Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

Кафедра мультимедійних систем факультету інформатики

Проектування та реалізація методів взаємодії на основі жестів у мультимедійних програмах, таких як безконтактні інтерфейси, для підвищення взаємодії та контролю користувачів

Текстова частина до кваліфікаційної роботи
за спеціальністю «Комп'ютерні науки» 122

Керівник кваліфікаційної роботи

доц. Афонін А.О.

_____

(підпис)

"____" _____ 2024 р.

Виконав студент КН-4

Черніков А.А.

"____" _____ 2024 р.

Київ – 2024

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

ЗАТВЕРДЖУЮ
Зав.кафедри мультимедійних систем,
Доцент., к. ф.-м. н. О.П. Жежерун
_____(підпис)
“____”_____2023 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ
на кваліфікаційну роботу

студенту 4-го року БП КН факультету інформатики

Чернікову Андрію Антоновичу

Тема: Проектування та реалізація методів взаємодії на основі жестів у мультимедійних програмах, таких як безконтактні інтерфейси, для підвищення взаємодії та контролю користувачів

Зміст ТЧ кваліфікаційної роботи:
Анотація
Вступ
1. Технологія взаємодії на основі жестів
2. Теоретичні підходи до виявлення позиції руки
3. Інструменти для розроблення Chrome додатку з жестовою взаємодією
4. Реалізація методів для збільшення контролю користувачів
Висновки
Список літератури

Дата видачі “__” _____ 2023 р. Керівник _____
(підпис)

Завдання отримав _____
(підпис)

Календарний план виконання кваліфікаційної роботи:

№ п/п	Назва етапу курсової роботи	Термін виконання етапу	Примітка
1.	Узгодження теми для кваліфікаційної роботи	06.10.2023	
2.	Пошук та аналіз джерел за темою роботи	06.10.2023 – 19.01.2024	
3.	Проектування методів взаємодії на основі жестів	25.01.2024 – 14.02.2024	
3.	Розроблення браузерного розширення	01.03.2024 – 22.03.2024	
4.	Складання плану для текстової частини роботи	25.03.2024 – 27.03.2024	
5.	Написання тексту роботи	02.04.2024 – 18.04.2024	
6.	Завершення програмної та текстової частини роботи	23.04.2024 – 27.04.2024	
7.	Захист кваліфікаційної роботи	30.05.2024	

Зміст

Анотація

У цій роботі описано реалізацію програми розширення для браузера на основі технології розпізнавання жестів, яка надає безконтактний інтерфейс для медіаплеєру YouTube та уможливлює жестову взаємодію між користувачем та обчислювальним пристроєм.

Вступ

Процес взаємодії людини із комп’ютером став невід’ємною частиною нашого життя. Від якості, та зручності, якого, залежить ефективність виконання повсякденних завдань, отримання нових навичок та загалом необхідної інформації. Останніми роками технологія розпізнавання жестів набирає популярності завдяки своїй здатності забезпечувати більш інтуїтивну взаємодію з цифровими пристроями. Однак, кількість та якість програм, які підтримують цю технологію, ще досить обмежена.

Метою цієї роботи є розширення можливих способів комунікації із комп’ютером, через додавання інтерфейсу розпізнавання жестів для збільшення доступності та контролю споживання медіаконтенту.

Завданням роботи є використання сучасних готових рішень для розроблення методів, що підтримують взаємодію користувача із комп’ютером з розпізнаванням безконтактних команд через відеокамеру.

Результатом роботи є розроблена програма-розширення для браузера, яка додає функціонал у контрольну панель відеоплеєра YouTube у вигляді кнопки, що дає змогу запускати та зупиняти процес зчитування жестових команд користувача, які впливають на перегляд відео.

Наукова новизна роботи полягає у тому, що аналогічних додатків для збільшення контролю користувачів у медіапрограмах - немає.

У першому розділі розглянуто галузь комп’ютерного зору, її підгалузь - технологію розпізнавання жестів, та перелік пристроїв, в яких основна взаємодія відбувається через використання жестових команд.

Другий розділ присвячено теоретичним аспектам розвитку жестової взаємодії та основним алгоритмам роботи нейронних мереж, на яких вона побудована.

У третьому розділі розглянуто розширення для Chrome браузера, як застосунок, що дає змогу доступитися до програм із медіаконтентом.

Представлено готове рішення для розпізнавання жестів у вигляді моделі MediaPipe.

Четвертий розділ є останнім та описує реалізацію методів розробленого застосунку. В ньому надано перелік команд, які відповідають за керування переглядом відео, та розглянуто обмеження застосунку.

1 Технологія взаємодії на основі жестів

1.1 Комп’ютерний зір

Людське око, це складний механізм, який функціонує, як досконалий оптичний прилад, що відповідає за виконання багатьох завдань. Комплексна робота нервів та сітківки ока надсилає сигнали мозку, які інтерпретуються людьми для ідентифікації найдрібніших деталей, спектру кольорів, їхньої відстані від спостерігача та загальному положенні об’єктів у просторі. В такий же спосіб комп’ютерний зір надає можливість машинам виконувати всі ці процеси.

Комп’ютерний зір - це напрям інформаційних технологій, що використовує штучний інтелект (ШІ) для аналізу візуальних даних, таких як фотографії чи відео, для отримання важливої інформації. Ці дані проходять обробку і використовуються для автоматизації команд, які відтворюють люди. Якщо штучний інтелект дає можливість обчислювальним машинам «мислити», то КЗ дає змогу їм «бачити» [1].

Сенсорний пристрій фіксує зображення в статиці або в динаміці, ця інформація надсилається на пристрій інтерпретації для класифікації образів, зображення розбивається на фрагменти, які порівнюються із заздалегідь зібраною бібліотекою зображень, та визначають, чи є збіг за відомими ознаками. Ці ознаки можуть бути, як загальним шаблоном, наприклад, контур руки з п’ятьма пальцями, так і конкретними деталями, як-от положення губ, що формують певну емоцію.

Однак КЗ має велику перевагу над людським зором, а саме – швидкість обробки. Машина із заздалегідь визначеним алгоритмом, та навченою на великій вибірці даних, може за лічені секунди ідентифікувати необхідну інформацію, раз за разом, з мінімальною похибкою. Це досягається завдяки алгоритмам, що називаються – нейронні мережі.

Нейронні мережі – це програми машинного навчання, або їх ще називають моделями, які приймають рішення подібно нейронам у людському мозку [2]. Кожна нейронна мережа складається з багатьох шарів та вузлів, у кожного такого вузла є своя вага та порогове значення, якщо певний наданий мережі елемент перевищує це порогове значення, то такий вузол активується і надсилає дані на наступний шар мережі. Водночас чим більше буде вага вузла за ідентифікації цього елементу, тим більша ймовірність його активації. Ці мережі зазвичай проходять попереднє тренування та оптимізацію, через коригування ваг та порогових значень, і різних тренувальних даних, після чого надаються у використання.

Повністю налаштовані мережі стають потужним інструментом для класифікації та кластеризації даних з високою швидкістю. Окрім цього, глибоке навчання дає змогу зберігати зображення, які мережа аналізує, тому під час її використання, вона перебуває в постійному розвитку.

1.2 Огляд технології розпізнавання жестів

Технологія розпізнавання жестів – це галузь комп’ютерного зору, яка надає безконтактний спосіб взаємодії людини з комп’ютером, який зі свого боку дає змогу обчислювальним машинам інтерпретувати жести, як команди. Основною метою такої взаємодії є покращення загального досвіду користування комп’ютером, щоб зробити його більш сприятливим до потреб користувача. З допомогою цієї технології, користувач може управляти курсором, у такий спосіб, щоби переміщувати його на екрані монітора, використовуючи пальці руки, а далі, наприклад, виконавши певний жест почати інтерпретувати рухи, як літери, для того, щоб ввести текст. Сукупність цих взаємодій, потенційно, може замінити використання приєднаних до комп’ютера пристроїв – миші та клавіатури.

Не дивлячись на те, що здебільшого підходи до розпізнавання жестів базуються на використанні комп’ютерного зору, оскільки це є найбільш

природнім варіантом, є також і підходи без використання камери, наприклад, спеціальні рукавички з розтяжними датчиками, які подають сигнали відповідно до положення пальців. Код до цієї рукавички працює у двійковій системі числення. Електричний опір сенсорів змінюється під час згинання або розгинання пальців, у такий спосіб кожен палець надсилає 0, або 1. Сукупність сигналів з усіх пальців утворює певний символ [3].

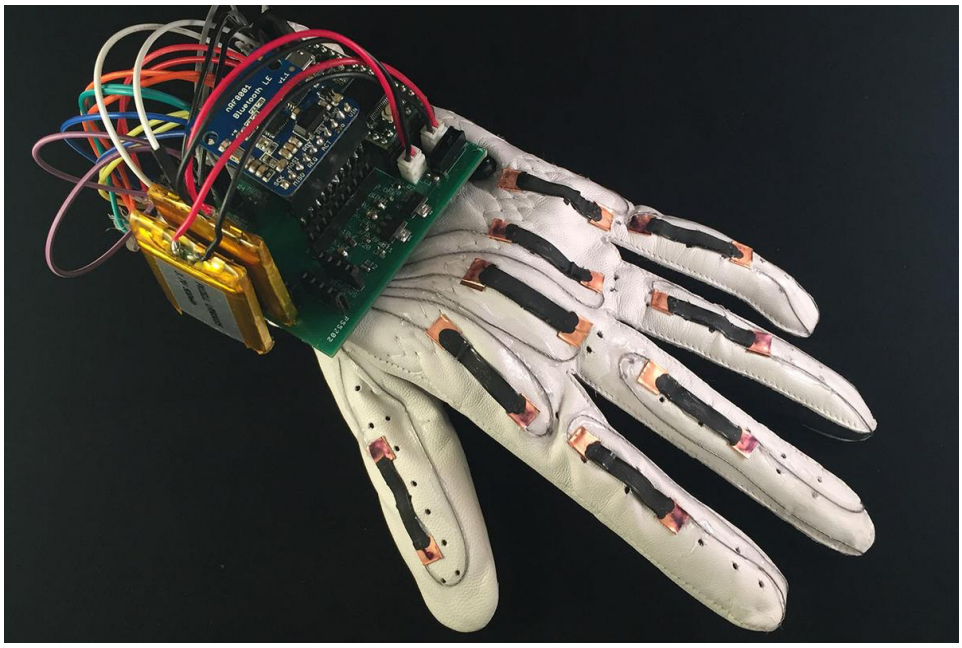

Рисунок 1. – Розумна рукавичка для перекладання мови жестів на текст

Для кращого розуміння цієї технології необхідно визначити поняття «жесту». Жестом називають рух частини тіла, частіше всього це відбувається з допомогою рук, метою якого, є вираження певної ідеї або значення[4]. У повсякденному житті люди використовують жести, як засіб невербальної комунікації, для того, щоб співрозмовник краще сприймав і обробляв вхідну інформацію. Окрім цього, жести слугують невід’ємним інструментом для взаємодії людей із порушеннями слуху або мовлення.

Процес розпізнавання жестів рук є складним, та включає кілька етапів. Першочергово користувач має «спілкуватись» з обчислювальною машиною, це відбувається через інтерфейси. Ця технологія є

альтернативним інтерфейсом з абстракцією більш високого рівня, ніж попередньо згадані миша та клавіатура. З боку комп'ютера взаємодія відбувається з використанням камери, яка передає дані із зображенням до певної програми для подальшої обробки. Використовуючи алгоритми машинного та глибинного навчання, програмне забезпечення виконує процес «feature extraction» (від англ. виокремлення особливостей), ідентифікуючи значущі зв'язки між особливостями, що формують жести. Якщо ці зв'язки відповідають заздалегідь визначеним жестам, виконується відповідна команда.

Підхід до розпізнавання жестів рук включає в себе декілька етапів, таких як: збір даних, обробка зображень, сегментація рук, вилучення ознак і класифікація жестів. Важливо відмітити, що жести бувають двох типів: статичні та динамічні. Статичний жест стосується фіксованої форми руки, у той час, як динамічний, навпаки, складається з послідовності її рухів, наприклад – махання [5]. Різниця в типах жестів суттєво впливає на хід розроблення систем розпізнавання рухів.

Методи для збору даних відрізняються залежно від підходу, чи то на основі комп'ютерного зору, чи на основі натільних сенсорів.

1.3 Пристрої розпізнавання жестів у мультимедійних програмах

1.3.1 Xbox Kinect

Xbox Kinect – пристрій для визначення рухів для ігрової платформи Xbox від компанії Microsoft. Дає змогу користувачам керувати системою з допомогою жестів, голосових команд та SDK Windows Kinect [6].

Рисунок 2. – Xbox Kinect

Має можливість розпізнавати статичні та динамічні безперервні рухи. Бібліотека ігор налічує велику кількість проектів, які використовують технологію розпізнавання жестів. Окрім вищерозглянутого розпізнавання жестів утворених руками, Kinect дає змогу відслідковувати рухи всього тіла в реальному часі. Така взаємодія зазвичай використовується в іграх, які відображають спортивні або танцювальні симулятори. Одна з найпопулярніших ігор із такою технологією – Just Dance. Це музикально-ритмічна гра, сенс якої – відтворювання рухів, що демонструються на екрані для отримання балів, чим більш точно був зроблений рух, тим більше очок заробляє гравець. Технологія підтримує можливість одночасного розпізнавання жестів до 4х гравців.

Окрім цього Kinect можна під’єднати до персонального комп’ютера й запрограмувати відповідно до своїх потреб.

1.3.2 Apple Vision Pro

Apple Vision Pro – це просторовий комп’ютер, який реалізує поєднання фізичного простору навколо користувача та цифрового контенту. Його функції дають змогу здійснювати навігацію, використовуючи, руки, голос та очі [7].

Рисунок 3. – Жестова взаємодія з Apple Vision Pro

Поява цього продукту на ринку є революційною, оскільки вона змінює парадигму навігації, надаючи, віртуальне середовище без фізичних девайсів інтерфейсу. Координація жестів та рухів очей уможливлює переміщення, масштабування та обирання віртуальних об’єктів. Роль миші відіграє напрямок погляду, даючи змогу користувачам плавно переміщуватися за інтерфейсом та обирати потрібні елементи. Після того, як курсор наведено на необхідну програму, можна виконати перелік дій: з’єднання вказівного та великого пальця для того, щоб обрати програму, подвійне клацання цими ж пальцями для її відкриття, для прокручування сторінки необхідно з’єднати пальці й затримати їх, після чого виконати рух в обраному напрямку, якщо виконувати попередню дію двома руками, то можна здійснити приближення, або обертання. Для відстеження рухів рук використовуються 8 камер із різних боків девайсу, та інфрачервоні датчики для слідкування за очима [8].

1.3.3 BMW Gesture control

Автомобільна індустрія не є винятком, коли справа стосується мультимедійних систем. Будь-який сучасний автомобіль пропонує тією чи іншою мірою сенсорний екран для контролю клімату, гучності музики чи налаштувань підвіски. Однак використання цього екрану за кермом не є цілком безпечним, оскільки для того щоби правильно виконати ту чи іншу дію, необхідно відводити погляд від дороги до екрану. Компанія BMW знайшла рішення цієї проблеми, інтегрувавши, розпізнавання жестів у систему iDrive для того, щоб водії були більш зосередженими за кермом.

Відслідковування жестів відбувається через 3D-камеру. Виконання простих рухів, може надати можливість: прийняти чи відхилити телефонний дзвінок, збільшити чи зменшити гучність звуку, встановити пункт призначення в програмі навігації, тощо[9].

Рисунок 4. – Використання системи BMW iDrive

1.3.4 Leap Motion Controller

Контролер компанії UltraLeap – це оптичний модуль відстеження рухів рук, який підключається до персонального комп'ютера або макбука. Під'єднаний модуль дає змогу грати в ігри, контролювати медіаконтент, та навіть використовувати для створення проектів.

Контролер використовує інфрачервоний сканер і датчик відстеження рухів. Інформація отримана в такий спосіб використовується для відтворення цифрової версії руки в реальному часі, для взаємодії з віртуальними об'єктами.

Рисунок 5. – Створення цифрової версії рук через Leap Motion Controller

Програмні інженери мають можливість розробляти власні додатки під контролер, що може збільшити його можливості для використання в багатьох сценаріях. Окрім цього, через властивості маніпулювання 3D-об'єктами, студенти мають можливість краще вивчати предмети, у яких є потреба в моделюванні. Таке програмне забезпечення може бути

використано, наприклад, студентами-хірургами для наочного вивчення анатомії [10].

1.4 Обмеження технології розпізнавання жестів.

Технологія розпізнавання жестів стрімко зростає в популярності останніми роками [11], оскільки в компаніях, які розробляють програмне забезпечення для споживання медіаконтенту є запит на покращення користувацького досвіду. Це зумовлено тим, що користувачі схильні витрачати більше часу в місцях, де відчувають контроль та зацікавленість. Однак ця технологія має перелік обмежень, через які, не всі компанії готові інтегрувати її у свої продукти.

Основною проблемою - є обробка рухів, які не є в переліку жестів для виконання дій. Перетин таких рухів, може спричиняти небажану поведінку програмного забезпечення, та активацію помилкових запитів. Окрім цього, якщо одні й ті самі положення тіла мають присвоєний статичний та безперервний жест, то можуть бути складнощі в тому, щоб відрізнити одне від іншого. Також важливу роль відіграє послідовність жестів, як у звичайній мові слова в різному порядку можуть утворювати різний сенс, так і в мові жестів, попередній та наступний жест можуть впливати на поточний, і накладатись одне на одного, таке явище називається коартикуляцією [12]. Цей ефект, може бути спричинений різними чинниками, такими як, форма руки кожної поодинокої людини, та індивідуальні особливості під час формування жесту. Ці аспекти можуть суттєво відрізнятися і мати різний рівень впливу на очікуваний результат.

Також варто зазначити, що системи, які розроблені з підтриманням відслідковування обох рук можуть мати проблеми з визначенням того, у який момент потрібно орієнтуватися на жест представлений однією рукою чи двома. Під час розроблення системи варто враховувати потенційні

випадки використання в середовищах з обмеженим освітленням та низькою продуктивністю компонентів комп’ютера.

Сукупність зазначених проблем, може знизити рівень досвіду користування інтерфейсом розпізнавання жестів, що змусить користувача повернутися до звичайних способів взаємодії, оскільки вони будуть більш зрозумілими та матимуть вищу точність виконання.

2 Теоретичні підходи до виявлення позиції руки

2.1 Розвиток методів ідентифікації

Поява алгоритмів глибинного навчання мала великий вплив на спосіб розпізнавання об’єктів. До їхнього виявлення, розробники покладалися на більш складні з боку обчислень методи, які включали евристики. Мета цих підходів полягала в тому, щоб за прийнятний проміжок часу знайти робоче рішення проблеми. Не дивлячись на те, що діапазон підходящих рішень, отриманий у такий спосіб знаходився відносно швидко, його точність була далеко не ідеальною [13]. Щоб збільшити цю точність, виконувався перелік оптимізацій, які збільшували вартість та складність розроблення.

Окрім цього, раніше дослідники використовували датчики глибини для виокремлення частин тіла. На сьогодні, перевага надається більш розповсюдженим та дешевим у виробництві – RGB-камерам.

2.2 Convolutional neural networks for image classification

Першим кроком для виявлення руки є використання класифікатора. Де на вхід отримується зображення для обробки, а на виході отримаємо ймовірність присутності шуканого об’єкта у вигляді вектору ознак. На роль класифікатора, може бути обрана будь-яка нейронна мережа, яка пройшла необхідне навчання на підходящому наборі зображень.

Такі мережі називають згортковими (Convolutional neural networks), оскільки принцип їхньої роботи полягає в тому, щоб зменшити (згорнути) зображення, залишаючи основні деталі. Ця операція необхідна для прискорення процесу визначення присутності об’єкта у вхідних даних, та передачі результатів на наступний етап. Основною перевагою згорткових нейронних мереж над звичайними є швидкість їхньої роботи.

CNN мають три основні типи шарів:

- Convolutional (згорктовий)
- Pooling (об’єднуючий)
- Fully-connected (повністю зв’язуючий)

Ці шари йдуть від згорткового до повністю зв’язуючого, та можуть утворювати сукупність однакових шарів згорткового або об’єднуючого типу, збільшуючи складність та точність CNN з кожним наступним шаром [14]. Процес починається з ідентифікації простих елементів, таких як, кольори або крайні пікселі об’єктів на межі із фоном зображення. Поступово просуваючись, мережа переходить до визначення більш складних деталей, таких як форма, що в підсумку призводить до виявлення шуканого об’єкта.

Основна кількість обчислень відбувається в першому згортковому шарі (Convolutional layer). CL складається з вхідних даних, фільтру, та співставлення ознак (feature mapping). Якщо взяти звичайне RGB зображення, то воно складається з трьох величин: ширини, висоти та глибини. CL обробляє зображення, використовуючи, фільтр, зазвичай це матриця 3x3, у якому задані шукані ознаки, якщо такі поля зображення мають цю ознаку, то відповідна частина перейде на наступний етап.

Сам процес є точковим добуток між пікселями зображення та фільтру, де у вихідному масиві отримується результат усіх операцій добутку.

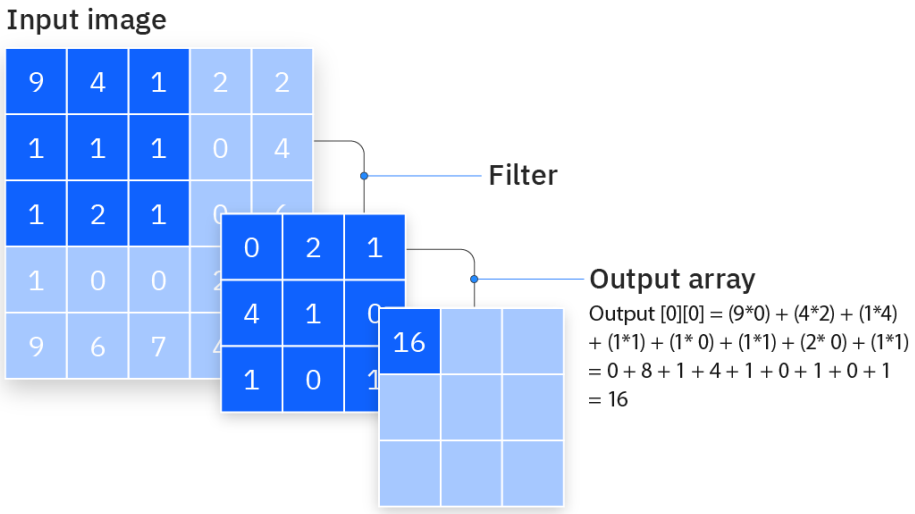

Рисунок 6. – Застосування фільтру ознак до вхідного зображення

Процес об’єднання шарів виконує операцію редукції, зменшуючи розмірність даних, що передаються до повністю зв’язуючого шару. Схожим чином, як із попереднім шаром, відбувається накладання фільтру, але в цей раз без коефіцієнтів. Натомість ядро мережі виконує операцію агрегації в межах свого фільтру. Агрегація може бути двох типів: об’єднанням за максимальним значенням, і за середнім. У першому варіанті обирається значення фільтру із найбільшим параметром, у другому випадку відповідно беруть середнє значення пікселів у межах фільтру. Не дивлячись на те, що під час цього процесу втрачається вагома частина інформації, він дає змогу збільшити продуктивність моделі та уникнути такого явища як overfitting, коли модель занадто пристосована до конкретного сценарію, і не може правильно розрізняти зображення з мінімальними відхиленням від тренувальних.

Останній повністю зв’язуючий (Fully connected) шар виконує очевидну з його назви дію, поєднує отримані результати (пікселі) з вихідних масивів попередніх шарів. Цей процес є необхідним, оскільки у зв’язуючому та об’єднувальному шарах, за замовченням немає функції, яка б поєднувала результати.

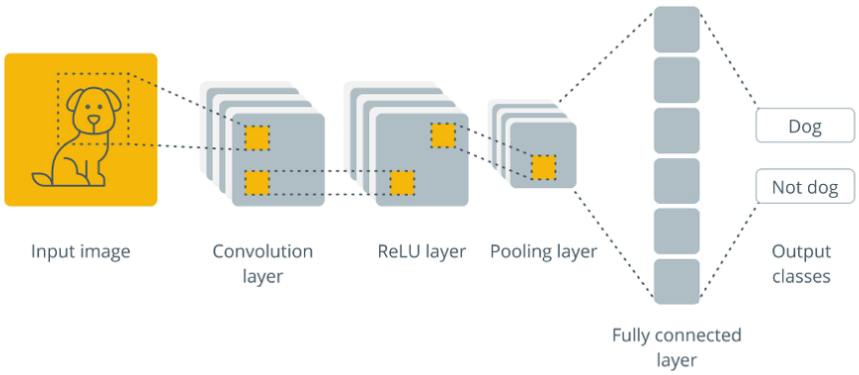

Рисунок 7. – Схема роботи згорткової нейронної мережі

2.3 Hand detection

Наступним етапом, після того, як нейронна мережа опрацювала та класифікувала зображення за ознакою присутності, є відповідне визначення, у якому конкретно місці на зображенні об’єкт розташований. Цей етап є важливим, оскільки подальша взаємодія з вхідними даними буде неможлива, маючи лише інформацію чи є представлена рука у відеопотоці.

Відео потік поділений на багато зображень, може містити багато об’єктів одночасно. Першочерговим та найпримітивнішим способом для локалізації руки, є створення прямокутної рамки навколо неї. Цю функцію виконують нейронні мережі, які мають готові рішення, як для попереднього етапу класифікації, так і для локалізації.

R-CNN (Region-based Convolutional neural network) – це нейронна мережа, де процес пошуку об’єкту виконується у два етапи: перший – розподіл зображення на регіони, другий – знаходження всіх об’єктів у кожному регіоні зображення, через класифікацію та процес згортання [15]. Підхід R-CNN пропонує поєднання можливостей звичайних CNN із додатковим розподілом певних частин зображення на регіони, їх називаються «кандидатськими регіонами» [16]. У кожному такому регіоні відбувається поодинокий пошук за значущими ознаками, з допомогою сторонніх інструментів, які можуть відрізнятися в різних реалізаціях. Після

генерації регіонів, отримані результати передають на вхід звичайному CNN для feature extraction. Зі свого боку результат CNN є вектором ознак, що представляє зміст за кожним із запропонованих регіонів. Додатковий функціонал R-CNN включає регресію за обмеженою областю, що й буде утворювати рамку навколо шуканого об’єкта.

Ключовими перевагами використання R-CNN є точність виявлення об’єктів, незалежно від їхньої кількості чи розміру, та гнучкість самого пошуку до індивідуальних потреб розробника. Однак, побічним ефектом цих переваг можна назвати обчислювальну складність підходу, як результат – повільність виконання. Це зумовлено тим, що кожен регіон обробляється окремо та послідовно, і час виведення результату цих обчислень є відносно повільним. Використання алгоритму є доцільним, якщо буде відбуватись обробка статичних зображень, однак за обробки відеопотоку затримка буде неприйнятною.

Автор оригінальної R-CNN побудував швидший алгоритм, звернувшись до недоліків попереднього рішення. Підхід здобув назву – Fast R-CNN, головною відмінністю якого є початкове згортання зображення, лише один раз, після якого вже відбувається розподіл на регіони. Оскільки вхідне зображення для регіонів у такому випадку має дуже малий розмір, то вилучення ключових ознак не займає багато часу.

Faster R-CNN є наступною ітерацією вдосконалення алгоритму, яке включає використання додаткової нейронної мережі – Region Proposal Network (RPN). Оскільки обидві попередні реалізації покладалися на вибірковий пошук, який додавав невиправдану трудомісткість та повільність, виникла потреба в більш елегантному рішенні. RPN є натренованою нейронною мережею, що допомагає краще пропонувати регіони і відповідно зменшувати кількість необхідних операції.

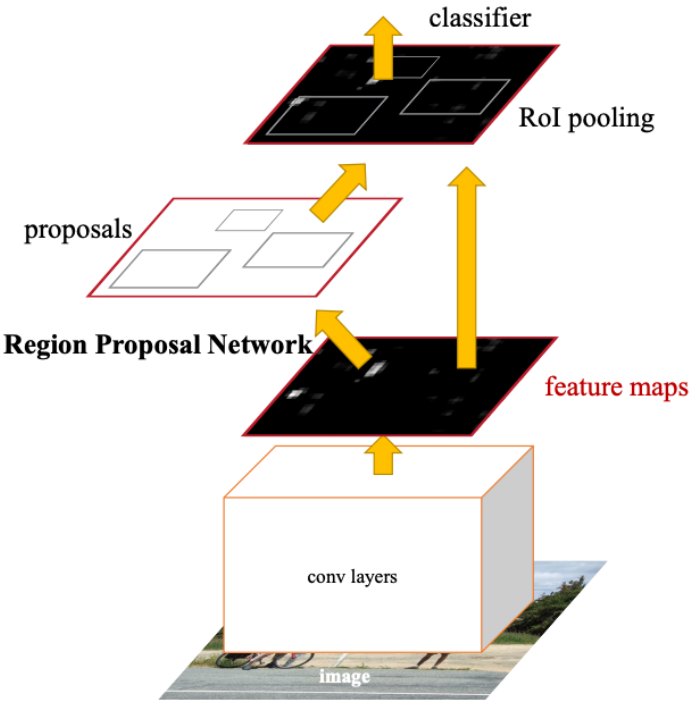

Рисунок 8. – Схема роботи моделі Faster R-CNN

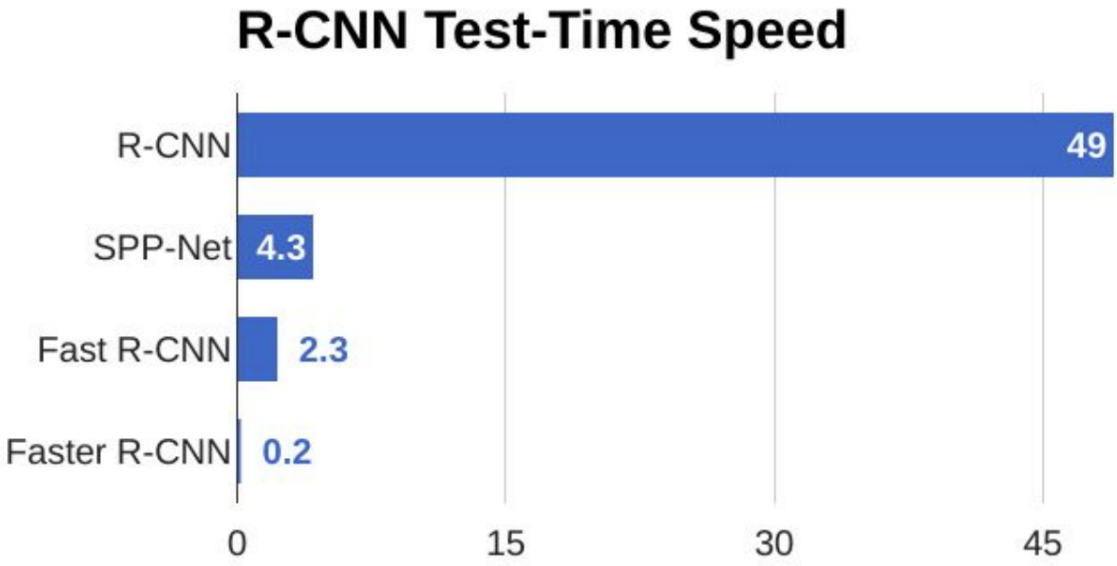

Рисунок 9. – Порівняння швидкодії варіацій R-CNN
(у секундах)

YOLO (You Only Look Once) – це алгоритм виявлення об’єктів, який суттєво відрізняється від попередніх, оскільки в ньому бракує розподілу

зображення за регіонами, іншими словами, вони не мають широкий погляд на картину, а лише можуть бачити поодинокі частинки. Алгоритм YOLO має єдину згорткову мережу, яка прогнозує обмежувальні рамки навколо об’єкта. Він працює на основі сітки (grid) розмірністю SxS [17], де кожній клітині призначається ймовірність належності до визначених класів. Цей алгоритм вирізняється більшою швидкодією, ніж попередні, тому він більш підходить до обробки даних у відеопотоці. До недоліків використання цієї мережі, можна віднести низький рівень виявлення дрібних об’єктів на зображенні та потреба у великому наборі тренувальних даних із різноманітними класами.

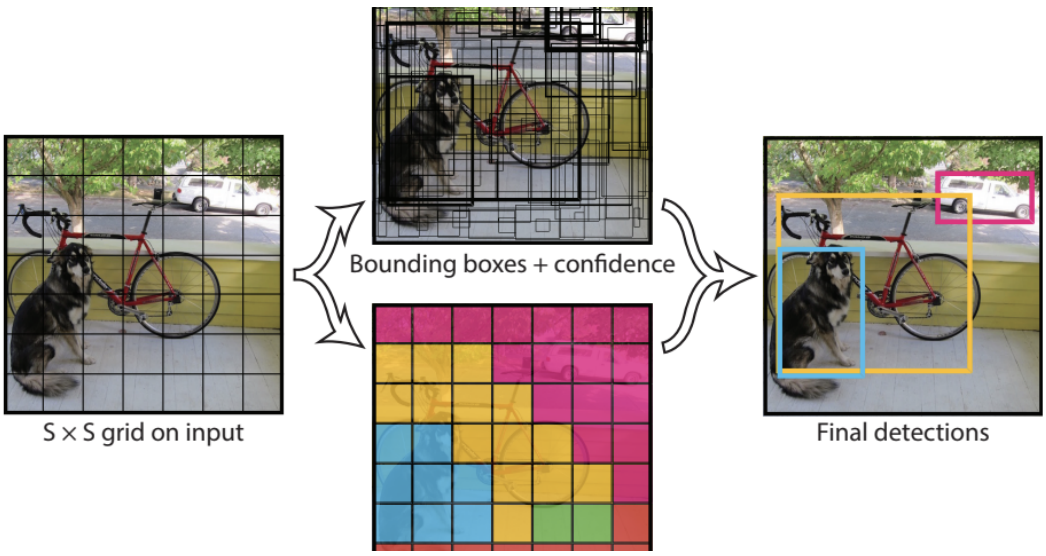

Рисунок 10. – Візуалізація роботи моделі YOLO

SSD (Single-Shot object Detection) – алгоритм який дослівно виконує виявлення об’єктів за один постріл (один прохід), без необхідності повторного виконання або повернення до попередніх станів шарів мережі. За цей прохід одночасно виконується класифікація та ідентифікація. Основну частину процесу виконує модель CNN VGG-16, яка працює з додатковими згортковими шарами для зменшення розмірності зображень.

Мережа разом із процесом згортання будує групу обмежувальних рамок навколо карти з відфільтрованих ознак (елементів об’єктів).

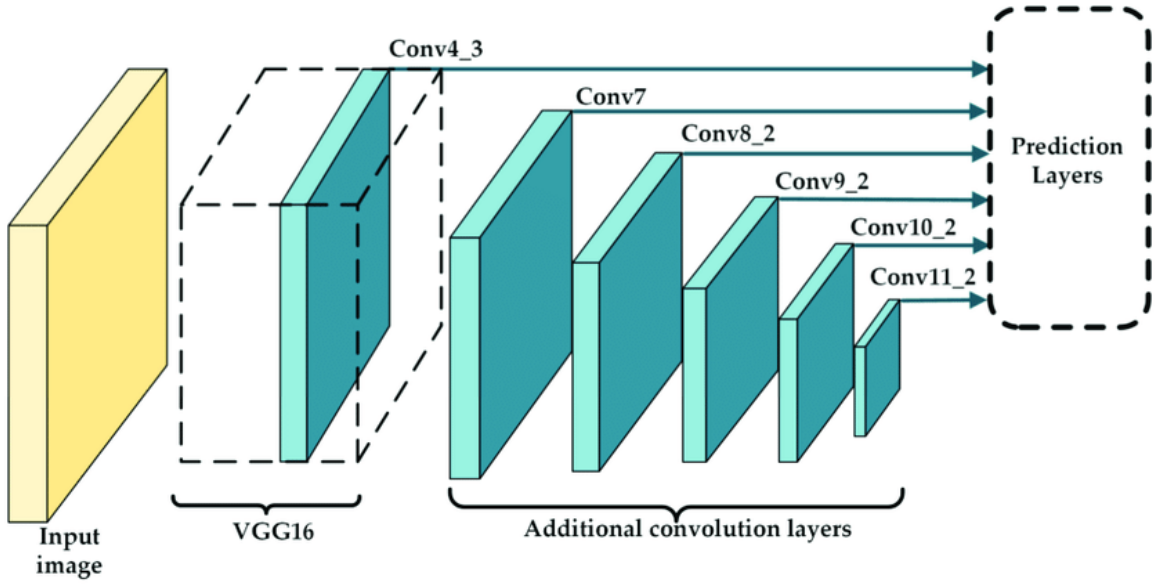

Рисунок 11. – Схема роботи моделі SSD

Незважаючи на те, що SSD та YOLO мають схожі алгоритми, їх використовують для різних сценаріїв. SSD на відміну від YOLO має декілька згорткових шарів різної розмірності, що дає змогу краще визначати, як малі, так і великі об’єкти. Такий підхід є більш точним, але водночас повільнішим за YOLO, що робить його менш привабливим для використання в застосунках із детекцію в реальному часі, там де точністю можна знехтувати на користь швидкодії.

Підбиваючи підсумки цього підрозділу, варто зазначити, що у всіх представлених алгоритмів є свої переваги та недоліки, і вибір конкретної моделі, залежить від завдань поставлених перед розробленням системи для знаходження об’єктів.

2.4 Gesture detection

Виявлення жестів є слідуючим етапом після класифікації та виявлення положення руки на зображенні. Для визначення жесту, лише наявність руки не є достатньою умовою, оскільки не зрозуміло, що саме вона робить, окрім цього, комп’ютер на поточному етапі ще не знає чи може рука щось робити. Одним із підходів для вирішення цієї проблеми є наскрізне навчання моделі на багатотисячній бібліотеці із зображеннями рук. Якщо взяти за приклад примітивну гру «Камінь, ножиці, папір», то людина зможе за долі секунди визначити, в якому стані гри перебуває рука. Для того, щоб комп’ютер міг із тією ж швидкістю визначати ці стани, потрібно щоб в модель було завантажено всі необхідні жести з різними варіаціями виконання, з відповідними позначками для комп’ютера, щоб відбувся процес «навчання».

Таке навчання не є складним процесом, оскільки нейронна мережа бере на себе всю логіку та навчається самостійно, без помітного навантаження на систему, яка вже навчена класифікувати та ідентифікувати об’єкти. Проблеми з’являються тоді, коли виникає потреба в модифікації системи, у тому, щоб додати нові жести, або прибрати старі, оскільки в такому випадку доведеться повторювати навчальний процес та збирати нові набори зображень із необхідними жестами.

Keypoints (landmarks) detection вирішує вищезгадану проблему через присвоєння орієнтирів ключовим точкам на руці. Модель tensorflow використана в цій роботі використовує шаблон із двадцяти однієї точки, із чотирма точками для кожного із пальців та однією точкою для зап’ястя.

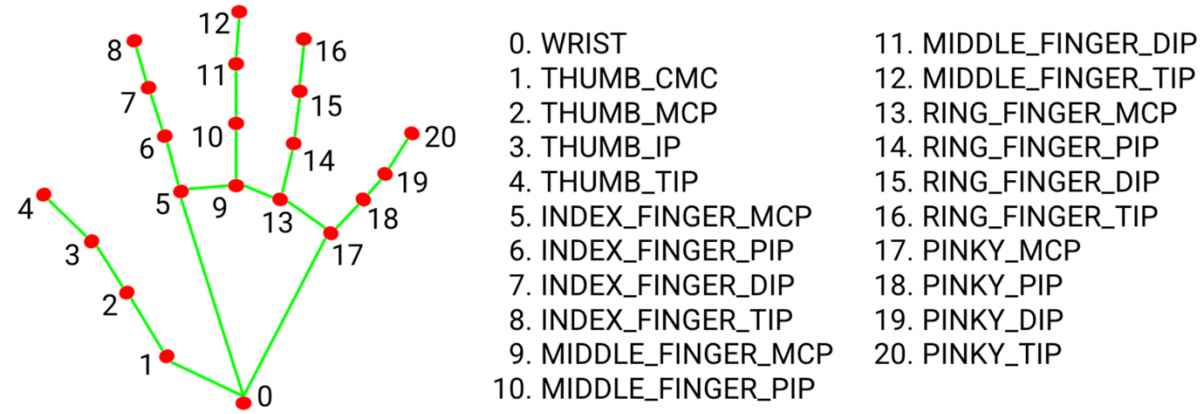

Рисунок 12. – Схема руки MediaPipe з ключовими точками

Цей підхід дає можливість розглядати руку, не як одне ціле, а як сукупність пов’язаних елементів. Інформація яку модель дістає про кожну точку дає змогу гнучкіше проектувати бібліотеку жестів, без необхідності в повторному тренуванні.

Останнім на сьогодні етапом розвитку технології розпізнавання жестів є додавання параметру глибини для визначення руки, тобто 3D keypoints extraction. Маючи, лише параметри ширини та висоти, можна проектувати методи для розпізнавання жестів, лише за умови, що виконуваний жест представлений перед камерою без зміни куту обертання. Цей аспект, може спричиняти труднощі у користувачів, якщо наприклад, в них є фізичні обмеження, для того щоби повернути кінцівку, але жест виконаний правильно.

Результат, який досягається з використанням 3D мереж є більш точним, але водночас і більш затратним за часом та ресурсами, оскільки початково однаково обчислюється положення руки в двовимірному просторі, після чого обчислюється глибина між визначеними ключовими точками.

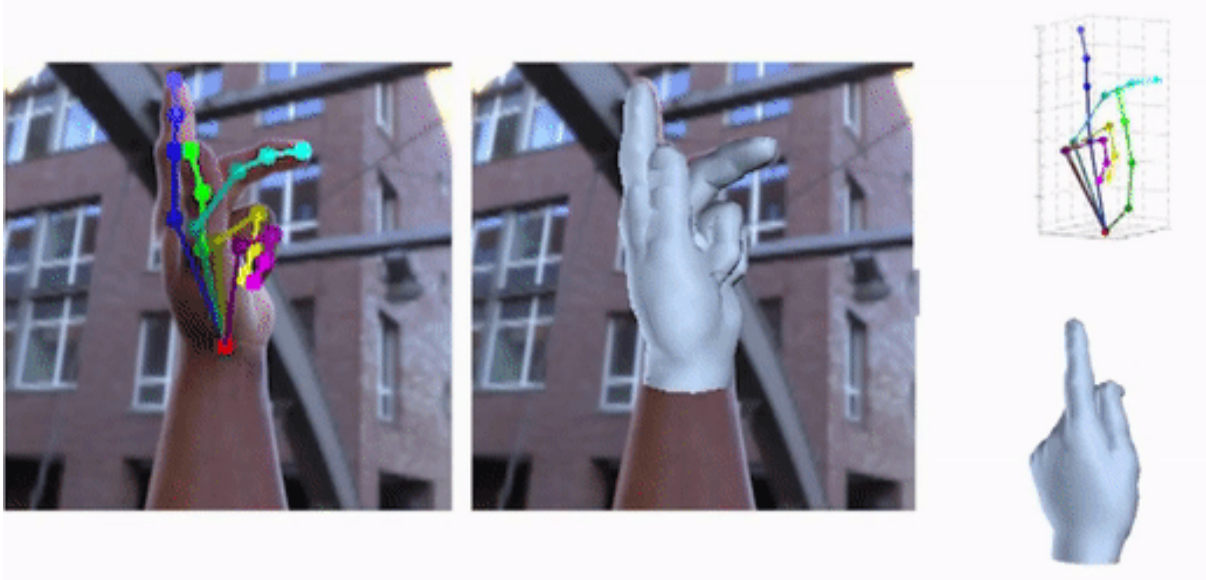

Рисунок 13. – Тривимірне розпізнавання ключових точок

З огляду на представлені вище алгоритми, може скластися враження, що технологія розпізнавання жестів досягнула своєї межі розвитку. Однак попереду ще багато роботи для того, щоб зробити цей інтерфейс максимально інтуїтивним для користувача, оскільки потреба у виявленні жестів не є постійною за використання цифрового продукту, саме тому розроблена система має бути налаштована в такий спосіб, щоб розуміти, у який момент їй варто розпізнавати жести, а в який - ні. Окрім цього кожній такій моделі треба вчитися на попередньому досвіді, для того, щоб ставати більш точною завдяки перебуванню в постійному розвитку.

3 Інструменти для розроблення Chrome додатку з жестовою взаємодією

3.1 Google Chrome extension

Для реалізації методів використання технології розпізнавання жестів обрано браузерне середовище, щоб охопити більшу кількість користувачів. Вибір зумовлений швидким процесом розроблення, через готові рішення мультимедійних систем, без необхідності інсталяції поодиноких програм.

Статистика за 2023 рік вказує на те, що другою медіаплатформою після Facebook за кількістю активних користувачів є YouTube, два з половиною мільярди щомісячних користувачів [18]. Оскільки відкритого доступу до контрольної панелі медіаплатформи немає, взаємодію з переглядом відео реалізовано через симуляцію дій користувача у вебсередовищі. Для додавання власного програмного забезпечення в браузер використано технологію браузерного розширення.

Google Chrome розширення (extension) – це програма, яку можна інсталювати в будь-який браузер, що працює на Chromium ядрі для збільшення функціональності браузера. Chromium є проектом із відкритим вихідним кодом, мета розроблення якого, створити більш швидке, стабільне та безпечне середовище для роботи в інтернеті [19].

Розширення для браузера, як і будь-який інший вебпродукт, розробляється з використанням HTML, CSS та JavaScript. Однак на відміну від звичайного вебзастосунку, розширення для успішного завантаження в браузер потребує наявності файлу manifest.json. Це файл, який містить метадані про всі конфігурації, які можуть бути налаштовані в браузері, як-от заголовок застосунку, його версія, та поодинокі специфічні аспекти: використання віддалених скриптів, фонові операції чи доступні вебресурси. Окрім цих параметрів, можна додатково задати default_icon для картинки, що буде відображатись у переліку розширень, та default_popup, для

відкривання HTML коду за клацанням на іконку додатку. У реалізації розширення з розпізнаванням жестів, цей параметр використано, як кнопку переходу на сторінку з інструкцією користування. Локальне додавання створеного розширення відбувається через URL «chrome://extensions» в режимі для розробників. Для того, щоб інші користувачі мали змогу скачати розроблене розширення, необхідно завантажити архів програми на Chrome Web Store разом із заповненою анкетою застосунку. В анкеті вказується мета розроблення цього застосунку, його можливості та функціонал, чи збирає він дані користувачів, та які зовнішні скрипти використовує. Спочатку анкета із заявкою перевіряється штучним інтелектом, після чого має пройти перевірку працівником Google. Цей крок необхідний для того, щоб упевнитись, що розширення не містить шкідливого програмного забезпечення, або може нанести потенційної шкоди користувачу. Швидкість прийняття рішення щодо публікації розширення залежить від кількості додаткових параметрів та зовнішніх скриптів.

3.2 Підбір моделі для реалізації функціоналу розширення

Процес проектування та створення методів для розпізнавального інтерфейсу розширення передбачає використання вищезгаданих нейронних мереж. Оскільки розроблення відбувалось із використанням мови програмування JavaScript, то і відповідна модель має бути побудована на цій мові.

Платформа Tensorflow є бібліотекою у відкритому доступі з програмами для використання штучного інтелекту та машинного навчання у своїх проектах. Конкретні рішення з використанням розпізнавання жестів рук є для різних мов програмування: Python, C++ та JavaScript. Модель обрана для розроблення створена на основі структури MediaPipe.

MediaPipe – це кросплатформенний фреймворк, для розроблення індивідуальних рішень із використанням машинного навчання для медіаконтенту в реальному часі [20]. Він використовується, як засіб розроблення програм, які потребують технології комп'ютерного зору під час роботи з відеоданими. Фреймворк містить у собі pipelines, що відповідають за роботу нейронної мережі, та алгоритми обробки або трансформування медіаданих.

Handpose – нейронна мережа, що належить до типу згорткових, розроблена з використанням MediaPipe, яка може розпізнавати лише одну руку в кадрі. Pipeline цієї моделі відповідає за відстеження скелету-руки, тобто двадцяти однієї 3D-точки окремо для правої чи лівої руки [21]. Модель на основі отриманих у реальному часі даних із відеокамери визначає чи присутня рука в кадрі, якщо так, то навколо неї будується обмежувальна рамка, а на руці встановлюються ключові точки, на виході отримаємо координати всіх цих елементів. Надалі ці координати можна використовувати для демонстрації роботи моделі, наприклад, встановивши для кожної точки елемент кругу, якому присвоюється колір на вибір.

Наразі версія моделі є експериментальною, тому за її використання можуть виникати неточності через визначення положення руки, їх розглянуто більш детально в реалізації браузерного додатку для YouTube. Поточна ефективність моделі, потенційно, може відрізнятися залежно від людських характеристик – статі, віку чи відтінку шкіри, це зумовлено тим, що тренування моделі проводилося на обмеженому наборі даних. Окрім цього, модель є непридатною для виявлення руки в рукавичці, або на занадто далекій відстані. Через нестабільну ефективність цієї нейронної мережі, її потрібно використовувати лише для тестових чи розважальних застосунків, тобто тих, де неточність прийняття рішень не є критичною.

4 Реалізація методів для збільшення контролю користувачів

4.1 Інтеграція інтерфейсу взаємодії з користувачем

Ініціалізація розроблення розширення відбувається зі створення та налаштування файлу manifest.json. Обов'язковим етапом є вказування поточної (третьої) версії маніфесту, оскільки він містить перелік додаткових заходів для перевірки застосунку на безпечність. Попередні версії за спроби завантаження в браузер будуть відхилені. З додаткових налаштувань вказано використаний скрипт та шаблон покликання на YouTube.

Завантаження моделі в браузерному середовищі відбувається з допомогою імпортів бібліотеки Tensorflow.js та handpose. Однак браузер сприймає лише чіткі інструкції JavaScript файлів, без змоги завантажити додаткові залежності. Для того, щоб надати необхідну інформацію з бібліотек, використано компонувальник модулів Webpack, його основна функція в тому, щоб зібрати всі додані локально залежності, і трансформувати їх у єдиний скрипт-файл зрозумілий для вебсередовища.

Наступним кроком є налаштування вебкамери вбудованої в пристрій користувача. Функція createVideo створює елемент відео, що буде відображати потік медіа-інформації з вебкамери. Запуск і зупинка отримання даних із відеопотоку регулюється відповідними функціями startHandposeDetection та stopHandposeDetection з прив'язкою до елементу відео за його ініціалізації та очищенням ресурсів після зупинки.

Функція runHandpose виконує завантаження моделі handpose, виклик є асинхронним для уникнення блокування інших операцій. Процес розпізнавання руки починається одразу після завантаження моделі. Окрім завантаження, нейронна мережа має встановлений інтервал 0.1 секунду. Це означає, що поточний стан руки буде обрахований 10 разів кожної секунди. Частота обрахування позиції руки зумовлена потребою в зменшенні

затримки для виконання жестів. Далі функція handDetection перевіряє чи надходять дані з відеокамери, якщо так, то модель виконує метод estimateHands з параметром video на вході. На виході, отримаємо приблизні координати двадцяти однієї точки, на базі яких, сформовані методи для ідентифікації жестів. Для тестувальної версії застосунку було використано елемент canvas для відображення стану відеопотоку в реальному часі з кольоровим відображенням ключових точок на руці та зв’язків між ними.

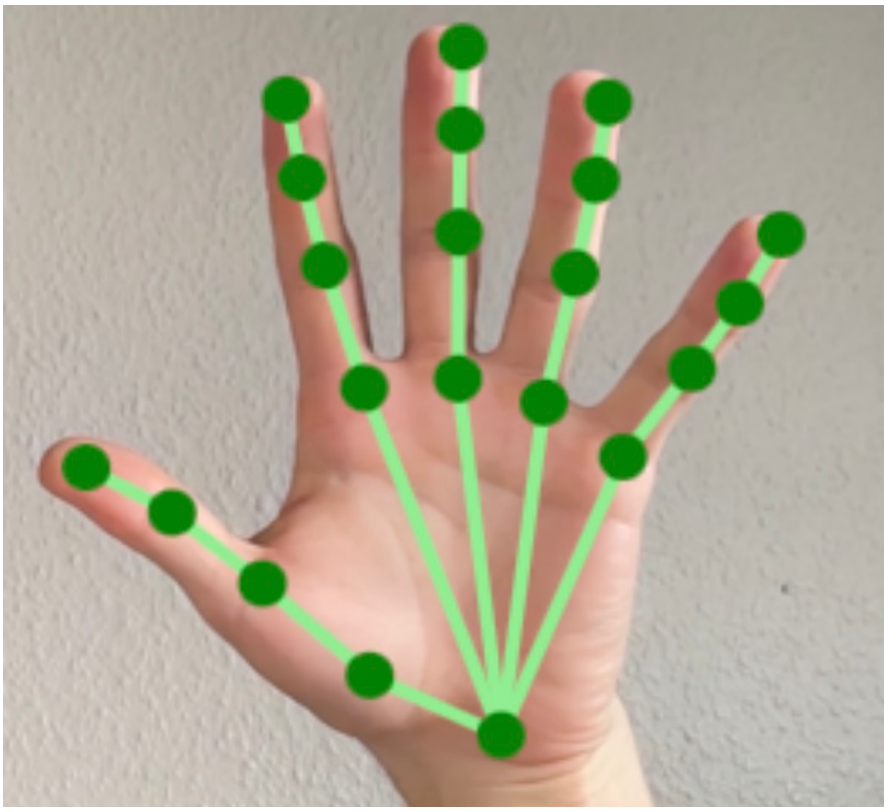

Рисунок 14. – Зафарбовані ключові точки руки

4.2 Методи ідентифікації та підрахунку пальців

Після отримання координат ключових точок для кожного пальця, необхідно встановити правила, щодо яких програма буде визначати стан кожного пальця, тобто чи розігнутий він, а також сукупність розігнутих пальців для подальшого присвоєння жестів відповідним комбінаціям. Якщо

взяти для прикладу вказівний палець, то звернувшись до схеми номерів ключових точок на руці, представленою в другому розділі, визначено: що за розташування вказівного пальця відповідають точки 5, 6, 7 та 8.

Для того, щоб визначити чи розігнутий палець, необхідно обрахувати довжину між початковою та кінцевою точкою кожного пальця, для вказівного це є точки 5 та 8, якщо отримана довжина більше за певне порогове значення, то палець вважаємо розігнутим. Порогове значення, відносно якого порівнюється довжина кожного пальця, має бути налаштованим таким чином, щоби правильно працювати для різних користувачів. Оскільки довжина руки та пальців у різних людей відрізняється, то й порогове значення буде різним, а отже, не може бути фіксованим. Після декількох спроб для пошуку оптимального порогового значення, вирішено взяти поділену навпіл довжину між точкою, яка відповідає за зап’ястя та точкою початку середнього пальця, ці точки позначені номерами 0 та 9. Аналогічно до вказівного пальця, підрахунки довжин виконані для середнього пальцю, безіменного та мізинцю.

Визначення стану розгину для великого пальцю, реалізовано іншим способом. Оскільки виконання жесту перед екраном передбачає, що чотири інших пальці спрямовані вгору, відповідно їхня довжина, та порогове значення обраховуються за віссю ординат. Великий палець на відміну від них, залежно від руки спрямовується в правий чи лівий бік, тому визначення його стану розгину реалізоване через вісь абсцис. Порогове значення для нього є різницею довжин між початковою точкою вказівного пальця та початковою точкою мізинцю.

Виконання цих обрахунків реалізовано у функціях countFingers та define_fingers, де за кожний розігнутий палець збільшується лічильник пальців, а елемент, який визначає назву пальця записується в масив. Отже, у такий спосіб, за повторному запуску програми можна вивести результати,

що будуть відображати кількість та назви розігнутих пальців кожні 100 мілісекунд.

4.3 Отримання доступу до відеоплеєру YouTube

Для того, щоб реалізація та виконання дій на основі жестів на сторінці перегляду відео була можливою, необхідно створити тригер. Роль цього тригера відіграє створена кнопка вбудована в DOM (Document object model) вебсторінки, коли користувач натискає на цю кнопку, запускається процес для визначення руки й обрахунок представлених пальців, якщо необхідно завершити процес, то користувачу необхідно клацнути на кнопку ще один раз. У файлі background.js виконується робота у фоновому режимі, розширення визначає чи є параметр URL новим або оновленим, а також чи включає поточна адреса вебсторінки основу «youtube.com/watch», якщо включає, то на основний скрипт надсилається повідомлення про успішне завантаження браузерної сторінки, яка містить YouTube відео.

Додавання кнопки відбувається після надсилання сторінкою повідомлення про успішне завантаження. Через DOM сторінки отримано доступ до елементів контрольної панелі відеоплеєра, після чого додано кнопку з обраним зображенням та присвоєно їхній click event, який передбачає запуск функції triggerHandEventHandler. Ця функція відповідає за запуск процесу розпізнавання руки, якщо процес не був до цього запущеним, в іншому випадку навпаки припиняє його.

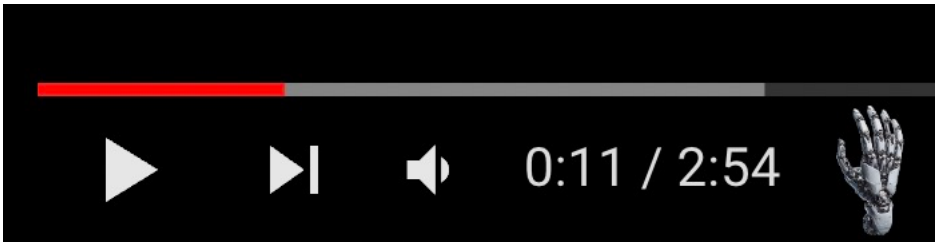

Рисунок 15. – Кнопка для запуску процесу розпізнавання жестів

4.4 Визначення жестів

Для того щоби присвоїти жести різним комбінаціям пальців необхідно визначити основний перелік дій медіаплеєру. Контрольна панель YouTube містить кнопки для активації представлених дій:

- Пуск/пауза
- Наступне відео
- Збільшити/зменшити гучність
- Ввімкнути/вимкнути звук
- Ввімкнути/вимкнути субтитри
- Налаштування
- Зменшити розмір відеоплеєра
- Режим театру
- Повноекранний режим

Щоб система контролю жестами не була перевантаженою різними діями, серед представленого вище переліку обрано найбільш важливі, вони включають: пуск/паузу та регулювання гучності з можливістю повністю вимикати звук. Додатково включено необхідну дію для перемотки відео вперед/назад, яку зазвичай виконують через кнопки з клавіатури.

Активація дій користувача може відбуватись або через симуляцію взаємодії з клавіатурою та мишкою користувача, або через зміни поточних станів змінних відеоплеєра. Оскільки виконання головного скрипту відбувається через браузерне середовище, а не локально на комп’ютері, враховано перелік обмежень. Єдиним девайсом, до якого, можна дістати доступ із браузера є камера. Для того, щоб взаємодіяти із клавіатурою або мишкою, необхідні справжні кліки, симуляція яких є неможливою. Через наявність цього обмеження, прийнято рішення взаємодіяти зі значеннями змінних в DOM сторінки.

Для тригеру активації дій пуск/пауза та ввімкнення/вимкнення звуку використано метод click на знайдених кнопках у контрольній панелі. Активація пуск/пауза виконується, якщо користувач демонструє руку зі всіма розігнутими пальцями.

Рисунок 16. – Жест що виконує дію пуск/пауза

Дія для ввімкнення/вимкнення звуку передбачає присутність середнього пальцю, безіменного та мізинцю, наявних у переліку розігнутих пальців.

Рисунок 17. – Жест що виконує дію ввімкнути/вимкнути звук

Для того, щоб уникнути ситуації з постійним виконанням поточної дії, коли жест ще не був змінений, створено перевірку для попереднього стану руки. Якщо попередня кількість обрахованих пальців збігається з поточною, дія не буде виконана повторно. Ця перевірка необхідна лише для жестів, де очікується єдиноразова дія, тобто вона не використовується для рухів пов’язаних із перемоткою відео та змінною гучності.

Активацію дії перемотки вирішено присвоїти жесту, де розігнутим є лише великий палець. Якщо користувач демонструє великий палець правої руки, то перемотка відбувається вліво, тобто в бік початку відео. Якщо представлено ліву руку, то розігнутий великий палець буде виконувати перемотку вправо, у бік кінця відео.

Рисунок 18. – Жест що виконує дію перемотки відео

Для того щоби програма могла визначати, якою є представлена рука, чи правою чи лівою, введено додаткову перевірку. Вона працює через положення ключової точки, яка є приблизно посередині руки, за номером 9. Оскільки камера, що надсилає відеопотік, має обмежений кут зору, його можна поділити навпіл умовною середньою лінією. Після визначення цієї середини, створено перевірку, щоб рука була більше або менше цього значення, у такий спосіб присвоєно статус правої або лівої руки.

Окрім цього, враховано ситуацію, коли користувач демонструє розігнутий великий палець правої руки, але рука розвернута, тобто камера бачить її зовнішню сторону, а не долоню. Для того, щоб визначити, яка сторона руки демонструється, необхідно порівняти будь-які значення точок пальців за віссю абсцис. За приклад взято ключові точки 5 та 9, якщо координати точки 9, за віссю абсцис, більше за координати точки 5, то рука повернута до камери долонею, якщо менше, то навпаки зовнішнім боком. Відповідно до цих перевірок визначено, чи перемотка має виконуватись у бік початку відео чи в його кінець.

На відміну від представлених вище жестів, попередній стан руки не враховується, тобто перемотка буде відбуватися доти, до поки жест не зміниться.

Оскільки немає змоги активувати дію перемотки з використанням кнопок клавіатури, вирішено напряму змінювати значення часу відео, щоби перемотка відбувалася в одну секунду за кожен обрахований жест перемотки. У процесі розроблення, виявлено, що значення змінюються лише в DOM вебсторінки, однак вони не впливають на стан відео, бо YouTube їх автоматично перезаписує. Для того, щоб обійти це обмеження, потрібно відштовхуватися від поточного часу в системі, на яке орієнтується відеоплеєр. Збільшення та зменшення змінної поточного часу, уможливлює активацію перемотки з розпізнаванням жестів.

Збільшення гучності присвоєно жесту із розігнутими вказівним та середнім пальцями. Зменшення відбувається, якщо демонструється лише вказівний.

Рисунок 19. – Жест, що виконує дію збільшення гучності

Рисунок 20. – Жест, що виконує дію зменшення гучності

На відміну від перемотки, значення гучності успішно перезаписується, але візуально це не можна відобразити через оригінальний повзунок у контрольній панелі, тому було створено власне відображення гучності, у відсотках. Де спочатку через завантаження моделі встановлюється поточне значення гучності, а потім регулюється в залежності від жестів користувача. Кожна ітерація збільшення чи зменшення звуку, змінює значення відображення на три десятих відсотка, таке значення встановлене для більш плавної зміни гучності.

Рисунок 21. – Відображення гучності звуку в контрольній панелі

4.5 Обмеження застосунку

Незважаючи на те, що здебільшого, розширення працює оптимально, іноді бувають помилкові ідентифікації. Наприклад, якщо користувач демонструє п’ять пальців, але програма, під час цього жесту, хоча б один раз помилково виявить іншу кількість, після чого знов буде виявляти п’ять, то відбудеться небажане активування дії. Окрім цього, якщо процес розпізнавання жестів буде запущений, але користувач не показує руку в кадрі, модель усе одно буде намагатись її знайти, і в умовах поганого освітлення, чи низької роздільної здатності камери, також може не правильно ідентифікувати представлене середовище, що призведе до випадкової активації дії. Причиною такої поведінки можна вважати той факт, що модель handpose, як було зазначено вище, вважається експериментальною, а отже, не претендує на ідеальну точність. Варто також додати, що обмежений перелік запропонованих користувачу жестів був обраний, не тільки для уникнення перевантаження можливих дій саме для користувача, а і для того, щоб модель працювала більш коректно. Оскільки через більшу кількість жестів на різних комбінаціях пальців, можливий їхній перетин, знову ж таки через згадану експериментальність моделі.

Отже, можна сказати, що наразі створений застосунок працює на достатньо точному рівні, для того, щоб користувач міг збільшити контроль над відеоплеєром, використовуючи жести для активації дій, які найбільше використовуються. Варто зазначити, що в майбутньому, коли буде створена нова модель, або вдосконалена поточна, розширення можна буде покращити, додаючи нові функції та більш складні комбінації жестів.

Висновки

У роботі проведено аналіз технології розпізнавання жестів для проектування безконтактної комунікації між людиною та комп’ютером. Розглянуто основні аспекти сучасних методів виявлення об’єктів, зокрема, їхні точність, швидкість роботи та застосування в різних сценаріях.

Наведено перелік успішних реалізацій пристроїв із використанням комп’ютерного зору, які демонструють можливості взаємодії з мультимедійними програмами.

Представлено реалізацію програми браузерного розширення для підвищення контролю користувачів над переглядом відео на платформі YouTube. У ній створено методи для підрахунку кількості представлених пальців та визначення їхнього стану розгину. Ці підходи спроектовано з допомогою бібліотеки Tensorflow, зокрема, моделі handpose для представлення руки у вигляді координатних точок. Окрім цього, реалізовано спосіб взаємодії з контрольною панеллю YouTube через DOM вебсторінки. Створено інтерфейс запуску та зупинки програми через спеціальну кнопку. Вивантажено готовий програмний продукт на платформу Chrome Web Store, для того, щоб інші користувачі мали змогу їм користуватись.

Визначено вектор розвитку застосунку для збільшення його функціональності та покращення точності та ефективності роботи.

Не дивлячись на той факт, що використана модель для розпізнавання та інтерпретації жестів іноді видає хибні припущення щодо положення руки, відчутно, що розроблена програма надає високоякісний інтерфейс, який є зручною альтернативою клавіатурі та миші.

Список літератури

1. What Is Computer Vision? Meaning, Examples, and Applications in 2022 [Електронний ресурс] // https://www.spiceworks.com/. – 2022. – Режим доступу до ресурсу: https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-computer-vision/.

2. What is a neural network? [Електронний ресурс] // https://www.ibm.com/uk-en – Режим доступу до ресурсу: https://www.ibm.com/topics/neural-networks.

3. Smart glove translates sign language gestures into text [Електронний ресурс] // https://newatlas.com/. – 2017. – Режим доступу до ресурсу: https://newatlas.com/sign-language-translate-glove/50474/.

4. gesture [Електронний ресурс] – Режим доступу до ресурсу: https://www.merriam-webster.com/dictionary/gesture.

5. An Exploration into Human–Computer Interaction: Hand Gesture Recognition Management in a Challenging Environment [Електронний ресурс] // https://link.springer.com/. – 2023. – Режим доступу до ресурсу: https://link.springer.com/article/10.1007/s42979-023-01751-y.

6. Gesture Commands with Kinect Gesture Recognition [Електронний ресурс] // medium.com. – 2018. – Режим доступу до ресурсу: https://medium.com/@sunx0578/gesture-commands-with-kinect-gesture-recognition-143a605150e6.

7. Apple Vision Pro and visionOS overview [Електронний ресурс] – Режим доступу до ресурсу: https://support.apple.com/en-gb/guide/apple-vision-pro/tan39b6bab8f/visionos.

8. Beyond Controllers: Apple’s Vision Pro Brings Hand Gestures and Eye Tracking to Virtual Worlds [Електронний ресурс] // https://www.encora.com/. – 2023. – Режим доступу до ресурсу:

https://www.encora.com/insights/beyond-controllers-apples-vision-pro-brings-hand-gestures-and-eye-tracking-to-virtual-worlds.

9. BMW gesture control - the next level of iDrive interaction [Електронний ресурс] // https://www.bimmer-tech.net/. – 2020. – Режим доступу до ресурсу: https://www.bimmer-tech.net/blog/item/124-bmw-gesture-control.

10. 7 Things about Leap Motion [Електронний ресурс] – Режим доступу до ресурсу: https://www.gvsu.edu/cms4/asset/7E70FBB5-0BBC-EF4C-A56CBB9121AECA7F/7_things_leap_motion.pdf.

11. The global gesture recognition market size is projected to grow from $24.78 billion in 2024 to $169.26 billion by 2032, at a CAGR of 27.1%... Read More at:- https://www.fortunebusinessinsights.com/industry-reports/gesture-recognition-market-100235 [Електронний ресурс] // https://www.fortunebusinessinsights.com/. – 2024. – Режим доступу до ресурсу: https://www.fortunebusinessinsights.com/industry-reports/gesture-recognition-market-100235.

12. A Review of the Hand Gesture Recognition System: Current Progress and Future Directions [Електронний ресурс] // https://ieeexplore.ieee.org/Xplore/home.jsp. – 2017. – Режим доступу до ресурсу: https://ieeexplore.ieee.org/abstract/document/9622242.

13. Introduction To The Heuristic Function In AI [Електронний ресурс] // https://www.simplilearn.com/. – 2023. – Режим доступу до ресурсу: https://www.simplilearn.com/tutorials/artificial-intelligence-tutorial/heuristic-function-in-ai#:~:text=Heuristics%20is%20a%20method%20of,rather%20than%20a%20perfect%20solution.

14. What are convolutional neural networks? [Електронний ресурс] // https://www.ibm.com/uk-en – Режим доступу до ресурсу: https://www.ibm.com/topics/convolutional-neural-networks.

15. Getting Started with R-CNN, Fast R-CNN, and Faster R-CNN [Електронний ресурс] // https://www.mathworks.com/?s_tid=gn_logo – Режим доступу до ресурсу: https://www.mathworks.com/help/vision/ug/getting-started-with-r-cnn-fast-r-cnn-and-faster-r-cnn.html.

16. What is R-CNN? [Електронний ресурс] // https://blog.roboflow.com/. – 2023. – Режим доступу до ресурсу: https://blog.roboflow.com/what-is-r-cnn/.

17. R-CNN, Fast R-CNN, Faster R-CNN, YOLO — Object Detection Algorithms [Електронний ресурс] // medium.com. – 2018. – Режим доступу до ресурсу: https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e.

18. The Top 10 Social Media Sites & Platforms [Електронний ресурс] // https://www.searchenginejournal.com/. – 2024. – Режим доступу до ресурсу: https://www.searchenginejournal.com/social-media/social-media-platforms/.

19. Chromium [Електронний ресурс] // https://www.chromium.org/chromium-projects/ – Режим доступу до ресурсу: https://www.chromium.org/Home/.

20. MediaPipe: Google’s Open Source Framework for Machine Learning solutions (2024 Guide) Read more at: https://viso.ai/computer-vision/mediapipe/ [Електронний ресурс] // https://viso.ai/. – 2024. – Режим доступу до ресурсу: https://viso.ai/computer-vision/mediapipe/#:~:text=Most%20MediaPipe%20solutions%20use%20different,Object%20Detection%20(Objectron)%20tasks.

21. MediaPipe Handpose [Електронний ресурс] // https://www.npmjs.com/. – 2024. – Режим доступу до ресурсу: https://www.npmjs.com/package/@tensorflow-models/handpose.