Вінницький національний технічний університет

Факультет інформаційних технологій та комп'ютерної інженерії

Кафедра програмного забезпечення

Пояснювальна записка

до магістерської кваліфікаційної роботи

магістр

(освітньо-кваліфікаційний рівень)

на тему: Розробка методу і програмних засобів обробки й проекціюнування веб-контенту з використанням доповненої реальності

Виконав: студент II курсу
групи 1ПІ-18 м
спеціальності
121 – Інженерія програмного
забезпечення
(шифр і назва напряму підготовки, спеціальності)
Богачук Г. В.
(прізвище та ініціали)
Керівник: к.т.н., доц. каф. ПЗ Войтко В.В.
(прізвище та ініціали)

ВНТУ – 2019

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра програмного забезпечення
Освітньо-кваліфікаційний рівень – магістр
Спеціальність 121 – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ
Завідувач кафедри ПЗ
Романюк О. Н.
"____"_____2019 року

З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Богачук Галині Володимирівні

1. Тема роботи – Розробка методу і програмних засобів обробки й проекціюнування веб-контенту з використанням доповненої реальності.

Керівник роботи: Войтко Вікторія Володимирівна, к.т.н., доцент кафедри ПЗ, затверджені наказом вищого навчального закладу від "__"_____2019 року №___

2. Строк подання студентом роботи

_____

3. Вихідні дані до роботи :

Мова програмування: Python

Технології: AR, Xcode, CoreML та Vision фреймворки

ОС: iOS

4. Зміст розрахунково-пояснювальної записки: вступ; аналіз інформаційного забезпечення та постановка задачі; розробка методу і моделей для оптимізації нейронних мереж та просторового сприйняття AR; розробка програми для проеціювання динамічного веб-контенту у доповненій реальності; тестування програмної системи; економічна частина; висновки; перелік посилань; додатки.

5. Перелік графічного матеріалу:

моделі системи; інтерфейс; структура баз даних; AR-модель; алгоритм роботи авторизації системи; блек-схема роботи додатку; приклад вигляду розробленої програми.

6. Консультанти розділів магістерської кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видано	завдання прийнято
1–4	к.т.н. Войтко В. В., доцент кафедри ПЗ		
5	к.е.н. Бальзан М. В., доцент кафедри ЕПВМ		

7. Дата видачі завдання_____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз інформаційного забезпечення та постановка задачі	07.10.2018 – 27.10-2019	Вик.
2	Розробка методу і моделей для оптимізації нейронних мереж та просторового сприйняття AR	28.10.2019 – 8.11.2019	Вик.
3	Розробка програми для проеціювання динамічного веб-контенту у доповненій реальності	9.11.2019 – 20.11.2019	Вик.
4	Тестування програмної системи	21.11.2019 – 01.12.2019	Вик.
5	Економічна частина	02.12.2019 – 07.12.2019	Вик.

Студент _____ **Богачук Г. В.**

(підпис) (прізвище та ініціали)

Керівник магістерської кваліфікаційної роботи _____ **Войтко В. В.**

(підпис) (прізвище та ініціали)

АНОТАЦІЯ

У магістерській кваліфікаційній роботі проведено детальний аналіз процесу обробки та проеціюнування web-контенту з використанням засобів доповненої реальності.

Запропоновано метод для проекціюнування динамічного веб-контенту у доповненій реальності, що дозволяє побудувати AR застосунок доповненої реальності із забезпеченням високої реалістичності зображень. Розроблено модель системи, що здатна тренуватися на обраному датасеті. Запропоновано спосіб постачання моделей на мобільний застосунок, зберігання та передача мобільному застосунку тривимірний, двовимірний або динамічний веб-контент у вигляді 3D моделей, зображень, відео, відеопотоку, анімацій, веб-сторінок, динамічної обробки інформації. Здійснюється передача отриманого результату на компонент доповненої реальності для подальшого проеціюнування.

Розроблена програма проеціюнування динамічного веб-контенту у доповненій реальності дозволяє моделювати процеси доповнення контенту. Програма розроблена мовою Python.

ABSTRACT

In the master's qualification work a detailed analysis of the process of processing and projection of web-content with the use of augmented reality was carried out.

We propose a method for projecting dynamic web content in augmented reality, which allows you to build an augmented reality AR application with high image realism. A model of a system capable of training on the selected dataset has been developed. A method of delivering models to a mobile application, storing and transmitting to a mobile application three-dimensional, two-dimensional or dynamic web content in the form of 3D models, images, videos, video stream, animations, web pages, dynamic information processing is proposed. The result is transferred to the augmented reality component for further projection.

The developed augmentation program for dynamic web content in augmented reality allows us to model the processes of augmenting the content. The program was developed in Python.

Зміст

ВСТУП

Обґрунтування вибору теми дослідження. Останнім часом завдяки активному розвитку технологій людство все частіше реалізує концепти, що були закладені ще наприкінці минулого сторіччя, зокрема такі, як реалізація віртуальної та доповненої реальності. Бурхливий розвиток мобільних технологій дозволяє реалізувати засоби доповненої реальності навіть на смартфоні користувача, утилізуючи як програмні, так і апаратні потужності пристрою [1].

Чому ж доповнена реальність стає такою популярною? Доповнена реальність (англ. augmented reality, далі – AR) — термін, що позначає всі проекти, спрямовані на доповнення реальності будь-якими віртуальними елементами. В книзі Брет Кінг «Епоха доповненої реальності» [1] йдеться про найважливіший аспект AR – розвиток апаратної складової. Реалізація доповненої реальності передбачає просторові обчислення та аналіз зображення з камери, тобто використовує засоби машинного навчання.

Методи доповненої реальності можуть бути використані у широкому спектрі застосувань, дозволяючи візуалізувати програмні об’єкти у проекції на реальний світ через камеру користувача, що може бути корисним у медицині, освіті, розважальній та навіть військових сферах. Вже зараз існують численні рішення з використанням AR: інтерактивні гіди, застосунки для тренувань, що показують модель тіла людини відносно тренажеру [2], медичні застосунки, що проеціюють внутрішні травми на тіло людини. Але у існуючих рішень є проблеми з відображенням та взаємодією з веб-сторінками, що обмежує можливості як розробника, так і користувача. У цієї технології є великий потенціал, її можливості постійно розвиваються, але деяких рішень ще просто не існує, зокрема методів взаємодії засобів доповненої реальності та веб-контенту.

Тому актуальною є розробка методу для проекціюнування динамічного веб-контенту у доповненій реальності.

Мета та завдання дослідження. Метою роботи є процес покращення якості взаємодії технологій доповненої реальності та веб-контенту, підвищення реалістичності доповнених елементів на веб-сторінці за рахунок використання розроблених моделей та методу проекціонування динамічного веб-контенту у доповненій реальності, що дозволяє побудувати AR застосунок доповненої реальності із забезпеченням високої реалістичності зображень.

Основними задачами дослідження є:

- Удосконалення методу проекціонування динамічного веб-контенту у доповненій реальності.
- Розробка моделей системи, що здатна тренуватися на обраному датасеті.
- Збереження моделей доповненої реальності.
- Постачання моделей на мобільний застосунок, зберігання та передача мобільному застосунку тривимірний, двовимірний або динамічний веб-контент у вигляді 3D моделей, зображень, відео, відеопотоку, анімацій, веб- сторінок, здійснення динамічної обробки інформації.
- Передача отриманого результату на компонент доповненої реальності для подальшого проекціонування.

Об'єктом дослідження є процес AR (доповненої реальності), що базується на маркерах, CoreML фреймворці, Vision фреймворці, Create ML утиліті, фреймворці для навчання моделі TuriCreate, програмі для розмітки фотографій RectLabel та скрипті Python.

Предметом дослідження є засоби релізації AR модулів.

Методи дослідження. У процесі досліджень використовувались:

- теорія нейронних мереж для аналізу та розпізнавання образів;

- методи 3D-моделювання для створення віртуальних моделей об’єктів;

- методи комп’ютерного моделювання для тестування роботи програми. моделі сприйняття оточення методами доповненої реальності.

Наукова новизна:

1. Подальшого розвитку дістав метод проекціонування динамічного веб-контенту у доповненій реальності, який, на відміну від існуючих,

орієнтований на проекціюнування динамічного веб-контенту у доповненій реальності за рахунок використання технології AR у динамічних веб-ресурсах, що дозволяє збільшити реалістичність зображень доповненої реальності.

2. Подальшого розвитку дістали моделі проекціюнування динамічного веб-контенту що, на відміну від існуючих, здатні тренувати на обраному датасеті та зберігати моделі доповненої реальності, постачати їх на мобільний застосунок, зберігати та передавати мобільному застосунку тривимірний, двовимірний або динамічний веб-контент у вигляді 3D моделей, зображень, відео, відеопотоку, анімацій, веб-сторінок, динамічно обробляти їх для подальшого використання у доповненій реальності та передавати отриманий результат на компонент доповненої реальності для подальшого проекціюнування, що забезпечує підвищення реалістичності доповненого контенту.

Процес передачі, генерації та проекціюнування контенту відбувається в реальному часі у працюючому додатку. Для вирішення задачі створення і постачання нейронних мереж необхідно налаштувати алгоритм тренування моделі нейронної мережі на віддаленому сервері із запропонованим датасетом, переведення цієї мережі у формат, що буде здатний працювати з мобільними технологіями доповненої реальності. Також варто враховувати те, що додаток працює з просторовими моделями, тому при отриманні веб-контенту і проекції його на реальний світ необхідно спочатку побудувати віртуальний об'єкт, на який цей контент буде проекціюнуватися, що враховує складну просторову геометрію розташування для побудови цього об'єкта.

Практична цінність одержаних результатів полягає в тому, що на основі отриманих в магістерській кваліфікаційній роботі теоретичних положень удосконалено метод проекціонування веб-контенту за допомогою технолігії віртуальної реальності та розроблено програмні засоби для проекціонування та візуалізації тривимірних зображень.

Зв’язок роботи з науковими програмами, планами, темами. Робота виконувалася відповідно до плану науково-дослідних робіт кафедри програмного забезпечення.

Роботу виконано на замовлення ТОВ «Кусто Агро», про що свідчить акт впровадження.

Апробація матеріалів магістерської кваліфікаційної роботи. Результати роботи доповідалися на двох конференціях: на Всеукраїнській науково-практичній Інтернет-конференції «Молодь в науці: дослідження, проблеми, перспективи - 2019» та на XLVII Міжнародній науково-технічній конференції факультету інформаційних технологій та комп'ютерної інженерії 2018 року.

Публікації. За тематикою дослідження опубліковано дві наукових публікації в матеріалах XLVII Міжнародної науково-технічної конференції факультету інформаційних технологій та комп'ютерної інженерії 2018 [3] та Всеукраїнської науково-практичної інтернет-конференції «Молодь в науці: дослідження, проблеми, перспективи-2019 [4].

Результати роботи, зокрема графічні зображення, подавалися на **міжнародні та всеукраїнські конкурси**, де отримали наступні перемоги:

- 1 місце у VI Міжнародному конкурсі науково-дослідних робіт «Ерудит» Дата проведення: з 15 червня 2016
- Диплом переможця (2 місце) у XIV Міжнародному бліц-конкурсі з комп’ютерної графіки та Веб-дизайну в номінації Веб-дизайн (краще інформаційне наповнення), 2017 р.
- 3 місце в у XIII Міжнародному Бліц-конкурсі з комп’ютерної графіки та Веб-дизайну в номінації «Краща 3Д-графіка», 2016
- Диплом учасника XIV Міжнародного бліц-конкурсу з комп’ютерної графіки та Веб-дизайну в номінації Веб-дизайн, 2017 р.
- Диплом за 3 місце в у Міжнародному Конкурсі з комп’ютерної графіки та Веб-дизайну в номінації «Краща 3Д-графіка», 2014 р.

- Диплом другого ступеня (2 місце) на Міжнародному конкурсі з 3Д-графіки та Веб-дизайну, в номінації «Best 2D raster graphics (photo collage)», 2015 р.
- Диплом учасника конкурсу «Геометрические модели и программы компьютерной графики», номінація «Компьютерный дизайн» 2015 р.
- Диплом лауреата 2 ступеня (2 місце) в XIX Міжнародному конкурсі комп'ютерної графіки та художнього фото «Зірковий проект» 2014
- 2 місце в XVIII Міжнародному конкурсі комп'ютерної графіки та художнього фото «Зірковий проект»,
- Диплом переможця VI-го Міжнародного конкурсу комп'ютерного малюнку та колажу «Творчество без границ», Болгарія, Хасково, 2015 р.
- Лауреат 2 ступеня XVII-го міжнародного конкурсу комп'ютерної графіки та художнього фото «Зірковий проект», 2014
- Диплом учасника II-го Відкритого міжнародного конкурсу комп'ютерної графіки «Волшебный мир сказок», 2014 р.
- Сертифікат учасника XII-го міжнародного молодіжного фестивалю інформаційних технологій, за участь в номінації «Web-дизайн», 2015 р.

Структура та обсяг роботи. Магістерська кваліфікаційна робота складається зі вступу, п'яти розділів, висновків, списку літератури, що містить 21 найменування, 4 додатка. Робота містить 15 ілюстрацій, 3 таблиці.

Робота складається з 5 розділів. У першому розділі було визначено актуальність роботи, виявлено проблему, яка полягає в тому, що для належного використання моделей нейронних мереж вони мають бути включені у бандл мобільного застосунку, що значно обмежує гнучкість розробки таких систем.

Досліджено поняття доповненої, змішаної та віртуальної реальностей, розділяти поняття віртуальної, доповненої та змішаної реальностей, які вимагають особливого обладнання, такого як шоломи, датчики, джерело струму для повної емуляції реальності, доповнена та змішана ж реальності.

Сформульовано задачі магістерської кваліфікафійної роботи.

У другому розділі розроблено математичну моделі нейронних мереж для розпізнавання образів та для класифікації зображень, розроблено згортуючу нейронну мережу, яку необхідно тренувати на запропонованому датасеті. Для вирішення проблеми враховано розролену математичну модель. Розроблено модель системи доповненої реальності, блок-схема\у модуля авторизації додатку.

У третьому розділі магістерської роботи проведено варіантний аналіз та обгрунтування вибору середовища розробки та аналіз вибору мови програмування і шаблону проектування. Для розробки додатку було обрано технологію Xcode та описано переваги цієї технології серед аналогів. За результатами аналізу, мовою програмування було обрано Python. Реалізований програмний AR застосунок.

Четвертий розділ показує результати тестування розробленого мобільного додатку за новим методом.

У п’ятому розділі розраховуються витрати на розробку програмного забезпечення, експлуатаційні витрати, обсяг роботи, пов'язаної з використанням програмного забезпечення, та економічний ефект від впровадження нового програмного продукту.

У висновках наведені основні результати дослідження.

У додатках міститься технічне завдання, акт впровадження, лістинг коду основних частин програми та ілюстративний матеріал до захисту магістерської кваліфікаційної роботи.

1 АНАЛІЗ ОСОБЛИВОСТЕЙ РОЗВИТКУ ДОПОВНЕНОЇ РЕАЛЬНОСТІ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ

1.1 Аналіз розвитку технологій доповненої реальності

Останнім часом завдяки активному розвитку технологій людство все частіше реалізує концепти, що були закладені ще наприкінці минулого сторіччя, зокрема такі, як реалізація віртуальної та доповненої реальності [3].

Особливо бурхливого розвитку набули мобільні технології, що обумовлює використання засобів доповненої реальності навіть на смартфоні користувача, утилізуючи як програмні, так і апаратні потужності пристрою.

Доповнена реальність (AR) - це інтерактивний досвід середовища реального світу, де об'єкти, що знаходяться в реальному світі, "доповнюються" комп'ютерно-генерованою перцептивною інформацією, іноді через декілька сенсорних модальностей, включаючи зорові, слухові, гаптичні, соматосенсорні і нюхові.

Накладена сенсорна інформація може бути конструктивною (тобто добавкою до природного середовища) або маскуючою (тобто маскуванням природного середовища) і безперешкодно переплітається з фізичним світом так, що вона сприймається як аспект реального середовища. Таким чином, доповнена реальність змінює постійне сприйняття середовища реального світу, на відміну від віртуальної реальності, що повністю замінює реальне середовище користувача імітаційною. Доповнена реальність пов'язана з двома в значній мірі синонімічними термінами: змішаною реальністю і комп'ютерно-опосередкованою реальністю.

Основна цінність розширеної реальності полягає в тому, що вона приносить компоненти цифрового світу до сприйняття людиною реального світу і робить це не як просте відображення даних, а через інтеграцію захоплюючих відчуттів, які сприймаються як природні частини навколишнього середовища. Перші функціональні системи AP, що забезпечували користувачам вражаючі враження про змішану реальність, були

винайдені на початку 1990-х років, починаючи з системи віртуальних світильників, розробленої в 1992 році в Армстронській лабораторії ВПС США. Ігрові підприємства, а тепер і інші галузі також зацікавлені в можливостях AP, наприклад, у обміні знаннями, освіті, управлінні інформаційним потоком і організації дистанційних зустрічей [1]. Доповнена реальність також трансформує світ освіти, де вміст може бути доступний шляхом сканування або перегляду зображення за допомогою мобільного пристрою або шляхом залучення до класу захоплюючих, без маркерів досвіду AR. Іншим прикладом є шоломи AR для будівельників, які відображають інформацію про будівельні об'єкти [4].

Доповнена реальність використовується для посилення природного середовища або ситуацій і пропонує сприйняття збагаченого досвіду. За допомогою передових технологій AR (наприклад, додавання комп'ютерного зору та розпізнавання об'єктів) інформація про навколишній реальний світ користувача стає інтерактивною та цифровою. Інформація про навколишнє середовище та його об'єкти накладається на реальний світ. Ця інформація може бути віртуальною або реальною, наприклад, бачачи іншу реальну відчутну або виміряну інформацію, таку як електромагнітні радіохвилі, накладені в точному узгодженні з місцем, де вони перебувають у просторі. Доповнена реальність також має величезний потенціал у зборі та обміну мовчазними знаннями. Методи розширення зазвичай виконуються в реальному часі і в семантичному контексті з елементами навколишнього середовища. Імперсивна перцептивна інформація іноді поєднується з додатковою інформацією, подібною оцінці над живим відеострімом спортивної події. Це поєднує в собі переваги технології розширеної реальності та технології відображення інтерфейсу (HUD)[5].

1.2 Виявлення проблем та актуалізація рішень

Зараз існують способи пакування та доставку контента у мобільний застосунок, проте основною проблемою цих способів є те, що зазвичай контент, що проекціюється, знаходиться у основному пакеті системи та обробляється під час компіляції та пакування системи у виконуваний файл, тому створення вищеописаної системи, що реалізує метод динамічного постачання контента, є необхідним.

Однією з ключових проблем у створенні досвіду доповненої реальності є належне закріплення віртуального вмісту в реальному світі; процес, який вимагає унікального набору сприйнятливий технологій, здатних відстежувати високу динаміку поверхні реальних об’єктів і належного проекціюнування цільової моделі на ці поверхні, для цього необхідні належним образом треновані нейронні мережі, що здатні класифікувати і визначати шукані об’єкти як якорі для проекціюнування контенту. Проблема полягає в тому, що для належного використання моделей нейронних мереж вони мають бути включені у бандл мобільного застосунку, що значно обмежує гнучкість розробки таких систем. Тому було вирішено дослідити цю проблему та запропонувати її вирішення шляхом динамічного постачання тренованих моделей до кінцевого користувача з можливістю подальшого використання у встановленому на пристрої користувача додатку [6].

Також великою проблемою є динамічність моделей нейронних мереж доповненої реальності. Зазвичай це заготовлені моделі що йдуть вже у SDK “з коробки”: обличчя, площини, qr-коди, собаки, кішки, без можливості змінити їх “на льоту”, а при спробі додати свою власну нейронну мережу користувач зустрінеться з проблемами сумісності, високого навантаження на систему та великого розміру цих моделей, тож окрім проблеми динамічності ще існує проблема оптимізації.

1.3. Аналіз понять змішаної, доповненої та віртуальної реальності

У 1994 році Пол Мілграм і Фуміо Кішино визначили змішану реальність як "... десь між екстремумами континууму віртуальності" (VC), де континуум віртуальності поширюється від абсолютно реального до повністю віртуального середовища з розширеною реальністю і збільшеною віртуальністю. між ними. Перша повністю занурена система змішаної реальності була платформою Virtual Fixtures, розробленою у ВПС США, лабораторіями Armstrong в 1992 році Луїсом Розенбергом, щоб дозволити користувачам керувати роботами в реальних середовищах, які включали реальні фізичні об'єкти та 3D віртуальні накладки, які називаються "світильниками" які додавалися, підвищують продуктивність виконання маніпуляційних завдань [7]. Опубліковані дослідження показали, що, впроваджуючи віртуальні об'єкти в реальний світ, значні збільшення продуктивності можуть бути досягнуті людиною.

Континуум змішаної реальності є однією з двох осей у концепції Стіва Манна про опосередковану реальність, реалізовану різними зварювальними шоломами та носими комп'ютерами, а також носимими фотографічними системами, які він створив у 1970-х та на початку 1980-х років, а друга ось - медіальний континуум. наприклад, зменшена реальність (як реалізована в зварювальному шоломі або окулярах, які можуть блокувати рекламу або замінювати реальну рекламу корисною інформацією)

Традиційно розглядається середовище віртуальної реальності, в якому учасник-спостерігач повністю занурюється і здатний взаємодіяти з повністю синтетичним світом, який може імітувати властивості деяких реальних середовищ. як існуючий, так і вигаданий, але він також може перевищувати межі фізичної реальності, створюючи світ, в якому фізичні закони, які зазвичай керують простором, часом, механікою, властивостями матеріалу і т.д., більше не тримаються. однак, те, що VR мітка також часто використовується у зв'язку з безліччю інших середовищ, до яких повне

занурення і повний синтез не обов'язково стосуються, але які потрапляють десь уздовж континууму віртуальності. підклас пов'язаних з ВР технологій, які передбачають злиття реальних і віртуальних світів, які ми називаємо загалом як рмішана реальність (MR)", схема екстремумів яких зображена на рисунку 1.2.

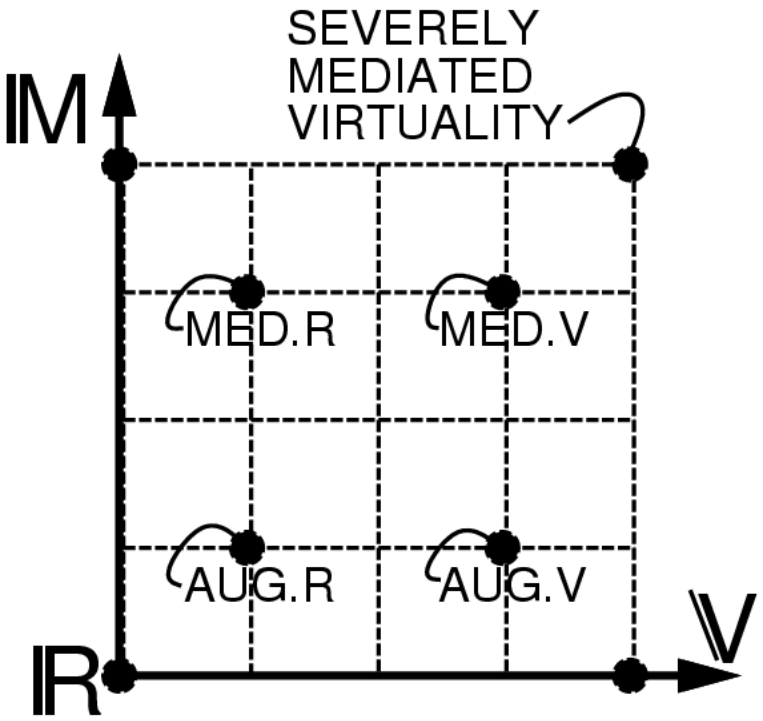


Рисунок 1.2 - Суворо опосередкована віртуальність

Вісь віртуальності (зліва направо) і вісь медіальності (знизу вгору) континууму опосередкованої реальності. Тут показані чотири приклади: доповнена реальність, розширена віртуальність, опосередкована реальність і опосередкована віртуальність на вісі віртуальності і медіальності. Це включає, наприклад, зменшену реальність (наприклад, комп'ютеризовані зварювальні шоломи, які відфільтровують і зменшують певні частини сцени) [8].

У контексті фізики термін "система інтерреальності" (див. рис. 1.2) відноситься до системи віртуальної реальності, пов'язаної з її колегою реального світу. Доповідь у випуску Physical Review E від травня 2007 року описує систему інтерреальності, що включає реальний фізичний маятник,

приєднаний до маятника, який існує тільки у віртуальній реальності, градацію переходу від реальності до власне віртуальності можна побачити на рисунку1.3.

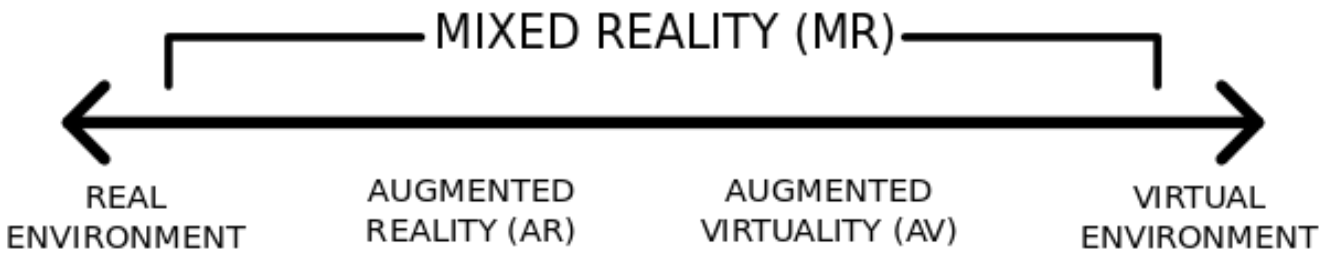


Рисунок 1.3 - Континуум реальності-віртуальності

Ця система, мабуть, має два стабільних станів руху: стан "подвійної реальності", при якому рух двох маятників є некорельованим і станом "змішаної реальності", в якому маятник має стабільний фазово-заблокований рух, який сильно корелює. Використання термінів "змішана реальність" і "інтерреальність" у контексті фізики чітко визначено, але може бути дещо іншим в інших областях.

Важливо чітко розділяти поняття віртуальної, доповненої та змішаної реальностей. Віртуальна реальність вимагає особливого обладнання, такого як шоломи, датчики, джерело струму для повної емуляції реальності, доповнена та змішана ж реальності, як це слідує з назви, використовують оточення та доповнюють його проекціями, що вимагає набагато менших затрат ресурсів. Саме тому доповнена реальність зараз набагато доступніша, перспективніша, та, як наслідок, актуальніша.

1.4 Аналіз існуючих рішень

Дослідимо існуючі рішення, щоб з’ясувати чи існує вирішення поставлених проблем та щоб дослідити найбільш розповсюджені підходи на ринку. Огляд існуючих інструментів:

Загалом, на ринку можна виділити два напрямки розвитку AR інструментарію: пропрієтарні засоби, серед яких найкращими є Apple ARKit

та Google ARCore пропрієтарні засоби, серед яких найкращими є Apple ARKit та Google засоби з відкритим вихідним кодом, зокрема OpenCV системи Короткий огляд обраних технологій:

ARKit - Завдяки своїм останнім чіпам A11 і глибоко сприймаючим камерам, Apple зробила крок вперед, випустивши ARKit для розробників iOS. Вона забезпечує постійний досвід AP, який зберігається між сесіями і може бути відновлений пізніше. Крім того, ці сеанси можуть бути спільними для декількох користувачів, кожен зі своїм пристроєм iOS. Звичайно, ARKit також забезпечує виявлення і відстеження об'єктів і об'єктів: ARKit 1.5 додав підтримку для виявлення 2D зображень, дозволяючи вам запускати досвід AR на основі 2D зображень, таких як плакати, ілюстрації або знаки; ARKit 2 розширює цю підтримку, пропонуючи повне відстеження 2D зображень, так що ви можете включати рухомі об'єкти, такі як коробки для виробів або журнали, у ваш досвід AR; ARKit 2 також додає можливість виявлення відомих 3D-об'єктів, таких як скульптури, іграшки або меблі [9].

ARCore - Google також розробила платформу для створення досвіду розширеної реальності для Android і iOS. Використовуючи різні API, ARCore дозволяє вашому телефону відчувати своє оточення, розуміти світ і взаємодіяти з інформацією. Деякі API доступні для Android та iOS, щоб дозволити спільний досвід AR. По суті, ARCore робить дві речі: відстежує позицію мобільного пристрою під час його руху і будує своє власне розуміння реального світу. Технологія відстеження руху ARCore використовує камеру телефону для виявлення цікавих точок, які називаються функціями, і відстежує, як ці точки рухаються з плином часу. Завдяки поєднанню переміщень цих точок і показань з інерційних датчиків телефону, ARCore визначає як положення, так і орієнтацію телефону, коли він рухається по простору. На додаток до ідентифікації ключових точок, ARCore може виявляти плоскі поверхні, такі як стіл або підлога, а також може оцінити середнє освітлення в районі навколо нього. Ці можливості поєднують ARCore, щоб побудувати своє власне розуміння навколишнього світу.ARCore

використовує камеру вашого телефону для відстеження руху, дозволяючи йому зрозуміти і відстежувати своє положення щодо світу. Після цього він використовує екологічне розуміння для визначення розміру та розташування всіх типів поверхонь: горизонтальних, вертикальних та кутових поверхонь, таких як земля, журнальний столик або стіни. Нарешті, він виконує оцінку світла для оцінки поточних умов освітлення. ARCore у роботі можна побачити на рисунку 1.4.

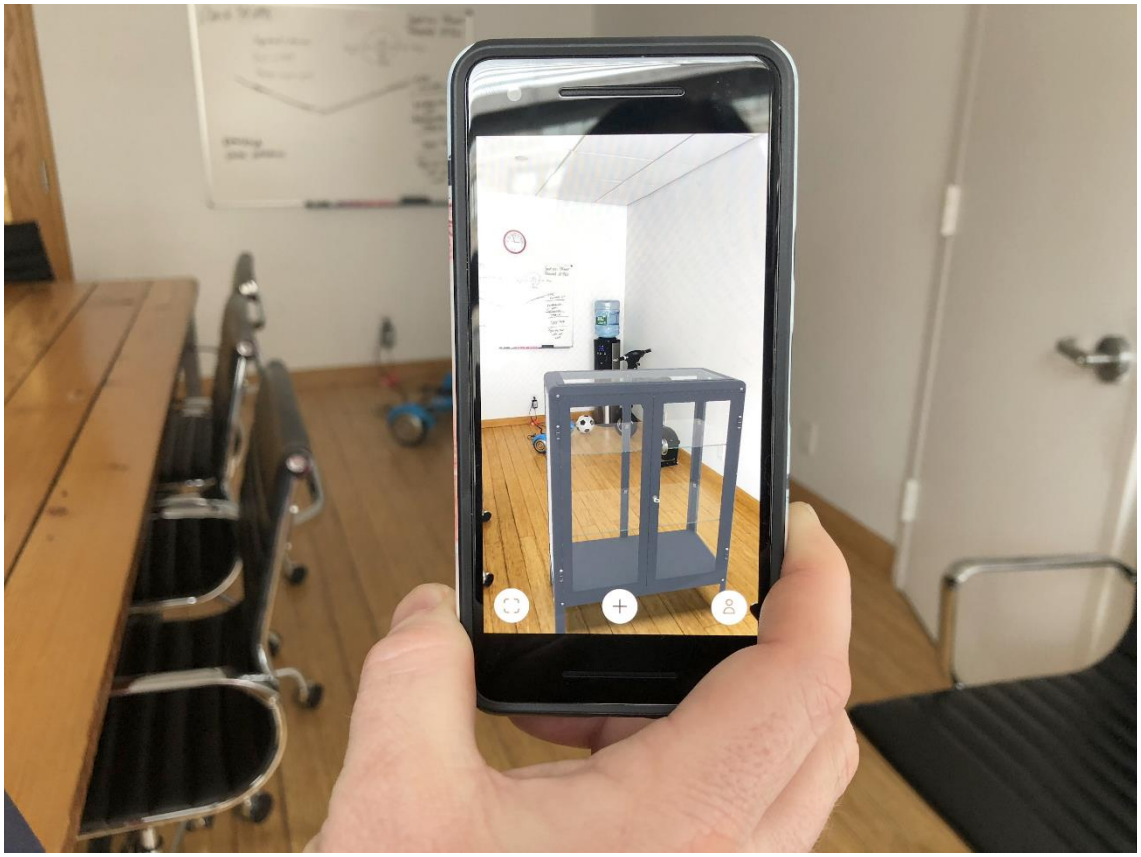


Рисунок 1.4 - Google ARCore

ARToolKit - це була перша широко доступна і відкрита бібліотека, орієнтована на AR. Для того, щоб створити сильну доповнену реальність, він використовує можливості відстеження відео, які обчислюють реальну позицію камери та орієнтацію щодо квадратних фізичних маркерів або природних маркерів об'єктів у реальному часі. Після того, як реальна позиція камери буде відома, віртуальна камера може бути розташована в одній точці, а моделі 3D

комп'ютерної графіки можуть бути намальовані, точно накладені на реальний маркер.

Одну з перших презентацій ARToolKit можна побачити на рисунку 1.5.

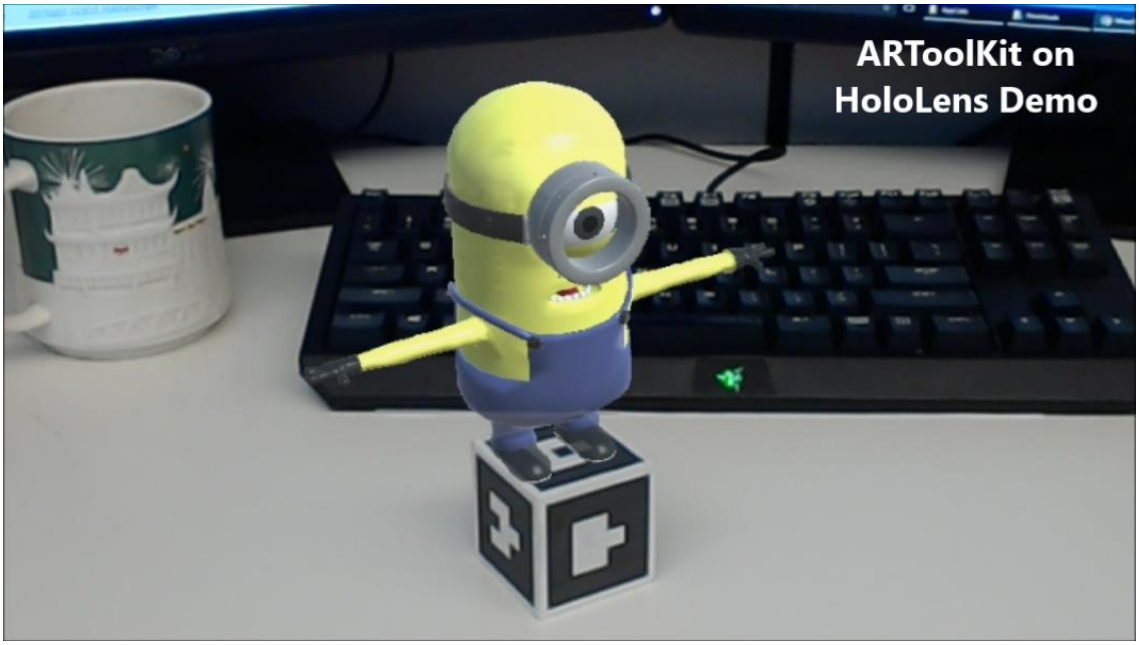


Рисунок 1.5 - ARToolKit

Таким чином, ARToolKit вирішує дві ключові проблеми в розширеній реальності: відстеження точок зору та взаємодії віртуальних об'єктів.

На жаль, при усій перспективності технології, вона застаріла та дуже важка у використанні порівняно з конкурентами.

Wikitude є постачальником мобільних додатків реальності (AR), що базується в Зальцбурзі, Австрія. Заснована у 2008 році, Wikitude спочатку зосереджувалася на забезпеченні досвіду розширеної реальності на основі розташування за допомогою програми Wikitude World Browser App. У 2012 році компанія реструктуризувала свою пропозицію, запустивши програму Wikitude SDK, основу розробки, що використовує технології розпізнавання та відстеження зображень, а також технології геолокації. Основний продукт компанії - Wikitude SDK. Вперше запущений у жовтні 2008 року, SDK включає в себе розпізнавання і відстеження зображень, рендеринг 3D-моделі, накладання відео, розташування на основі AR. У 2017 році Wikitude запустила

свою технологію SLAM (Simultaneous Localization And Mapping), яка дозволяє розпізнавання та відстеження об'єктів, а також миттєве відстеження без маркер. Крос-платформенний SDK доступний для операційних систем Android, iOS і Windows, оптимізований для декількох смарт-пристроїв для окулярів. Додаток Wikitude був першим загальнодоступним додатком, який використовував підхід, заснований на розташуванні, до розширеної реальності. Приклад інструментарія WikiTude можна побачити на рисунку 1.6.

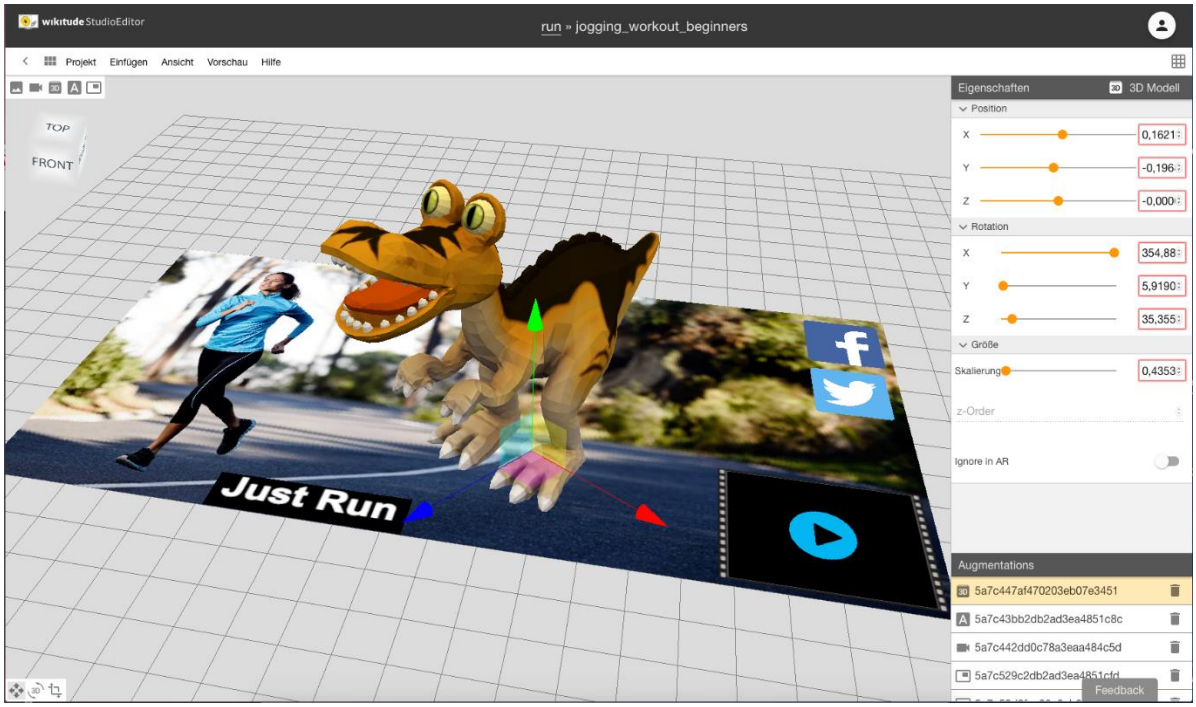


Рисунок 1.6 - WikiTude

Для даної роботи було обрано Apple ARKit як найбільш стабільний, доступний та простий інструмент розробки AR-застосуваннь.

Засіб розробки доповненої реальності ARKit. Принцип роботи ARKit як маркерного AR засобу дуже простий. Фактично ARKit це три фреймворка Apple в одному:

1. Vision - фреймворк обробки візуальної інформації
2. CoreML - фреймворк машинного навчання, що дозволяє визначати зображення та об’єкти як маркери.

3. CoreMotion -- фреймворк що відповідає за взаємодію з сенсорами пристрою, використовуються для трекінгу об’єктів

Схему роботи цих компонентів можна побачити на рисунку 1.7.

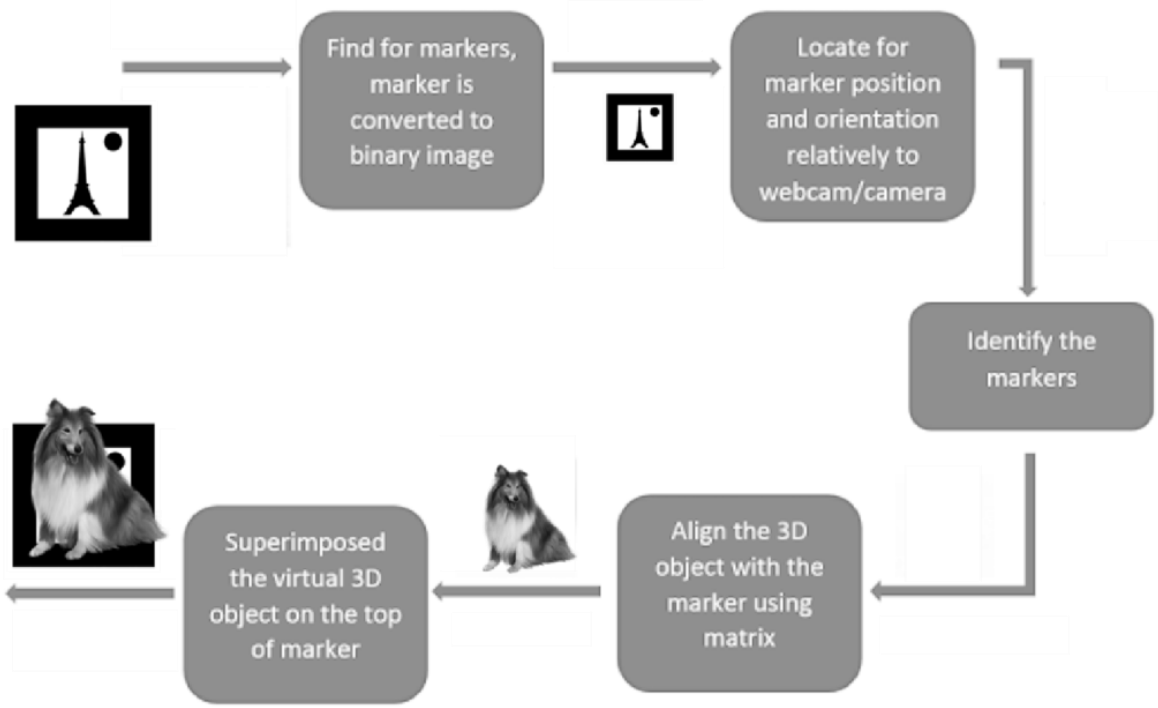


Рисунок 1.6 - Схема роботи ARKit

Маркером може бути що завгодно, друк 2D зображень, об'єкти реального світу, як автомобіль або людське обличчя, загальновідомий як ImageTarget. Після того, як додаток розширеної реальності розпізнає ціль за допомогою камери, він обробляє зображення і збільшує віртуальний вміст, як 3D-модель, відео, зображення або інші засоби масової інформації. Наприклад, ви можете побачити, що плакат фільму оживає і зіграти трейлер фільму. Поки ви дивитеся на плакат через «вікно» дисплея, ви можете бачити розширену реальність замість простої старої ванільної реальності. Використовуючи інтелектуальні алгоритми та інші датчики, такі як акселерометри та гіроскопи, пристрій може утримувати доповнені елементи у відповідності з зображенням реального світу. Використовуючи мобільний додаток, камера мобільного телефону визначає та інтерпретує маркер (ImageTarget). Програмне забезпечення аналізує маркер і створює віртуальне накладання зображення на

екрані мобільного телефону, прив'язане до положення камери. Це означає, що програма працює з камерою для інтерпретації кутів і відстані мобільного телефону від маркера.

1.5 Постановка задач дослідження

Об’єктом дослідження є дослідження засобів та методів обробки та проекціюнування web-контенту з використанням засобів доповненої реальності.

У світлі високого зросту можливостей нейронних мереж та полегшення доступу до машинного навчання протягом останніх років, було прийнято рішення досліджувати методи побудування доповненої реальності, у сфері доповненої реальності доволі гостро стоїть проблема постачання динамічного контенту, тому конкретизація теми зводиться саме до дослідження методів вирішення проблеми проекціюнування веб-контенту засобами доповненої реальності, також, враховуючи розповсюдження та доступність мобільних пристроїв, збільшення їх потужностей і покращення роботи їх сенсорів було обрано саме мобільні пристрої як цільову платформу.

Таким чином, для досягнення мети магістерської кваліфікаційної роботи необхідно запропонувати систему, що може вирішити цей комплекс задач, а саме буде здатна:

–тренувати на обраному датасеті та зберігати моделі доповненої реальності;

–конвертувати ці моделі у необхідний застосунку формат;

–постачати їх на мобільний застосунок;

–зберігати та передавати мобільному застосунку тривимірний, двовимірний або динамічний веб-контент у вигляді 3D моделей, зображень, відео, відеопотоку, анімацій, веб-сторінок;

–динамічно обробляти цей контент для подальшого використання у доповненій реальності;

–будувати модель відображення отриманого контента та передавати отриманий результат на компонент доповненої реальності для подальшого ну.

Процес передачі, генерації та проекціюнування контента має відбуватися в реальному часу у працюючому додатку (runtime).

Для вирішення задачі створення і постачання нейронних мереж необхідно налаштувати алгоритм тренування моделі нейронної мережі на віддаленому сервері із запропонованим датасетом, переведення цієї мережі у формат, що буде здатний працювати з мобільними технологіями доповненої реальності.

При вирішенні другої задачі варто враховувати те, що доповнена реальність оперує просторовими моделями, тому при отриманні веб-контенту і проекції його на реальний світ необхідно спочатку побудувати віртуальний об’єкт, на який цей контент буде проекціюнуватися, що враховує складну просторову геометрію розташування для будування цього об’єкта.

1.6 Висновки

В першому розділі було визначено актуальність роботи, виявлено проблему, яка полягає в тому, що для належного використання моделей нейронних мереж вони мають бути включені у бандл мобільного застосунку, що значно обмежує гнучкість розробки таких систем. Тому було вирішено дослідити цю проблему та запропонувати її вирішення шляхом динамічного постачання тренованих моделей до кінцевого користувача з можливістю подальшого використання у встановленому на пристрої користувача додатку.

Досліджено поняття доповненої, змішаної та віртуальної реальностей, розділяти поняття віртуальної, доповненої та змішаної реальностей, які вимагають особливого обладнання, такого як шоломи, датчики, джерело струму для повної емуляції реальності, доповнена та змішана ж реальності. Поставлено задачі магістерської кваліфікафійної роботи.

2 РОЗРОБКА МЕТОДУ І МОДЕЛЕЙ ДЛЯ ОПТИМІЗАЦІЇ НЕЙРОННИХ МЕРЕЖ ТА ПРОСТОРОВОГО СПРИЙНЯТТЯ AR

Для реалізації мети магістерської кваліфікаційної роботи необхідно оптимізувати нейронні менежі та просторове сприйняття AR.

2.1 Розробка математичної моделі нейронних мереж для розпізнавання образів

Для класифікації зображень використовумо згортуючу нейронну мережу, яку необхідно тренувати на запропонованому датасеті. Зазвичай це тривіальна задача, однак оскільки ця нейронная мережа буде використовуватися мобільним пристроєм, гостро стоїть проблема розміру цієї мережі. Для вирішення цієї проблеми необхідно врахувати математичну модель мережя для її оптимізації.

Найбільш трудомістким будівельним блоком CNN, кон'юнктурним шаром, є згортання 3D вхідних даних з серією 3D ядер. Складність конволюційного шару є квадратичною з трьома гіпер-параметрами: розміром ядра, кількістю каналів і просторовими вимірами. З іншого боку, баланс повинен бути обережним контролем серед них, щоб забезпечити доступність обчислень. На практиці широко використовуються 1 × 1 і 3 × 3 ядра, а збільшення каналів часто супроводжується зменшенням просторових розмірів [10].

Якщо розглядати шари нейронної мережі як матрицю, що складається з ядер як параметрів, то можна досягти зменшення розміру нейронної мережі шляхом зменшення кількості параметрів матриці. Цього можна досягти шляхом факторизації згортань, а саме зменшення кількості з'єднань / параметрів без зменшення ефективності мережі.

Розглянемо вхідні дані I в

R^{h*w*c},

де h, w, c висота, ширина і кількість каналів вхідних характеристичних карт, і згорткове ядро K у

$$R^{k*k*c*n},$$

де k - розмір згорткового ядра і n є число ядер.

Робота стандартного згортаного шару визначається за формулою

$$O \in R^{h*w*c} = K * I$$

представлена наступною формулою (2.1):

$$O(x, y, j) = \sum_{i=1}^{c} \sum_{u=1}^{k} \sum_{v=1}^{k} K(u, v, i, j) I(y + u - 1, x + v - 1, i) \quad (2.1)$$

Складність згорткового шару за кількістю множень, дорівнює

$$nck^2hw$$

Ілюстрація роботи згорткового шару зображена на рисунку 2.1.

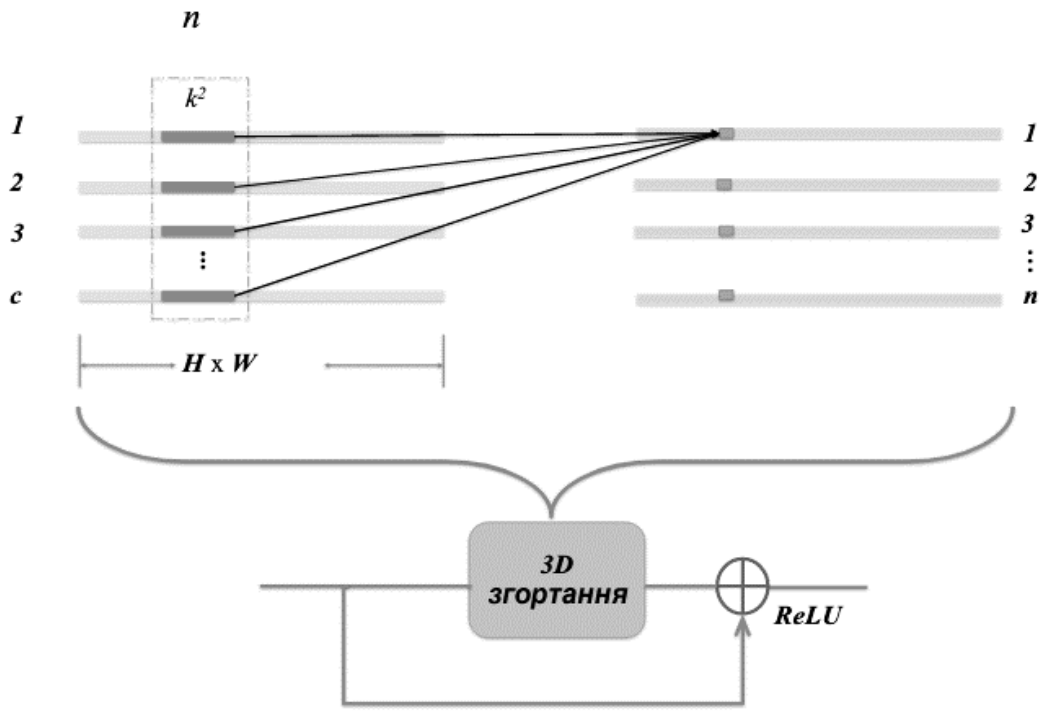


Рисунок 2.1 – Стандартний згортковий шар з залишковим з'єднанням

Оскільки складність квадратична з розміром ядра, в самих останніх моделях CNN, розмір ядра обмежується до 3 × 3 для контролю загального часу роботи.

У стандартних згорткових шарах вихідні особливості виробляються шляхом згортання групи 3D ядер з вхідними ознаками вздовж просторових вимірів. Таку операцію 3D-згортки можна розглядати як комбінацію 2D просторової згортки всередині кожного каналу (внутрішньоканальна згортка) і лінійної проекції через канали. Для кожного вихідного каналу на кожному вхідному каналі виконується просторова згортка. Просторова згортка здатна захоплювати локальну структурну інформацію, тоді як лінійна проекція перетворює простір ознак для вивчення необхідної нелінійності в шарах нейронів. Коли кількість вхідних і вихідних каналів велика, зазвичай сотні, такий 3D-згортковий шар вимагає величезної кількості обчислень [11]. Природна ідея полягає в тому, що 2D просторова згортка і проекція лінійних каналів можуть бути розформовані і виконані окремо.

4D згорткове ядро K може бути розкладено на основні фільтри

$$B \in R^{k*k*b*c}$$

і тензор лінійної проекції

$$P \in R^{b*c*n},$$

де b - число основ, так що (2.2):

$$K(u,v,i,j) = \sum_{t=1}^{b} B(u,v,t,i)P(t,i,j) \qquad (2.2)$$

Для кожного вхідного каналу згорткове ядро розкладається вздовж просторових вимірів. Операція згорткового шару в рівнянні 1 перетворюється в

$$J(y,x,t,i) = \sum_{u=1}^{k}\sum_{v=1}^{k} B(u,v,t,i)I(y+u-1,x+v-1,i)$$

$$O(x,y,j) = \sum_{i=1}^{c}\sum_{t=1}^{b} P(t,i,j)J(y,x,t,i),$$

де J є $R^{h*w*b*c}$. Кожен канал вхідної карти уявлення спочатку згортається за допомогою b 2D фільтрів (баз), генеруючи проміжну карту J, з якої число каналів становить b разів на вхід. Потім використовується лінійна проекція каналу для проекціюнування проміжної карти характеристик на виході. Вирівнювання цих двох операцій (2D просторова згортання і лінійна проекція) дає нам більшу свободу моделювання за допомогою налаштування кількості 2D баз b і розміру ядра k . Складність такого шару - bck2hw + bcnhw. Перший член відповідає 2D просторовій згортці, а другий - лінійній проекції. Як правило, k2 є набагато меншим, ніж n, і більшість обчислень споживається лінійною проекцією, яка є лінійною з числом базисного фільтра b. При b = k2 це еквівалентно повнорозмірному розкладанню згорткового ядра для кожного вхідного каналу вздовж просторових вимірів. Складність лінійної проекції така ж, як і стандартний згортковий шар; Завдяки присвоєнню b <k2 складність нижча, ніж стандартний згортковий шар у низькому порядку.

Розглядаючи використовану Inception v3 модель, процес факторизації одного з її модулей зображено на рисунку 2.2.

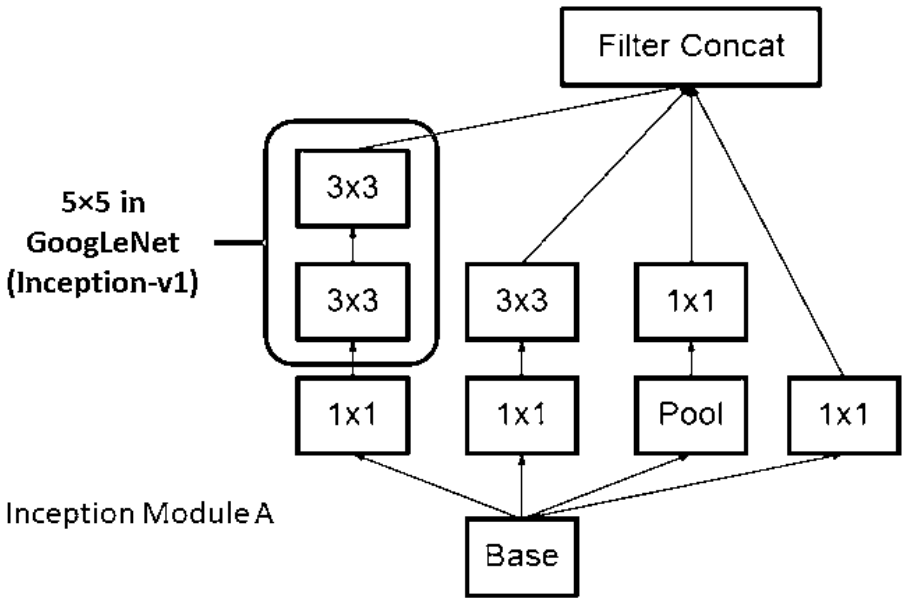


Рисунок 2.2 - Inception Module A з використанням факторизації

Ключове спостереження полягає в тому, що замість згортання b 2D фільтрів (баз) з кожним вхідним каналом одночасно і виконують над ними

лінійну проекцію, ми можемо послідовно організувати звивини, інтерліруючи з лінійною проекцією. Вищезгаданий згортковий шар з b 2D фільтрами може бути перетворений в каркас, який має b шари. У кожному шарі кожен вхідний канал спочатку згортається з одним 2D фільтром (основа), тоді лінійна проекція застосовується до всіх каналів. Таким чином, кількість каналів підтримується так само, як і вхідні дані по всіх слоях. Зауважимо, що така послідовна організація зберігає однакову обчислювальну складність, збільшуючи глибину та можливості навчання. Таким чином, складність кожного шару буде

$$ck^2hw + cnhw$$

Що значно пришвидшить роботу моделі та зменшить її розмір.

2.2 Розробка математичної моделі сприйняття навколишнього середовища методами доповненої реальності

При будуванні доповненої реальності головною задачей є сприйняття сенсорами пристрої оточення та будування його моделі на основі отриманих даних. Хоча у різних системах підходу можуть відрізнятися деталями, основний принцип залишається незмінним. Мета розширеної реальності полягає в тому, щоб забезпечити сприйняття комп'ютером простору з розумінням людини. У комп'ютерній науці простір просто метафора для загальноприйнятих і науково обгрунтованих концепцій простору, часу і матерії. Комп'ютерне розуміння простору - це не що інше, як математично визначене 3D представлення об'єктів, місцеположення та матерії [12]. Його можна просто зрозуміти за допомогою систем координат без необхідності заплутати жаргон, як гіпер-нерухомість або альтернативні всесвіти. Хоча це, безумовно, цікаві думки експериментів.

Те, що сприймається як розташування об'єктів у навколишньому середовищі, є реконструкція світлових моделей на сітківці. Візуальний простір

у комп'ютерній графіці можна визначити як сприйманий простір або візуальну сцену 3D-віртуального простору, що відчувається учасником.Віртуальний простір, в якому існує об'єкт, називається простором об'єкта. Це прямий аналог візуального простору.

Три типи різних математичних систем координат використовуються для компонування та програмування додатків віртуальної та розширеної реальності:

1. Декартові координати.
2. Сферичні полярні координати.
3. Циліндричні координати.

Декартові системи координат використовуються в основному за рахунок їх простоти, і більшість віртуальних просторів визначаються нею. Система координат на основі x-y-z є точною для визначення розташування 3D-об'єктів у віртуальному просторі. Три координатні площини перпендикулярні один одному. Відстані та місцеположення визначаються від точки виникнення, яка є точкою, де три площини перетинаються один з одним. Ця система використовується в основному для визначення візуальних координат 3D-об'єктів [13].

Декартова система визначає положення 3D-об'єктів часто по відношенню до початкової точки. Система розміщення сферичних полярних координат використовується для визначення об'єктів і особливостей по відношенню до положення користувачів. Ця система використовується в основному для відображення віртуального джерела звуку або відображення сферичного відео у випадку занурення VR першої особи. Сферична система координат заснована на перпендикулярних площинах, що розділяють сферу і складається з трьох елементів: азимут, висота і відстань. Азимут - це кут від початкової точки в горизонтальній / заземленій площині, а висота - кут у вертикальній площині. Відстань - це величина або діапазон від початку, що можна побачити на рисунку 2.3.

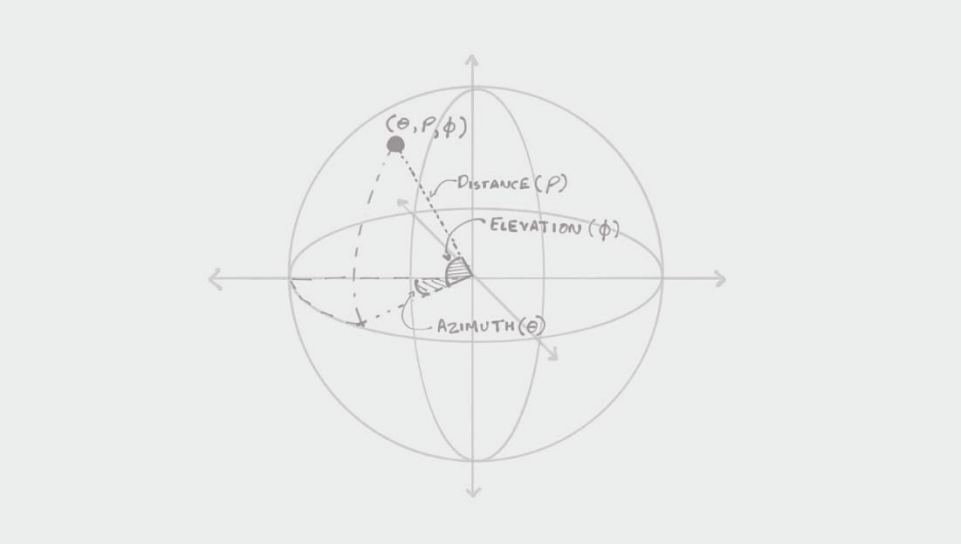


Рисунок 2.3 – Сферичні декартові координати

Циліндрична система система використовується в основному у програмах VR для перегляду панорам 360 градусів. Циліндрична система дає можливість точного відображення та вирівнювання нерухомих зображень для перекриття для зшивання країв у панорам. Система складається з центральної опорної осі (L) з точкою початку (O). Радіальна відстань (ρ) визначається з походження (O). Кутову координату (is) визначають для радіальної відстані (ρ) разом з висотою (z). Хоча ця система хороша для сценаріїв, які вимагають ротаційної симетрії, вона обмежена з точки зору її вертикального вигляду [13].

Полярні координати можна назвати радіальною відстанню або радіусом і кутом або азимутом, відповідно. Третю координату можна назвати висотою (якщо площина відліку горизонтальна). Лінію, перпендикулярну до площини відліку, яка проходить через центр координат можна назвати циліндричною віссю.

Далі зображення осей циліндричних координат на рисунку 2.4.

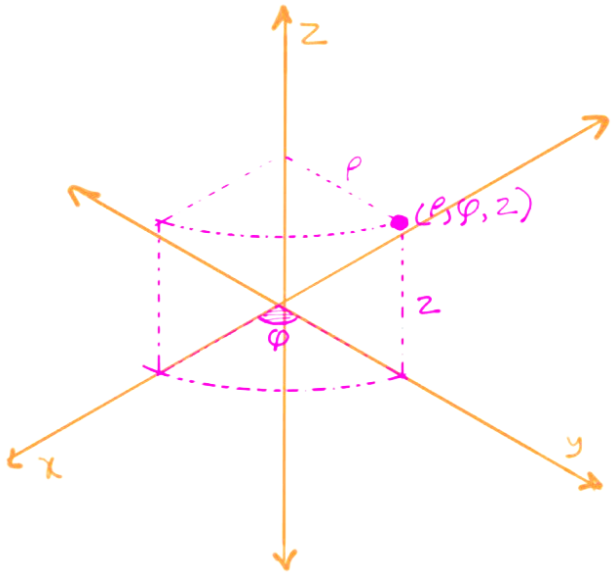


Рисунок 2.4 - Циліндрична система координат.

Необхідно визначити орієнтацію і обертання точок зору та об'єктів користувача разом з їх положенням у віртуальному просторі. Знання цієї інформації особливо важливо при відстеженні, де користувач шукає або знає орієнтацію віртуальних об'єктів щодо зорового простору.

Також у віртуальній і доповненій реальності загальноприйнято визначати орієнтацію і обертання з трьома незалежними значеннями. Вони називаються roll (x), pitch (y) і yaw (z) і відомі як кути Тейта-Баяна. Поєднання положення (x-y-z) і орієнтації (roll-pitch-yaw) називається шістьма осями свободи (6 DOF).

Схематичне зображення осей розташування на рисунку 2.5.

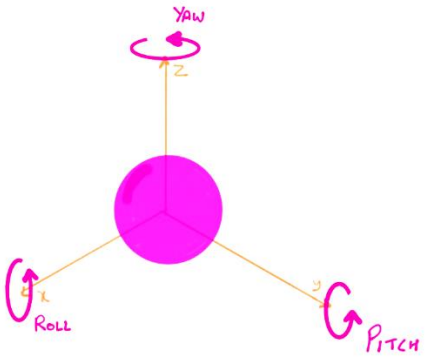


Рисунок 2.5 – Кути Тейта-Баяна

Однак цікавим питанням є сприйняття відстаней органами доповненої реальності, для цього необхідно провести обчислення у реальному світі, використовуючи сферичні координати.

Розрахуємо положення користувача:

Радіус Землі в екваторі становить 3 956,547 кілометрів. Позначимо його через s. s = 3,956,547 кілометрів.

Перетворення широт в кілометри буде простішим, оскільки можна наблизити широтні лінії до дуги кола. Відомо, як обчислити окружність кола (2πr), і також відомо, що коло має 360∘. Таким чином, якщо розподіляється окружність Землі на 360 градусів, можна отримати наближення до кілометрів на градус широти t. Точність цього наближення є достатньою для нашого застосування.

$$t = \frac{2\pi}{360}$$

Розрахунок кілометрів на градус довготи g буде не таким простим. Пересування на градус довжини поруч з екватором означає прогулянку на більшу відстань, ніж прогулянка на один градус довготи ближче до полюсів.

Таким чином, отримаємо що кожний градус довготи дорівнює

$$g = t\cos(a_y \frac{\pi}{180})$$

Де g це кілометри на ступінь довжини 180∘

Наш користувач стоїть біля:

$$a = \left[\frac{ax}{ay}\right] = [\frac{\text{Широта пристрою}}{\text{Довгота пристрою}}]$$

Користувач буде спрямовувати пристрій до напрямку, і цей напрямок буде робити кут α з істинним північчю (заголовком). Тепер цей кут α буде виміряний по лініях широти, які є вертикальними, і еквівалентні осі y, або будь-якій з його паралелей у декартовій площині. Однак у декартовій площині звичайною практикою є вимірювання кутів на осі x, тому кут α необхідно

перетворити на кут ψ, що дасть нам напрмок відносно горизонтальної лінії, паралельної осі x.

$$\psi = \frac{\pi}{2} - a$$

Оскільки тепер відмо, як перетворити з градусів на кілометри, і навпаки, можна обчислити точки b і c з урахуванням відстані r.

$$b = \begin{bmatrix} b_x \\ b_y \end{bmatrix} = \begin{bmatrix} \frac{r}{g}\cos\left(\psi + \frac{\phi}{2}\right) + a_x \\ \frac{r}{t}\sin\left(\psi + \frac{\phi}{2}\right) + a_y \end{bmatrix}$$

$$c = \begin{bmatrix} c_x \\ c_y \end{bmatrix} = \begin{bmatrix} \frac{r}{g}\cos\left(\psi - \frac{\phi}{2}\right) + a_x \\ \frac{r}{t}\sin\left(\psi - \frac{\phi}{2}\right) + a_y \end{bmatrix}$$

Враховуючи те, що відомий набір місць, кожне з яких має власну довготу і широту. Завдання полягає в тому, щоб визначити, які з них потраплятимуть в область інтересу, тому бути видимими і мати відповідну анотацію на екрані. Потрібно буде переглядати набір міток місця і для кожного місця a потрібно обчислити, чи координати довготи і широти знаходяться в межах області інтересу.

Маємо, що проекція вектора ap$^{\rightarrow}$ на вектор ab$^{\rightarrow}$ може бути обчислена за формулою (2.3):

$$\text{proj}_{\overrightarrow{ab}}\overrightarrow{ap} = \|\overrightarrow{ap}\|\cos\omega \tag{2.3}$$

Крім того, вектор між ap$^{\rightarrow}$ і ab$^{\rightarrow}$ є:

$$\overrightarrow{ap}\cdot\overrightarrow{ab} = \|\overrightarrow{ab}\|\|\overrightarrow{ap}\|\cos\omega$$

Тому отримано:

$$\text{proj}_{\overrightarrow{ab}}\overrightarrow{ap} = \frac{\overrightarrow{ap}\cdot\overrightarrow{ab}}{\|\overrightarrow{ab}\|}$$

Потрібно виразити $\text{proj}_{\overrightarrow{ab}}\overrightarrow{ap}$ як координату вектора ab, або іншими словами, обчислити власне значення для нього. Це досягається діленням його на норму ab. Назвемо цю власну величину λ.

$$\lambda = \frac{\text{proj}_{\overrightarrow{ab}}\overrightarrow{ap}}{\|\overrightarrow{ab}\|} \Rightarrow \lambda = \frac{\overrightarrow{ap} \cdot \overrightarrow{ab}}{\|\overrightarrow{ab}\|^2}$$

Тепер нам потрібно обчислити проекцію вектора ap → на вектор ac →:

$$\text{proj}_{\overrightarrow{ac}}\overrightarrow{ap} = \frac{\overrightarrow{ap} \cdot \overrightarrow{ac}}{\|\overrightarrow{ac}\|}$$

Розділення проекції на норму вектора ac→ дало нам власне значення і дозволяє виразити його координати в термінах ac →. Назвемо це власне значення σ.

$$\sigma = \frac{\text{proj}_{\overrightarrow{ac}}\overrightarrow{ap}}{\|\overrightarrow{ac}\|} \Rightarrow \sigma = \frac{\overrightarrow{ap} \cdot \overrightarrow{ac}}{\|\overrightarrow{ac}\|^2}$$

При λ і σ в руці можна перейти до останнього етапу визначення того, чи міститься мітка в межах видимої області, що представляє інтерес.

Дуга окружності, що визначає межу відстані, може бути обчислена за допомогою теореми Піфагора:

$$(\lambda\overrightarrow{ab})^2 + (\sigma\overrightarrow{ac})^2 = r^2$$

Однак довжина векторів ab→ і ac→ мають однакову довжину і дорівнюють довжині радіус r. Таким чином, для спрощення рівняння можна розділити обидві сторони на r2.

$$\lambda^2\frac{\overrightarrow{ab}^2}{r^2} + \sigma^2\frac{\overrightarrow{ac}^2}{r^2} = \frac{r^2}{r^2}$$

Залишаючи нас з:

$$\lambda^2 + \sigma^2 = 1$$

Тепер можна визначити, чи мітка місця p знаходиться в межах інтересу, тому є видимою. Власні значення λ і σ повинні бути позитивними числами (інакше позначка місця буде розташована за спостерігачем), а при об'єднанні, використовуючи теорему Піфагора, вони повинні бути меншими або рівними 1, тому вони не далі, ніж межа задається радіусом сегмента дуги. p (λ, σ) дорівнює:

$$\forall\ (\lambda > 0) \wedge (\sigma > 0) \wedge (\lambda^2 + \sigma^2 \leq 1)$$

для видимого спектра, та всі інші значення для невидимого.

Наступним кроком є використання отриманого результату у застосунку. Більш ранні методи AR спиралися на безліч датчиків на додаток до камери. Бібліотеки програмного забезпечення, такі як OpenCV, Vuforia, ARCore, ARKit, MRKit, дозволили обчислення цих значеннь на невеликих обчислювальних пристроях, таких як смартфон, з дивовижною точністю. Ці бібліотеки використовують алгоритми машинного навчання для розміщення 3D-об'єктів у навколишньому середовищі і вимагають введення лише цифрової камери. Економічність та простота цих алгоритмів з точки зору вимог датчиків значною мірою була відповідальною за наступне розповсюдження AR в останній час.

2.3 Розробка моделей системи доповненої реальності

Запропоновану модель системи доповненої реальності зображено на рисунку 2.6.

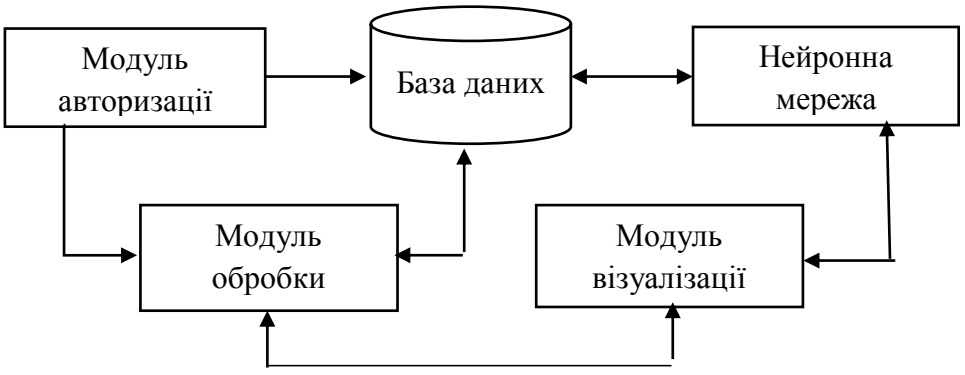


Рисунок 2.6 – Модель системи доповненої реальності

Система складається з модулів, а саме модуль авторизації, база даних, модуль обробки зображень, нейронна мережа доповненої рельності та модуль візуалізації.

Модуть авторизації користувачів, який працює за алгоритмом, наведеним на рисунку 2.7.

Рисунок 2.7 - Блок-схема модуля авторизації додатку

2.4 Розробка методу проекціонування веб-контенту з використанням доповненої реальності

Покроковий опис методу:

1. Заповнення бази даних тривимірними елементами. Користувач може власноруч заповнити базу даних зображеннями різних форматів. Додаток підтримує розширення .png, .jpg, JPEG та інші.

2. Опрацювання робочої карти. Робоча карта опрацьовується на вибраних фреймворках, а саме CoreML та Vision фреймворки.

3. Розрахунок математичної моделі за формулами. Математична модель використовує нейронні мережі для розпізнавання образів, а також для класифікації зображень використовується згортуюча нейронна мережа.

4. Розрахунок за створеної моделлю. Моделі розраховуються за формулами 2.1 -2.3.

5. Доповнння реальності з викристанням засобаму Turi Create. Turi Create фокусується на домені програми, що дозволяє побудувати архітектуру моделі та з'єднувальних шарів.

6. Доповнння реальності з викристанням засобаму reate ML

7. Використання моделі у застосунку засобами CoreML, цей фреймворк дозволяє визначати зображення та об’єкти як маркери.

2.5 Висновки

В другому розділі розроблено математичну моделі нейронних мереж для розпізнавання образів та для класифікації зображень, розроблено згортуючу нейронну мережу, яку необхідно тренувати на запропонованому датасеті. Для вирішення проблеми враховано розролену математичну модель. Розроблено модель системи доповненої реальності, блок-схема\у модуля авторизації додатку.

Розроблено метод проекціонування веб-контенту з використанням доповненої реальності. Описано формули розрахунків нейромережі, що буде використовуватись в програмному додатку.

3 РОЗРОБКА ПРОГРАМИ ДЛЯ ПРОЕКЦІОНУВАННЯ ДИНАМІЧНОГО ВЕБ-КОНТЕНТУ У ДОПОВНЕНІЙ РЕАЛЬНОСТІ

3.1 Варіантний аналіз і обгрунтування вибору середовища розробки

В якості основного середовища було обрано продукт Xcode, лінійку продуктів Apple, що включає інтегровану середу розробки програмного забезпечення та ряд інших інструментів. Ці продукти дозволяють розробляти як консольні програми, так і програми з графічним інтерфейсом, включаючи підтримку технології UIKit, ARKit, CoreML, Vision, для розробки програмного забезпечення для MacOS, iOS, iPadOS, watchOS і tvOS [14].

Основним додатком комплекту є інтегрована середовище розробки (IDE), також названа Xcode. Набір Xcode включає більшу частину документації розробника Apple, а також вбудований інтерфейс Builder - додаток, що використовується для побудови графічних інтерфейсів користувача. До Xcode 4.1, пакет Xcode містив модифіковану версію GNU Compiler Collection. В Xcode 3.1 до Xcode 4.6.3, він включав компілятор LLVM-GCC, з передніми кінцями з колекції компіляторів GNU і генератора коду на основі LLVM. У Xcode 3.2 і пізніших версіях він включав компілятор Clang C / C ++ / Objective-C, з новими письмовими кінцями і генератором коду на основі LLVM, а також статичним аналізатором Clang. Починаючи з Xcode 4.2, компілятор Clang став типовим Компілятор, починаючи з Xcode 5.0, Clang був єдиним наданим компілятором.

Всі інструменти встановлені на macOS, це серія графічних операційних систем, розроблених і реалізованих компанією Apple Inc. з 2001 року. Це основна операційна система для комп'ютерів Apple Mac. На ринку настільних комп'ютерів, ноутбуків і домашніх комп'ютерів, а також за допомогою веб-застосувань, він є другим найбільш широко використовуваним настільним ОС після Microsoft Windows.

macOS - друга велика серія операційних систем Macintosh. Перший розмовно називається "класичний" Mac OS, який був введений в 1984 році, і

остаточний випуск якого був Mac OS 9 в 1999 році. Перша десктопна версія, Mac OS X 10.0, була випущена в березні 2001 року, з її першим. оновлення, 10.1, прибуває пізніше цього року. Після цього Apple почала називати свої випуски після великих кішок, які тривали до OS X 10.8 Mountain Lion. Починаючи з OS X 10.9 Mavericks, релізи отримали назву за місцем розташування в Каліфорнії. Apple скоротила назву на "OS X" у 2012 році, а потім змінила її на "macOS" у 2016 році, прийнявши номенклатуру, яку вони використовували для своїх інших операційних систем, iOS, watchOS і tvOS. Остання версія macOS Mojave, яка була публічно випущена у вересні 2018 року та є найбільш продвинутою операційною системою для розробки AR застосунків [15].

У період з 1999 по 2009 рік Apple продала окрему серію операційних систем Mac OS X Server. Початкова версія, Mac OS X Server 1.0, була випущена в 1999 році з інтерфейсом користувача, схожим на Mac OS 8.5. Після цього нові версії були введені одночасно з настільною версією Mac OS X. Починаючи з Mac OS X 10.7 Lion, функції сервера були доступні як окремий пакет на Mac App Store, на Mac OS X server є можливість розробляти веб-застосунки з використанням мови програмування Swift.

iOS (раніше iPhone OS) - це мобільна операційна система, створена і розроблена компанією Apple Inc. виключно для апаратного забезпечення. Це операційна система, яка в даний час керує багатьма мобільними пристроями компанії, включаючи iPhone, iPad і iPod Touch. Це друга за популярністю мобільна операційна система в усьому світі після Android.

Спочатку презентований у 2007 році для iPhone, iOS був розширений на підтримку інших пристроїв Apple, таких як iPod Touch (вересень 2007 року) і iPad (січень 2010 року). Станом на березень 2018 року, Apple App Store містить більше 2,1 мільйона iOS-додатків, з яких 1 мільйон є рідними для iPad. Ці мобільні програми колективно завантажуються більше 130 мільярдів разів.

Інтерфейс користувача iOS заснований на безпосередній маніпуляції, використовуючи жести з мультитач. Елементи управління інтерфейсом

складаються з повзунків, перемикачів і кнопок. Взаємодія з операційною системою включає в себе жести, такі як удари, натискання, затискання та зворотний поштовх, кожна з яких має певні визначення в контексті операційної системи iOS та її мультитач-інтерфейсу. Внутрішні акселерометри використовуються деякими додатками для реагування на струшування пристрою (один загальний результат - команда скасування) або обертання в трьох вимірах (один загальний результат - перемикання між портретом і пейзажем). Компанія Apple отримала високу оцінку за включення до iOS повноцінних функцій доступності, що дозволяє користувачам з вадами зору та слуху правильно використовувати її продукти [16].

Основні версії iOS випускаються щорічно. Поточна версія, iOS 12, була випущена 17 вересня 2018 року. Вона доступна для всіх пристроїв iOS з 64-розрядними процесорами; iPhone 5S і пізніші моделі iPhone, iPad (2017), iPad Air, а потім моделі Air Air, всі моделі iPad Pro, iPad Mini 2 і пізніші моделі iPad Mini, а також IPod Touch шостого покоління. На всіх останніх iOS-пристроях iOS регулярно перевіряє наявність оновлення, а якщо є, підкаже користувачеві дозволити автоматичну інсталяцію.

Прийшовши восени 2019 року, iOS 13 буде випущений для громадськості, представляючи незначні налаштування інтерфейсу користувача, а також багато нових функцій, таких як оновлений додаток Reminders, клавіатура swipe, розширене додаток Photos і новий темний режим. iOS 13 завершиться підтримкою декількох пристроїв iOS, включаючи iPhone 5s, iPod Touch (6-е покоління), iPhone 6 і iPhone 6 Plus, які все ще складають понад 10% всіх пристроїв iOS.

Випуск iOS 13 буде виключно для iPhone і iPod touch, оскільки iPad буде відгалужуватися в нову операційну систему на базі iOS під назвою iPadOS. Крім того, iPadOS також зменшить підтримку iPad, які мають менше 2 Гб оперативної пам'яті, такі як iPad Air, iPad mini 2 і iPad mini 3. Підтримку ARKit було введено з iOS 11, ARKit 2 було випущено на поточну версію iOS 12, наступна версія iOS 13, матиме підтримку ARKit 3.0, найбільш розвинену SDK

для розробки AR застосунків [17]. Отже, для розробки було обрано технологію Xcode.

3.2 Варіантний аналіз і обгрунтування вибору мови програмування і шаблону проектування

Пакети для кодування нейронних мереж існують у більшості популярних мов програмування, включаючи Matlab, Octave, C, C ++, C #, Ruby, Perl, Java, Javascript, і PHP, однак найбільш поширеною мовою для розробки комплексних нейронних мереж є мова програмування Python.

Python - це мова програмування високого рівня, призначена для читаності коду та ефективного синтаксису, що дозволяє виражати поняття в меншій кількості рядків коду, ніж мови C ++ або Java. Двома бібліотеками Python, що мають особливе значення для створення нейронних мереж, є NumPy і Theano. NumPy - це пакет Python, який містить безліч інструментів для наукових обчислень, включаючи об'єкт N-масиву, функції мовлення, лінійну алгебру та можливості випадкових чисел. NumPy пропонує ефективний багатовимірний контейнер з загальними даними з довільним визначенням типів даних, що дозволяє легко інтегруватися з широким спектром баз даних. Функції NumPy слугують фундаментальними будівельними блоками для наукових обчислень, і багато з його особливостей є невід'ємною частиною розробки нейронних мереж у Python.

NumPy - це бібліотечний пакет для мови програмування Python, який можна використовувати для розробки нейронних мереж, серед інших наукових обчислювальних завдань. Theano - бібліотека Python, яка визначає, оптимізує і оцінює математичні вирази, інтегрується з NumPy [18]. Він має модульні тестування та функції самостійної перевірки для виявлення помилок; ефективна символічна диференціація; динамічне генерування коду C для швидкої оцінки експресії; і здатний використовувати блок обробки графіки (GPU) прозоро для дуже швидких обчислень, які інтенсивно використовують

дані. Він може моделювати обчислення у вигляді графіків і використовувати ланцюгове правило для обчислення складних градієнтів. Це особливо застосовне до нейронних мереж, оскільки вони легко виражаються у вигляді графіків обчислень. Використовуючи бібліотеку Theano, нейронні мережі можуть бути написані більш стисло і виконуватися набагато швидше, особливо якщо використовується GPU.

У екосистемі Apple використовуються дві мови програмування: більш старий і традиційний Objective-C, та Swift.

Objective-C - це універсальна, об'єктно-орієнтована мова програмування, яка додає повідомлення в стилі Smalltalk мовою програмування C.

Це була основна мова програмування, підтримувана Apple для операційних систем macOS, iOS і iPadOS, та їх відповідних інтерфейсів прикладного програмування (API) Cocoa і Cocoa Touch до введення Swift. Мова програмування Objective-C була розроблена на початку 1980-х років. Він був обраний як основна мова, яку використовує NeXT для своєї операційної системи NeXTSTEP, з якої виведені macOS і iOS. Портативні програми Objective-C, які не використовують бібліотеки Cocoa або Cocoa Touch, або ті, які використовують частини, які можуть бути перенесені або перевиконані для інших систем, також можуть бути зібрані для будь-якої системи, що підтримується GNU Compiler Collection (GCC) або Clang. Файли програмної реалізації Objective-C, як правило, мають розширення .m filename, тоді як файли "header / interface" Objective-C мають розширення .h, так само як і C-файли заголовків.

Оскільки Objective-C було виведено з C, було випущено Objective-C++ для сумісності з C++ відповідно. Файли Objective-C ++ позначаються розширенням .mm.Objective-C ++Objective-C ++ - це мовний варіант, прийнятий інтерфейсом до GNU Compiler Collection і Clang, який може компілювати вихідні файли, які використовують комбінацію синтаксису C ++ і Objective-C. Objective-C ++ додає до C ++ розширення, які додає Objective-C до C. Оскільки нічого не робиться для уніфікації семантики за різними

функціями мови, застосовуються певні обмеження: Клас C ++ не може бути отриманий з класу Objective-C і навпаки.

Однак Objective-C є потужним інструментом для використання більш низькорівневих API написаних на C та C++, наприклад OpenCV, що є дуже потужним інструментом Computer Vision.

Однак основною мовою розробки AR застосунку було обрано Swift як найбільш сучасну та потужну.

Swift - це універсальна, складна мова програмування, розроблена компанією Apple для iOS, macOS, watchOS, tvOS, і Linux. Swift розроблений для роботи з Cocoa та Cocoa Touch продуктів Apple. Вона побудована з відкритим вихідним кодом LLVM компілятора і була включена в Xcode з версії 6, випущеної в 2014 році. На платформах Apple вона використовує бібліотеку виконання Objective-C, яка дозволяє запускати код C, Objective-C, C ++ і Swift. в межах однієї програми. Apple призначила Swift підтримати багато основних концепцій, пов'язаних з Objective-C, особливо динамічну диспетчеризацію, широко розповсюджене пізнє зв'язування, розширюване програмування та подібні функції, але "безпечнішим" способом, що полегшує уловлювання програмних помилок; Swift має функції, що стосуються деяких поширених помилок, таких як нульовий покажчик [19].

Таким чином Swift можна використовувати разом із Objective-C, що в свою чергу дає можливість користуватися Objective-C++ та C++ кодом, що є необхідним для повноцінної розробки комплексної AR системи з використанням Computer Vision, а також повноцінну розробку web-сервера для зберігання та розповсюдження тренованих моделей та контенту.

Зазвичай для розробки мобільних застосунків використовуються архітектури сімейства MVX, де найбільш розповсюдженим є MVC (model-view-controller), або більш продвинутий MVP (model-view-presenter), однак ці архітектури занадто прості для комплексної задачі адаптації AR даних, тому у мобільному компоненті системи було вирішено використати архітектурний шаблон Model-view-viewmodel (MVVM) як найбільш відповідну для

розділення відображення (View) у вигляді ARKit сутностей, сховища моделей нейронних мереж та веб-контенту (Model) та адаптера, що переводить ці моделі у сприйнятливий для відображення вид (ViewModel).

MVVM полегшує поділ розробки графічного інтерфейсу користувача - мовою розмітки або кодом GUI - від розробки бізнес-логіки або локальної логіки (дані моделі), у нашому випадку – розділ бізнес-логіки AR компонентів від користувацького інтерфейсу AR та від даних AR у їх звичайному вигляді.

Схему роботи MVVM зображено на рисунку 3.1.

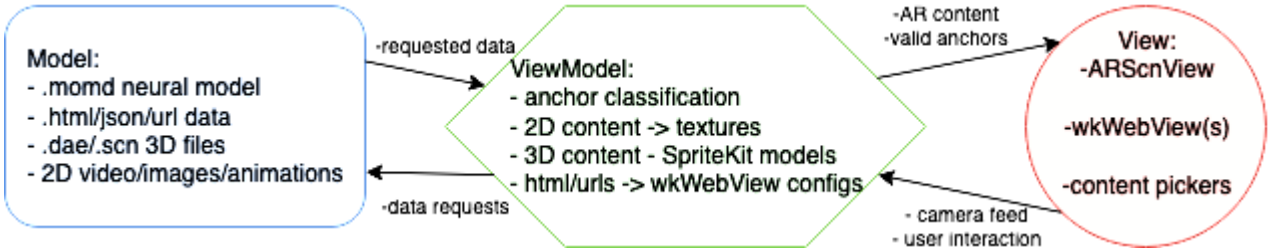


Рисунок 3.1 - MVVM схема AR додатку

Модель перегляду MVVM є перетворювачем значень, тобто модель перегляду відповідає за викриття (перетворення) об'єктів даних з моделі таким чином, що об'єкти легко керуються і подаються. У цьому відношенні модель перегляду є більш моделлю, ніж перегляд, і обробляє більшість, якщо не всю, логіку відображення перегляду. Модель перегляду може реалізовувати шаблон посередника, організовуючи доступ до локальної логіки навколо набору випадків використання, що підтримується видом. \ TMVVM є варіацією шаблону дизайну моделі презентації Мартіна Фаулера.

MVVM аналогічно описує стан і поведінку перегляду, але модель презентації абстрагує уявлення (створює модель перегляду) таким чином, що не залежить від конкретної платформи інтерфейсу користувача. \ TMVVM був винайдений архітекторами Майкрософт Кеном Купером і Тедом Петерсом для спрощення програмування користувальницьких інтерфейсів за допомогою подій. Шаблон був включений у Windows Presentation Foundation (WPF)

(Microsoft. NET графічної системи) і Silverlight (WPF в інтернет-додатків похідних). Джон Госсман (John Gossman), один з архітекторів Microsoft WPF і Silverlight, оголосив MVVM у своєму блозі в 2005 році. Модель-view-viewmodel також називається модель-view-binder, особливо в реалізаціях, які не залучають платформу .NET. ZK (фреймворк веб-додатків, написаний на Java) і KnockoutJS (бібліотека JavaScript) використовують модель-view-binder.

3.3 Програмна реалізація AR застосунку

ARKit стек приймає контент у вигляді 2D або 3D об’єктів та проекціює його на точки (anchors), знайдені методами комп’ютерного зору та оброблені нейронною мережею;

SpriteKit та SceneKit дозволяють створювати 2- та 3-вимірні моделі на основі динамічно отриманого контенту та будування просторових моделей на основі точок знайдених ARKit;

CoreML фреймворк дозволяє використовувати моделі, отримані ззовні для класифікації оточення, отриманого з Vision;

Vision фреймворк дозволяє використовувати сенсори мобільного пристрою як інструменти машинного зору та введення оточення для класифікації нейронною мережею, OpenCV додатково покращує результати роботи Vision;

Create ML утиліта дає можливість створювати прості класифікатори для подальшого використання CoreML або конвертувати складні існуючі в необхідний формат [20].

Як було вказано в останньому пункті, CreateML при розробці CoreML не може створювати складні моделі, а лише базові класифікатори зображень, звуку та тексту.

Тому для складних комплексних ситуацій необхідно використати додатковий алгоритм генерації складних моделей:

- фреймворк для навчання моделі (TuriCreate/TensorFlow);
- платформа для запуску цієї основи для навчальних цілей (Google Colab);

- програми для розмітки фотографій (RectLabel);
- скрипти Python, які будуть використовуватися для перетворення цих анотацій у формат, необхідний фреймворкам.
- скрипт Python для оптимізації отриманої нейронної мережі

Таким чином, задача динамічного постачання моделей та контенту до кінцевого застосунку зводиться до алгоритму тренування та адаптації нейронної мережі у необхідний формат та перебудування контенту відповідно.

Алгоритм роботи наведеної системи враховує складні випадки та є досить гнучким для переважної більшості випадків [21].

Рисунок 3.2 зображає алгоритм роботи системи:

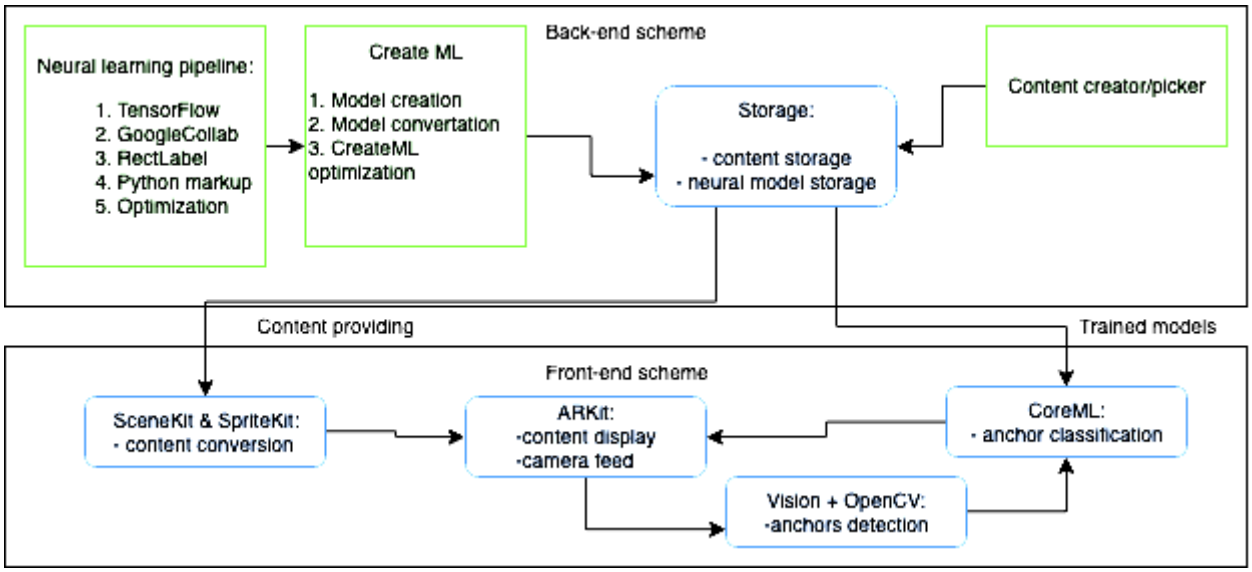


Рисунок 3.2 – Загальна схема системи генерації та проекціонування динамічного веб-контенту.

Таким чином, є чіткий розподіл на генерацію, перетворення та відображення контенту.

ARKit використовує візуальну інерційну одометрію (VIO) для відстеження руху пристрою та навколишнього світу. VIO запобігає входженню на основі AVFoundation з датчика камери з даними руху пристрою, знятими за допомогою CoreMotion.

Клас ARViewController керує сеансом AR і відображає вміст AR накладання у вигляді SpriteKit. ARKit захоплює відеокадри з камери і подає їх до контролера подання в методі session (_: didUpdate :), який потім викликає метод classifyCurrentImage () для запуску класифікатора зображень Vision.

Далі наведено фрагмент коду ініціалізації AR стеку.

```
func session(_ session: ARSession, didUpdate frame: ARFrame) {
          guard currentBuffer == nil, case .normal =
frame.camera.trackingState else {
     return
     }

     self.currentBuffer = frame.capturedImage
     classifyCurrentImage()}
```

Метод classifyCurrentImage () використовує властивість currentBuffer контролера перегляду, щоб відстежувати, чи Vision зараз обробляє зображення, перш ніж починати інше завдання Vision.

Більшість завдань комп'ютерного зору не є агностичними для обертання, тому важливо передати орієнтацію зображення відносно пристрою.

Фрагмент коду з’ясування орієнтації зображення.

```
let orientation =
CGImagePropertyOrientation(UIDevice.current.orientation)

let requestHandler = VNImageRequestHandler(cvPixelBuffer:
currentBuffer!, orientation: orientation)
visionQueue.async {
     do {
               defer { self.currentBuffer = nil }
     try requestHandler.perform([self.classificationRequest])
     } catch {
     print("Error: Vision request failed with error \"\(error)\"")
     }
}
```

Обмежено обробку на один буфер одночасно для виконання. Камера переробляє кінцевий пул буферів пікселів, тому збереження занадто великої кількості буферів для обробки може призвести до уповільнення камери та

вимкнення сеансу зйомки. Передача декількох буферів Vision для обробки призведе до уповільнення обробки кожного зображення, додавання затримки і зменшення кількості процесорів і накладних витрат GPU для візуалізації AR.

Крім того, додаток дозволяє налаштування usesCPUOnly для свого запиту Vision, звільняючи GPU для використання у візуалізації.

Метод processClassifications (для: error :) зберігає мітку результату найкращого відповідності, створену класифікатором зображення, і відображає її в кутку екрана. Потім користувач може натиснути на сцену AR, щоб помістити цю позначку в реальне положення. Розміщення мітки вимагає двох основних кроків.

По-перше, розпізнавач жестів торкання спрацьовує дію placeLabelAtLocation (відправника :). Цей метод використовує метод ARKit hitTest (_: types :), щоб оцінити позицію 3D у реальному світі, що відповідає крану, і додає прив'язку до AR сесії в цій позиції.

Фрагмент коду прив’язки об’єкта до маркера.

```
@IBAction func placeLabelAtLocation(sender: UITapGestureRecognizer) {
      let hitLocationInView = sender.location(in: sceneView)
      let hitTestResults = sceneView.hitTest(hitLocationInView, types:
[.featurePoint, .estimatedHorizontalPlane])
      if let result = hitTestResults.first {

                  let anchor = ARAnchor(transform:
result.worldTransform)
      sceneView.session.add(anchor: anchor)

                  anchorLabels[anchor.identifier] = identifierString
      }
}
```

Далі, після того, як ARKit автоматично створить вузол SpriteKit для доданого якоря, метод перегляду (_: didAdd: for :) надає вміст для цього вузла. У цьому випадку клас ARLabelNode створює стилізовану текстову мітку, використовуючи рядок, наданий класифікатором зображень.

Фрагмент коду створення ще одного об’єкта і додавання до маркера.

```
func view(_ view: ARSKView, didAdd node: SKNode, for anchor: ARAnchor)
{
      guard let labelText = anchorLabels[anchor.identifier] else {
      fatalError("missing expected associated label for anchor")
      }
      let label = TemplateLabelNode(text: labelText)
      node.addChild(label)
}
```

Таким чином будуємо AR об’єкти на розпізнаних маркерах.

Блок-схема роботи додатку зображена на рисунку 3.3.

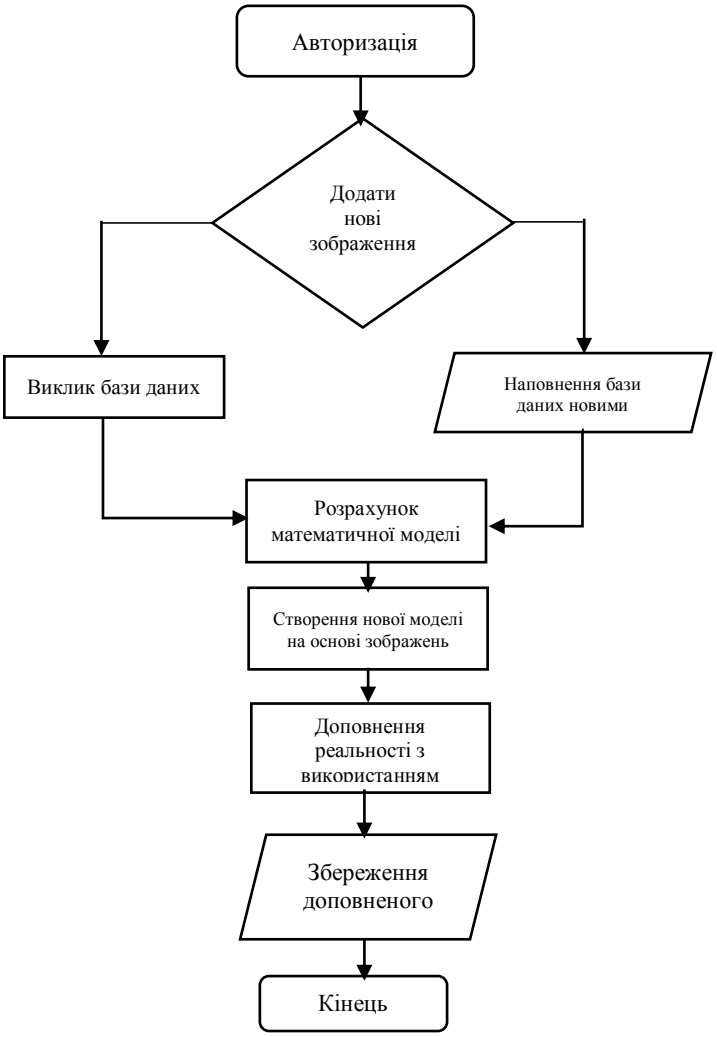


Рисунок 3.3 – Блок-схема роботи додатку

3.4 Тренування моделі засобами Turi Create

Turi спочатку був стартапом в Сіетлі, відомим своєю продукцією GraphLab, що спрощує загальні завдання машинного навчання. У 2016 році

Apple придбала Turi, а пізніше в грудні 2017 року зробила Turi Create проектом з відкритим кодом.

Головним плюсом Turi є те, що система спрощує розробку власних моделей машинного навчання, щоб надавати можливість додавати рекомендації, виявлення об'єктів, класифікацію зображень, подібність зображень або класифікацію діяльності.

Також вона проста у використанні, візуальна, гнучка і експортується в сумісні з CoreML моделі. Отже, створені моделі можна використовувати відразу в iOS, macOS, tvOS і додатках watchOS без додаткового перетворення.

Turi Create фокусується на домені програми, а не на домені моделі. Це означає, що нам не потрібно турбуватися про побудову архітектури моделі та з'єднувальних шарів, а справа полягає з API високого рівня для виконання завдань, таких як завантаження зображень, навчання та оцінювання. Це також означає, що не потрібно використовувати фахівців з машинного навчання.

Turi Create автоматично використовує графічні процесори Mac для виконання таких завдань:

• Класифікація зображень (macOS 10.13+)

• Подібність зображення (macOS 10.13+)

• Виявлення об’єктів (macOS 10.14+, лише дискретний GPU)

• Класифікація активності (macOS 10.14+, лише дискретні GPU)

Наступний фрагмент коду демонструє процес тренування моделі за допомогою Python скрипта.

```
import turicreate as tc
import os
data = tc.image_analysis.load_images('dataset', with_path=True)
data['hero_name'] = data['path'].apply(lambda path:
os.path.basename(os.path.dirname(path)))
data.save('turi.sframe')
data.explore()
data = tc.SFrame('turi.sframe')
train_data, test_data = data.random_split(0.8)
```

```
    model = tc.image_classifier.create(train_data,
target='hero_name')
    predictions = model.predict(test_data)
    metrics = model.evaluate(test_data)
    print(metrics['accuracy'])
    model.save('turi.model')
    model.export_coreml('model/TuriCreate.mlmodel')
```

На виході отримали треновану модель.

3.5 Тренування моделі та конвертація засобами Create ML

У 2018 році Apple представила Core ML: швидкий спосіб імпортувати попередньо навчені моделі машинного навчання в додаток з якомога меншим кодом! У цьому році, за допомогою програми Create ML, Apple надає розробникам можливість створювати власні моделі машинного навчання прямо в Xcode Playgrounds! Нам потрібен лише певний обсяг даних. На даний момент Create ML дозволяє переносити текст, зображення та таблиці як дані. Однак, оскільки це являє собою більшість додатків ML, це повинно слугувати цілі добре. На рисунку 3.4 зображено принцип роботи Create ML.

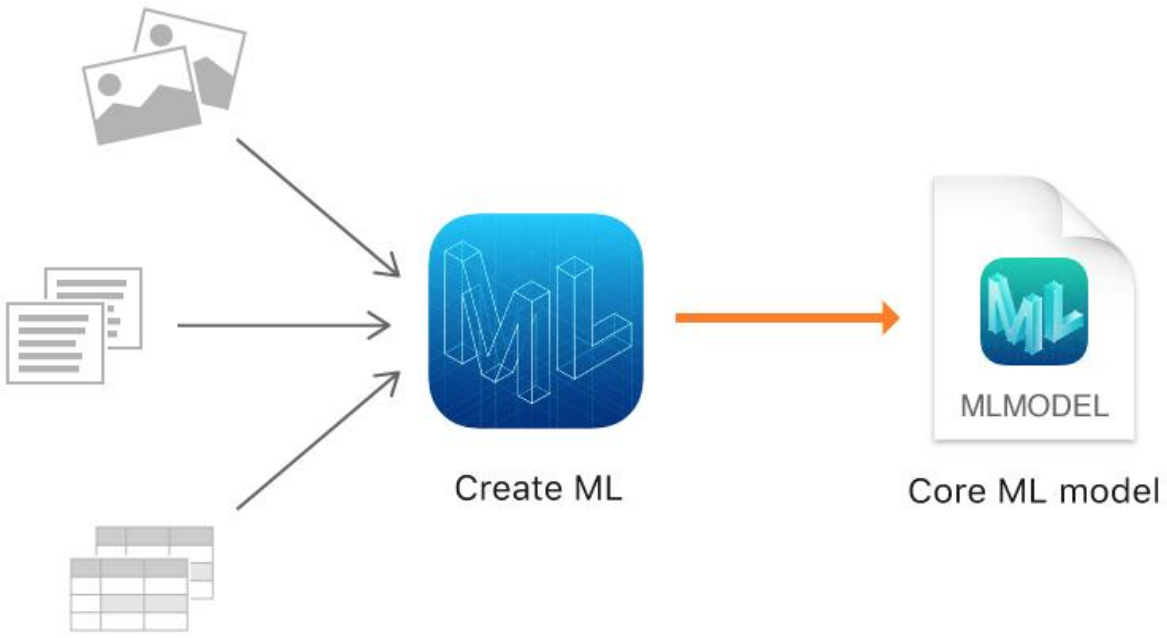


Рисунок 3.4 – Принцип роботи Create ML.

Модель є результатом застосування алгоритму машинного навчання до набору навчальних даних. Можна використовувати модель для прогнозування на основі нових вхідних даних. Наприклад, модель, яка пройшла підготовку з історичних цін на житло в регіоні, може спрогнозувати ціну будинку, враховуючи кількість спалень і ванних кімнат.

Класифікатор зображень - це модель машинного навчання, яка розпізнає зображення. Коли надаєте йому зображення, він відповідає міткою для цього зображення.

Діаграма, що показує, як класифікатор зображення передбачає позначку "Жираф" із зображення жирафа.Ви тренуєте класифікатор зображень, показуючи йому багато прикладів зображень, які ви вже позначили. Наприклад, можна навчити класифікатор зображень, щоб розпізнати тварин сафарі, показуючи йому різноманітні фотографії слонів, жирафів, левів і так далі.

На рисунку 3.5 зображено процес конвертації моделі у робочий формат.

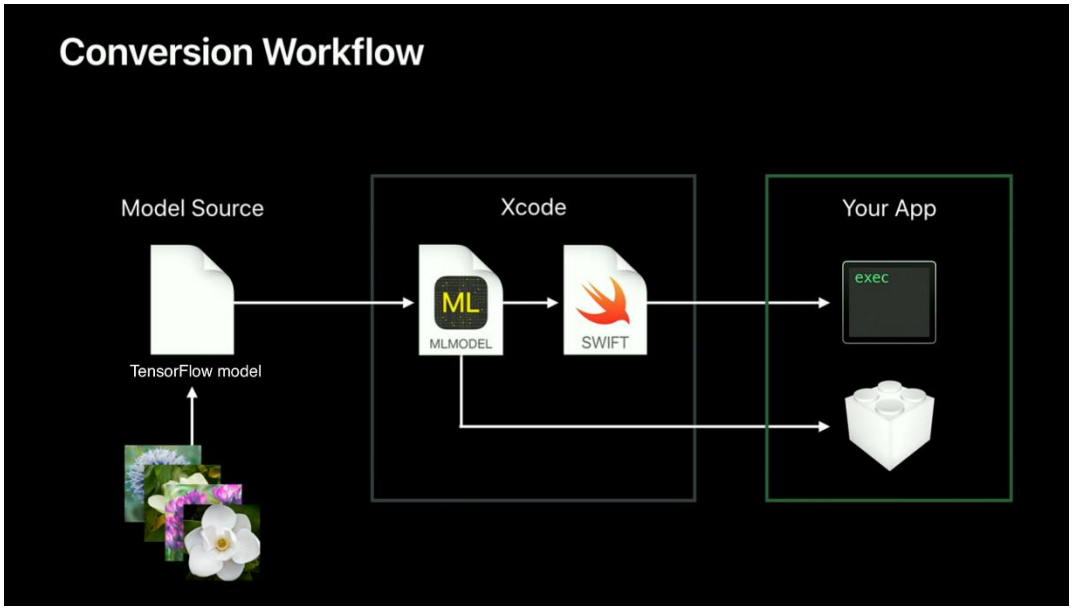


Рисунок 3.5 – Конвертація моделі TensorFlow у .mlmodel формат.

Остаточна модель міститься у файлі .mlmodel. Це новий відкритий формат файлів, який описує шари у вашій моделі, входи та виходи, мітки класів і будь-яку попередню обробку, яку необхідно виконати на даних. Вона також містить всі вивчені параметри (ваги та упередження). Потім ми можемо просто завантажити .mlmodel в Xcode і почати робити прогнози.

Щоб полегшити ситуацію, припустимо, що у нас вже є джерело моделі, яке навчається з даними. Існують тонни цієї моделі в Інтернеті. Ми можемо зосередитися на частині перетворення між джерелом моделі та MLModel.

Наразі для конверсії підтримуються наступні формати:

– TensorFlow

– Keras

– Kaffe

– scikit-learn

– XGBoost

– libsvm

Далі отримана модель записується до сховища даних.

3.6 Використання моделі у застосунку засобами CoreML

Швидкість роботи нейронних мереж на мобільних пристроях забезпечується BNNS та MPSCNN:

– BNNS (Basic Neural Network Subroutines), Основні підпрограми для нейронних мереж, є частиною рамки прискорення, колекції математичних функцій, які повністю використовують швидкі векторні інструкції процесора.

– MPSCNN є частиною Metal Performance Shaders, бібліотеки оптимізованих обчислювальних ядер, які працюють на GPU, а не на процесорі.

У своєму нинішньому вигляді BNNS і MPSCNN корисні для виконання висновків на згорткових нейронних мережах.

У порівнянні з TuriCreate, де створюється нейронна мережа з нуля шляхом побудови обчислювального графа, BNNS і MPSCNN є API більш високого рівня - математичних обчислень як таких значно менше.

Це також має недоліки: BNNS і MPSCNN набагато менш здатні, ніж інші фреймворки, такі як TuriCreate. Легше їх запустити, але це обмежує глибоке навчання.

На рисунку 3.6 зображено рівні роботи з CoreML.

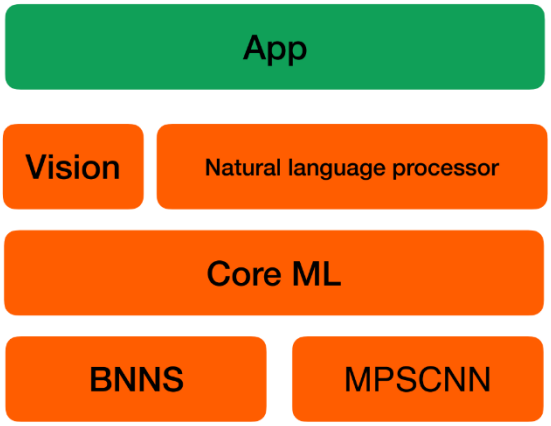


Рисунок 3.6 – Рівні роботи CoreML.

Фрагмент коду використання CoreML з поясненнями:

```
// Allocate memory for storing intermediate and final results.
var tempBuffer1: [Float] = . . .
var tempBuffer2: [Float] = . . .
var results: [Float] = . . .

// Apply the first layer to the input data (for example an image).
BNNSFilterApply(convLayer, inputData, &tempBuffer1)

// Apply the second layer to the output of the first layer.
BNNSFilterApply(poolLayer, tempBuffer1, &tempBuffer2)

// Apply the third and final layer. The results are typically a
// probability distribution.
BNNSFilterApply(fcLayer, tempBuffer2, &results)
```

Оскільки дані перетікають з одного шару в інший в нейронній мережі, кожен шар перетворює дані деяким чином. Як частина цього перетворення, шар застосовує функцію активації (activation function) . Без цих функцій активації нейронні мережі не зможуть навчатися.

Існує широкий вибір функцій активації, і BNNS і MPSCNN підтримують найбільш поширені:

- ReL
- logistic sigmoid
- t*anh* та масштабований *tanh*
- *a*bsolute value
- функція ідентичності, яка передає дані, не змінюючи її
- лінійна (тільки MPSCNN)

BNNS визначає два типи BNNSActivationFunctionRectifiedLinear і BNNSActivationFunctionLeakyRectifiedLinear, але в MPSCNN є лише один MPSCNNNeuronReLU, який має параметр "alpha" для того, щоб зробити ReLU витікним чи ні. Так само і для tanh і масштабованого tanh.

Можна з упевненістю сказати, що MPSCNN набуває більш гнучкого та адаптованого підходу, ніж BNNS. Це те, що відбувається на всьому API. Властивості обох методів наведені у таблиці 2.1.

Таблиця 3.1 — Порівняння BNNS та MPSCNN.

API	BNNS	MPSCNN
Компонент обробки	На основі ЦП	На основі GPU
Мова розробки	API на основі C	Адаптований для Swift
Ключова властивість	Швидка робота	Дозволяє створити власні функції активації

На рисунку 3.7 зображено результати замірів швикдості роботи обох API у консолі.

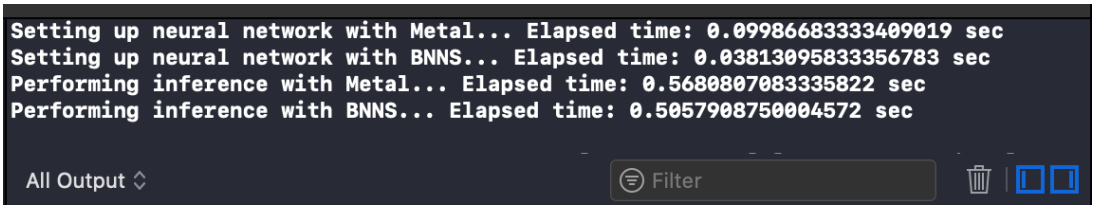


Рисунок 3.7 – Результати замірів швикдості роботи обох API у консолі

Хоча BNNS працює швидше, аніж MPSCNN, було вирішено використовувати CoreML з MPSCNN завдяки підтримці Swift, та тому що ARKit використовує Metal framework для рендерінгу.

3.7 Висновки

У третьому розділі магістерської роботи проведено варіантний аналіз та обгрунтування вибору середовища розробки та аналіз вибору мови програмування і шаблону проектування. Для розробки додатку було обрано технологію Xcode та описано переваги цієї технології серед аналогів. За результатами аналізу, мовою програмування було обрано Python.

Реалізований програмний AR застосунок, який використовує ARKit стек, що приймає контент у вигляді 2D або 3D об’єктів та проекціює його на точки, знайдені методами комп’ютерного зору та оброблені нейронною мережею; SpriteKit та SceneKit дозволяють створювати 2- та 3-вимірні моделі на основі динамічно отриманого контенту та будування просторових моделей на основі точок знайдених ARKit; CoreML фреймворк дозволяє використовувати моделі, отримані ззовні для класифікації оточення, отриманого з Vision; Vision фреймворк дозволяє використовувати сенсори мобільного пристрою як інструменти машинного зору та введення оточення для класифікації нейронною мережею, OpenCV додатково покращує результати роботи Vision

Проведено тренування моделі засобами Turi Create, тренування моделі та конвертація засобами Create ML. Розроблено спосіб використання моделі у застосунку засобами CoreML.

4 ТЕСТУВАННЯ ЕСТУВАННЯ ПРОГРАМНОГО ПРОДУКТУ

4.1 Обгрунтування актуальності проведення тестування

Зазвичай у мобільних застосунках використовується Unit Testing та UI testing, що підходить для перевірки роботи даного AR застосунку.

Unit тестування - це рівень тестування програмного забезпечення, в якому перевіряються окремі одиниці / компоненти програмного забезпечення.

Мета полягає в тому, щоб перевірити, що кожна одиниця програмного забезпечення виконується так, як вона була розроблена. Пристрій є найменшою частиною будь-якого програмного забезпечення, що перевіряється. Зазвичай він має один або декілька входів і зазвичай єдиний вихід. У процедурному програмуванні одиниця може бути окремою програмою, функцією, процедурою тощо. У об'єктно-орієнтованому програмуванні найменша одиниця - це метод, який може належати базовому / супер-класу, абстрактному класу або похідному / дочірньому класу. (Деякі з них розглядають модуль програми як одиницю. Це не рекомендується, оскільки в цьому модулі, ймовірно, буде багато окремих одиниць.) Для полегшення тестування використовуються модульні тестові рамки, драйвери, заглушки та макетні / підроблені об'єкти. .

Unit тести реалізуються за допомогою QUnit в якості тестової основи. Тестова папка містить автоматизовані тестові набори.

Метод модульного тестування повинен відповідати узгодженому формату, щоб полегшити його розуміння іншими розробниками. Наступним форматом, як правило, вважається найкраща практика:

- Arrange;
- Act;
- Assert.

По-перше, для створення екземпляра класу, що тестується, а також будь-яких залежностей, які він може мати, може бути написаний деякий код установки. Це розділ "Впорядкувати" тестового модуля.

Нарешті, при виконанні даного методу або власності тест повинен перевірити, чи метод чи властивість зробили саме те, що він повинен був зробити. Це розділ Assert тесту. Під час фази твердження, затвердження методів викликаються для порівняння фактичних результатів з очікуваними результатами. Якщо фактичні результати є такими, як очікується, проходить модульний тест. Якщо ні, тест провалиться.

Можна запустити програму в Xcode і вставити нерухоме зображення.

Це допомагає перевірити можливість виявлення зображень. Цей підхід використовується для розміщення нерухомого зображення і надає йому декілька різних варіантів, наприклад, чорно-білу або розмитої версії, і перевіряє, чи правильно інтерпретувати зображення.

4.2 Тестування роботи розробленого додатку

Екран авторизації реалізований таким чином, як зображено на рисунку 4.1.

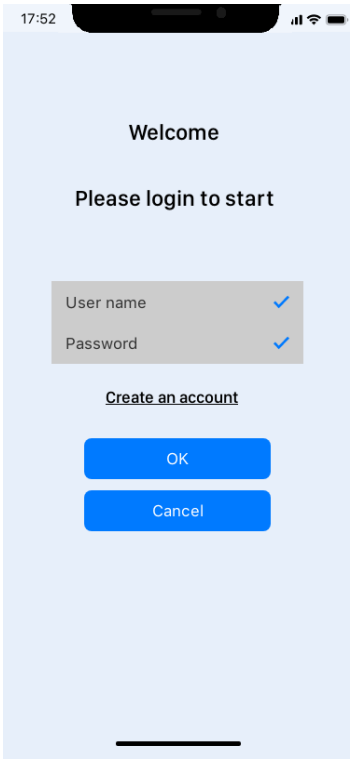


Рисунок 4.1 – Екран авторизації користувача

Коли користувач авторизувався, йому пропонується вибрати вхідні зображення для доповнення реальності (рис 4.2).

Рисунок 4.2 – Вибір зображень для доповнення

Після вибору рисунку, відбувається перехід до камери смартфону та процес доповнення зображення в реальну часі (рис 4.3 – рис. 4.7).

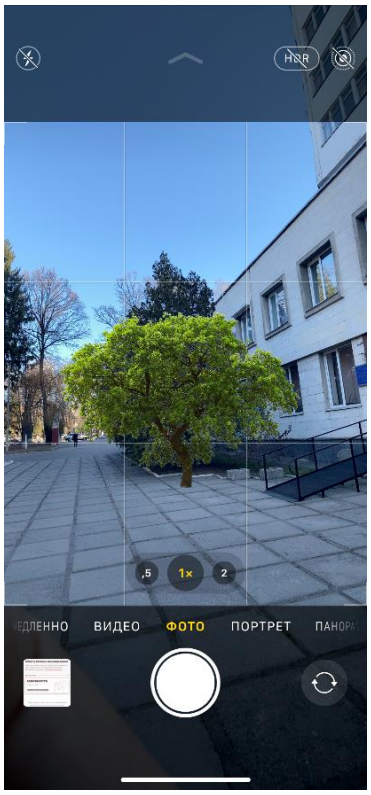


Рисунок 4.3 – Приклад роботи програми

Рисунок 4.4 – Приклад роботи програми

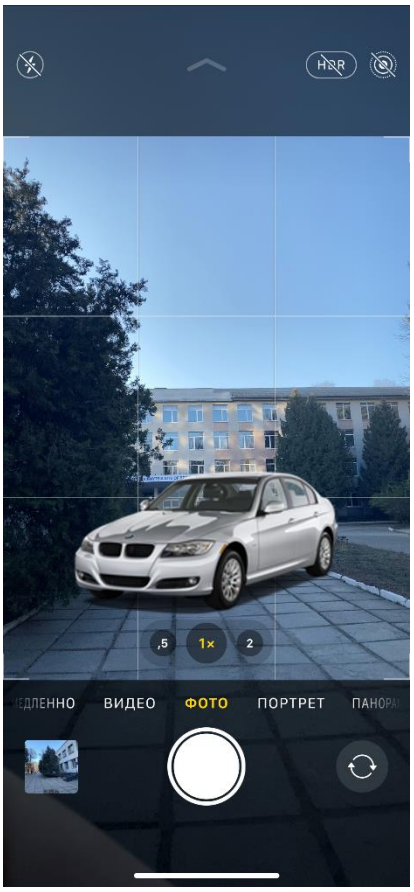


Рисунок 4.5 – Приклад роботи програми

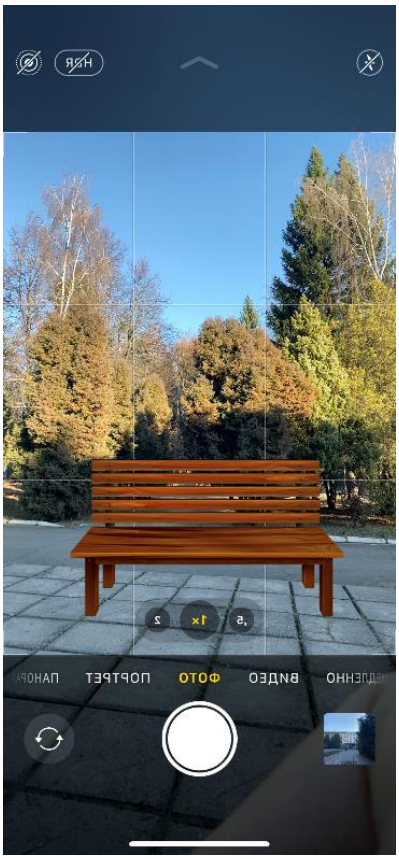


Рисунок 4.6 – Приклад роботи програми

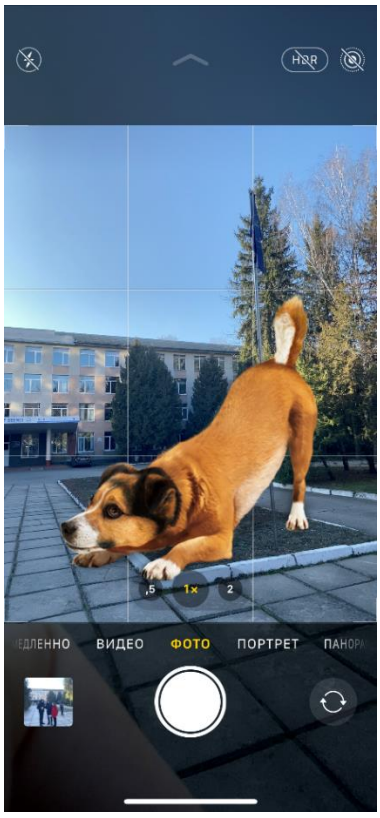


Рисунок 4.7 – Приклад роботи програми

4.3 Висновки

У четвертому розділі проведено Unit Testing та UI testing, що підходять для перевірки роботи даного AR застосунку.

Було перевірено, що кожна одиниця програмного забезпечення виконується так, як вона була розроблена та перевірено можливість виявлення зображень.

Протестовано головну сторінку додатку, екран авторизації, вибору зображень та роботу додатку з 4 різними зображеннями.

5 ЕКОНОМІЧНА ЧАСТИНА

5.1 Оцінювання комерційного потенціалу розробки

Метою проведення технологічного аудиту є оцінювання комерційного потенціалу розробки. Для проведення технологічного аудиту було залучено 2-х незалежних експертів.

Такими експертами будуть Войтко В.В. та Крилик Л.В.

Здійснюємо оцінювання комерційного потенціалу розробки за 12-ма критеріями за 5-ти бальною шкалою.

Результати оцінювання комерційного потенціалу розробки наведено в таблиці 5.1.

Таблиця 5.1 – Результати оцінювання комерційного потенціалу розробки

Критерії	Прізвище, ініціали, посада експерта	
	1. Войтко В.В.	2.Крилик Л.В.
	Бали, виставлені експертами:	
1	4	4
2	3	3
3	3	4
4	4	4
5	3	3
6	3	4
7	4	3
8	3	4
9	4	4
10	4	4
11	3	4
12	3	4

Сума балів	СБ1 = 42	СБ2 = 45
Середньоарифметична сума балів $\overline{СБ}$	$\overline{СБ} = \frac{\sum_{1}^{3} СБ_i}{2} = 43,5$	

Отже, з отриманих даних таблиці 5.1 видно, що нова розробка має високий рівень комерційного потенціалу.

5.2 Прогнозування витрат на виконання науково-дослідної роботи та конструкторсько–технологічної роботи

Для розробки нового програмного продукту необхідні такі витрати.

Основна заробітна плата для розробників визначається за формулою (5.1):

$$З_о = \frac{М}{Т_р} \cdot t, \quad (5.1)$$

де М- місячний посадовий оклад конкретного розробника;

$Т_р$ - кількість робочих днів у місяці, $Т_р$ = 21 день;

t - число днів роботи розробника, t = 40 днів.

Розрахунки заробітних плат для керівника і програміста наведені в таблиці 5.2.

Таблиця 5.2 – Розрахунки основної заробітної плати

Працівник	Оклад М, грн.	Оплата за робочий день, грн.	Число днів роботи, t	Витрати на оплату праці, грн.
Науковий керівник	6500	309,52	8	2476,16
Інженер-програміст	4000	190,47	40	7618,8

Всього:	10094,16

Розрахуємо додаткову заробітну плату:

$$Здод= 0,1 \cdot 10094,16=1009,41 \text{ (грн.)}$$

Нарахування на заробітну плату операторів НЗП розраховується як 37,5…40% від суми їхньої основної та додаткової заробітної плати:

$$Нзп= (Зо + Зр) \cdot \frac{\beta}{100}, \quad (5.2)$$

$$Нзп= (10094,16+1009,41) \cdot \frac{36,3}{100} = 4030,88 \text{ (грн.)}.$$

Розрахунок амортизаційних витрат для програмного забезпечення виконується за такою формулою:

$$А=\frac{Ц \cdot Н_а}{100} \cdot \frac{Т}{12}, \quad (5.3)$$

де Ц – балансова вартість обладнання, грн;

$Н_а$ – річна норма амортизаційних відрахувань % (для програмного забезпечення 25%);

Т – Термін використання (Т=3 міс.).

Таблиця 5.3 – Розрахунок амортизаційних відрахувань

Найменування програмного забезпечення	Балансова вартість, грн.	Норма амортизації, %	Термін використання, міс.	Величина амортизаційних відрахувань, грн
Персональний комп’ютер	9000	25	3	562,5
Всього:				562,5

Розрахуємо витрати на комплектуючі. Витрати на комплектуючі розрахуємо за формулою:

$$K = \sum_1^n Н_i \cdot Ц_i \cdot К_i, \quad (5.4)$$

де n – кількість комплектуючих;

Ні - кількість комплектуючих і-го виду;

Ці – покупна ціна комплектуючих і-го виду, грн;

Кі – коефіцієнт транспортних витрат (приймемо Кі = 1,1).

Таблиця 5.4 - Витрати на комплектуючі, що були використані для розробки ПЗ.

Найменування матеріалу	Одиниці виміру	Ціна, грн.	Витрачено	Вартість витрачених матеріалів, грн.
Флешка	шт.	150	1	150
Пачка паперу	уп.	100	1	100
Ручка	шт.	5	1	5
Всього з урахуванням транспортних витрат				280,5

Витрати на силову електроенергію розраховуються за формулою:

$$В_е = В \cdot П \cdot Ф \cdot К_п ; \quad (5.5)$$

де В – вартість 1кВт-години електроенергії (В=1,7 грн/кВт);

П – установлена потужність комп’ютера (П=0,6кВт);

Ф – фактична кількість годин роботи комп’ютера (Ф=195 год.);

$К_п$ – коефіцієнт використання потужності ($К_п < 1$, $К_п = 0,8$).

$$Ве = 1,7 \cdot 0,6 \cdot 195 \cdot 0,8 = 159,12 \text{ (грн.)}$$

Розрахуємо інші витрати Він.

Інші витрати Ів можна прийняти як (100…300)% від суми основної заробітної плати розробників та робітників, які були виконували дану роботу, тобто:

$$В_{ін} = (1..3) \cdot (З_о + З_р). \quad (5.6)$$

Отже, розрахуємо інші витрати:

$В_{ін} = 1 * (10094{,}16 + 1009{,}41) = 11103{,}57$ (грн).

Сума всіх попередніх статей витрат дає витрати на виконання даної частини роботи:

$$В = З_о + З_д + Н_{зп} + А + К + В_е + І_в$$

В = 10094,16+1009,41+4030,88+562,5+280,5+159,12+11103,57= 27240,14 (грн.)

Розрахуємо загальну вартість наукової роботи $B_{заг}$ за формулою:

$$В_{заг} = \frac{В_{ін}}{\alpha} \quad (5.7)$$

де α – частка витрат, які безпосередньо здійснює виконавець даного етапу роботи, у відн. одиницях = 1.

$$В_{заг} = \frac{27240{,}14}{1} = 27240{,}14$$

Прогнозування загальних витрат ЗВ на виконання та впровадження результатів виконаної наукової роботи здійснюється за формулою:

$$ЗВ = \frac{В_{заг}}{\beta} \quad (5.8)$$

де β – коефіцієнт, який характеризує етап (стадію) виконання даної роботи.

Отже, розрахуємо загальні витрати:

$$ЗВ = \frac{27240,14}{0,9} = 30266,82 \text{ (грн.)}$$

5.3 Прогнозування комерційних ефектів від реалізації результатів розробки.

Спрогнозуємо отримання прибутку від реалізації результатів нашої розробки. Зростання чистого прибутку можна оцінити у теперішній вартості грошей. Це забезпечить підприємству (організації) надходження додаткових коштів, які дозволять покращити фінансові результати діяльності.

Оцінка зростання чистого прибутку підприємства від впровадження результатів наукової розробки. У цьому випадку збільшення чистого прибутку підприємства ΔПі для кожного із років, протягом яких очікується отримання позитивних результатів від впровадження розробки, розраховується за формулою:

$$\Delta П_i = \sum_1^n (\Delta П_я \cdot N + П_я \Delta N)_i \quad (5.9)$$

де ΔПя – покращення основного якісного показника від впровадження результатів розробки у даному році;

N – основний кількісний показник, який визначає діяльність підприємства у даному році до впровадження результатів наукової розробки;

ΔN – покращення основного кількісного показника діяльності підприємства від впровадження результатів розробки;

Пя – основний якісний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки;

n – кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки.

У результаті впровадження результатів наукової розробки витрати на виготовлення програмного продукту зменшуються на 30 грн (що автоматично спричинить збільшення чистого прибутку підприємства на 30 грн), а кількість користувачів, які будуть користуватись збільшиться: протягом першого року – на 150 користувачів, протягом другого року – на 100 користувачів, протягом третього року – 50 користувачів. Реалізація програмного продукту до впровадження результатів наукової розробки складала 1000 користувачів, а прибуток, що отримував розробник до впровадження результатів наукової розробки – 300 грн.

Спрогнозуємо збільшення чистого прибутку від впровадження результатів наукової розробки у кожному році відносно базового.

Отже, збільшення чистого продукту $^{\Delta П_1}$ протягом першого року складатиме:

$$\Delta П_1 = 30 \cdot 1000 + (300 + 30) \cdot 150 = 79500 \text{ грн.}$$

Протягом другого року:

$$\Delta П_2 = 30 \cdot 1000 + (300 + 30) \cdot (150 + 100) = 112500 \text{ грн.}$$

Протягом третього року:

$$\Delta П_3 = 30 \cdot 1000 + (300 + 30) \cdot (150 + 100 + 50) = 129000 \text{ грн.}$$

5.4 Розрахунок ефективності вкладених інвестицій та період їх окупності

Визначимо абсолютну і відносну ефективність вкладених інвестором інвестицій та розрахуємо термін окупності.

Абсолютна ефективність $Е_{абс}$вкладених інвестицій розраховується за формулою:

$$Е_{абс} = (ПП - PV), \quad (5.10)$$

де $^{\Delta П_i}$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДДКР, грн;

т – період часу, протягом якого виявляються результати впровадженої НДДКР, 3 роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні; для України цей показник знаходиться на рівні 0,1;

t – період часу (в роках) від моменту отримання чистого прибутку до точки 2, 3,4.

Рисунок, що характеризує рух платежів (інвестицій та додаткових прибутків) буде мати вигляд, рисунок 5.1.

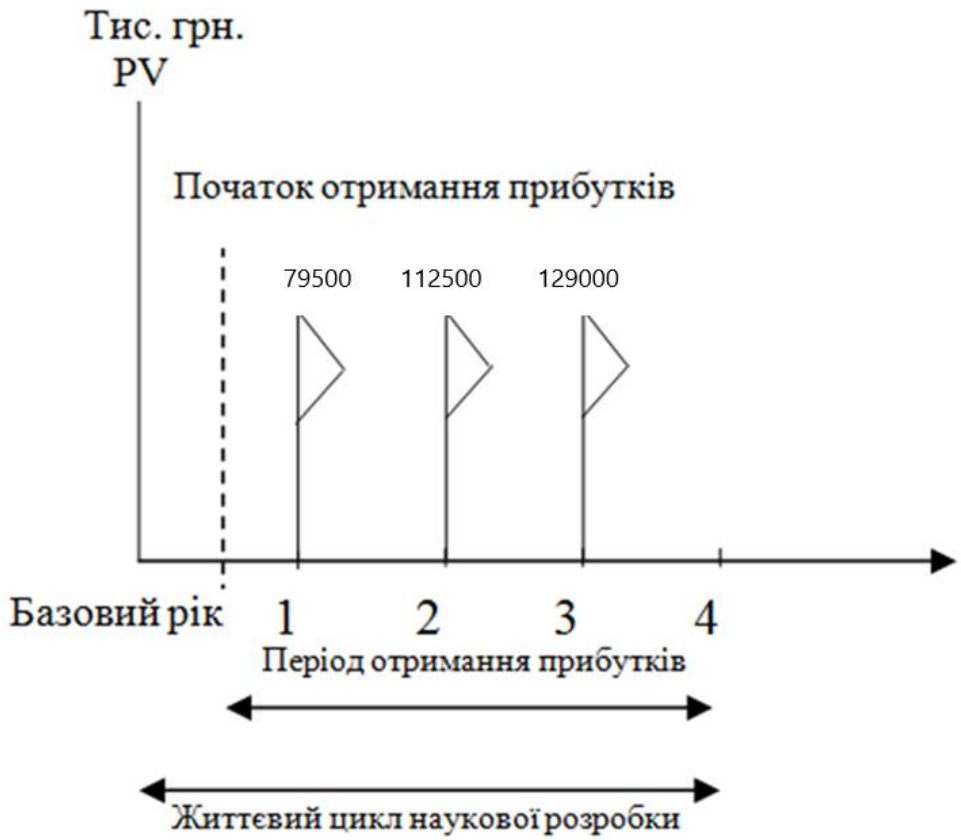


Рисунок 5.1 – Вісь часу з фіксацією платежів, що мають місце під час розробки та впровадження результатів НДДКР

Розрахуємо вартість чистих прибутків за формулою:

$$ПП = \sum_1^m \frac{\Delta П_i}{(1+\tau)^t} \quad (5.11)$$

де $\Delta П_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДДКР, грн;

т – період часу, протягом якого виявляються результати впровадженої НДДКР, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні; для України цей показник знаходиться на рівні 0,1;

t – період часу (в роках) від моменту отримання чистого прибутку до точки.

Отже, розрахуємо вартість чистого прибутку:

$$ПП = \frac{30266,82}{(1+0,1)^0} + \frac{79500}{(1+0,1)^2} + \frac{112500}{(1+0,1)^3} + \frac{129000}{(1+0,1)^4} = 268600,93 \text{ (грн.)}$$

Тоді розрахуємо $E_{абс}$:

$$Е_{абс} = 268600,93 - 30266,82 = 238334,11 \text{ грн.}$$

Оскільки $E_{абс} > 0$, то вкладання коштів на виконання та впровадження результатів НДДКР буде доцільним.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій Ев за формулою:

$$Е_в = \sqrt[Т]{1 + \frac{Е_{абс}}{PV}} - 1 \qquad (5.12)$$

де Еабс – абсолютна ефективність вкладених інвестицій, грн;

PV – теперішня вартість інвестицій PV = ЗВ, грн;

Тж – життєвий цикл наукової розробки, роки.

Тоді будемо мати:

$$E_в = \sqrt[3]{1 + \frac{238334,11}{30266,82}} - 1 = 1,07 \text{ або } 107\ \%$$

Далі, розраховану величина Ев порівнюємо з мінімальною (бар'єрною) ставкою дисконтування τмін, яка визначає ту мінімальну дохідність, нижче за яку інвестиції вкладатися не будуть. У загальному вигляді мінімальна (бар'єрна) ставка дисконтування τмін визначається за формулою:

$$\tau = d + f,$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2019 році в Україні d = 0,2;

f – показник, що характеризує ризикованість вкладень, величина f = 0,1.

$$\tau = 0,2 + 0,1 = 0,3$$

Оскільки Ев = 107% > τмін = 0,3 = 30%, то у інвестор буде зацікавлений вкладати гроші в дану наукову розробку.

Термін окупності вкладених у реалізацію наукового проекту інвестицій. Термін окупності вкладених у реалізацію наукового проекту інвестицій Ток розраховується за формулою:

$$T_{ок} = \frac{1}{E_в}$$

$$T_{ок} = \frac{1}{1,07} = 0,93 \text{ року}$$

Обрахувавши термін окупності даної наукової розробки, можна зробити висновок, що фінансування даної наукової розробки буде доцільним.

5.5 Висновки

У п’ятому розділі було проведено оцінювання комерційного потенціалу розробки, виконано технологічний аудит для оцінювання комерційного потенціалу розробки. Було залучено 2-х незалежних експерта. Прогнозовано витрати на виконання науково-дослідної роботи та конструкторсько–технологічної роботи.

Прогнозовано отримання прибутку від реалізації результатів розробки. Зростання чистого прибутку оцінено у теперішній вартості грошей. Визначено абсолютну і відносну ефективність вкладених інвестором інвестицій та розраховано термін окупності.

ВИСНОВКИ

У магістерській кваліфікаційній роботі удосконалено метод проекціонування динамічного веб-контенту у доповненій реальності, розроблено модель системи, що здатна тренуватися на обраному датасеті, розроблено програмний додаток для зберігання та передачі тривимірного, двовимірного або динамічного веб-контенту у вигляді 3D моделей, зображень, відео, відеопотоку, анімацій, веб-сторінок, здійснення динамічної обробки інформації.

На основі отриманих в магістерській кваліфікаційній роботі теоретичних положень запропоновано новий метод та досконалішу систему проекціонування веб-контенту за допомогою технології віртуальної реальності та розроблено програмні засоби для проекціонування та візуалізації тривимірних зображень.

Досліджено актуальність даної розробки. Було проаналізовано стан даної проблеми на сьогоднішній день. Розглянуто основні аналоги, визначено їх особливості та недоліки і розроблено порівняння з власним програмним продуктом та методом.

Досліджено процес AR доповненої реальності, що базується на маркерах, CoreML фреймворці, Vision фреймворці, Create ML утиліті, фреймворці для навчання моделі TuriCreate, програмі для розмітки фотографій RectLabel та скрипіт Python.

Покращено якість взаємодії технологій доповненої реальності та веб-контенту, підвищено реалістичність доповнених елементів на веб-сторінці за рахунок використання розроблених моделей та методу проекціонування динамічного веб-контенту у доповненій реальності, що дозволило побудувати AR застосунок доповненої реальності із забезпеченням високої реалістичності зображень.

У процесі досліджень було використано теорію нейронних мереж для аналізу та розпізнавання образів, засоби 3D-моделювання для створення

віртуальних моделей об’єктів, методи комп’ютерного моделювання для тестування роботи програми, моделі сприйняття оточення методами доповненої реальності.

Подальшого розвитку дістав метод проекціонування динамічного веб-контенту у доповненій реальності, який, на відміну від існуючих, орієнтований проекціонування динамічного веб-контенту у доповненій реальності за рахунок використання технології AR у динамічних веб-ресурсах, що дозволяє збільшити реалістичність зображень доповненої реальності.

Подальшого розвитку дістали моделі проекціонування динамічного веб-контенту що, на відміну від існуючих, здатні тренувати на обраному датасеті та зберігати моделі доповненої реальності, постачати їх на мобільний застосунок, зберігати та передавати мобільному застосунку тривимірний, двовимірний або динамічний веб-контент у вигляді 3D моделей, зображень, відео, відеопотоку, анімацій, веб-сторінок, динамічно обробляти їх для подальшого використання у доповненій реальності та передавати отриманий результат на компонент доповненої реальності для подальшого проекціонування, що забезпечує підвищення реалістичності доповненого контенту.

Процес передачі, генерації та проекціонування контенту відбувся в реальному часі у працюючому додатку. Для вирішення задачі створення і постачання нейронних мереж було налаштовано алгоритм тренування моделі нейронної мережі на віддаленому сервері із запропонованим датасетом, переведення цієї мережі у формат, що буде здатний працювати з мобільними технологіями доповненої реальності.

За результати дослідження було зроблено дві наукові публікації. Результати доповідались на двох конференціях та здобули перемогу в 13 міжнародних конкурсах.

Роботу виконано на замовлення ТОВ «Кусто Агро», про що свідчить акт впровадження.

ПЕРЕЛІК ПОСИЛАНЬ

1. Богачук Г.В. Розробка методу і програмних засобів обробки й проекціюнування веб-контенту з використанням доповненої реальності / В.В.Войтко, С.В. Бевз, С.М. Бурбело, Н.Є. Барчук, О.В. Гаврилюк, Г.В.Богачук // Всеукраїнська науково-практична Інтернет-конференція «Молодь в науці: дослідження, проблеми, перспективи - 2019». [електронний ресурс] // Режим доступу: https://conferences.vntu.edu.ua/index.php/mn/mn2020/author/submission/8485

2. Богачук Г.В. Розробка веб-ресурсу агро компанії/ Тези доповіді/ Матеріали XLVII Міжнародної науково-технічної конференції факультету інформаційних технологій та комп'ютерної інженерії 2018 [Електронний ресурс] // Режим доступу: https://conf.vntu.edu.ua/index.php/all-fitki/all-fitki-2018/paper/view/5196 .

3. Богачук Г.В. Розробка сайту «Вінниччина – перлина Поділля» / Г.В. Богачук, І. В. Кобися, А. В. Волошина, В. М. Задорожний, М. А. Гусак, В.В.Войтко // XLV Науково-технічна конференція факультету інформаційних технологій та комп'ютерної інженерії (2016). [Електронний ресурс] // Режим доступу: https://conf.vntu.edu.ua/index.php/all-fitki/all-fitki-2016/paper/view/5246.

4. Шаев Ю.М. Виртуальная реальность. Пролегомены к онтологии ISBN:978-5-9710-6301-8 2016

5. Slyusar, Vadym Artificial intelligence as the basis of future control networks, 2019.

2. P. Milgram and A. F. Kishino, Taxonomy of Mixed Reality Visual Displays Архивировано 3 ноября 2009 года. IEICE Transactions on Information and Systems, E77-D (12), 1994 – pp. 1321—1329.

3. Войтко В.В., Богачук Г.В. XLVII Міжнародна науково-технічна конференція факультету інформаційних технологій та комп'ютерної інженерії 2018 та Всеукраїнської науково-практичної інтернет-конференції «Молодь в науці: дослідження, проблеми, перспективи» - 2019.

4. Иванова А. Технологии виртуальной и дополненной реальности: возможности и препятствия применения // Стратегические решения и риск-менеджмент. — 2018. — Вып. 3 (108). — ISSN 2618-947X.

5. Яковлев Б. С., Пустов С. И. Классификация и перспективные направления использования технологии дополненной реальности // Известия Тульского государственного университета. Технические науки. — 2013.

6. R. Azuma, A. Survey of Augmented Reality Presence: Teleoperators and Virtual Environments, August 1997 – pp. 355—385.

7. Ray Wanderlich, ARKit by Tutorials// Razeware LLC -p. 32-37, 2019

8. Schueffel, Patrick (2017). The Concise Fintech Compendium. Fribourg: School of Management Fribourg/Switzerland.

9. Wu, Hsin-Kai; Lee, Silvia Wen-Yu; Chang, Hsin-Yi; Liang, Jyh-Chong (March 2013). "Current status, opportunities and challenges of augmented reality in education". Computers & Education. 62: 41–49.

10. Ray Wanderlich, ARKit by Tutorials// Razeware LLC, 2019 – p. 32-37

11. Steuer,"Defining Virtual Reality: Dimensions Determining Telepresence" (PDF). Archived from the original (PDF) on 24 May 2016. Retrieved 27 November 2018., Department of Communication, Stanford University. 15 October 1993.

12. Chen, Brian (25 August 2009). "If You're Not Seeing Data, You're Not Seeing". Wired. Retrieved 18 June 2019.

13. Mann, Steve; Feiner, Steve; Harner, Soren; Ali, Mir Adnan; Janzen, Ryan; Hansen, Jayse; Baldassi, Stefano (15 January 2015). "Wearable Computing, 3D Aug* Reality, Photographic/Videographic Gesture Sensing, and Veillance". Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction - TEI '14. ACM. pp. 497–500. doi:10.1145/2677199.2683590. ISBN 9781450333054

14. Ma, Minhua; C. Jain, Lakhmi; Anderson, Paul (2014). Virtual, Augmented Reality and Serious Games for Healthcare 1. Springer Publishing. p. 120. ISBN 978-3-642-54816-1.

15. Stewart-Smith, Hanna. Education with Augmented Reality: AR textbooks released in Japan, ZDnet, 4 April 2012.

16. Штучні нейронні мережі: Про обчислення, М.А. Новотарській, Б.Б. Нестеренко [Текст] – Інститут математики НАН України, Київ, 2004 – 127с.

17. Drummond T., Rosten E. Machine Learning for high-speed corner detection - 9th European Conference on Computer Vision (ECCV), 2006 – p. 430-443.

18. Lowe D.G. Distinctive Image Features from Scale-Invariant Keypoints [Текст] – International Journal of Computer Vision, 2004 – p. 91-110.

19. Ian Goodfellow, Yoshua Bengio, Deep Learning (Adaptive Computation and Machine Learning series) // The MIT Press, 2016 – p. 621-638.

20. Frank Millstein, Deep Learning: 2 Manuscripts - Deep Learning With Keras And Convolutional Neural Networks In Python // Paperback, 2018 – p.117-129.

21. Joshua Newman, Machine Learning with Core ML: An iOS developer's guide to implementing machine learning in mobile apps // Packt Publishing, 2018 – p. 45-48.

ДОДАТКИ

Додаток А – Технічне завдання

Міністерство освіти і науки України

Вінницький національний технічний університет

Факультет інформаційних технологій та комп’ютерної інженерії

ЗАТВЕРДЖУЮ

д.т.н., проф. О. Н. Романюк

"____ "_____2019 р.

Технічне завдання

на магістерську кваліфікаційну роботу

«Розробка методу і програмних засобів обробки й проекціюнування веб-контенту з використанням доповненої реальності»

за спеціальністю

121 – Інженерія програмного забезпечення

Керівник магістерської кваліфікаційної роботи:

_____ д.т.н., проф. О.Н.Романюк

"____"_____2019 р.

Виконала:

_____студентка гр.1ПІ-18м Г.В. Богачук

"____"_____2019 р.

Вінниця – 2019 року

1. Найменування та галузь застосування

Магістерська кваліфікаційна робота: «Розробка методу і програмних засобів обробки й проекціюнування веб-контенту з використанням доповненої реальності».

Галузь застосування - системи комп’ютерної графіки.

2. Підстава для розробки.

Підставою для виконання магістерської кваліфікаційної роботи (МКР) є індивідуальне завдання на МКР та наказ № ректора по ВНТУ про закріплення тем МКР.

3. Мета та призначення розробки.

Метою роботи є процес покращення якості взаємодії технологій доповненої реальності та веб-контенту, підвищення реалістичності доповнених елементів на веб-сторінці за рахунок використання розроблених моделей та методу проеціювання динамічного веб-контенту у доповненій реальності, що дозволяє побудувати AR застосунок доповненої реальності із забезпеченням високої реалістичності зображень.

Призначення роботи – розробка методу та системи проекціонування веб-контенту за допомогою технології віртуальної реальності та програмних засобів для проекціонування та візуалізації тривимірних зображень.

3 Вихідні дані для проведення НДР

Перелік основних літературних джерел, на основі яких буде виконуватись МКР.

1. Шаев Ю.М. Виртуальная реальность. Пролегомены к онтологии ISBN:978-5-9710-6301-8 2016

2. Slyusar, Vadym Artificial intelligence as the basis of future control networks, 2019.

3. P. Milgram and A. F. Kishino, Taxonomy of Mixed Reality Visual Displays Архивировано 3 ноября 2009 года. IEICE Transactions on Information and Systems, E77-D (12), pp. 1321—1329, 1994.

4. Иванова А. Технологии виртуальной и дополненной реальности: возможности и препятствия применения // Стратегические решения и риск-менеджмент. — 2018. — Вып. 3 (108). — ISSN 2618-947X.

5. Яковлев Б. С., Пустов С. И. Классификация и перспективные направления использования технологии дополненной реальности // Известия Тульского государственного университета. Технические науки. — 2013.

4. Технічні вимоги

Методи доповнення реальності, які базуються на маркерах, CoreML фреймворк, Vision фреймворк, Create ML утиліта, фреймворк для навчання моделі TuriCreate, програма для розмітки фотографій RectLabel та скрипти Python.

5. Конструктивні вимоги.

Мобільний додаток повинен відповідати всім вимогам, повинен бути зручним та зрозумілим у використанні.

Графічна та текстова документація повинна відповідати діючим стандартам України.

6. Перелік технічної документації, що пред'являється по закінченню робіт:

– пояснювальна записка до МКР;

– технічне завдання;

– лістинги програми;

– Акт впровадження.

8. Вимоги до рівня уніфікації та стандартизації

При розробці програмних засобів слід дотримуватися уніфікації і ДСТУ.

9. Стадії та етапи розробки:

№ з/п	Назва етапів магістерської кваліфікаційної Роботи	Строк виконання етапів роботи
1	Аналіз інформаційного забезпечення та постановка задачі	07.10.2018 –
2	Розробка методу і моделей для оптимізації нейронних мереж та просторового сприйняття AR	27.10-2019
3	Розробка програми для проеціювання динамічного веб-контенту у доповненій реальності	28.10.2019 – 8.11.2019
4	Тестування програмної системи	9.11.2019 – 20.11.2019
5	Економічна частина	21.11.2019 – 3.12.2019

10. Порядок контролю та прийняття.

Виконання етапів магістерської кваліфікаційної роботи контролюється керівником згідно з графіком виконання роботи.

Прийняття магістерської кваліфікаційної роботи здійснюється ДЕК, затвердженою зав. кафедрою згідно з графіком

Додаток Б – Акт впровадження

Додаток В – Програмний код додатку

classifier_resnet.py

```
import turicreate as turi

datasetId = 0
def __init__(self, id):
      self.datasetId = id
url = FileManager.urlFor(self.datasetId)
data = turi.image_analysis.load_images(url)
data["type"] = data["path"].apply(lambda path: "Type1" if "type2" in
path else "Types")
data.save("type1orType2.sframe")
data.explore()
dataBuffer = turi.SFrame("type1orType2.sframe")
trainingBuffers, testingBuffers = dataBuffer.random_split(0.9)
model = turi.image_classifier.create(trainingBuffers, target="type",
model="resnet-50")
evaluations = model.evaluate(testingBuffers)

model.save("TypeClassifier.model")
model.export_coreml("TypeClassifier.mlmodel")
```

VisionClassifier+CoreML.swift

```
import Foundation
import CoreML
import Vision

var classificationRequest: VNCoreMLRequest

func loadModel() -> VNCoreMLRequest {
      let model = try VNCoreMLModel(for: MobileNet().model)

      let request = VNCoreMLRequest(model: model, completionHandler: {
      [weak self] request, error in
          self?.processClassifications(for: request, error: error)
      })
      request.imageCropAndScaleOption = .centerCrop
      self.classificationRequest = request
      return request
}
```

```
func initiateHandler(){
      DispatchQueue.global(qos: .userInitiated).async {
          let handler = VNImageRequestHandler(ciImage: ciImage,
      orientation: orientation)
          do {
              try handler.perform([self.classificationRequest])
          } catch {
              print("Failed to perform
      classification.\n\(error.localizedDescription)")
          }
      }
}

func processClassifications(for request: VNRequest, error: Error?) {
    DispatchQueue.main.async {
        guard let results = request.results else {
            self.classificationLabel.text = "Unable to classify
image.\n\(error!.localizedDescription)"
            return
        }
            let classifications = results as!
[VNClassificationObservation]
      }
}
```

ContentConverter3D.swift

```
     class func loadObjectNamed(_ objectName: String, fromSceneNamed
sceneName: String) -> CAAnimation? {
          let url = FileManager(urlForResource: sceneName, ofType:
"DAE")
          #if os(OSX)
               let options: [SCNSceneSource.LoadingOption: Any] =
[SCNSceneSource.LoadingOption.convertToYUp: true]
          #else
               let options: [SCNSceneSource.LoadingOption: Any] = [:]
          #endif
          let sceneSource = SCNSceneSource(url: url, options: options)
          let animation =
sceneSource?.entryWithIdentifier(animationName, withClass:
CAAnimation.self)
          animation?.fadeInDuration = 0.3
          animation?.fadeOutDuration = 0.3
          return animation
     }
```

ARViewController.swift

```
class ARViewController: UIViewController, ARSCNViewDelegate {

      @IBOutlet var sceneView: ARSCNView!

      var nodeModel:SCNNode!
      let nodeName = ""

      override func viewDidLoad() {
      super.viewDidLoad()
      sceneView.delegate = self

      sceneView.showsStatistics = true
      sceneView.antialiasingMode = .multisampling4X

            let scene = SCNScene()
      sceneView.scene = scene

      let modelScene = SCNScene(named:
            "art.scnassets/cherub/cherub.dae")!

      nodeModel =  modelScene.rootNode.childNode(withName: nodeName,
recursively: true)
      }
}
```

Додаток Г – Ілюстративний матеріал

ІЛЮСТРАТИВНИЙ МАТЕРІАЛ ДО ЗАХИСТУ МАГІСТЕРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Завідувач кафедри ПЗ, д. т. н., професор _____ О. Н. Романюк

Науковий керівник, к. т. н., доцент кафедри ПЗ_____ В. В. Войтко

Рецензент, к.т.н, доцент кафедри КН_____ Л. В. Крилик

Нормоконтроль, к. т. н., проф. кафедри ПЗ _____ В. В. Войтко

Виконавець, студент групи 2ПІ-17м _____ Г. В. Богачук

Мета

- **Метою роботи** є процес покращення якості взаємодії технолігій доповненої реальності та веб-контенту, підвищення реалітичності доповнених елементів на веб-сторінці за рахунок використання розроблених моделей та методу проеціювання динамічного веб-контенту у доповненій реальності, що дозволяє побудувати AR застосунок доповненої реальності із забезпеченням високої реалістичності зображень.

Об'єкт та предмет

- **Об'єктом дослідження** є процес AR (доповненої реальності), що базується на маркерах, CoreML фреймворці, Vision фреймворці, Create ML утиліті, фреймворці для навчання моделі TuriCreate, програмі для розмітки фотографій RectLabel та скрипті Python.
- **Предметом** дослідження є засоби релізації AR модулів.

Основні задачі дослідження:

- Удосконалення методу проеціювання динамічного веб-контенту у доповненій реальності.
- Розробка моделей системи, що здатна тренуватися на обраному датасеті.
- Збереження моделей доповненої реальності.
- Постачання моделей на мобільний застосунок, зберігання та передача мобільному застосунку тривимірний, двовимірний або динамічний веб-контент у вигляді 3D моделей, зображень, відео, відеопотоку, анімацій, веб- сторінок, здійснення динамічної обробки інформації.
- Передача отриманого результату на компонент доповненої реальності для подальшого проеціювання.

Наукова новизна:

- 1. Подальшого розвитку дістав метод проеціювання динамічного веб-контенту у доповненій реальності, який, на відміну від існуючих, орієнтований проеціювання динамічного веб-контенту у доповненій реальності за рахунок використання технології AR у динамічних веб-ресурсах, що дозволяє збільшити реалістичність зображень доповненої реальності.
- 2. Подальшого розвитку дістали моделі проеціювання динамічного веб-контенту що, на відміну від існуючих, здатні тренувати на обраному датасеті та зберігати моделі доповненої реальності, постачати їх на мобільний застосунок, зберігати та передавати мобільному застосунку тривимірний, двовимірний або динамічний веб-контент у вигляді 3D моделей, зображень, відео, відеопотоку, анімацій, веб-сторінок, динамічно обробляти їх для подальшого використання у доповненій реальності та передавати отриманий результат на компонент доповненої реальності для подальшого проеціювання, що забезпечує підвищення реалістичності доповненого контенту.

Покроковий опис методу:

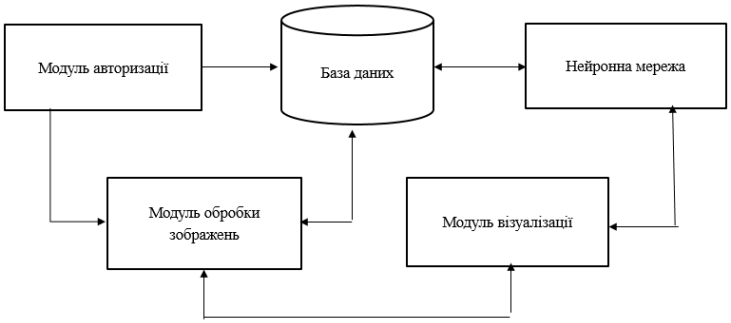
- 1. Заповнення бази даних тривимірними елементами
- 2. Опрацювання робочої карти
- 3. Розрахунок математичної моделі за формулами
- 4. Розрахунок за створеної моделлю
- 5. Доповнння реальності з використанням засобаму Turi Create
- 6. Доповнння реальності з використанням засобаму reate ML
- 7. Використання моделі у застосунку засобами CoreML

Технології:

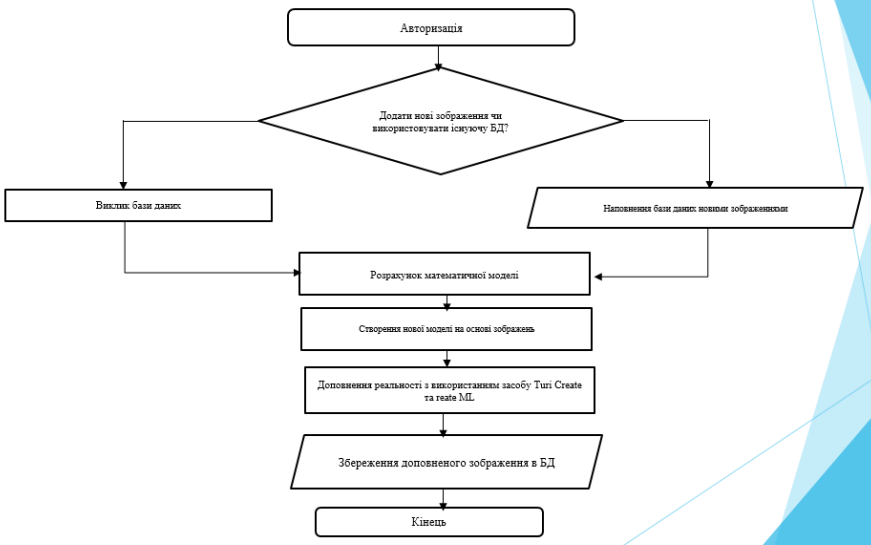
- Xcode

- Python

Модель системи доповненої реальності

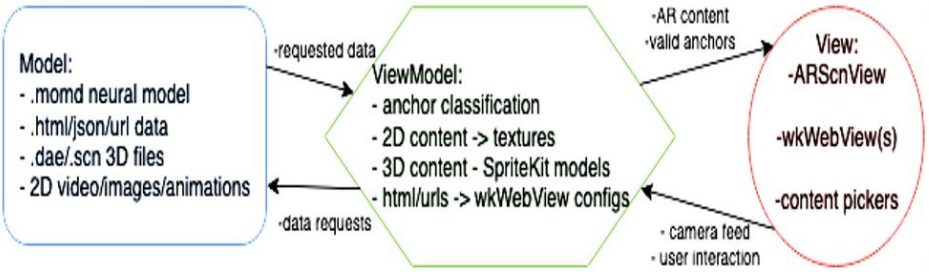


Блок-схема роботи додатку

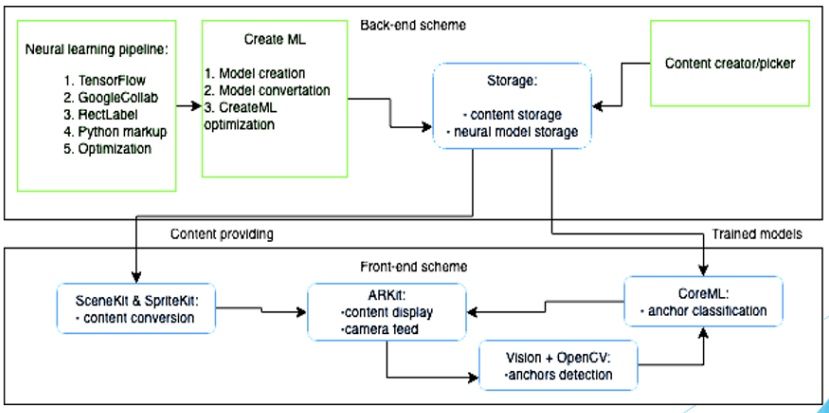


Блок-схема модуля авторизації

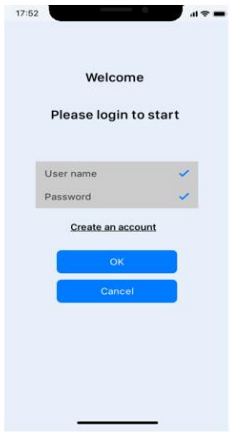
Model-view-viewmodel

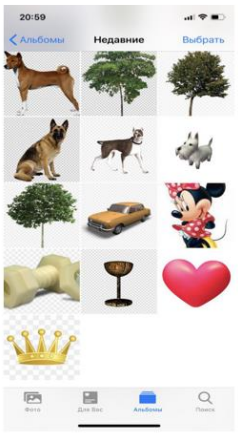


Загальна схема системи генерації та проеціювання динамічного веб-контенту



Екран авторизації та вибору зображень для доповнення





Приклад роботи

Приклад роботи

Прибуток:

Протягом третього року:

- **129000** грн.

Абсолютна ефективність ▶ **238334, 11 грн**

Відносна ефективність ▶ **107 %**

Термін окупності ▶ **0,93 року**

Висновки

- В роботі запропоновано новий метод та досконалішу систему проекціонування веб-контенту за допомогою технолігї віртуальної реальності та розроблено програмні засоби для проекціонування та візуалізації тривимірних зображень.
- Досліджено актуальність даної розробки. Було проаналізовано стан даної проблеми на сьогоднішній день. Розглянуто основні аналоги, визначено їх особливості та недоліки і розроблено порівняння з власним програмним продуктом та методом.
- Досліджено процес AR доповненої реальності, що базується на маркерах, CoreML фреймворці, Vision фреймворці, Create ML утиліті, фреймворці для навчання моделі TuriCreate, програмі для розмітки фотографій RectLabel та скрипті Python.
- Покращено якість взаємодії технолігій доповненої реальності та веб-контенту, підвищенно реалітичность доповнених елементів на веб-сторінці за рахунок використання розроблених моделей та методу проеціювання динамічного веб-контенту у доповненій реальності, що дозволило побудувати AR застосунок доповненої реальності із забезпеченням високої реалістичності зображень.
- У процесі досліджень було використано теорію нейронних мереж для аналізу та розпізнавання образів, засоби 3D-моделювання для створення віртуальних моделей об'єктів, методи комп'ютерного моделювання для тестування роботи програми, моделі сприйняття оточення методами доповненої реальності.
- Подальшого розвитку дістав метод проеціювання динамічного веб-контенту у доповненій реальності, який, на відміну від існуючих, орієнтований проеціювання динамічного веб-контенту у доповненій реальності за рахунок використання технології AR у динамічних веб-ресурсах, що дозволяє збільшити реалістичність зображень доповненої реальності.
- Подальшого розвитку дістали моделі проеціювання динамічного веб-контенту що, на відміну від існуючих, здатні тренувати на обраному датасеті та зберігати моделі доповненої реальності, постачати їх на мобільний застосунок, зберігати та передавати мобільному застосунку тривимірний, двовимірний або динамічний веб-контент у вигляді 3D моделей, зображень, відео, відеопотоку, анімацій, веб-сторінок, динамічно обробляти їх для подальшого використання у доповненій реальності та передавати отриманий результат на компонент доповненої реальності для подальшого проеціювання, що забезпечує підвищеня реалістичності доповненого контенту.