

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Харківський національний університет радіоелектроніки
Факультет Комп'ютерних наук
Кафедра Програмної інженерії

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

другий (магістерський)
(рівень вищої освіти)

Дослідження підходів до розпізнавання та анотації об'єктів в доповненій
реальності для путівників.

Виконав:
студент 2 курсу групи ІПЗм-20-2
Черненко Є. А.

(прізвище, ініціали)

Спеціальніс 121 – Інженерія програмного
ть забезпечення

Тип програми Освітньо-наукова

Керівник проф. Білоус Н.В.

(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____

З.В. Дудар

2022 р.

Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук

Кафедра _____ Програмної інженерії

Рівень вищої освіти _____ другий (магістерський)

Спеціальність _____ 121– Інженерія програмного забезпечення
(код і повна назва)

Тип програми _____ освітньо-наукова програма

Освітня програма _____ Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

«__» _____ 202_ р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студента _____ Черненка Єгора Анатолійовича
(прізвище, ім'я, по батькові)

1. Тема роботи «Дослідження підходів до розпізнавання та анотації об'єктів в доповненій реальності для путівників.»

затверджена наказом університету від «__» _____ 202__ р. №__

2. Термін подання студентом роботи до екзаменаційної комісії «__» _____ 202__ р.

3. Вихідні дані до роботи Технічне завдання, календарний план, методичні вказівки.

4. Перелік питань, що потрібно опрацювати в роботі мета роботи, аналіз предметної галузі і постановка задачі, дослідження технологій розпізнавання, локалізації та анотації об'єктів та інтеграції доповненої реальності.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1.	Аналіз предметної галузі	31 березня 2022	<i>виконано</i>
2.	Огляд існуючих методів	10 квітня 2022.	<i>виконано</i>
3.	Розробка алгоритмів, проектування та розробка ПЗ	15 квітня 2022	<i>виконано</i>
4.	Підготовка пояснювальної записки	20 квітня 2022	<i>виконано</i>
5.	Спецчастина	28 квітня 2022	<i>виконано</i>
6.	Підготовка презентації та доповіді	03 травня 2022	<i>виконано</i>
7.	Попередній захист	16 травня 2022	<i>виконано</i>
8.	Нормоконтроль, рецензування	17 травня 2022.	<i>виконано</i>
9.	Занесення роботи в електронний архів	18 травня 2022.	<i>виконано</i>
10.	Допуск до захисту в зав. кафедри	19 травня 2022	<i>виконано</i>

Дата видачі завдання «28» березня 2022 р.

Керівник, проф. _____ Наталія БІЛОУС

Завдання прийняв до виконання _____ (Черненко Є.А.)

РЕФЕРАТ / ABSTRACT

Кваліфікаційна робота магістра містить: 75 с., 21 рис., 6 дод., 25 джерел.

ДОПОВНЕНА РЕАЛЬНІСТЬ, МАШИНЕ НАВЧАННЯ, АНОТАЦІЯ, COREML, ARKIT, TENSORFLOW.

Об'єктом дослідження є підходи до розпізнавання та анотації об'єктів в доповненій реальності.

Метою роботи є аналіз проблем розпізнавання та локалізації об'єктів та методи їх вирішення, а також пошук оптимального способу інтеграції розпізнавання об'єктів в доповненій реальності.

Методи розробки базуються на таких технологіях, як Swift, ARKit, Vision, а також інструменти машинного навчання TensorFlow та CoreML.

В результаті роботи було досліджено підходи до реалізації розпізнавання та анотації об'єктів в доповненій реальності використовуючи архітектуру виявлення об'єктів R-CNN та інтеграції технології доповненої реальності ARKit.

AUGMENTED REALITY, MACHINE LEARNING, ANOTATION, COREML, ARKIT, TENSORFLOW.

The object of the study is the problem of recognition and annotations of objects in augmented reality.

The aim of the work is the analysis of the problems of recognition and localization of objects and the methods of their development, as well as the search for the optimal way of integrating the recognition of objects in augmented reality.

Development methods are based on such technologies as Swift, ARKit, Vision, as well as TensorFlow and CoreML machine learning tools.

As a result, the work was carried out to the implementation of the recognition of the annotation of objects in augmented reality using the architecture of objects manifestation R-CNN and the integration of augmented reality technology ARKit.

Умови публікації пояснювальної записки

Я, _____ Черненко Єгор Анатолійович _____
(прізвище, ім'я, по батькові)

студент групи ІІЗм-20-2 здобувач вищої освіти на другому (магістерському) рівні

кафедра програмної інженерії _____,
(повна назва кафедри)

заявляю: моя кваліфікаційна робота на тему Дослідження підходів до розпізнавання та анотації об'єктів в доповненій реальності для путівників.,

(назва роботи)

що буде представлена до ЕК для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу EIArKhNURE. Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

ЗМІСТ

ВСТУП.....	8
1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ.....	11
1.1 Опис предметної галузі.....	11
1.2 Аналіз підходів та методів рішення.....	14
1.3 Аналіз існуючих систем.....	21
1.4 Обґрунтування цілей дослідження.....	25
2 АНАЛІЗ ТА ОБМЕЖЕННЯ МЕТОДІВ РІШЕННЯ.....	27
2.1 Локалізація об’єктів розпізнавання.....	27
2.2 Швидкість розпізнавання в режимі реального часу.....	29
2.3 Обмеження даних.....	31
2.4 Аналіз продуктивності GPU на операційній системі iOS.....	32
3 ВИБІР МЕТОДУ НА ПОСТАНОВКА ЗАДАЧІ.....	37
3.1 Вибір методу локалізація об’єктів.....	37
3.2 Вибір методу реалізації доповненої реальності.....	39
3.3 Основні функції мобільного додатку.....	41
3.4 Проектування архітектури програмного забезпечення.....	43
3.5 Інтеграція CoreML та ARKit.....	44
ВИСНОВКИ.....	47
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	49
ДОДАТОК А.....	Error! Bookmark not defined.
Перелік джерел посилання за науковими напрямками керівника ...	Error! Bookmark not defined.
та науковців кафедри програмної інженерії.....	Error! Bookmark not defined.
ДОДАТОК Б.....	Error! Bookmark not defined.
Звіт результатів перевірки на унікальність тексту ...	Error! Bookmark not defined.
ДОДАТОК В.....	Error! Bookmark not defined.
Слайди презентації.....	Error! Bookmark not defined.

ДОДАТОК Г	Error! Bookmark not defined.
Код програми	Error! Bookmark not defined.
ДОДАТОК Д	Error! Bookmark not defined.
Апробація роботи.....	Error! Bookmark not defined.
ДОДАТОК Е.....	Error! Bookmark not defined.
Експертний висновок результатів перевірки кваліфікаційної роботи на відповідність оформлення вимогам ДСТУ	Error! Bookmark not defined.

ВСТУП

Доповнена реальність тренд сучасності, адже відкриває новий вид взаємодії людини та комп'ютерних систем. Це не просто модна технологія, вона справді має потенціал у багатьох сферах нашого життя, наприклад: освіта (особливо у галузях, де візуалізації допомагає засвоєнню матеріалу), роздрібна торгівля (віртуальні примірки одягу та предметів інтер'єру), ігри, медицина, туризм та багато іншого.

Машинне навчання вже більше інтегроване у сучасне програмне забезпечення та прекрасно справляється з рутинними завданнями задля збереження людського ресурсу.

На сьогодні є актуальною проблема взаємодії цих технологій, адже користувацький досвід використання доповненою реальності може бути суттєво покращений за допомогою використання машинного навчання, а саме розпізнавання об'єктів.

Метою роботи є створення оптимального способу інтеграції розпізнавання об'єктів в доповненій реальності.

Об'єктом дослідження є аналіз проблем розпізнавання та локалізації об'єктів та методи їх вирішення.

Предметом дослідження є дослідження підходів до розпізнавання та анотації об'єктів в доповненій реальності для путівників.

Методом дослідження є порівняння та пошук збалансованого засобу реалізації доповненої реальності та анотації об'єктів за допомогою тестів окремих технологій та їх способів взаємодії. Також, досліджується продуктивність даної інтеграції, адже всі обчислення проходять на смартфонах, які мають значно менші потужності за сучасні комп'ютери чи консолі.

Елементом наукової новизни є підхід до та технічне рішення взаємодії двох технологій раніше не використаних сумісно, доповненої реальності та розпізнавання об'єктів з анотацією.

Результатом роботи є отриманий програмний продукт, в якому реалізовано використання доповненої реальності з анотацією розпізнаних об'єктів на мобільному пристрою в режимі реального часу.

Популярність туризму з кожним роком б'є рекорди, ми часто бачимо звіти у СМІ, що якесь європейське місто відвідала рекордна кількість людей. Це ставить під сумнів багато звичних речей, таких як транспорт, готельно-рестораний бізнес та інше, адже з великим навантаженням з'являється потреба до оптимізації багатьох процесів [1].

Навантаження на гідів також зростає, особливо в часи «високого сезону», коли для того щоб завітати на екскурсію по місту, треба займати чергу або реєструватись заздалегідь. Існує альтернативний варіант – розробити план маршруту самостійно та власноруч знаходити інформацію про історію міста, про відомих людей які там проживали або інші цікаві факти. Так, це можливо це найефективніший спосіб дізнатися, те що саме цікаво насамперед туристу, але це дуже часозатратний процес і ціна помилки дуже висока, зазвичай мандрівник має обмежену кількість днів у подорожі, тому небажано гаяти час заблукавши десь у місті, тим паче це може бути небезпечно.

Сучасні мобільні пристрою дозволяють отримати інформацію в одну мить. Також, існують безліч допоміжних аплікацій, які стануть у нагоді мандрівнику, наприклад: офлайн карти, купівля білетів на транспорт та бронювання готелів. Крім цього, існують ще агрегатори, які просто збирають все до купи, але плата за таку уніфікацію – це досить поверхнева інформація про все аспекти туризму в тій чи іншій точці світу.

Останнім часом все більшу популярність здобувають додатки «аудіогіди» або «маршрути». Головним їх недоліком є якість та обмежений функціонал.

Програмна система для путівників на прикладі додатку «Мандри по Харкову» використовує анотацію об'єктів в доповненій реальності, що забезпечує

новий рівень взаємодії користувача та смартфона. Наприклад, навести камеру девайсу на історичну пам'ятку та побачити в доповненій реальності якусь анімацію або сцену. Також, за допомогою ресурсів 3x вимірного моделювання та AR відобразити зруйновані споруди в їх первозданному вигляді.

Основні положення роботи доповідалось на XII Міжнародній науково-практичній конференції “MODERN DIRECTIONS OF SCIENTIFIC RESEARCH DEVELOPMENT”, (м. Чикаго, США, 2022).

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Опис предметної галузі

Проведений на сьогодні аналіз предметної галузі показав що, в наш час з'явилась тенденцію до розвитку метавсесвітів, оскільки люди все більше і більше використовують інтернет, який інтегрувався у кожний аспект нашого життя, від навчання та роботи до медицини та романтичних відносин. На сьогодні звичайних інтерфейсів як комп'ютер чи телефон вже недостатньо, тому що вони вже не в змозі задовольнити всі потреби людства.

Слово «метавсесвіт» часто пов'язане з дистопічним, кіберпанковим романом Ніла Стівенсона «Snow Crash» 1992 року, і багато хто бачить новітнє натхнення в сліпучому колоді досвіду, що лежить в основі роману Ернеста Клайна 2011 року «Першому гравцю приготуватись». Однак метавсесвіт далекий від наукової фантастики [2].

Онлайн-спільноти існували принаймні з середини 1980-х років і зросли в 1990-х роках за допомогою чат-кімнат, програми миттєвих повідомлень AOL і перших сайтів соціальних мереж. Гра World of Warcraft стала постійною соціальною сценою для мільйонів на початку 2000-х років, і спільноти продовжували розвиватися в іграх і навколо них. Сьогодні увійти у Fortnite, приєднатися до чату з друзями через консольну платформу та почати гру з ними, особливо для молодших поколінь, є таким же соціальним досвідом, як і більшість інших фізичних взаємодій.

У віртуальній реальності (VR), доповненій реальності (AR) або просто на екрані, метавсесвіт обіцяє забезпечити більше перекриття нашого цифрового та фізичного життя в багатстві, соціалізації, продуктивності, покупках та розвагах. Ці два світи вже переплетені, гарнітура не потрібна: подумайте про додаток Uber, який повідомляє вам за даними про місцезнаходження, наскільки далеко

знаходиться автомобіль. Подумайте, як Netflix оцінює те, що ви дивилися раніше, щоб зробити пропозиції. Подумайте, як сканер LiDAR на новіших iPhone може виконувати 3D-сканування вашого оточення. За своєю суттю метавсесвіт (також відомий багатьом як «Web3») є еволюцією нашого нинішнього Інтернету.

Туризм це ще одна перспективна галузь метавсесвіту, яка має нескінчену кількість варіантів використання «Web3». Вже сьогодні кожний мандрівник має декілька додатків у своєму смартфоні, які мають спільну місію – полегшити життя туриста та зменшити в стресс на новій і невідомій місцевості. Більш продвинуті мандрівники мають на своєму девайсі більше десяти додатків категорії «Travel», серед них: пошук житла («Booking», «AirBnB»), пошук авіаквитків («Flight Hooper», «Aviasales»), навігація («maps.me», «Waze») та сервіси рекомендації («TripAdvisor») [3].

На сьогодні людство ще не готово для повноцінного занурення до метавсесвіту, на це існує багато цілком обґрунтованих причин. По-перше, це стосується вимог передових цифрових технологій. Metaverse пропонує багато нових і передових технологій, таких як гарнітури VR, тактильні пристрої, блокчейн та інші вимоги. Однак не кожна людина на цій планеті має доступ до передових технологій. Наприклад, швидке підключення до Інтернету є однією з обов'язкових вимог для участі в метавсесвіті[4]. Більшість людей у всьому світі не мають доступу до швидкого Інтернету і не можуть використати весь потенціал метавсесвіту. Крім того, недоліки діджитал світу також вказують на необхідність розширених комунікаційних засобів і гаджетів. Багато людей не можуть дозволити собі високоякісні гарнітури VR для входу в метавсесвіт. По-друге, багато з існуючих сьогодні цифрових рішень пов'язані з проблемами конфіденційності та безпеки. Основний порядок денний для критики цифрових рішень зводиться до того, що вони збирають дані від користувачів. Такі дані можна використовувати для нав'язливої онлайн-реклами та крадіжки особистих даних. Крім того, компанії не змогли повністю вирішити ці проблеми. Отже, мінуси метавсесвіту також можуть викликати проблеми, пов'язані з ризиками конфіденційності та безпеки. Будучи онлайн-простором, метавсесвіт може

призвести до нових проблем у безпеці та конфіденційності для окремих осіб та установ.

Альтернативною технологією взаємодії з метавсесвітом є «Augmented Reality», або скорочено AR (див. рис. 1.1).



Рисунок 1.1 – Приклад доповненої реальності [5]

AR — це інтерактивний досвід реального середовища, де об'єкти, що знаходяться в реальному світі, покращуються за допомогою комп'ютерної перцептивної інформації, використовуючи візуальні сенсори. AR можна визначити як систему, яка включає в себе три основні функції: комбінацію реального та віртуального світів, взаємодію в реальному часі та точну 3D-реєстрацію віртуальних і реальних об'єктів. Накладена сенсорна інформація може бути конструктивною (тобто доповненням до природного середовища) або деструктивною (тобто маскуванням природного середовища). Цей досвід плавно переплітається з фізичним світом таким чином, що він сприймається як аспект реального середовища, що занурює в них. Таким чином, доповнена реальність змінює постійне сприйняття реального середовища, тоді як віртуальна реальність повністю замінює реальне середовище користувача змодельованим.

Доповнена реальність пов'язана з двома переважно синонімічними термінами: змішана реальність і комп'ютерно-опосередкована реальність. доповнена реальність.

Одна з недоцінених груп аплікацій та потенційної сферою AR вважається – путівники. Зазвичай вони використовуються туристами для кращого орієнтування у незнайомій місцевості. Композиція путівника часто підпорядкована рекомендованим маршрутам огляду пам'яток місцевості, що описується, що містить відомості про країну, регіон, місто, туристичний маршрут. Найбільш популярні в містах з великою історією та високою концентрацією історичних пам'яток. Але наразі немає жодного справді сучасного додатка путівника, який би задовольняв вимогам сучасності. Саме тому існує безліч можливостей для розвитку цієї ідеї.

1.2 Аналіз підходів та методів рішення

З використанням сучасних технологій розпізнавання образів є можливість поліпшити користувацький досвід засобів путівника. На сьогодні існує безліч можливостей реалізувати засіб розпізнавання об'єктів. Технології які використовуються цілком задовольняють сучасні потреби, тому можна озброїтись майже будь-якою. Головне це – знайти або створити правильний датасет для навчання моделі і вдало реалізувати свій додаток. Популярні підходи на основі глибокого навчання, що використовують згорткові нейронні мережі (CNN), такі як R-CNN (див. рис. 1.2) і YOLO v2, автоматично навчаються виявляти об'єкти в зображеннях. Існує два ключових підходи виявлення об'єктів за допомогою глибокого навчання.

Перший, створення та навчання користувацького детектора об'єктів. Щоб навчити користувацький детектор об'єктів з нуля, потрібно розробити архітектуру мережі, щоб дізнатися про особливості що притаманні об'єкту. Також, потрібно зібрати дуже великий набір позначених даних для навчання CNN [7]. Результати спеціального детектора об'єктів можуть бути чудовими. Тим не менш,

налаштування шарів та ваг в CNN вручну вимагає багато часу та навчальних даних.

Другий спосіб, це – заздалегідь підготовлений детектор предметів. Багато робочих процесів виявлення об'єктів, що використовують глибоке навчання, використовують навчання передачі, підхід, який дає змогу почати з попередньо навченої мережі, а потім точно налаштувати її для своєї програми. Цей метод може забезпечити швидші результати, оскільки детектори об'єктів вже навчені на тисячах або навіть мільйонах зображень. Отже, для даної програмної системи буде використано підготовлений детектор предметів.

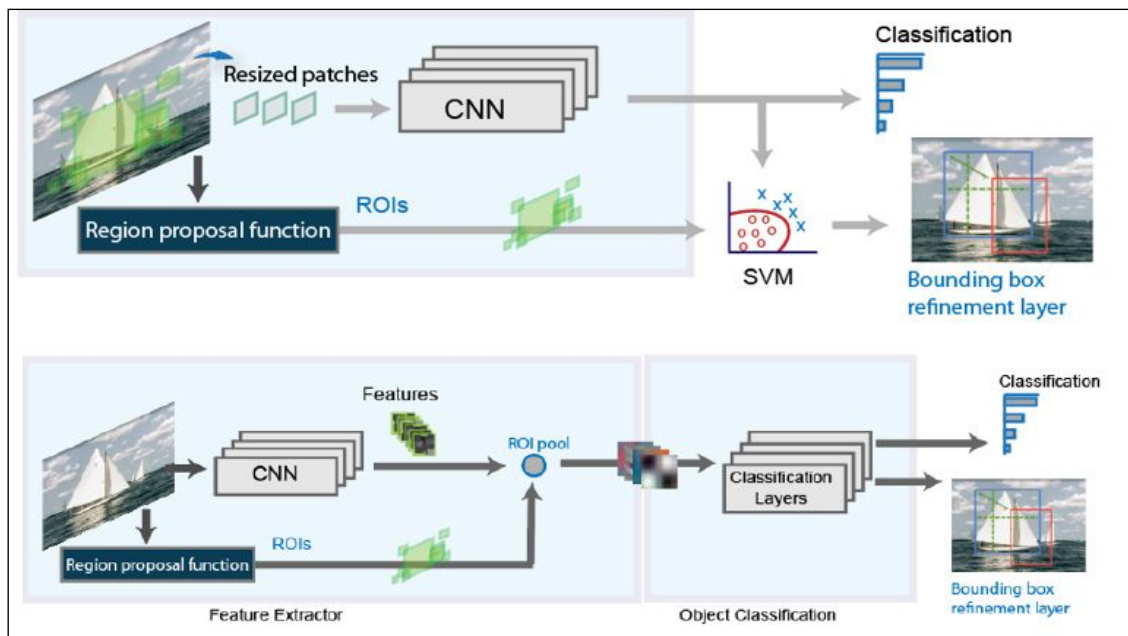


Рисунок 1.2 – Високорівнева архітектура виявлення об'єктів R-CNN (зверху) і швидкого R-CNN (знизу) [6]

Після визначення детектору треба визначитись з типом мережі. Існує два основних види: одно і двоетапна.

Двоетапна мережа, така як R-CNN та її варіанти, визначає пропозиції регіонів або підмножини зображення, які можуть містити об'єкт. На другому етапі класифікуються об'єкти в межах регіону пропозицій. Двоступеневі мережі можуть досягати дуже точних результатів виявлення об'єктів; однак вони, як правило, повільніші, ніж одноступінчасті мережі.

В одноступінчатих мережах, таких як YOLO v2 (див. рис. 1.3), CNN створює мережеві прогнози для регіонів по всьому зображенню за допомогою блоків прив'язки, і передбачення декодуються для генерування кінцевих обмежувальних рамок для об'єктів. Одноступінчасті мережі можуть бути набагато швидшими, ніж двоступеневі мережі, але вони можуть не досягати такого ж рівня точності, особливо для сцен, що містять невеликі об'єкти.

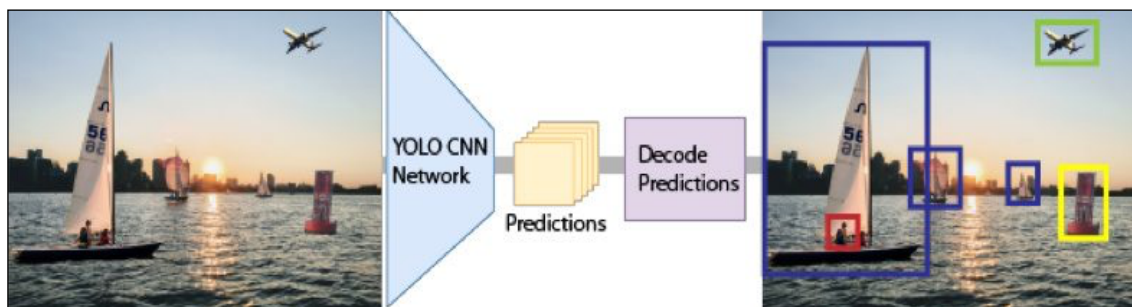


Рисунок 1.3 – Огляд виявлення об'єктів YOLO v2 [6]

Оскільки для додатку путівника маленькі об'єкти не є пріоритними, тому що, більшість визначних пам'яток є або меморіали, або споруди. Таким чином одноетапна модель YOLO v2 є вдалим вибором для реалізації заданої програмної системи.

Наступна особливо важлива частина в розробці моделі це збір та тренування моделі. Для того, щоб розпочати тренування треба підготувати датасет, це можна зробити за допомогою спеціальної програми розмітки [8]. Зазвичай ці програми дуже прості та мають один і той же функціонал, тому був обраний інструмент LabelBox (див. рис 1.4), оскільки він надає безкоштовну ліцензію і має зручний інтерфейс.

Для створення датасету було вилучено фотокартки визначних пам'яток Харкова, які доступні в інтернеті.

Отримавши підготовлений датасет його можна використати для «донавчання» моделі YOLO v2. Для тренування також існує багато інструментів, але через те що вона буде використовуватись на мобільній платформі iOS вибір дещо скорочується до двох TensorFlow та CoreML. TensorFlow — це безкоштовна

бібліотека програмного забезпечення з відкритим вихідним кодом для машинного навчання та штучного інтелекту. Його можна використовувати в ряді завдань, але він зосереджений на навчанні та висновку глибоких нейронних мереж. Це дуже популярний інструмент та має багато мануалів використання, але він більш придатний використання для платформи Android.

CoreML — це базова платформа машинного навчання від Apple, яка базується на шейдерах Accelerate, BNNS і Metal Performance. Core ML оптимізує продуктивність на пристрої, використовуючи центральний процесор, графічний процесор і Apple Neural Engine (ANE), мінімізуючи при цьому обсяг пам'яті та енергоспоживання. Він надає моделі машинного навчання, які можна інтегрувати в програми iOS, і підтримує аналіз зображень, обробку природної мови, перетворення звуку в текст і аналіз звуку.



Рисунок 1.4 – Набір визначних фотокарток визначних пам'яток Харкова.

Програми можуть скористатися перевагами CoreML без необхідності підключення до мережі або викликів API, оскільки платформа CoreML працює за допомогою обчислень на пристрої. Виконання моделі суворо на пристрої користувача усуває будь-яку потребу в мережевому з'єднанні, що допомагає зберегти конфіденційність даних користувача і реагувати ваш додаток. Головним

недоліком напевно є те що CoreML це дуже простий інструмент, який немає багато налаштувань. Насамперед цей інструмент орієнтований на людей які не дуже розуміються на машинному навчанні або мають лише базові навички. Крім того, бібліотека орієнтована на iOS та MacOS, тому це створює певні перешкоди для розробників які використовують інші платформи. Наприклад, якщо команда має окрему людину яка займається машинним навчанням та використовує дистрибутив Linux чи Windows, але це можна вирішити іншим способом.

Існує більше продвинутый інструмент для роботи з CoreML, для його використання підійде будь-яка операційна система. Все, що потрібно розробнику це вміння використовувати мову програмування Python та базові навички роботи з командною строкою. Назва цієї бібліотеки Coremltools, вона дозволяє створювати CoreML моделі без прив'язки до платформи. Ця бібліотека має значні переваги над своїм GUI аналогом, завдяки більш точному процесу навчання, який можна зупинити та виправити на будь-якому етапі. Тим більше coremltools конвертувати моделі зі сторонніх навчальних бібліотек, таких як TensorFlow і PyTorch, в Core ML (див. рис. 1.5).

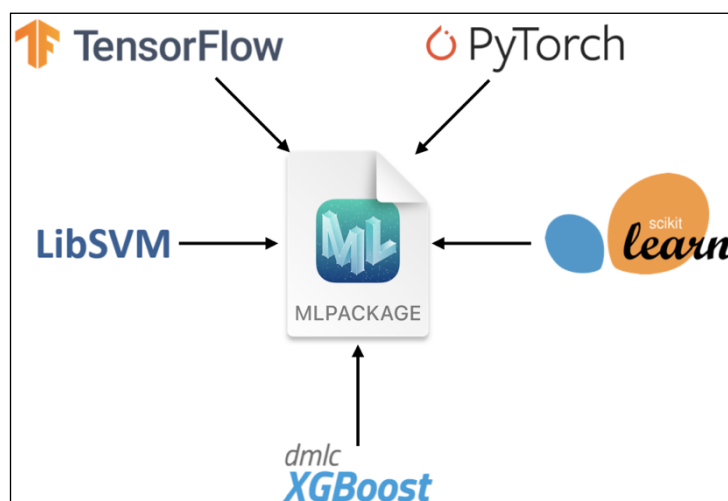


Рисунок 1.5 – Приклад моделей які можна конвертувати в CoreML подібну.

Модель зроблену за допомогою TensorFlow можна конвертувати в CoreML подібну, але в цьому немає жодного сенсу, оскільки YOLO v2 доступний для CoreML.

Для створення так званого метавсесвіту, який побудований на потужностях доповненої реальності. Доповнений метавсесвіт (тобто злиття реального і віртуального світів в єдину імерсивну та єдину реальність) результат введення в зорове поле будь-яких сенсорних даних з метою доповнення відомостей про оточення та зміни сприйняття навколишнього середовища. Інструментів реалізацій AR існує безліч: Vuforia, Wikitude, ARKit, ARCore, ARToolKit. Вони так само, як і інструменти для машинного навчання мають певні обмеження стосовно до платформи використання, тому ARCore та ARToolKit не будуть розглянуті, оскільки вони виключають можливість їх використання на операційній системі iOS.

Vuforia – це SDK доповненої реальності, який займає перше місце в більшості списків «Найпопулярнішої доповненої реальності» і не без підстав. Vuforia пропонує ряд продуктів для розробки AR досвіду, включаючи Vuforia Engine, Studio і Chalk. Програмне забезпечення підтримує створення як на основі маркерів, так і без маркерів AR і має кілька ключових функцій, які роблять його одним із найкращих для розпізнавання об'єктів та 3D-моделювання. Ці функції включають Ground Plane (для додавання вмісту до горизонтальних поверхонь), Visual Camera (розширює підтримувані візуальні джерела за межі мобільних телефонів і планшетів) і VuMarks (користувацькі маркери, які можна використовувати для розпізнавання обличчя Vuforia, а також кодування даних). Головним недоліком цієї бібліотеки, це складність та надлишкова завантаженість, яка недоречна для аплікації для путівників, оскільки там будуть використовуватись базові функції AR.

Інший інструмент розроблений компанією Apple, отже він має повну інтеграцію з системою iOS і використовує ресурси мобільного пристрою дуже ефективно. ARKit покладається на дані датчика камери та додаткові дані (наприклад, з гіроскопа та акселерометра), щоб виявляти й аналізувати оточення користувачів для візуалізації AR. ARKit також підтримує швидке відстеження руху, відстеження обличчя, Quick Look (відображення моделей і сцен, які можна

легко переміщати і масштабувати) і різні ефекти візуалізації. Apple докладас зусиль, щоб зробити AR простою та зрозумілою для розробників [10].

Варто зауважити, що також існує платформа Unity для роботи з доповненою реальністю, але вона суттєво збільшує вага додатку та накладає обмеження на інтеграцію з CoreML.

Унікальність програмної системи для путівників досягається комбінацією інструментів машинного навчання та доповненої реальності.



Рисунок 1.6 – Приклад інтеграції CoreML та ARKit

Комбінація цих технологій дозволить зробити розпізнавання та анотацію історичних пам'яток прямо в смартфоні користувача. Окрім цього, це відкриває широкий спектр можливостей для ознайомлення туристів з минулим нашої батьківщини. Наприклад, визначні споруди чи пам'ятки зруйновані під час

підступної агресії Росії навесні 2022 року можна буде відобразити на екрані смартфона, як ці будівлі виглядали до бойових дій.

Wikitude – це ще один чудовий вибір для розробки програмного забезпечення AR. Wikitude підходить для розробки додатків AR для пристроїв iOS, Android і Smart Glasses. Як і більшість найкращих програм розробки AR, Wikitude підтримує різноманітні методи та технології відстеження, але також включає геолокацію та функції масштабування на основі відстані. На жаль, цей інструмент не здобув широкого використання на платформі iOS, тому це може викликати певні складності під час розробки [9].

1.3 Аналіз існуючих систем

В мобільних магазинах додатків існує безліч аплікацій путівників по майже будь-якому місту, але майже всі мають досить низьку якість та обмежений функціонал.

В Інтернеті є майже все. Однак, щоб знайти серед океану «мусорної» інформації цінну, толкову та перевірену, ви потратите драгоціний час. Тому «живі відгуки», як правило, дуже суперечать. Діапазон оцінок одного і того ж готелю чи ресторану може варіюватись від «чудово» до «обходьте стороною за кілометр». А ще вони часто бувають рекламними.

Час бумажних путівників чи туристичних карт вже давно пройшов. Паперовий путівник довгий час був єдиним способом отримати перевірену корисну інформацію. На їх зміну прийшли тематичні форуми або спеціальні веб-сайти для багатьох країн, але для отримання інформації потрібно «нагунлити» та перебрати досить багато ресурсів. Тому майже кожне місто в світі має додатки цікавих місцевостей у своєму регіоні.

Наприклад, путівник по Стамбулу, найпопулярніший додаток по цій тематиці.

Напевно цей додаток виконує поставлену задачу, але воно трохи не відповідає вимогам сучасної програми і зовсім не використовує потенціал сучасних смартфонів.

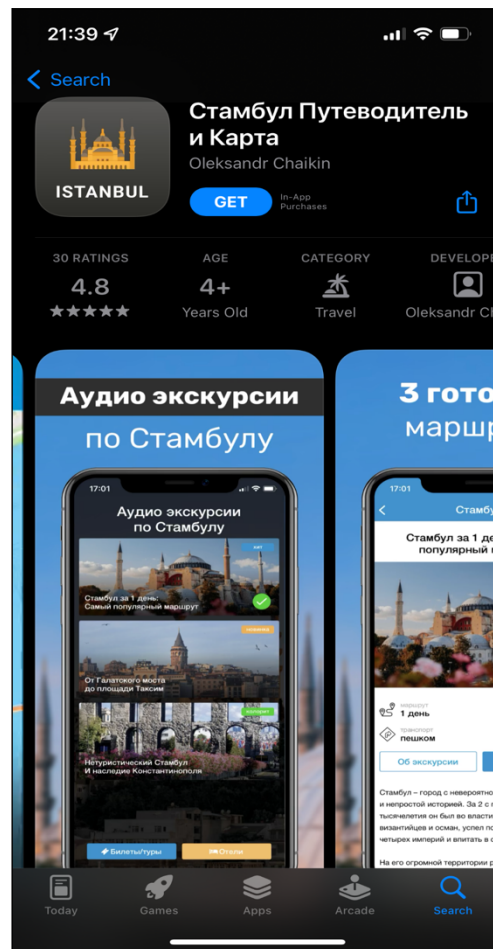


Рисунок 1.7 – Путівник по Стамбулу.

Основний функціонал це: карта історичних пам'яток (без навігації) та аудіогід. Звичайно, це не заважає йому користуватись не аби якою популярністю та мати досить високий рейтинг, але не заперечує той факт, що такий спосіб отримання інформації для планування подорожей сильно застарів та не відповідає вимогам сучасного світу. Подібних додатків безліч і вони забруднюють магазини мобільних додатків на всіх платформах. Це погано, тому що, люди звикають неякісного ПО.

Tripadvisor – це агрегатор для туриста, який нараховує безліч функцій, наприклад: список рекомендованих місць, бронювання готелю, відгуки та інше. Мабуть, саме через аплікація виглядає перезавантажено і не виконує більшість функцій належним чином.

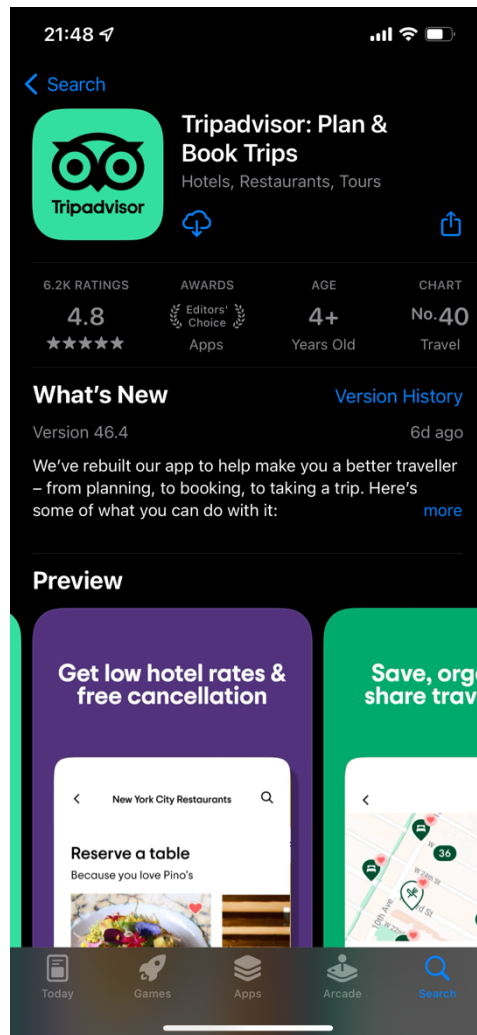


Рисунок 1.8 – Tripadvisor.

Розглянувши доступні додатки на даний момент, можна зробити висновки, що аналогів немає. Найбільш наближений функціонал має «Путівник по Стамбулу» та інші аплікації від розробника. Головною перевагою путівника з доповненою реальністю – є новий рівень інтеграції користувача та споживання інформації.

Варто зауважити, що компанія Snap яка є флагманом розробки мобільної доповненої реальності наразі тестує функціонал під назвою Landmarks [11].

Останнє оновлення Lens Studio тепер включає ще більше шаблонів для розробників доповненої реальності, зокрема Landmarkers, які дають можливість творцям створювати об'єктиви, які можуть перетворювати найбільш культові пам'ятки світу в режимі реального часу (див. рис. 1.9) .

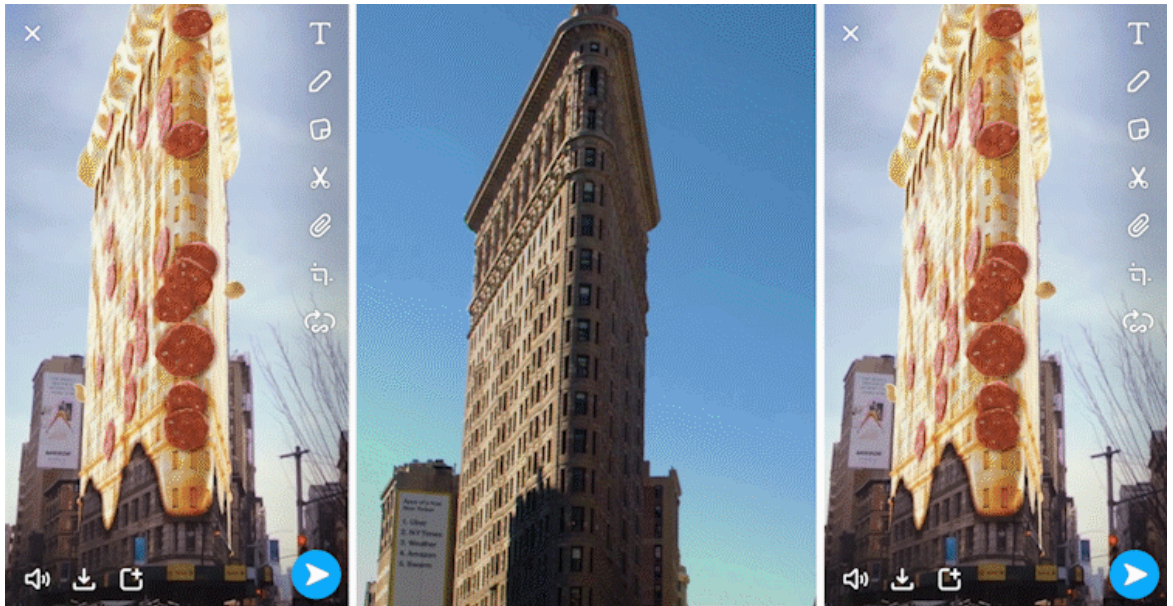


Рисунок 1.9 – Додаток Snap з прикладом Flatiron Building

Також є нові шаблони для відстеження рук, відстеження тіла тощо, які ви можете використовувати для створення лінз, які Snapchatters по всьому світу можуть розблокувати та грати з ними. Нажаль, зараз створення подібних сцен не доступне для широкого спектру користувачів.

Наступною подібною системою є Google Maps, яка включає функцію AR Navigation (рис. 1.10). Головною ідеєю є отримання підказок в режимі доповненої реальності.

В основі цієї реалізації лежить використання декількох тригерів для визначення місця розташування та напрямок підказки. До них відносяться геолокація та показники акселерометр.

Оновлена функція Live View AR, яка накладає цифрові довідники поверх реального світу, щоб надавати маршрути, коли ви дивитеся на дисплей свого телефону, тепер працює в приміщенні. Скажімо, ви перебуваєте в аеропорту і вам

потрібно знайти вихід або банкомат. Ви шукаєте те, що шукаєте на Картах Google, і маркери направлятимуть вас за допомогою стрілок та інших цифрових індикаторів.



Рисунок 1.10 – Google Maps AR Navigation

Існують менше популярні аналоги цього додатку, які використовуються для навігації в приміщеннях. В якості тригерів вони використовують спеціально позначки, QR-коди або технологію iBeacon для позиціонування користувача у приміщенні.

1.4 Обґрунтування цілей дослідження

Дослідити методи інтеграції розпізнавання об'єктів в доповненій реальності та реалізувати програмний продукт у вигляді мобільно додатку для платформи iOS. Який призначений для мандрівників або просто туристів які хочуть дізнатися про певне місто або країну використовуючи інтерактивний додаток. Даний

програмний продукт цілком може замінити дорого гіда, крім цього, туристи не повинні слідувати чіткими вказівкам чи дотримуватись певного маршруту, та можуть обрати його самі.

Головним чином аплікація стане у нагоді людям, яким цікава історія міста та споруд, деякі зруйновані об'єкти можна буде побачити у первозданному вигляді або побачити анімацію навколо архітектурної пам'ятки у режимі доповненої реальності.

Мобільні додатки на разі є один з найбільш популярних засобів взаємодії людини зі смартфоном. Переважна частина користувачів обирає окрему аплікацію, якщо вона доступна, а не мобільну версію сайту. Якісний мобільний додаток є більш чуйним аніж веб-сайт, тому що, обробка зображень відбувається безпосередньо на девайсі використовуючи вбудовану відео-карту, коли мобільна версія сайту відкрити у браузері обробляється процесором та має певний проміжний шар, який впливає на плавність роботи. Також, перевагою аплікацій є те, що доступ до апаратних можливостей мобільно пристрою значно простіший та ефективніший за будь-який веб-сайт.

Озброївшись цими знаннями було вирішено реалізувати мобільний додаток для путівників, адже за допомогою сучасних технологій обробки зображень, які можна запускати прямо на смартфоні можна досягти особливих переваг в економії трафіку (який може бути доволі дорогим в певних країнах) та швидкодії, оскільки не потребує обміну важкими медіа файлами з сервером.

2 АНАЛІЗ ТА ОБМЕЖЕННЯ МЕТОДІВ РІШЕННЯ

2.1 Локалізація об'єктів розпізнавання.

Першим серйозним ускладненням виявлення об'єктів є його додаткова мета – визначення положення об'єктів, що зазвичай називають завданням локалізації об'єктів (див. рис. 2.1) [12].

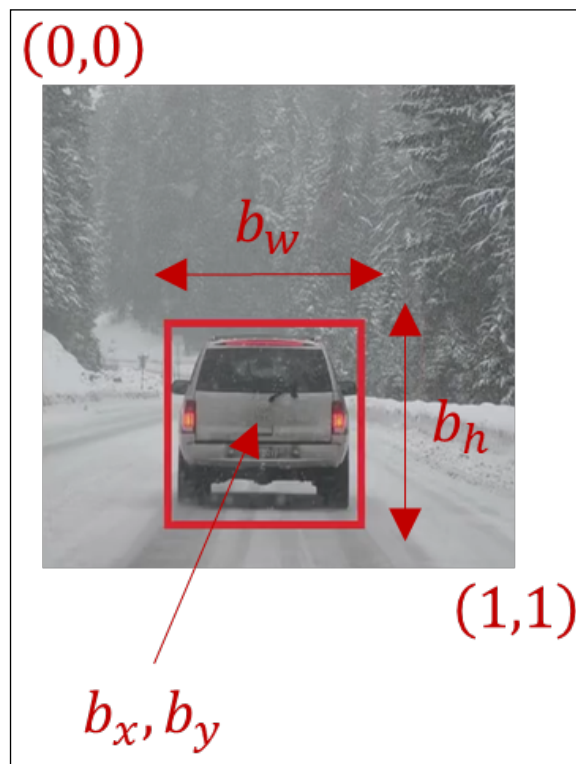


Рисунок 2.1 – Приклад результату алгоритму локалізації.

Указуючи обмежувальний прямокутник, червоний прямокутник вимагає вказати середину, тож це точка b_x, b_y , а також висота, яка буде b_h , а також ширина b_w цієї обмежувальної рамки. Тепер, якщо навчальний набір містить не тільки мітку класу об'єкта, яку наша нейронна мережа намагається передбачити тут, але він також містить 4 додаткові числа, що дають обмежувальний квадрат, тоді ми можемо використовувати навчання з наглядом, щоб вивести наш

алгоритм не просто мітку класу, а також 4 параметри, які вказують нам, де знаходиться обмежувальна рамка об'єкта, який ми виявили. У цьому прикладі b_x може бути приблизно 0,5, тому що це приблизно половина шляху праворуч до зображення, b_y може бути десь 0,7, оскільки це 70% шляху вниз до зображення, b_h може бути приблизно 0,3, оскільки висота цього червоного квадрата становить 30 % загальної висоти зображення, а b_w може становити 0,4, оскільки ширина червоного квадрата становить 0,4 від загальної ширини всього зображення.

Щоб вирішити цю проблему, дослідники найчастіше використовують функцію втрати багатозадачності, щоб уникнути як неправильну класифікацію, так і помилки локалізації [14].

Регіональні CNN представляють один популярний клас фреймворків виявлення об'єктів. Ці методи полягають у створенні пропозицій регіонів, де, ймовірно, будуть розташовані об'єкти, з подальшою обробкою CNN для класифікації та подальшого уточнення розташування об'єктів. Росс Гіршик та ін. розробили Fast R-CNN, щоб покращити свої початкові результати з R-CNN. Як випливає з назви, Fast R-CNN забезпечує значне прискорення, але точність також підвищується, оскільки завдання класифікації та локалізації оптимізовані за допомогою однієї уніфікованої функції втрати багатозадачності. Ефективність моделі для класифікації зображень оцінюється за допомогою середньої помилки класифікації для передбачених міток класів. Ефективність моделі для локалізації окремого об'єкта оцінюється за допомогою відстані між очікуваним і прогнозованим обмежуючим квадратом для очікуваного класу. У той час як продуктивність моделі для розпізнавання об'єктів оцінюється за допомогою точності та запам'ятовування кожного з найкращих відповідних обмежувальних рамок для відомих об'єктів на зображенні.

2.2 Швидкість розпізнавання в режимі реального часу.

Алгоритми виявлення об'єктів повинні не тільки точно класифікувати та локалізувати важливі об'єкти, але й бути неймовірно швидкими під час прогнозування, щоб відповідати вимогам обробки відео в реальному часі [15]. Кілька ключових удосконалень протягом багатьох років підвищили швидкість цих алгоритмів, покращивши час тестування з 0,02 кадру в секунду (fps) R-CNN до вражаючих 155 кадрів в секунду для Fast YOLO.

Fast R-CNN і Faster R-CNN мають на меті прискорити оригінальний підхід R-CNN (див. рис. 2.2). R-CNN використовує вибіркового пошуку для створення 2000 регіонів, що представляють інтерес (RoIs) і передає кожен RoI через базу CNN окремо, що спричиняє масове вузьке місце, оскільки обробка CNN є досить повільною. Швидкий R-CNN натомість надсилає все зображення через базу CNN лише один раз, а потім узгоджує ROI, створені за допомогою вибіркового пошуку, з картою функцій CNN, що дозволяє скоротити час обробки в 20 разів.

Хоча Fast R-CNN набагато швидший, ніж R-CNN, існує ще один швидкісний бар'єр. Швидкому R-CNN потрібно приблизно секунду для виявлення об'єкта на одному зображенні, а вибіркового пошуку займає менше секунди. Швидший R-CNN замінює селективний пошук окремою субнейронною мережею для генерації RoI, прискорюючи ще в 10 разів і таким чином швидкість досягає приблизно 7–18 кадрів в секунду.

Незважаючи на ці вражаючі покращення, відео зазвичай знімаються зі швидкістю принаймні 24 кадри в секунду, що означає, що швидший R-CNN, швидше за все, не встигатиме. Регіональні методи складаються з двох окремих етапів: пропозиції регіонів та їх обробки. Таке поділ завдань виявляється дещо неефективним. Інший основний тип систем виявлення об'єктів замість цього спирається на єдиний підхід з одним станом. Ці так звані одноразові детектори повністю локалізують і класифікують об'єкти під час одного проходу по зображенню, що значно скорочує час перевірки.

Один з таких одноразових детекторів YOLO починається з розкладки сітки над зображенням і дозволяє кожній клітинці сітки виявляти фіксовану кількість об'єктів різного розміру. Для кожного справжнього об'єкта, наявного на зображенні, клітинка сітки, пов'язана з центром об'єкта, відповідає за прогнозування цього об'єкта. Складна, багатотермінова функція втрат гарантує, що вся локалізація та класифікація відбуваються в рамках одного процесу. Одна версія цього методу, Fast YOLO, навіть досягла швидкості 155 кадрів в секунду; однак точність класифікації та локалізації різко падає на цій підвищеній швидкості.

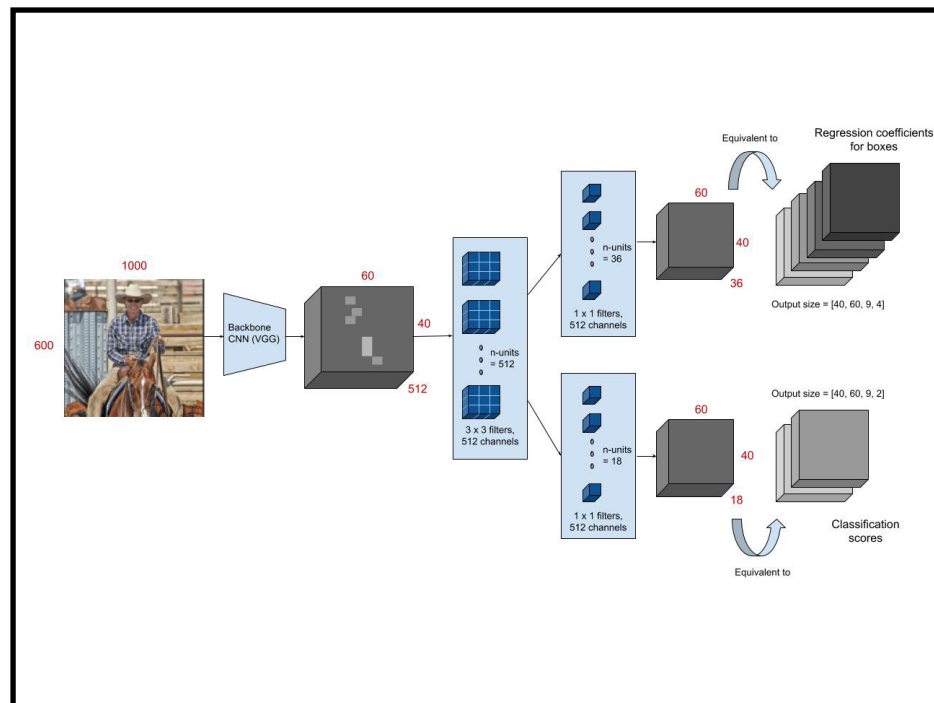


Рисунок 2.2 – Алгоритм роботи Fast R-CNN

Зрештою, сучасні алгоритми виявлення об'єктів намагаються знайти баланс між швидкістю та точністю. На ці результати впливають декілька варіантів дизайну за межами рамки виявлення. Наприклад, YOLOv3 дозволяє зображення з різною роздільною здатністю: зображення з високою роздільною здатністю зазвичай мають кращу точність, але повільніший час обробки, і навпаки, для зображень із низькою роздільною здатністю. Вибір бази CNN також впливає на співвідношення швидкості та точність.

2.3 Обмеження даних

Обмежена кількість анотованих даних, доступних наразі для виявлення об'єктів, виявляється ще однією істотною перешкодою. Набори даних виявлення об'єктів, як правило, містять приклади реальної істини для приблизно від десятка до ста класів об'єктів[16], тоді як набори даних класифікації зображень можуть включати понад 100 000 класів. Крім того, краудсорсинг часто безкоштовно створює теги класифікації зображень (наприклад, аналізуючи текст підписів до фотографій, наданих користувачем). Збір міток істини разом із точними обмежуючими рамками для виявлення об'єктів, однак, залишається неймовірно виснажливою роботою.

Набір даних COCO, наданий корпорацією Майкрософт, наразі є одним із найкращих доступних даних для виявлення об'єктів. COCO містить 300 000 сегментованих зображень з 80 різними категоріями об'єктів з дуже точними мітками розташування. Кожне зображення містить в середньому близько 7 об'єктів, і елементи відображаються в дуже широких масштабах. Яким би корисним не був цей набір даних, типи об'єктів за межами цих 80 вибраних класів не будуть розпізнаватися, якщо навчання проводиться виключно на COCO.

Дуже цікавий підхід до зменшення дефіциту даних походить від YOLO9000, другої версії YOLO. YOLO9000 містить багато важливих оновлень у YOLO, але він також спрямований на скорочення розриву між наборами даних між виявленням об'єктів та класифікацією зображень. YOLO9000 тренується одночасно на COCO і ImageNet, наборі даних класифікації зображень із десятками тисяч класів об'єктів. Інформація COCO допомагає точно знаходити об'єкти, а ImageNet розширює «словник» класифікації YOLO. Ієрархічне дерево Word дозволяє YOLO9000 спочатку виявити концепцію об'єкта (наприклад, «тварина/собака»), а потім детально ознайомитися з деталями (наприклад, «сибірський хаскі»). Цей підхід, здається, добре працює для концепцій, відомих

COCO, як тварини, але погано працює на менш поширених концепціях, оскільки пропозиція RoI надходить виключно від навчання з COCO.

Важливу роль відіграє унікальність проекту та об'єктів які користувач в перушу чергу буде розпізнавати. Головним чином це пам'ятники або споруди, тому жодна з наявних моделей не підходить, але базову модель YOLO можна взяти за основу та до навчити її.

2.4 Аналіз продуктивності GPU на операційній системі iOS

Існує велика кількість технологій реалізованих для роботи з трьох вимірною графікою на мобільній платформі iOS. Далі йдеться про їх порівняння.

RealityKit — наймолодший SDK в сімействі технологій рендеринга Apple [17]. Цей високорівневий фреймворк був випущений у 2019 році. RealityKit створений для проектів AR / VR, має спрощені налаштування для роботи з багатьма користувачами та може використовуватися на iOS / macOS. Виконує багатопотокове відтворення. Немає застарілого Objective-C, RealityKit підтримує тільки Swift і, швидше, декларативний синтаксис (як у SwiftUI). Основна перевага RealityKit – він може доповнювати/змінювати/налаштовувати сцени, що надходять із програми Reality Composer, і може бути потужним розширенням для ARKit, хоча він також виглядає як окремий AR SDK. У RealityKit основними одиницями є сутності (ModelEntity, AnchorEntity, TriggerVolume, BodyTrackedEntity, PointLight, SpotLight, DirectionalLight і PerspectiveCamera), які мають компоненти і можуть бути створені з таких ресурсів, як ModelEntity [18]. Фреймворк запускає систему компонентів об'єктів (ECS) на ЦП для керування такими завданнями, як фізика, анімація, обробка звуку та мережева синхронізація. Але він покладається на апаратне забезпечення Metal і GPU для виконання багатопотокового візуалізації. RealityKit має шість матеріалів: UnlitMaterial,

SimpleMaterial, PhysicallyBasedMaterial (з 18 AOV для контролю зовнішнього вигляду матеріалу), OcclusionMaterial, VideoMaterial і, звичайно, CustomMaterial. Данна технологія має наступну діаграму класів (див. рис. 2.3).

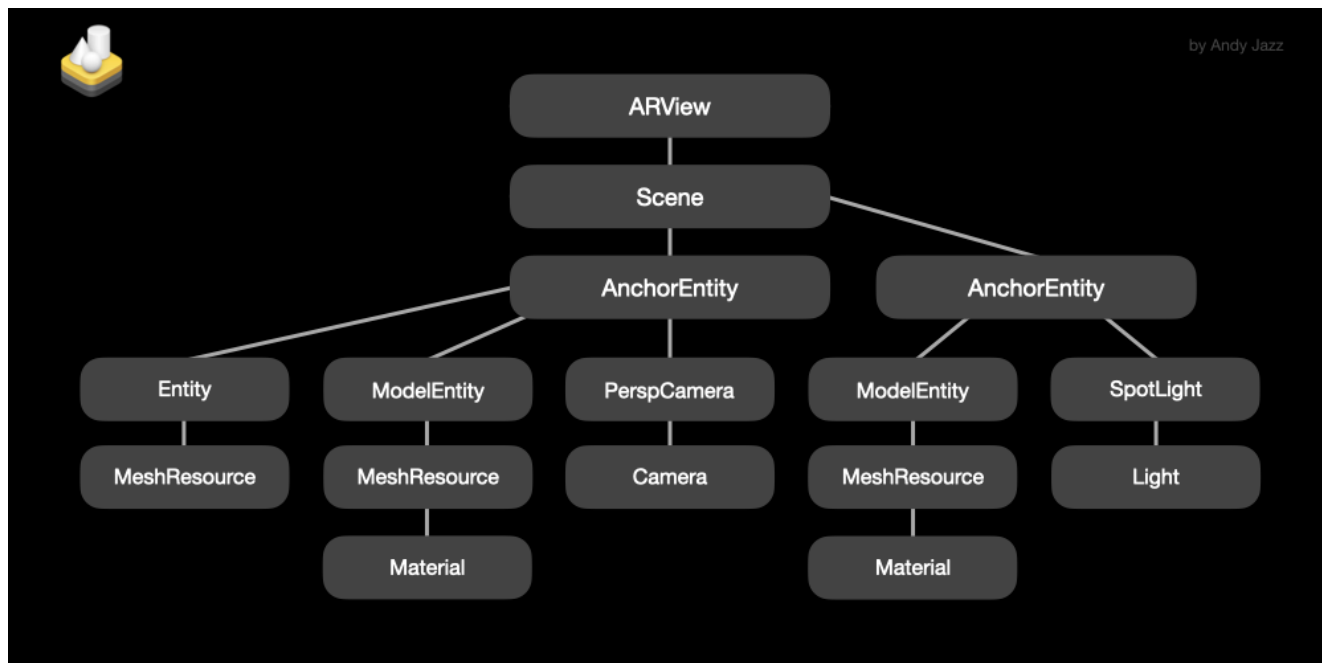


Рисунок 2.3 - Ієрархія RealityKit.

RealityKit читає файли у форматах .usdz, .rcproject та .reality. Підтримує анімацію трансформації та активів, динаміку твердого тіла, матеріали PBR, освітлення на основі зображення HDR і просторове аудіо. Усі моделі сцени мають бути прив'язані якорями (клас AnchorEntity) [19]. Framework автоматично генерує та використовує мірмарс, які є серією прогресивно низьких варіантів текстури об'єктів, які покращують час візуалізації при застосуванні до віддалених об'єктів. RealityKit працює з полігональною сіткою, створеною за допомогою функції реконструкції сцени.

Висновок, RealityKit надає високоякісну технологію візуалізації та сучасні можливості AR «з коробки». Підтримує сканер LiDAR. Підтримує інструменти фотограмметрії. Відтворює поведінку Reality Composer через API сповіщень. RealityKit можна використовувати як самостійний фреймворк або як партнер ARKit і MetalKit. Починаючи з iOS 15, ми маємо доступ до фрагментних/піксельних шейдерів і модифікаторів геометрії за допомогою

сценаріїв Metal і CustomMaterials. Програмне забезпечення сімейства Reality має інструменти CLI та GUI для швидкого та легкого перетворення USDZ.

RealityKit працює з розкадровками UIKit або з інтерфейсами SwiftUI. Він має мінімум стандартного коду. Наприклад, RealityKit має дуже просту настройку для зіткнення моделей і для жестів (панорама, поворот, зведення), включаючи альтернативні 2D жести. І тут композиція над успадкуванням, тож це скоріше фреймворк програмування, орієнтованого на протокол – тісне зв'язок у більшості випадків більше не є проблемою у вашому коді. RealityKit ідеально вписується в реактивну парадигму Combine, яка допомагає обробляти видавців, передплатників та асинхронні події. Яскравим прикладом цього є загальний метод екземпляра `subscribe(to:on:_:)`, який повертає об'єкт, що представляє підписку на потік подій, наприклад `SceneEvents.Update.self`, який запускається один раз за інтервал кадру (60 кадрів в секунду).

SceneKit також є фреймворком високого рівня. Найстаріший у сімействі технологій візуалізації Apple. Він був випущений у 2012 році. SceneKit був задуманий для віртуальної реальності та може працювати на iOS / macOS. Для проектів AR ви можете використовувати його лише разом з ARKit. SceneKit підтримує Objective-C і Swift. У SceneKit основним блоком є вузол (клас `SCNNode`), який має власну ієрархію і може зберігати світло (`SCNLight`), або камеру (`SCNCamera`), або геометрію (`SCNGeometry`), або систему частинок (`SCNParticleSystem`), або аудіо програвачі (`SCNAudioPlayer`) (див. рис. 2.5). Основна перевага SceneKit – його можна легко налаштувати, він може змінювати геометрію та матеріали під час виконання, у нього є морфери, скіннери та обмеження, він відтворює сцену зі швидкістю до 120 кадрів в секунду та має розширені налаштування для системи частинок. Існують шейдери Blinn, Constant, Lambert, Phong, ShadowOnly і PBR.

Шейдер оклюзії також доступний для нас у SceneKit, але в спеціальному вигляді (тут немає готового матеріалу для оклюзії, який ми можемо знайти в RealityKit).

SceneKit читає файли у форматах .usdz, .dae та .scn. Підтримує вкладену анімацію активів, динаміку, частинки, матеріали PBR, HDR IBL і просторове аудіо. Для неявної та явної анімації перетворення будь-якого вузла можна використовувати класи SCNAction, SCNTransaction та CAAnimation. Хоча налаштування колізій у SceneKit дещо складніші. Щоб створити модульну і масштабовану архітектуру гри за допомогою SceneKit, нам потрібно реалізувати шаблон сутності-компонента GameplayKit [20].

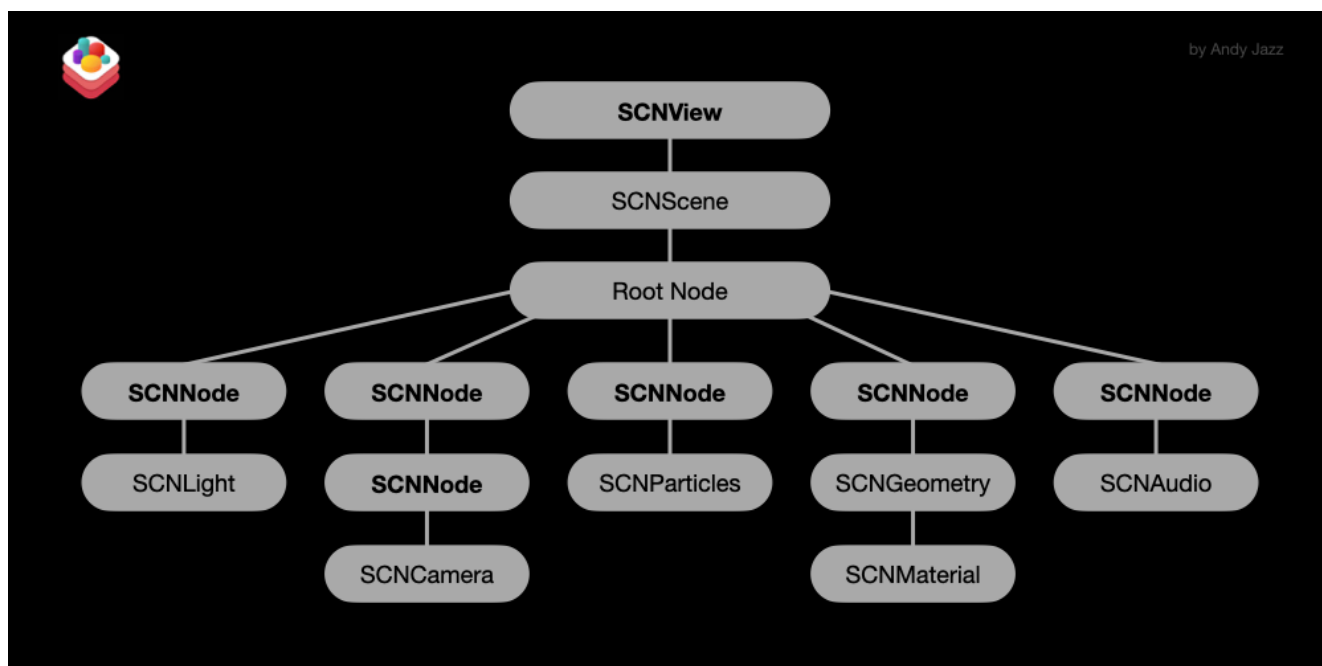


Рисунок 2.5 – Ієрархія SceneKit.

Висновок, SceneKit дає вам якісну технологію візуалізації (але спочатку вам потрібно налаштувати шейдери на основі фізичної бази), хоча для проектів AR ви можете використовувати її лише з ARKit. SceneKit легко налаштовується і може використовуватися з Swift і Objective-C, а також надає вам набір корисних методів екземпляра `renderer(...)`, що надходять від протоколу `ARSCNViewDelegate`, що дозволяє оновлювати моделі AR і відстежувати прив'язки зі швидкістю 60 кадрів в секунду. Працює з UIKit і SwiftUI (незважаючи на те, що в Xcode немає шаблону SceneKit+SwiftUI). Існують очевидні причини, що Apple може відмовитися від цієї платформи протягом наступних 3 років – SceneKit не оновлювався з 2017 року (за винятком незначних змін, таких як властивості

матеріалу clearCoat або SSR). Але SceneKit все ще має ряд переваг перед RealityKit 2.0. Розробники Swift забувають про те, що програми Objective-C SceneKit забезпечують швидкий час компіляції.

Наступна технологія – Metal. Якщо бути точним, Metal — це не технологія візуалізації, а скоріше прискорювач графічного процесора з можливістю використання мови розширеного затінення (MSL). Він був випущений у 2014 році. Це низькорівневий фреймворк. Metal реалізовано скрізь – у RealityKit, SceneKit, ARKit, CoreML, Vision, AVFoundation тощо [19]. Metal поєднує функції, подібні до OpenGL та OpenCL, під капотом лише одного API. Звичайно, Metal можна використовувати як засіб візуалізації для просунутої 3D-графіки. Метал виявляє не тільки відбиття, але й заломлення та явища розсіювання на поверхні.

Згідно з документацією Apple: «Metal — це мова програмування на основі C++, яку розробники можуть використовувати для написання коду, який виконується на графічному процесорі для графіки та паралельних обчислень загального призначення. Оскільки Metal заснований на C++, розробники знайдуть його знайомим і простий у використанні. З Metal, як графічні, так і обчислювальні програми можна писати на єдиній уніфікованій мові, що забезпечує більш тісну інтеграцію між ними» [21].

Висновок, розробники зазвичай використовують Metal framework для створення високоякісного графічного рендерингу для ігор зі складним 3D-середовищем, для програм обробки відео, таких як Final Cut Pro і Nuke, для 3D-додатків, таких як Maya, або для наукових додатків для великих даних, які мають працювати для наукових досліджень. дослідження. Зверніть увагу, що трасування променів у Metal набагато швидше, ніж у RealityKit.

3 ВИБІР МЕТОДУ НА ПОСТАНОВКА ЗАДАЧІ.

3.1 Вибір методу локалізація об'єктів.

Мета будь-якого алгоритму машинного навчання полягає в тому, щоб передбачити максимально близькі значення істинності. Для цього будь-який контрольований алгоритм машинного навчання використовує функцію втрат, а алгоритми навчаються, мінімізуючи втрати за допомогою оптимізації ваги або параметрів.

Оскільки локалізація об'єкта є проблемою регресії [22], можна використовувати будь-яку функцію втрат регресії, застосовну до N-вимірному масиву. Метод втрат L2 широко використовується практиками галузі та дослідницькими спільнотами.

Відстань L2, також відома як евклідова відстань. Це відстань між двома точками в N-вимірному просторі і обчислюється шляхом застосування теореми Піфагора щодо декартових координат точок. Цей підхід працює для будь-якого N-вимірному простору.

Щоб зрозуміти відстань L2, припустимо дві точки P і Q в тривимірному просторі. Точки P і Q представлені у вигляді (p_1, p_2, p_3) і (q_1, q_2, q_3) в декартових системах координат. Відстань між цими точками визначається як:

$$(P, Q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + (p_3 - q_3)^2}$$

В даному експерименті розглядається порівняльна характеристика алгоритмів оптимізації які використовуються у TensorFlow та CoreML.

Кращім алгоритмом можна вважати той що, може мінімізувати евклідову відстань між істиною та прогнозованим значенням.

У відкритих джерелах немає інформації який саме алгоритм застосований у цих технологіях. Отже, зробивши датасет з 100 зображень ми зможемо отримати

середню похибку. На рисунку 3.1 можна побачити, що CoreML має трохи меншу похибку, та локалізує kota краще, майже не торкаючись зайвих деталей на фото.

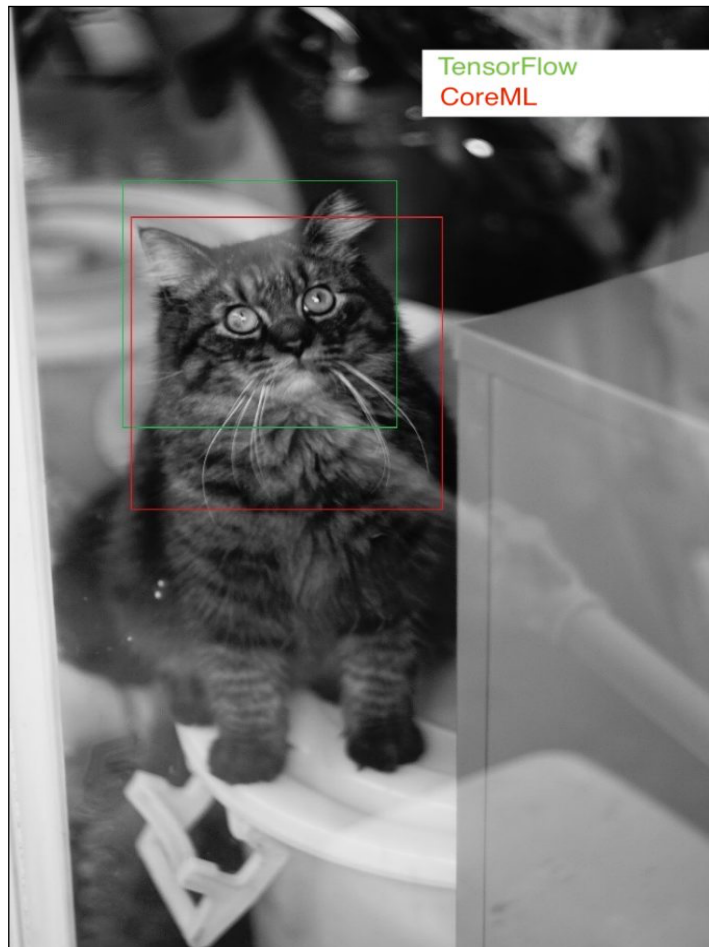


Рисунок 3.1 – Приклад порівняння алгоритмів локалізації

На цьому прикладі можна роздивитися що, що технологія CoreML має більш точний та акуратний результат, а саме обмежувальну рамку, яка не зачіпає інші частини kota чи навколишнього середовища (див. рис. 3.2).

Для повноцінної картини було проаналізовано середню похибку між істиною обмежувальною рамкою та результатами роботи цих технологій, та отримана такі результати.

Похибка результату вимірювання, властива умовам статичної вимірності, тобто при вимірюванні постійних величин після завершення перехідних процесів в елементах приладів та перетворювачів.

Статична похибка використана при його вимірі за допомогою постійної величини.

	A	B	C
1	Файл	TensorFlow	CoreML
91	13fewr5243.jpg	3.63	1.38
92	85sot81pos.jpg	2.7	1.82
93	3falbgh5re1.jpg	5.21	3.65
94	48askdvvfse.jpg	3.47	2.21
95	59qpjfse3e4.jpg	1.91	1.78
96	t09bvpaxtw2.jpg	6.67	1.84
97	ldxlv13d0a0.jpg	3.42	0.87
98	laqot3225g8.jpg	4.07	2.54
99	f23f33asd53.jpg	5.83	1.09
100	lls241242dh.jpg	2.2	2.74
101	b83ms92ps8.jpg	4.12	3.13
102	Похибка	3.93	2.095454545

Рисунок 3.2 – Результати експерименту

Отже, можна зробити висновок, що технологія CoreML має значну перевагу, а саме має похибку майже у два рази менше ніж TensorFlow. Тому, CoreML була обрана для вирішення питань локалізації об'єктів.

3.2 Вибір методу реалізації доповненої реальності

Було обрано дві технології які надають змогу побудувати доповнену реальність на мобільному пристрої.

По-перше, Unity — це кросплатформний ігровий двигун, підтримує різноманітні платформи ПК, мобільні, консольні і віртуальні платформ. Він особливо популярний для розробки мобільних ігор для iOS. Він вважається простим у використанні для початківців розробників і популярним для розробки

інді-ігор. Двигун можна використовувати для створення тривимірних (3D) і двовимірних (2D) ігор, а також інтерактивних симуляцій та інших можливостей.

По-друге, SceneKit — це інтерфейс програмування 3D-графіки для платформ Apple Inc., написаний на Objective-C. Це фреймворк високого рівня, розроблений для забезпечення простого у використанні рівня над API нижнього рівня, як-от OpenGL і Metal [23]. SceneKit підтримує об'єктний графік сцени разом із фізичним механізмом, системою частинок та посиланнями на Core Animation та інші фреймворки, щоб легко анімувати цей дисплей. Подання SceneKit можна змішувати з іншими видами, наприклад, дозволяючи відображення 2D-дисплею SpriteKit на поверхню об'єкта в SceneKit, або використовувати його для рендеру об'єктів в доповненій реальності за допомогою так званою надбудови ARKit.

Дослідження продуктивності показало (див. рисунок 3.3), що різниці в FPS за однакових умов для обох платформ майже немає.

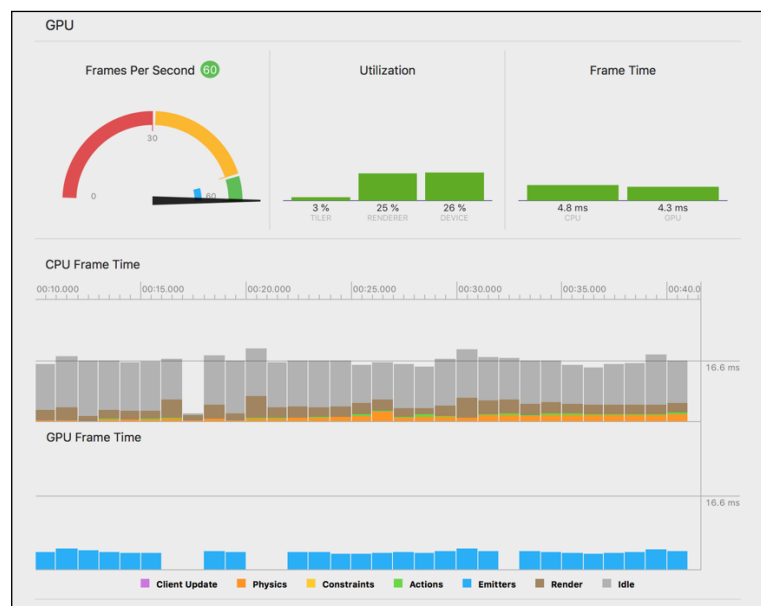


Рисунок 3.3 – Дослідження FPS.

Іншим показником продуктивності є швидкість завантаження додатку, в цьому тесті безпрецедентний переможець ARKit, 12 секунд проти 4 секунд. Основною причиною такої різниці є факт того, що Unity – це статична бібліотека,

яка завантажується при запуску додатку. В той час як ARKit динамічний фреймворк, який дуже тісно інтегрований у операційну систему iOS.

Варто зазначити, що вага додатку також суттєво відрізняється, проект з Unity важить на 120 МБ більше за конкурента.

Також, інтеграція ARKit з CoreML виглядає дещо простішою порівнюючи з Unity. Оскільки для того, щоб передати потік відео з камери, який займає Unity, треба розробити прошарок використовуючи Objective-C Bridging Header, це є певним бар'єром як для розробників так і життєздатності програми, тому що, зміна API той чи іншої технології потребує також змін у Bridging Header. Останній, та мабуть головний недолік, це перетворення координат локалізації об'єкту між CoreML та Unity, розрахунки виглядають дещо об'ємнішими за подібну реалізацію з ARKit, та головне, що вони втрачають точність при зміні положення мобільного пристрою.

Взявши до уваги недоліки та складнощі інтеграції CoreML з Unity було обрано ARKit в якості засобу реалізації доповненої реальності. Треба зазначити, що Unity має надлишковий функціонал та краще підходить перш за все для мобільних ігор [24].

3.3 Основні функції мобільного додатку

Мобільна аплікація повинна задовольняти базові вимоги мандрівника, а також надавати змогу побачити анотації в доповненій реальності при настанні певних умов.

На прикладі додатку «путівник по Харкову» для операційної системи iOS реалізований наступний функціонал:

- отримання загальної інформації про місто;
- карта з відмітками та прокладеним маршрутом;

- детальна інформація по кожній відмітці (історична пам'ятка, споруда, тощо.)
- перехід до режиму AR.

Для переходу режим AR користувач повинен бути в межах відміченої на карті, натиснути спеціальну кнопку «AR» на екрані смартфона та надати доступ для використання камери, після чого навести свій девайс на об'єкт, в результаті користувач побачить 3D модель або анімацію, яка доповнює реальність.

Мобільний додаток складається з декількох головних екранів та спливаючих вікон. На (рис. 3.4) можна побачити зліва головний екран з двома кнопками «AR» та «Мапа», остання відкриває перегляд маршруту.

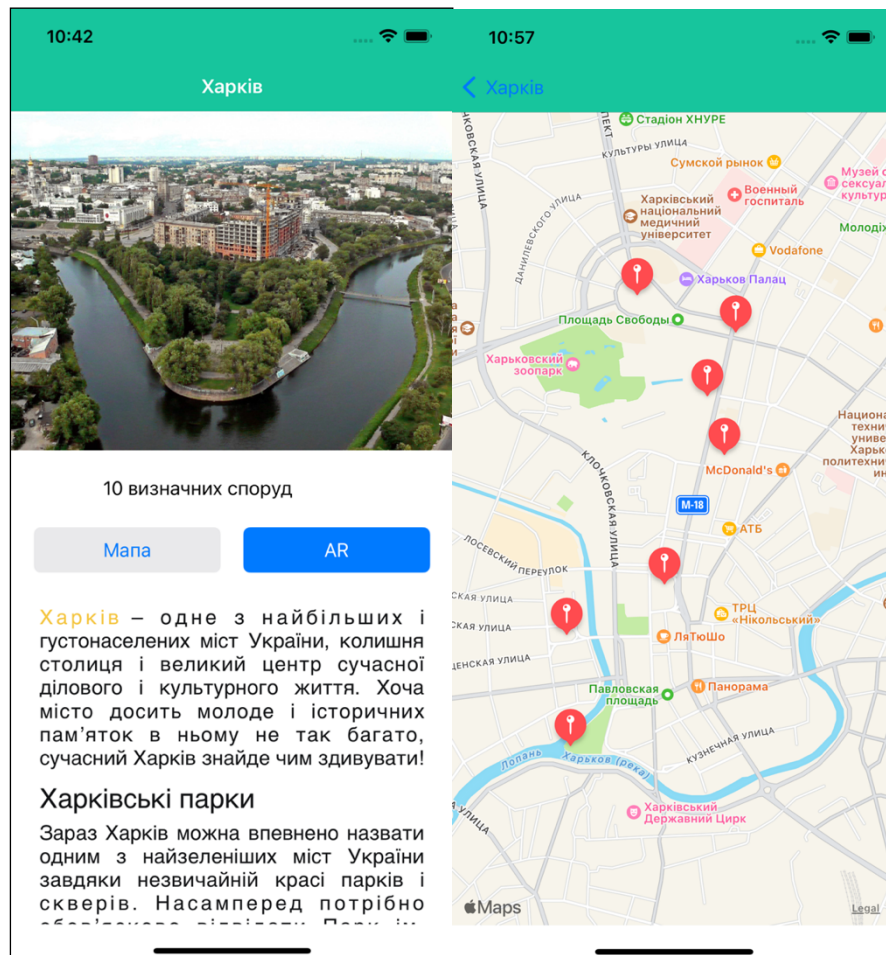


Рисунок 3.4 – Головні екрани додатку

Точки на мапі мають інформацію про позначення місце та дозволяють прокласти маршрут.

На екрані «AR» продемонстрована (рис 3.5) дослідницька частина роботи, а саме анотація об'єктів у доповненій реальності.

Для отримання анотацій користувач повинен навести камеру смартфона на об'єкт після чого він побачить анімацію чи анотацію.



Рисунок 3.5 – Екран «AR»

Анотації відображаються у вигляді трьох вимірного тексту, який додається у визначене датасетом місце.

3.4 Проектування архітектури програмного забезпечення

У якості архітектури мобільного додатка було обрано MVVM (див. рис. 3.6). MVVM полегшує відокремлення розробки графічного інтерфейсу від розробки бізнес логіки відомої як модель (можна також сказати, що це

відокремлення представлення від моделі). Модель представлення є частиною, яка відповідає за перетворення даних для їх подальшої підтримки і використання.

Окрім цього, не аби якою перевагою обраного шаблону є його популярність, тому поріг входу до проекту для інженерів буде мінімальним. Архітектура реалізована за допомогою мови програмування Swift.

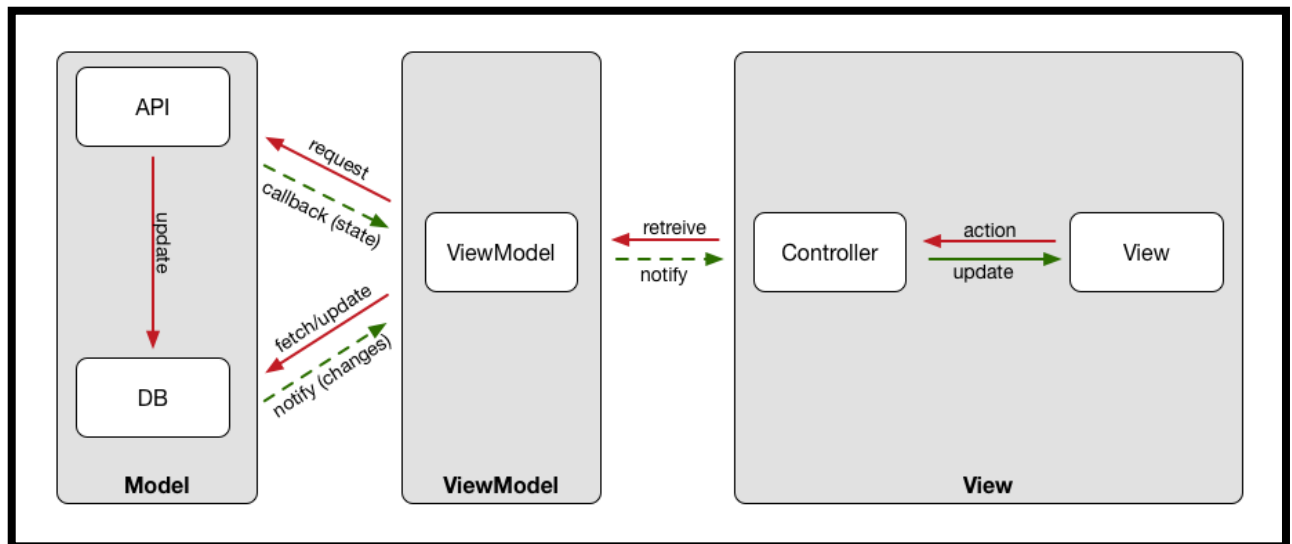


Рисунок 3.6 – UML діаграма шаблону MVVM.

Модель представлення більше схожа на звичайну модель, ніж на представлення і оброблює більшість, якщо не всю, логіку відображення даних. Модель представлення може також реалізовувати патерн медіатор, організовуючи доступ до будь-яких сервісів логіки навколо множини правил використання, які підтримуються представленням.

3.5 Інтеграція CoreML та ARKit.

Програма, що розробляється, має дві основні сутності. Перша з них — це служба виявлення об'єктів: вона приймає зображення як вхідні дані та повертає

обмежувальну рамку та мітку класу розпізнаного об'єкта. Другий — ViewController, місце, де відбувається вся магія AR.

Сервіс розпізнавання та локалізації об'єктів має 3 основні дії:

– ініціалізація моделі Core ML: у цьому випадку ми працюємо з представленням Swifty попередньо навченої моделі машинного навчання;

– ініціалізація запиту на аналіз зображення, який використовує модель Core ML для обробки вхідних зображень;

– метод «Detection», що створює екземпляр обробника для виконання запитів Vision для одного зображення. Він використовує піксельний буфер Core Video, оскільки його можна легко взяти з поточного ARFrame, буфер пікселів не зберігає інформацію про поточну орієнтацію зображення, тому ми беремо поточну орієнтацію пристрою та відображаємо її у форматі орієнтації Exif [25].

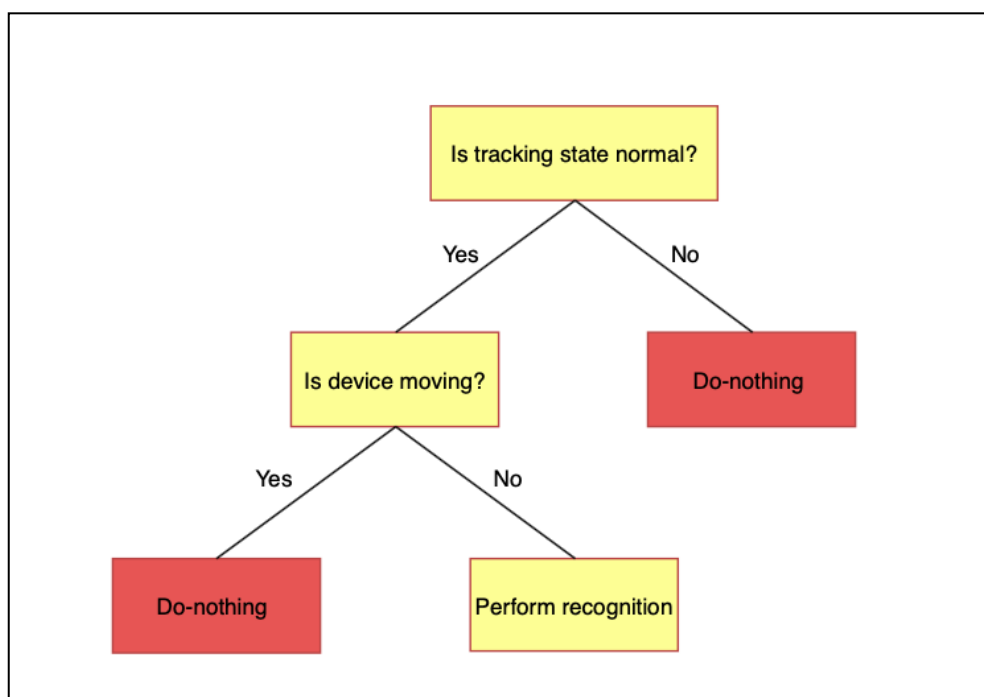


Рисунок 3.7 – Логіка оцінки стабільності сцени ARKit

Коли точність розпізнавання більше 90% запускається алгоритм анотації. ViewController відповідає за циклічні виклики служби виявлення об'єктів і розміщення анотацій, коли об'єкт розпізнається. Крім того, контролер підвищує

точність розміщення анотацій — він не дозволяє розмістити анотацію, якщо пристрій рухається, об'єкт знаходиться занадто далеко від камери тощо.

Щоб зробити автоматичну анотацію, нам потрібно постійно зациклювати процес отримання знімка поточного кадру та відправлення його до служби виявлення. Але процес розпізнавання запускається тільки коли сцена є стабільною, що визначається просто логікою.

Контролер може відстежувати всі зміни в поточному сеансі AR. Існує багато випадків, коли в певному стані нам не потрібно запускати виявлення об'єктів, тому ми встановлюємо для `isDetectionEnabled` значення `false` (наприклад, коли відстеження AR обмежено, відбувається ініціалізація або поверхня не виявлена). Цей контролер подання реалізує всі методи з `ARSessionDelegate`, і ці методи викликають `onSessionUpdate`, який визначає стан сеансу та розглядає, чи слід продовжувати виявлення. Коли пристрій рухається занадто швидко, це негативно впливає на точність анотації. Реалізована логіка, яка відстежує рух пристрою та встановлює `isDetectionEnabled` в `false` щоразу, коли користувач перевищить обмеження швидкості.

На першому етапі анотації об'єктів виключається ситуацію, коли для одного виявленого об'єкта розміщується більше однієї анотації. Створена центральна точка розпізнаного об'єкта на екрані за допомогою системи координат `sceneView`. Метод `hitTest` шукає в сцені рендерера об'єкти, що відповідають точці, і використовує `SCNHitTestSearchMode.all`, щоб знайти всі перетини посланого променя з вузлами. Якщо в отриманому результаті він знаходить `BubbleNode` (фактична назва анотації, відфільтрована за назвою вузла), виконання методу закінчується.

Наступний крок — знайти площину та точку у світовій системі координат, де слід розташувати вузол. Це робиться за допомогою методів `raycastQuery` та `raycast(raycastQuery)`.

Останній етап це — розміщення анотації на кореновому вузлі сцени. Встановлюється `worldPosition` нещодавно створеного `BubbleNode` в позицію, визначену на другому кроці.

ВИСНОВКИ

В ході виконання кваліфікаційної роботи магістра була досліджена предметна область інтеграції машинного навчання для анотації та доповненої реальності. Була проаналізована необхідність інтеграції систем. Було досліджено окремі засоби машинного навчання для реалізації розпізнавання об'єктів на мобільному пристрої, порівняно їх ефективність та продуктивність. Також було досліджено засоби реалізації доповненої реальності та проаналізована продуктивність з точки зору технологія та людського ресурсу.

Головним чином було розглянуто та порівняно існуючі методи інтеграції доповненої реальності та анотації об'єктів. Також, було розглянуті способи використання моделей машинного навчання та засоби створення доповненої реальності на мобільній платформі iOS у режимі реального часу.

На основі зібраної інформації було обрано технологію CoreML, яка дозволяє використовувати аналіз потокового відео прямо на смартфоні. Було розглянуто та порівняно різні методи взаємодії цих технологій. В результаті було обрано ARKit для створення доповненої реальності, оскільки інші засоби не показали свою ефективність. Було проаналізовано та реалізовано способи підключення та взаємодії обраних систем.

На основі дослідницької роботи було розроблено мобільний додаток, путівник по місту Харкову, в якому реалізовано взаємодія користувача з історичними пам'ятками та іншими спорудами у вигляді анотації об'єктів у доповненій реальності.

Отже, можна зробити висновок, що проведені дослідження та інтеграція є успішною та є предметом технічного ноу-хау, оскільки вся обробка відео потоку зроблена на смартфоні у режимі реального часу.

В подальшому дану програмну систему можна розвивати та додавати нові функції, наприклад, складні анімації. Також, результати досліджень можна

застосовувати у багатьох сферах, де передбачується такий вид взаємодії користувача та мобільного додатку.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Новосний портал «RBC Україна». URL: <https://travel.rbc.ua/ukr/show/vosstanovlenie-stranstviy-kotorym-vydalsya-1640687852.html>
2. Журнал Time. URL: <https://time.com/6116826/what-is-the-metaverse/>
3. OprahDaily. URL: <https://www.oprahdaily.com/life/g25563106/best-travel-apps>
4. 101 Blockchain. URL: <https://101blockchains.com/pros-cons-of-metaverse/>
5. Telekom. URL: <https://www.telekom.com/en/company/details/this-is-augmented-reality-614136>
6. Geron A. Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. USA: O'Reilly Media, 2019. 956 с.
7. Mathworks. URL: <https://www.mathworks.com/discovery/object-detection.html>
8. Hramm, O., Bilous, N., Ahebian, I. Configurable Cell Segmentation Solution Using Hough Circles Transform and Watershed Algorithm. Proceedings of the International Conference on Advanced Optoelectronics and Lasers, CAOL, 2019, 2019-September, стр. 602–605, 9019493.
9. Himanshu Singh. Practical Machine Learning and Image Processing: For Facial Recognition, Object Detection, and Pattern Recognition Using Python. USA: Apress, 2019. 184 с
10. Program Ace. URL: <https://program-ace.com/blog/augmented-reality-sdk/>
11. Digital information world. URL: <https://www.digitalinformationworld.com/2019/04/snapchat-landmarkers-ar-lens-studio.html>

12. Білоус Н. В., Черненко Є. А. Проблеми та рішення локалізації об'єктів. XII Міжнародна науково-практична конференція “MODERN DIRECTIONS OF SCIENTIFIC RESEARCH DEVELOPMENT”, 18-20 травня 2022 року Чикаго, США, стр 1-3.
13. Toward Data Science. URL: <https://towardsdatascience.com/faster-r-cnn-for-object-detection-a-technical-summary-474c5b857b46>
14. Vaibhav V. Computer Vision Using Deep Learning: Neural Network. USA: Apress, 2021. 332 с.
15. Apple Documentation. URL: <https://developer.apple.com/documentation/realitykit>
16. Medium: Mac O`Clock. URL: <https://medium.com/@arkit/realitykit-911-entity-component-system-ecs-bfe0520e0e8e>
17. Medium: Geek Culture. URL: <https://medium.com/@arkit/realitykit-911-fake-shadows-for-stationary-models-78387aabf42c>
18. 3D Graphics Rendering Cookbook: A comprehensive guide to exploring rendering algorithms in modern OpenGL and Vulkan. USA: Packt Publishing, 2021, 670 с.
19. MetalkitOrg. URL: <http://metalkit.org/2018/07/14/raytracing-with-metal.html>
20. Apple Developer. URL <https://developer.apple.com/documentation/modelio>
21. Metal Programming Guide: Tutorial and Reference via Swift. USA: Addison-Wesley Professional, 2015. 352 с.
22. Shcherbakova, G., Krylov, V., Logvinov, O., Bilous, N. Adjustment of wavelet function parameters for analysis of non-stationary periodic signals with multistart optimization. 2017 4th International Scientific-Practical Conference Problems of Infocommunications Science and Technology, PIC S and T 2017 – Proceedings, 2017, 2018-January, стр. 110–112.
23. Claudia D., Timothy H. Augmented Reality and Virtual Reality: Empowering Human, Place and Business. USA: Springer, 2018. 395 с.

24. Apple Developer. URL:
<https://developer.apple.com/documentation/scenekit>
25. Jayven N. Mastering ARKit: Apple's Augmented Reality App. USA:
Apress, 2022. 568 c.