифікації зображень. Модель CNN генерувала вектор з 4096 елементів для опису вмісту зображення, який потім подавався на лінійну SVM. Для кожного набору класів навчалася окрема SVM.

2.1.2. Модель fast R-CNN

Після великого успіху моделі R-CNN, в Microsoft Research запропоновано розширення для вирішення проблем зі швидкістю R-CNN

[33]. Дослідження починається з обговорення обмежень R-CNN, які можна сформулювати наступним чином:

- Навчання є багатоетапним процесом. Воно передбачає підготовку та роботу трьох окремих моделей.

- Навчання є дорогим з точки зору простору та часу. Навчання глибокої CNN на такій великій кількості рекомендацій регіонів для кожного зображення займає багато часу.

- Виявлення об'єктів є повільним. Робити прогнози за допомогою глибокої CNN на такій великій кількості ідей області є трудомістким.

Їхня робота прискорила вилучення ознак, але це був алгоритм кешування прямого проходу. Fast R-CNN пропонується як єдина модель, а не конвеєр, для навчання та виведення регіонів і класифікацій безпосередньо. Архітектура моделі використовує фотографію як вхідні дані для створення набору рекомендацій щодо області, які потім обробляються глибокою згортковою нейронною мережею.

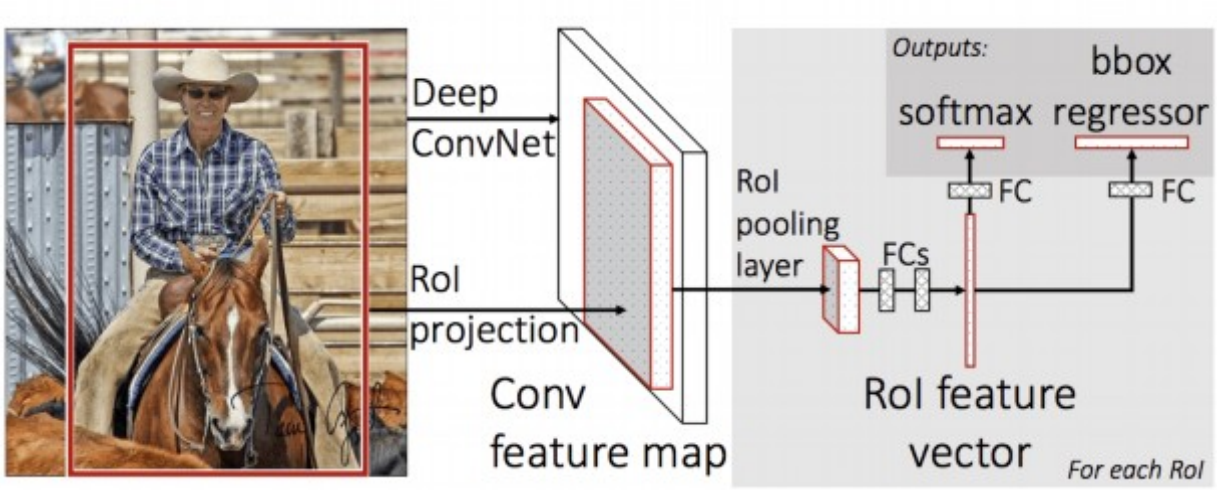

Рис. 2.3. Архітектура моделі Fast R-CNN

Для вилучення ознак використовується попередньо навчена CNN, така як VGG-16. Спеціальний шар, який називається шаром об'єднання регіонів інтересу (RoI Pooling), витягує ознаки, характерні для певного вхідного регіону-кандидата після глибокої CNN. Повністю зв'язаний шар інтерпретує

вихідні дані CNN. Модель розділяється на два виходи: один для прогнозування класу через шар softmax, а інший для рамки обмеження через лінійний вихід. Ця процедура виконується для кожного регіону інтересу на заданому зображенні. Архітектура моделі наведена на рисунку 2.3.

2.2. Підхід та алгоритми реалізації процесу розпізнавання об’єктів

Невід’ємними частинами рішення є модель виявлення об’єктів і рендеринг 3D-сцени в Інтернеті. Для вирішення проблеми розпізнавання об'єктів ми використовували YOLO.

YOLO (You Only Look Once) - це система розпізнавання об'єктів у реальному часі. Замість того, щоб сканувати зображення по частинах або генерувати регіони-кандидати, як це роблять інші методи (наприклад, R-CNN), YOLO розглядає зображення лише один раз, розділяючи його на сітку та передбачаючи рамки обмеження та ймовірності класів для кожної комірки сітки. YOLO дуже швидкий, деякі версії можуть обробляти відео в реальному часі з високою частотою кадрів і досягає високої точності виявлення об'єктів, конкуруючи з іншими передовими методами.

Він продемонстрував багатообіцяючі результати в області розпізнавання зображень. Щоб розробити наші власні чотири алгоритми, були параметри, які важливо витягнути із зображення. Завдання полягало в тому, щоб отримати дані про зображення, такі як оцінка достовірності, опорна точка (x, y), висота та ширина об’єкта, назва класу та область; вилучення цих важливих деталей після розпізнавання забезпечило нам гнучкість для виконання наших алгоритмів. Крім того, рішення для виявлення об’єктів із алгоритмами оповідання було інтегровано з 3D-сценою у веб-середовищі, відтвореним за допомогою технології X3DOM.

Для цього дослідження користувачів ми обмежилися двома гіпотезами. Ми називаємо наші гіпотези H1 і H2. Перш ніж пояснити нашу гіпотезу, ми повинні зрозуміти чотири алгоритми та дві категорії.

Таблиця 2.1.

Алгоритми

Object Centric	Environment Centric
1. Count 2. Prominence	3. LTR - Left to Right 4. BTP - Bottom to Top

"Об'єктно-орієнтований" алгоритм один - (Кількість): Цей алгоритм надає точну кількість різних об'єктів у середовищі метавсесвіту. Цей алгоритм підрахунку входить до категорії "об'єктно-орієнтованих" алгоритмів.

"Об'єктно-орієнтований" алгоритм два - (Визначність): Алгоритм визначності надає результати на основі площі об'єктів у метавсесвіті, від найбільшого до найменшого.

"Середовищно-орієнтовані" алгоритми: Ці алгоритми також називають просторовими алгоритмами.

"Середовищно-орієнтований" алгоритм три - (Зліва направо): Цей алгоритм називається "зліва направо". Алгоритм входить до категорії просторових середовищно-орієнтованих алгоритмів. Він використовує правила координатної геометрії для визначення положення об'єктів на сцені від лівої просторової точки до правої просторової точки.

"Середовищно-орієнтований" алгоритм чотири - (Знизу вгору): Цей алгоритм називається "знизу вгору". Алгоритм входить до категорії просторових середовищно-орієнтованих алгоритмів. Він використовує правила координатної геометрії для визначення положення об'єктів на сцені від нижньої (найближчої на сцені) просторової точки до верхньої (найдальшої) просторової точки.

Як ми вже згадували в наших гіпотезах H1 та H2, ми реалізували вищезгадані алгоритми, щоб з'ясувати, який набір алгоритмів буде краще працювати для якої конкретної категорії завдань. Перша категорія - це

завдання пошуку, а інша - завдання узагальнення. Обидва завдання, пошуку та узагальнення, мають набір питань. Кожен набір питань має різні метавсесвіти для дослідження та відповіді на зазначені в завданні питання. Користувач повинен мати можливість легко орієнтуватися на сцені та виконувати те, що його просять.

На основі літератури та нашого дизайну алгоритмів ми висуваємо гіпотезу, що:

H1: "Об'єктно-орієнтовані" алгоритми будуть кращими для завдань пошуку.

H2: "Середовищно-орієнтовані" алгоритми будуть кращими для завдань узагальнення.

2.3. Архітектура пропонованого рішення

2.3.1. Алгоритм виявлення об'єктів - YOLO

Дизайн архітектури складається з кількох компонентів, які сприяють функціональному наскрізному сервісу і представлений на рисунку 2.4. X3DOM у верхньому лівому куті — це стандарт, який допоміг нам відтворити 3D-сцену у веб-браузері.

Користувачі можуть переміщатися по 3D-сцені та натискати зображення за допомогою кнопки знімка екрана зеленого кольору. PNG створюється та надсилається до хмарної інфраструктури через API, що містить модуль машинного навчання, модуль алгоритму та систему оповідання. Модуль машинного навчання виводить JSON. JSON подається як вхідні дані для вибраного типу алгоритму, і генерується опис. Опис надсилається до модуля розповіді через API для розповіді користувачеві за допомогою мови.

На рисунку 2.4 зображено архітектуру алгоритму обробки 3D-сцени з веб-сторінки та генерації її опису, що включає в себе декілька етапів та модулів:

1. Вхідні дані:

- Користувач переглядає 3D-сцену на веб-сторінці за допомогою X3D-програвача.

- Користувач робить знімок екрану (скріншот) цієї сцени.

2. Обробка знімка екрану:

- Знімок екрану (у форматі PNG) відправляється на хмарну інфраструктуру через Flask API.

- Модуль машинного навчання. Зображення обробляється сервісом розпізнавання зображень (Image Captioning Service). Сервіс генерує текстовий опис зображення у форматі JSON.

- Модуль алгоритмів. JSON-файл з описом аналізується за допомогою обраного алгоритму. Об'єктно-орієнтовані алгоритми: Count – підраховує кількість об'єктів на сцені, а Prominence визначає порядок об'єктів за їх розміром (від найбільшого до найменшого).

- Середовищно-орієнтовані алгоритми. LTR (Зліва направо): Визначає порядок об'єктів зліва направо. BTT (Знизу вгору): Визначає порядок об'єктів знизу вгору.

- Модуль нарації. На основі обраного алгоритму та текстового опису генерується наратив (SCENE DESCRIPTION) - опис сцени з урахуванням обраного способу представлення інформації.

3. Вихідні дані:

На основі нарративу формується остаточний опис сцени, який може бути представлений у різних форматах: текстовий опис та звуковий опис (Sonification).

Додаткові можливості. API-POST & GET: дозволяє взаємодіяти з хмарною інфраструктурою, відправляти запити та отримувати результати.

Загалом, цей алгоритм дозволяє автоматично генерувати опис 3D-сцени з веб-сторінки, використовуючи машинне навчання та різні алгоритми аналізу. Це буде корисним для людей з вадами зору, для створення аудіогідів, для аналізу віртуальних середовищ тощо.

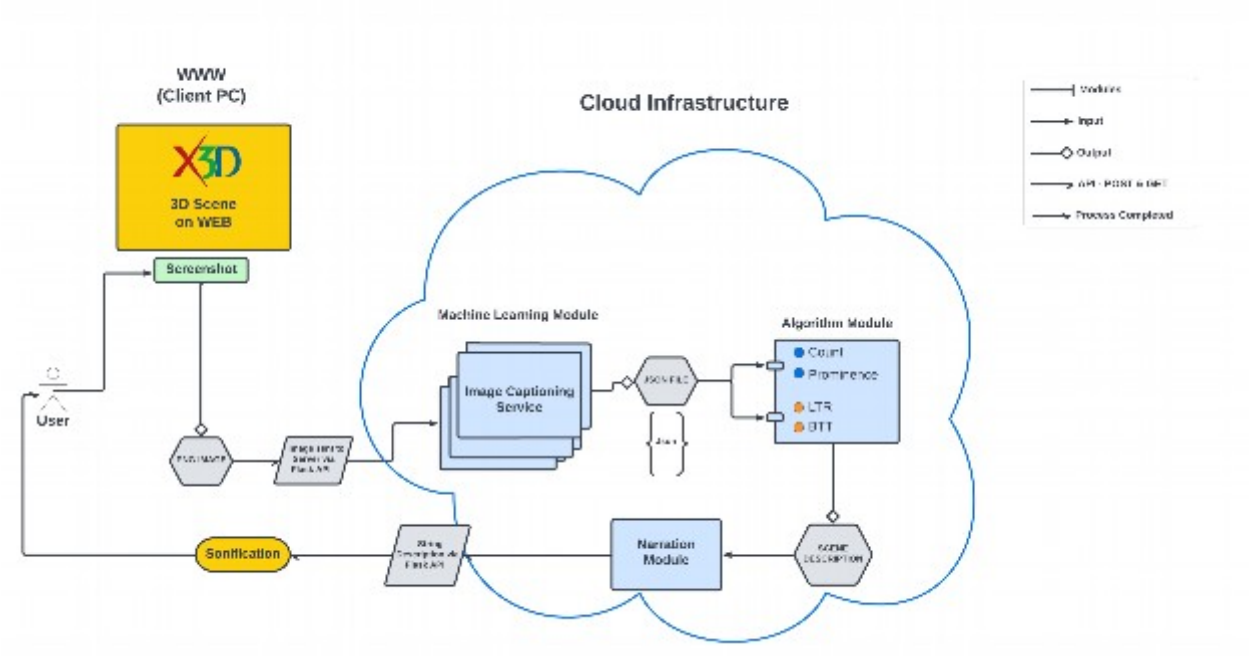

Рис. 2.4. Архітектура алгоритму обробки 3D-сцени

2.3.2. Модель YOLO

Алгоритм YOLO створює рамки для виявлення об’єктів. Обмежувальні прямокутники на зображеннях генеруються за допомогою R-CNN та інших алгоритмів для техніки пропозиції регіону перед виконанням класифікатора на запропонованих прямокутниках. Одна згортка нейронної мережі прогнозує багато обмежувальних прямокутників і ймовірності класів для цих коробок одночасно. Ця техніка робить Yolo нескладним. YOLO ефективний у розгортанні через відсутність складного конвеєра, оскільки проблема виявлення визначається як проблема регресії.

Крім того, коли частина класифікації завершена, реалізується постобробка для уточнення полів, видалення дублікатів і повторної оцінки балів залежно від інших елементів на зображенні [34].

Коли YOLO створює прогнози, він оцінює все зображення. Якщо ми порівняємо YOLO з підходом на основі ковзання та методами на основі пропозицій регіону, YOLO бачить цілісне зображення під час навчання та тестування. YOLO кодує контекстну інформацію про класи та їхній вигляд. У

порівнянні з Fast R-CNN YOLO створює менше половини кількості фонових помилок.

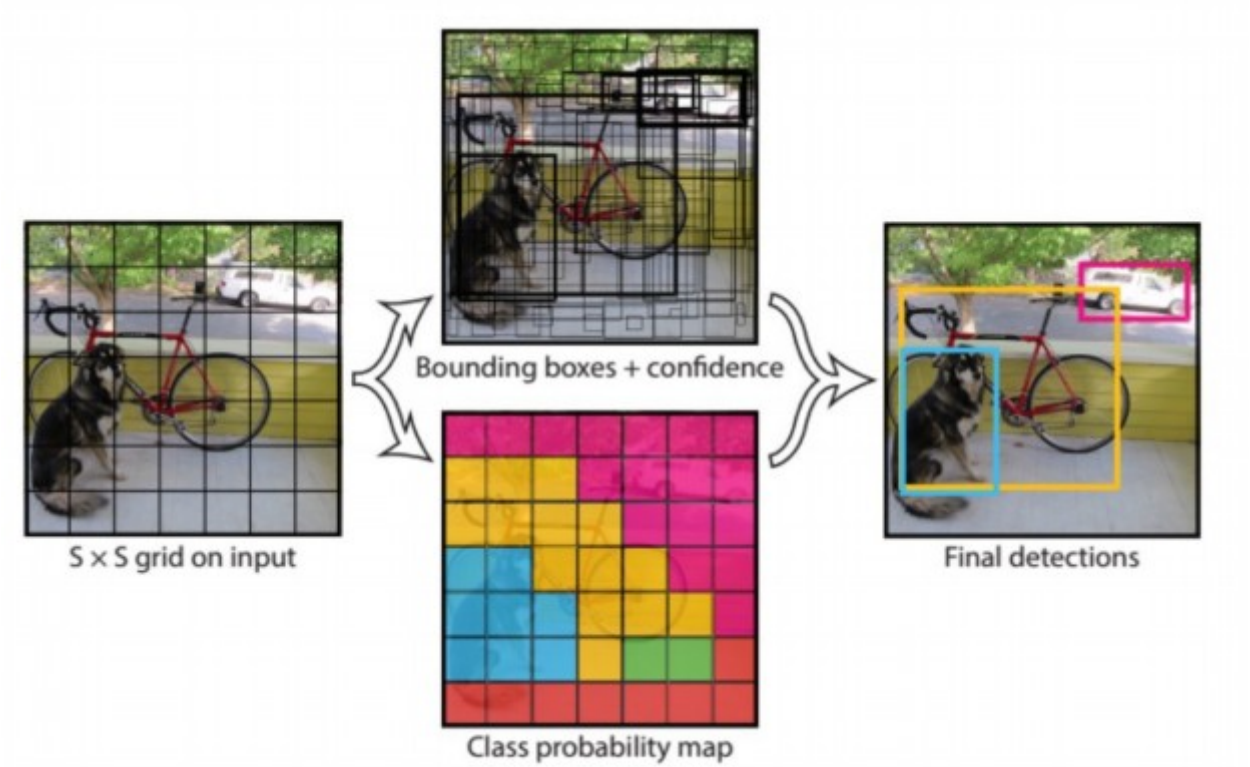

Рис. 2.5. Модель YOLO

Навчання YOLO можна узагальнити, оскільки воно вивчає представлення об’єктів таким чином. Коли YOLO навчався на різних зображеннях і творах мистецтва, він перевершив інші кращі алгоритми виявлення, такі як R-CNN і DPM. Завдяки своїй узагальненій природі, YOLO з меншою ймовірністю виявиться невдалим, якщо застосувати його до нових доменів або неочікуваних вхідних даних [33].

Замість того, щоб використовувати підхід пропозиції регіону, модель YOLO ділить зображення на сітки S x S. Кожна комірка сітки прогнозує B обмежувальні прямокутники та їхні оцінки достовірності, щоб визначити, чи потрапляє клас у рамки. Впевненість визначається як Pr(object) X IOU (truth, pred), що представляє впевненість класу в коробці та точність координат

коробки. У результаті кожна коробка повинна передбачити п'ять параметрів: x, y, w, h і впевненість.

Кожна клітинка сітки також передбачила Pr(Class(i)IObject). У результаті достовірність для кожного ящика дорівнює Pr(Class Object) X Pr(object) X IOU (truth) = Pr(Class (i)) X IOU (truth, pred). Тензор ASX SX (BX 5 + C) можна використовувати для представлення загальних змінних для прогнозування.

2.3.3. Згорточна нейронна мережа моделі YOLO

Автори реалізували цю модель у вигляді згорткової нейронної мережі та перевірили її на наборі даних виявлення VOC PASCAL [35]. Ранні згорткові шари мережі виділяють візуальні характеристики, тоді як повністю пов’язані шари передбачають вихідні ймовірності та координати. Дизайн мережі базується на моделі категоризації зображень GoogLeNet [36]. Мережа складається з 24 згорткових шарів, за якими слідують два пов’язані між собою шари. Замість початкових модулів GoogLeNet вони просто використовують 11 редукційних шарів, за якими слідують 3 X 3 згорткові шари [37]. Всю мережу показано на рисунку 2.6.

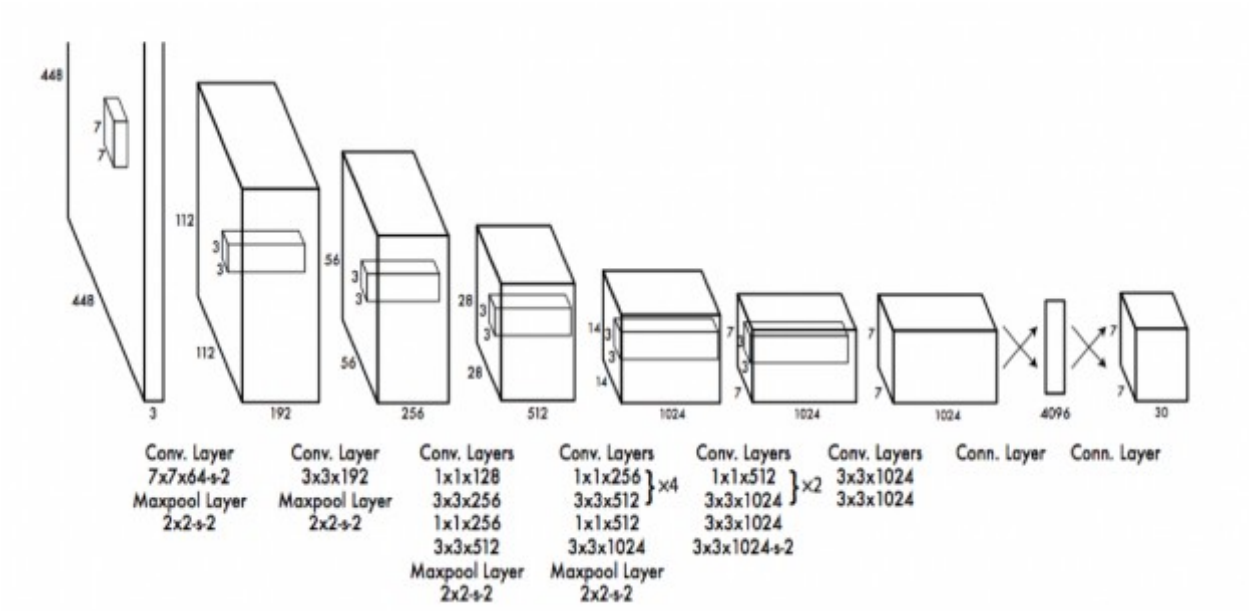

Рис. 2.6. Архітектура згорткової нейронної мережі для виявлення об'єктів

2.4. Процес навчання мережі

Для навчання згорткового шару використовувався набір даних ImageNet з 1000 класів [38]. Спочатку для попереднього навчання використовувалися перші двадцять згорткових шарів, а потім шар усередненого пулінгу та повністю зв'язаний шар. Модель навчається тижнями та досягає точності 5% у топ-5 при тестуванні на валідаційному наборі ImageNet 2012, що еквівалентно моделям GoogleNet. Вони використовували фреймворк Darknet для висновування та навчання.

Модель була модифікована для покращення виявлення об'єктів. Чотири згорткові шари та два повністю зв'язані шари були об'єднані з випадково призначеними вагами. Ймовірності та координати рамки обмеження прогнозуються на останньому шарі моделі. Для останнього шару використовується лінійна функція активації, а всі інші шари використовували функцію активації leaky rectified linear:

$$\phi(x) = \begin{cases} x, & \text{if } x > 0 \\ 0.1x, & \text{otherwise} \end{cases}$$

Вони оптимізують суму квадратів помилок на виході моделі. Вони також використовують суму квадратів помилок, оскільки її легко оптимізувати, хоча вона не повністю відповідає оптимізації середньої точності. Крім того, вона однаково зважує помилки локалізації та класифікації, що може бути небажаним. Крім того, багато комірок сітки на кожному зображенні порожні. Це призводить до того, що оцінки "впевненості" цих комірок падають до нуля, часто перекриваючи градієнт від комірок, які містять об'єкти. Це може призвести до нестабільності моделі та ранньої дивергенції навчання. Щоб вирішити цю проблему, вони збільшують втрати від прогнозів координат рамки обмеження, одночасно зменшуючи

втрати від прогнозів впевненості для порожніх рамок. Вони використовують два параметри λcoord λnoobj.

Під час навчання вони оптимізують наступну багатокомпонентну функцію втрат [39]:

$$\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]$$

$$+ \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$$

$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left(C_i - \hat{C}_i \right)^2$$

$$+ \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{noobj} \left(C_i - \hat{C}_i \right)^2$$

$$+ \sum_{i=0}^{S^2} 1_{ij}^{obj} \sum_{c \in classes} \left(p_i(c) - \hat{p}_i(c) \right)^2$$

Модель навчається з використанням PASCAL VOC 2007 протягом приблизно 135 епох на навчальних та валідаційних наборах даних 2012 року. Розмір пакету становить 64, імпульс - 0.9, а коефіцієнт згасання - 0.0005 під час навчання. Спочатку швидкість навчання для кожної епохи поступово збільшується з 10^{-3} до 10^{-2}. Початок з високої швидкості навчання призвів до проблеми нестабільного градієнта, що часто призводило до дивергенції моделі. Навчання продовжувалося з 10^{-2} протягом 75 епох, потім з 10^{-3} протягом 30 епох і, нарешті, з 10^{-4} протягом 30 епох. Для мінімізації перенавчання використовувалися методи відсіву та доповнення даних. Випадкове масштабування та перенесення до 20%.

2.4.1. Обмеження YOLO

Через те, що кожна комірка прогнозує дві рамки та має лише один клас, YOLO має просторові обмеження на прогнози [39]. Це обмежує кількість

об'єктів, які алгоритм може передбачити. Це створює проблеми з груповими об'єктами. Узагальнюваність моделі не дуже добра для нових або незвичайних співвідношень сторін. В кінці навчання на функції втрат, яка оцінює продуктивність виявлення, функція втрат розглядає помилки, які є подібними в малих і великих рамках обмеження.

Нарешті, хоча вони навчаються на функції втрат, яка апроксимує продуктивність виявлення, функція втрат розглядає однакові помилки в малих і великих рамках обмеження. Незначна помилка у великій рамці зазвичай нешкідлива, тоді як невелика помилка в маленькій рамці має набагато більший вплив на IOU. Неправильні локалізації є нашим найважливішим джерелом помилок.

2.4.2. Прогнозування меж обмеження

Для прогнозування меж обмеження використовуються кластери розмірів та опорні рамки [1, 2]. Мережа прогнозує чотири координати: tx, ty, tw та th. Якщо комірка зміщена відносно верхнього лівого кута зображення на (cx, cy), а апріорна рамка обмеження має ширину та висоту pw, ph, то прогнози виглядають наступним чином:

$$b_x = \sigma(t_x) + c_x \qquad b_y = \sigma(t_y + c_y) \qquad b_w = p_w e^{t_w} \qquad b_h = p_h e^{t_h}$$

Під час навчання використовується сума квадратів помилок втрат. Наприклад, якщо прогнозована координата має істинне значення ˆt*, тоді істинне значення градієнта мінус прогноз: ˆt* − t*

YOLOv3 використовує алгоритм логістичної регресії для прогнозування оцінки об'єкта для рамок обмеження. Він працює таким чином, що якщо апріорна рамка обмеження перекриває об'єкт істинного значення більше, ніж будь-яка інша, то вона повинна бути 1. Прогноз

ігнорується, якщо апріорна рамка обмеження не є найкращою, але перекриває об'єкт істинного значення більше, ніж поріг [40].

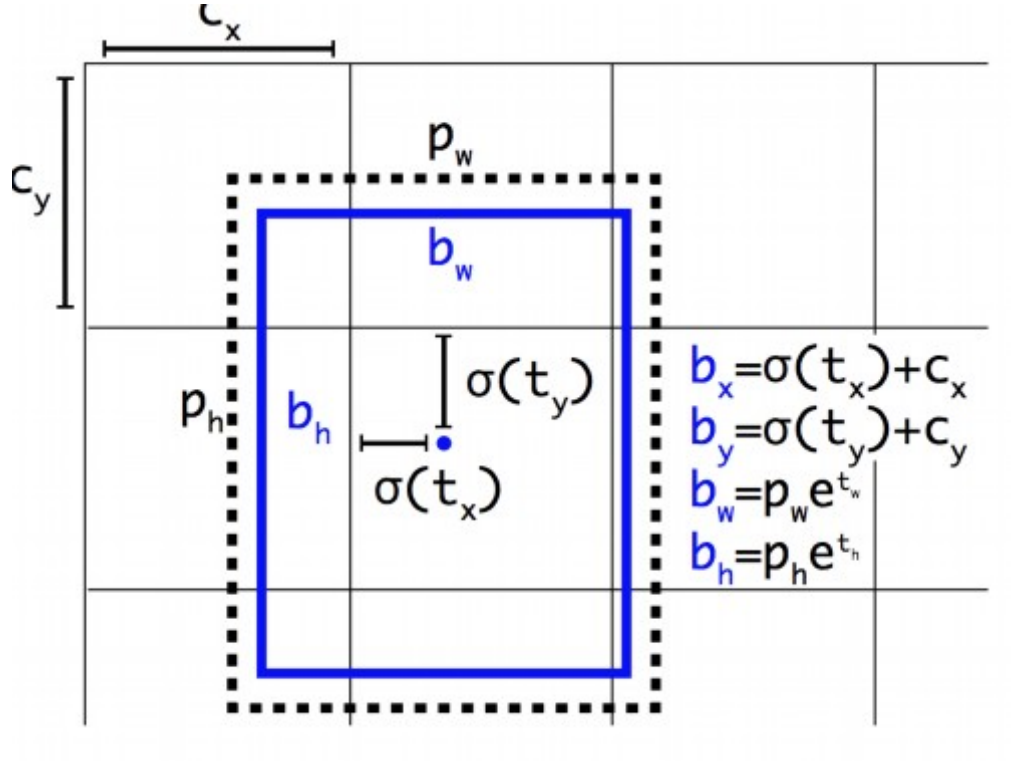

Рис. 2.7. Зображення рамок обмеження з розмірами [1, 2]

2.4.3. Середовище метавсесвіту, X3DOM та мікрофреймворк Flask

Для програми ми створили середовище метавсесвіту на основі Web3D. X3DOM [4 - 7] - це бібліотека з відкритим кодом, яку ми використовуємо для візуалізації сцен X3D, що складаються з віртуальних об'єктів X3D та GLTF. X3DOM дозволяє розробникам використовувати стандарт X3D для додавання 3D-середовища виконання до полотна WebGL у веб-браузері та об'єктної моделі документа (DOM), доступної для Javascript на стороні клієнта.

За веб-адресою у своєму браузері користувачі переміщуються по 3D-сцені за допомогою інтерактивної камери з перспективою; у будь-який момент вони можуть натиснути кнопку, щоб зробити знімок екрана сцени. Потім веб-сторінка надсилає ці знімки екрана зі сцени на сервер, де знаходиться служба розпізнавання зображень та модель, і повертається опис.

Flask - це мікрофреймворк для веб-розробки, реалізований за допомогою Python. Він не вимагає жодних спеціальних інструментів чи бібліотек. Flask не має шару абстракції бази даних, перевірки форм або будь-яких інших компонентів, де сторонні бібліотеки використовуються для забезпечення загальних функцій. Отже, щоб зробити нашу програму незалежною від численних сторонніх інтеграцій, ми використовували Flask для з'єднання наших компонентів через API.

2.5. JSON-структура та алгоритми виявлення об'єктів

Цей розділ містить детальну інформацію про результати виявлення об'єктів YOLO та чотири реалізовані та протестовані алгоритми.


```
{"0": {
        "Index": "0",
        "Object": "person",
        "Score": "0.9999959999999999",
        "AnchorPoint": "(4204, 3227)",
        "Height": "661",
        "Width": "302",
        "Area": "199622"
    }
},

{"1": {
        "Index": "1",
        "Object": "person",
        "Score": "1.0",
        "AnchorPoint": "(328, 3211)",
        "Height": "593",
        "Width": "348",
        "Area": "206364"
    }
}
```

Algorithm 1 Object Centric Algorithm - Count

Require: JSON OUTPUT
Ensure: "String" with scene description
description;
if length of JSON == 0 **then**
description += "Sorry, we can't find any relevant objects detected"; **return** description
end if
if length of JSON == 1 **then**
object = JSON[0];
description += "There is a" + ',' + object['Object']; **return** description
end if
if length of JSON > 1 **then**
description += "There are "
object_map = "Empty Dictionary"
loop: in JSON and add the objects to object_map
loop: in object map and count the objects based on key and value and add a string value to description; **return** description
end if

Algorithm 2 Object Centric Algorithm - Prominence

Require: JSON OUTPUT
Ensure: "String" with Prominence description
area_dictionary;
if length of JSON == 0 **then**
description += "Sorry, we can't find any relevant objects detected"; **return** description
end if
if length of JSON == 1 **then**
object = JSON[0];
description_two += "The" + " " + object['Object'] + " " + 'is the prominent object in the scene' **return** description_two
end if
if length of JSON > 1 **then**
description_three += 'The most prominent objects from biggest to smallest are' + " "
loop: in JSON and add the objects and the area to area_dictionary
sort: the dictionary in descending order
loop: in area_dictionary and add the object to description_three **return** description_three
end if

Algorithm 3 Environment Centric Algorithm - Left to Right

Require: JSON OUTPUT
Ensure: "String" with Left to Right description, result="The object in the image from left to right are";
lefttoright_dictionary;
if length of JSON == 0 **then**
description += "Sorry, we can't find any relevant objects detected"; **return** description
end if
if length of JSON == 1 **then**
object = JSON[0];
description_two += "The object in the image from left to right" + obj['Object'] **return** description_two
end if
if length of JSON > 1 **then**
loop: in JSON add objects and anchor point to the lefttoright_dictionary;
sort: lefttoright_dictionary
loop: in lefttoright_dictionary; and add the object to result; **return** result;
end if

Algorithm 4 Environment Centric Algorithm - Bottom to Top

Require: JSON OUTPUT
Ensure: "String" with Left to Right description, result="The object in the image from bottom to top are";
bottomtotop_dictionary;
if length of JSON == 0 **then**
description += "Sorry, we can't find any relevant objects detected"; **return** description
end if
if length of JSON == 1 **then**
object = JSON[0];
description_two +="The object in the image from bottom to top" + obj['Object']
return description_two
end if
if length of JSON > 1 **then**
loop: in JSON add objects and anchor point to the bottomtotop_dictionary;
sort: bottomtotop_dictionary
loop: in bottomtotop_dictionary; and add the object to result; **return** result;
end if

Висновки до розділу

У другому розділі розглянуто моделі та алгоритми, які базуються на застосуванні нейронних мереж для розпізнавання об'єктів і зображень, із акцентом на методах глибокого навчання.

Було проаналізовано сучасні підходи до розпізнавання об'єктів, зокрема сімейство моделей Region-Based Convolutional Neural Network (R-CNN) і його вдосконалені версії, такі як Fast R-CNN. Ці моделі продемонстрували високу точність у розпізнаванні, але мають певні недоліки щодо швидкості обробки. Представлено методологію реалізації процесу розпізнавання з урахуванням сучасних алгоритмів і підходів. Особливу увагу приділено поєднанню архітектур глибокого навчання з оптимізаційними техніками для підвищення продуктивності.

Розроблено архітектуру розпізнавання об’єктів на основі моделі YOLO (You Only Look Once). Проаналізовано її основні компоненти, включаючи згорточну нейронну мережу та алгоритм виявлення. Детально розглянуто принципи роботи YOLO, що забезпечують баланс між швидкістю та точністю. Описано процес підготовки та навчання мережі, зокрема врахування обмежень YOLO та методів їх подолання. Наведено підходи до прогнозування меж об'єктів і визначення їхніх координат у зображеннях.

Розглянуто можливості використання нейронних мереж у віртуальних середовищах, таких як метавсесвіт, із застосуванням інструментів X3DOM і Flask. Це підвищує інтерактивність і зручність впровадження нейронних мереж у реальні системи. Запропоновано використання JSON-структур для передачі даних і розроблено алгоритми, що ефективно інтегруються з нейронними мережами для виявлення об’єктів.

Таким чином, у розділі запропоновано оптимізовану архітектуру розпізнавання об'єктів, побудовану на базі сучасних підходів, а також розроблено алгоритми й середовище для інтеграції цього рішення у практичні додатки.

РОЗДІЛ 3. ІМПЛЕМЕНТАЦІЯ МОДЕЛЕЙ ТА АЛГОРИТМІВ ВИКОРИСТАННЯ ПРИРОДНОЇ МОВИ ТА НЕЙРОННИХ МЕРЕЖ ПРИ ОБРОБЦІ ЗОБРАЖЕНЬ ОБ'ЄКТІВ

3.1. Представлення дизайну інтерфейсу користувача системи виявлення об'єктів

У цьому розділі ми представляємо дизайн дослідження користувачів та наші методи оцінки. Мета дослідження - порівняти чотири різні алгоритми за рейтингами в випадкових наборах 3D-світів у мережі. Дані, зібрані в ході цього дослідження користувачів, будуть проаналізовані за допомогою комплексних статистичних методів, щоб відповісти на наші запитання та обґрунтувати нашу гіпотезу.

Дослідження користувачів було розроблено згідно з планом. Окрім усіх компонентів, таких як алгоритм виявлення об'єктів, X3DOM [4] (підтримує 3D-сцени в мережі) та API, що з'єднують систему від початку до кінця, я розробив систему опитування для проведення дослідження. Система опитування була створена для того, щоб показувати користувачам випадкові сцени з випадковими алгоритмами та випадково призначеними типами завдань.

Найголовніше, що кожен користувач мав різну комбінацію завдань, світів та алгоритмів, що робило дослідження повністю неупередженим. Додаток для цього дослідження використовував реалізацію X3DOM [5, 6] для запуску програми з 3D-світами і сумісний з інтернет-браузерами, такими як Chrome, Safari, Firefox, Opera тощо.

Таким чином, було проведено наступний дизайн дослідження користувачів, де суб'єкти взаємодіяли у веб-браузері настільного комп'ютера/ноутбука, щоб переміщатися по сцені, робити знімки, слухати описи та оцінювати їх.

Випадкове призначення питань починається з рандомізації типу завдання. Типи завдань:

1. Пошук. Користувачеві ставиться запитання, щоб знайти відповідні об'єкти на сцені. Відповідними об'єктами може бути будь-який об'єкт, присутній в середовищі. Наприклад: "Знайдіть стілець/стільці на сцені та зробіть знімок".

2. Узагальнення. Це друге завдання, де користувача просять "Отримати ширший вигляд сцени та зробити знімок". Користувач зменшує масштаб і знаходить точку огляду, яку можна вважати ширшим виглядом, де він може чітко бачити кілька об'єктів.

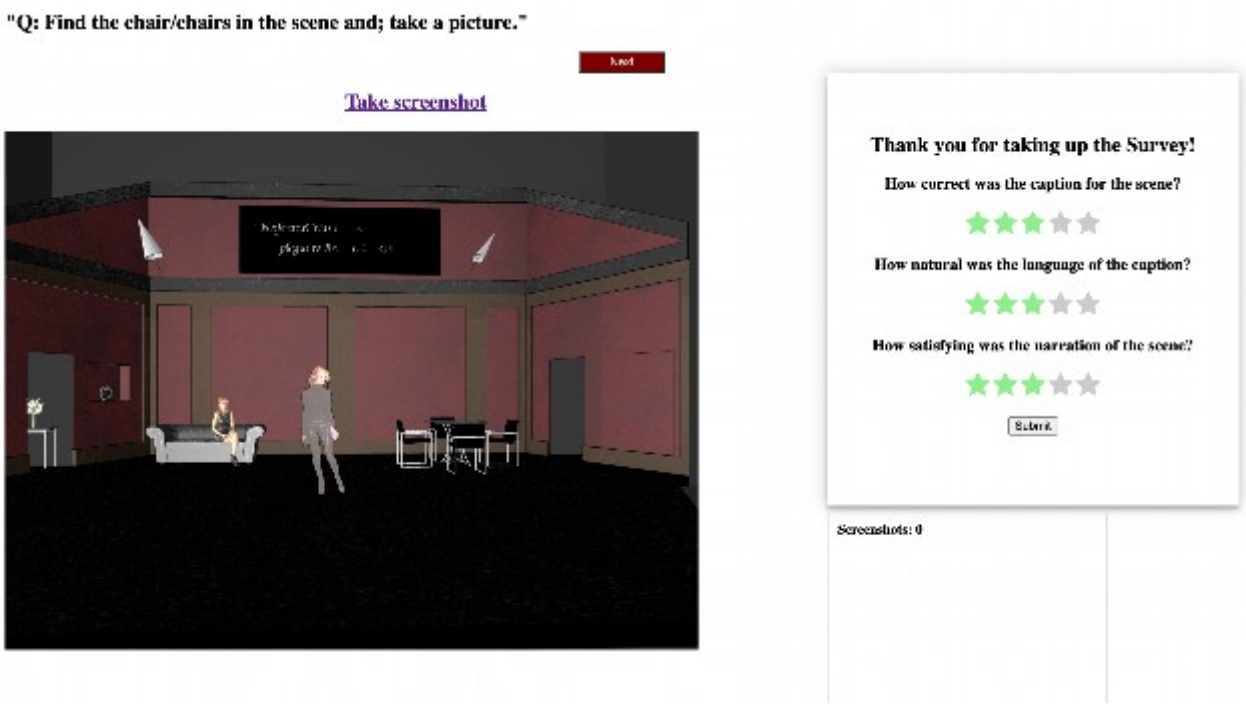

Рис. 3.1. Тип завдання: Пошук

На зображенні вище представлено один зі світів. Всього є вісім світів для 24 випробувань, проведених користувачами. Після прослуховування описів, заснованих на чотирьох різних алгоритмах, користувачеві було поставлено три запитання:

1. Наскільки правильним був опис сцени?

2. Наскільки природною була мова опису?

3. Наскільки задовільним було озвучення сцени?

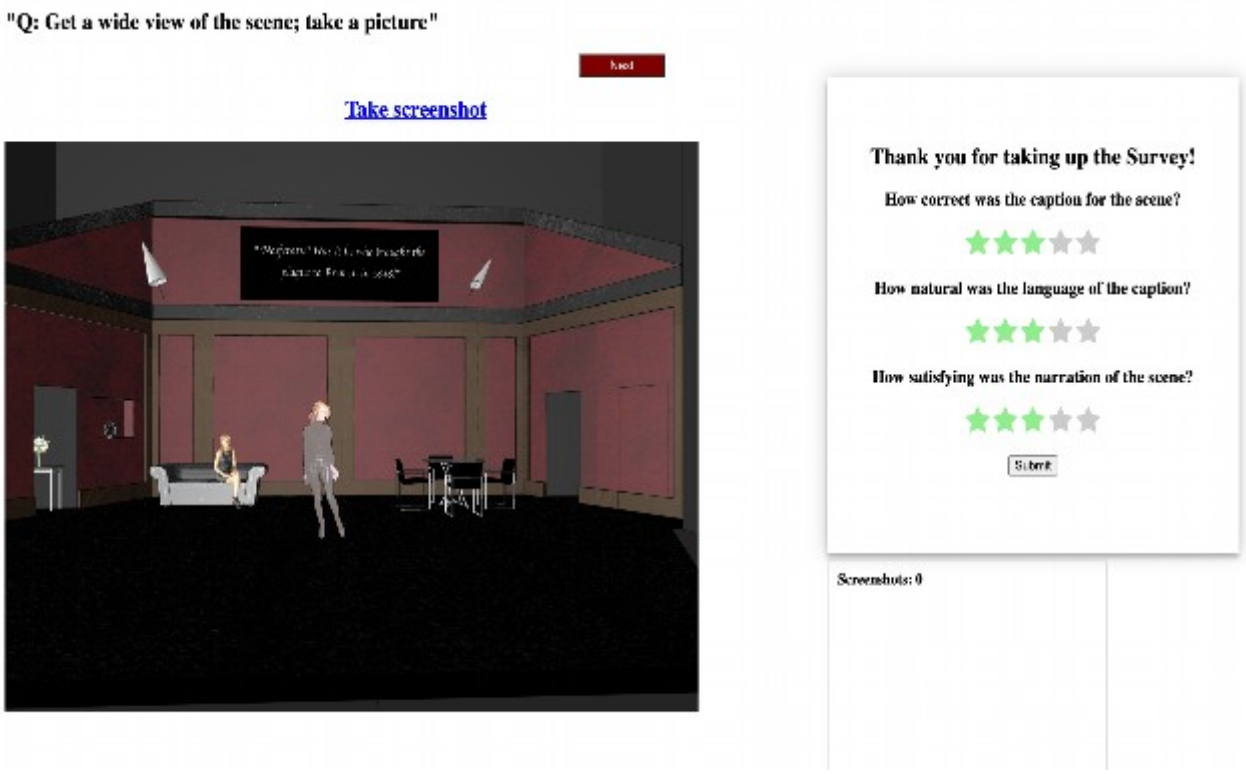

Рис. 3.2. Тип завдання: Узагальнення

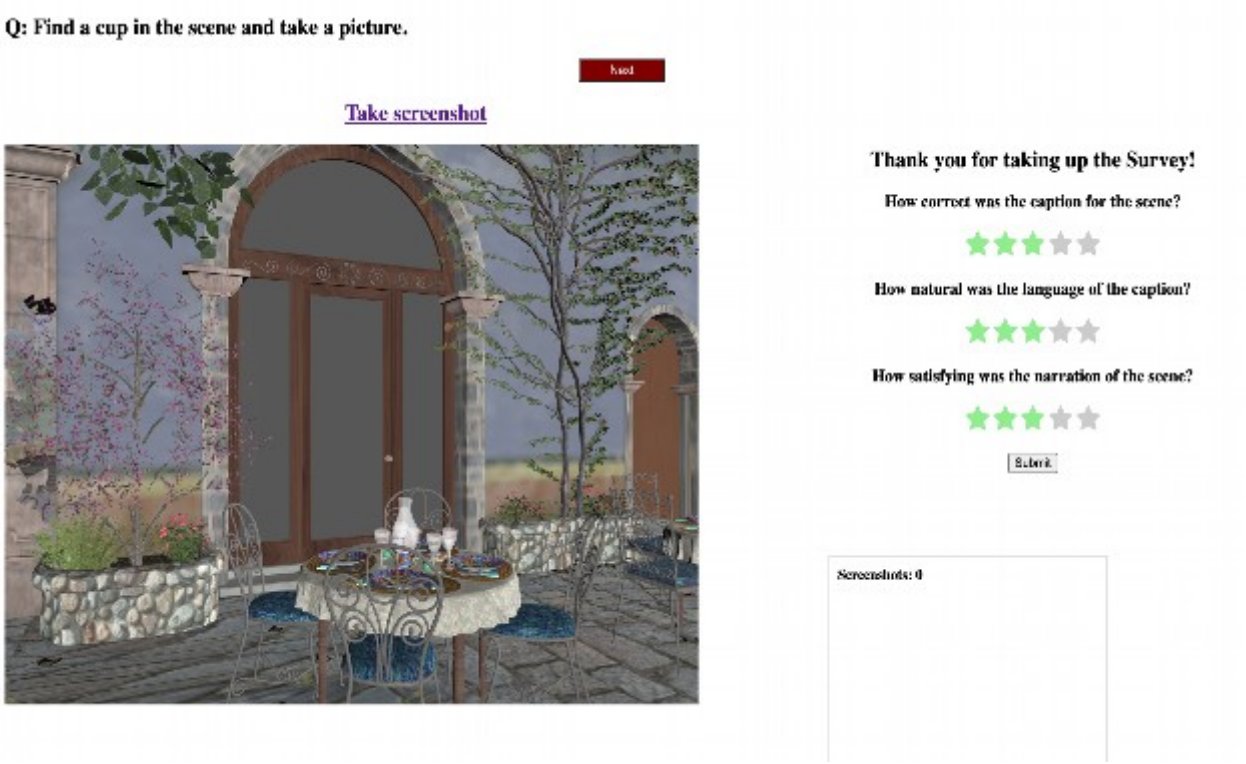

Рис. 3.3. Тип завдання: пошук

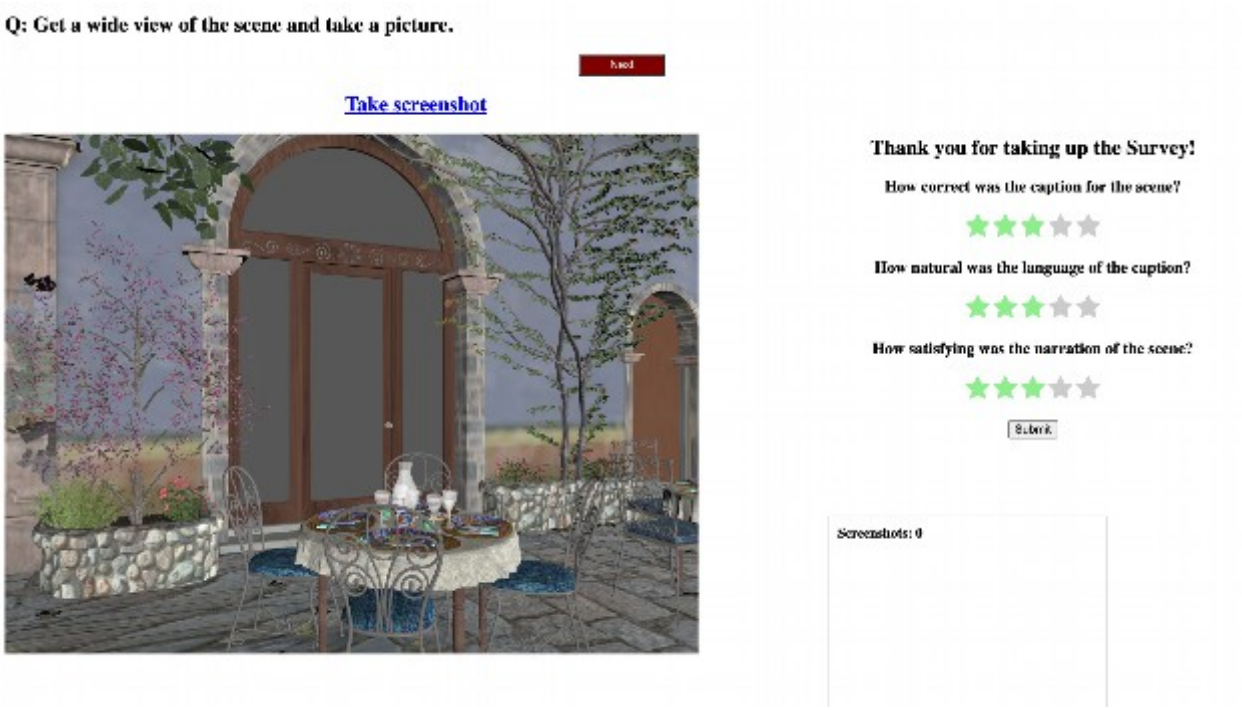

Рис. 3.4. Тип завдання: Узагальнення

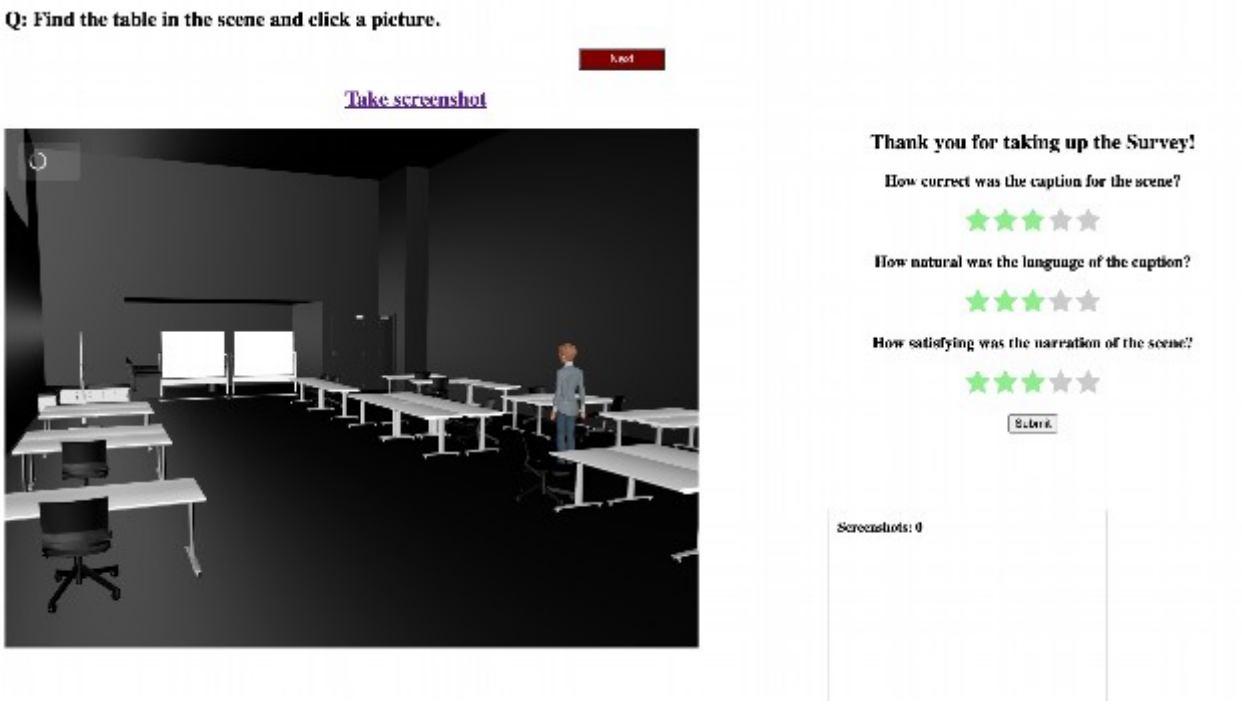

Рис. 3.5. Тип завдання: пошук

Як зазначалося, користувач має 24 випробування, і в кожному випробуванні була обрана випадкова сцена з загальної кількості восьми 3D-

світів. У кожному випробуванні також був обраний випадковий алгоритм для опису.

Описи алгоритмів для наведених вище знімків екрана:

- КІЛЬКІСТЬ: НА СЦЕНІ ТРИ СТІЛЬЦІ. (рис. 3.2)

- ВИЗНАЧНІСТЬ: ОБ'ЄКТИ ВІД НАЙБІЛЬШОГО ДО НАЙМЕНШОГО: ОБІДНІЙ СТІЛ, СТІЛЕЦЬ, ТАРІЛКА ТА КЕЛИХ ДЛЯ ВИНА (рис. 3.3).

- ЗНИЗУ ВГОРУ: ОБ'ЄКТИ ЗНИЗУ ВГОРУ: СТІЛЕЦЬ, ОБІДНІЙ СТІЛ, ТАРІЛКА ТА КЕЛИХ ДЛЯ ВИНА. (рис. 3.4)

- ЗЛІВА НАПРАВО: ОБ'ЄКТИ ЗЛІВА НАПРАВО: СТІЛЕЦЬ, СТІЛ, ЛЮДИНА (рис. 3.5).

Імітаційно дослідження користувачів проводилося онлайн, тому під час опитування, коли користувачі намагалися робити знімки згідно з поставленими запитаннями за допомогою кнопки "Зробити знімок екрана", їм спочатку доводилося переміщатися по сцені, щоб зробити правильний знімок. Тому було проведено попереднє навчання, щоб полегшити користувачам навігацію по сцені. У процесі навчання користувачі навчалися комфортно переміщатися по сцені та виконувати кроки, необхідні в опитуванні.

Елементи керування були пояснені користувачам, а також були доступні в письмовій формі. Ключові інструкції, які були надані кожному користувачеві, стосувалися 3D-навігації та керування сценою:

- Ліва кнопка миші - переміщення по сцені (збільшення масштабу)

- Права кнопка миші - віддалення від місця розташування (зменшення масштабу)

- E - режим огляду (політ по сцені)

- W - режим ходьби

- U - прямування вгору по сцені, 90 градусів

Процедура дослідження

- Перейти до опитування.

- Показати 3D-сцену з формою оцінювання, і це повторюється з випадковими сценами та алгоритмами 24 рази для користувача.

- Оцінити систему після прослуховування опису в кожному випробуванні.

- Завершити дослідження.

3.2. Результати імплементації моделей та алгоритмів

3.2.1. Двофакторний дисперсійний аналіз з повторними вимірюваннями

Зібраний набір даних має два фактори (незалежні змінні): завдання та алгоритми. Завдання має два рівні: пошук та узагальнення, тоді як алгоритми мають чотири рівні (1-4). Однак дослідження користувачів, яке було проведено, має кілька записів оцінок від одного користувача, хоча для різних завдань, але ці завдання повторювалися. Тому, щоб зберегти точність результатів, ми вирішили використати двофакторний дисперсійний аналіз з повторними вимірюваннями. Ми можемо одночасно оцінити, як тип завдань та алгоритми впливають на оцінки користувачів.

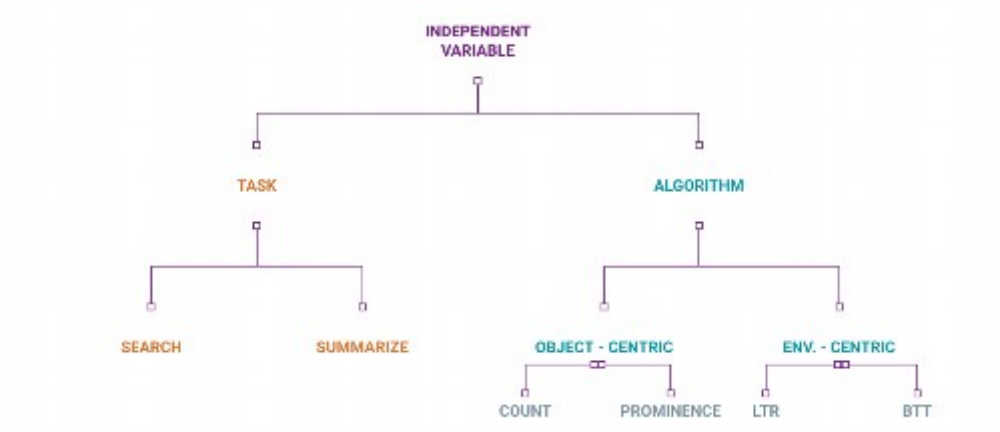

Рис. 3.6. Дерево змінних дисперсійного аналізу (ANOVA - Analysis of Variance)

За допомогою двофакторного дисперсійного аналізу ми можемо перевірити три гіпотези. Нульова гіпотеза:

1. Немає значного впливу типу завдання на оцінки користувачів щодо опису.

2. Немає значного впливу алгоритмів на оцінки користувачів щодо опису.

3. Немає значного впливу як завдань, так і алгоритмів на оцінки користувачів щодо опису.

Значення (p), отримане з аналізу ANOVA для завдання, алгоритмів та взаємодії, є статистично значущим, коли ($p < 0,05$).

	user_id	trial	task	algo	world	rating_one	rating_two	rating_three
0	20	2	summarize	3	2	5	5	5
1	20	3	search	4	4	5	4	5
2	20	4	summarize	1	8	5	5	5
3	20	17	search	1	1	5	5	5
4	20	8	summarize	4	5	5	5	1
...	...	...	...	...	...	...	...	...
972	35	14	search	1	3	1	3	3
973	35	13	summarize	2	3	3	3	3
974	35	12	search	3	7	1	3	3
975	35	24	summarize	1	7	4	3	4
976	35	1	search	3	1	4	2	3

Рис. 3.7. Дані опитування

ANOVA дозволяє нам визначити статистично значущу групу, яка впливає на рейтинги. Однак, щоб рухатися вперед з результатами ANOVA з повторними вимірюваннями, ми перетворюємо фрейм даних на 8 стовпців та три різні набори даних для кожного рейтингу.

Назви стовпців використовуються в таблицях та результатах, тому необхідно розуміти, що вони означають.

Пошук - t1 або завдання 1

t1a1: завдання 1 та алгоритм 1

t1a2: завдання 1 та алгоритм 2

t1a3: завдання 1 та алгоритм 3

t1a4: завдання 1 та алгоритм 4

Узагальнення - t2 або завдання 2

t2a1: завдання 2 та алгоритм 1

t2a2: завдання 2 та алгоритм 2

t2a3: завдання 2 та алгоритм 3

t2a4: завдання 2 та алгоритм 4

Multivariate Tests[a]

Effect		Sig.	Partial Eta Squared	Noncent. Parameter
algo_factor	Pillai's Trace	.003	.113	14.749
	Wilks' Lambda	.003	.113	14.749
	Hotelling's Trace	.003	.113	14.749
	Roy's Largest Root	.003	.113	14.749
task_factor	Pillai's Trace	.795	.001	.068
	Wilks' Lambda	.795	.001	.068
	Hotelling's Trace	.795	.001	.068
	Roy's Largest Root	.795	.001	.068
algo_factor * task_factor	Pillai's Trace	<.001	.164	22.760
	Wilks' Lambda	<.001	.164	22.760
	Hotelling's Trace	<.001	.164	22.760
	Roy's Largest Root	<.001	.164	22.760

Рис. 3.8. Дисперсійний аналіз завдання та алгоритми за рейтингом один - Правильність опису

З рисунка 3.8 ми бачимо, що фактор алгоритму має значний вплив на рейтинг один незалежно, тоді як фактор алгоритму та фактор завдання в поєднанні також мають значний вплив на рейтинг один. Але фактор завдання сам по собі не показує значного впливу на рейтинги.

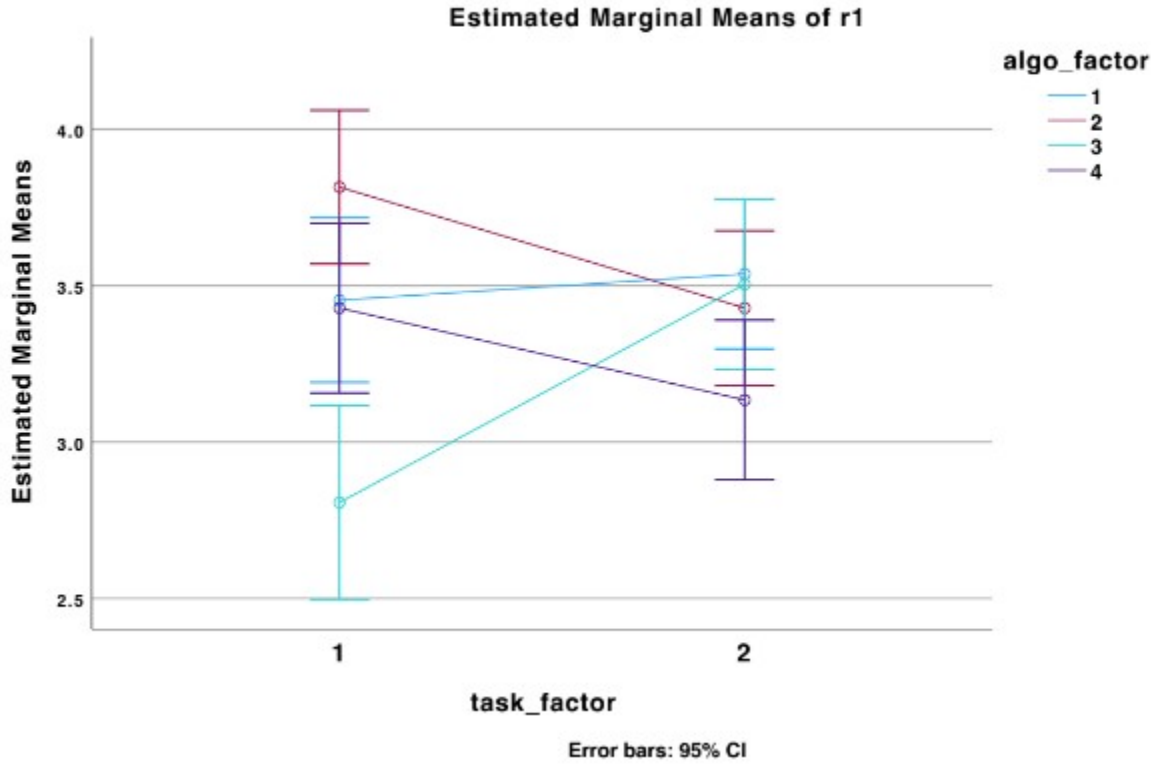

Рис. 3.9. Продуктивність завдання проти алгоритму для рейтингу правильності

Графічне представлення продуктивності алгоритмів відносно типів завдань допомагає нам зробити висновок, що:

- Продуктивність алгоритму 1 зросла з точки зору рейтингу для завдань узагальнення порівняно із завданнями пошуку.

- Продуктивність алгоритму 2 з точки зору рейтингів була високою для завдань пошуку, але значно знизилася порівняно із завданнями узагальнення.

- Продуктивність алгоритму 3 з точки зору рейтингів демонструє різке зростання від завдань пошуку до завдань узагальнення, хоча для завдань узагальнення рейтинги були вищими.

- Алгоритм 4 показав кращі результати для типу завдання пошуку порівняно із завданнями узагальнення.

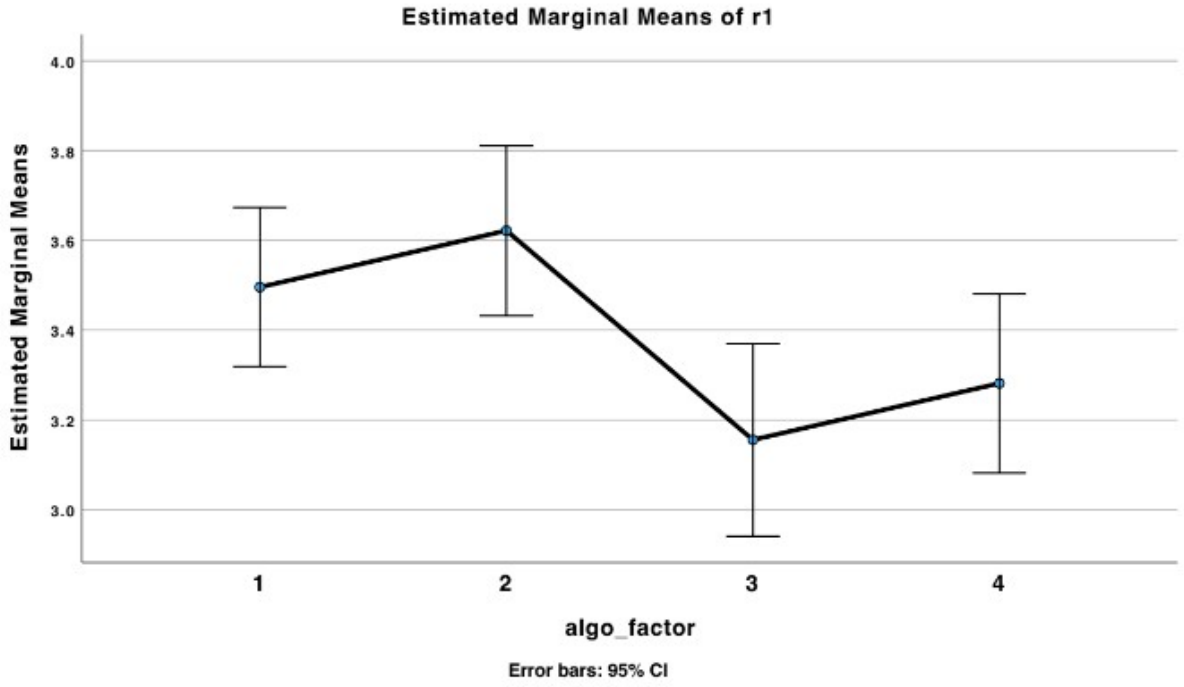

Рис. 3.10. Продуктивність алгоритму для рейтингу правильності

Рисунок 3.10 показує, що алгоритм 2 мав кращу продуктивність рейтингу порівняно з іншими алгоритмами.

Multivariate Tests[a]

Effect		Sig.	Partial Eta Squared	Noncent. Parameter
algo_factor	Pillai's Trace	<.001	.359	64.928
	Wilks' Lambda	<.001	.359	64.928
	Hotelling's Trace	<.001	.359	64.928
	Roy's Largest Root	<.001	.359	64.928
task_factor	Pillai's Trace	.485	.004	.492
	Wilks' Lambda	.485	.004	.492
	Hotelling's Trace	.485	.004	.492
	Roy's Largest Root	.485	.004	.492
algo_factor * task_factor	Pillai's Trace	.036	.071	8.814
	Wilks' Lambda	.036	.071	8.814
	Hotelling's Trace	.036	.071	8.814
	Roy's Largest Root	.036	.071	8.814

Рис. 3.11. Дисперсійний аналіз завдання та алгоритми за рейтингом два - Природність опису

З рисунка 3.11 ми бачимо, що фактор алгоритму має значний вплив на рейтинг два незалежно, тоді як фактор алгоритму та фактор завдання в поєднанні також мають значний вплив на рейтинг два. Але фактор завдання сам по собі не показує значного впливу на рейтинги.

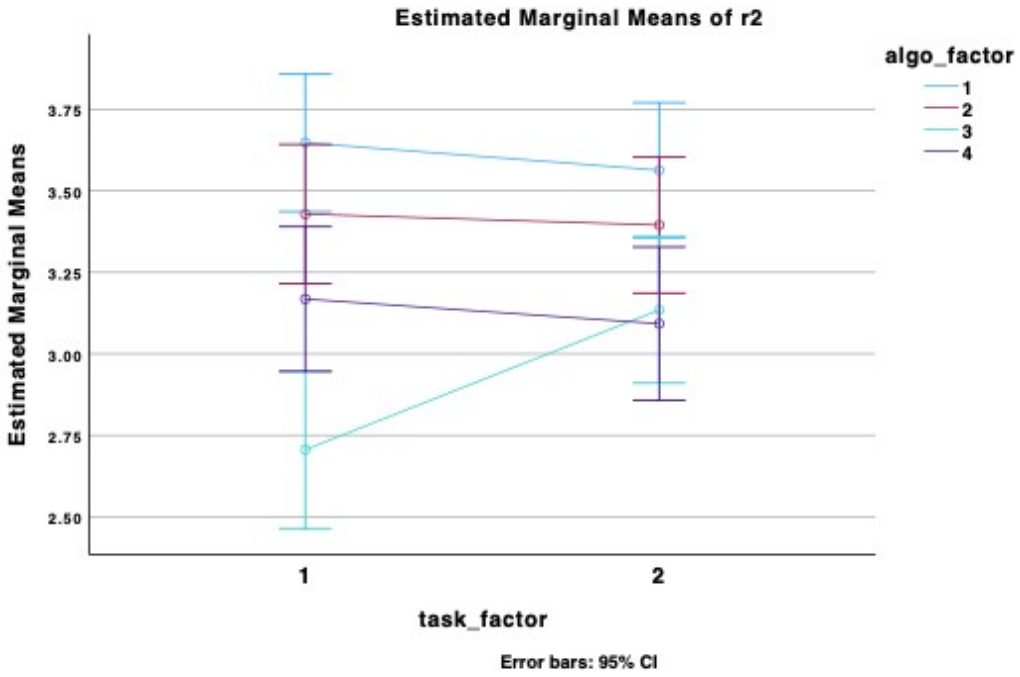

Рис. 3.12. Продуктивність завдання проти алгоритму для рейтингу природності

Графічне представлення продуктивності алгоритмів відносно типів завдань допомагає нам зробити висновок, що:

- Продуктивність алгоритму 1 знизилася з точки зору рейтингу для завдань узагальнення порівняно із завданнями пошуку.

- Продуктивність алгоритму 2 з точки зору рейтингів була високою для завдань пошуку, але незначно знизилася порівняно із завданнями узагальнення.

- Продуктивність алгоритму 3 з точки зору рейтингів демонструє різке зростання від завдань пошуку до завдань узагальнення.

- Алгоритм 4 показав кращі результати для типу завдання пошуку порівняно із завданнями узагальнення.

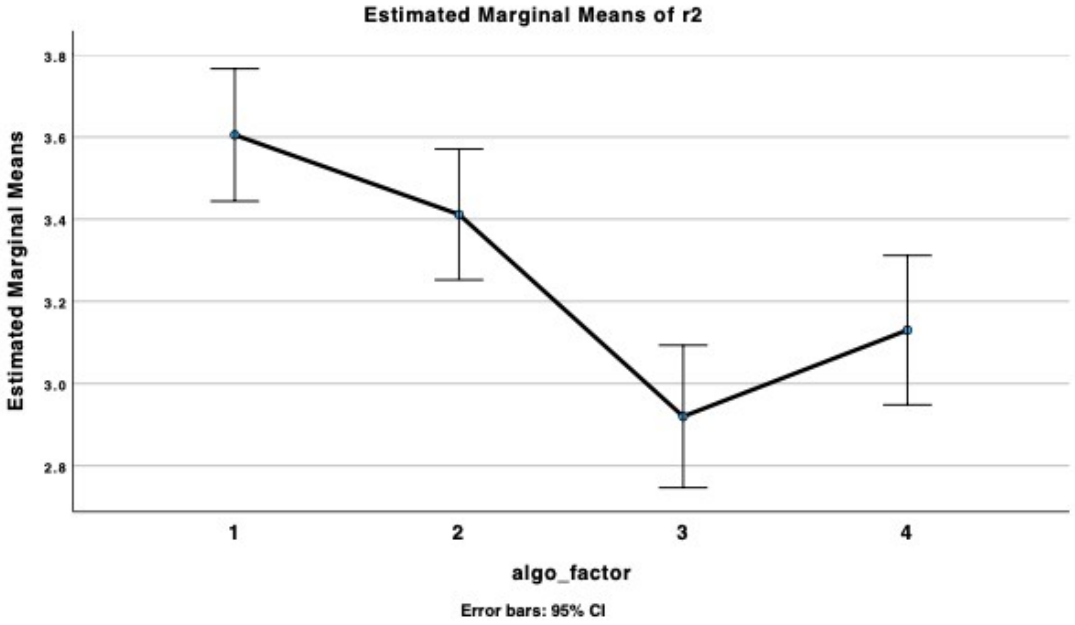

Рис. 3.13. Продуктивність алгоритму для рейтингу природності

Алгоритм 1 показав найкращі результати в цілому.

Multivariate Tests[a]

Effect		Sig.	Partial Eta Squared	Noncent. Parameter
algo_factor	Pillai's Trace	<.001	.174	24.468
	Wilks' Lambda	<.001	.174	24.468
	Hotelling's Trace	<.001	.174	24.468
	Roy's Largest Root	<.001	.174	24.468
task_factor	Pillai's Trace	.046	.033	4.083
	Wilks' Lambda	.046	.033	4.083
	Hotelling's Trace	.046	.033	4.083
	Roy's Largest Root	.046	.033	4.083
algo_factor * task_factor	Pillai's Trace	<.001	.187	26.617
	Wilks' Lambda	<.001	.187	26.617
	Hotelling's Trace	<.001	.187	26.617
	Roy's Largest Root	<.001	.187	26.617

Рис. 3.14. Дисперсійний аналіз завдання та алгоритми за рейтингом три - Рейтинг задоволення

З рисунка 3.14 ми бачимо, що фактор алгоритму має значний вплив на рейтинг три незалежно, тоді як фактор алгоритму та фактор завдання в поєднанні також мають значний вплив на рейтинг три. Однак цього разу фактор завдання також показує значний вплив на рейтинги.

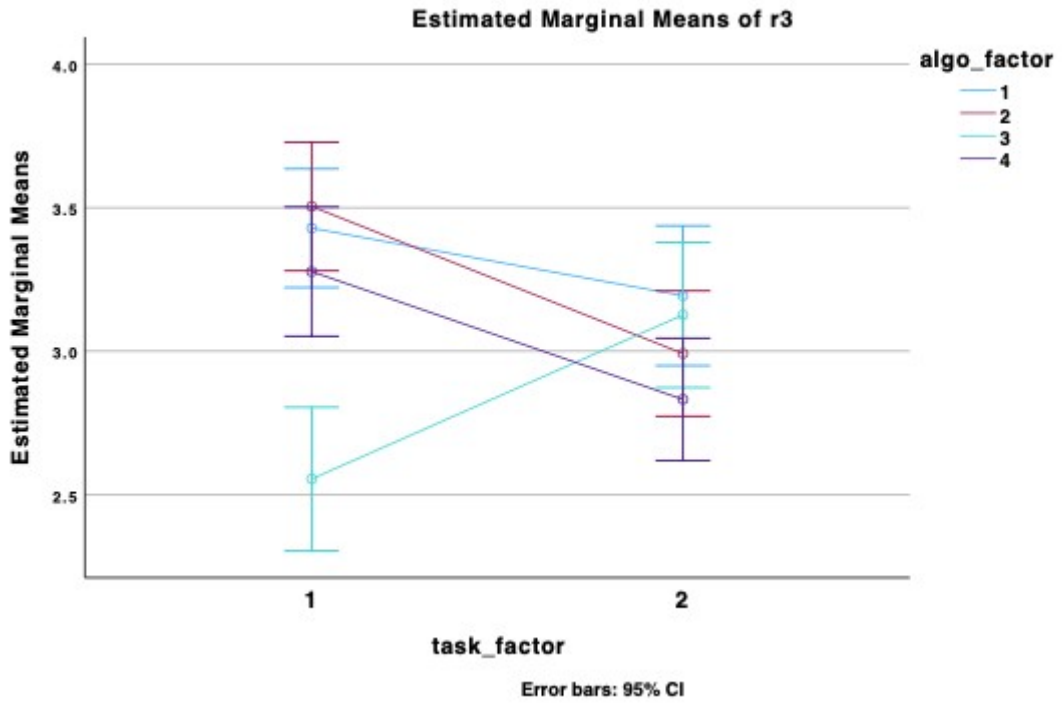

Рис. 3.15. Продуктивність алгоритму для рейтингу природності

Графічне представлення продуктивності алгоритмів відносно типів завдань допомагає нам зробити висновок, що:

- Продуктивність алгоритму 1 знизилася з точки зору рейтингу для завдань узагальнення порівняно із завданнями пошуку.

- Продуктивність алгоритму 2 з точки зору рейтингів була високою для завдань пошуку, але значно знизилася порівняно із завданнями узагальнення.

- Продуктивність алгоритму 3 з точки зору рейтингів демонструє різке зростання від завдань пошуку до завдань узагальнення.

- Алгоритм 4 показав кращі результати для типу завдання пошуку порівняно із завданнями узагальнення.

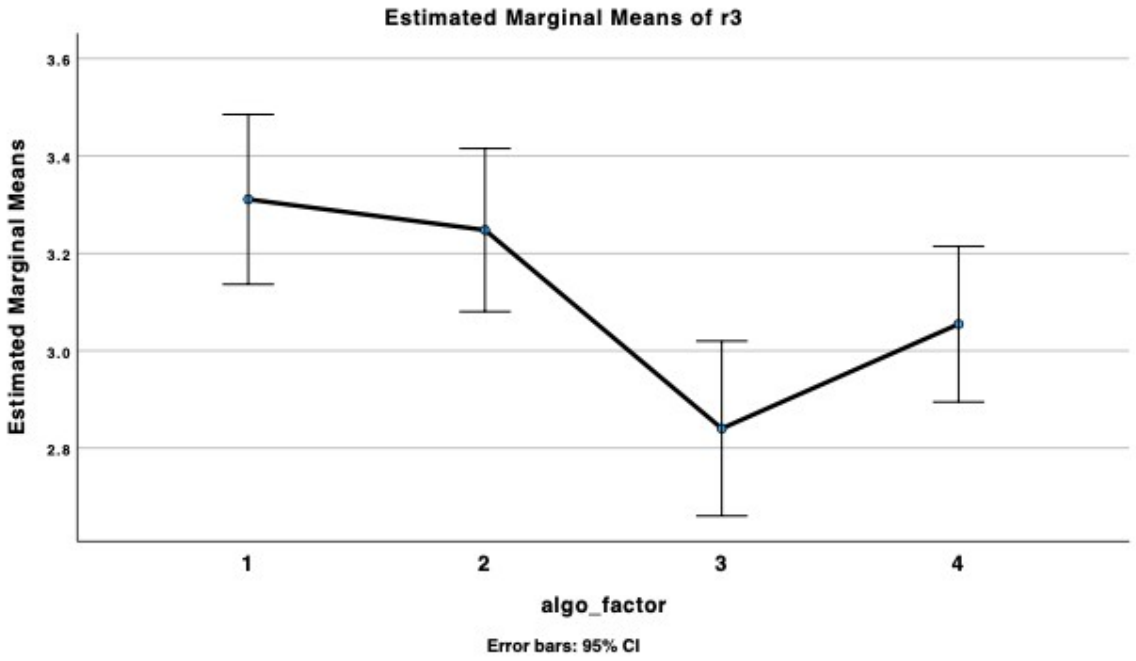

Рис. 3.16. Продуктивність алгоритму для природності опису

Діаграми взаємодії нижче допомагають нам зрозуміти взаємодію, яка здається досить значною. Цей тип діаграми також відомий як профільна діаграма, яка використовується для ефектів взаємодії.

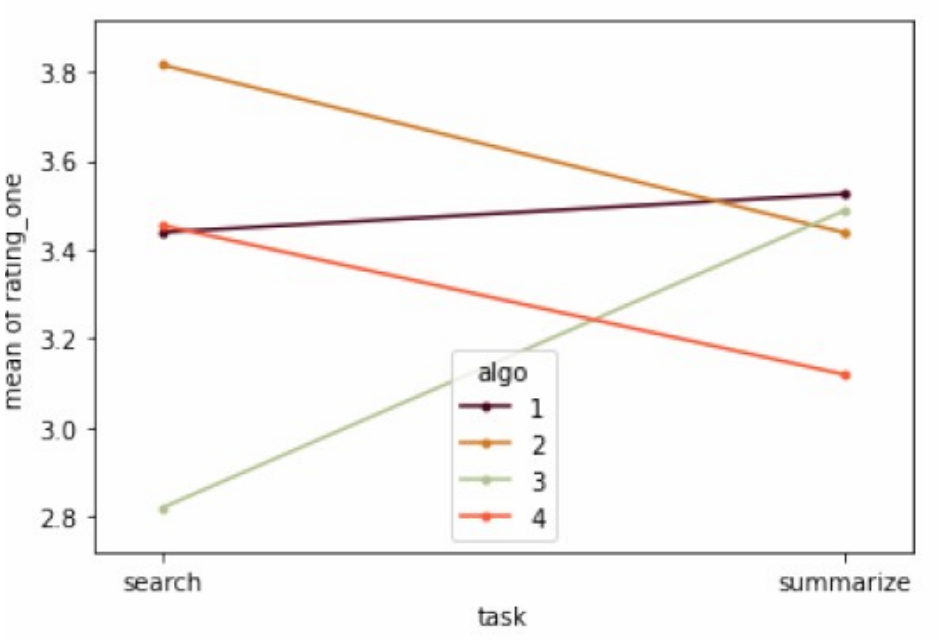

Рис. 3.17. Взаємодія алгоритмів завдань - рейтинг правильності

Графіки допомагають нам візуалізувати середні значення відповідей, отриманих від користувача. Діаграма базується на двох основних факторах - Завданні та Алгоритмі. Обидва вони відображені на одному графіку, щоб зрозуміти ефекти взаємодії.

Ми можемо зробити висновок з діаграми взаємодії, що ефект взаємодії між алгоритмами та завданнями є значним. У деяких випадках чотири побудовані лінії не паралельні, коли графіки приблизно паралельні, це вказує на відсутність значної взаємодії.

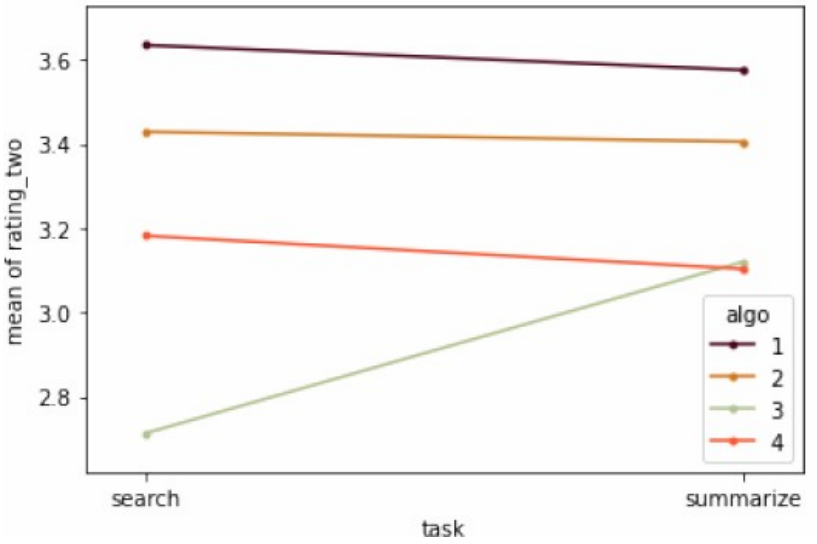

Рис. 3.18. Взаємодія алгоритмів завдань - рейтинг природності

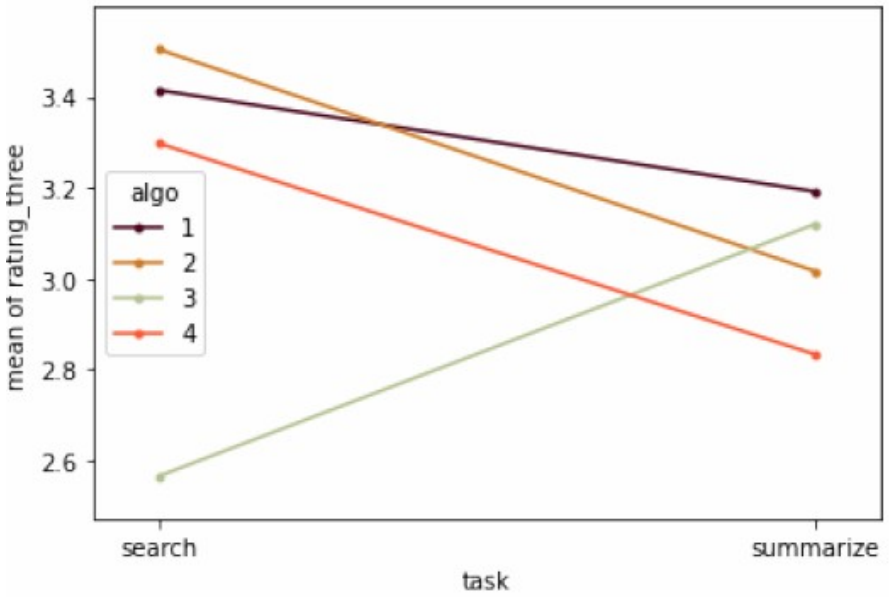

Рис. 3.19. Взаємодія алгоритмів завдань - рейтинг задоволення

3.2.2. Множинне попарне порівняння (пост-хок тест)

Множинне попарне порівняння (пост-хок тест) - це статистичний метод, який використовується для визначення того, які саме групи відрізняються одна від одної після того, як ANOVA показав значну різницю між середніми значеннями трьох або більше груп.

Після аналізу за допомогою дисперсійного аналізу (ANOVA) ми впевнені, що маємо інформацію про те, що різні алгоритми та рейтинги мають значний вплив на оцінки користувачів. Але ANOVA не говорить нам, яка саме категорія алгоритмів або категорія завдань суттєво відрізняється. Однак, щоб зрозуміти індивідуальний вплив групи, ми провели тест Тьюкі HSD та використали загальну лінійну модель, щоб визначити різноманітний набір пар, сформованих між завданням та алгоритмами, які мають значний вплив. Тут виконуються множинні попарні порівняння (пост-хок порівняння).

Як згадувалося раніше, ми об'єднали алгоритм 1 та алгоритм 2 (Кількість та Визначність) відповідно в одну категорію, названу Об'єктно-орієнтовані, та алгоритм 3 та алгоритм 4 (Зліва направо та Знизу вгору) відповідно як Середовищно-орієнтовані або Просторові алгоритми. Крім того, ми провели тестування, групуючи завдання пошуку з обома категоріями та узагальнюючи завдання з обома категоріями, щоб зробити висновок, яка категорія має значний вплив на рейтинги.

На графіках та в таблиці нижче в наступному розділі OC = Об'єктно-орієнтовані та EC = Середовищно-орієнтовані.

Для основного ефекту: Групи за рейтингом

Пост-хок тести проводяться, щоб з'ясувати, чи є певні відмінності між групами, коли ми виявили, що ANOVA є значущим. Пост-хок тест контролює рівень помилок, який розраховується між групами або для всієї сукупності. Пост-хок тест змінює p-значення (корекція Бонферроні) або критичні значення (Тьюкі HSD).

Multivariate Tests[a]

Effect		Sig.	Partial Eta Squared	Noncent. Parameter
task_factor	Pillai's Trace	<.001	.062	15.996
	Wilks' Lambda	<.001	.062	15.996
	Hotelling's Trace	<.001	.062	15.996
	Roy's Largest Root	<.001	.062	15.996
algo_category	Pillai's Trace	.923	.000	.009
	Wilks' Lambda	.923	.000	.009
	Hotelling's Trace	.923	.000	.009
	Roy's Largest Root	.923	.000	.009
task_factor * algo_category	Pillai's Trace	.151	.009	2.074
	Wilks' Lambda	.151	.009	2.074
	Hotelling's Trace	.151	.009	2.074
	Roy's Largest Root	.151	.009	2.074

Рис. 3.20. Вплив групи на рейтинг один - Правильність

Як показують результати, фактор завдання має значний вплив, а фактор завдання та категорії разом не мають значного впливу. Пізніше ми дізнаємося, яка категорія алгоритмів найкраще підходить для якого типу завдання.

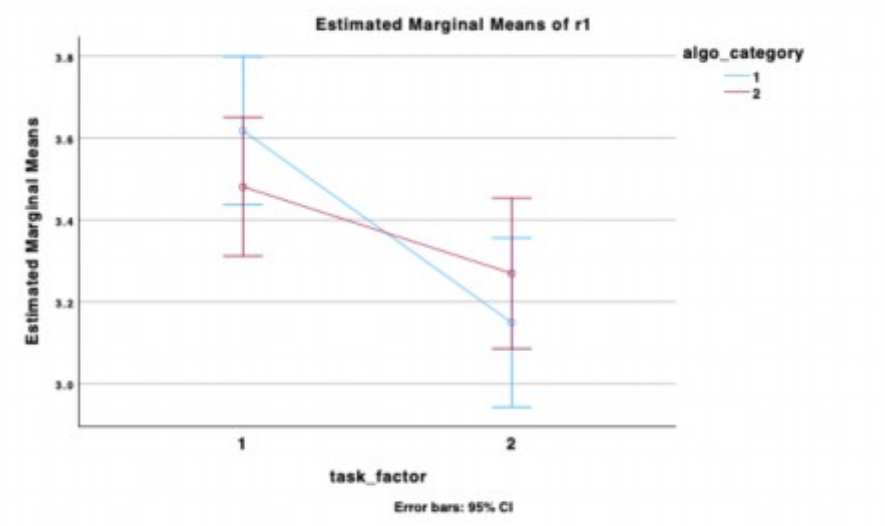

Рис. 3.21. Вплив групи на рейтинг один – Правильність

За допомогою графіків ми зрозуміли, що категорія алгоритмів 1, яка означає Об'єктно-орієнтований алгоритм, показала кращі результати для завдань пошуку, а категорія алгоритмів 2, Просторові або Середовищно-орієнтовані, досягла кращих рейтингів для завдань узагальнення, якщо

розглядати рейтинг правильності. У межах груп категорія алгоритмів 1 та категорія алгоритмів 2 показали кращі результати для завдання пошуку порівняно з узагальненням.

Multivariate Tests[a]

Effect		Sig.	Partial Eta Squared	Noncent. Parameter
task_factor	Pillai's Trace	<.001	.161	46.109
	Wilks' Lambda	<.001	.161	46.109
	Hotelling's Trace	<.001	.161	46.109
	Roy's Largest Root	<.001	.161	46.109
algo_category	Pillai's Trace	.475	.002	.512
	Wilks' Lambda	.475	.002	.512
	Hotelling's Trace	.475	.002	.512
	Roy's Largest Root	.475	.002	.512
task_factor * algo_category	Pillai's Trace	.125	.010	2.370
	Wilks' Lambda	.125	.010	2.370
	Hotelling's Trace	.125	.010	2.370
	Roy's Largest Root	.125	.010	2.370

Рис. 3.22. Вплив групи на Рейтинг Два - Природність

Як показують результати, фактор завдання має значний вплив, а фактор завдання та категорії алгоритмів разом не мають значного впливу. Пізніше ми дізнаємося, яка категорія алгоритмів найкраще підходить для якого типу завдання.

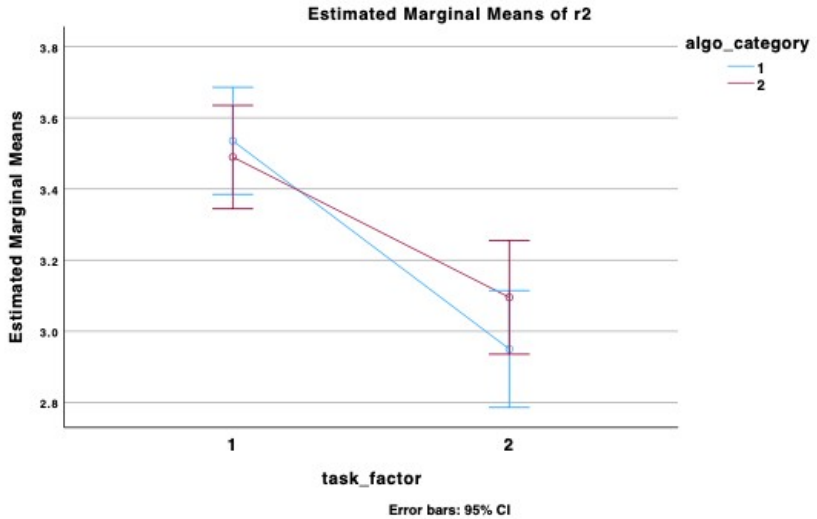

Рис. 3.23. Вплив групи на Рейтинг Два - Природність

За допомогою графіків ми зрозуміли, що категорія алгоритмів 1, яка означає Об'єктно-орієнтований алгоритм, показала кращі результати для завдань пошуку, а категорія алгоритмів 2, Просторові або Середовищно-орієнтовані, досягла кращих рейтингів для завдань узагальнення, якщо розглядати рейтинг природності. У межах груп категорія алгоритмів 1 та категорія алгоритмів 2 показали кращі результати для завдання пошуку порівняно з узагальненням.

Щоб зрозуміти загалом найкращу категорію, нам потрібно поглянути на графік нижче. Загалом, категорія алгоритмів 2 показала кращі результати для будь-якого типу завдання з точки зору рейтингів.

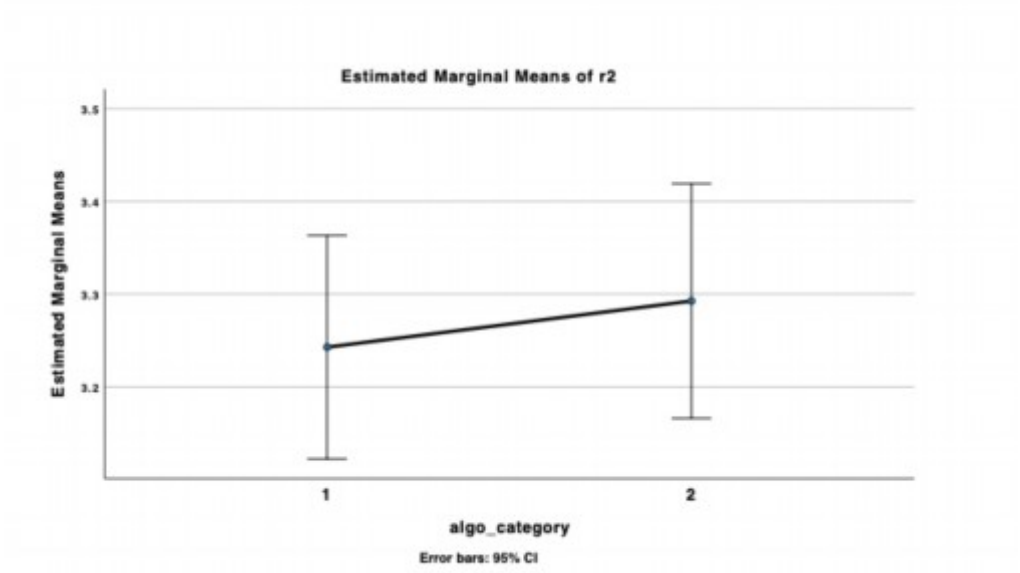

Рис. 3.24. Вплив групи на Рейтинг Два - Природність

Multivariate Tests[a]

Effect		Sig.	Partial Eta Squared	Noncent. Parameter
task_factor	Pillai's Trace	<.001	.075	19.533
	Wilks' Lambda	<.001	.075	19.533
	Hotelling's Trace	<.001	.075	19.533
	Roy's Largest Root	<.001	.075	19.533
algo_actegory	Pillai's Trace	.012	.026	6.445
	Wilks' Lambda	.012	.026	6.445
	Hotelling's Trace	.012	.026	6.445
	Roy's Largest Root	.012	.026	6.445
task_factor * algo_actegory	Pillai's Trace	.014	.025	6.091
	Wilks' Lambda	.014	.025	6.091
	Hotelling's Trace	.014	.025	6.091
	Roy's Largest Root	.014	.025	6.091

Рис. 3.25. Вплив групи на Рейтинг Три - Задоволення

Як показують результати, фактор завдання має значний вплив, а фактор завдання та категорії алгоритмів разом також мають значний вплив, тоді як категорії алгоритмів також мають значний вплив. Пізніше ми дізнаємося, яка категорія алгоритмів найкраще підходить для якого типу завдання.

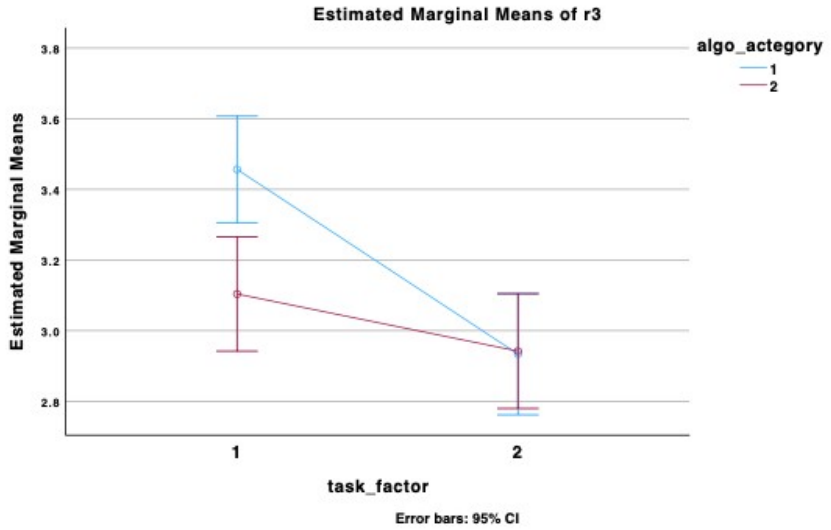

Рис. 3.26. Вплив групи на Рейтинг Три - Задоволення

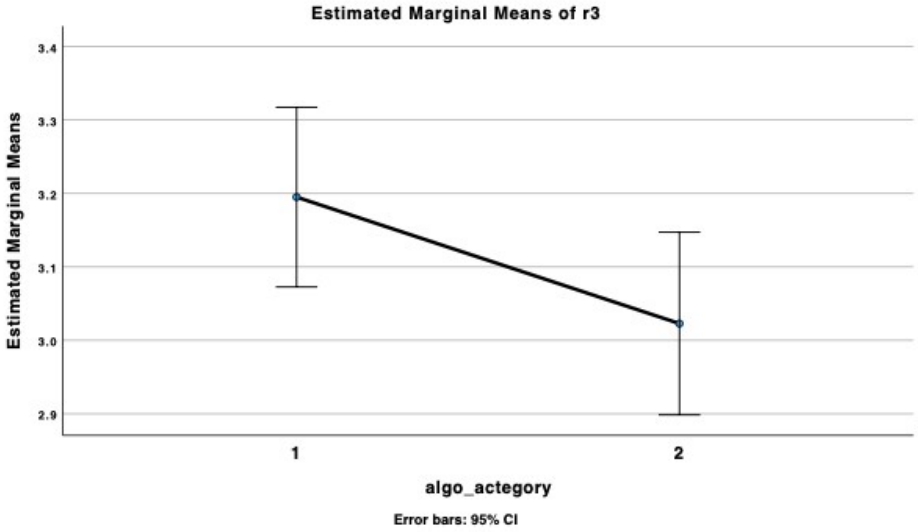

Рис. 3.27. Вплив групи на Рейтинг Три - Задоволення

За допомогою графіків ми зрозуміли, що категорія алгоритмів 1, яка означає Об'єктно-орієнтований алгоритм, показала кращі результати для завдань пошуку. Обидві категорії алгоритмів були подібними з точки зору

продуктивності, коли йдеться про задоволення в Просторовому або Середовищно-орієнтованому. У межах груп категорія алгоритмів 1 та категорія алгоритмів 2 показали кращі результати для завдання пошуку порівняно з узагальненням.

Щоб зрозуміти загалом найкращу категорію, нам потрібно поглянути на графік вище. Загалом, категорія алгоритмів 1 показала кращі результати для будь-якого типу завдання з точки зору рейтингів задоволення.

3.3. Проведення аналізу результатів за допомогою підходу непараметричної кореляції

Щоб глибше зануритися в наш аналіз та отримати конкретні відповіді щодо кореляції між зором та рейтингами, ми вирішили використати методи Спірмена та Кендалла для знаходження кореляції між групами стану зору та рейтингами, які вони дали за правильність опису, природність розповіді та задоволення від розповіді.

Опис термінів, що використовуються в таблиці:

- algo: група всіх чотирьох алгоритмів (кількість, визначність, зліва направо, знизу вгору)

- rating_one: Наскільки правильним був опис сцени?

- rating_two: Наскільки природною була розповідь?

- rating_three: Наскільки задовільною була розповідь?

Перш ніж рухатися далі та розуміти значущість груп зору на рейтинги, нам потрібно зрозуміти різні типи груп.

Опис стану зору:

Група 1: Гіперметропія (ви можете чітко бачити віддалені об'єкти, але об'єкти поблизу можуть бути розмитими) - 2,17%

Група 2: Нормальний зір - 43,48%

Група 3: Дальтонізм - 0%

Група 4: Міопія (стан, при якому близькі об'єкти видно чітко, а далекі - ні) - 54,35%

Correlations

			vision_status	algo	rating_one
Kendall's tau_b	vision_status	Correlation Coefficient	1.000	.003	-.016
		Sig. (2-tailed)	.	.904	.577
		N	977	977	977
	algo	Correlation Coefficient	.003	1.000	-.061*
		Sig. (2-tailed)	.904	.	.020
		N	977	977	977
	rating_one	Correlation Coefficient	-.016	-.061*	1.000
		Sig. (2-tailed)	.577	.020	.
		N	977	977	977
	rating_two	Correlation Coefficient	.093**	-.136**	.400**
		Sig. (2-tailed)	.001	<.001	<.001
		N	977	977	977
	rating_three	Correlation Coefficient	.090**	-.079**	.502**
		Sig. (2-tailed)	.001	.003	<.001
		N	977	977	977
	task_auto	Correlation Coefficient	.006	.010	-.010
		Sig. (2-tailed)	.848	.723	.732
		N	977	977	977
Spearman's rho	vision_status	Correlation Coefficient	1.000	.004	-.018
		Sig. (2-tailed)	.	.904	.573
		N	977	977	977
	algo	Correlation Coefficient	.004	1.000	-.077*
		Sig. (2-tailed)	.904	.	.017
		N	977	977	977
	rating_one	Correlation Coefficient	-.018	-.077*	1.000
		Sig. (2-tailed)	.573	.017	.
		N	977	977	977
	rating_two	Correlation Coefficient	.104**	-.165**	.478**
		Sig. (2-tailed)	.001	<.001	<.001
		N	977	977	977
	rating_three	Correlation Coefficient	.101**	-.097**	.580**
		Sig. (2-tailed)	.002	.002	<.001
		N	977	977	977
	task_auto	Correlation Coefficient	.006	.011	-.011
		Sig. (2-tailed)	.848	.723	.732
		N	977	977	977

Рис. 3.28. Таблиця непараметричної кореляції

Згідно з коефіцієнтами кореляції та значущими значеннями, ми дійшли висновку, що стан зору та рейтинг два, а також стан зору та рейтинг три тісно корелюють, і отримані рейтинги сильно залежать від стану зору, або ми можемо сказати, що стан зору тут досить значущий.

Стан зору та рейтинг два (природність)

Група стану зору 4 дала найвищі оцінки за природність розповіді, більше 50% групи оцінили від 3 до 5. Найвищий рейтинг 3 отримала більшість користувачів з групи.

Ми можемо побачити це на графіку нижче для візуального розуміння.

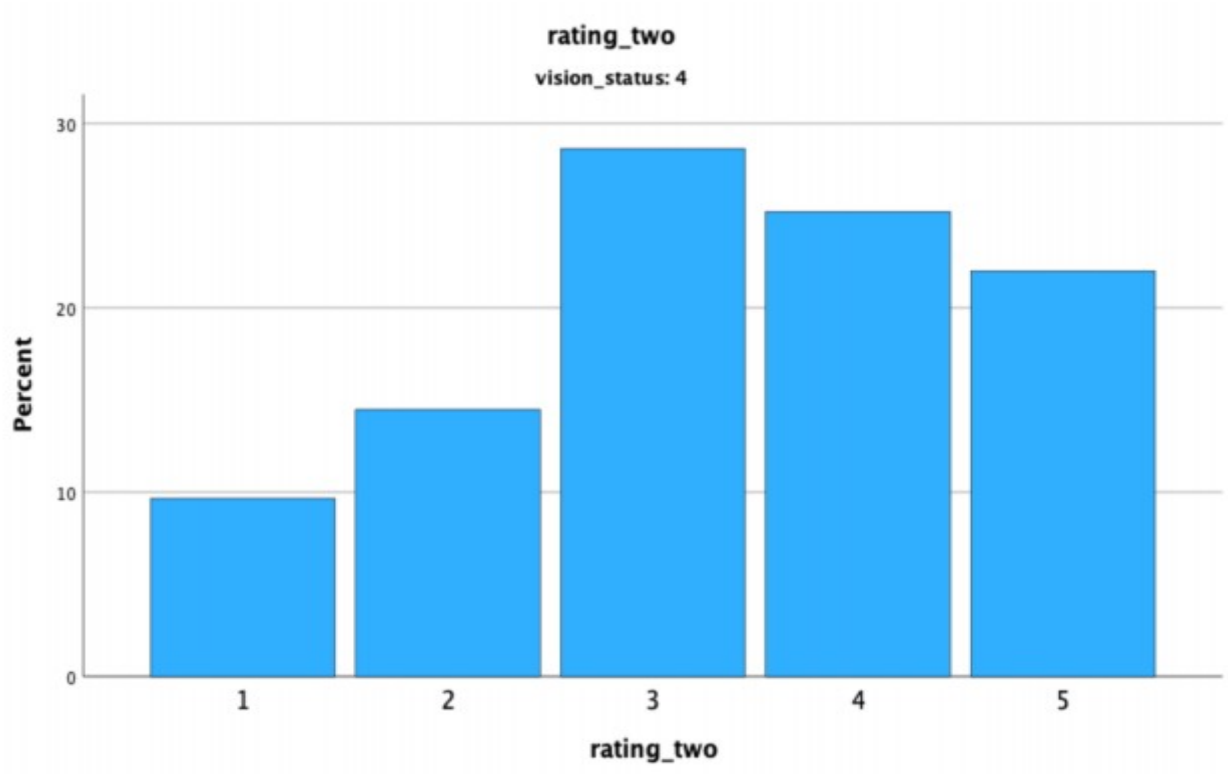

Рис. 3.30. Група стану зору 4 (міопія) та рейтинг природності

Група стану зору 2, яка є категорією нормального зору, оцінила 3 приблизно 30% користувачів у групі, але більше 50% оцінили 3 і вище, як ми можемо бачити на рисунку 3.31. Це означає, що люди з нормальним зором вважали природність розповіді хорошою, як і користувачі з міопією.

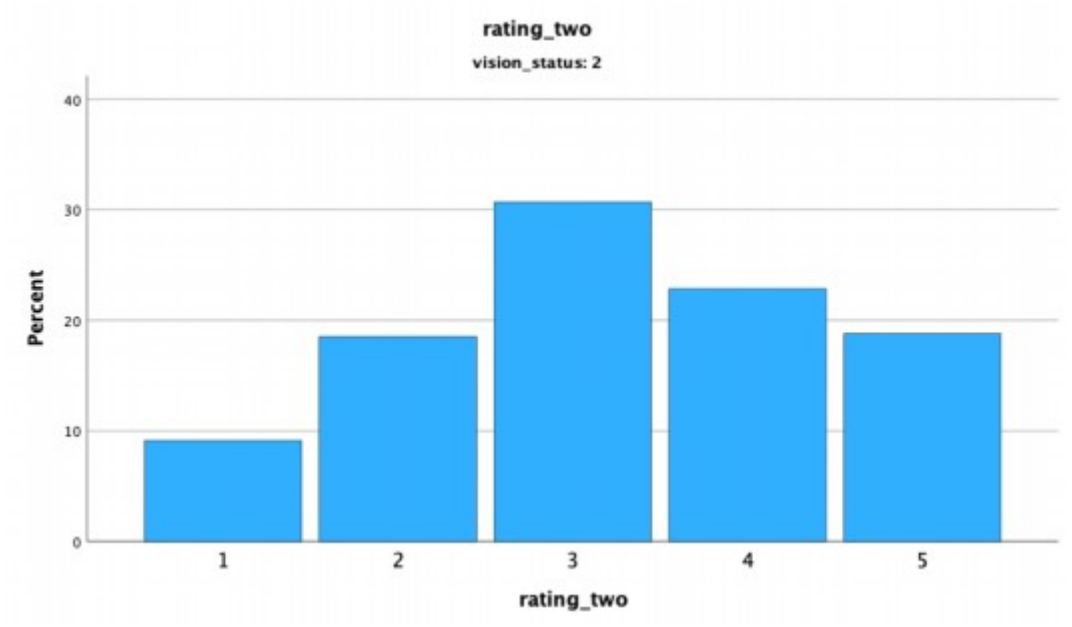

Рис. 3.31. Група стану зору 2 (нормальний) та рейтинг природності

Стан зору та рейтинг три (задоволення)

У цій категорії ми обговорили різні категорії зору та те, як вони оцінювали задоволення від того, наскільки вони були задоволені після прослуховування розповіді від системи.

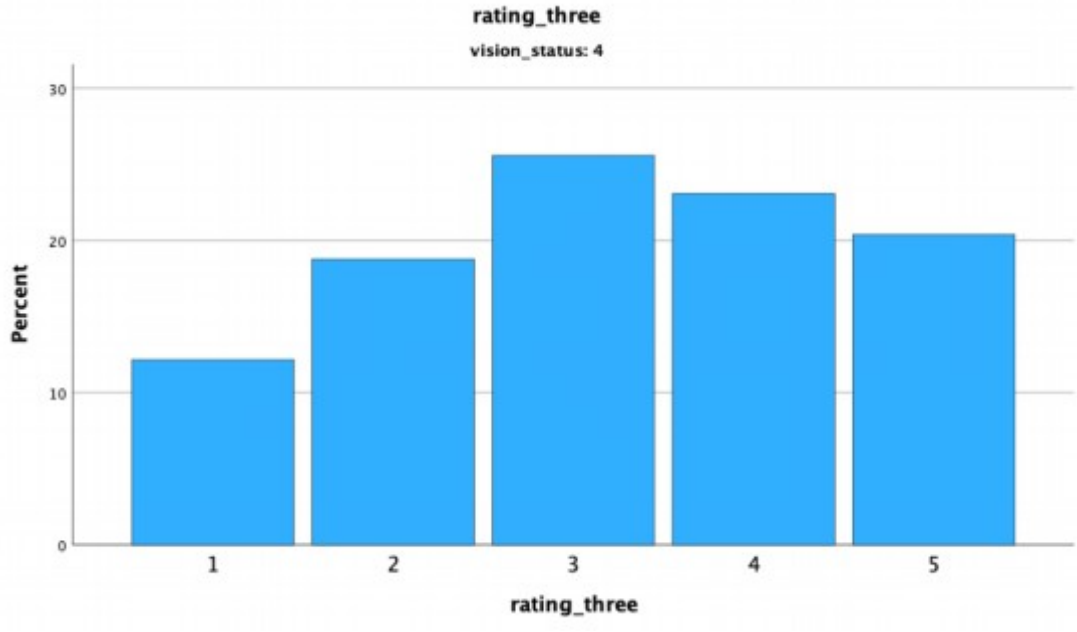

Рис. 3.32. Група стану зору 4 (міопія) та рейтинг задоволення

Найбільш задоволеними користувачами, чия група оцінила задоволення від розповіді найвище, була група зі станом зору 4 (міопія).

Більше 60% групи дали оцінку 3 і вище. Як ми можемо бачити на рисунку 3.32, найвища оцінка користувачів була три, але значна кількість користувачів з групи 4 була задоволена розповіддю, яку вони почули.

Враховуючи результат, що група користувачів з міопією оцінила вище за природність та задоволення, ми глибше занурилися у розуміння того, які алгоритми були найкращими для групи користувачів з міопією, які вони оцінили найвище за різними питаннями рейтингу.

Pairwise Comparisons

Measure	(I) algo	(J) algo	Mean Difference (I-J)	Std. Error	Sig.[b]	95% Confidence Interval for Difference[b] Lower Bound	Upper Bound
Correctness	1	2	-.058	.147	1.000	-.452	.336
		3	.210	.169	1.000	-.243	.663
		4	.101	.161	1.000	-.329	.532
	2	1	.058	.147	1.000	-.336	.452
		3	.268	.176	.779	-.203	.739
		4	.159	.153	1.000	-.251	.570
	3	1	-.210	.169	1.000	-.663	.243
		2	-.268	.176	.779	-.739	.203
		4	-.109	.160	1.000	-.538	.320
	4	1	-.101	.161	1.000	-.532	.329
		2	-.159	.153	1.000	-.570	.251
		3	.109	.160	1.000	-.320	.538
Naturalness	1	2	.181	.113	.672	-.122	.484
		3	.674*	.123	<.001	.344	1.003
		4	.377*	.124	.018	.044	.710
	2	1	-.181	.113	.672	-.484	.122
		3	.493*	.130	.001	.144	.841
		4	.196	.122	.668	-.131	.522
	3	1	-.674*	.123	<.001	-1.003	-.344
		2	-.493*	.130	.001	-.841	-.144
		4	-.297	.131	.147	-.647	.053
	4	1	-.377*	.124	.018	-.710	-.044
		2	-.196	.122	.668	-.522	.131
		3	.297	.131	.147	-.053	.647
Satisfaction	1	2	.159	.123	1.000	-.169	.488
		3	.580*	.144	<.001	.194	.965
		4	.341	.134	.071	-.017	.698
	2	1	-.159	.123	1.000	-.488	.169
		3	.420*	.154	.043	.008	.832
		4	.181	.137	1.000	-.184	.547
	3	1	-.580*	.144	<.001	-.965	-.194
		2	-.420*	.154	.043	-.832	-.008
		4	-.239	.138	.510	-.608	.130
	4	1	-.341	.134	.071	-.698	.017
		2	-.181	.137	1.000	-.547	.184
		3	.239	.138	.510	-.130	.608

Based on estimated marginal means

*. The mean difference is significant at the .05 level.

b. Adjustment for multiple comparisons: Bonferroni.

Рис. 3.33. Таблиця значущості стану зору (міопія) та алгоритму

Таблиця значущості вказує на зв'язок між алгоритмом та рейтингом. Ми побудували оцінені граничні середні значення рейтингу правильності, природності та задоволення.

Рис. 3.34 показує, що алгоритм 2 (Визначність) був найкращим за рейтингом правильності, як оцінили учасники групи з міопією (зір 4).

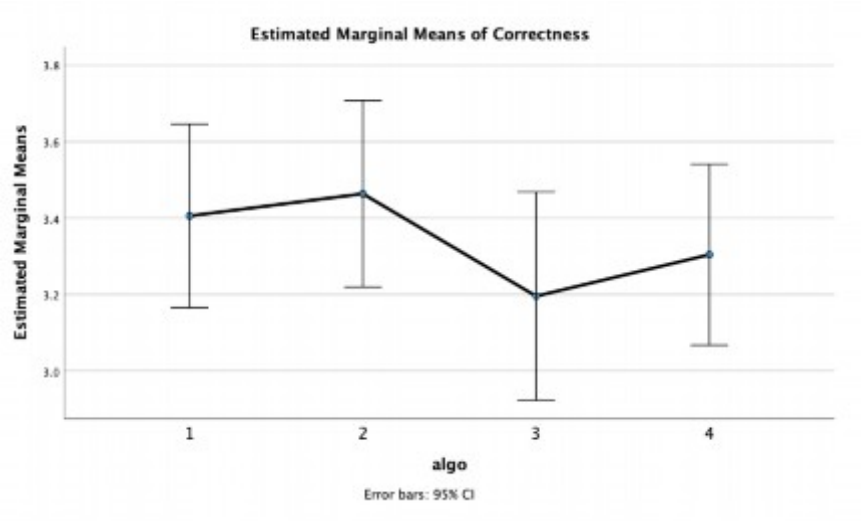

Рис. 3.34. Група користувачів з міопією та продуктивність алгоритму за правильністю

Рис. 3.35 показує, що алгоритм 1 (Кількість) був найкращим за рейтингом природності, як оцінили учасники групи з міопією (зір 4).

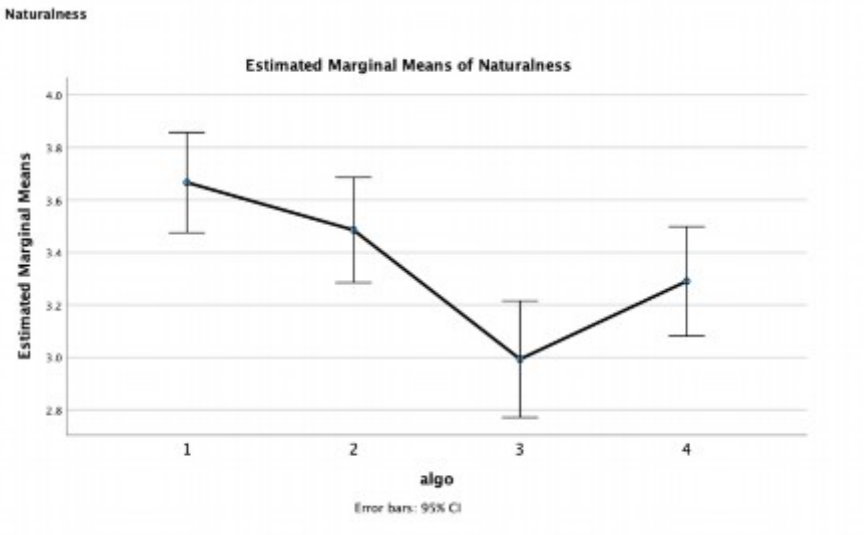

Рис. 3.35. Група користувачів з міопією та продуктивність алгоритму за природністю

Рис. 3.36 показує, що алгоритм 1 (Кількість) був найкращим за рейтингом задоволення, як оцінили учасники групи з міопією (зір 4).

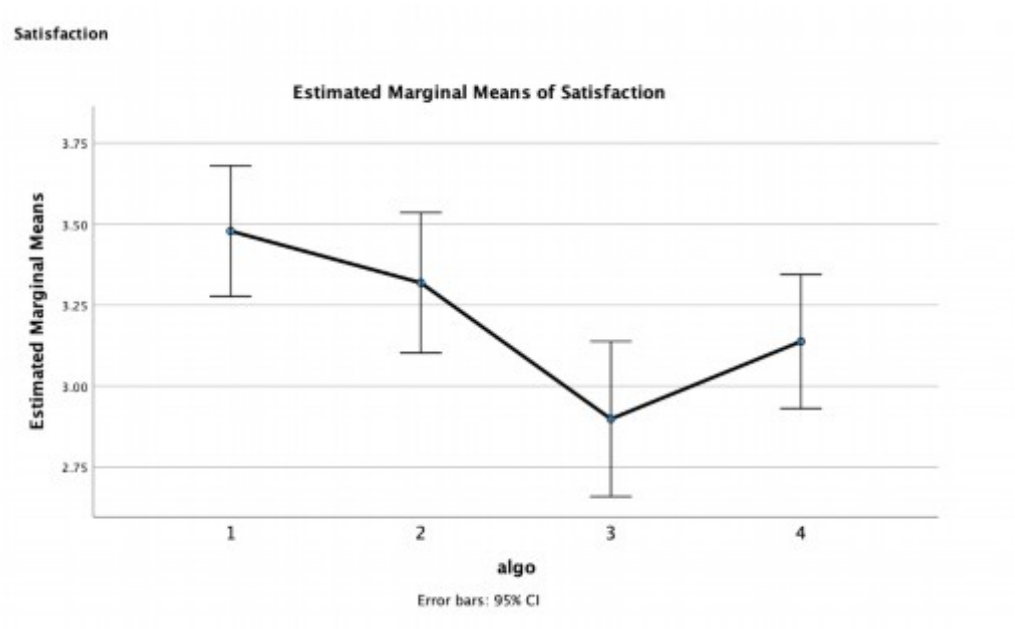

Рис. 3.36. Група користувачів з міопією та продуктивність алгоритму за задоволенням

Висновки до розділу

Отже, в цьому розділі проведено імплементацію моделей і алгоритмів, що використовують нейронні мережі та методи обробки природної мови для аналізу зображень об’єктів. Розроблено та представлено дизайн інтерфейсу користувача для системи виявлення об'єктів. Інтерфейс створено з акцентом на інтуїтивність, зручність та інтерактивність, що забезпечує ефективну взаємодію користувача із системою. Реалізовано алгоритми обробки зображень із використанням нейронних мереж, а також оцінено їхню ефективність на практичних задачах.

Проведено кількісний аналіз залежностей між факторами, які впливають на результати роботи моделей, із використанням повторних вимірювань. Отримані дані свідчать про значущість вибраних параметрів.

Виконано детальне порівняння різних конфігурацій моделей і параметрів. Це дозволило визначити найефективніші підходи для конкретних задач. Проведено оцінку результатів роботи моделей за допомогою непараметричних методів кореляції. Цей підхід дозволив визначити зв'язки між вхідними даними та продуктивністю алгоритмів, а також оцінити їхню надійність і узгодженість.

Таким чином, у цьому розділі розроблено та впроваджено ключові елементи системи виявлення об’єктів, включаючи зручний інтерфейс користувача, алгоритмічну базу та методи аналізу отриманих результатів. Результати імплементації підтвердили ефективність використаних підходів для задач розпізнавання об'єктів і обробки зображень.

ВИСНОВКИ

У магістерській роботі досліджено методи та моделі, що базуються на застосуванні природної мови та нейронних мереж для обробки зображень і розпізнавання об’єктів. У результаті дослідження розроблено комплексне рішення для автоматизованого аналізу та розуміння сцени у віртуальному середовищі.

Запропоноване рішення інтегрує кілька компонентів, зокрема алгоритми машинного навчання для виявлення об'єктів та X3DOM для візуалізації 3D-сцен у браузері. Користувачам надано можливість взаємодіяти зі сценою через навігацію, створення знімків екрана та отримання описів сцени. Опис сцени формується на основі чотирьох алгоритмів: за кількістю об'єктів, визначністю, послідовністю зліва направо та знизу вгору.

Для оцінки ефективності цих алгоритмів було проведено імітаційне дослідження користувачів. Головною гіпотезою дослідження було визначення, який із запропонованих алгоритмів найбільш придатний для завдань узагальнення та завдань пошуку. Аналіз результатів показав:

- Просторові алгоритми (зліва направо, знизу вгору) продемонстрували кращу ефективність у завданнях узагальнення, ніж у завданнях пошуку.

- Алгоритми, орієнтовані на об'єкти (за кількістю, визначністю), показали вищу ефективність у завданнях пошуку.

- Загалом, алгоритм "зліва направо" отримав найнижчі оцінки, тоді як об'єктно-орієнтовані алгоритми були найбільш корисними для завдань пошуку, а середовищно-орієнтовані — для завдань узагальнення.

Дослідження також виявило обмеження системи. Зокрема:

- Оптимізація візуалізації (зменшення якості текстур та оптимізація геометрії) призвела до зниження точності алгоритмів в окремих випадках.

- Навігація сценою в браузері виявилася складною для деяких користувачів через апаратні обмеження та особливості використання трекпада або миші.

- Опис сцени іноді перешкоджав природним звукам середовища, що могло впливати на користувацький досвід.

Для подолання цих обмежень запропоновано подальше вдосконалення системи:

- Інтеграція більш складних моделей обробки природної мови, зокрема алгоритмів глибокого навчання, для створення гнучкіших та детальніших описів.

- Розробка персоналізованих віртуальних середовищ, які враховуватимуть індивідуальні потреби користувачів із різними типами вад зору.

- Оптимізація продуктивності системи для покращення якості взаємодії та точності розпізнавання об’єктів.

Таким чином, отримані результати підтверджують ефективність використання комбінованих алгоритмів для вирішення завдань розпізнавання та опису сцен, що створює основу для подальших досліджень у напрямку адаптивних віртуальних середовищ

ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. J. Redmon and A. Farhadi, “Yolo9000: Better, faster, stronger,” in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6517–6525, 2017.
2. J. Behr, P. Eschler, Y. Jung, and M. Zöllner, “X3dom: A dom-based html5/x3d integration model,” in Proceedings of the 14th International Conference on 3D Web Technology, Web3D ’09, (New York, NY, USA), p. 127–135, Association for Computing Machinery, 2009.
3. Vaswani, A., et al. "Attention is All You Need." NeurIPS (2017).
4. Xie, N., et al. "Unsupervised Data Augmentation for Consistency Training." NeurIPS (2020).
5. Dosovitskiy, A., et al. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale." ICLR (2021).
6. J. Behr, Y. Jung, T. Drevensek, and A. Aderhold, “Dynamic and interactive aspects of x3dom,” in Proceedings of the 16th International Conference on 3D Web Technology, Web3D ’11, (New York, NY, USA), p. 81–87, Association for Computing Machinery, 2011.
7. J. Behr, Y. Jung, J. Keil, T. Drevensek, M. Zoellner, P. Eschler, and D. Fellner, “A scalable architecture for the html5/x3d integration model x3dom,” in Proceedings of the 15th International Conference on Web 3D Technology, Web3D ’10, (New York, NY, USA), p. 185–194, Association for Computing Machinery, 2010.
8. Ramesh, A., et al. "Hierarchical Text-to-Image Synthesis Using CLIP." Proceedings of NeurIPS (2022).
9. K. Vines, C. Hughes, L. Alexander, C. Calvert, C. Colwell, H. Holmes, C. Kotecki, K. Parks, and V. Pearson, “Sonification of numerical data for education,” Open Learning: The Journal of Open, Distance and e-Learning, vol. 34, pp. 19–39, 01 2019.

10. A. Constantinescu, K. Müller, M. Haurilet, V. Petrausch, and R. Stiefelhagen, “Bring the environment to life: A sonification module for people with visual impairments to improve situation awareness,” in Proceedings of the 2020 International Conference on Multimodal Interaction, ICMI ’20, (New York, NY, USA), p. 50–59, Association for Computing Machinery, 2020.
11. M. Geronazzo, A. Bedin, L. G. Brayda, C. Campus, and F. Avanzini, “Interactive spatial sonification for non-visual exploration of virtual maps,” Int. J. Hum. Comput. Stud., vol. 85, pp. 4–15, 2016.
12. Anderson, P., et al. "Bottom-up and Top-down Attention for Image Captioning and Visual Question Answering." IEEE Transactions on Pattern Analysis and Machine Intelligence (2018).
13. Radford, A., et al. "Learning Transferable Visual Models from Natural Language Supervision." Proceedings of the International Conference on Machine Learning (2021).
14. Li, J., et al. "Oscar: Object-semantics Aligned Pre-training for Vision-Language Tasks." ECCV (2020).
15. Chen, Y.-C., et al. "UNITER: UNiversal Image-TExt Representation Learning." EMNLP (2019).
16. Tan, H., Bansal, M. "LXMERT: Learning Cross-Modality Encoder Representations from Transformers." EMNLP (2019).
17. Su, W., et al. "VL-BERT: Pre-training of Generic Visual-Linguistic Representations." ICLR (2020).
18. Kim, D., et al. "ViLT: Vision-and-Language Transformer Without Convolution or Region Supervision." Proceedings of ICML (2021).
19. Ramesh, A., et al. "Zero-Shot Text-to-Image Generation." Proceedings of ICML (2021).
20. Xu, K., et al. "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention." ICML (2015).

21. Huang, L., et al. "Attention on Attention for Image Captioning." ICCV (2019).
22. Li, L., et al. "Blip: Bootstrapped Language-Image Pre-training for Unified Vision-Language Understanding and Generation." NeurIPS (2022).
23. Cho, J., et al. "Unifying Vision-and-Language Tasks via Text Generation." EMNLP (2021).
24. Dong, L., et al. "Unified Language Model Pre-training for Natural Language Understanding and Generation." NeurIPS (2019).
25. Zhang, X., et al. "Enhancing Vision-and-Language Pre-training with Label Supervision." ICLR (2022).
26. Yang, S., et al. "Visual Dialog." Proceedings of CVPR (2017).
27. Gao, P., et al. "Transformer for Image Captioning." Proceedings of CVPR (2021).
28. Lin, T.-Y., et al. "Microsoft COCO: Common Objects in Context." ECCV (2014).
29. Devlin, J., et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." NAACL-HLT (2019).
30. Brown, T., et al. "Language Models Are Few-Shot Learners." NeurIPS (2020).
31. Wang, L., et al. "ImageBERT: Cross-modal Pre-training with Large-scale Image-Text Data." ECCV (2020).
32. Carion, N., et al. "End-to-End Object Detection with Transformers." ECCV (2020).
33. Raffel, C., et al. "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer." JMLR (2020).
34. Hao, W., et al. "Gibbs Sampling with People." ICML (2022).
35. Zhang, H., et al. "StackGAN: Text to Photo-Realistic Image Synthesis with Stacked Generative Adversarial Networks." ICCV (2017).
36. Johnson, J., et al. "Image Generation from Scene Graphs." CVPR (2018).

37. Heusel, M., et al. "GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium." NeurIPS (2017).
38. Chen, M., et al. "Generative Pretrained Transformer with Multiple Modalities." NeurIPS (2021).
39. Hu, R., et al. "Exploiting Semantic Context for Vision-and-Language Reasoning." CVPR (2020).
40. Shen, T., et al. "How to Integrate Language and Vision: Better Models and Stronger Benchmarks." ICLR (2021).
41. Liu, W., et al. "Exploring Context in Vision-Language Embeddings for Image Captioning." CVPR (2022).
42. Ji, Z., et al. "Text-to-Image Diffusion Models: A Comprehensive Review." arXiv (2023).
43. Wu, J., et al. "NLP-Driven Image Editing with Diffusion Models." IEEE Transactions (2023).
44. Gao, Y., et al. "Advances in Vision and Language Tasks." Springer (2022).
45. Shen, Z., et al. "Understanding Vision-Language Pre-training with Attention Mechanisms." Springer (2022).
46. Wei, X., et al. "Multi-modal Pre-training for Generative AI." IEEE Journals (2023).
47. Zhang, L., et al. "Self-supervised Learning for Multimodal Systems." MDPI Applied Sciences (2023).
48. D. Ahmetovic, F. Avanzini, A. Baratè, C. Bernareggi, G. Galimberti, L. A. Ludovico, S. Mascetti, and G. Presti, “Sonification of rotation instructions to support navigation of people with visual impairment,” in 2019 IEEE International Conference on Pervasive Computing and Communications (PerCom, pp. 1–10, 2019.
49. N. Aziz, T. Stockman, and R. Stewart, “An investigation into customisable automatically generated auditory route overviews for pre-navigation,” 06 2019.

50. M. Ferati, S. Mannheimer, and D. Bolchini, “Usability evaluation of acoustic interfaces for the blind,” SIGDOC’11 - Proceedings of the 29th ACM International Conference on Design of Communication, 10 2011.
51. M. Everingham, S. M. Eslami, L. Gool, C. K. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes challenge: A retrospective,” Int. J. Comput. Vision, vol. 111, p. 98–136, jan 2015.