Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра мультимедійних систем факультету інформатики

Особливості використання машинного навчання та доповненої реальності на пристроях на базі Android

Текстова частина до курсової роботи
за спеціальністю «Комп'ютерні науки та інформаційні технології» - 122

Керівник курсової роботи
старший викладач
Борозенний С.О.
_____
(Підпис)
"___" _____ 2020 року
Виконала студентка КНІТ-4
Радзієвська О.В.
"___" _____ 2020 року

Київ 2020

Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

Кафедра мультимедійних систем факультету інформатики

ЗАТВЕРДЖУЮ
Викладач кафедри мультимедійних систем,
доцент _____ Жежерун О.П.
(підпис)
„____”_____2019 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ
на курсову роботу
студенту Радзієвській Олександрі Володимирівні
факультету інформатики 4 курсу бакалаврської програми
ТЕМА: **Особливості використання машинного навчання та доповненої реальності на пристроях на базі Android**

Зміст ТЧ до курсової роботи:
- Індивідуальне завдання
- Вступ
- Огляд теоретичного матеріалу і здійснення дослідження
- Висновки

Список літератури
Додатки (за необхідністю)
Список посилань

Дата видачі „___” _____ 2019 р. Керівник_____
(підпис)
Завдання отримав _____

(підпис)

Тема: **Особливості використання машинного навчання та доповненої реальності на пристроях на базі Android**

№ п/п	Назва етапу курсової роботи	Термін виконання етапу	Примітка
1.	Отримання завдання на курсову роботу	23.12.2019	
2.	Огляд літератури за темою роботи	12.01.2020	
3.	Проведення досліджень	28.01.2020	
3.	Опис результатів дослідження	10.03.2020	
4.	Аналіз отриманих результатів з керівником	08.03.2020	
6.	Корегування роботи	28.03.2020	
7.	Створення презентації та написання доповіді	30.03.2020	
8.	Остаточне оформлення пояснювальної роботи та слайдів.	02.04.2020	
9.	Захист курсової роботи	Квітень 2020	

Студент _____

Керівник _____

“_____”_____

ЗМІСТ

Анотація

У роботі описані технології машинного навчання і доповненої реальності та їх перспективи, зокрема при використанні на мобільних пристроях. Розглянуто інструменти ML Kit та ARCore від Google, досліджено їхні переваги та недоліки. На основі цих інструментів було створено фреймворк ARMarker, який може бути використаний іншими розробниками для легкої інтеграції даних технологій у свої додатки.

ВСТУП

Ідея машинного навчання з’явилася ще у 1952 році, та брак потужностей значний час стримував його розвиток. Бурхливий розвиток машинного навчання розпочався у 1990-х роках, і з того часу ця технологія зробила значний крок вперед, освоївши величезний спектр задач. Сьогодні машинне навчання використовується практично у всіх сферах життя: медицині, страхуванні, економіці, математиці, мовознавстві, рекламі, пошукових системах, створенні ігор, дослідженні генів тощо та має значний потенціал.

На даний момент технології досягли такого рівня, що складні обчислення можна здійснювати навіть на телефонах, що відкриває нові можливості з використання машинного навчання у повсякденному житті. Раніше можна було проводити навчання лише пересилаючи дані на хмарний сервері, але такий підхід створює ряд проблем і незручностей: затримка у роботі системи, питання безпеки і приватності, висока вартість ресурсів, необхідних для централізованої обробки даних, обов’язковість швидкого доступу до інтернету. Машинне навчання безпосередньо на пристроях покращить досвід користування додатками та дасть змогу розширити застосування для відстежування стану здоров’я, розпізнавання мовлення, сервіси з рекомендацій, пропозицій реклами тощо.

Сьогодні для розробки додатків з використанням машинного навчання під iOS та Android використовується платформа ML Kit від компанії Google та бібліотеку TensorFlow, крім того Apple розробляє свою платформу Core ML. ML Kit працює на Android починаючи з API рівня 21.

Доповнена реальність ще одна популярна технологія, що має величезний потенціал не лише в сфері розваг, а й в освіті, медицині, промисловості, авіації, маркетингові, дизайні, туризмі та інших сферах життях. Термін “доповнена

реальність” був запропонований ще у 1990 році та активно заговорили про нього у 2013, коли були представлені Google Glass. Сьогодні Apple і Google активно ведуть розробку своїх інструментів ARKit та ARCore, що дозволяє легко створювати додатки з використанням віртуальної реальності. За прогнозами [9] до 2025 року вартість доповненої і віртуальної реальності у медицині сягне 5 млрд $ та 4 млрд $ у сфері туризму. Доповнена реальність може внести якісні зміни у підходах до освіти, покращити досвід покупок тощо. AR підтримується на мобільних пристроях починаючи з Android 7.0 (Nougat) та iOS 11 - тобто на нових пристроях, досить високої цінової категорії. Але поступово ця ситуація змінюється, і скоро така технологія буде доступною для широкого кола користувачів.

Підсумовуючи написане вище, можна зробити висновки, що машинне навчання та доповнена реальність мають величезний потенціал, а можливість використання цих технологій на мобільних пристроях може якісно покращить життя у різних сферах. Тим не менш, цей напрям досить новий, технології проходять стадію активної розробки і ще не всі розробники готові працювати з ними.

Метою роботи є дослідження інструментів для роботи з цими технологіями, а також створення фреймворку для спрощення використання цих технологій у своїх додатках.

Текстова частина курсової роботи складається з трьох розділів.

У першому розділі описуються нейронні мережі, як один з типів машинного навчання, розглядаються особливості машинного навчання на мобільних пристроях, аналізується інструментарій ML Kit, його можливості та недоліки.

Другий розділ присвячений доповненій реальності. У ньому досліджується принцип роботи та перспективи AR, а також аналізується фреймворк ARCore.

Третій розділ описує розроблений фреймворк: його можливості, структуру та перспективи для вдосконалення.

Постановка задачі:

1) дослідити машинне навчання та доповнену реальності в розрізі їх використання на мобільних пристроях на платформі Android;

2) проаналізувати інструментарій для роботи з машинним навчанням ML Kit та доповненою реальністю ARCore, дослідити їх можливості у рамках поставлених задач;

3) створити фреймворк, який матиме змогу розпізнавати об’єкти та QR-коди за допомогою ML Kit та створювати мітки в доповненій реальності.

РОЗДІЛ 1. МАШИННЕ НАВЧАННЯ НА БАЗІ ML KIT

1.1 Машинне навчання

Машинне навчання (machine learning) – це підрозділ штучного інтелекту, який використовує статистичні методи для надання комп'ютеру здатності покращувати свої результати у вирішенні певної задачі, без того, щоб бути програмованим явно.

Важливим напрямком машинного навчання є нейронні мережі. Штучна нейронна мережа – це багатошарова мережа штучних нейронів, яка імітує роботу нейронів у мозку і використовується для класифікації, створення передбачень тощо. Такі мережі мають здатність навчатися і виконувати задачі без спеціального написання алгоритму під кожну конкретну задачу.

Нейронні мережі використовуються для завдань, які вимагають аналітичних обрахунків, подібних до тих, що здатна виконувати людина. Найпопулярнішими задачами для нейронних мереж є класифікація та кластеризація.

Класифікація має на меті віднести кожен об'єкт до одного зі скінченного списку класів ґрунтуючись на його властивостях. Класифікація належить до навчання з вчителем, тобто наперед задана певна множина об'єктів, помічених класом, до якого вони належить, і навчання відбувається на основі цього масиву даних. Класифікація може бути використана для позначення листа як спаму, визначення, чи поверне людина кредит, розпізнавання дорожнього знаку, прогнозування погоди тощо.

Кластеризація – розбиття множини об'єктів на групи таким чином, що об'єкти одного класу були більш схожі між собою ніж на об'єкти інших кластерів по якому-небуть критерію. Задача кластеризації відноситься до

навчання без вчителя і логіка віднесення певного об’єкту до певного кластеру може бути неочевидна для людини. Кластеризація може бути використана для створення рекомендацій, виявлення аномалій, групуванню новин за змістом тощо.

Для подальшої роботи з машинним навчанням нам потрібно розуміти, як працюють нейронні мережі. Основними компонентами нейронної мережі є нейрони, що з’єднані зв’язками і утворюють шари. Кожен нейрон має ваговий коефіцієнт, який показує наскільки важливим є даний нейрон у прийнятті рішення.

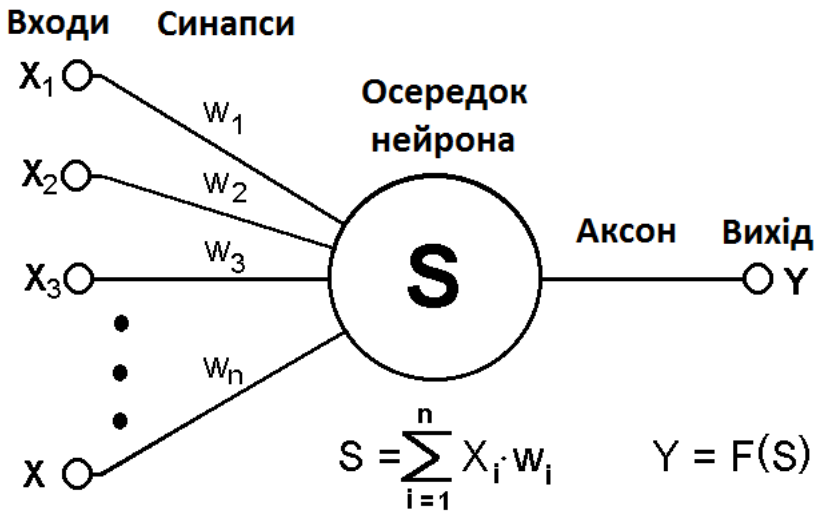

Рис. 1.1. Схема роботи формального нейрона

На вхід штучного нейрону надходять сигнали x_1, x_2 … x_n (вектор X), які є виходами від інших нейронів. Кожен з сигналів домножується на відповідну вагу w_1, w_2 … w_n. Результати сумуються в суматорі (позначений як S на рис.1.1) і подаються на вихід [3].

На практиці найчастіше використовують багатошарові нейронні мережі (рис. 1.2.). Перед навчанням нейронної мережі вхідні дані рекомендується

нормалізувати, тобто звести до діапазону від -1 до 1. Ваги W потрібно ініціалізувати випадковим чином, адже так досягається найкращий ефект від навчання. Обрахунок відбувається за схемою, описаною вище. До результату застосовується функція активації. Такий алгоритм називається прямим проходом (forward propagation).

Для оцінки якості тренованої мережі використовується зворотний прохід (back propagation). Спочатку результат роботи нейронної мережі порівнюється з результатами на тестовій вибірці і знаходиться різниця значень. Вираховується значення на яке потрібно оновити вагові коефіцієнти і відбувається послідовне оновлення всіх ваг. Технічно навчання й полягає в такому знаходженні коефіцієнтів зв'язків між нейронами.

Цикл прямого проходу називається епохою. З кожним циклом навчання похибка обчислень зменшується. Коли вона перестає перевищувати певне значення, вважається що мережа пройшла навчання і готова вирішувати реальні задачі.

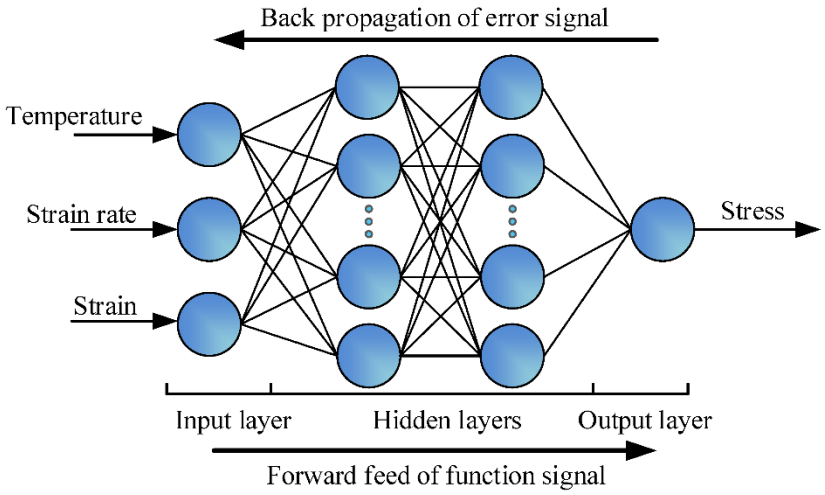

Рис. 1.2. Чотиришарова нейронна мережа

На практиці навчання нейронної мережі зводиться до множення матриць великих розмірностей, що диктує вимоги до потужностей пристрою, на якому ці обчислення проводяться.

1.2 Особливості машинного навчання на мобільних пристроях

Машинне навчання має величезні перспективи у вирішенні широкого ряду задач у повсякденному життя. Так як машинне навчання потребує значних ресурсів, то певний час його можливості були обмеженні для використання пересічним користувачем, але активний розвиток мобільних пристроїв зробив реальним використання цієї технології на телефонах навіть без стабільного підключення до Інтернету. Це має низку переваг у порівнянні з навчанням в хмарі.

Першим важливим аспектом є швидкість роботи застосунку. Обмін інформацією між пристроєм і хмарою може займати значний час. Швидкість роботи є дуже важливим критерієм сучасного застосунку і тому затримка більше кількох секунд може бути критичним фактором і сприяти негативному досвіду користувача.

Мобільні стосунки більшість часу доступаються до Інтернету через мобільну мережу, і користувачі зацікавлені в мінімізації її використання, оскільки це дорого, а також знижує енергоефективність. Крім того, на даний момент якісна мобільна мережа є далеко не всюди (зокрема в тунелях метро, деяких селах і на трасах між містами). Можливість роботи офлайн дасть поштовх для розвитку застосунків, що не могли бути втілені раніше в повному обсязі через необхідність стабільного підключення до Інтернету, наприклад застосунки для відстеження стану здоров’я.

Енергоефективність зокрема також відіграє важливу роль у формуванні досвіду користувача, адже важлива риса мобільних пристроїв це автономність від точок живлення. Крім того, низька енергоефективність створює проблему у довготерміновій перспективі, адже від великої кількості циклів розрядів і зарядів батарея зношується.

Надзвичайно важливим аспектом в сучасному світі є конфіденційність і безпека. Оскільки вся робота відбувається безпосередньо на пристрої, користувач уникає всіх небезпек, що пов’язані з передачею даних, зокрема їх перехоплення кіберзлочинцями. Гарантія безпеки і приватності особистих даних користувача є вимогою GDPR (General Data Protection Regulation - Загальний регламент захисту даних). Прикладом вдалого використання машинного навчання на пристрої в розрізі безпеки та приватності може слугувати розпізнавання обличчя за допомогою нейронної мережі безпосередньо на пристрої.

Також машинне навчання на пристрої в порівнянні з обробкою в хмарі має переваги і для розробників. Уникаючи значної централізованої обробки даних виробник застосунку може значно зекономити на серверах. Також економиться час на налаштування і підтримку серверів. До того ж знижуються вимоги до ширини каналу передачі даних.

Отже, навчання на пристрої значно економить ресурси, гроші та час як для користувачів так і для розробників. Гіганти Google і Apple активно ведуть розробку в цьому напрямі, створюючи інструментарій для зручної розробки застосунків з машинним навчанням, додаючи все нові і нові функції.

1.3 Особливості ML Kit

ML Kit for Firebase був представлений на Google I/O 2018. ML Kit – це мультиплатформений мобільний SDK від Google, що надає можливість розробникам легко інтегрувати високоточні наперед навчені глибинні моделі машинного навчання в мобільні застосунки на Android та iOS. Поки існує лише beta-версія ML Kit. Підтримка ML Kit Android починається з версії 5.0 (Lollipop). На презентації у 2018 було представлено 5 функцій, сьогодні ж ця кількість збільшилась до 11:

- розпізнавання тексту;
- розпізнавання облич;
- сканування штрих-кодів;
- маркування зображень;
- виявлення та відстеження об’єктів;
- розпізнавання пам’яток;
- розпізнавання мови тексту;
- переклад;
- генерація «розумних» відповідей;
- створення власних моделей з AutoML;
- задання власних моделей [2].

Перевагою ML Kit є легкість використання: для вирішення більшості завдань досить написати декілька стрічок коду. Для початку роботи з ML Kit не потрібно мати глибинних знань з нейронних мереж та оптимізації моделей. Тим не менш, на випадок, якщо існуючого функціоналу недостатньо для вирішення задачі, ML Kit надає зручну API, яка дозволяє використовувати власні TensorFlow Lite моделі (рис. 1.3.) в мобільних додатках. Якщо модель

занадто складна розробник може запустити її тренування в хмарі та організувати динамічне завантаження в додаток. Це дає можливість зменшити розмір додатку і при цьому оновлювати модель окремо від оновлення застосунку.

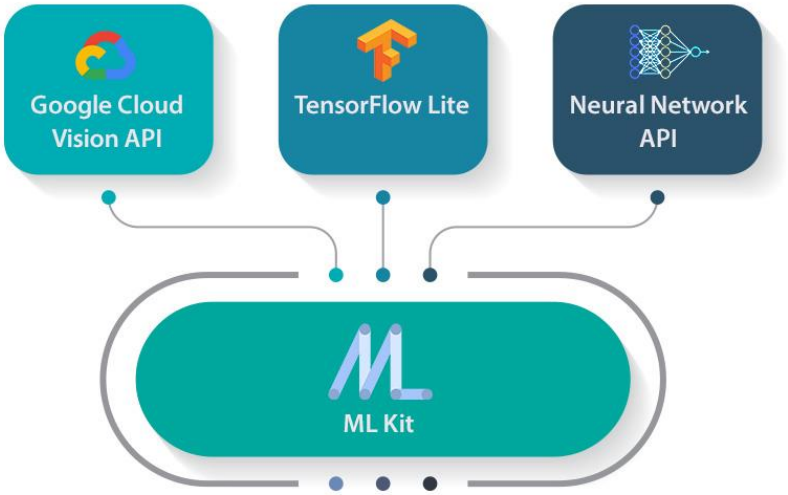

Рис. 1.3. Компоненти ML Kit

ML Kit забезпечує два режими роботи: на пристрої та в хмарі. Розробники можуть обирати режим залежно від їх потреб. Тренування моделей на пристрої швидше та не потребує стабільного підключення до інтернету, але хмарні моделі мають більше можливостей і дають вищу точність.

Також ці два режими мають ще деякі особливості. Розпізнавання пам’яток працює лише в хмарі, а розпізнавання мов та маркування зображень мають більше можливостей ніж при тренуванні на пристрої (більша кількість мов, та об’єктів, які можна розпізнати). На пристрої API пропонує роботу з функціоналом для розпізнавання облич, мови, виявлення та відстеження об’єктів, скануванням QR-кодів, «розумними» віповідями, AutoML та власними кастомними моделями.

Для використання на пристрої інструментарій повністю безкоштовний. Також Google безкоштовно надає функціонал в хмарі для перших 1000 звернень на місяць, кожні наступні 1000 звернень коштують 1,5$.

Як згадувалось, важливим аспектом при роботі з хмарними рішеннями є приватність. Тому Google гарантує, що дані користувача видаляються з хмарного API відразу після роботи з ними.

На жаль, ML Kit також містить ряд недоліків. Зокрема розмір власних моделей часто може перевищувати розмір застосунку в цілому, що створює незручності як для розробника так і для користувача. Також, як було зазначено вище, хоча SDK було презентовано два роки тому, досі наявна лише beta-версія, а отже вона може містити певні недоліки і помилки, що є нормою для beta-версії [4] .

Висновки до розділу 1

У цьому розділі було розглянуто принцип роботи нейронні мережі, як одного з видів машинного навчання. Проаналізовано особливості навчання на пристрої та розглянуті його переваги у порівнянні з навчанням в хмарі. Також було обговорено можливості та недоліки інструментарію ML Kit.

РОЗДІЛ 2. ДОПОВНЕНА РЕАЛЬНІСТЬ НА БАЗІ AR CORE

2.1 Доповнена реальність

Доповнена реальність (Augmented Reality) – це доповнення фізичного світу будь-якими віртуальними даними (візуальними, аудіо тощо) у режимі реального часу.

Загалом термін «доповнена реальність» не новий – він був запропонований дослідником Томом Коделом ще у 1990 році. З того часу доповнена реальність здебільшого використовувалась у наукових лабораторіях та воєнних цілях. У 1977 дослідник Рональд Азума визначив доповнену реальність як систему, яка:

- Поєднує віртуальне і реальне;
- Взаємодіє в реальному часі;
- Працює в 3D [6].

Активно ж говорити про доповнену реальність почали у 2013 році, коли Google представив свої Google Glass. Хоча вони насправді не мали особливого відношення до цієї технології, їх поява дала поштовх до розвитку цього напрямку.

Важливо розрізняти доповнену реальність з віртуальною. Віртуальна реальність (Virtual Reality) повністю будує штучний світ, який передається людині через органи чуття: зір, слух, дотик тощо, та імітує вплив людини на цей світ, та реакцію світу на такий вплив. Водночас, доповнена реальність лише додає окремі штучні елементи в сприйняття реального світу.

Віртуальна і доповнена реальність мають різний принцип робота, а тому і різні способи використання. Але варто зауважити, що досвід занурення у

віртуальну реальність може бути досить стресовим для людей зі слабким вестибулярним апаратом, викликати запаморочення та дискомфорт в очах.

Все описане далі, буде стосуватися доповнення реального світу візуальними штучними об’єктами.

Є декілька основних технологій роботи з доповненою реальністю. Найпопулярнішим з них є підхід з використанням мобільного пристрою. У такому випадку штучні об’єкти зображуються на екрані смартфону або планшету. Сучасний мобільний пристрій здебільшого має весь набір необхідних інструментів для роботи з доповненою реальністю: акселерометр, гіроскоп, достатньо якісна камера і екран з високою роздільною здатністю [8].

Загальний алгоритм роботи системи доповненої реальності починається з аналізу відеопотоку, що надходить з камери, знаходження цілі (маркеру, чи певного фізичного об’єкта) і його аналізу (відстань, позиція, розмір, орієнтація відносно до камери). Після цього програма генерує відповідне доповнення, та об’єднує це доповнення з реальним зображенням. Результат виводиться на екран пристрою [7].

Існує декілька типів «мішеней» для доповненої реальності, на основі яких система може створювати штучні об’єкти. Далі будуть перераховані найважливіші з них.

Популярною є доповнена реальність, що базується на маркерах (Marker-based). Найбільш базовим є маркер з широкими рамками та вписаним всередину ідентифікатором (рис. 2.1.). Такі маркери дуже легко розпізнаються програмним забезпеченням, витрачаючи мало ресурсів на обробку. Крім того з таким варіантом маркеру мінімізуються ризики збоїв, що пов’язані з навколишнім середовищем (наприклад якість освітлення).

Рис. 2.1
Звичайний маркер

Рис. 2.2. Зображення як маркер

Рис. 2.3. Кодований маркет

У доповненій реальності, що базується на зображення, маркером є не квадрат з чіткими чорними рамками, а будь-яке 2D зображення (рис. 2.2.). Такий підхід зручно використовувати в поліграфії для створення анімованих картинок. Користувач може навіть на здогадуватись, що частини картинки утворюють «мішені» для доповненої реальності.

GPS- або координатно-орієнтована доповнена реальність. При такому підході віртуальний об’єкт прив’язується до конкретної точки на карті, а пристрій використовує вбудовану систему глобального позиціювання (GPS), датчик швидкості або акселерометр, цифровий компас для того щоб отримати інформацію про розміщення пристрою в просторі. Часто використовується у додатках, що орієнтовані на пошук потрібних місць та побудови маршрутів. Прикладом може слугувати й популярна гра Pokemon Go.

«Справжня» доповнена реальність працює без маркерів. Тобто їй не потрібен конкретний об’єкт-тригер. Вона здатна розпізнавати і відстежувати 3D об’єкти.

Крім того об’єкти доповненої реальності можуть «чіплятись» на закодовані маркери (рис. 2.3.), текст та геометричні фігури [7].

У 2017 році Google та Apple представили свої інструментарії для роботи з AR – AR Core та ARKit відповідно, і зараз ведуть їхню активну розробку, що

дає можливість звичайним розробникам легко створювати свої додатки з використанням доповненої реальності.

Хоча сьогодні доповнена реальність асоціюється зі сферою розваг, передбачається величезний стрибок у використанні AR в багатьох сферах до 2025 року [9], зокрема в туризмі, де 84% користувачів будуть зацікавлені у використанні доповненої реальності в подорожах (рис. 2.4.). Наприклад, за допомогою AR можна побачити, як виглядало місто багато років тому, отримати додаткову інформацію про маршрут тощо.

Прогнозується також активне використання доповненої реальності в медицині як для навчання, так і для різноманітних маніпуляцій (візуалізація ультразвукової діагностики) та навіть при хірургічних втручаннях.

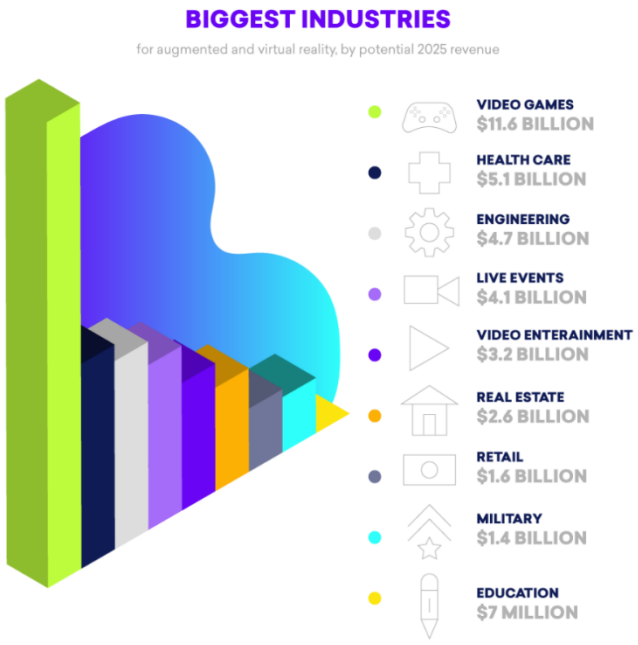

Рис. 2.4.Орієнтовна об’єми AR та VR в різних сферах у 2025 році

Доповнена реальність може бути хорошим інструментом в закладах освіти, для візуалізації інформації. Крім того вже зараз AR активно використовується в дизайні інтер’єру та маркетингу. Очікується значне

зацікавлення цією технологією у військовій справі, будівництві, авіації, ремонтних службах тощо.

2.2 Характеристики та особливості роботи ARCore від Google

ARCore - це мультиплатформений SDK від компанії Google, який дозволяє створювати додатки доповненої реальності. ARCore був представлений Google у лютому 2018 року. Це був не перший проект цієї компанії в сфері доповненої реальності. З 2014 року Google займався розробкою проекту Tango, але ця платформа була дуже залежною від апаратних характеристик пристроїв, тому компанія згорнула цей проект на користь ARCore.

На даний момент ARCore доступний для Android, Android NDK, Unity для Android, Unity для iOS, iOS та Unreal (Tango був доступний лише на пристроях випущених спеціально під нього). Для кожного з них існує офіційна документація від Google. Підтримка ARCore Android починається з версії 7.0 (Nougat).

ARCore має можливості:

- розділяти світ на умовні одиниці, знаходити об’єкти, виявляти їх розміри та інші параметри;
- відслідковувати рух, у тому числі розрізняти цілеспрямований рух від хаотичного;
- знаходити джерело світла, розрізняти тіні, сприймати освітлення;
- виявляти джерела звуку, розпізнавати голоси, обличчя, жести та таке інше [5].

AR Core використовує три основні технології: відслідковування руху (motion tracking), розуміння навколишнього середовища (environment understanding) і оцінка освітлення (light estimation).

Використовуючи гіроскоп та камеру для відслідковування опорних точок у кімнаті (точки в яких буде знаходитись віртуальний об’єкт), ARCore визначає положення та орієнтацію пристрою під час руху. При цьому віртуальні точки завжди залишаються там, де вони були розташовані з початку.

Розуміння навколишнього середовища дозволяє смартфону оцінювати розмір і положення різних типів поверхонь: вертикальних, горизонтальних та кутових. Для цього ARCore шукає кластери характерних точок, які лежать в одній площині, і робить їх доступними для застосунку у якості поверхні. В основі розуміння оточення лежить технологія SLAM (simultaneous localization and mapping). Головною задачею SLAM є побудова і постійне оновлення карти простору з одночасним відслідковуванням положення пристрою, який рухається в цьому просторі. Варто зазначити, що ArCore успадкував від Tango проблеми з розпізнаванням зеркальних поверхонь.

Lighting Estimation API аналізує особливості освітлення (рис. 2.5) на зображенні і надає детальну інформацію про освітлення даної сцени. Завдяки цьому розробник може використовувати ці дані для регулювання освітлення своїх віртуальних об’єктів. Це має важливе значення для реалістичності зображення та якості занурення користувача. На рис. 2.6, 2.7 можна прослідкувати, як змінюється зовнішній вигляд одного і того ж об’єкта в залежності від освітлення навколишнього середовища.

Перед початком відбудовування доповненої реальності застосунок на основі ARCore будує власний віртуальний світ на основі фізичного. Під час переміщення телефону, він запам’ятовує предмети, а також відслідковує як

деякі предмети рухаються стосовно камери. За рахунок цього, віртуальний об’єкт, розміщений в якомусь місці, залишиться там стояти, навіть якщо користувач покине приміщення, а потім повернеться до нього знову.

Варто зазначити, що ARCore зокрема використовує Sceneform SDK – 3D-фреймворк, що дозволяє розробникам створювати ARCore застосунки без

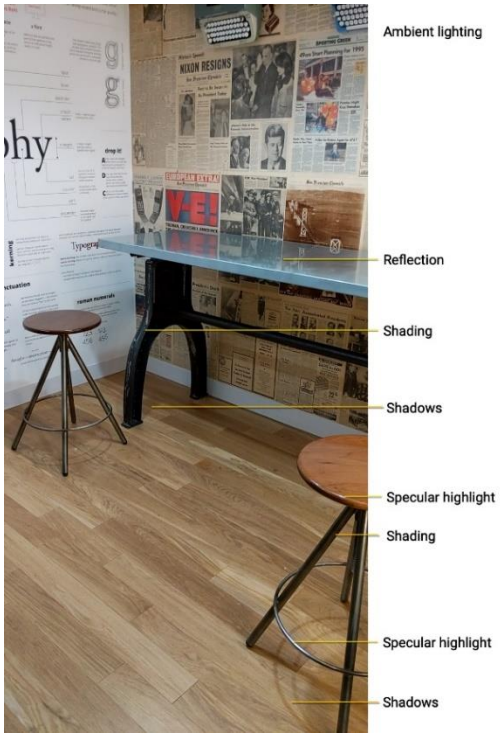

Рис. 2.5. Елементи освітлення, що аналізуються Lighting Estimation API

Рис. 2.6.Освітлення за допомогою Lighting Estimation API

Рис. 2.7. Освітлення за допомогою Lighting Estimation API

знання OpenGL (графічна бібліотека для написання завтосунків, що використовують 2D та 3D комп’ютерну графіку).

В останніх оновленнях ARCore покращилась фізика, алгоритми пошуку шляху та взаємодія з площинами. Пристрої тепер краще розпізнають глибину, використовуючи лише камеру. За рахунок цього віртуальні об’єкти можуть розміщуватись за, а не лише перед предметами реального світу (рис. 2.8.).

Також віртуальні персонажі перестали проходити через реальні об’єкти, запущені м’ячі відбиваються від стін, а користувач отримав можливість малювати не лише на стінах.

Рис. 2.8. Покращення ARCore в роботі з глибиною

Висновки до розділу 2

У цьому розділі було розглянуто технологію доповненої реальності, та її перспективи в різних сферах життя. Було досліджено та проаналізовано інструментарій ARCore від Google.

РОЗДІЛ 3. ПОЄДНАННЯ МАШИННОГО НАВЧАННЯ ТА ДОПОВНЕНОЇ РЕАЛЬНОСТІ ПРИ РОЗРОБЦІ ВЛАСНОГО ФРЕЙМВОРКУ ARMARKER

3.1 Принцип роботи фреймворку та актуальність розробки

Як було описано в попередніх розділах, машинне навчання, зокрема нейронні мережі, демонструє високу точність у розпізнаванні об’єктів на зображенні. У той же час за допомогою доповненої реальності можна додавати віртуальні об’єкти в навколишнє середовище. Поєднуючи ці дві технології можна помічати розпізнані об’єкти мітками в доповненій реальності, які будуть нести в собі додаткову інформацію. Це може бути корисним у створенні довідкових систем на підприємствах, навчанні, туризмі тощо.

Сьогодні простежується тенденція приховувати складну реалізацію, надаючи зручний інтерфейс, адже використання складних технологій на практиці вимагає значних ресурсів, тому що розробникам потрібен час на опанування цих технологій. Широке розповсюдження застосунків доповненої реальності в багатьох сферах стримується через те, що мало розробників мають знання у відповідній сфері чи готові витрачати час на її вивчення. Зважаючи на ці фактори, метою розробки фреймворку ARMarker було створення інструментарію, який міг би допомогти розробникам у побудові застосунків, пов’язаних з задачами розпізнавання та маркування об’єктів без необхідності розуміння роботи інструментів ML Kit та AR Core. Ідея для розробки була запозичена у студента Олександра Франківа [10], який займався створенням аналогічного фреймворку під систему iOS. Основною задачею розробленого фреймворку є розпізнавання об’єктів реального світу чи QR кодів та відображення відповідних міток поруч з ними у доповненій

реальності. Важливо є те, що фреймворк досить гнучкий, має декілька режимів роботи та може налаштовуватись розробником відповідно до потреб.

Як вже згадувалось, фреймворк може працювати у двох режимах: розпізнавання QR кодів та розпізнавання образів. Розпізнавання образів працює, якщо задана модель нейронної мережі, в іншому випадку фреймворк працює в режимі розпізнавання QR-кодів. Для маркування можна використовувати різні типи міток:

1. Текстова мітка з заданим заповненням (використовується за замовчуванням) (рис. 3.1.).
2. Мітка з віртуальним відеоплеєром, для програвання відео за заданим URL (рис. 3.3.).
3. Тривимірний об’єкт заданий користувачем (рис. 3.2.).
4. Власна створена користувачем мітка (має успадковуватись від класу RenderableLabel).

Рис. 3.1. Текстова мітка

Рис. 3.2. Мітка – тривимірний об’єкт

Рис. 4 Відео мітка

3.2 Особливості будови фреймворку

Як було зазначено основна робота фреймворку ARMarker відбувається на базі інструментів Google ML Kit та ARCore. Далі буде описано архітектуру фреймворку, важливі деталі реалізації та особливості поєднання цих двох інструментів.

Фреймворк написаний на Android (Kotlin) та використовує інструменти Firebase ML Kit, AR Core і бібліотеку Sceneform від Google та вбудований пакет Android.Graphics.

Головним класом фреймворку є контролер ARLabeler (рис. 3.4.), в параметри якого передається екземпляр ARFragment, на якому буде здійснюватись відображення об'єктів, мітка, яка буде використана для відображення (якщо параметр відсутній, буде використана текстова мітка за замовчуванням) та за бажанням назва нейронної моделі. Відповідно до цих параметрів контролер визначає, який режим розпізнавання буде

```
class ARLabeler (arFragment: ArFragment, renderableLabel: RenderableLabel? = null, model: String? = null)
```

Рис. 3.4. Визначення контролера ARLabeler

здійснюватись та викликає відповідний клас аналізатора. Всі класи аналізатора реалізують кастомний інтерфейс Analyzer та перевизначають метод runDetection(bitmap: Bitmap). Всі типи міток реалізують інтерфейс RenderableLabel (рис. 3.5). Також можна створювати власні мітки на основі цього інтерфейсу.

```
interface  RenderableLabel{
      fun addLabelToScene(arFragment: ArFragment, anchor: Anchor)
      fun setTextToLabel(text: String)
}
```

Рис. 3.5.Інтерфейс RenderableLabel

Важливою деталлю є те, що для ML Kit стандартним є використання бібліотеки CameraX API, що значно полегшує роботу з камерою. Бібліотека

містить три основні методи: Preview (відображення вікна камери на екрані застосунку), Image analysis (аналіз зображення – метод для роботи зокрема і з MLKit) та Image capture (збереження зображення у високій якості) [1]. Але для роботи з ARCore необхідний такий елемент, який не лише може відображати відеопотік з камери, а й додавати до нього відрендерені об’єкти, а отже від використання CameraX довелось відмовитись на користь ARFragment. У такому випадку на вхід до аналізатора ML Kit подається екземпляр класу Bitmap, який створюється на основі пікселів з зображення на екрані. Розпізнавання відбувається у паралельному потоці.

Стандартний сценарій застосунку на основі фреймворку ARMarker у режимі розпізнавання об’єктів починається з пошуку об’єктів на зображенні за допомогою функції Object Detection and Tracking із ML Kit. У результаті можна отримати координати прямокутника, в якому знаходиться знайдений об’єкт. Насправді ця функція може знаходити відразу декілька об’єктів на зображенні, але на даному етапі фреймвок працює лише з одним, який він визначає найпріоритетнішим. На основі знайденого фрагменту будується Bitmap, який розглядається як окреме зображення і обробляється іншою функцією ML Kit Label Image. Це важливо, адже Label Image не вміє класифікувати окремі об’єкти на зображенні, а лише все зображення в цілому.

Розпізнавання здійснюється за допомогою заданої попередньо натренованої моделі. Результатом такого розпізнавання є список з n кращих результатів. У даній реалізації береться лише перший результат. Після цього відбувається рендеринг мітки обраного типу. Якір мітки ставиться в центр екрану з розпізнаним зображенням.

У режимі розпізнавання QR-міток використовується функція Scan Barcodes ML Kit. Рендеринг мітки відбувається аналогічним чином. Якщо QR-код містить URL відео, воно може бути відображено на відео-мітці.

Для використання ARMarker в своєму проекті достатньо додати його як окремий модуль до проекту, та підключити його. Для підключення потрібно перейти у файл build.gradle (app) та додати залежність в dependencies (рис.

```
implementation project(':arlabeler')
```

Рис. 3.6.Підключення фреймворку до застосунку

3.6.). Після цього необхідно створити екземпляр контролера ARLabeler з даного фреймворку та передати першим параметром ARFragment. Інші параметри опціональні і задаються відповідно до потреб.

3.3 Можливості для вдосконалення фреймворку

Фреймворк ARMarker має широкі можливості для покращення. Найпріоритетнішим завданням є покращення розрахунку точки прикріплення якоря з врахуванням глибини зображення. Також можна забезпечити можливість розпізнавання декількох об’єктів та надати користувачеві можливість отримувати не один, а декілька результатів розпізнавання для того, щоб можна було самостійно вирішувати як ранжувати результати. Крім того цікавою задачею є розширення функціоналу так, щоб фреймворк мав можливість ставити мітки не лише на статичні об’єкти, а й прикріплювати їх до рухомих об’єктів.

Важливо зауважити, що оскільки ARMarker побудований на основі інших інструментів, то він перебирає всі їх переваги та недоліки. Як вже згадувалось ML Kit доступний лише в бета-версії, а отже знаходиться на стадії активної розробки і внесення змін. За час розробки було помічено, що він має проблеми з витоками пам’яті, що може спричиняти аварійне завершення програми при розпізнаванні відеопотоку на пристроях з низькими технічними

характеристиками. Тим не менш, якість розпізнавання та класифікації об’єктів досить висока. AR Core має певні проблеми з розпізнаванням вертикальних поверхонь, а також все ще не ідеально взаємодіє з реальними предметами навколишнього простору.

Отже якість розпізнавання, швидкодія та продуктивність ARMarker напряму залежить від інструментів ML Kit та AR Core, а оскільки вони активно розвиваються, то з кожним оновленням цих інструментів буде покращуватись і розроблений фреймворк.

Висновки до розділу 3

У цьому розділі було описано розроблений фреймворк ARMarker, його принципи роботи, архітектуру, технічні особливості та перспективи використання. Також було розглянуто недоліки фреймворку та можливості для вдосконалення.

ВИСНОВОК

Машинне навчання та доповнена реальність мають великий потенціал поза індустрією ігр та можуть внести якісні зміни у різні сфери життя. Особливо це актуально в час, коли ці технології стали доступними пересічним користувачам на їхніх мобільних пристроях. Великі компанії, зокрема Google та Apple ведуть активну розробку інструментів, але середньостатистичні розробники ще не почали активно використовувати їх у своїх додатках, адже вивчення цих технологій потребує значних витрат.

Отже, керуючись факторами, описаними вище, в рамках даної роботи було розроблено фреймворк ARMarker для використання доповненої реальності та машинного навчання у власних застосунках без необхідності розуміння принципів роботи ML Kit та ARCore. Фреймворк досить гнучкий, дозволяє користувачеві налаштовувати режим відповідно до потреб, а також додавати свої типи міток. Його використання може значно пришвидшити розробку продукту, а отже і позитивно вплинути на його собівартість. Фреймворк може бути використаним для створення довідкових систем, додатків для туризму, навчання тощо.

Розроблений фреймворк має широкі можливості як для вдосконалення існуючих функції, так і для створення нових.

ВИКОРИСТАНІ ДЖЕРЕЛА

1. Android Documentation [Електронний ресурс] / Google Developers – Режим доступу до ресурсу: https://developer.android.com/docs
2. ML Kit for Firebase Documentation [Електронний ресурс] / Google Developers – Режим доступу до ресурсу: https://firebase.google.com/docs/ml-kit.
3. Штучні нейронні мережі обчислення / М.А. Новотарський Б.Б. Нестеренко., 2014.
4. ML Kit for Firebase: features, capabilities, pros and cons [Електронний ресурс] / Vitaly Kuprenko., 2019 – Режим доступу до ресурсу: https://towardsdatascience.com/ml-kit-for-firebase-features-capabilities-pros-and-cons-a182b4299cc
5. ARCore Documentation [Електронний ресурс] / Google Developers – Режим доступу до ресурсу: https://developers.google.com/ar.
6. A Survey of Augmented Reality Presence: Teleoperators and Virtual Environments/ R. Azuma., 1977. - 355—385 с.
7. Augmented Reality for Developers / K. Babilinski, J. Linowes., 2017.
8. OverLay: Practical Mobile Augmented Reality / Puneet Jain, Justin Manweiler, Romit Roy Choudhury.
9. Predictions for the Future of Augmented Reality [Електронний ресурс] / 2018. – Режим доступу до ресурсу: https://hackernoon.com/predictions-for-the-future-of-augmented-reality-63c7b8c9d794
10. Особливості застосування машинного навчання та доповненої реальності на пристроях на базі iOS/ Олександр Франків, 2019