

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет
Кафедра

Комп'ютерної інженерії та управління
Комп'ютерних інтелектуальних технологій та систем

КВАЛІФІКАЦІЙНА РОБОТА **Пояснювальна записка**

рівень вищої освіти

другий (магістерський)

Застосування технологій машинного навчання
для розпізнавання жестів рукою

Виконав:

студент 2 курсу, групи КІТм-20-1

Кушаков П.К.

Спеціальність 123 Комп'ютерна
інженерія Тип програми освітньо-
професійна

Освітня програма Комп'ютерні
інтелектуальні технології

Керівник доц. Сердюк Н. М

Допускається до захисту

(підпис)

Зав. кафедри

(підпис) проф. Руденко О.Г.

2021 р.

5.Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням кафедри)

6.Консультанти розділів роботи (п.6 включається до завдання за наявністю

консультантів згідно до наказу, зазначеному у п.1)

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

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Видача та узгодження теми проекту	08.11.2021	
2	Аналіз проблеми та огляд існуючих рішень	15.11.2021	
3	Вибір технології розробки та інструментальних засобів	22.11.2021	
4	Розробка алгоритмічного забезпечення	24.11.2021	
5	Розробка програмних модулів	29.12.2021	
6	Відлагодження програмних модулів	01.12.2021	
7	Оформлення матеріалів атестаційної роботи	03.12.2021	
8	Перевірка виконаного проекту керівником	10.12.2021	
9	Захист проекту	17.12.2021	

Дата видачі завдання 8 листопада 2021р.

Студент _____
(підпис)

Керівник роботи _____ доц. Сердюк Н. М
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи містить: 72 сторінки, 10 рисунків, 4 додатки, 20 джерел переліку посилань.

Метою кваліфікаційної роботи є розробка програмного забезпечення технології машинного навчання для розпізнавання жестів рукою для керування комп'ютером без використання клавіатури чи миші.

У ході виконання кваліфікаційної роботи був проведений аналіз технології машинного навчання для визначеної предметної області, їх основні недоліки та переваги. Проаналізовано основні існуючі платформи машинного навчання. З усіх представлених варіантів було обрано платформу MediaPipe. Було розроблено програмне забезпечення, яке дозволяє завдяки алгоритмам заснованим на машинному навчанню рухати курсором або керувати налаштуванням гучності комп'ютера застосовуючи лише стандартну веб-камеру.

МАШИННЕ НАВЧАННЯ, ВЕБ-КАМЕРА, КОМП'ЮТЕРНИЙ ЗІР,
ФРЕЙМВОРК

ABSTRACT

Qualification s thesis contains: 72 pages, 10 figures, 4 appendices, 20 sources.

The major goal of this thesis is to develop hand gesture recognition machine learning based software technology to control computer without the use of a keyboard or a mouse.

During the qualification work the machine learning technology for a chosen subject area and their main disadvantages and advantages were analyzed. The main existing machine learning platforms were analyzed. From all the presented options, the MediaPipe platform was chosen. Software that allows machine-based algorithms to move the cursor or change control computer volume settings using only a standard web-camera has been developed.

MACHINE LEARNING, WEB-CAMERA, COMPUTER VISION,
FRAMEWORK

АНОТАЦІЯ

У часи швидкого розвитку апаратного та програмного забезпечення з'являється потреба у нових методах взаємодії. В останні десятиліття клавіатура та миша відіграють значну роль у взаємодії людини з комп'ютером, але технологія комп'ютерного зору для розпізнавання жестів також стає важливою частиною взаємодії між людиною та комп'ютером. Зокрема, такі технології, як розпізнавання жестів, отримують велику увагу.

Актуальність дослідження полягає у тому, що технології машинного навчання дозволяють комп'ютерам та системам отримувати значну інформацію із цифрових зображень, відео та інших візуальних входів, і на основі цих вихідних даних програми можуть виконувати різноманітні дії. Комп'ютерний зір, що працює на основі машинного навчання, може використовуватися для розмітки фотографій у соціальних мережах, для радіологічної візуалізації в галузі охорони здоров'я та в самокерованих автомобілях. Здатність сприймати форму та рух рук за допомогою комп'ютерного зору може бути життєво важливим компонентом у покращенні взаємодії з користувачами в різних технологічних сферах та платформах. Наприклад, це може стати основою для розуміння мови жестів та управління жестами рук. На сьогодні існує та практично застосовується технологія від Meta для Oculus Quest, яка дозволяє ефективно відстежувати рухи рук і пальців. Проте для роботи із цією розробкою потрібні окуляри віртуальної реальності Oculus Quest.

Метою кваліфікаційної роботи є розробка програмного забезпечення технології машинного навчання для розпізнавання жестів рукою для керування комп'ютером без використання клавіатури чи миші.

Для реалізації поставленої мети необхідно вирішити наступні завдання:

- аналіз існуючої проблеми, застосування машинного навчання для розпізнавання жестів та огляд літературних джерел;
- огляд сучасного інструментарію для розробки моделей машинного навчання з визначеної проблеми;
- вибір моделей для технологій машинного навчання;
- програмна реалізація запропонованих моделей.

Об'єктом дослідження є процес розпізнавання жестів рукою за допомогою моделей технології машинного навчання.

Предмет дослідження являє собою процес перетворення зображень руки які отримані з веб-камери на координатні точки

Необхідність спеціальних датчиків та складного програмного забезпечення донедавна формувала бар'єр для створення та впровадження такого виду інтерактивних інструментів в більшості для розробників, особливо у середовищах з обмеженими ресурсами. На щастя, досягнення машинного навчання та розробка швидких та ефективних моделей виявлення об'єктів тепер дають нам можливість виконувати відстеження, використовуючи лише зображення з однієї простої камери без додаткових інструментів таких, як інфрачервоні датчики, лідар тощо. Поєднання цих досягнень із сучасними бібліотеками та фреймворками забезпечує ефективну основу для створення інтерактивного досвіду та відкриває людям можливість експериментувати з виявленням об'єктів та відстеженням рук.

Здатність сприймати форму та рух рук може бути життєво важливим компонентом у покращенні взаємодії з користувачами в різних технологічних сферах та платформах. Наприклад, це може стати основою для розуміння мови жестів та управління жестами рук, а також може забезпечити накладання цифрового вмісту та інформації поверх фізичного світу в доповненій реальності.

Існує багато платформ для роботи з машинним навчанням, аналіз яких та, зокрема, розробка власних технологій є метою даного проекту. У процесі роботи було розглянуто деякі з них.

- Firebase ML Kit – це мобільний SDK від Google для Android та iOS, який дозволяє використовувати машинне навчання у програмах. Для роботи з інструментом на початку також не потрібно мати знань про машинне навчання. ML Kit постачається з набором готових до використання API для загальних випадків використання, і тому для інтегрування технології потрібно лише кілька рядків коду, щоб впровадити ці API у програми [10].
- TensorFlow – одна з найпопулярніших систем машинного навчання та глибокого навчання, яку використовують розробники та дослідники. Спочатку запущений в 2007 році командою Google Brain, TensorFlow став повноцінною платформою машинного навчання [11]. Моделі, навчені TensorFlow, можна застосовувати на комп'ютерах, браузерях і навіть мікроконтролерах. Ця широка підтримка робить TensorFlow унікальним продуктом.
- PyTorch – це система глибокого навчання з відкритим кодом, створена бути гнучкою та модульною для досліджень зі стабільністю та підтримкою, яка необхідна для розгортання виробництва. Він заснований на Torch, структурі для виконання швидких обчислень, що на початку був написаний на мові програмування C [12].
- DeepLearning4j – один з небагатьох фреймворків машинного навчання, написаних на Java та націлених на віртуальну машину Java. Він створений групою розробників машинного навчання, що базується у Сан-Франциско і комерційно підтримується стартапом SkyMind. DeepLearning4j був переданий Фонду Eclipse у жовтні 2017 року [14]. Бібліотека сумісна з Clojure та Scala.
- MediaPipe Hands – це високоякісне рішення для відстеження рук і пальців. Він використовує машинне навчання, щоб вивести 21 тривимірний орієнтир руки лише з одного кадру. У той час як інші сучасні підходи в

першу чергу покладаються на потужні робочі середовища для виведення, цей метод досягає ефективного результату в реальному часі навіть на мобільному телефоні та масштабується до декількох рук [20].

Як приклади існуючих рішень можна привести два проекти щодо використання машинного навчання для відстеження рук: технологія, що розроблена Meta для окулярів віртуальної реальності Oculus Quest [19] та бібліотека Handtrack.js [18] для платформи машинного навчання з відкритим кодом TensorFlow.

Технологія від Meta для Oculus Quest дозволяє ефективно відстежувати рухи рук і пальців. Проте для роботи із цією розробкою потрібні окуляри віртуальної реальності Oculus Quest, які включають в себе набір з декількох інфрачервоних камер. Щодо технології бібліотеки Handtrack.js для платформи машинного навчання TensorFlow, вона не настільки детально визначає і відстежує кінцівки рук, проте для роботи з цією технологією потрібна лише веб-камера. На відміну від технології Meta, яка вже є повноцінним продуктом, Handtrack.js є лише базою для майбутніх розробок.

Проаналізувавши різні рішення для машинного навчання, які мають функціонал для відстеження рук, було обрано інструмент MediaPipe Hands. Ця платформа використовує декілька моделей машинного навчання, що працюють разом. Перша – це модель виявлення долоні, яка обробляє зображення і повертає орієнтовну рамку обмеження руки. Друга являє собою модель розмітки долоні, яка працює на обрізаній області зображення, що визначається детектором долоні, і повертає ключові точки долоні у просторі кадру з високою точністю.

Надання точно обрізаного зображення руки на модель розмітки долоні різко зменшує потребу у додаткових даних і замість цього дозволяє мережі присвятити більшу частину своїх можливостей для поліпшення точності прогнозування. Крім того, виділене зображення долоні також може генеруватися на основі орієнтирів рук, визначених у попередньому кадрі, і лише тоді, коли модель розмітки долоні

більше не може ідентифікувати присутність руки, викликається перша модель виявлення долоні.

Завдяки платформі MediaPipe на мові програмування python було створено три програми. Першою було розроблено інструмент розпізнавання та підрахунку пальців долоні. Програма у циклі за допомогою базового модулю отримує зображення, розпізнає і розмічає долоню та знаходить координати. Після знаходження пальців підраховується кількість відкритих кінцівок і виводиться їх кількість і так, доки програма не зупиниться. Щоб виявити, чи знаходиться палець у відкритому положенні, беруться координати його кінця та координати опорної точки цього ж пальця, але тієї, що знаходиться нижче. У процесі вони порівнюються, і за умови, якщо координати нижньої опорної точки показують, що вона вище, то палець вважається закритим і навпаки. Великий палець вважається згорнутим, коли його координати стали ліворуч чи праворуч відносно певної опорної точки для використаної руки.

Другою було створено програму для управління налаштуваннями гучності. Програма знаходить великий та вказівний пальці, а потім використовує їх взаємну відстань один від іншого для налаштування звуку. Коли пальці віддаляються, гучність збільшується, а коли наближаються – зменшується. Якщо пальці зімкнути, то звук повністю вимикається. Також модуль має функціонал відображення налаштувань гучності в реальному часі, який підраховується за відступом між пальцями.

Третьою розроблено програму для управління курсором комп'ютера. За допомогою даної технології можна використовувати руку замість комп'ютерної миші, тобто переміщати курсор екраном та натискати на необхідні елементи. Програма розпізнає два жести: один піднятий вказівний палець та два піднятих пальці (вказівний та середній). У першому випадку, коли піднятий лише один вказівний палець, вона буде знаходитись у режимі переміщення курсору по

робочій поверхні, у другому, а саме, коли буде додатково піднятий середній палець, програма перейде в режим натискання.

На прикладі даного програмного забезпечення можна зробити висновок, що отримані розробки можуть бути корисними для використання у навчальних програмах для дітей дошкільного віку, як варіант більш інтуїтивного засобу вводу, та, наприклад, для використання у якості інтерфейсу взаємодії з терміналами оплати та банкоматами.

Корисність даних розробок полягає у тому, що кожен день терміналами і банкоматами користуються сотні людей і для взаємодії з ними необхідно їх торкатися, а це, у свою чергу, може вести до поширенню різних захворювань. Тому використання таких програм у великому масштабі може позитивно вплинути на життя великої кількості людей, тим паче, що для її роботи потребується лише звичайна камера та процесор.

ЗМІСТ

ВСТУП.....	15
1 КЛАСИФІКАЦІЯ ЗАГАЛЬНИХ МЕТОДІВ ТА ПРОБЛЕМ МАШИННОГО НАВЧАННЯ ДЛЯ РОЗПІЗНАВАННЯ ЖЕСТІВ.....	17
2 ОГЛЯД СУЧАСНОГО ІНСТРУМЕНТАРІЮ ДЛЯ РОЗРОБКИ МОДЕЛЕЙ МАШИННОГО НАВЧАННЯ.....	26
2.1 Firebase ML Kit.....	26
2.2 TensorFlow.....	28
2.3 PyTorch.....	30
2.4 Apache MXNet.....	31
2.5 Eclipse Deeplearning4j.....	32
2.6 XGBoost.....	33
2.7 Paddle Paddle.....	35
2.8 Open Neural Networx Exchange.....	35
2.9 Аналіз існуючих рішень.....	36
2.9.1 Відстеження рук на Oculus Quest.....	36
2.9.2 Handtrack.js.....	38
2.9.3 Висновок з аналізу існуючих додатків.....	40

3 ВИБІР МОДЕЛЕЙ ТЕХНОЛОГІЙ МАШИННОГО НАВЧАННЯ ДЛЯ РОЗПІЗНАВАННЯ ЖЕСТІВ.....	42
3.1 Вибір мови програмування та середовища для реалізації моделей.....	42
3.2 Принципи роботи MediaPipe Hand для відстеження рук і пальців.....	42
3.2.1 Модель виявлення долоні.....	43
3.2.2 Модель розмітки руки.....	45
3.2.3 Вхідні параметри налаштування моделі.....	46
3.2.4 Огляд вихідних даних.....	47
4 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ВІДСТЕЖЕННЯ ДОЛОНЕЙ.....	48
4.1 Модуль відстеження руки.....	48
4.2 Програма розпізнавання та підрахунку пальців долоні.....	49
4.4 Програма для управління курсором.....	53
ВИСНОВКИ.....	56
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ.....	58
Додаток А	59
Додаток Б.....	65
Додаток В.....	68
Додаток Г.....	70

ВСТУП

Технологія комп'ютерного зору для розпізнавання жестів стає важливою частиною взаємодії людини та комп'ютера. В останні десятиліття клавіатура та миша відіграють значну роль у взаємодії людини з комп'ютером. Однак, через швидкий розвиток апаратного та програмного забезпечення, з'являється потреба в нових методах взаємодії. Зокрема, такі технології, як розпізнавання жестів, отримують велику увагу.

Актуальність дослідження полягає у тому, що технології машинного навчання дозволяють комп'ютерам та системам отримувати значну інформацію із цифрових зображень, відео та інших візуальних входів, і на основі цих вихідних даних програми можуть виконувати різноманітні дії. Комп'ютерний зір, що працює на основі машинного навчання, може використовуватися для розмітки фотографій у соціальних мережах, для радіологічної візуалізації в галузі охорони здоров'я та в самокерованих автомобілях. Здатність сприймати форму та рух рук за допомогою комп'ютерного зору може бути життєво важливим компонентом у покращенні взаємодії з користувачами в різних технологічних сферах та платформах. Наприклад, це може стати основою для розуміння мови жестів та управління жестами рук.

Метою кваліфікаційної роботи є розробка програмного забезпечення технології машинного навчання для розпізнавання жестів рукою для керування комп'ютером без використання клавіатури чи миші.

Для реалізації поставленої мети необхідно вирішити наступні завдання:

- аналіз існуючої проблеми, застосування машинного навчання для розпізнавання жестів та огляд літературних джерел;

- огляд сучасного інструментарію для розробки моделей машинного навчання з визначеної проблеми;
- вибір моделей для технологій машинного навчання;
- програмна реалізація запропонованих моделей.

Об'єктом дослідження є процес розпізнавання жестів рукою за допомогою моделей технології машинного навчання.

Предмет дослідження являє собою процес перетворення зображень руки які отримані з веб-камери на координатні точки

Донедавна для відстеження частин людського тіла в режимі реального часу були потрібні спеціальні 3D-датчики та висококласне програмне забезпечення. Цей високий бар'єр зробив створення та впровадження такого роду інтерактивних інструментів великим викликом для більшості розробників, особливо в середовищах з обмеженими ресурсами. На щастя, досягнення машинного навчання та розробка швидких моделей виявлення об'єктів тепер дають нам можливість виконувати відстеження, використовуючи лише зображення з камери. Поєднання цих досягнень із сучасними бібліотеками та фреймворками забезпечує ефективну основу для створення інтерактивного досвіду та відкриває людям можливість експериментувати з виявленням об'єктів та відстеженням рук.

Структура кваліфікаційної роботи складається із вступу, чотирьох розділів, висновків і переліку використаних джерел.

1 КЛАСИФІКАЦІЯ ЗАГАЛЬНИХ МЕТОДІВ ТА ПРОБЛЕМ МАШИННОГО НАВЧАННЯ ДЛЯ РОЗПІЗНАВАННЯ ЖЕСТІВ

Машинне навчання – це галузь штучного інтелекту та інформатики, яка зосереджена на використанні даних та алгоритмів для імітації способу навчання людей, поступово покращуючи його точність.

Машинне навчання є важливою складовою зростаючої галузі науки про дані. За допомогою статистичних методів алгоритми навчаються робити класифікації або прогнозування, розкриваючи ключові моменти в рамках проектів з обробки даних.

Оскільки глибоке навчання та машинне навчання, як правило, використовуються як взаємозамінні поняття, варто зазначити нюанси між ними. Машинне навчання, глибоке навчання та нейронні мережі – це всі підгалузі штучного інтелекту. Однак глибоке навчання насправді є підполем машинного навчання, а нейронні мережі – підполем глибокого навчання.

Спосіб, яким відрізняються глибоке навчання та машинне навчання, полягає в тому, як навчається кожен алгоритм. Поглиблене навчання автоматизує більшу частину процесу вилучення особливостей, усуваючи деякі необхідні втручання людини вручну та дозволяючи використовувати більші масиви даних. Класичне, або "неглибоке", машинне навчання більше залежить від втручання людини у навчання. Людські експерти визначають набір функцій, щоб зрозуміти відмінності між введеними даними, як правило, для вивчення потрібні більш структуровані дані.

Глибоке машинне навчання може використовувати мічені набори даних, також відомо як контрольоване навчання, для інформування свого алгоритму, але це не обов'язково вимагає маркованого набору даних. Воно може поглинати неструктуровані дані в необробленому вигляді (наприклад, текст, зображення), а

також може автоматично визначати набір функцій, які відрізняють різні категорії даних один від одного. На відміну від машинного навчання, воно не вимагає втручання людини для обробки даних, що дозволяє масштабувати машинне навчання більш цікавими способами. Поглибленому навчанню та нейронним мережам в першу чергу приписують прискорення прогресу в таких областях, як комп'ютерний зір, обробка природної мови та розпізнавання мови.

Нейронні мережі або штучні нейронні мережі складаються з шарів вузлів, що містять вхідний рівень, один або кілька прихованих шарів, і вихідний рівень. Кожен вузол, або штучний нейрон, з'єднується з іншим і має відповідну вагу та поріг. Якщо вихід будь-якого окремого вузла перевищує вказане порогове значення, цей вузол активується, надсилаючи дані на наступний рівень мережі. В іншому випадку дані не передаються на наступний рівень мережі. "Глибина" в глибокому навчанні просто стосується глибини шарів нейронної мережі. Нейронну мережу, яка складається з більш ніж трьох шарів, які включали б вхідні та вихідні дані, можна вважати алгоритмом глибокого навчання або глибокою нейронною мережею. Нейронна мережа, яка має лише два-три шари, є лише основною нейронною мережею [4].

Роботу алгоритму машинного навчання можна поділити на три основні частини:

- Процес прийняття рішень: загалом алгоритми машинного навчання використовуються для прогнозування або класифікації. На основі деяких вхідних даних, які можуть бути позначені або не позначені, ваш алгоритм створить оцінку щодо закономірності в даних.
- Функція помилки: функція помилки служить для оцінки передбачення моделі. Якщо є відомі приклади, функція помилки може провести порівняння для оцінки точності моделі.
- Процес оптимізації моделі: якщо модель може краще відповідати точкам даних навчального набору, то ваги коригуються, щоб

зменшити невідповідність між відомим прикладом та оцінкою моделі. Алгоритм повторюватиме цей процес оцінки та оптимізації, оновлюючи ваги автономно, доки не буде досягнуто поріг точності.

Класифікувати машинне навчання можна у три основні категорії:

а) контрольоване навчання, також відоме як кероване машинне навчання, визначається використанням маркованих наборів даних для навчання алгоритмів, які дозволяють класифікувати дані або точно прогнозувати результати. Коли вхідні дані подаються в модель, вона регулює свою вагу, поки модель не буде встановлена належним чином. Це відбувається як частина процесу перехресної перевірки, щоб гарантувати, що модель дозволяє уникнути перетренування або недотренування. Навчання під контролем допомагає організаціям вирішувати різноманітні масштабні проблеми, наприклад класифікувати спам в окремій папці з папки "Вхідні".

б) Навчання без нагляду, також відоме як машинне навчання без нагляду, використовує алгоритми машинного навчання для аналізу та кластеризації не розмічених наборів даних. Ці алгоритми виявляють приховані шаблони або групування даних без потреби втручання людини. Його здатність виявляти схожість та розбіжності в інформації роблять це ідеальним рішенням для дослідницького аналізу даних, стратегій перехресних продажів, сегментації клієнтів, розпізнавання зображень та зразків. Він також використовується для зменшення кількості об'єктів у моделі за допомогою процесу зменшення розмірності; Аналіз основних компонентів та декомпозиція особливих значень - два загальних підходи для цього.

в) Навчання під наглядом пропонує дещо середнє між навчанням під наглядом та без нагляду. Під час навчання він використовує менший набір маркованих даних для керівництва класифікацією та вилучення ознак із більшого, немаркованого набору даних. Навчання під наглядом може вирішити проблему недостатньої кількості маркованих даних (або неможливості дозволити собі

маркування достатньої кількості даних) для підготовки керованого алгоритму навчання.

Розвиток технологій машинного навчання безсумнівно, полегшив життя людей. Однак впровадження машинного навчання на підприємствах також викликало низку етичних проблем, пов'язаних із технологіями машинного навчання. Деякі з них включають:

– Технологічна сингулярність. Неможливо думати, що автомобіль без водія ніколи не потрапить в дорожньо-транспортну пригоду, але хто відповідальний за таких обставин досі не вирішене питання. Чи слід продовжувати розробляти повністю автономні транспортні засоби, чи треба обмежити інтеграцію цієї технології для створення лише напіваавтономних транспортних засобів, що сприяють підвищенню безпеки серед водіїв? Саме такі етичні суперечки відбуваються в міру розвитку нових, інноваційних технологій штучного інтелекту.

– Вплив штучного інтелекту на робочі місця. Багато громадського уваги навколо штучного інтелекту пов'язаного з можливістю втрати роботи, це занепокоєння, мабуть, слід переформулювати. З кожною руйнівною новою технологією бачимо, що ринковий попит на певні робочі ролі змінюється. Наприклад, коли розглядаємо автомобільну промисловість, багато виробників, як General Motors, переходять до фокусу на виробництво електромобілів, щоб задовольнити зелені ініціативи [9]. Енергетична галузь не зникає, але джерело енергії переходить від палива до електрики. Подібним чином слід розглядати штучний інтелект, де штучний інтелект містить попит на роботу в інших сфери. Потрібні будуть люди, які допоможуть керувати цими системами, оскільки дані зростають і змінюються щодня. Все ще будуть потрібні ресурси для вирішення більш складних проблем у галузях, на які найімовірніше вплинуть зміни в попиті на роботу, наприклад, обслуговування клієнтів. Важливим аспектом штучного інтелекту та його впливу на ринок праці буде сприяння переходу людей до цих нових сфер ринкового попиту.

– Конфіденційність. Конфіденційність, як правило, обговорюється в контексті конфіденційності даних, захисту даних та безпеки даних, і ці занепокоєння дозволили політикам досягти тут більших успіхів за останні роки. Наприклад, у 2016 році було створено законодавство GDPR для захисту персональних даних людей у Європейському Союзі та Європейському економічному просторі, надаючи фізичним особам більше контролю над своїми даними [9]. У США окремі штати розробляють політику, таку як Каліфорнійський закон про конфіденційність споживачів, який вимагає від підприємств інформувати споживачів про збір своїх даних. Це недавнє законодавство змусило компанії переосмислити, як вони зберігають та використовують особисті дані. Як результат, інвестиції в безпеку стають дедалі більшим пріоритетом для бізнесу, оскільки вони прагнуть усунути будь-які вразливі місця та можливості для нагляду, злому та кібератак.

– Упередженість та дискримінація. Випадки упередженості та дискримінації в ряді інтелектуальних систем викликали багато етичних питань щодо використання штучного інтелекту. Як можливо захистити себе від упередженості та дискримінації, коли самі дані навчання можуть піддаватися упередженості. Хоча компанії, як правило, мають не погані наміри навколо своїх зусиль щодо автоматизації, існують деякі непередбачувані наслідки включення машинного навчання в практику найму. У своїх зусиллях автоматизувати та спростити процес, Amazon ненавмисно упереджує потенційних кандидатів на роботу за статтю для відкритих технічних ролей, і зрештою їм довелося скасувати проект. Подібно до таких подій, Harvard Business Review порушило інші гострі питання щодо використання штучного інтелекту в процесі найму, наприклад, які дані ви повинні мати можливість використовувати при оцінці кандидата на посаду.

Упередженість та дискримінація також не обмежуються функцією людських ресурсів, її можна знайти в ряді додатків - від програмного забезпечення для розпізнавання облич до алгоритмів соціальних мереж.

Оскільки підприємства усвідомлюють ризики, пов'язані зі штучним інтелектом, вони також активізують цю дискусію навколо етики та цінностей штучного інтелекту. Наприклад, у 2019 році генеральний директор IBM Арвінд Крішна сказав, що IBM припинила свою діяльність із загальновизнаного розпізнавання та аналізу обличчя IBM, наголосивши, що "IBM твердо виступає проти і не погоджується з використанням будь-якої технології, включаючи технологію розпізнавання обличчя, запропоновану іншими продавцями, спостереження, расове профілювання, порушення основних прав і свобод людини або будь-яка мета, яка не узгоджується з нашими цінностями та Принципами довіри та прозорості " [9].

– Підзвітність. Оскільки законодавства, яке регулює практику штучного інтелекту, не існує законодавства, немає реального механізму правозастосування, який би гарантував, що використовується етичний штучний інтелект. Нинішні стимули для компаній дотримуватися цих вказівок є негативними наслідками неетичної системи штучного інтелекту. Щоб заповнити прогалину, етичні рамки виникли як частина співпраці між етиками та дослідниками для управління, побудови та розподілу моделей штучного інтелекту в суспільстві. Однак на даний момент вони служать лише керівництвом, а дослідження показують, що поєднання розподіленої відповідальності та відсутність передбачення потенційних наслідків не обов'язково сприяє запобіганню шкоді суспільству .

Виходячі з огляду, майже необмежена кількість доступних даних, доступне зберігання даних та зростання менш дорогих та потужніших засобів обробки даних сприяли зростанню актуальності машинного навчання. Зараз багато галузей розробляють більш надійні моделі машинного навчання, здатні аналізувати більші та складніші дані, забезпечуючи при цьому швидші та точніші результати в

широких масштабах. Інструменти машинного навчання дозволяють організаціям швидше виявляти вигідні можливості та потенційні ризики.

Практичне застосування машинного навчання зумовлює результати, які можуть суттєво вплинути на результат роботи компаній, підприємств та життя звичайних людей. Нові методи в цій галузі швидко розвиваються і розширюють застосування машинного навчання майже до безмежних можливостей. Галузі, які залежать від великої кількості даних - і їм потрібна система для їх ефективного та точного аналізу, сприйняли машинне навчання як найкращий спосіб побудови моделей, стратегій та планування.

Машинне навчання вже використовується в багатьох областях. Наприклад у маркетинговому секторі багато компаній успішно впровадили машинне навчання, щоб збільшити та підвищити задоволеність споживачів. Сайти електронної комерції та соціальних медіа використовують машинне навчання, щоб проаналізувати історію ваших покупок та пошуку та надати рекомендації щодо інших предметів для придбання, виходячи з ваших минулих звичок. Багато експертів вважають, що майбутнє роздрібної торгівлі визначатиметься за допомогою штучного інтелекту та машинного навчання, оскільки бізнес-програми глибокого навчання стають ще більш вправними у збірці, аналізі та використанні даних для персоналізації досвіду покупок та розробки індивідуальних цільових маркетингових кампаній. Інтернет-чат-боти замінюють людей на шляху клієнта. Вони відповідають на часто задавані питання щодо таких тем, як доставка, або надають персоналізовані поради, здійснюють перехресні продажі товарів або пропонують розміри для користувачів, змінюючи наше мислення про залучення клієнтів на веб-сайтах та платформах соціальних медіа. Прикладом можуть бути боти для обміну повідомленнями на сайтах електронної комерції з віртуальними агентами, програми обміну повідомленнями, такі як Slack та Meta Messenger, а також завдання, які зазвичай виконують віртуальні помічники та голосові помічники.

Використовуючи дані про минулу поведінку споживача, алгоритми можуть допомогти виявити тенденції даних, які можуть бути використані для розробки більш ефективних стратегій перехресних продажів. Це використовується для надання відповідних додаткових рекомендацій клієнтам під час оформлення замовлення в інтернет-магазинах.

У сфері транспортування ефективність та точність є ключовими для прибутковості, так само здатність прогнозувати та пом'якшувати потенційні проблеми. Функції аналізу та моделювання даних машинного навчання ідеально поєднуються з бізнесом у сферах доставки, громадського транспорту та вантажного транспорту. Машинне навчання використовує алгоритми для пошуку факторів, які позитивно та негативно впливають на успіх ланцюга поставок, роблячи машинне навчання критично важливим компонентом управління ланцюгами поставок. В рамках логістики машинне навчання полегшує можливість планувальників оптимізувати вибір оператора, рейтинг, маршрутизацію та процеси контролю якості, що економить гроші та підвищує ефективність. Здатність машинного навчання одночасно аналізувати тисячі точок даних та застосовувати алгоритми швидше, ніж будь-яка людина, дозволяє машинному навчанню вирішувати проблеми, які люди ще не визначили.

Поширення датчиків та пристроїв, які можна носити, контролюючи все: від частоти пульсу та кроків до рівня кисню та цукру та навіть режимів сну створили значний обсяг даних, що дозволяє лікарям оцінювати стан здоров'я своїх пацієнтів у режимі реального часу. Один новий алгоритм машинного навчання виявляє ракові пухлини на мамографіях, інший визначає рак шкіри, третій може аналізувати зображення сітківки для діагностики діабетичної ретинопатії [2].

Багато мобільних пристроїв включають розпізнавання мови в свої системи для здійснення голосового пошуку або спрощення доступу до текстових повідомлень.

Технології машинного навчання дозволяє комп'ютерам та системам отримувати значну інформацію із цифрових зображень, відео та інших візуальних входів, і на основі цих входів вона може вживати заходів. Ця здатність надавати рекомендації відрізняє її від завдань розпізнавання зображень. Комп'ютерний зір, що працює на основі машинного навчання, може використовуватися для розмітки фотографій у соціальних мережах, радіологічної візуалізації в галузі охорони здоров'я та самокерованих автомобілів в автомобільній промисловості.

2 ОГЛЯД СУЧАСНОГО ІНСТРУМЕНТАРІЮ ДЛЯ РОЗРОБКИ МОДЕЛЕЙ МАШИННОГО НАВЧАННЯ

2.1 Firebase ML Kit

Firebase ML Kit - це мобільний SDK від Google для Android та iOS, який дозволяє використовувати машинне навчання програмах. Для початку також не потрібно мати знань про машинне навчання. ML Kit постачається з набором готових до використання API для загальних випадків використання, і потрібно лише кілька рядків коду, щоб впровадити ці API у програми [10].

Нині майже все цифрове газети, статті, книги. Отже можна письмовий та надрукований текст оцифрувати або навіть перекласти. API розпізнавання тексту ML Kit дозволяє розпізнавати будь-яку мову, що базується на латині, у текстовій формі. Ця функція може бути використана для автоматизації точного введення даних для кредитних або візитних карток. Ви можете сфотографувати квитанцію, а програма на вашому смартфоні може оплатити її. Хмарний API дозволяє розробникам витягувати текст із більш складних зображень документів. Тоді цей текст можна використовувати для перекладу документів або збільшення доступності.

Також API ML Kit дозволяє розпізнавати предмети на зображеннях. Немає необхідності надавати будь-які додаткові метадані, а розпізнавання доступне також у режимі офлайн (лише базове) та у хмарі. Використовуючи цю функцію, ви можете виконати автоматичне формування метаданих та модерацію вмісту. Зберігати, групувати або позначати фотографії зараз дуже просто.

Визнання орієнтиру дозволяє розпізнати добре відомі орієнтири на зображенні. Крім цього, також вказуються геокоординати орієнтиру. Тому можна побачити, де було зроблено цей знімок. Цю інформацію можна використовувати

для генерації метаданих, надаючи користувачам персоналізований досвід. Або у своєму додатку для туристичного путівника, щоб допомогти своєму користувачеві отримати інформацію про найближче місце або вигляд.

API розпізнавання обличчя обличчя використовується для виявлення обличчя на зображеннях. Ця функція використовується в режимі офлайн. За допомогою цього користувачі API можуть витягувати обличчя із зображень та редагувати їх за допомогою різних фільтрів або генерувати аватари з фотографій. ML Kit надає можливість застосувати функцію розпізнавання обличчя в режимі реального часу, і її можна легко інтегрувати у відеочати.

API Виявлення та відстеження об'єктів, який працює на пристрої, допоможе користувачам локалізувати та відстежувати об'єкти на зображенні або в реальному часі з камери. Виявлений об'єкт можна віднести до обраної загальної категорії. Якщо користувачам потрібно створити досвід візуального пошуку в реальному часі, функція виявлення та відстеження об'єктів може бути вигідною. Коли всі об'єкти виявлені, користувачі можуть відправити їх у хмарний сервер або власну модель, якщо розробники побудують цю модель.

Існує безліч рішень для сканування штрих-коду, але більшість із них вимагали певного положення телефону. Це сканування виконується на пристрої, і підключення до Інтернету не потрібно. Ця функція дуже зручна для виявлення закодованих даних і може робити це в різних положеннях та під різними кутами штрих-коду. Забезпечуючи набагато кращий досвід використання.

API ідентифікації мови дозволяє ідентифікувати мову з одного текстового рядка. Ця функція може бути особливо корисною для перекладачів чи істориків, яким потрібно з'ясувати, якою мовою написано зображення чи документ.

API перекладу можна використовувати для перекладу тексту до 59 мов, а користувачі можуть переключатися між мовами або використовувати API ідентифікації мови для виявлення мови. API перекладу використовує ті самі моделі, що й офлайн-програма Google Translate.

Smart Reply API генерує пропозиції відповідей на основі всієї розмови. І ця функція може створити повну пропозицію. Це дозволяє користувачам швидко відповідати на повідомлення, оскільки всі відповіді генеруються автоматично. На даний момент підтримується лише англійська мова, і цю функцію можна використовувати лише у випадкових розмовах, але, безсумнівно, вона може допомогти користувачеві в таких програмах як чати.

Якщо існуючі API набору ML Kit не надають необхідних функцій, тоді ви можете створити власні моделі машинного навчання. Додаткові власні моделі можуть стати дуже важкими для програми та зробити програму занадто великою. Поточний ліміт становить 40 МБ для моделі, що може бути трохи проблематичним для складних рішень. Якщо модель все ще занадто важка, можливо залишити її в хмарі, а потім організувати динамічне завантаження в програму. Це дозволяє зменшити розмір програми, і користувачі швидше завантажать програму. Окрім цього, оновлення моделей також є динамічними, і це важливо. Це означає, що модель буде оновлена, тоді як вся програма не буде оновлена.

Firebase ML Kit - це дуже потужний інструмент, який можна легко інтегрувати та перенести додаток на новий сучасний рівень, надаючи корисні функції для користувачів.

2.2 TensorFlow

TensorFlow - одна з найпопулярніших систем машинного навчання та глибокого навчання, яку використовують розробники та дослідники. Спочатку запущений в 2007 році командою Google Brain, TensorFlow дозрів і став повноцінною платформою машинного навчання [11]. Це виходить за рамки навчання для підтримки підготовки даних, розробки особливостей та обслуговування моделей.

TensorFlow може працювати на стандартному процесорі та спеціалізованих прискорювачах штучного інтелекту, включаючи GPU та TPU. Він доступний на 64-розрядних платформах Linux, macOS, Windows та мобільних обчислювальних платформах, включаючи Android та iOS. Моделі, навчені TensorFlow, можна застосовувати на комп'ютерах, браузерах і навіть мікроконтролерах. Ця широка підтримка робить TensorFlow унікальним продуктом.

Основну бібліотеку TensorFlow можна встановити як модуль Python, що працює на платформах AMD та ARM. TensorFlow.js - це бібліотека JavaScript для навчання та розгортання моделей у браузері та на Node.js. Для мобільних пристроїв, пристроїв IoT та розподілених пристроїв TensorFlow Lite можна використовувати для взаємодії з моделями.

TensorFlow Extended (TFX) - це платформа для впровадження машинного навчання, що охоплює весь робочий процес, необхідний для збору даних, прийому даних, перевірки даних, навчання моделі, аналізу моделі, розгортання та виведення. Підприємства використовують TFX для реалізації проектів машинного навчання.

У TensorFlow 1.x код був написаний для визначення обчислювального графіку з подальшим виконанням. Це було громіздко і трудомістко для розробників, що впроваджують нейронні мережі. Запуск TensorFlow 2.0 приніс дві суттєві зміни - стрімке виконання та інтеграцію з Keras. Стрімке виконання дозволяє виконувати код без визначення обчислювального графіка, що спрощує процес розробки та налагодження. Keras, високорівневий API машинного навчання, інтегрований з TensorFlow 2.0, забезпечуючи знайомий робочий процес визначення нейронної мережі та її навчання.

TensorFlow Hub - це колекція попередньо навчених моделей, які розробники можуть використовувати для висновків у різних середовищах, включаючи хмару, комп'ютер, браузер та в розподіленому обчисленні.

TFServe є компонентом платформи TensorFlow, гнучкої, високопродуктивної системи обслуговування для моделей машинного навчання, розробленої для виробничих середовищ. Він може обслуговувати декілька моделей у стандартному форматі для надання великомасштабної послуги взаємодії з платформою.

Незалежно від того, що розробляється комп'ютерний зір чи обробка природної мови, TensorFlow - це зріла та надійна платформа машинного навчання з великими можливостями.

2.3 PyTorch

PyTorch - це система глибокого навчання з відкритим кодом, створена бути гнучкою та модульною для досліджень, зі стабільністю та підтримкою, необхідною для розгортання виробництва [12]. Він заснований на Torch, структурі для виконання швидких обчислень, спочатку написаній на C.

У порівнянні з іншими системами глибокого навчання, PyTorch має більш коротку криву навчання для розробників Python. Хоча інтерфейс Python вважається більш досконалим і основною мовою розробки, PyTorch також має інтерфейс C ++. Meta інвестував в інший фреймворк, який називається Convolutional Architecture for Fast Feature Embedding (Caffe2), який був об'єднаний з PyTorch у 2018 році.

З моменту своєї першої версії PyTorch реалізував стрімке виконання, що надихнуло TensorFlow 2.0. Однією з переваг PyTorch є його сумісність з NumPy. Перетворення об'єктів NumPy в тензори вбудовано в основу основних структур даних PyTorch. Розробники можуть легко переключатися між тензорними об'єктами PyTorch та масивами NumPy.

Впровадження нейронної мережі в PyTorch простіше та більш інтуїтивне, ніж в інших фреймворках. Завдяки підтримці процесорів і графічних процесорів, складні нейронні мережі можна навчити за допомогою великих наборів даних.

Meta і Amazon Web Services (AWS) співпрацювали над розробкою TorchServe, механізму з відкритим кодом для моделей PyTorch. TorchServe забезпечує легке обслуговування з низькою затримкою, щоб забезпечити високопродуктивну взаємодію. Він включає обробники за замовчуванням для найпоширеніших програм, таких як виявлення об'єктів та класифікація тексту, тому розробникам не потрібно писати власний код для розгортання моделей.

TorchElastic - це інструмент з відкритим кодом для навчання глибоких нейронних мереж різного масштабу за допомогою Kubernetes. Це дозволяє розподілені навчальні роботи PyTorch виконувати відмовостійку та еластичну манеру навчання. Контролер TorchElastic для Kubernetes (TECK) - це вбудована реалізація Kubernetes інтерфейсу PyTorch Elastic, який автоматично управляє життєвим циклом Kubernetes послуг, необхідних для навчання TorchElastic.

PyTorch Hub надає можливості для вивчення попередньо підготовлених моделей та експериментів з ними.

2.4 Apache MXNet

Apache MXNet, випущений у 2017 році, є одним із останніх учасників екосистеми глибокого навчання [13]. Його унікальність полягає у підтримці різних мов, включаючи C ++, Python, Java, Julia, Matlab, JavaScript, Go, R, Scala, Perl та Wolfram Language.

Apache MXNet був розроблений Карлосом Гостріном з Вашингтонського університету та дослідниками з Університету Карнегі Меллона. Amazon вибрала

його як бажану основу для глибокого навчання для створення комерційних продуктів та керованих пропозицій платформи машинного навчання AWS. Сьогодні більшість попередньо побудованих моделей та алгоритмів, доступних на Amazon SageMaker, реалізовані за допомогою Apache MXNet.

Для розробників Python MXNet надає комплексний та гнучкий API для розробників з різним рівнем досвіду та широкими вимогами. Подібно до того, як Keras надає зручний для розробників API високого рівня для TensorFlow, Apache MXNet виставляє Gluon API, який забезпечує чистий, простий API для глибокого навчання. Gluon має спеціалізовані API, GluonCV, GluonNLP та GluonTS, призначені для комп'ютерного зору, обробки природної мови та аналізу часових рядів.

Apache MXNet може використовувати як процесори, так і графічні процесори для навчання. При використанні в хмарному середовищі він використовує переваги масштабованої інфраструктури графічного процесора, пропонованих Amazon EC2. Він тісно інтегрований з Horovod, розподіленим інструментом глибокого навчання від Uber, для підтримки широко розподіленого навчання GPU. Багатомодельний сервер із відкритим кодом може бути використаний для обслуговування моделей, навчених Apache MXNet.

2.5 Eclipse Deeplearning4j

Deeplearning4j - один з небагатьох фреймворків машинного навчання, написаних на Java, націлених на віртуальну машину Java. Він розроблений групою розробників машинного навчання, що базується в Сан-Франциско, і комерційно підтримується стартапом Skymind. Deeplearning4j був переданий Фонду Eclipse у жовтні 2017 року [14]. Бібліотека сумісна з Clojure та Scala.

Для кластеризації та розподіленого навчання Deeplearning4j інтегрований з Apache Spark та Apache Hadoop. Він також інтегрований із середовищем

виконання NVIDIA CUDA для виконання операцій на графічному процесорі та розподіленого навчання по кількох графічних процесорах.

Deeplearning4j включає n-мірний клас масиву з використанням ND4J, що дозволяє проводити наукові обчислення в Java та Scala, порівняно з функціями, які NumPy надає Python. Він може бути ефективно використаний як бібліотека для виконання лінійної алгебри та маніпуляцій з матрицями для навчання та умовиводу.

Deeplearning4j може бути використаний для навчання моделей, які можуть виконувати класифікацію зображень, виявлення об'єктів, сегментацію зображень, обробку природної мови та прогнозування часових рядів.

Сервер моделей SKIL також може імпортувати моделі з фреймворків Python, таких як Tensorflow, Keras, Theano та CNTK. Хоча Deeplearning4j порівняно менш популярний, ніж TensorFlow та PyTorch, він набирає популярності серед розробників Java.

2.6 XGBoost

XGBoost - найкраща бібліотека для навчання моделей, заснованих на лінійній регресії та класифікації. XGBoost розшифровується як eXtreme Gradient Boosting, техніка, яка часто використовується під контролем навчання при навчанні моделей на основі високоструктурованих наборів даних. Тянькі Чень створив його у Вашингтонському університеті в травні 2014 року [15]. Цікаво, що він також є співавтором Apache MXNet.

XGBoost написаний на C++ з інтерфейсами для Python, R, Julia та Java. Він інтегрований із Scikit-learn як один із підтримуваних модельних алгоритмів. XGBoost заснована на техніці додавання нових моделей, які передбачають залишки або помилки попередніх моделей, а потім додаються разом, щоб зробити

остаточне прогнозування. Він використовує алгоритм градієнтного спуску, щоб мінімізувати втрати при додаванні нових моделей.

Хоча основи глибокого навчання дають кращі результати при навчанні комп'ютерному зору, обробці природної мови та моделям часових рядів, вони є надмірним для навчання наборів даних з лінійною роздільністю. XGBoost найкраще працює, коли навчає моделі на основі менших наборів даних з відсутніми даними. Він неявно виконує техніку, звану імпутацією даних, яка компенсує відсутні значення в наборі даних.

XGBoost можна навчити на менш дорогих багатоядерних процесорах замість розподіленого кластера графічних процесорів. Коли обсяг даних обмежений, XGBoost виявляється найбільш ефективною та доступною бібліотекою машинного навчання для підготовки точних моделей.

Коли дані гарно структуровані, що представляють одновимірний масив, для навчання моделей використовується комбінація Pandas та XGBoost. Зображення та відео, які перекладаються у багатовимірні масиви або тензори, найкраще працюють із фреймворками для глибокого навчання, такими як TensorFlow та PyTorch.

Для структурованого, табличного, одновимірного набору даних із відсутніми значеннями, і потрібна неперевершена швидкість виконання на центральних процесорах з високою точністю прогнозування, XGBoost гарний вибір.

2.7 Paddle Paddle

PaddlePaddle (PArallel Distributed Deep LEarning) - це незалежна платформа глибокого навчання з відкритим кодом, запущена Baidu у 2016 році [16]. Це проста у використанні, ефективна, гнучка та масштабована платформа глибокого навчання. Спочатку вчені та інженери розробили її, щоб застосувати глибоке

навчання в багатьох продуктах Baidu, таких як обробка природної мови, переклад та комп'ютерне бачення.

PaddlePaddle підтримує широкий спектр архітектур нейронних мереж та алгоритмів оптимізації. За допомогою PaddlePaddle можна використовувати багато центральних чи графічних процесорів, щоб пришвидшити навчання, досягаючи високої пропускну здатності та продуктивності завдяки оптимізованій комунікації між процесами.

PaddlePaddle пропонує 146 алгоритмів і вдосконалив понад 200 моделей попередньої підготовки, деякі з яких мають відкритий код, що сприяє швидкому розвитку додатків. На платформі також розміщені набори інструментів для передових дослідницьких цілей, такі як Paddle Quantum для квантових обчислювальних моделей та Paddle Graph Learning для моделей навчання заснованих на графах. Розробники Python можуть легко розпочати роботу з фреймворком, встановивши спеціальній модуль.

2.8 Open Neural Network Exchange

Проект Open Neural Network Exchange був ініційований AWS, Meta та Microsoft у 2017 році. У 2019 році він був прийнятий в Linux Foundation AI [17].

Хоча ONNX і не є наскрізним фреймворком, таким як TensorFlow або PyTorch, він заслуговує на увагу інженерів та користувачів машинного навчання. ONNX забезпечує взаємодію з моделями, навченими різними системами глибокого навчання. Наприклад, модель, навчену в PyTorch, можна експортувати в ONNX, яку можна імпортувати в TensorFlow для обробки.

ONNX має три компоненти - внутрішній рівень, оптимізований для роботи з прискорювачами штучного інтелекту, таких як Intel OpenVINO Toolkit та NVIDIA TensorRT, процес, який може робити висновок про моделі ONNX; набір інструментів для експорту та імпорту моделей з однієї платформи в іншу.

Корпорація Майкрософт має середовище виконання ONNX з відкритим кодом та додала підтримку навчання моделей. Ця можливість перетворила ONNX на повноцінну платформу, яка може навчати моделі та виводити результат.

ONNX доступний для розробників як бібліотека Python. Після встановлення легко експортувати моделі з одного формату в інший. PyTorch та Apache MXNet мають власні інструменти для експорту моделей у формат ONNX. У ONNX Model Zoo є попередньо навчені моделі із зору, мови та мовлення.

ONNX намагається зменшити фрагментацію систем глибокого навчання завдяки взаємодії з іншими платформами. Безперечно, це крок у правильному напрямку, який позитивно вплине на екосистему штучного інтелекту.

2.9 Аналіз існуючих рішень

2.9.1 Відстеження рук на Oculus Quest

Дослідники та інженери з Meta Reality Labs та Oculus розробили єдину систему відстеження рук для VR, яка повністю покладається на монохромні камери. У системі не використовується активна технологія зондування глибини або будь-яке додаткове обладнання (наприклад, інструментальні рукавички). Ця технологія застосовується як частина програмного забезпечення для Oculus Quest, бездротової автономної гарнітури VR, яка доступна для споживачів [19].

Використовуючи чотири камери Quest у поєднанні з новими методами глибокого навчання та відстеження на основі моделей, досягається великий обсяг взаємодії для ручного відстеження. Обробка виконується повністю на пристрої, і система оптимізована для підтримки жестів для взаємодії.

Глибокі нейронні мережі використовуються для прогнозування розташування рук людини, а також орієнтирів, таких як суглоби рук. Потім ці

орієнтири використовуються для реконструкції 26-ти ступенів свободи рук і пальців людини. Результатом є 3D-модель, яка включає конфігурацію та геометрію поверхні кисті. API дозволять розробникам використовувати ці 3D-моделі, щоб увімкнути нові механізми взаємодії в своїх програмах або керувати користувальницьким інтерфейсом.

Використовується нова архітектура відстеження, яка робить точну оцінку рук з низьким рівнем тремтіння, яка надійно працює в широкому діапазоні середовищ, а також ефективну квантовану структуру нейронної мережі, яка забезпечує відстеження рук у реальному часі на мобільному процесорі без шкоди виділеним ресурсам. для користувацьких програм.

Точне відстеження рук відкриє цілий ряд нових вражень, а також спрощує та поліпшує досвід використання Oculus Quest. Наприклад, люди можуть зробити паузу у фільмі у віртуальній реальності лише одним жестом, і висловитись більш природно в соціальних додатках. У корпоративних програмах викладач міг би проводити навчальний клас на основі VR, не маючи необхідності підтримувати парк спарених, заряджених контролерів.

У більш широкому сенсі, відстеження рук зробить VR більш природним та інтуїтивно зрозумілим, а також допоможе розробникам створити нові способи взаємодії людей у віртуальних світах.

2.9.2 Handtrack.js

Мета бібліотеки Handtrack.js для TensorFlow - абстрагуватися від кроків, пов'язаних із завантаженням файлів моделі, забезпечити корисні функції та дозволити користувачеві виявляти руки на зображенні без попереднього досвіду машинного навчання [18]. Не потрібно навчати модель, і не потрібно експортувати графі або збережені моделі. Можливо просто розпочати роботу, включивши Handtrack.js у веб-програму та викликавши бібліотечні методи.

Робота Handtrack.js забезпечується згортковою нейронною мережею, яка виконує завдання виявлення об'єкта - на основі зображення є можливість ідентифікувати наявність об'єктів (рук) та їх розташування (обмежувальне поле). Він був побудований з використанням інструментів екосистеми Tensorflow – переважно Tensorflow Python та Tensorflow.js. Tensorflow.js дозволяє розробникам навчати моделі (онлайн) безпосередньо у браузері або імпортувати попередньо навчені моделі (в автономному режимі) для виводу у браузері.

Загалом процес побудови Handtrack.js можна розділити на 4 основні етапи - збір даних, навчання моделі, перетворення моделі та групування і виведення моделей. Перші 3 кроки, які зосереджені на навчанні моделі, можна виконувати в автономному режимі, використовуючи різні серверні мови (наприклад, Tensorflow python та Tensorflow node.js). Останній крок, який зосереджений на завантаженні моделі в браузер для виведення в Інтернет, реалізований за допомогою Tensorflow.js.

- Збір даних: Набір даних Egohands складається з 4800 зображень людської руки з анотаціями обмежувальної рамки в різних налаштуваннях (внутрішній, зовнішній, егоцентрична точка зору), зроблених за допомогою пристрою Google Glass. Кожне зображення позначено полігонами розташування рук. Кожен багатокутник спочатку очищається, а потім перетворюється на обмежувальну рамку.
- Навчання моделі: Далі модель виявлення об'єктів навчається за допомогою API виявлення об'єктів Tensorflow. Архітектура SSD Mobilenet, оптимізована для швидкості та розгортання в середовищах з обмеженими ресурсами, була обрана для підтримки низької затримки. Навчальний процес виконується за допомогою пітон-коду Tensorflow і може запускатися на локальному графічному процесорі або за допомогою ресурсів хмарних обчислень, таких як Google Compute Engine, Google ML Engine та Cloudera Machine Learning.

- Перетворення моделі: Отримана навчена модель потім перетворюється у формат, який можна завантажити у браузер для виведення. Tensorflow.js пропонує перетворювач моделей, який дозволяє перетворити попередньо навчені моделі у формат веб-моделі Tensorflow.js. На даний момент він підтримує такі формати моделей: модулі Tensorflow savedModel, модель Keras та Tensorflow Hub. Розмір моделі Handtrack.js становить 18,5 Мб після перетворення і розділена на 5 файлів вагів моделі та файл маніфесту.
- Групування моделей: Цей крок зосереджений на створенні бібліотеки javascript з основними методами, які завантажують перетворену модель та виконують вивід (передбачення) з урахуванням вхідного зображення. Зокрема, метод Tensorflow.js loadGraphModel використовується для завантаження перетвореної моделі, тоді як метод executeAsync використовується для виведення. Для зручності розповсюдження бібліотека також була опублікована на npm - таким чином розробники javascript, які створюють свої програми за допомогою інструментів збірки, можуть імпортувати менеджери пакетів бібліотек, такі як npm або yarn. Крім того, зменшена версія бібліотеки також доступна через jsdelivr.

Машинне навчання в браузері – захоплююча і зростаюча сфера. Використовуючи такі інструменти, як Tensorflow.js, розробники javascript можуть легко використовувати навчені моделі Tensorflow безпосередньо в браузері та створювати цікаві, інтерактивні враження. Машинне навчання у браузері також приносить такі привабливі переваги, як конфіденційність, простота розповсюдження та зменшена затримка (в деяких випадках). З розвитком активних дослідницьких областей, таких як стиснення моделей, квантування моделей, дистиляція моделей, стає все більш можливим ефективно запускати складні, дуже точні моделі на обмежених ресурсах пристроїв, таких як браузери. Крім того, досягнення веб-стандартів для прискорених обчислень (наприклад, WebGPU, WebAssembly) означають, що програми на базі браузера стають

швидшими. Все це вказує на майбутнє, де моделі машинного навчання (подібно до тривимірної веб-графіки) можна ефективно виконувати у браузері.

2.9.3 Висновок з аналізу існуючих додатків

Виявлено визначення поняття машинне навчання та його основні проблеми. Здатність сприймати форму та рух рук може бути життєво важливим компонентом у покращенні взаємодії з користувачами в різних технологічних сферах та платформах. Наприклад, це може стати основою для розуміння мови жестів та управління жестами рук, а також може забезпечити накладання цифрового вмісту та інформації поверх фізичного світу в доповненій реальності.

Було проаналізовано два рішення щодо використання машинного навчання для відстеження рук: технологія що розроблена Meta для окулярів віртуальної реальності Oculus Quest для та бібліотека Handtrack.js для платформи машинного навчання з відкритим кодом TensorFlow.

Технологія від Meta для Oculus Quest дозволяє ефективно відстежувати рухи рук і пальців, але для роботи із цією розробкою потрібні окуляри віртуальної реальності Oculus Quest. Бібліотека Handtrack.js для платформи машинного навчання TensorFlow не настільки детально визначає і відстежує кінцівки рук. Проте для роботи з цією технологією потрібна лише веб-камера. На відміну від технології Meta, яка вже є повноцінним продуктом, Handtrack.js є лише базою для майбутніх розробок.

3 ВИБІР МОДЕЛЕЙ ТЕХНОЛОГІЙ МАШИННОГО НАВЧАННЯ ДЛЯ РОЗПІЗНАВАННЯ ЖЕСТІВ

3.1 Вибір мови програмування та середовища для реалізації моделей

Python – це мова програмування з відкритим кодом. Вона використовує концепцію об'єктно-орієнтованого програмування, яка допомагає програмістам розробляти великі програми, розділяючи їх на менші будівельні блоки. Він був розроблений Гідо ван Россумом в 1990 році [3]. Однією з основних причин, популярності Python, є наявність численних модулів, які є загальнодоступними для комерційного використання. Крім того, вивчення python порівняно легше, ніж інші мови програмування, через його меншу складність у синтаксисі та семантиці. На сьогоднішній день популярним використанням python є data science, машинне навчання, веб-розробка тощо.

Одним з найкращих IDE для Python є PyCharm. Причиною цього є те, що PyCharm був спеціально створений для Python, тому всі плагіни, розширення та додаткові функції спеціально спрямовані на покращення процесу кодування за допомогою Python. PyCharm попередньо інтегрований з різними серіями бібліотек Python та базами даних, включаючи Oracle та MySQL. Він також має дуже корисний інструмент автоматичного доповнення, який, як тільки ви розумієте, що треба зробити з кодом, може пришвидшити час, який витрачається на написання та редагування коду.

3.2 Принципи роботи MediaPipe Hand для відстеження рук і пальців

MediaPipe Hands – це високоякісне рішення для відстеження рук і пальців. Він використовує машинне навчання, щоб вивести 21 тривимірний орієнтир руки

лише з одного кадру. У той час як інші сучасні підходи в першу чергу покладаються на потужні робочі середовища для виведення, цей метод досягає хорошого результату в реальному часі навіть на мобільному телефоні і навіть масштабується до декількох рук .

MediaPipe Hands використовує декілька моделей машинного навчання, що працюють разом [20]. Модель виявлення долоні, яка обробляє зображення і повертає орієнтовну рамку обмеження руки. Модель розмітки долоні, яка працює на обрізаній області зображення, визначеній детектором долоні, і повертає ключові точки у тривимірному просторі з високою точністю.

Надання точно обрізаного зображення руки на модель розмітки долоні різко зменшує потребу у додаткових даних і замість цього дозволяє мережі присвятити більшу частину своїх можливостей для поліпшення точності прогнозування. Крім того, виділене зображення долоні також може генеруватися на основі орієнтирів рук, визначених у попередньому кадрі, і лише тоді, коли модель розмітки долоні більше не може ідентифікувати присутність руки, для переміщення руки викликається перша модель виявлення долоні.

3.2.1 Модель виявлення долоні

Виявлення долоні є надзвичайно складною задачею: модель повинна працювати на різних розмірах долонь з великим діапазоном масштабу щодо кадру зображення і мати можливість виявляти перекриті та самоперекриті частини долоні. Тоді як обличчя мають високо контрастні візерунки, наприклад, в області очей та рота, відсутність таких рис у руках утруднює їх надійне виявлення лише за їх візуальними особливостями. Натомість надання додаткового контексту, наприклад, руки , тіла чи особистих рис, допомагає точній локалізації рук.

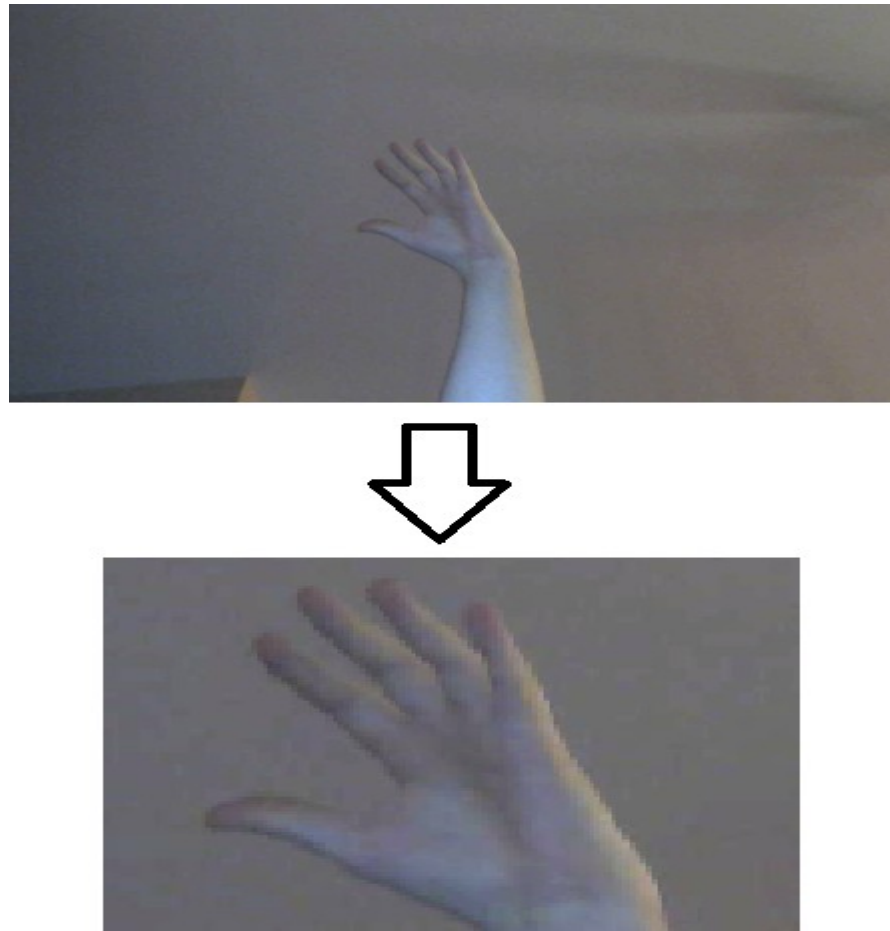


Рисунок 3.1 – Приклад роботи моделі виявлення долоні

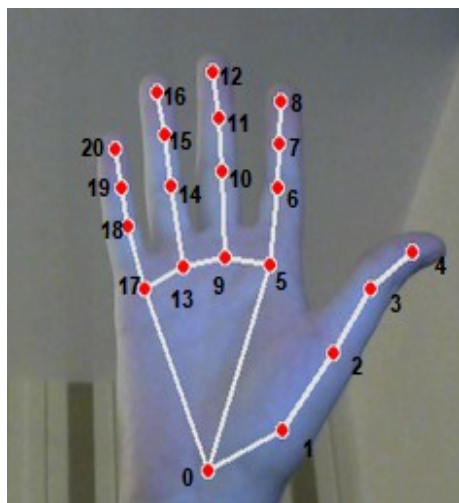
Даний метод вирішує вищезазначені проблеми, використовуючи різні стратегії. По-перше, тренується детектор долоні замість детектора руки, оскільки оцінка границь цільних предметів, таких як долоні та кулаки, значно простіша, ніж виявлення долоні з пальцями які рухаються. Окрім того, оскільки долоні є меншими об'єктами, алгоритм добре працює навіть у випадках коли долоня перекриваю руку, таких як рукостискання. Більше того, долоні можна представити за допомогою квадратних обмежувальних границь (якорів у термінології машинного навчання), ігноруючи інші пропорції, і отже, зменшуючи кількість якорів у 3-5 разів. По-друге, екстрактор функцій кодера-декодера використовується для розширеного усвідомлення контексту сцени навіть для

невеликих об'єктів. Наприкінці, мінімізуються фокальні втрати під час тренувань для підтримки великої кількості якорів, що виникають внаслідок великої дисперсії масштабу.

За допомогою вищезазначених методів досягається середня точність у 95,7% при виявленні долоні. Використання методу перехресної ентропії без декодера дає базовий рівень лише в 86,22% [20].

3.2.2 Модель розмітки руки

Після виявлення долоні на зображенні наступна модель розмітки руки виконує точну локалізацію ключових точок в межах виявлених областей рук за допомогою регресії, тобто прямого прогнозування координат. Модель засвоює послідовне представлення внутрішньої пози долоні і є стійкою навіть до частково закритих рук та само оклюзій.



- | | |
|-----------------------|-----------------------|
| 0. WRIST | 11. MIDDLE_FINGER_DIP |
| 1. THUMB_CMC | 12. MIDDLE_FINGER_TIP |
| 2. THUMB_MCP | 13. RING_FINGER_MCP |
| 3. THUMB_IP | 14. RING_FINGER_PIP |
| 4. THUMB_TIP | 15. RING_FINGER_DIP |
| 5. INDEX_FINGER_MCP | 16. RING_FINGER_TIP |
| 6. INDEX_FINGER_PIP | 17. PINKY_MCP |
| 7. INDEX_FINGER_DIP | 18. PINKY_PIP |
| 8. INDEX_FINGER_TIP | 19. PINKY_DIP |
| 9. MIDDLE_FINGER_MCP | 20. PINKY_TIP |
| 10. MIDDLE_FINGER_PIP | |

Рисунок 3.2 – Ключові точки долоні

Для отримання результату, розробники моделі вручну розмітили близько 30 тис. реальних зображень [6]. Щоб краще охопити можливі пози рук і забезпечити

додатковий огляд за природою геометрії рук, також було надано високоякісну синтетичну модель руки на різних фонах і розмічено відповідні координати.

3.2.3 Вхідні параметри налаштування моделі

Розглянемо вхідні параметри для налаштування моделі:

– `static_image_mode`. Якщо встановлено значення `false`, рішення розглядає вхідні зображення як відеопотік. Він спробує виявити руки на перших вхідних зображеннях, а після успішного виявлення додатково локалізує орієнтири рук. На наступних зображеннях, як тільки всі руки кількості яких дорівнює `max_num_hands` виявляються і відповідні орієнтири долоней локалізуються, він просто відстежує ці орієнтири, не викликаючи іншого виявлення, поки не втратить слід руки. Це зменшує затримку і ідеально підходить для обробки відеокadrів. Якщо встановлено значення `true`, розпізнавання рук запускається на кожному вхідному зображенні, ідеально підходить для обробки партії статичних, можливо, не пов'язаних між собою зображень.

– `max_num_hands`. Максимальна кількість рук для виявлення. За замовчуванням 2.

– `min_detection_confidence`. Мінімальне значення впевненості від 0.0 до 1.0 з моделі розпізнавання рук, щоб виявлення вважалось успішним. За замовчуванням 0.5.

– `min_tracking_confidence`. Мінімальне значення довіри від 0.0 до 1.0 з моделі відстеження орієнтира для долоней, які слід вважати успішно відстеженими, або інакше виявлення рук буде автоматично викликане на наступному вхідному зображенні. Якщо встановити його на більш високе значення, це може підвищити надійність рішення за рахунок більшої затримки. Ігнорується, якщо `static_image_mode` має значення `true`, де виявлення рук просто запускається на кожному зображенні. За замовчуванням 0.5.

3.2.4 Огляд вихідних даних

Розглянемо види даних які є результатом роботи моделі :

– `multi_hand_landmarks`. Колекція виявлених або відстежених рук, де кожна рука представлена у вигляді списку з 21 орієнтира рук, і кожен орієнтир складається з `x`, `y` та `z`. `x` та `y` нормалізуються від 0.0 до 1.0 відповідно до ширини та висоти зображення. `z` являє собою глибину орієнтира, причому глибина у зап'ястя є початком, і чим менше значення, тим ближче орієнтир до камери. Величина `z` використовує приблизно такий же масштаб, що і `x`.

– `multi_handedness`. Колекція виявлених або відстежуваних рук з виявленням це ліва чи права рука. Кожен елемент який представляю долоню складається з `label` та `score`. Параметр `label` - це параметр із значенням "left" або "right". Параметр `score` - це передбачувана ймовірність рука ліва чи права. Виявленням це ліва чи права визначається за умови, що вхідне зображення відображається дзеркально, тобто зроблено за допомогою фронтальної чи селфі-камери. Якщо це не так треба поміняти місцями вихідні дані.

4 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ВІДСТЕЖЕННЯ ДОЛОНЕЙ

4.1 Модуль відстеження руки

Цей модуль відповідає за такі функції як визначення правильного сигналу з веб-камери і його обробку таку, як знаходження, відслідковування та відображення “скелету” руки і максимальну кількість підтримуємих рук. Він також використовується як основа програми, щоб коли треба дізнатися координати будь якої змінної яка відповідає за якусь кінцівку долоні ми могли зробити лише зробити запит і отримати координати. Це дозволяє швидше створювати нові програми для відстеження долоней без необхідності переписати код кожний раз.

Модуль включає в себе такі функції:

- функція ініціації, яка відповідає за роботу методів із модулю `mediaripre` та до якої передаються такі параметри, як максимальна кількість рук які відстежуються, впевненість виявлення яка відповідає за те при якій впевненості алгоритма у тому, що він знайшов руку буде відбувати новий пошук долоні, параметр впевненості у відстеженні руху, та параметр який відповідає за те буде відбуватися пошук руки кожний новий кадр, чи тільки коли буде втрачено руку.
- функція знаходження руки, яка приймає зображення з камери та розмічає орієнтири долоні.
- функцію знаходження координат, яка знаходить координати кожної з 21 точки долоні на яку вона розмічена. Функція отримує вже розмічене

- зображення, його висоту і ширину і відносно цих даних видаю точні координати запитуваної точки та видає їй ідентифікаційний номер.
- головну функцію, яка відповідає за отримання зображення з камери, та у циклі обробляти кадри за допомогою функцій знаходження руки і знаходження координат, а також відображення швидкості обробки кадрів за допомогою відображення лічильника кадрів за секунду поверх вихідного зображення.

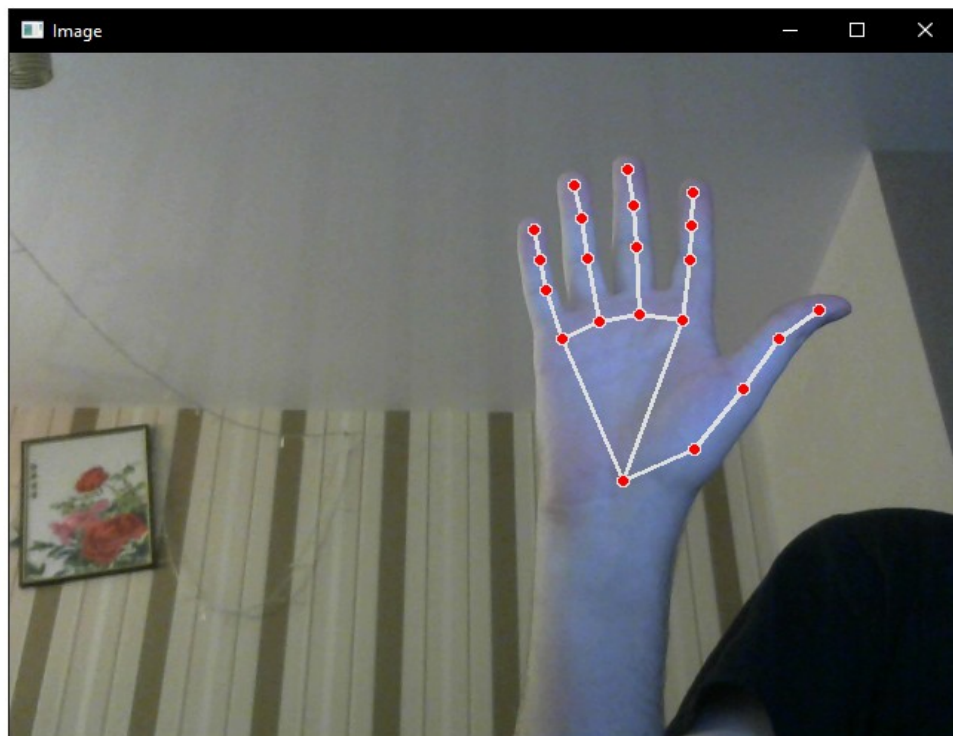


Рисунок 4.1 – Приклад роботи функції знаходження координат

4.2 Програма розпізнавання та підрахунку пальців долоні

Дана програма знаходить кінцівки пальців та підраховує, а потім відображає їх кількість цифрою в інтерфейсі та зображенням з цифрою.

Програма у циклі за допомогою базового модулю отримує зображення, розпізнає і розмічає долоню та знаходить координати.

Щоб виявити чи знаходиться палець у відкритому положенні беруться координати його кінця та координати опорної точки цього ж пальця, але яка знаходиться нижче, вони порівнюються і якщо координати нижньої опорної точки показують, що вона вище то палець вважається закритим і навпаки, якщо координати вищої опорної точки виявляються вище, то палець вважається відкритим.



Рисунок 4.2 – Приклад розпізнавання двох пальців

Для великого пальця ця логіка не працює, бо його дуже важко зігнути вниз, частіше ми згортаємо його всередину руки. З цього робимо висновок, що для

виявлення згорнутий палець чи ні нам потрібно координати не висоти, а широти. Тобто палець вважається згорнутим, коли його координати стали ліворуч чи праворуч відносно певної опорної точки для використаної руки.

4.3 Програма управління налаштуваннями гучності

Програма знаходить великий та вказівний палець, а потім використовує їх взаємну відстань один від іншого для налаштування звуку. Коли пальці віддаляються гучність збільшується, а коли наближаються зменшується. Якщо пальці зімкнути та звук повністю вимикається. Також модуль має функціонал відображення гучності в реальному часі яка підраховується по відступу між пальцями.

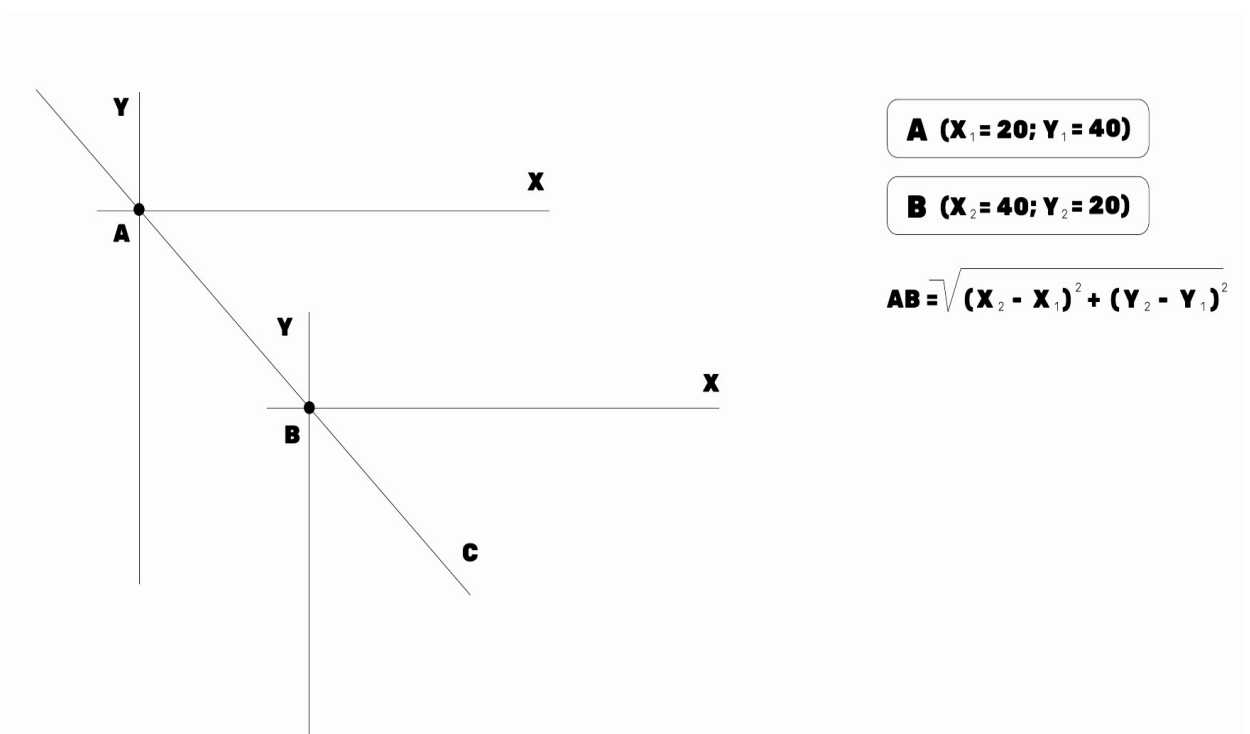


Рисунок 4.4 – Схема розрахунку довжини лінії між двох точок

Програма для регулювання гучності використовує відстань між координатами великого і вказівного пальця. Розмір лінії між пальцями

визначається завдяки функції знаходження гіпотенузи. По перше знаходяться різниця між координатами висоти і широти, а потім ця різниця використовується у якості довжини катетів уявного прямокутного трикутника і завдяки теоремі Піфагора кожного нового кадру знаходиться довжина уявної лінії між вказівним і великим пальцем.

Для контролю встановлюється максимальний і мінімальний розмір лінії між пальцями. Коли довжина лінії сягає мінімальної звук повністю відключається. Максимальна межа використовується, щоб програма розуміла ріст довжини відносно умовної шкали для того щоб зміни узагалі відбувалися.

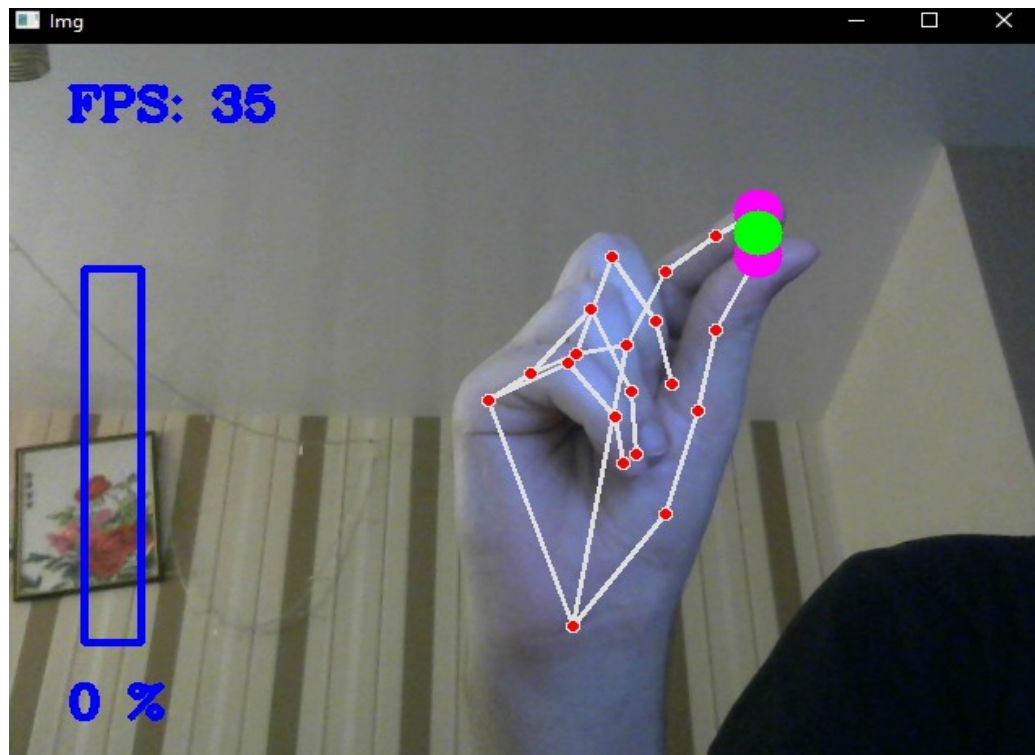


Рисунок 4.5 – Приклад повного відключення звуку

Для кращого візуального сприйняття кінці пальців на відеовиході позначаються точками і між ними малюється лінія, також всередині цієї лінії

також є точка, яка представляє собою мінімальний розмір лінії і при її досягненні звук повністю виключається.

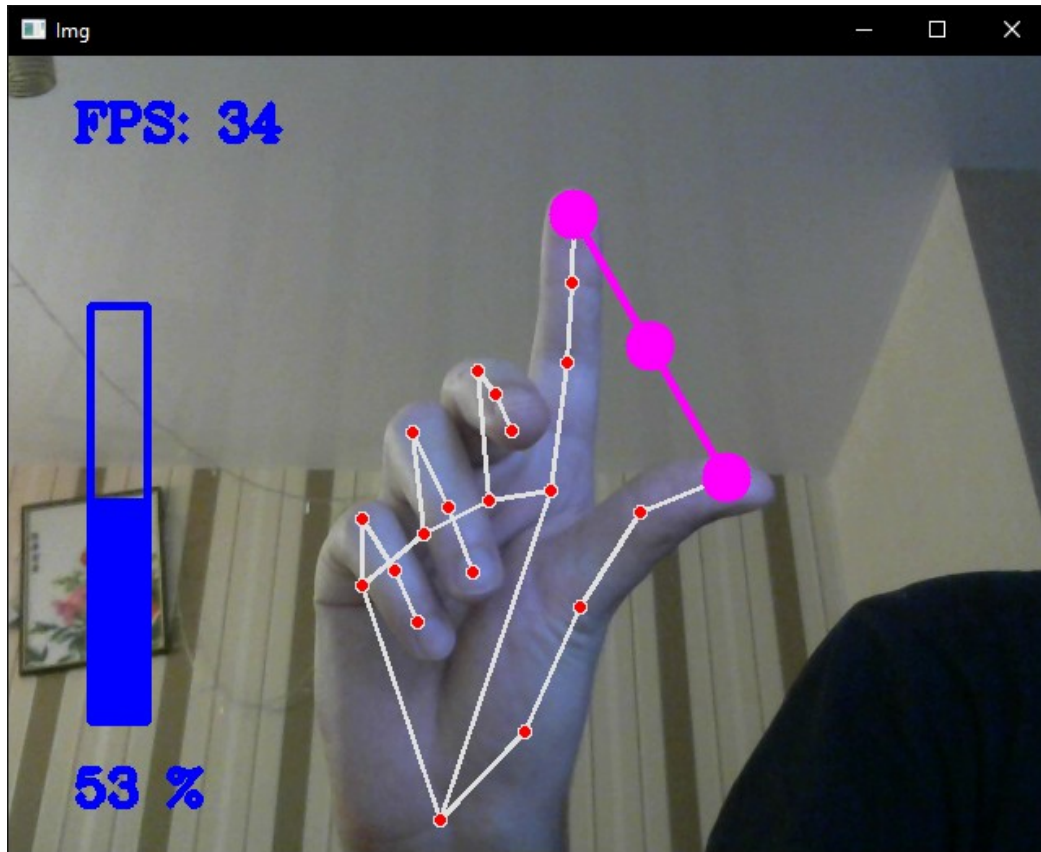


Рисунок 4.6 – Приклад регулювання звуку

4.4 Програма для управління курсором

За допомогою даної програми можна використовувати руку замість комп'ютерної миші, тобто переміщати курсор по екрану та натискати на необхідні елементи.

По-перше, за допомогою базового модулю відстеження руки програма знаходить і розмічає долоню, потім бере точні координати вказівного та середнього пальців. Програма розпізнає коли піднятий лише один вказівний

палець, то вона буде знаходитись у режимі переміщення курсору по робочій поверхні, а коли буде додатково піднятий середній палець програма перейде в режим натискання.



Рисунок 4.7 – Приклад режиму натискання

Режим натискання працює завдяки визначенню відстані між вказівним та середнім пальцем, коли відстань менше певного значення то спрацьовує функція, яка відповідає за емуляцію кліку комп'ютерної миші.

В режимі руху курсором координати ширини з камери інвертуються, бо зображення з камери дзеркальне, а значить при русі пальця вправо, координати отримані з камери будуть вказувати на протилежний напрям відносно напрямку руху пальця користувача.

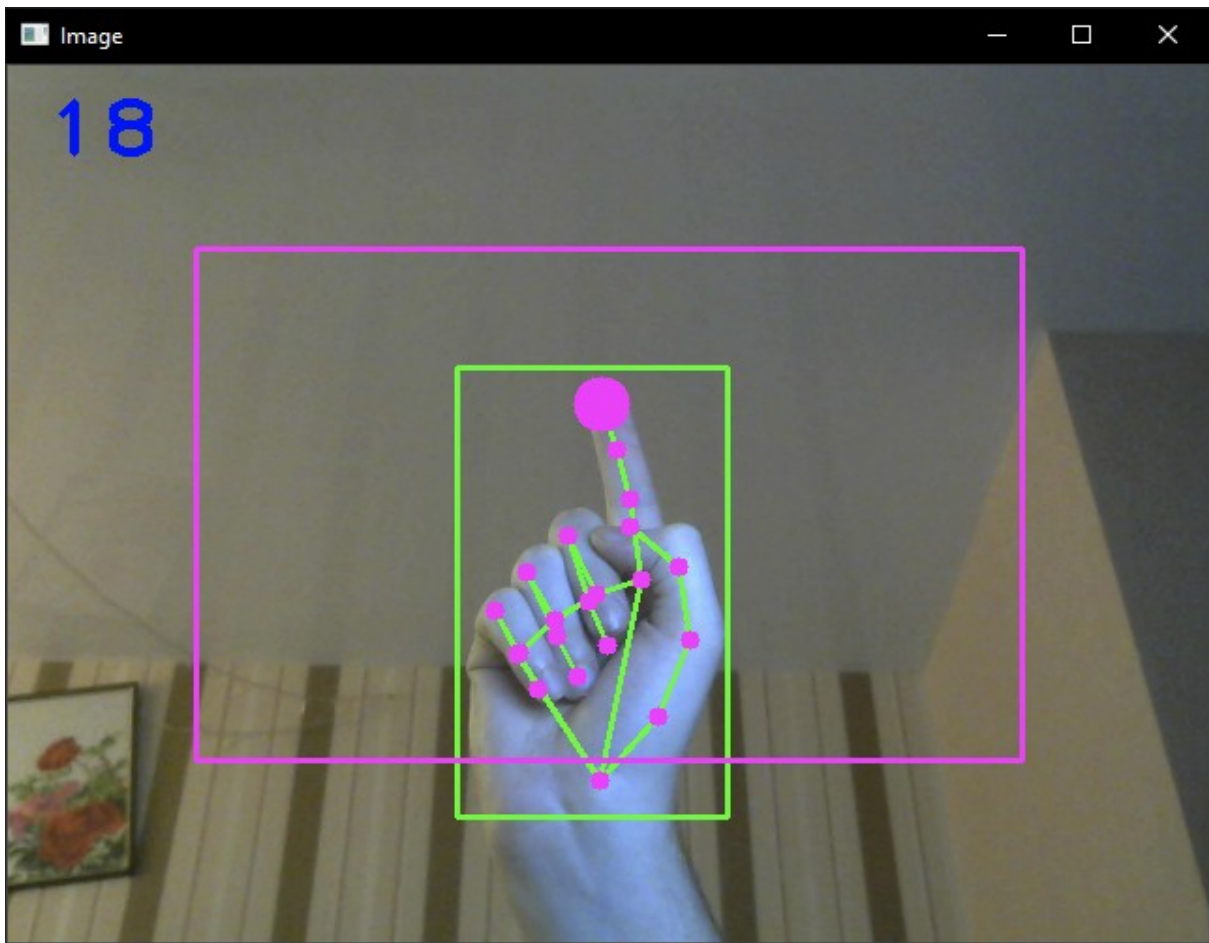


Рисунок 4.8 – Приклад режиму руху курсору

Для того, щоб у режимі руху курсор не рухався занадто різко використовується функція яка робить рух плавнішим завдяки обробці змінних які відповідають за координати пальця. Функція руху миші використовує вже нові значення координат.

ВИСНОВКИ

В ході виконання кваліфікаційної роботи було розроблено програмне забезпечення технології машинного навчання для розпізнавання жестів рукою для керування комп'ютером без використання клавіатури чи миші. У результаті огляду літературних джерел проведено аналіз існуючої проблеми, розглянуто сфери застосування машинного навчання для розпізнавання жестів. Досліджено сучасний інструментарій для використання технологій машинного навчання у визначеній специфіці. Зокрема було розглянуто платформу MediaPipe, а саме, її особливості та стратегії вирішення проблем машинного навчання за допомогою цієї розробки. Перевагами даної технології можна назвати високу точність сканування долоні та помірне використання ресурсів комп'ютера, яке дозволяє використовувати для обробки вхідного зображення центральний процесор замість відеокарти, що дає можливість запускати програму на більшій кількості пристроїв. Також було проаналізовано технічні параметри розробки, які будуть використовуватися при створенні проекту.

Було написано програмне забезпечення технології машинного навчання за допомогою мови програмування python та модуля MediaPipe. Воно ідентифікує пальці долоні та використовує їх як засіб вводу інформації або як один з методів керування комп'ютером, наприклад, керування гучністю за допомогою руху пальців замість класичних методів вводу, таких як комп'ютерна миша або екран з тач-інтерфейсом. Це доводить, що дані розробки можуть бути корисними для використання у навчальних програмах для дітей дошкільного віку як більш інтуїтивний засіб вводу, так і, наприклад, для використання у якості інтерфейсу взаємодії з терміналами оплати та банкоматами. На сьогодні існує та практично застосовується технологія від Meta для Oculus Quest, яка дозволяє ефективно

відстежувати рухи рук і пальців. Проте для роботи із цією розробкою потрібні окуляри віртуальної реальності Oculus Quest.

Корисність розробки полягає у тому, що кожен день терміналами і банкоматами користуються сотні людей, і для взаємодії з пристроями необхідно їх торкатися, а це, у свою чергу, може сприяти поширенню різних захворювань. Тому використання таких програм у великому масштабі може позитивно вплинути на життя великої кількості людей, тим паче, що для її роботи потребується лише звичайна камера.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Кушаков П.К. Поліпшення використання додатку «Шкільний асистент» за допомогою нейромереж. Об'єднані наукою: перспективи міждисциплінарних досліджень: матеріали VII всеукраїнської науково-практичної конференції студентів, аспірантів та молодих вчених., м. Київ, 12-13 листопада 2020 р. Київ, 2020. С. 144-145.
2. Кушаков П.К. Поліпшення якості медичного обслуговування за допомогою нейромереж. Радіоелектроніка та молодь у XXI столітті: матеріали 25-го міжнародного молодіжного форуму., м. Харків, 20-22 квітня 2021 р. Харків, 2021. С. 197-198.
3. Downey A. Think Python: How to Think Like a Computer Scientist, 2nd edition. Needham: Green Tea Press, 2015. 36 с.
4. Raschka S., Mirjalili V. Python Machine Learning. Livery Place: Packt Publishing Ltd, 2019. 296 с.
5. Bradski G., Kaehler A. Learning OpenCV: Computer Vision with the OpenCV Library. Sebastopol: O'Reilly Media, Inc, 2011. 53 с.
6. Solem J. E. Programming Computer Vision with Python: Tools and Algorithms for Analyzing Images. Sebastopol: O'Reilly Media, Inc, 2012. 101 с.
7. Lakshmanan V., Görner M., Gillard R. Practical Machine Learning for Computer Vision. Sebastopol: O'Reilly Media, Inc, 2021. 201 с.
8. James A., Edward R. Talking Nets An Oral History of Neural Networks. Cambridge: The MIT Press, 2020. 10 с.
9. Christian B. The Alignment Problem: Machine Learning and Human Values. New York: W. W. Norton & Company, 2020. 25 с.

10. ML Kit for Firebase. URL: <https://firebase.google.com/docs/ml-kit> (дата звернення: 10.11.2021).
11. Introduction to TensorFlow. URL: <https://www.tensorflow.org/learn> (дата звернення: 11.10.2021).
12. PyTorch documentation. URL: <https://pytorch.org/docs/stable/index.html> (дата звернення: 11.11.2021).
13. Features. URL: <https://mxnet.apache.org/versions/master/features> (дата звернення: 11.11.2021).
14. Eclipse Deeplearning4j URL: <https://projects.eclipse.org/proposals/eclipse-deeplearning4j> (дата звернення: 11.11.2021).
15. XGBoost Documentation URL: <https://xgboost.readthedocs.io/en/stable/> (дата звернення: 11.11.2021).
16. Introduction to Tensor URL: https://www.paddlepaddle.org.cn/documentation/docs/en/guides/index_en.html (дата звернення: 11.11.2021).
17. About ONNX URL: <https://onnx.ai/about.html> (дата звернення: 11.11.2021).
18. Handtrack.js: tracking hand interactions in the browser using Tensorflow.js and 3 lines of code URL: <https://blog.tensorflow.org/2019/11/handtrackjs-tracking-hand-interactions.html> (дата звернення: 11.11.2021).
19. How Oculus Quest Hand Tracking Technology Works URL: <https://www.online-tech-tips.com/gaming/how-oculus-quest-hand-tracking-technology-works/> (дата звернення: 11.11.2021).
20. Exquisite hand and finger tracking in web browsers with MediaPipe's machine learning models URL: <https://towardsdatascience.com/exquisite-hand-and-finger-tracking-in-web-browsers-with-mediapipe-machine-learning-models-2c4c2beee5df> (дата звернення: 11.11.2021).