

Вінницький національний технічний університет
(повне найменування вищого навчального закладу)

Факультет інтелектуальних інформаційних технологій та автоматизації
(повне найменування інституту, назва факультету (відділення))


Кафедра комп'ютерних наук
(повна назва кафедри (предметної, циклової комісії))

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

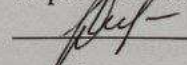
на тему:

**«Інформаційна технологія жестового керування
WEB-додатком на основі згорткових нейронних мереж»**

Виконав: студент 2-го курсу, групи 1КН-21м
спеціальності 122 «Комп'ютерні науки»
(шифр і назва напрямку підготовки, спеціальності)

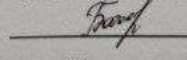
 Каліцінський В.В.
(прізвище та ініціали)

Керівник: к.т.н., доцент каф. КН


 Колесницький О.К.
(прізвище та ініціали)

« 16 » 12 2022 р.

Опонент: к.т.н., доцент каф. АІТ

 Богач І.В.
(прізвище та ініціали)

« 16 » 12 2022 р.

 Допущено до захисту

Завідувач кафедри КН

д.т.н., проф. Яровий А.А.
(прізвище та ініціали)

« 16 » 12 2022 р.

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та
автоматизації Кафедра комп'ютерних наук
Рівень вищої освіти II-й (магістерський)
Галузь знань – 12 «Інформаційні технології»
Спеціальність – 122 «Комп'ютерні науки»
Освітньо-професійна програма – «Системи штучного інтелекту»

ЗАТВЕРДЖУЮ
Директор ТОВ «ДСОФТВЕР»
Бондар Н.П.



ЗАТВЕРДЖУЮ
Завідувач кафедри КН
д.т.н., проф. Яровий А.А.

19. вересня 2022 року

ЗАВДАННЯ

НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Каліцінський Владислав Вікторович

(прізвище, ім'я, по батькові)

1. Тема роботи Інформаційна технологія жестового керування WEB-додатком на основі згорткових нейронних мереж

керівник роботи к.т.н., доцент кафедри КН Колесницький О.К.

затверджені наказом вищого навчального закладу від "14" 09 2022 року № 203

2. Строк подання студентом роботи 18 листопада 2022 року

3. Вихідні дані до роботи:

Браузерне середовище роботи додатку із підтримкою веб-камери, кількість жестів, що розпізнаються не менше 9, достовірність розпізнавання — не менше 85%, обсяг навчальної вибірки не менше 400, обсяг тестової вибірки не менше 400, легка додатку за посиланням, інтерфейс користувача має бути інтуїтивно зрозумілим, інтеграція програми із зовнішніми додатками не менше 2.

4. Зміст текстової частини:

Вступ, аналіз предметної області жестового керування WEB-додатком, проектування інформаційної технології жестового керування WEB-додатком, програмна реалізація інформаційної технології жестового керування WEB-додатком на основі згорткових нейронних мереж, аналіз результатів тестування, розробка інструкції користувача, економічна частина, висновки, список використаних джерел, додатки

5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень)

Архітектура запропонованої згорткової нейронної мережі, UML-діаграма класів інформаційної технології, Загальний алгоритм роботи ядра, Робочі вікна програми, Порівняння достовірності розпізнавання аналогів.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	виконав прийняв
1-4	Колесницький О.К., к.т.н., доц. каф. КН	 18.09.2022	 15.12.2022
5	Нікіфорова Л. О., к. е. н., доц. каф. ЕПВМ	 22.09.2022	 14.12.2022

7. Дата видачі завдання 14.09 2022 року

КАЛЕНДАРНИЙ ПЛАН

№ етапу	Назва етапу	Термін виконання		Очікувані результати
		початок	кінець	
1	Аналіз сучасного стану розвитку систем розпізнавання жестів та обґрунтування доцільності розробки	14.09	20.09	Аналітичний огляд літературних джерел, задачі дослідження розділ 1
2	Обґрунтування методу розв'язання задачі, обґрунтування вибору інструментів технічної реалізації	25.09	30.09	Моделі, розділ 2
3	Проектування інформаційної технології. Програмна реалізація інформаційної технології.	09.10	25.10	розділ 2, 3
4	Підготовка економічної частини	03.11	28.11	розділ 4
5	Апробація та/або впровадження результатів дослідження	30.11	09.12	тези доповідей/впровадження
6	Оформлення пояснювальної записки, графічного матеріалу та презентації	11.12	18.12.	Пояснювальна записка, графічний матеріал, презентація

Студент
Керівник роботи

(підпис)

(підпис)

Каліцінський В.В.
Колесницький О.К.

АНОТАЦІЯ

УДК 004.8

Каліцінський В.В. Інформаційна технологія жестового керування WEB-додатком на основі згорткових нейронних мереж. Магістерська кваліфікаційна робота зі спеціальності 122 – комп'ютерні науки, освітня програма - комп'ютерні науки. Вінниця: ВНТУ, 2022. 103 с.

На укр. мові. Бібліогр.: 26 назв; рис.: 39; табл. 7.

Дана магістерська кваліфікаційна робота присвячена розробці програмного забезпечення розпізнавання жестів у браузерному середовищі, а саме керування веб-додатком. За основу для реалізації було обрано згорткову нейронну мережу. Інформаційна технологія реалізована мовою програмування JavaScript з використанням бібліотек React.js та Tensorflow.js. Вхідною інформацією системи є відео-потік з веб-камери, а вихідним результатом – виконання відповідної дії на раніше заданий жест. Розроблений додаток продемонстрував на 5% кращу достовірність розпізнавання в порівнянні з аналогами, а також довів свою модульність інтеграції з іншими веб-додатками.

Графічна частина складається з 6 плакатів.

У економічному розділі розраховано суму витрат на розробку та виготовлення нового технічного рішення, яка складає 1916461 гривень, спрогнозовано орієнтовану величину витрат по кожній з статей витрат, розраховано чистий прибуток, термін окупності витрат для виробника 0,58 роки та економічний ефект для споживача при використанні даної розробки.

Ключові слова: веб-додаток, жести, веб-камера, розпізнавання.

ABSTRACT

Kalitsynskyi V.V. Information technology for gesture control of a WEB application based on convolutional neural networks. Master's thesis in the specialty 122 - computer sciences, educational program - computer science. Vinnytsia: VNTU, 2022. 103 p.

In Ukrainian language. Bibliographer: 26 titles; fig .: 39; table 7.

This master's thesis is devoted to the development of gesture recognition software in a browser environment, namely the management of a web application. A convolutional neural network was chosen as the basis for implementation. Information technology is implemented with the JavaScript programming language using the React.js and Tensorflow.js libraries. The input information of the system is a video stream from a web camera, and the output is the execution of a corresponding action on a previously specified gesture. The developed module demonstrated 5% better recognition accuracy compared to its counterparts, and also proved its modular integration with other web applications.

The graphic part consists of 6 posters.

The economic section calculates the amount of costs for the development and manufacture of a new technical solution, which is 1916461 hryvnia, predicted the estimated cost of each of the cost items, calculated net profit, payback period for the manufacturer 0,58 years and the economic effect for consumers using this development.

Keywords: web application, gestures, webcam, recognition.

ЗМІСТ

ВСТУП	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ЖЕСТОВОГО КЕРУВАННЯ WEB-ДОДАТКОМ.....	9
1.1 Постановка задачі жестового керування WEB-додатком	9
1.2 Огляд сучасних підходів жестового керування та обґрунтування вибору нейромережевого методу	11
1.3 Обґрунтування вибору аналогу до інформаційної технології жестового керування WEB-додатком.....	13
1.4 Висновок до розділу 1	16
2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ЖЕСТОВОГО КЕРУВАННЯ WEB-ДОДАТКОМ.....	17
2.1 Обґрунтування вибору типу нейронної мережі.....	17
2.1.1 Багатошаровий персептрон.....	19
2.1.2 Рекурентні нейронні мережі	23
2.1.3 Згорткова нейронна мережа.....	24
2.2 Обґрунтування вибору архітектури згорткової нейронної мережі	25
2.3 Структура, математична модель та порядок функціонування згорткової нейронної мережі	27
2.4 Вимоги до вхідного сигналу згорткової нейронної мережі	38
2.5 Архітектура інформаційної технології керування WEB-додатком на основі згорткових нейронних мереж	39
2.6 Розробка алгоритму функціонування інформаційної технології жестового керування WEB-додатком.....	41
2.7 Висновок до розділу 2	42
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ЖЕСТОВОГО КЕРУВАННЯ WEB-ДОДАТКОМ НА ОСНОВІ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ.....	43

3.1 Обґрунтування вибору мови та середовища програмування для інформаційної технології жестового керування WEB-додатком	43
3.2 Реалізація інформаційної технології жестового керування WEB-додатком з використанням бібліотек React.js та Tensorflow.js	46
3.3 Специфіка програмної реалізація інформаційної технології жестового керування WEB-додатком.....	48
3.4 Висновок до розділу 3	53
4 ТЕСТУВАННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ РОБОТИ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ЖЕСТОВОГО КЕРУВАННЯ WEB-ДОДАТКОМ НА ОСНОВІ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ.....	54
4.1 Тестування роботи інформаційної технології жестового керування WEB-додатком	54
4.2 Аналіз результатів тестування інформаційної технології жестового керування web-додатком.....	63
4.3 Висновок до розділу 4	64
5 ЕКОНОМІЧНА ЧАСТИНА	65
5.1 Комерційний та технологічний аудит науково-технічної розробки	65
5.2 Прогнозування витрат на виконання науково-дослідної (дослідно-конструкторської) роботи	69
5.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором.....	75
5.4 Висновок до розділу 5	80
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	83
Додаток А (обов'язковий) Результат перевірки на плагіат в онлайн-системі UNICHECK	87
Додаток Б (обов'язковий) Лістинг програми	88
Додаток В (обов'язковий) ІЛЮСТРАТИВНА ЧАСТИНА.....	93
Додаток Г (довідниковий) Інструкція користувача	100
Додаток Д (довідниковий) Довідка про впровадження.....	103

ВСТУП

Актуальність теми дослідження. На сьогоднішній день користувачі ПК звикли до того, що взаємодія з комп'ютером відбувається за допомогою клавіатури та миші. Але бувають випадки, коли не зручно використовувати те чи інше. Наприклад, під час доповіді чи презентації. Тому почали задумуватись, чи є ще якісь альтернативні методи взаємодії з комп'ютером.

Одним із них є жестове керування. Ні для кого не секрет, що під час спілкування, люди звикли жестикулювати, тим сам вони проявляють свої емоції та акцентують увагу на деталях. Також до прикладу соціум прийшов до того, що було створено мову жестів, для взаємодії з людьми з обмеженими можливостями. Так чому таким же чином людина не може взаємодіяти з комп'ютером?

Розпізнавання жестів може полегшити керування технікою, у тих випадках, коли складно використовувати звичайні пристрої вводу і в цьому нам може допомогти комп'ютерний зір.

Комп'ютерний зір це доволі новий, але стрімко розвиваючий напрям в системах штучного інтелекту. Він дає змогу аналізувати та знаходити на зображеннях та відео певні образи та елементи. Все частіше для задач комп'ютерного зору застосовують штучні нейронні мережі.

Отже, задача розпізнавання жестів носить доволі важливий прикладний характер, і для її реалізації у веб середовищі відсутні достатньо ефективні класичні методи, що спричинює можливість застосування засобів штучного інтелекту для досягнення кращих показників достовірності розпізнавання жестів.

Зв'язок роботи з науковими програмами, планами, темами. Магістерська кваліфікаційна робота виконана відповідно до напрямку наукових досліджень кафедри комп'ютерних наук Вінницького національного технічного університету 22 К1 «Моделі, методи, технології та пристрої

інтелектуальних інформаційних систем управління, економіки, навчання та комунікацій» та плану наукової та навчально-методичної роботи кафедри.

Мета та завдання дослідження. Метою магістерської кваліфікаційної роботи є підвищення достовірності розпізнавання жестів за рахунок використання згорткових нейронних мереж, розширення функціональних можливостей завдяки запровадженню розпізнавання жестів двох та більше рук, а також реалізація модульності системи, щоб дати змогу інтегруватися з будь-якими іншими веб-додатками.

Задачі дослідження:

1. Аналіз предметної області та огляд сучасних методів розпізнавання жестів, обґрунтування доцільності розробки;
2. Проектування інформаційної технології жестового керування WEB-додатком;
3. Програмна реалізація інформаційної технології жестового керування WEB-додатком;
4. Тестування програмного забезпечення жестового керування WEB-додатком;
5. Розробка інструкції користувача.

Об'єктом дослідження є процес комп'ютеризованого розпізнавання жестів.

Предметом дослідження є інформаційна технологія та програмне забезпечення, що реалізовує процес розпізнавання жестів.

Наукова новизна одержаних результатів полягає в наступному:

1. Набула подальшого розвитку інформаційна технологія розпізнавання жестів, яка відрізняється використанням згорткової нейронної мережі, що дозволити підвищити достовірність розпізнавання жестів.

Практичне значення одержаних результатів полягає у наступному:

1. Розроблена архітектура функціонування модульної системи розпізнавання жестів у WEB-середовищі.
2. Розроблено загальний алгоритм покадрового аналізу жестів.
3. Розроблено програмне забезпечення таким чином, що воно являє собою модуль, який можна інтегрувати(додати) до вже існуючого додатку без складнощів.

Достовірність теоретичних положень магістерської кваліфікаційної роботи підтверджується строгістю постановки задач, коректним застосуванням математичних методів під час доведення наукових положень, строгим виведенням аналітичних співвідношень, порівнянням результатів з відомими.

Особистий внесок здобувача. Усі результати, наведені у магістерській кваліфікаційній роботі, отримані самостійно.

Апробація результатів роботи. Результати роботи були апробовані на двох конференціях:

1. Науково-технічна «Молодь в науці: дослідження, проблеми та перспективи (МН-2021)», доповідь «Розробка інформаційної системи жестового керування веб-додатком на основі згорткових нейронних мереж.»[1];
2. Науково-технічна «Молодь в науці: дослідження, проблеми та перспективи (МН-2022)», доповідь «Розробка інформаційної технології жестового керування веб-додатком на основі згорткових нейронних мереж»[2];

Публікації: За результатами магістерської кваліфікаційної роботи опубліковано 2 тези доповіді конференцій[1, 2], подано заявку на твір(комп'ютерну програму) та отримано довідку про впровадження на підприємстві «ДСофтвер».

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ЖЕСТОВОГО КЕРУВАННЯ WEB-ДОДАТКОМ

1.1 Постановка задачі жестового керування WEB-додатком

Мишка та клавіатура відіграють значну роль у взаємодії людини і комп'ютера. Однак, завдяки швидкому розвитку програмного та апаратного програмного забезпечення, все частіше почало виникати потреба у нових методах людино-комп'ютерної взаємодії (ЛКВ). Тому, увага була прикута до таких технологій, як розпізнавання жестів та розпізнавання мови у сфері ЛКВ.

Людино-машинна взаємодія — це міждисциплінарний науковий напрямок, що вивчає взаємодію між людьми і машинами. Предметом ЛКВ є дослідження, планування і розробка методів взаємодії людини з машиною, де в ролі машини може виступати ПК, комп'ютерна система значних масштабів, система управління процесами і т.д. [3]. Під процесом взаємодії розуміється будь-яка комунікація між людиною і машиною. Одним із методів ЛКВ, який отримав широке поширення в останні роки, є взаємодія, засноване на жестах людини [4, 5].

Жести — це певні рухи тіла, які є невербальною способом передачі інформації. Рух пальців, рук, плечей, голови, міміка особи: всі перераховані є жестами. За допомогою жестів людина може передати самостійну інформаційну одиницю, доповнити словесний ряд, передати почуття і пр. Звичайно жести поділяються на статичні (сприймаються одномоментно) і динамічні (що сприймаються в деякий період часу) жести, які мають певну інтерпретацію в ручних абетках і безконтактних людино-машинних інтерфейсів. У ЛКВ жести використовуються для передачі інформації в комп'ютер, який в подальшому може використовуватися для ідентифікації людини, управління авто, літальним апаратом, персонаж в грі і т.д.

Комп'ютерне розпізнавання жестів може забезпечити більш природний інтерфейс ЛКВ. Розпізнавання жестів дозволяє комп'ютерам бути більш доступними для людей з обмеженими фізичними можливостями та робить взаємодію природнішою в іграх або віртуальному світі 3D. Використання розпізнавання жестів, дасть змогу вказувати пальцем на екран комп'ютера, а курсор буде відповідно переміщатися[6].

Розрізняють дві категорії жестів: динамічні і статичні. Динамічний жест - це рухливий жест, представлений послідовністю зображень. Статичний жест - це конкретна конфігурація руки або пози, представлена єдиним зображенням[7].

Чому люди прагнуть використовувати жести, а не просто торкатися пристрою? Прагнення до безконтактного досвіду та питання гігієни є головними поштовхом попиту на безконтактні технології. Розпізнавання жестів також може забезпечити кращу ергономіку для споживчих пристроїв. Ще одним рушієм ринку є зростання біометричних систем у багатьох сферах життя людей - від автомобілів до будинків і магазинів.

Під час пандемії не дивно, що люди неохоче використовують сенсорні екрани в громадських місцях. В інших випадках натискання маленьких піктограм або випадкове натискання неправильного поля збільшує розчарування та змушує людей шукати кращого досвіду взаємодії.

Розпізнавання жестів рук в режимі реального часу для комп'ютерних взаємодій – це наступний крок у технологічній еволюції, який потребують для сучасні споживчі. Окрім випадків, коли вам не зручно торкатися обладнання, відстеження жестів можна застосовувати в середовищах доповненої та віртуальної реальності, розпізнаванні мови жестів та іграх[8].

У даній роботі буде розглядатись задача керування WEB-додатком на основі комп'ютерного зору.

1.2 Огляд сучасних підходів жестового керування та обґрунтування вибору нейромережевого методу

Комп'ютерний зір, як науковий напрям, тісно пов'язаний з теорією розпізнавання образів та машинним навчанням, але належить до більш спеціалізованої технології створення штучних систем, які отримують інформацію з зображень[8].

Область комп'ютерного зору може бути охарактеризована як молода та різноманітна. І, хоча існують більш ранні роботи, можна сказати, що тільки з кінця 1970-х почалось інтенсивне вивчення цієї проблеми, коли комп'ютери змогли керувати обробкою великих наборів даних, таких як зображення. Однак, ці дослідження зазвичай починались з інших галузей, і, відповідно, нема стандартного формулювання проблеми комп'ютерного зору. Також, і це навіть більш важливо, нема стандартного формулювання того, як повинна вирішуватись проблема комп'ютерного зору. Замість того, існує маса методів для вирішення різноманітних строго визначених задач комп'ютерного зору, де методи часто залежать від задач і рідше коли можуть бути узагальнені для широкого кола застосування. Багато з методів та застосувань все ще знаходяться на стадії фундаментальних досліджень, але все більша кількість методів знаходить застосування в комерційних продуктах, де вони часто складають частину складнішої системи, яка може вирішувати складні задачі (наприклад, в галузі медичних зображень або вимірювання та контролю якості в процесах виробництва). У більшості практичних застосувань комп'ютерного зору, комп'ютери попередньо запрограмовані для вирішення окремих задач, але методи, що базуються на знаннях, стають все більше узагальненими[9].

Важливою частиною в області машинного навчання, є автоматичне планування або прийняття рішень в системах, які можуть виконувати механічні дії, такі як пересування робота через деяке середовище. Цей тип обробки зазвичай потребує вхідних даних, що надаються системами

комп'ютерного зору, діють як відеосенсор і надають високорівневу інформацію про середовище та роботу.

Машинне навчання застосовують у ряді обчислювальних задач, у яких розробка та програмування явних алгоритмів з доброю продуктивністю є складною або нездійсненною; до прикладів таких додатків належать фільтрування електронної пошти, виявлення мережних вторгників або зловмисних інсайдерів, що добиваються витоків даних, оптичне розпізнавання символів (ОПС), навчання ранжуванню та комп'ютерний зір[10].

Класичні методи розпізнавання об'єктів, комп'ютерного зору і машинного навчання ділять умовно на три групи:

- методи фільтрації;
- методи логічного аналізу;
- методи навчання.

Для завдань комп'ютерного зору, фільтрація зазвичай використовується для попередньої обробки зображення перед аналізом його морфологічних ознак, але зустрічаються і задачі, в яких достатньо буде використати тільки фільтрацію (наприклад, в задачах машинного зору).

За допомогою фільтрації можна одержати результати, що будуть придатні для обробки, але більшість завдань напрямку комп'ютерного зору потребують аналізу безпосередньо структури зображення, для цього досліджуються і впроваджуються методи логічного аналізу зображень. Методи математичної морфології є наслідком теорії аналізу і обробки геометричних структур. Ці методи реалізуються елементарними операціями. Операції двійкової морфології є певним перетворенням впорядкованої множини або деякої області зображення за допомогою структурного елемента. Структурний елемент представляється у вигляді двійкового зображення довільного розміру і структури, але як правило використовуються симетричні елементи, такі як коло фіксованого діаметру або прямокутник фіксованого

розміру. Методи математичної морфології дають можливість видаляти шуми з двійкових зображень, а також впроваджують алгоритми пошуку контурів, але на практиці ці методи використовуються в поєднанні з іншими алгоритмами.

Штучна нейронна мережа представляє концептуальну модель біологічної нейронної мережі і побудована із пов'язаних між собою різним чином шарів штучних нейронів, які організують загальну структуру і безпосередньо впливають на роботу один одного. У більшості архітектур штучних нейронних мереж, активність нейрона визначається сукупністю зовнішнього сумарного впливу інших нейронів на даний нейрон. З моменту започаткування технології штучних нейронних мереж розвивалися досить окремо в порівнянні з класичними методами. Завдання нейронної мережі полягає в перетворенні вхідного вектора у вихідний вектор, що регулюється вагою і топологією мережі.

Аналіз методів вирішення задач комп'ютерного зору обґрунтував переваги нейромережевого методу над усіма розглянутими. Принцип роботи нейронів штучної мережі дає можливість виявити його активність, що визначається вагами і його функцією активації.

1.3 Обґрунтування вибору аналогу до інформаційної технології жестового керування WEB-додатком

Програмні продукти, у яких використовується розпізнавання жестів, це не новина, але це представлено у вузьких ситуаціях, де інтелектуальна ситстема має тільки декілька шляхів взаємодії, за основу можна взяти такі відомі приклади:

GestureTek займається технологією розпізнавання жестів і створює мультисенсорні дисплеї, плівки, таблиці, накладки, інтерактивні підлоги, стіни, вікна та екрани. GestureTek відзначений понад 4000 нагородами за

досягнення в інтерактивних дисплеях на підлозі, вікнах, стінах, барних стільницях, столах і прилавках, у торгових центрах, на торгових виставках і виставках, у рекламних і цифрових рекламних мережах, парках розваг і атракціонах, готелях і нічних клубах, корпоративні місця та громадських місцях. Він має мультизапатентовану технологію управління відео жестами (VGC), яка дозволяє користувачам керувати мультимедійним вмістом, отримувати доступ до інформації, маніпулювати спецефектами, навіть занурюватися в інтерактивний 3D віртуальний світ. Компанія Qualcomm придбала всю інтелектуальну власність, пов'язану з її технологією розпізнавання жестів для рішень мобільної та побутової електроніки[11].

Судячи з відгуків аудиторії користувачів, є велика розбіжність в точності розпізнавання, деякі надавали результат 70%, а інші користувачі надсилали звіти з 95% точністю розпізнавання жестів. Виробник поясню, що є багато факторів впливу на точність, а саме освітлення в приміщенні та апаратні потужності обчислювальної машини користувача.



Рисунок 1.1 – GestureTek Illuminate Display

Іншим прикладом є TableFX — це технологія обчислення поверхні столу, яка перетворює звичайні горизонтальні поверхні на інтерактивні столи,

інтерактивні панелі та інтерактивні стільниці. Ця настільна технологія є інструментом для представлення інтерактивної настільної реклами, розваг, ігор та спеціальних ефектів. Використовуючи передову технологію камери та запатентоване програмне забезпечення для відстеження тіла, TableFX створює динамічний інтерактивний стіл з інтерактивним управлінням жестами. Користувачі можуть взаємодіяти з захоплюючими візуальними ефектами, іграми та рекламою простими рухами рук і тіла. Цифрові столи GestureTek також можуть постачатися з мультитач-інтерфейсом на екрані настільного комп'ютера, забезпечуючи справді унікальний інтерактивний обчислювальний інтерфейс поверхні столу[12]. TableFX є частиною серії GestureFX, яка також включає GroundFX, WallFX і ScreenFX. Інтерактивний дисплей продемонстровано на рис.1.2.

Використовуючи передову технологію камери та запатентоване програмне забезпечення для відстеження тіла, TableFX створює динамічну інтерактивну таблицю з управління жестами. Користувачі можуть взаємодіяти із захоплюючими візуальними ефектами, іграми та рекламою простими рухами рук і тіла. Під час тестування даного продукту, як заявляє представник департаменту розробки, точність розпізнавання жестів пристрою сягає до 80-85%.



Рисунок 1.2 – TableFX Interactive Display

ManoMotion – забезпечує 2D/3D відстеження рук і керування жестами в реальному часі. Мінімальні вимоги до обчислювальної техніки та потужності. Стандартні датчики, без спеціального обладнання. Інноваційна архітектура спрямована не лише на те, щоб запропонувати відстеження рук світового класу, але й використати можливість взаємодії реальних рук і віртуальних об'єктів[13]. Основна інновація ManoMotion представляє нову запатентовану концепцію, яка використовує функцію пошуку набору золотих рук, що дає змогу досягти безпрецедентної ефективності, застосовуючи відстеження рук за допомогою AI/ML.

1.4 Висновок до розділу 1

Отже проаналізувавши існуючі програмні продукти та реалізації, було поставлено за мету реалізувати інтелектуальний модуль, який зможе показати достовірність розпізнавання 85% і вище, в порівнянні з аналогами, а також розробити програмне забезпечення, яке дасть змогу інтегруватися з будь-яким WEB-додатком.

2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ЖЕСТОВОГО КЕРУВАННЯ WEB-ДОДАТКОМ

2.1 Обґрунтування вибору типу нейронної мережі

Нейронна мережа – один із напрямків штучного інтелекту, мета якого змоделювати аналітичні механізми, що здійснюються людським мозком. Завдання, які вирішує типова нейронна мережа – класифікація, передбачення та розпізнавання. Нейронні мережі здатні самостійно навчатися і розвиватися, будуючи свій досвід на помилках. Нейромережі це послідовність нейронів, з'єднаних між собою синапсами.

Структура нейронної мережі прийшла у світ програмування прямо з біології. Завдяки такій структурі машина знаходить здатність аналізувати і навіть запам'ятовувати різну інформацію. Також нейронні мережі здатні як аналізувати вхідну інформацію, а й відтворювати її з власної пам'яті.

Іншими словами, нейромережа це машинна інтерпретація мозку людини, в якому знаходяться мільйони нейронів, що передають інформацію у вигляді електричних імпульсів.

Біологічна мережа являє собою нервову систему людини, та сукупність нейронів в мозку, завдяки якій ми думаємо, приймаємо ті чи інші рішення, сприймаємо світ навколо нас. Біологічний нейрон – це спеціальна клітина, що складається з ядра, тіла і відростків, до того ж вона має тісний зв'язок з тисячами інших нейронів. Через цей зв'язок то і діло передаються електрохімічні імпульси, що призводять всю нейронну мережу в стан збудження або навпаки спокою. Наприклад, якась приємна і водночас хвилююча подія (зустріч коханої людини, перемога в змаганні і т. д.) породить електрохімічний імпульс у нейронної мережі, яка розташовується в нашій голові, що призведе до її збудження. Як наслідок, нейронна мережа в нашому

мозку своє збудження передасть до інших органів нашого тіла і призведе до підвищеного серцебиття, більш частого моргання очей, тощо.

На рисунку 2.1 наведена спрощена модель біологічної нейронної мережі мозку. Видно, що нейрон складається з тіла клітини і ядра, тіло клітини, в свою чергу, має безліч відгалужень волокон, названих дендритами. Довгі дендрити називаються аксонами і мають протяжність багато більшу, ніж показано на малюнку, за допомогою аксонів здійснюється зв'язок між нейронами, завдяки ним і працює біологічна нейронна мережа.

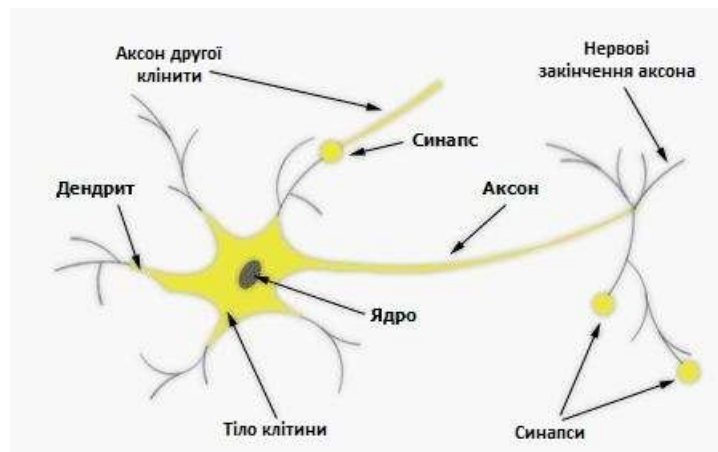


Рисунок 2.1 – Модель біологічного нейрона

Під штучними нейронними мережами прийнято розуміти обчислювальні системи, які мають здібності до самонавчання, поступового підвищення своєї продуктивності. Основними елементами структури нейронної мережі є:

Класичні методи розпізнавання об'єктів, комп'ютерного зору і машинного навчання ділять умовно на три групи:

- Штучні нейрони, що представляють собою елементарні, пов'язані між собою одиниці;
- Синапс – це з'єднання, що використовується для відправки-отримання інформації між нейронами.;
- Сигнал – власне інформація, що підлягає передачі.

Нарахоують чимало парадигм нейронних мереж, які можуть використовуватись для задачі апроксимації функцій. Серед них можна звернути увагу на такі нейронні мережі, як одношаровий персептрон, багатошаровий персептрон, мережа Хопфілда, Хемінга, згорткова нейронна мережа. Мережу Кохонена можна одразу відкинути, оскільки вона працює у режимі кластеризації. Для програмної реалізації інформаційної технології потрібно обрати таку нейронну мережу, яка була б найбільш ефективною і дала б можливість досягти поставлену у роботі мету – реалізувати інформаційну технологію, яка зможе продемонструвати достовірність розпізнавання 85% і вище, в порівнянні з аналогами.

2.1.1 Багатошаровий персептрон

Під час досліджень функцій людського мозку, дослідники спроектували апаратні моделі біологічного нейрона і системи його з'єднань. Коли нейрофізіологи досягли більш глибокого розуміння нервової системи людини, ці спроби стали трактуватися, як дуже грубі апроксимації. Проте, було досягнуто вражаючих результатів, стимулювавши подальші дослідження, що призвели до створення складніших мереж. Перші вивчення штучних нейронних мереж було зроблено в 1943 р. Згодом Маккалох і Пітс аналізували мережні парадигми для розпізнавання зображень, що піддаються зсувам і поворотам, використовуючи при цьому просту нейронну модель (рис. 2.1.). Елемент \sum домножує кожний вхід x_i на вагу w_i і сумує входи.

Якщо ця сума більше певного заданого порогового значення, вихід дорівнює одиниці, інакше – нулю. Ці системи отримали назву персептрон.

На рисунку 2.2 представлена схема багатошарового персептрона. Нейрони показані кружечками, зв'язки між нейронами – стрілками.

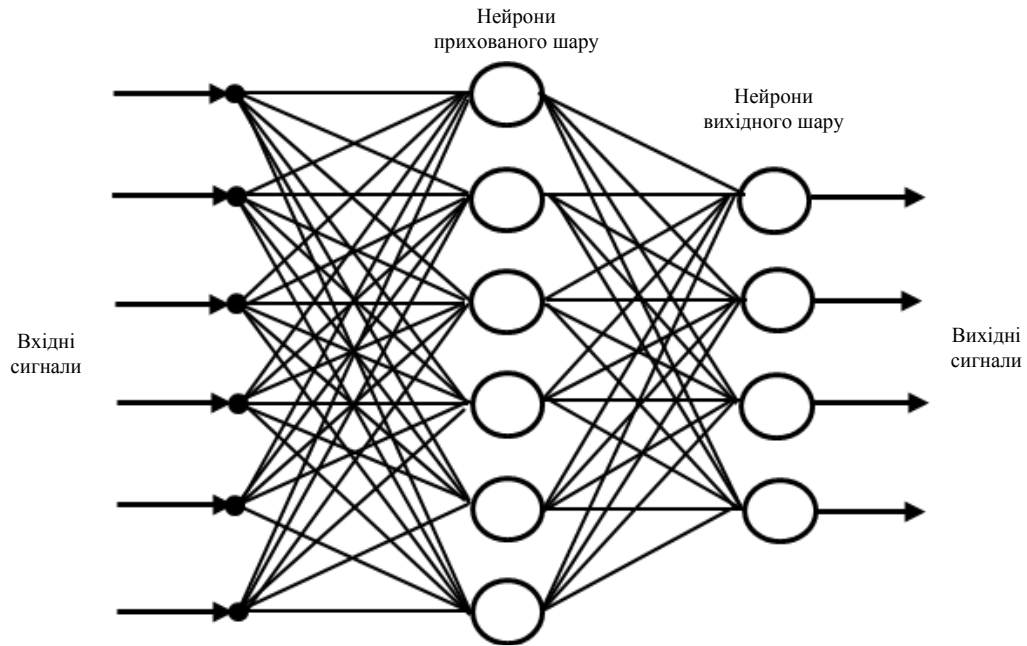


Рисунок 2.2 – Структура багатошарового перцептрона

Позначимо входи n -го шару нейронів $x_j^{[n]}$, вагові коефіцієнти w_{ij} . Нейрони цього шару обраховують відповідні комбінації a_i :

$$a_i^{[n]} = \sum_j w_{ij}^{[n]} x_j^{[n]} \quad (2.1)$$

і передають їх на наступний шар, пропустивши через нелінійну функцію активації:

$$x_i^{[n+1]} = f(a_i^{[n]}) \quad (2.2)$$

Нелінійна функція f може мати різний вигляд, як зображено на рисунку 2.2. Найпоширенішою є нелінійна функція з насиченням, так звана логістична функція або сигмоїда (тобто функція S-подібного виду):

$$f(x) = \frac{1}{1 + e^{-\alpha x}}, \quad (2.3)$$

При зменшенні α сигмоїда стає пологішою, в межі при $\alpha=0$ перетворюючись в горизонтальну лінію на рівні 0.5, при збільшенні α сигмоїда наближається до функції одиничного стрибка з порогом T в точці $x = 0$. З виразу очевидно що для сигмоїди, вихідне значення нейрона належить діапазону $[0,1]$. Одна з цінних властивостей сигмоїдальної функції – простий вираз для її похідної.

$$f'(x) = \alpha \cdot f(x) \cdot (1 - f(x))$$

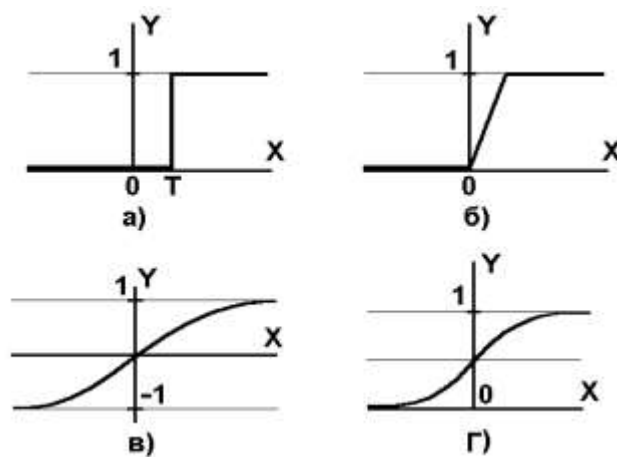


Рисунок 2.3 – а) функція одиничного стрибка, б) лінійний поріг (гістерезис), в) гіперболічний тангенс; г) сигмо їда

Щоб побудувати алгоритм навчання, потрібно знати похідну помилки по кожному з ваг мережі:

$$\frac{\partial E}{\partial w_{ij}^{[n]}} = \frac{\partial E}{\partial a_i^{[n]}} \frac{\partial a_i^{[n]}}{\partial w_{ij}^{[n]}} \equiv \delta_i^{[n]} x_j^{[n]}, \quad (2.4)$$

Тобто, внесок у загальну помилку кожної ваги підраховуються локально, простим множенням зв'язки нейрона $\delta_i^{[n]}$ на значення відповідного входу.

Входи кожного шару обчислюються послідовно від першого шару до останнього під час прямого розповсюдження сигналу:

$$x_i^{[n+1]} = f\left(\sum_j w_{ij}^{[n]} x_j^{[n]}\right), \quad (2.5)$$

а зв'язки кожного шару підраховуються під час зворотного поширення помилки від останнього шару до першого:

$$\delta_i^{[n]} = f'(a_i^{[n]}) \sum_k w_{ki}^{[n+1]} \delta_k^{[n+1]}, \quad (2.6)$$

Багатошаровий перцептрон застосовується в таких задачах:

- розпізнавання образів;
- класифікація;
- прогнозування;
- експертні системи.

Недоліками багатошарового перцептрона є:

1) багатокритеріальна задача оптимізації в методі зворотного розповсюдження представляється як набір однокритеріальних. Тобто на кожній ітерації змінюються значення параметрів мережі, покращуючи роботу лише з одним прикладом навчальної вибірки. Такий підхід суттєво зменшує швидкість навчання;

2) класичний метод зворотного розповсюдження належить до алгоритмів з лінійною збіжністю, для підвищення швидкості збіжності потрібно використати матриці других похідних функції помилки.

2.1.2 Рекурентні нейронні мережі

Рекурентними називають штучні нейронні мережі, в яких, поряд із прямими зв'язками, направленними від входів (рецепторів) мережі до її виходів (ефекторів), є зворотні, що мають протилежний напрямок. На відміну від штучних нейромереж прямого поширення, які здійснюють статичну проекцію поданих на входи векторів даних (стимулів) у вихідні вектори (реакції), рекурентні нейромережі є динамічними системами, що оперують з послідовностями вхідних даних, перетворюючи їх на послідовності реакцій.

Поведінка рекурентних нейромереж віддзеркалює набуті при навчанні стереотипи, що робить їх близькими до цілеспрямованих адаптивних динамічних систем, запрограмованих на досягнення заздалегідь визначених цілей. Але на відміну від останніх програмування поведінки рекурентних нейромереж здійснюється шляхом навчання на прикладах, що не потребує формального визначення цілей. Вони здатні ефективно діяти в умовах невизначеності, зокрема, вирішувати задачі адаптивного керування поведінкою складних систем у нестационарному оточенні, приймати оперативні рішення в системах ситуаційного управління тощо.

Рекурентна нейронна мережа організована за принципом збереження результуючого сигналу шару та його передачі знову на вхід, для прогнозування нового результату шару. В даній мережі перший шар подібний до нейронної мережі прямого поширення з добутком суми ваг і функцій. Навчання починається після першого обчислення ваг. Тобто до опрацювання наступної ітерації кожен нейрон запам'ятає певну інформацію, яку він зберіг на попередній ітерації. У ході виконання цього процесу дати можливість нейронній мережі працювати над прямим поширенням похибки та зберігати інформацію, яка необхідна для наступного використання. Однією із найпоширеніших сфер застосування рекурентних нейронних мереж є здійснення машинного рерайту іншомовних речень і творів.

Але для розпізнавання жестів цей вид мережі занадто ефективний, що призведе до перенавчання мережі. Тобто ті параметрами, з якими мережа гарно працювала, вона захоче покращити, і буде шукати залежності там де їх немає і це призведе до погіршення точності і продуктивності.

2.1.3 Згорткова нейронна мережа

Згорткові нейронні мережі схожі до нейронних мереж прямого поширення, в яких нейрони мають тренувальну вагу та упередження. Цей тип мережі, як правило використовують для обробки зображень та сигналів.

В порівнянні з мережею прямого поширення, де кожен вхідний нейрон з'єднується з вихідним нейроном в наступному шарі, в згорткових мережах для одержання вихідних значень використовуються згортки над кожним вхідним шаром. Для операції згортки застосовується матриця ваг малого розміру, яка «пробігається» по всьому поточному шарі, створюючи після кожного зсуву сигнал активації для нейрона наступного шару з аналогічною позицією. Така матриця отримала назву - ядро згортки.

Згорткова нейронна мережа за рахунок застосування операції згортки дозволяє водночас зменшити кількість інформації, що зберігається в пам'яті, за рахунок чого краще справляється з картинками більш високої роздільної здатності, і виділити опорні ознаки зображення, такі як ребра, контури або грані. На наступному рівні обробки з цих ребер і граней можна розпізнати повторювані фрагменти текстур, які далі можуть скластися в фрагменти зображення.

По суті кожен шар нейронної мережі використовує власне перетворення. Якщо на перших шарах мережа оперує такими поняттями як "ребра", "грані" і т.п, то далі використовуються поняття "текстура", "частини об'єктів". В результаті такого опрацювання ми можемо правильно класифікувати картинку або виділити на кінцевому етапі потрібний об'єкт на зображенні[15].

На рисунку 2.4 зображений, один із етап під час якого мережа аналізує зображення частинами. Тобто вхідний сигнал надається частково, як фільтр. Це дає змогу мережі запам'ятовувати зображення порціями та проводити необхідні розрахунки. Ці обрахунки включають в себе перетворення зображення з колірної моделі RGB або HSI в сірий масштаб. Згодом, відмінності в значеннях пікселів допоможуть визначити границі зображень, які можна буде класифікувати на різні категорії.

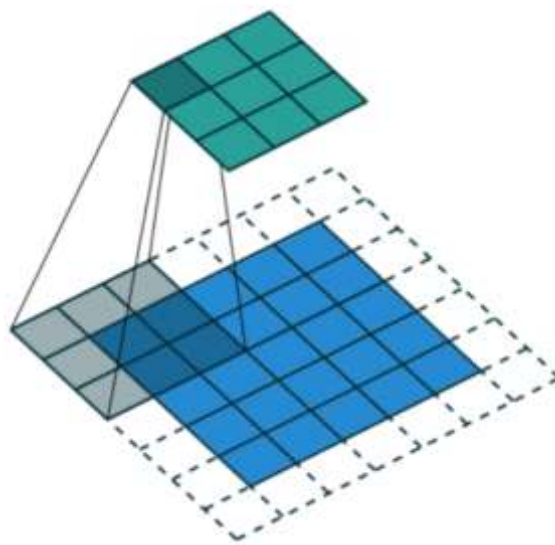


Рисунок 2.4 - Накладання фільтру згортки на зображення

Таким чином, розглянувши переваги та недоліки зазначених трьох видів нейронних мереж, оберемо для програмної реалізації згорткову нейронну мережу, так як вона має найточніші механізми аналізу зображення, які допоможуть вирішити поставлену задачу.

2.2 Обґрунтування вибору архітектури згорткової нейронної мережі

Згорткові нейронні мережі(CNN) застосовуються досить широко і в різних областях. Першим і, по суті, найбільш тривіальним завданням, яке

навчилися вирішувати за допомогою нейронних мереж, стала класифікація зображень. Класифікації за допомогою CNN активно застосовуються в медицині: можна навчити нейронну мережу класифікації хвороб або симптомів, наприклад, для МРТ-діагностики рис. 2.5.

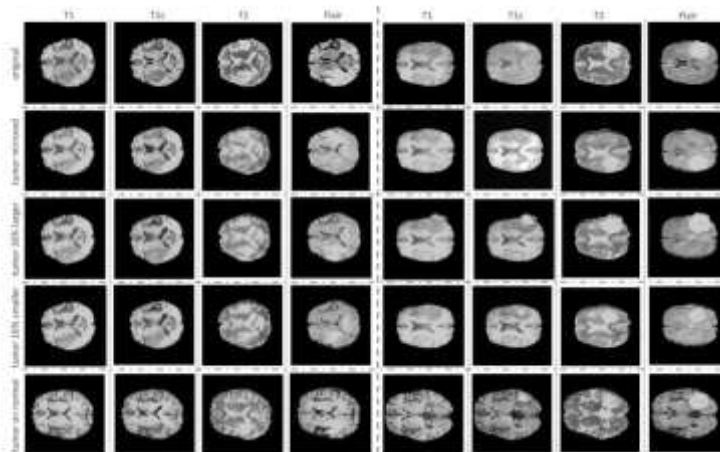


Рисунок 2.5 – Приклад МРТ знімку використаного для навчання CNN

В агробізнесі розробляється і впроваджується методика аналізу та розпізнавання зображень, при якій дані отримують від відкритих супутників, таких як LSAT, і використовують для прогнозування майбутньої врожайності конкретних земель рис. 2.6.



Рисунок 2.6 – Приклад результату аналізу врожайності земельних ділянок на основі супутникового знімку

Розпізнавання об'єктів на фото і відео за допомогою нейронних мереж застосовується в безпілотному транспорті, відеоспостереженні, системах контролю доступу та системах "розумного будинку". Зустрічається також масковане розпізнавання об'єктів з виділенням контуру, при якому ми також можемо отримувати чіткі контури об'єкта за допомогою згорткових нейронних мереж. Також нейронні мережі можна використовувати для виділення людей або окремих частин тіла людини на фото або відео, для побудови їхніх скелетів, поз. Такий підхід застосовується, наприклад, для відеоаналітики[14].

Отож з практики користувачів CNN можна зробити висновок, що згорткова мережа чудово підходить для вирішення поставленої задачі розпізнавання жестів рук людини.

2.3 Структура, математична модель та порядок функціонування згорткової нейронної мережі

Більшість алгоритмів машинного навчання включають в себе гіперпараметри – змінні, значення яких встановлюється перед оптимізацією параметрів моделі. Встановлення значень гіперпараметрів можна розглядати як вибір моделі, яку необхідно використати для певного набору даних. Гіперпараметри часто встановлюються вручну або обираються за допомогою деякого алгоритму пошуку оптимальних параметрів. Нейромережі можуть мати багато гіперпараметрів, зокрема ті, які визначають структуру самої мережі та процедуру її навчання[15].

В процесі навчання мережі результуюча модель буде залежати не тільки від обраної структури, але і від методу, який використовується для встановлення параметрів мережі. Сам спосіб навчання може мати багато гіперпараметрів. Стохастичний градієнтний спуск оновлює параметри мережі з використанням градієнтного спуску на підмножині навчальних даних, що періодично переміщується або вважається нескінченною. Нехай $t < T$ – певна

ітерація, тоді параметри θ оновлюватимуться на даній ітерації, як показано в рівнянні 2.7:

$$\theta^{(t)} \leftarrow \theta^{(t-1)} - \epsilon_t \frac{1}{B} \sum_{t'=Bt+1}^{B(t+1)} \frac{\partial L(z_{t'}, \theta)}{\partial \theta}, \quad (2.7)$$

де z_t – це приклад тренувальної множини. Гіперпараметрами в даному випадку є: функція втрат L , темп навчання на ітерації t – ϵ_t , розмір пакету прикладів B , та кількість ітерацій T .

Темп навчання

Темп навчання ϵ_t вирішує наскільки швидко градієнт оновлення прямує до глобального мінімуму. Якщо темп навчання дуже низький, модель буде сходитися надто повільно, а якщо навпаки, тоді модель буде неправильно натренована. Коли ϵ_t змінюється з ітераціями, необхідно визначити початкову швидкість навчання ϵ_0 та спосіб обчислення ϵ_t . Для нейронних мереж із стандартизованими вхідними даними або даними, що мають значення в діапазоні $[0,1]$, ϵ_0 зазвичай встановлюється в межах 1 та 10^{-6} , а найкращим значенням для першої спроби є 0,01. Проте, зважаючи на важливість ϵ_0 , даний параметр потребує точного налаштування[16].

Темп навчання ϵ_t зазвичай зменшується зі зміною ітерації. Якщо використати загальновідоме евристичне припущення, спочатку потрібно знайти гідне початкове значення темпу навчання, після чого його треба точно налаштувати. Одним з підходів є ініціалізація $\epsilon_t = \epsilon_0$, де $t < \tau$ (τ - новий гіперпараметр), після чого нове значення темпу навчання розраховується за формулою $\epsilon_t = \epsilon_0 / t^\alpha$ (α - ще один гіперпараметр). Необхідно використовувати менші значення α , коли функція, на якій шукається глобальний мінімум, не є опуклою, і використовується певний метод градієнтного усереднення (наприклад, імпульс). Також загальноприйнятою практикою є адаптивне встановлення значення τ в залежності від ітерації.

Функція втрат

Функція втрат порівнює вже відомі вихідні дані для тренувального прикладу та значення, яке було отримане під час навчання. В загальному випадку функція втрат – квадрат віддалі між двома точками, яка визначається за рівнянням 2.8:

$$L = \frac{1}{2} \sum_i (y_i - z_i)^2, \quad (2.8)$$

де y_i – i -те значення вихідного шару мережі, а z_i - це i -те значення відповіді, що визначені в тренувальній множині. Коефіцієнт $1/2$ використовується для того, щоб зробити градієнт більш простим. Коли вихідні значення нейронної мережі розглядаються як розподіл імовірності (наприклад, використовується вихідний шар softmax), загальним і теоретично доцільним є використання функції втрат перехресної ентропії, яка виглядає як рівняння 3:

$$L = - \sum_i y_i \log(z_i), \quad (2.9)$$

Число тренувальних ітерацій

Часто використовуваним способом визначення T є принцип ранньої зупинки. Рання зупинка працює за принципом припинення тренування після того, як продуктивність навчання на етапі валідації перестав збільшуватись (тобто похибка починає поступово рости). Це доволі потужний засіб попередження перенавчання, оскільки налаштуванням інших гіперпараметрів можна знехтувати.

Імпульс

Найпоширенішою методикою є «гладке» оновлення градієнта, використовуючи інтеграційний фільтр разом з параметром β , як показано у функції 2.10:

$$\bar{g} \leftarrow (1 - \beta)\bar{g} + \beta \frac{\partial L(z_t, \theta)}{\partial \theta}, \quad (2.10)$$

де \bar{g} може бути використаний замість «справжнього» оновлення градієнта при застосуванні градієнтного спуску. Деякі математичні підходи можуть забезпечити більш швидкий спуск градієнта при використанні відповідного імпульсу, однак для випадкового градієнтного спуску, стандартне градієнтне оновлення ($\beta = 1$) з гармонійно зменшеним темпом навчання є оптимальними.

Сама структура нейронної мережі містить в собі численні гіперпараметри, включаючи розмір і нелінійність кожного шару. Чисельні властивості ваг також певним чином обмежені, а їх ініціалізація може сильно впливати на показники продуктивності. Попередня обробка вхідних даних також може бути важливою для забезпечення спуску градієнта. Багато гіперпараметрів можуть варіюватися в різних шарах [16].

Кількість прихованих шарів

Значна кількість прихованих шарів дає змогу нейронній мережі добре натренуватись, тому навіть при використанні регуляризації дуже важливо створювати приховані шари необхідної розмірності. Як правило, використовують однакову розмірність для всіх прихованих шарів, так як цей спосіб покращує роботу. В доповненні, при використанні першого прихованого шару, який більше, ніж вхідний шар – є сприятливою практикою. Використовуючи безконтрольне попереднє навчання, шари слід бути набагато більшими, ніж при виключно контрольованій оптимізації.

Відсікання вагових компонент

Зачасту, щоб зменшити перенавчання, до критерію тренування додають регуляризацію ваг мережі. Для збереження значення ваг мережі θ близькими

до нуля, регуляризація L2 додає $\lambda_2 \sum_i \theta_i^2$, а L1 додає $\lambda_1 \sum_i |\theta_i|$, де λ_2 та λ_1 - множники лагранжа, які визначають, наскільки важливою є дана регуляризація.

Нелінійність

Найчастіше використовуваними нелінійностями є:

- сигмоїда $1/(1 + e^{-a})$, яка плавно конвертує вхідні дані у діапазоні $[0,1]$;
- формула, що рівносильна сигмоїді в діапазоні $[-1,1]$;
- випрямляч (ReLU) $\max(0, a)$, або порогова функція.

Сигмоїда може призвести до проблем, у випадку використання глибокими мережами з наглядачем на вихідному рівні.

Згортка є відносною елементарною операцією, яка базується на ядрі – невелика матриця ваг, що аналізує вхідні дані та виконує почергове множення елементів цільової частини вхідного сигналу, а потім резюмує усі значення у вигляді одного вихідного пікселя. Ядро повторює цей процес для кожного місця, де воно проходить, перетворюючи одну 2D матрицю ознак на іншу. Вихідні ознаки є, зважені суми вхідних ознак, розташованих приблизно в одному місці вхідного пікселя на вхідному шарі. Необмежено від того, чи знаходиться ознака входу в границях чи визначається безпосередньо, а також чи знаходиться вона в області ядра, що створює вихідний сигнал, чи ні – розмір ядра прямо засвідчує про те, скільки вхідних ознак поєднуються для створення нової ознаки виходу.

3 ₀	3 ₁	2 ₂	1	0
0 ₂	0 ₂	1 ₀	3	1
3 ₀	1 ₁	2 ₂	2	3
2	0	0	2	2
2	0	0	0	1

12.0	12.0	17.0
10.0	17.0	19.0
9.0	6.0	14.0

Рисунок 2.7 – Приклад згортки та створення нового ядра

Згортковий шар відносно гостро контрастує з повністю підключеним шаром. У наведеному вище прикладі $5 \times 5 = 25$ вхідних ознак і $3 \times 3 = 9$ вихідних ознак. У випадку стандартного повнозв'язного шару матриця мала би $25 \times 9 = 225$ параметрів, причому кожна ознака виходу була би зваженою сумою кожної окремої ознаки введення. Згортки дозволяють зробити це перетворення лише за 9 параметрами, з кожною ознакою виходу, а, отже, замість того, щоб «дивитися» на кожну ознаку входу, достатньо тільки «поглянути» на ознаки виходу, що знаходяться приблизно в тому самому місці [17].

Пікселі по краях ніколи не розташовуються в центрі ядра, оскільки ядро не виходить за границі. Це не завжди зручно тому, що часто потрібно, щоб розмір вихідного сигналу був еквівалентний вхідному.

Падинг виконує кошовну функцію, а саме додає до країв додаткові «фальшиві» пікселі. Таким чином, ядро при переміщенні може дозволити початковим пікселям краю знаходитись у центрі, одночасно поширюючись на фальшиві пікселі за границями, що дозволяє виводити вихідний сигнал такого ж розміру, як і вхідний.

Страйдинг

Часто при запуску згорткового шару необхідно отримати вихід з меншим розміром, ніж вхід. Це звичайна ситуація у згорткових нейронних мережах, де розмір просторових розмірностей зменшується при збільшенні кількості каналів. Одним із способів досягнення цього є використання шару агрегування, ще одним – страйдинг (див. рис. 2.8).

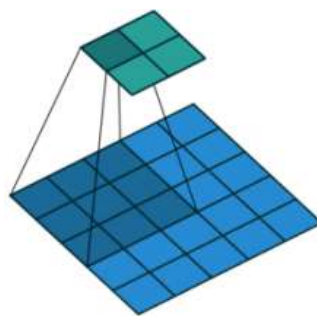


Рисунок 2.8 – Використання страйдингу у згортковій мережі

Ідея страйду полягає в тому, щоб пропустити деякі розташування ядра під час згортки. Страйд розміром 1 означає, що нове розташування ядра обирається із кроком в один піксель, тобто, звичайна робота згорткового шару. При страйді розміром 2 – нове розташування ядра обирається із кроком в два пікселі, пропускаючи всі інші розташування в процесі згортки, зменшуючи їх приблизно вдвічі. Відповідно, страйд розміром 3 передбачає пропуск кожного третього розташування.

Сьогодні для зменшення розмірів внутрішніх шарів, більш сучасні мережі, такі як архітектура ResNet, повністю відмовляються від шарів агрегування на користь згорткових шарів з використанням страйду.

Багатоканальна версія згортки

Більшість зображень на вході мають 3 канали, та з просуванням вглиб мережі це число тільки збільшується. Зручно розглядати канали, як альтернативну версію певного зображення, яка підкреслює деякі аспекти цього зображення та уникає інших. На рис. 2.9 зображено три варіанти одного й того самого зображення, розкладеного на одноканальні зображення по каналу кольору.

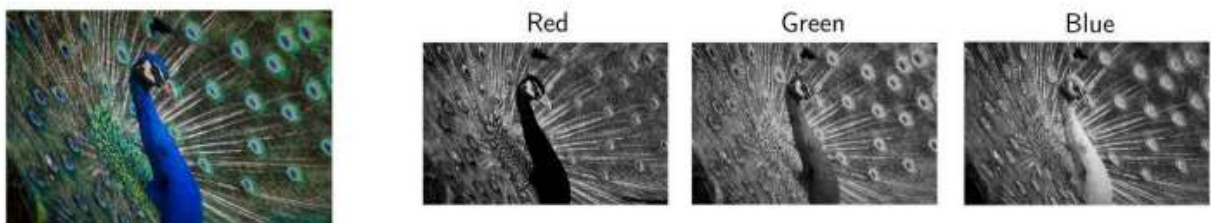


Рисунок 2.9 – Розкладання зображення на альтернативні одноканальні зображення

Більшість часу нейронні мережі мають справу із зображеннями RGB з трьома каналами кольорів. Важливо врахувати при цьому, що у випадку з одним каналом, термін фільтр та ядро є взаємозамінними, тоді, як насправді, вони є досить різними категоріями. Фактично, кожен фільтр – це сукупність

ядер, де для кожного вхідного каналу шару існує одне унікальне ядро. Один фільтр у згортковому шарі обробляє один і тільки один вхідний канал. При цьому певне ядро фільтра «ковзає» над відповідними вхідними каналами, створюючи їх оброблену версію. Деякі ядра можуть мати більшу вагу, ніж інші, що дозволяє їм надавати перевагу певним каналам вводу (наприклад, фільтр може мати ядро червоного каналу з більшою вагою, отже, більше відповідати відмінностям в ознаках червоного каналу).

Потім кожна оброблена версія одноканального вхідного сигналу сумується з іншими для формування одного спільного каналу. Ядра кожного фільтра створюють одну версію кожного каналу, а фільтр, у цілому, виробляє один загальний вихідний канал.

Такі перетворення використовуються для будь-якої кількості фільтрів, кожен з яких за допомогою описаного вище процесу обробляє вхід з власним набором ядер і скалярним зміщенням, створюючи власний вихідний канал. Потім вони об'єднуються всі разом, щоб отримати загальний результат, причому кількість вихідних каналів – це кількість фільтрів, що були використані. Зазвичай, застосовується нелінійність, перш ніж передавати це як вхід в інший шар згортки, який потім повторює цей процес.

Згортки залишаються лінійними перетвореннями

Розглянемо такий випадок: нехай існує вхідний сигнал 4×4 , і необхідно перетворити його на сітку 2×2 . Якщо використати мережу прямого поширення, вхідний сигнал 4×4 трансформується у вектор довжиною 16, що пройде через повнозв'язний шар із вхідними даними довжиною 16 та вихідним довжиною 4. Можна візуалізувати матрицю ваг W для шару (рис. 2.10)

$$\begin{bmatrix} w_{1,1} & w_{1,2} & w_{1,3} & w_{1,4} & w_{1,5} & w_{1,6} & w_{1,7} & w_{1,8} & w_{1,9} & w_{1,10} & w_{1,11} & w_{1,12} & w_{1,13} & w_{1,14} & w_{1,15} & w_{1,16} \\ w_{2,1} & w_{2,2} & w_{2,3} & w_{2,4} & w_{2,5} & w_{2,6} & w_{2,7} & w_{2,8} & w_{2,9} & w_{2,10} & w_{2,11} & w_{2,12} & w_{2,13} & w_{2,14} & w_{2,15} & w_{2,16} \\ w_{3,1} & w_{3,2} & w_{3,3} & w_{3,4} & w_{3,5} & w_{3,6} & w_{3,7} & w_{3,8} & w_{3,9} & w_{3,10} & w_{3,11} & w_{3,12} & w_{3,13} & w_{3,14} & w_{3,15} & w_{3,16} \\ w_{4,1} & w_{4,2} & w_{4,3} & w_{4,4} & w_{4,5} & w_{4,6} & w_{4,7} & w_{4,8} & w_{4,9} & w_{4,10} & w_{4,11} & w_{4,12} & w_{4,13} & w_{4,14} & w_{4,15} & w_{4,16} \end{bmatrix}$$

Рисунок 2.10 – Матричне представлення ваг повнозв'язного шару

Хоча операція з ядром згортки може на перший погляд виглядати дещо дивно, вона все ще є лінійним перетворенням з еквівалентною матрицею. Якщо необхідно застосувати ядро K розміром 3 на зміненому входному сигналі 4×4 для отримання вихідного сигналу 2×2 , еквівалентна матриця перетворення буде виглядати [18]:

$$\begin{bmatrix} k_{1,1} & k_{1,2} & k_{1,3} & 0 & k_{2,1} & k_{2,2} & k_{2,3} & 0 & k_{3,1} & k_{3,2} & k_{3,3} & 0 & 0 & 0 & 0 & 0 \\ 0 & k_{1,1} & k_{1,2} & k_{1,3} & 0 & k_{2,1} & k_{2,2} & k_{2,3} & 0 & k_{3,1} & k_{3,2} & k_{3,3} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & k_{1,1} & k_{1,2} & k_{1,3} & 0 & k_{2,1} & k_{2,2} & k_{2,3} & 0 & k_{3,1} & k_{3,2} & k_{3,3} & 0 \\ 0 & 0 & 0 & 0 & 0 & k_{1,1} & k_{1,2} & k_{1,3} & 0 & k_{2,1} & k_{2,2} & k_{2,3} & 0 & k_{3,1} & k_{3,2} & k_{3,3} \end{bmatrix}$$

Рисунок 2.11 – Матричне представлення ваг згорткового шару

З матриці видно, що використовуються лише 9 параметрів. Згортка, як і раніше, є лінійним перетворенням, але, в той самий час, це також різний вид перетворення. Для матриці з 64 елементами існує всього 9 параметрів, які повторно використовуються кілька разів. Кожен вихідний вузол бачить лише вибране число входів (ті, що знаходяться всередині 3×3 ядра), і немає взаємодії з будь-якими іншими входами тому, що ваги для них встановлено на 0.

Функція активації Softmax

Функція Softmax приймає N -мірний вектор дійсних чисел і перетворює його в вектор значень в діапазоні $(0,1)$, який визначається за формулою:

$$p_i = \frac{e^{a_i}}{\sum_{k=1}^N e^{a_k}}, \quad (2.11)$$

Як видно з назви, функція softmax є спрощеною версією функції max. Замість того, щоб вибирати одне максимальне значення, функція пропорційно розподіляє значення, щоб в сумі отримати одиницю.

Ця властивість функції softmax, яка виводить розподіл імовірності, робить її придатною для імовірнісного тлумачення у завданнях класифікації.

Щоб зробити функцію softmax чисельно стабільною, необхідно нормалізувати значення вектора шляхом множення чисельника і знаменника на константу C .

$$\begin{aligned}
 p_i &= \frac{e^{a_i}}{\sum_{k=1}^N e^{a_k}} \\
 &= \frac{C e^{a_i}}{C \sum_{k=1}^N e^{a_k}} \\
 &= \frac{e^{a_i + \log(C)}}{\sum_{k=1}^N e^{a_k + \log(C)}} ,
 \end{aligned} \tag{2.12}$$

Можна обрати довільне значення для $\log(C)$, хоча, зазвичай, обирається $\log(C) = -\max(a)$, оскільки, він зміщує всі елементи вектору в діапазон від від'ємних значень до нуля.

Через бажану властивість функції softmax, що визначає розподіл імовірності, дана функція використовується як останній шар в нейронних мережах. Для цього потрібно обчислити похідну або градієнт і повернути його назад до попереднього шару під час зворотного розповсюдження похибки.

$$\frac{\partial p_i}{\partial a_j} = \frac{\partial \frac{e^{a_i}}{\sum_{k=1}^N e^{a_k}}}{\partial a_j} , \tag{2.13}$$

З загального виду градієнту можна зробити висновок, що похідну функції Softmax дуже просто обчислити, тому її використання є виправданим з огляду на швидкість обчислень.

Архітектура мережі

Архітектура згорткових мереж складається з різних шарів, що виконують перетворення вхідного об'єму даних. Кожен шар згорткової мережі складається з нейронів, з'єднаних з вузлами попередніх шарів, таким чином,

вихід у даному вузлі для шару L є вихідною функцією вузлів у шарі $L-1$. На прикладі п'яти основних шарів даної мережі наведемо архітектуру, що пропонується для розгляду[19]:

1. Згортковий шар. Його параметри містять набори фільтрів (або ядер) для навчання мережі, які мають невелике рецептивне поле, що простягаються на всю глибину вхідного об'єму. Кожен фільтр здійснює згортку по ширині та висоті вхідного об'єму під час прямого проходу. Виконується обчислення скалярного добутку даних фільтру та входу, і формується двовимірний карта активації цього фільтру. Перші згорткові шари вивчають низькорівневі особливості, такі як краї, лінії та кути. Наступні шари вивчають більш складні особливості (наприклад, частини та моделі).

2. Шари підвибірки. Вони реалізовані нелінійними функціями для реалізації підвибірки, серед яких найпоширенішою є максимізаційна підвибірка. Вона розділяє вхідне зображення на набір прямокутників без перекриття, і для кожної такої підобласті виводить її максимум. Ідея створення цього шару полягає в тому, що якщо ознаку було знайдено, то її точне положення не так важливе, як її положення відносно інших ознак. Функцією підвибіркового шару є поступове скорочення просторового розміру представлення для зменшення об'єму параметрів (наприклад вага нейрону) та обчислень у мережі.

3. Шари активації - функції активації імітують поведінку аксона нейрона, що запускає сигнал при контакті з подразником. Ключовою особливістю згорткових мереж є функція активації ReLU.

4. Шари виключення, які генерують випадковий набір активацій, обнуляючи їх значення. Це означає, що мережа повинна надавати правильну класифікацію, навіть якщо деякий набір активацій виключено.

5. Повнозв'язні шари слугують як провідники між шарами, де всі виходи попереднього шару пов'язані з усіма входами повнозв'язних шарів.

На рисунку 2.12 наведено варіант архітектури, що спроектована на основі SSD (англ. single-shot detector) архітектури.

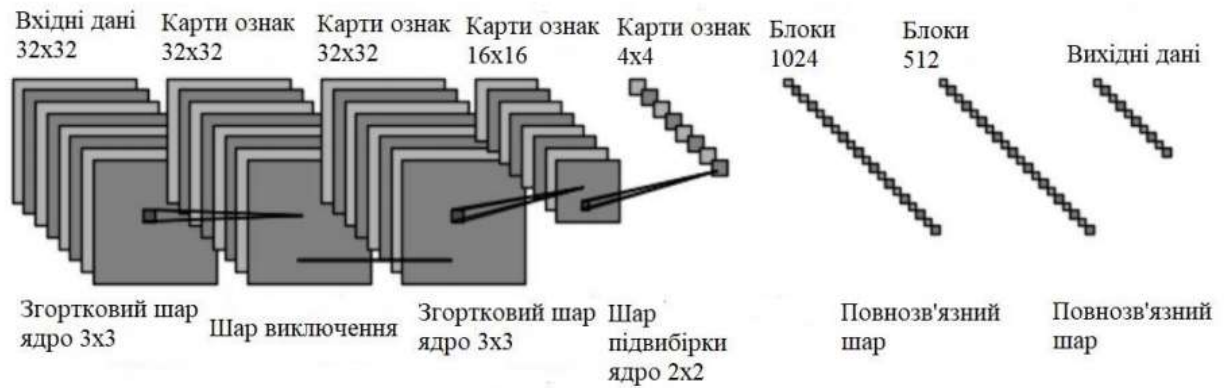


Рисунок 2.12 – Архітектура запропонованої згорткової нейронної мережі, що спроектована на основі SSD архітектури.

Отже в даному розділі було розглянуто основні елементи згорткової мережі та особливості вибору параметрів для навчання нейронних мереж. Обґрунтовано використання функції активації Softmax, за допомогою якої відбувається визначення ймовірності того, до якого класу належить об'єкт, а також взято за основу архітектуру мережі, яка спроектована на базі SSD архітектури.

2.4 Вимоги до вхідного сигналу згорткової нейронної мережі

Відносно поставленої задачі нейронній мережі на вхід слід отримувати відео-потік, мінімальне розширення якого має відповідати хоча б 640x480 пікселів, для коректного аналізу. Далі мережа покадрово аналізує відео, тобто отримує зображення в певний момент часу. Кожне зображення буде оброблятися за трьома каналами кольорів, тобто за RGB колірною моделлю. Якщо зображення побудовано на основі іншої моделі, його слід конвертувати із допомогою вбудованих функцій в RGB. Після виконання всіх підготовчих

операцій, мережа розпочинає обробку каналів. Потім кожна оброблена версія одноканального вхідного сигналу сумується з іншими для формування одного спільного каналу.

2.5 Архітектура інформаційної технології керування WEB-додатком на основі згорткових нейронних мереж

На поточному етапі проектування розроблено алгоритм функціонування ядра інформаційної системи, але іншим завданням даної роботи є розробка модуля спроможного до масштабування. Тобто необхідно інформаційну систему розробити так, щоб вона легко інтегрувалась, як у новий проект так і уже готовий. Задля цього розроблювана система повинна мати можливість підключення у вигляді модуля.

Отже необхідно спроектувати систему так, щоб мати можливість її використовувати у вигляді пакету або бібліотеки. Необхідно уже розроблене ядро інформаційної технології помістити в пакет, та надати відповідний інтерфейс по взаємодії з ним.

Запропоновану реалізацію можна представити у вигляді UML-діаграми класів. Схема складається з декількох секцій. Безпосередньо ядро розпізнавання жестів, яке вхідними параметрами отримує відео потік з цільової системи інтеграції та декілька систем, які використовують функціонал ядра розпізнавання жестів. Цільова система «підписується» на той список жестів, який необхідний у її функціоналі.

На рис. 2.13. показано як дві системи, а саме відео-додаток і музичний плеєр використовують функціонал розпізнавання з одного пакета.

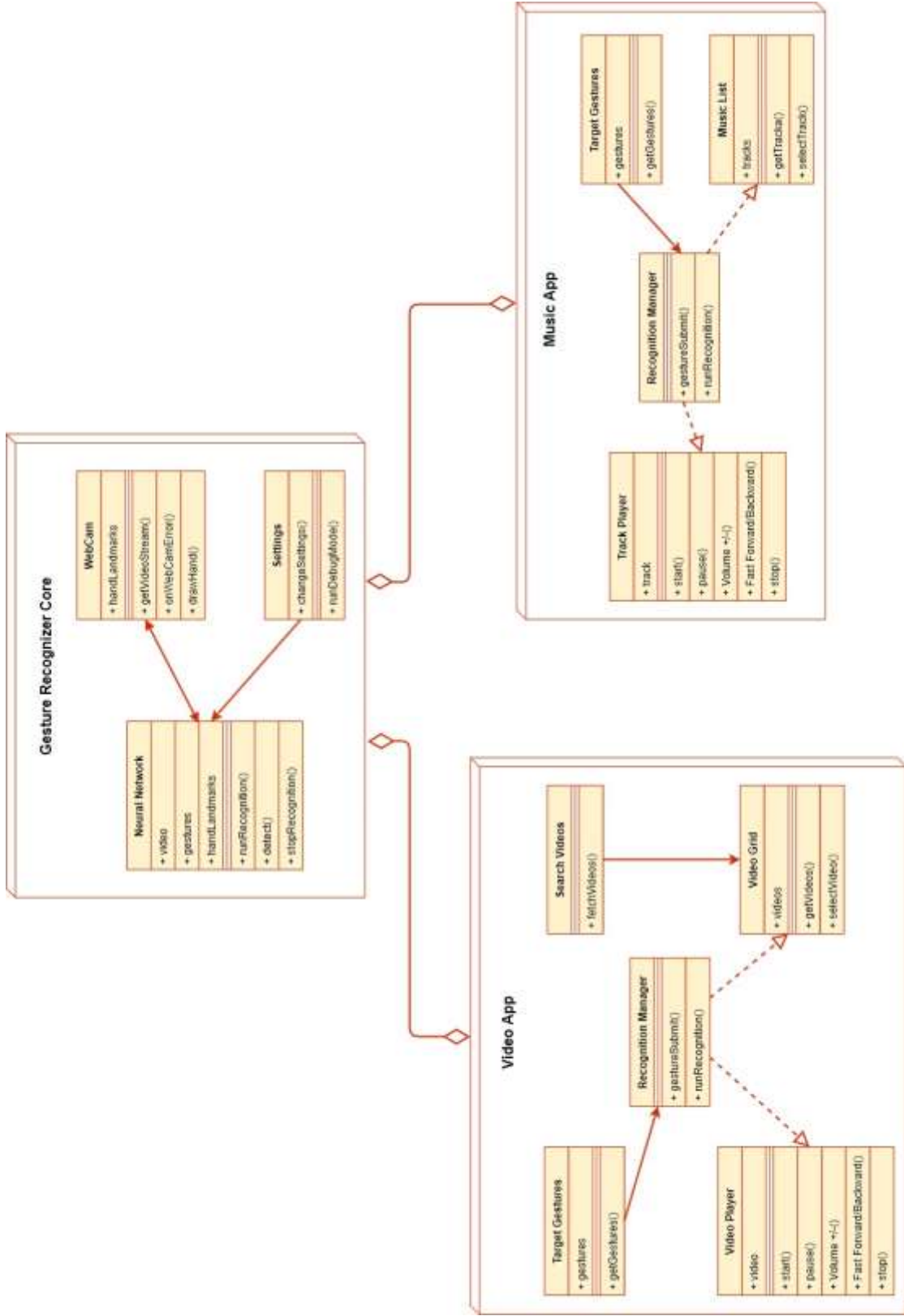


Рисунок 2.13 – UML-діаграма класів інформаційної технології жестового керування WEB-додатком на основі згорткових нейронних мереж

2.6 Розробка алгоритму функціонування інформаційної технології жестового керування WEB-додатком

Вхідними даними системи приймає відео-потік з веб-камери та список цільових жестів, а вихідними – опис позиції руки в просторі та список впевненості у жестах.

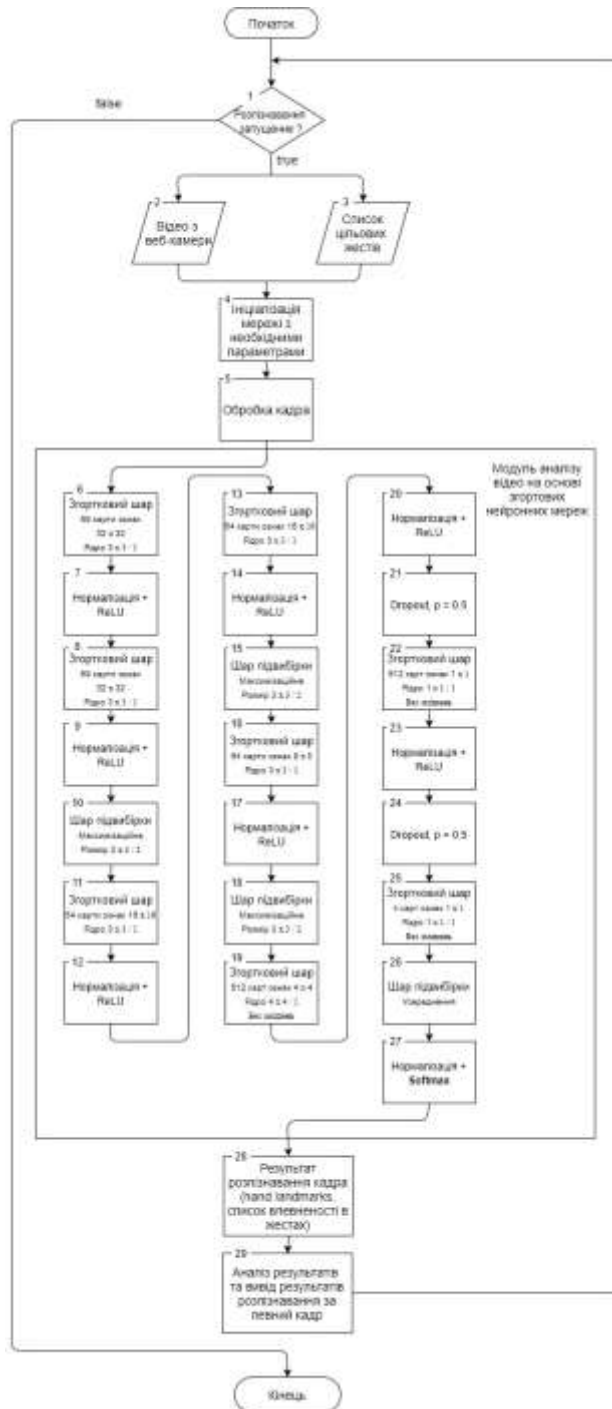


Рисунок 2.14 – Загальний алгоритм роботи ядра інформаційної технології

Алгоритм роботи проектованої інформаційної системи повинен забезпечувати таку послідовність дій:

1. Запуск режиму розпізнавання.
2. Ініціалізація мережі з поданими на вхід параметрами, а саме відео та список цільових жестів.
4. Покадровий аналіз відео модулем на основі згорткових нейронних мереж, процес детально зображений на рис. 2.14
5. Проаналізувати результати розпізнавання мережі.
6. Вивести результати користувачу у вигляді підкреслення країв руки на відео та виконати відповідну операцію у випадку вдалого знаходження жесту зі списку цільових.

2.7 Висновок до розділу 2

Отже у даному розділі було розглянуто декілька видів нейронних мереж і обрано для програмної реалізації згорткову нейронну мережу, так як вона має найточніші механізми аналізу зображення, які допоможуть вирішити поставлену задачу. Також було проаналізовано структуру, математичну модель та принцип функціонування згорткової нейронної мережі. Було сформовано вимоги до вхідного сигналу нейронної мережі. Розроблено алгоритм функціонування інформаційної технології, а також запропоновано архітектуру, яка представляє собою ядро системи обгорнуте у вигляді пакету, який у свою чергу надає інтерфейс задля зручної інтеграції інформаційної технології у будь-які інші системи.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ЖЕСТОВОГО КЕРУВАННЯ WEB-ДОДАТКОМ НА ОСНОВІ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ

3.1 Обґрунтування вибору мови та середовища програмування для інформаційної технології жестового керування WEB-додатком

Визначимо інструменти, за допомогою яких буде створюватись програмна реалізація інтелектуальної системи жестового керування браузером на основі згорткових нейронних мереж. Зокрема, необхідно визначити мову програмування для створення веб-клієнта, панелі адміністрування, програмного інтерфейсу, універсального кодувальника речень, а також системи керування базами даних для збереження даних.

Для аналізу і порівняння було обрано 3 мови програмування: Python, C++, JavaScript.

Python – мова програмування, що швидко розвивається. Мова має простий і зрозумілий синтаксис. Це значно спрощує процес розробки. Має динамічну типізацію та автоматичний контроль пам'яті та механізм обробки виключень. Мова підтримується великою кількістю бібліотек, як, наприклад, Django. Найчастіше використовується для створення веб-додатків. Використання бібліотек і низький поріг входження зробили Python однією з найбільш популярних мов програмування [20].

C++ – мова програмування високого рівня з підтримкою кількох парадигм програмування: узагальненої та процедурної, об'єктно-орієнтованої. Мову використовують для системного програмування, розробки програмного забезпечення, написання драйверів, потужних серверних та клієнтських програм, а також для розробки розважальних програм, наприклад, відеоігор. Мова C++ з появою перших трансляторів знайшла відразу ж дуже широке розповсюдження, на ній було створено величезну кількість програм і

застосунків. У міру накопичення досвіду створення великих програмних систем спливали недоліки, які спонукали до пошуку альтернативних рішень [21].

JavaScript – динамічна, об’єктно-орієнтована мова програмування. Найчастіше використовується для створення сценаріїв веб-сторінок, що надає можливість на стороні клієнта взаємодіяти з користувачем, керувати браузером, обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд веб-сторінки [22].

Гнучкість цієї мови дозволяє використовувати безліч шаблонів програмування стосовно до конкретних умов. У JavaScript взагалі відсутня статична типізація, тобто в момент оголошення не вказується тип змінної, він присвоюється лише разом із присвоєнням або зміною значення, що полегшує роботу з несумісністю типів. Прийняття нових стандартів призвело до спрощення синтаксису JavaScript і додало класи, що робить JavaScript повноцінною об’єктно-орієнтованою мовою. У таблиці 3.1 наведена порівняльна характеристика мов програмування[23].

У таблиці 3.1 наведена порівняльна характеристика мов програмування.

Таблиця 3.1 – Порівняльна характеристика мов програмування

Мова/ Критерій	Динамічна типізація	Статична типізація	Виконання на клієнт- частині	Пряме підключенн я скриптів	Необхідніс ть у роботі з браузером
Python	+	-	-	-	-
C++	+	+	-	-	-
JavaScript	+	-	+	+	+

Також для реалізації поставленої задачі було обрано середовище розробки WebStorm.

JetBrains Webstorm – інтегроване середовище розробки для JavaScript, HTML та CSS від компанії JetBrains, розроблена на основі платформи IntelliJ IDEA. WebStorm є спеціалізованою версією PhpStorm, пропонуючи підмножину з його можливостей. WebStorm постачається з перед-установленим плагінами JavaScript. WebStorm забезпечує авто доповнення, аналіз коду на льоту, навігацію по коду та інтеграцію з системами управління версіями. Важливою перевагою інтегрованого середовища розробки WebStorm є робота з проектами. Перевагами є вбудований Debugger, яким з легкістю можна перевірити коректність написаного коду, також можна встановити точки зупинки. Підтримує Node.js також є простий уніфікований інтерфейс для роботи з Git, GitHub. Зручний, приємний для роботи інтерфейс. Webstorm пропонує велику кількість готових шаблонів веб-додатків, що спрощує первинне налаштування системи. Крім того, середа розробки підтримує популярні препроцесори CSS, такі як LESS, SASS, SCSS. Є можливість встановлення плагіна Emmet, який значно пришвидшує розробку[24]

Єдиним недоліком WebStorm є платна основа розповсюдження. Однак компанія підтримує програму безкоштовного ліцензування студентів. Таким чином, WebStorm є найкращим середовищем розробки на JavaScript.

Отже для створення програмного забезпечення було обрано мову програмування JavaScript, тому що вона є незамінною при роботі на клієнтській частині браузера. Простий для розуміння синтаксис з можливостями динамічних змін перетворень типів, прибирання сміття, оголошення анонімних функцій таким чином дипломна робота буде написана з використанням однієї мови програмування.

3.2 Реалізація інформаційної технології жестового керування WEB-додатком з використанням бібліотек React.js та Tensorflow.js

Інтерфейс користувача буде реалізований бібліотекою React.js. Для створення цього проекту необхідно запустити середовище для веб-розробки WebStorm. Далі створити новий проект. З'явиться вікно представлене на рис. 3.1

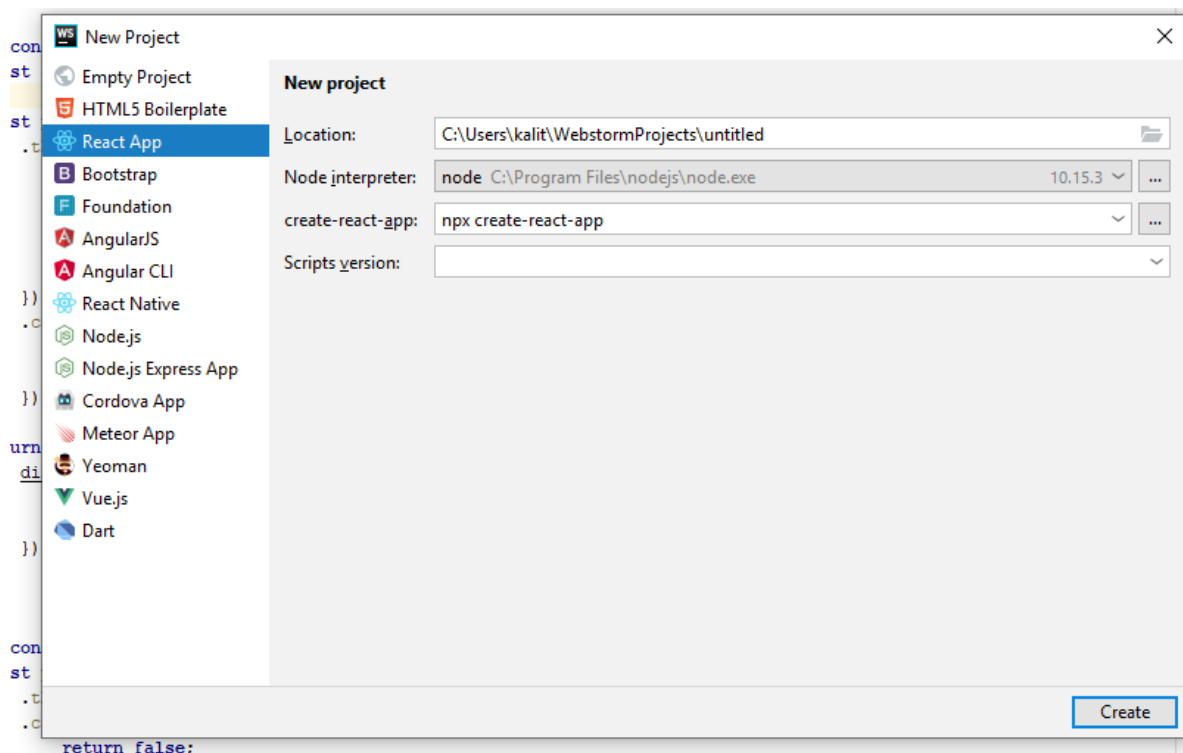


Рисунок 3.1 – Створення нового проекту в WebStorm

У цьому вікні потрібно обрати проект React App. В полі location необхідно ввести шлях до директорії, після чого натиснути Create.

Після цього створиться проект і всі необхідні файли для початку роботи. Це початковий клас App унаслідований від класу React бібліотеки React.js. Далі можна редагувати даний клас або створювати інші.

У разі необхідності додаткових пакетів ми може інстальювати їх за допомогою менеджера пакетів(npm), виконавши команду `npm install <назва пакету>` [25].

Далі всі необхідні компоненти ми зможем імпортувати за допомогою команди `import` вказавши назву та шлях, звідки імпортується, як представлено на рис. 3.2 [26].

```

App.js
1  import React, { useRef, PureComponent } from "react";
2  import { isEqual, isEmpty, isNumber } from "lodash";
3  import classNames from "classnames";
4

```

Рисунок 3.2 – Приклад імпортування компонентів

За основу було прийнято рішення взяти макет відомої відео платформи Youtube, вигляд головної сторінки продемонстровано на рис. 3.3. Функціонал даного сервісу повністю підходить, щоб продемонструвати елементарні функції, які можуть бути адаптовані у вигляді жестів.

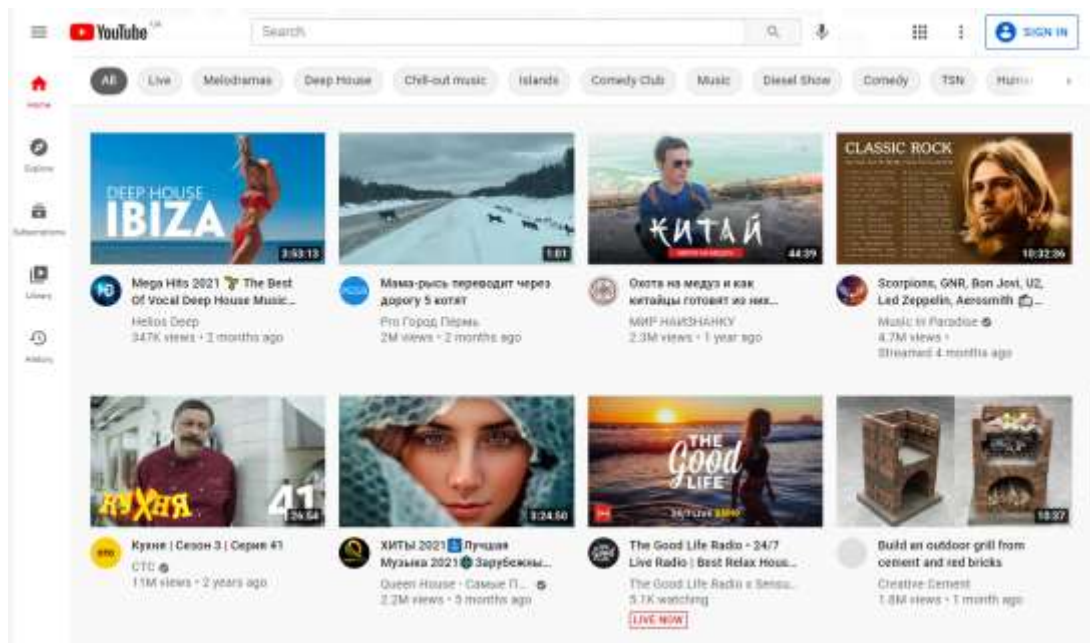


Рисунок 3.3 – Вигляд головної сторінки відео-платформи Youtube

Отже необхідно реалізувати наступний список функцій, які зможуть виконуватись за допомогою жестів:

- вибір наступного/попереднього відео у списку;
- функція підтвердження дії, еквівалентна клавіші Enter;
- старт/пауза відео плеєра;
- додати/знизити рівень гучності відтворення відео;
- перемотка відео вперед/назад;
- функція зупинки/виходу, еквівалентна клавіші Esc.

3.3 Специфіка програмної реалізація інформаційної технології жестового керування WEB-додатком

Для реалізації модуля розпізнавання жестів, було взято за основу бібліотеку Tensorflow.js, а саме одну з її моделей «hand-pose-detection». Також було використано додатковий класифікатор «fingerpose», для зручності обробки результату розпізнавання.

Дана мережа приймає на вхід відео-потік. Відео-потік можна отримати вбудованими засобами браузера, головне щоб веб-камера була підключена та користувач дозволив сайту використовувати її.

Також другим параметром мережа приймає список цільових жестів. Кожен жест необхідно описати, а саме вказати всі можливі позиції пальців руки, які цікавлять. Палець описується за допомогою двох властивостей. Перша це напрям, тобто вгору, вниз, по діагоналі доверху-вліво, по діагоналі донизу-вправо. Друга властивість це ступінь зігнутості пальця, тобто він може бути прямим, напівзігнутим або повністю зігнутим. До кожної властивості додається певний коефіцієнт важливості даної позиції пальця, тобто якщо певна позиція пальця важлива в загальному для жесту, тоді потрібно присвоїти більший ваговий коефіцієнт і якщо певна позиція не корисна для описуваного жесту, тоді навпаки потрібно присвоїти нижчий, або навіть від'ємний ваговий

коефіцієнт. Опис жесту «вправо», який відповідає за вибір наступного відео, продемонстровано на рис. 3.4.

```

20  const rightGesture = new GestureDescription('right');
21
22  // Thumb
23  rightGesture.addCurl(Thumb, NoCurl, confidence: 1);
24  rightGesture.addCurl(Thumb, HalfCurl, confidence: -5);
25  rightGesture.addCurl(Thumb, FullCurl, confidence: -5);
26  rightGesture.addDirection(Thumb, DiagonalUpLeft, confidence: 0.7);
27  rightGesture.addDirection(Thumb, HorizontalLeft, confidence: 1);
28  rightGesture.addDirection(Thumb, DiagonalDownLeft, confidence: 0.7);
29
30  rightGesture.addDirection(Thumb, VerticalUp, confidence: -5);
31  rightGesture.addDirection(Thumb, VerticalDown, confidence: -5);
32  rightGesture.addDirection(Thumb, DiagonalUpRight, confidence: -5);
33  rightGesture.addDirection(Thumb, HorizontalRight, confidence: -5);
34  rightGesture.addDirection(Thumb, DiagonalDownRight, confidence: -5);
35
36  // Rest
37  for(let finger of [Index, Middle, Ring, Pinky]){
38    rightGesture.addCurl(finger, FullCurl, confidence: 1);
39    rightGesture.addCurl(finger, NoCurl, confidence: -3);
40    rightGesture.addCurl(finger, HalfCurl, confidence: -2);
41    rightGesture.addDirection(finger, VerticalUp, confidence: 1);
42    rightGesture.addDirection(finger, DiagonalUpLeft, confidence: 0.9);
43    rightGesture.addDirection(finger, DiagonalUpRight, confidence: 0.9);
44  }
45
46  export default rightGesture

```

Рисунок 3.4 – Приклад опису жесту «Вправо»

Отже описавши всі необхідні жести та отримавши відео-потік, можна викликати метод розпізнавання з цими параметрами, як продемонстровано на рис. 3.5.

```

315
316
317 // Make Detections
318 const hand = await ml5.vision.hands(video);
319
320 if (hand.length > 0) {
321   const SE = new fp.GestureEstimator([
322     highFiveGesture,
323     fistGesture,
324     okGesture,
325     rightGesture,
326     leftGesture,
327     oneFingerUp,
328     oneFingerDown,
329     twoFingerUp,
330     twoFingerDown,
331   ]);
332   const result = await SE.estimate(hand[0].landmarks, debugMode ? debugConfidence : confidence);
333

```

Рисунок 3.5 – Приклад виклику методу розпізнавання з параметрами відео та списком цільових жестів

Так як система вмiє обробляти декiлька рук одночасно, необхідно вказати чiткi правила, за якими програма буде виконувати одну конкретну дiю коли було розпiзнано бiльше нiж один жест.

Це реалiзується досить просто, а саме вибравши зi списку розпiзнаних жестiв, жест у якого коефiцiєнт впевненостi серед iнших найвищий. Приклад програмної реалiзацiї цього процесу продемонстровано на рис. 3.6

```

if (
  this.camera?.video?.readyState === 4
) {
  const video = this.camera.video;
  const { confidence } = this.recognitionSettings;
  const originalEstimation = await this.detector.estimateHands(video);
  const adoptedEstimation = originalEstimation.map((item) => ({
    landmarks: item.keypoints?.map(({x, y}) => ([x, y, 0]))
  }));

  if (adoptedEstimation.length > 0) {
    const temp = adoptedEstimation.map((hand) => {
      if (_.isEmpty(hand?.landmarks)) {
        return { confidence: null, gesture: null };
      }

      const result = this.gestureEstimator.estimate(hand.landmarks, confidence);

      if (_.isEmpty(result?.gestures)) {
        return { confidence: null, gesture: null };
      }

      const confidenceList = result.gestures.map(prediction => prediction.score);

      const maxConfidence = Math.max(...confidenceList);
      const maxConfidenceGestureIndex = confidenceList.indexOf(maxConfidence);
      const gesture = result.gestures[maxConfidenceGestureIndex].name;

      return { confidence: maxConfidence, gesture };
    });

    const newGesture = _.maxBy(temp, iteratee: 'confidence')?.gesture;

    this.confirmGesture( gesture: newGesture || null);
  } else {
    this.confirmGesture( gesture: null);
  }

  if (this.observingRecognition) {
    this.camera.setHands(originalEstimation);
  }
}

```

Рисунок 3.6 – Приклад фрагменту коду, який приймає рiшення щодо вибору одного жесту серед списку розпiзнаних

Додаткова впевненість, крім результату роботи мережі, у коректності розпізнаного жесту, у даній реалізації досягається наступним чином. Після одержання результату розпізнавання мережі, щоб система була більш впевненіша, що користувач показав саме той жест, який планував, а не просто зробив спонтанний змах рукою, було додано певну постійну затримку. Ця затримка очікує певний час, до прикладу дві секунди, на протязі якого передбачається, що користувач не змінить цільовий жест. Якщо жест не змінився, програма може бути більш впевненіша, що користувач демонстрував цей жест певний час, тому що він хотів продемонструвати саме те. Якщо жест був змінений, або взагалі перестав розпізнаватися під час очкування цієї затримки, це дасть програмі зрозуміти, що це був випадковий, не цільовий жест користувача. Час затримки можна динамічно змінювати у налаштуваннях програми під час її роботи, він підлаштовується вручну в залежності від користувацького комфорту. Приклад програмної реалізації цього процесу продемонстровано на рис. 3.7

```
confirmGesture(gesture) {
  const { confirmTime } = this.recognitionSettings;

  if (gesture !== this.tempGesture) {
    clearTimeout(this.confirmTimer);
    this.tempGesture = gesture;

    if (gesture) {
      this.confirmTimer = setTimeout( handler: () => {
        this.handleGestureSubmit(gesture);
        this.tempGesture = null;
      }, confirmTime);
    } else {
      this.confirmTimer = null;
    }
  }
}
```

Рисунок 3.7 – Приклад фрагменту коду, який приймає рішення щодо вибору одного жесту серед списку розпізнаних

Якщо всі дії виконано вірно, запустивши програму можна побачити пряму трансляцію відео з веб-камери. Піднявши руку, система розпізнає її, та підсвітить фаланги пальців, як продемонстровано на рис. 3.8

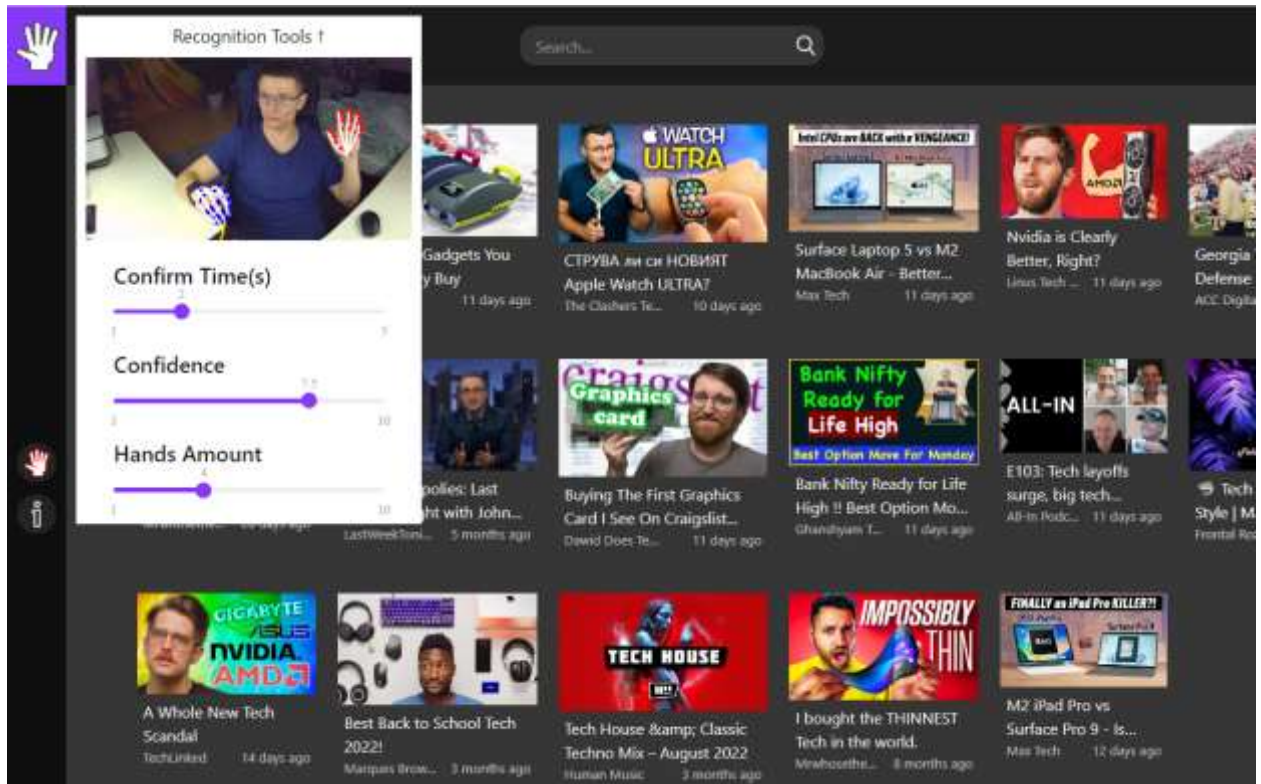


Рисунок 3.8 – Приклад вдало запущеного розпізнавання у альтернативному додатку веб-платформи

Щоб довести модульність розробленої системи розпізнавання було реалізовано інший, схожий веб-додаток, який використовує теж саме ядро розпізнавання. А саме це програма для прослуховування музики, тобто музичний плеєр, яким можна також керувати за допомогою жестів, тим самим змінюючи трек, додати гучність, перемотувати пісню, як продемонстровано на рис. 3.8.

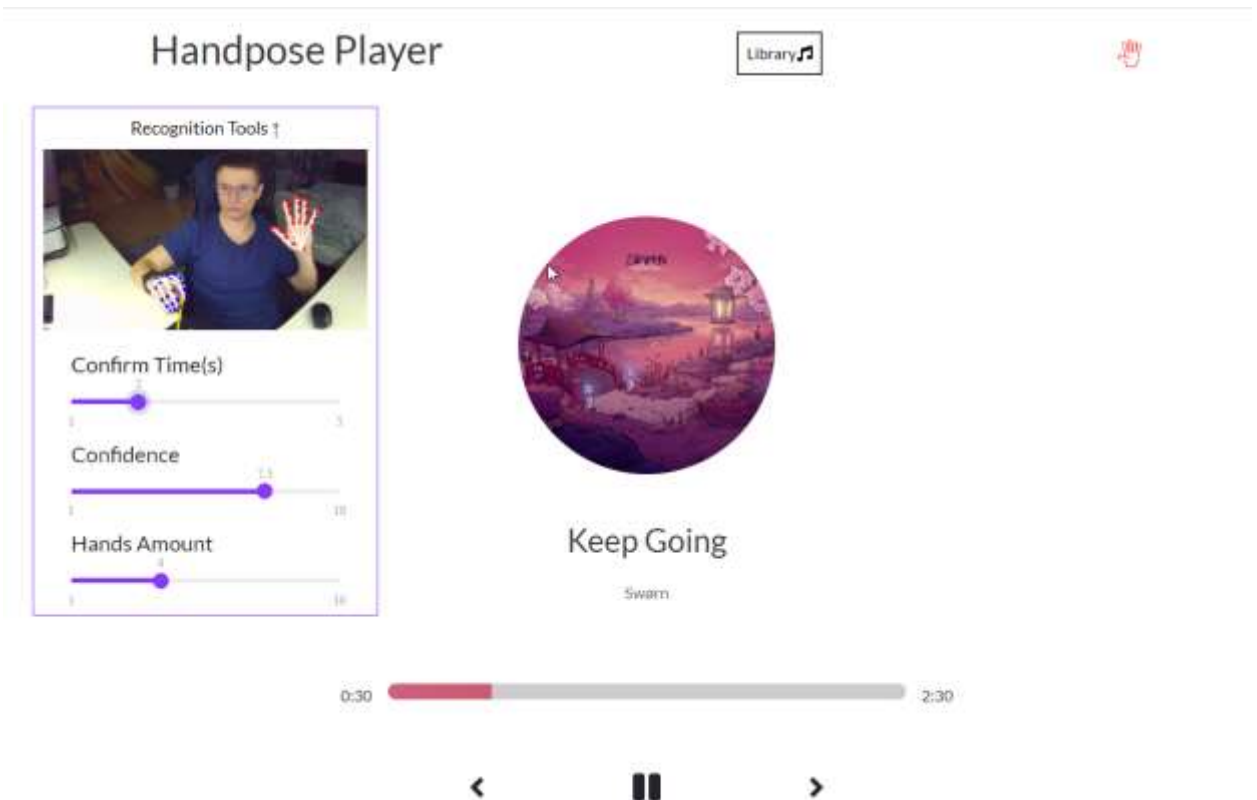


Рисунок 3.8 – Приклад розпізнавання у веб-додатку музичний плеєр

3.4 Висновок до розділу 3

Таким чином у даному розділі було проаналізовано три мови програмування. Для реалізації поставленої задачі було обрано мову програмування JavaScript, так як це єдина мова, яка надає гнучкі можливості реалізації цільового функціоналу у браузері. Також було обґрунтовано вибір середовища розробки WebStorm. На основі підбраної конфігурації технічних інструментів було реалізовано інтелектуальну систему засобами бібліотек React.js та Tensorflow.js.

4 ТЕСТУВАННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ РОБОТИ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ЖЕСТОВОГО КЕРУВАННЯ WEB-ДОДАТКОМ НА ОСНОВІ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ

4.1 Тестування роботи інформаційної технології жестового керування WEB-додатком

Тестування буде проводитися за наступними критеріями:

- веб-клієнт інтелектуальної системи доступний за посиланням в мережі Інтернет;
- при відкритті сторінки відображаються відповідні початкові відео;
- режим розпізнавання вмикається із допомогою відповідної кнопки;
- демонстрований жест виконує відповідно задану йому функцію;

Перейшовши за посиланням <https://handp0se.firebaseio.com> відображається головна сторінка додатку, яка представлена на рис. 4.1.

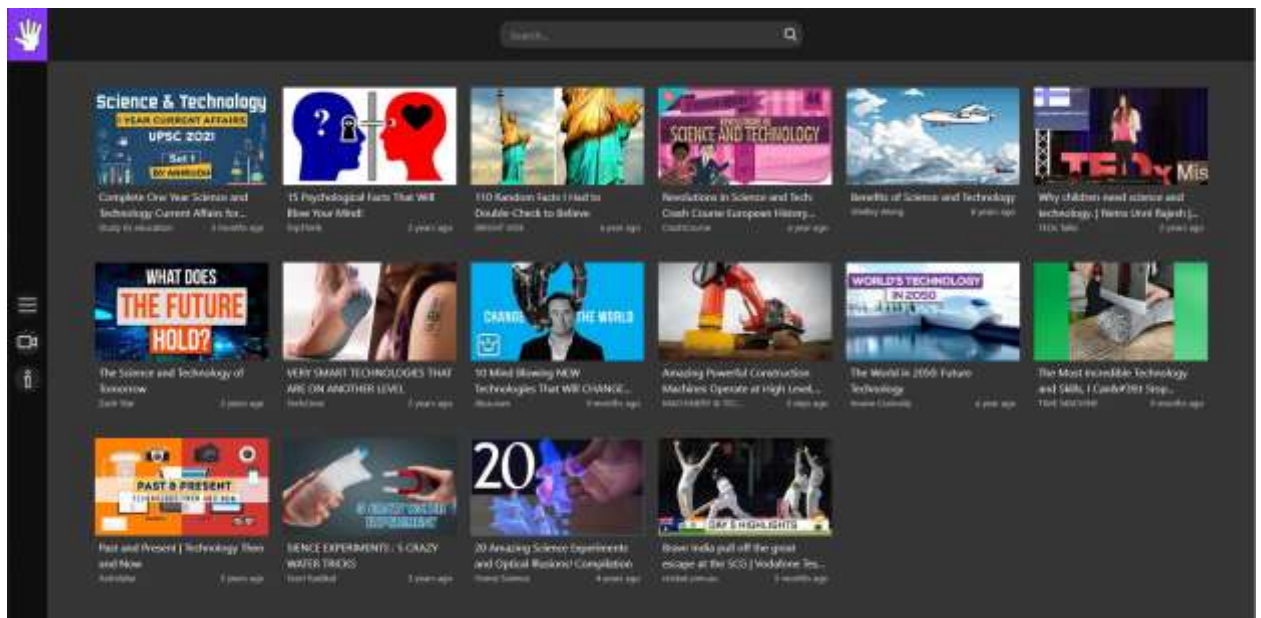


Рисунок 4.1 – Головна сторінка веб-додатку

Натиснувши відповідну кнопку, спостерігаємо увімкнення режиму розпізнавання. Загорається червоним кольором індикатор камери, що символізує про розпочатий процес розпізнавання жестів.

Тестування було розпочато з жесту «вліво». Очікується, що додаток розпочне підсвічувати відео по черзі у списку, як показано на рис. 4.2.

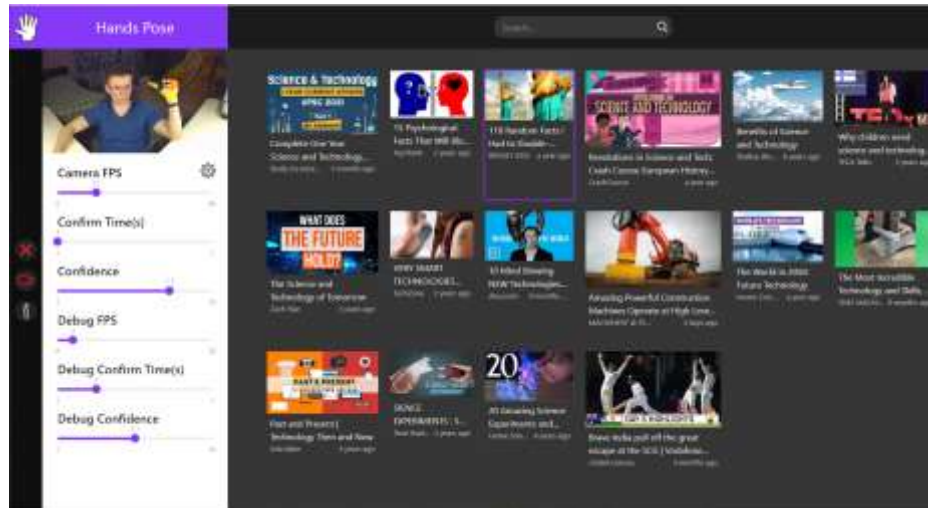


Рисунок 4.2 – Жест «вліво»

Далі перестаємо показувати жест «вліво» та показуємо жест «підтвердження» та очікуємо запуск відео, як продемонстровано на рис. 4.3.

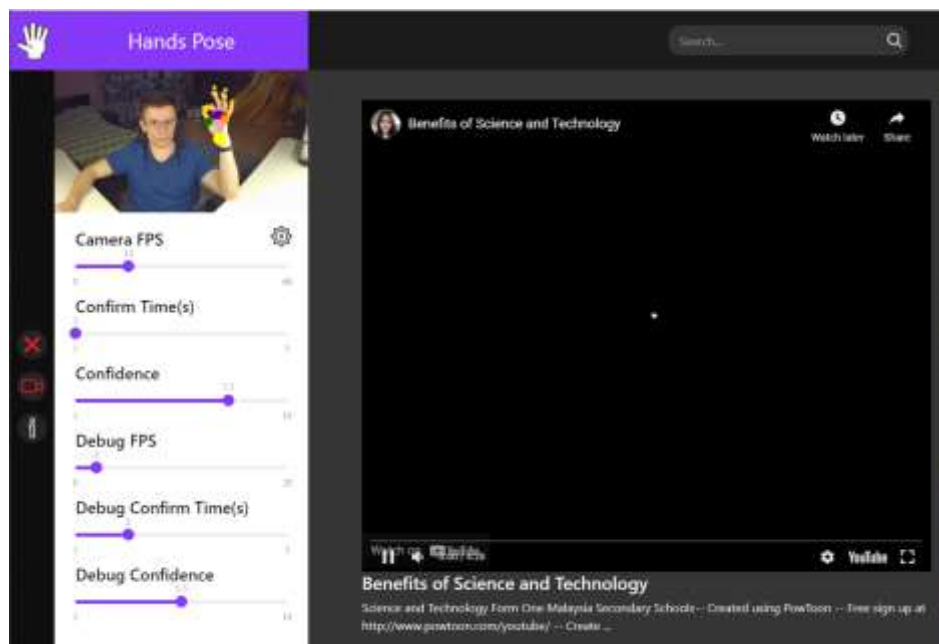


Рисунок 4.3 – Жест «підтвердження»

Наступним кроком можна збільшити рівень звучності відео із допомогою жесту «додати гучності» очікується, що рівень гучності відео буде зростати, як показано на рис. 4.4.

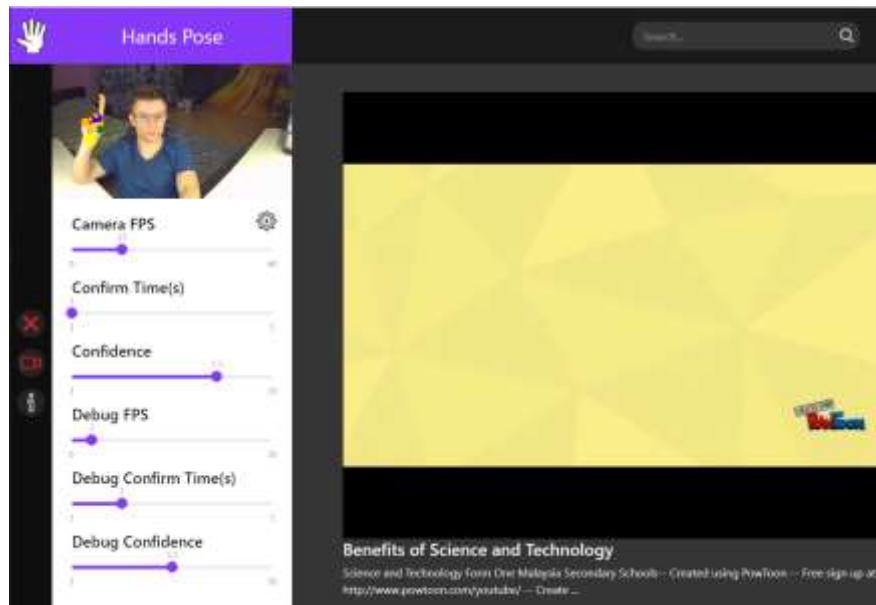


Рисунок 4.4 – Жест «додати гучності»

Далі перевіряємо функцію перемотування відео за допомогою «перемотка відео вперед», передбачається, що відео поступово буде перемотуватися на 10 секунд вперед, як показано на рис. 4.5.

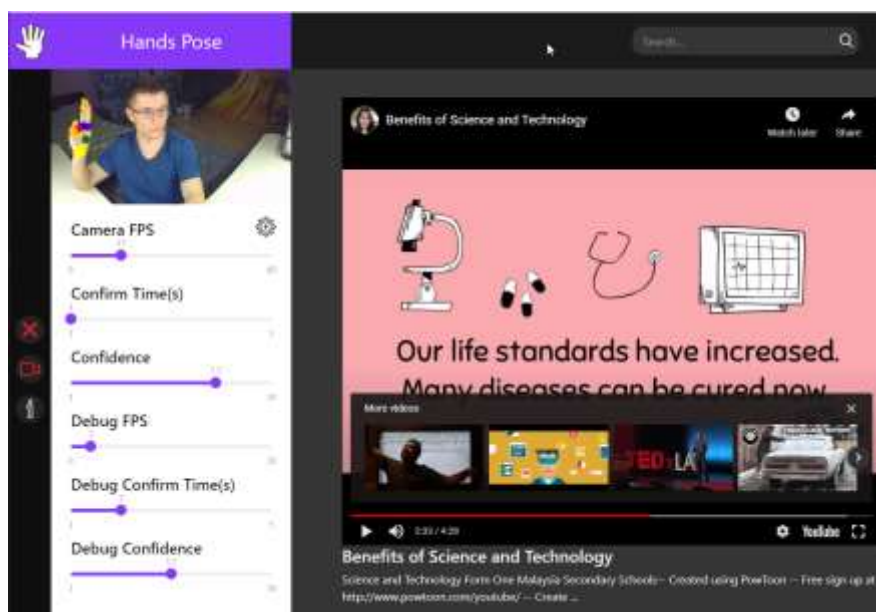


Рисунок 4.5 – Жест «перемотка відео вперед»

Аналогічно тестуємо жести-антагоністи, тобто зниження гучності, перемотка назад, вибір попереднього відео у списку, а також жест, який зупиняє відео та вимикає програвач.

Отже було протестовано додаток із допомогою веб-камери Xiaomi Xiaovv USB 1080P з роздільною здатністю 1920*1080 пікселів і частотою оновлення кадрів 30. Кількість спроб дорівнює 75. Система не розпізнала 10 жестів, але в загальному можна констатувати достовірність 87%. Дане тестування проводилось на базі комп'ютера з процесором Intel Core i5-11600K та з дискретною відеокартою NVIDIA GeForce 570Ti.

Також на тестування було залучено двох колег з іншими апаратними конфігураціями комп'ютера, щоб порівняти та протестувати достовірність розпізнавання жестів.

Колегою Юрієм було протестовано ідентичний набір жестів. На наборі спроб, кількість яких дорівнює 65, достовірність аналізу жестів сягає 85%. В конфігурації його комп'ютера відсутня дискретна відео-карта, але в його процесор Intel Core i5-10700K, інтегрована графіка Intel HD Graphics 6000. Тестування проводилось за допомогою веб-камери LOGITECH C270 з роздільною здатністю 1366*768 пікселів. Деякі знімки екрану під час тестування продемонстровані на рисунку 4.6.



Рисунок 4.6 – Тестувальник Юрій, жест «старт/пауза»

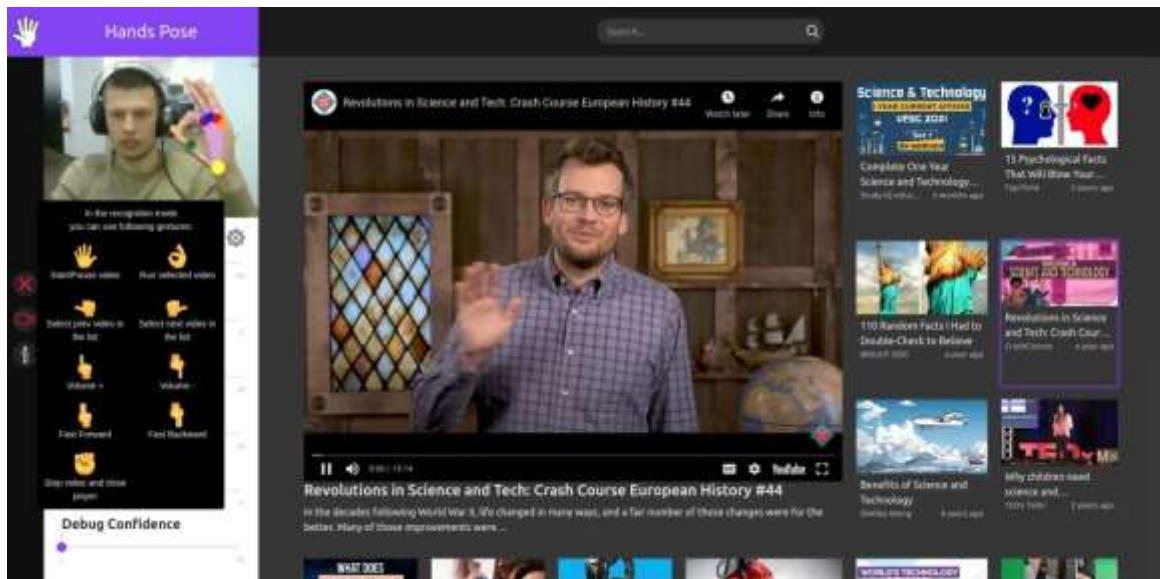


Рисунок 4.7 – Тестувальник Юрій, жест «підтвердження»

Також колегою Надією було протестовано ідентичний набір жестів, програмний модуль продемонстрував найкращі результати серед усіх випробувань. Кількість спроб дорівнювала 60, а невдало розпізнаних жестів всього 6, тому достовірність дорівнює 90%. Тестування проводились на сучасному ноутбукові MacBook Pro 13, хоч даний пристрій немає дискретної відео-карти, але в процес вбудована графіка Intel Iris Plus Graphics 640. Інтегрована веб-камера PhotoBooth MBs підтримує роздільну здатність 1920*1080 пікселів. Деякі знімки екрану під час тестування продемонстровані на рисунку 4.8.

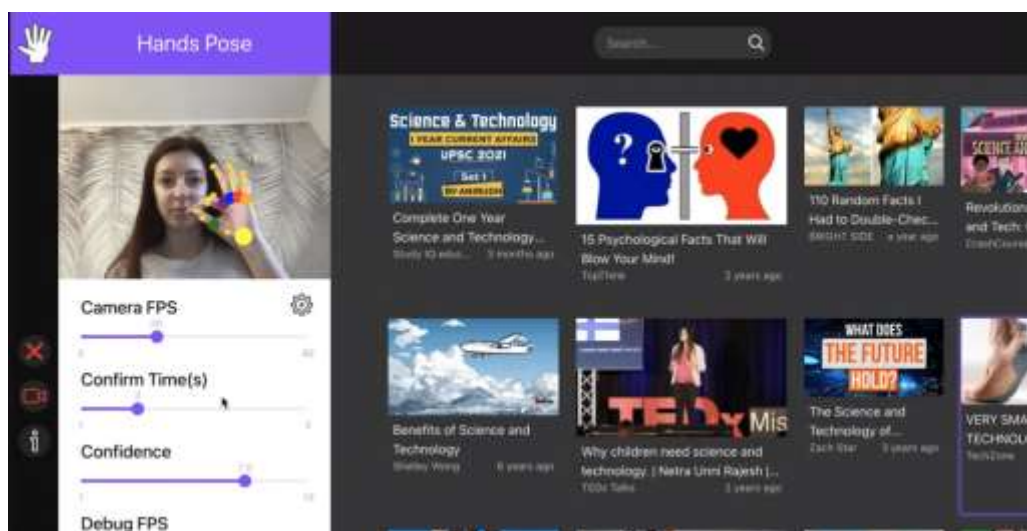


Рисунок 4.8 – Тестувальник Надія, жест «підтвердження»

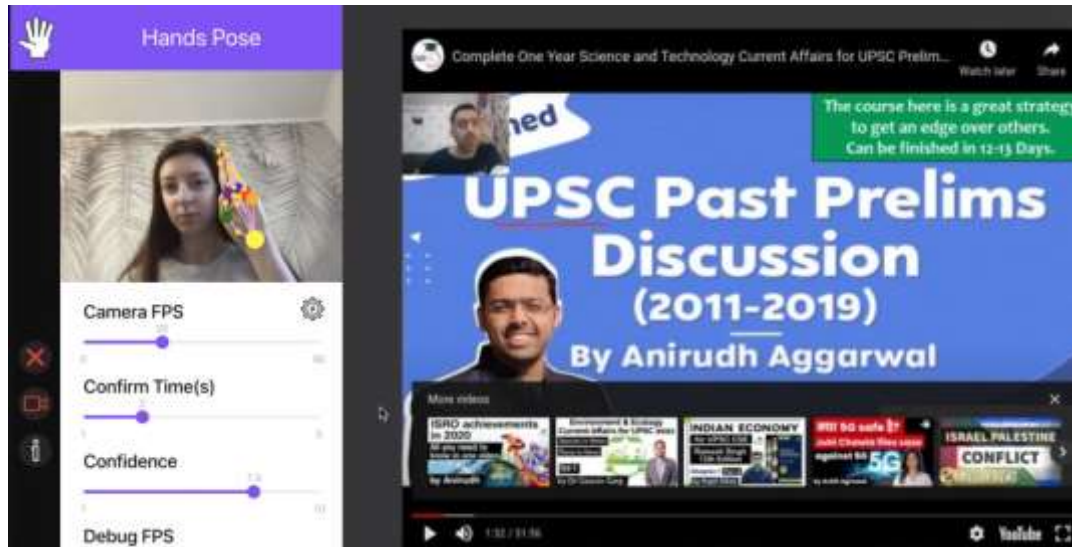


Рисунок 4.9 – Тестувальник Надія, жест «перемотка відео вперед»

Під час тестування було перевірено роботу програмного модулю із допомогою веб-камери Defender C-090 з низькою роздільною здатність, а саме 160*180 пікселів. Як і очікувалось, достовірність розпізнавання не вразила, але елементарні жести, такі як «старт/пауза» чи «вперед/назад», додаток без проблем розумів. З більш складнішими жестами, такими як «знизити гучність» чи «перемотати назад» виникали проблеми. Мережа не могла чітко зрозуміти комбінацію фаланг пальців. Один і прикладів тестування із допомогою цієї веб-камери продемонстровано на рис. 4.11.

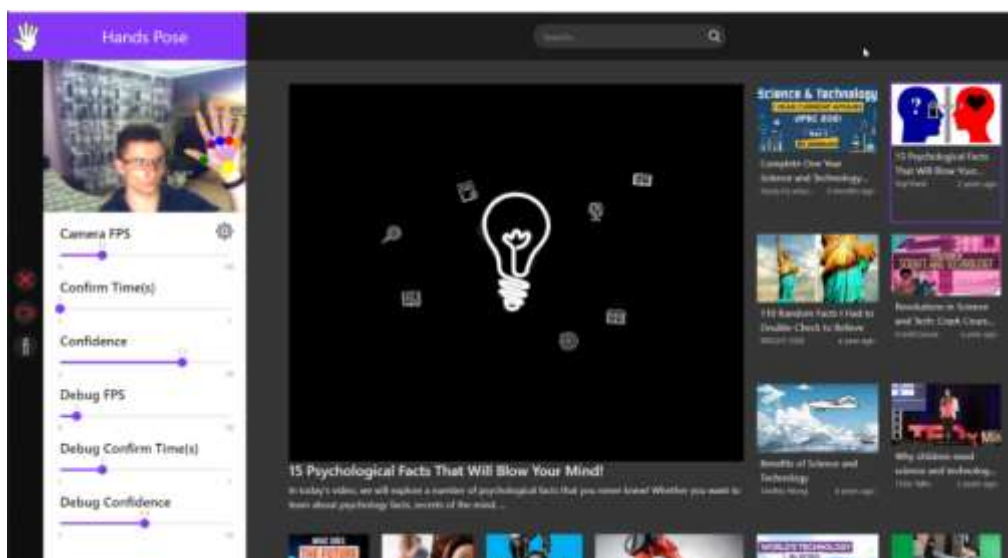


Рисунок 4.11 – Жест «старт/пауза» веб-камери з низьким розширенням

Також було протестовано іншу програмну реалізацію, яка використовує теж саме ядро розпізнавання жестів. А саме це музичний плеєр, який реагує на тіж самі жести що були використанні вище. На рисунку 4.12 продемонстровано жест запуску плеєра з вибраною композицією, за допомогою відповідного жесту.

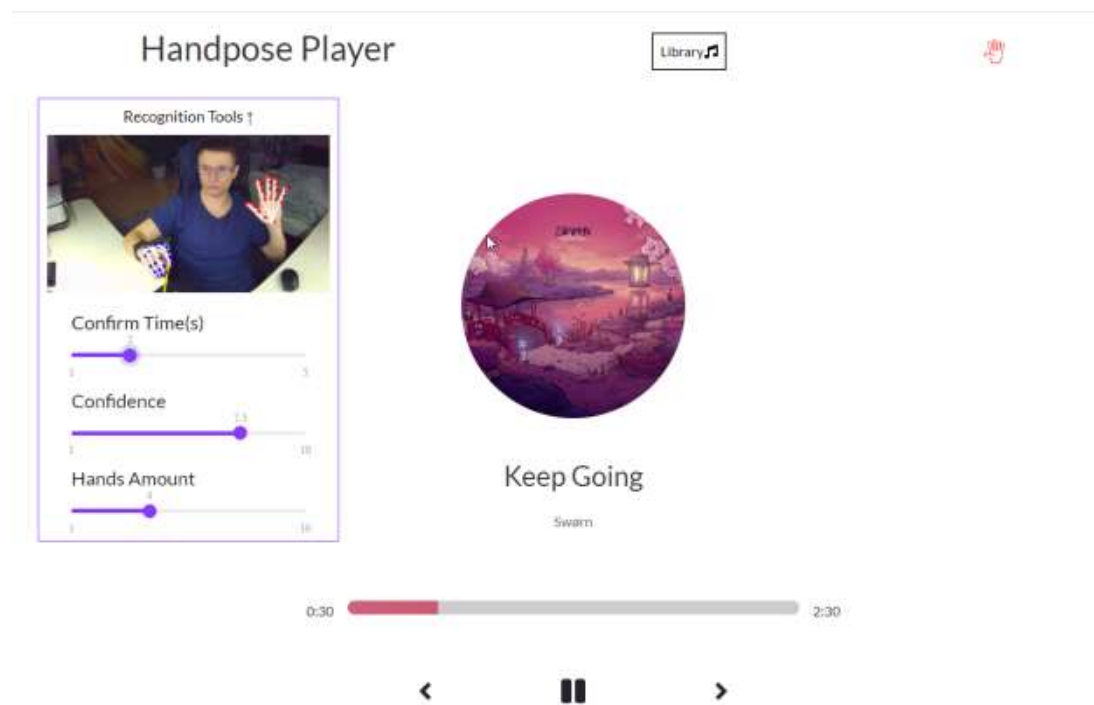


Рисунок 4.12 – Жест «старт/пауза» у веб-додатку музичного плеєра

Наступний жестом демонструємо «вправо» та очікуємо відтворення наступного треку, як продемонстровано на рис. 4.13.

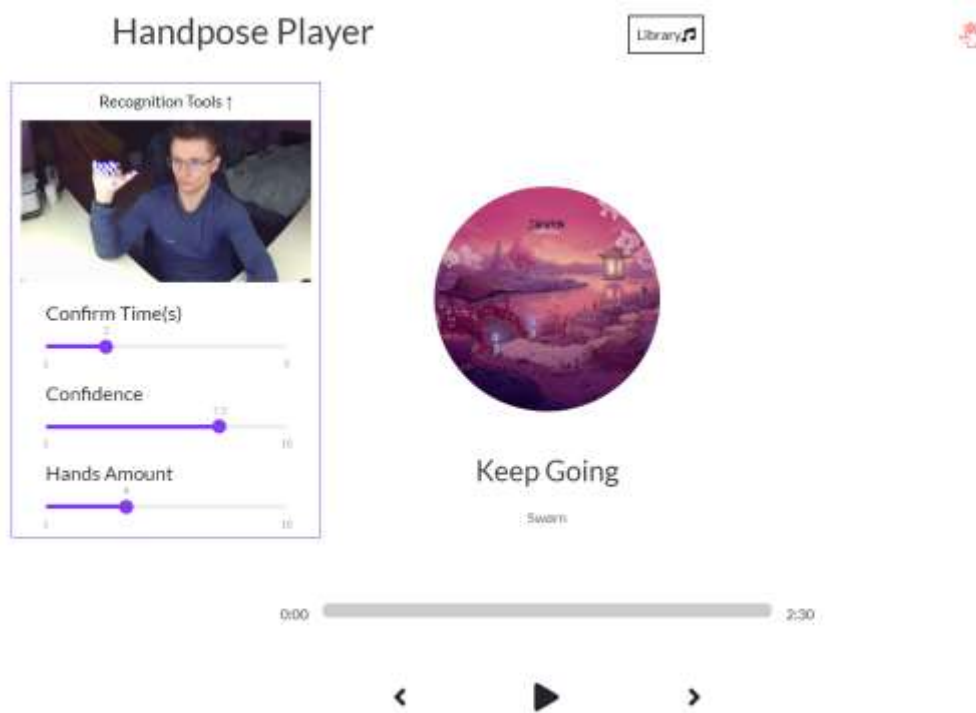


Рисунок 4.13 – Жест «вправо», наступний трек

Наступним кроком можна збільшити рівень звучності музики із допомогою жесту «додати гучності» очікується, що рівень гучності треку буде зростати, як показано на рис. 4.14.

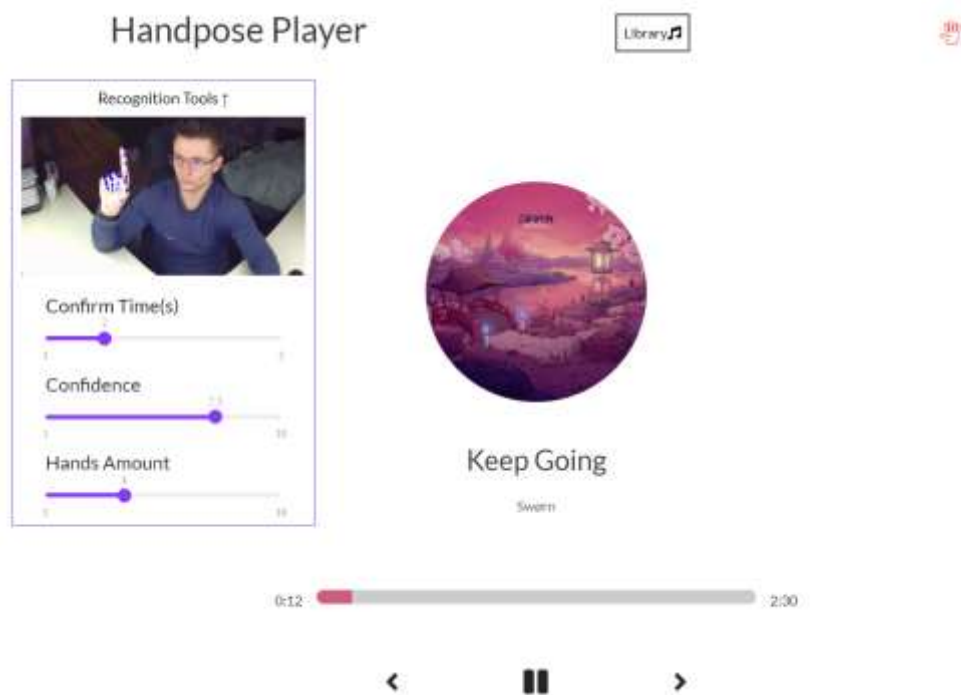


Рисунок 4.14 – Жест «додати гучності»

Далі перевіряємо функцію перемотування музики за допомогою жесту «перемотка музики вперед», передбачається, що трек поступово буде перемотуватися на 10 секунд вперед, як показано на рис. 4.15.

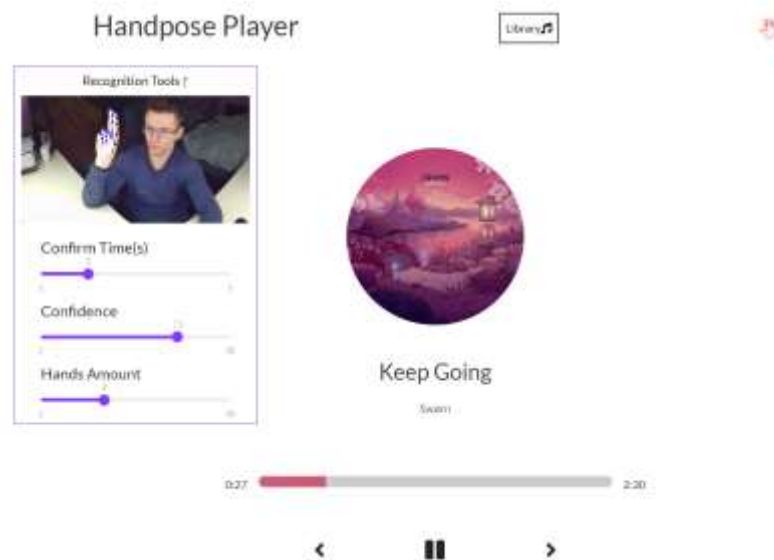


Рисунок 4.15 – Жест «перемотка музики вперед»

Останнім жестом тестування перевіримо «стоп» жест, після виконання якого передбачається, що відтворення музики моментально завершиться та буде змінено поточний трек на перший по списку музичних творів, як показано на рис. 4.16.

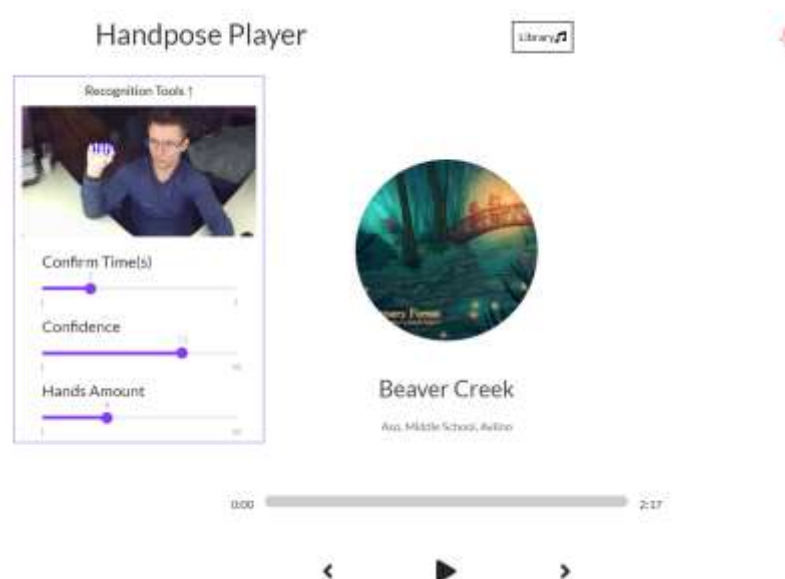


Рисунок 4.16 – Жест «стоп»

4.2 Аналіз результатів тестування інформаційної технології жестового керування web-додатком

Для порівняння результатів достовірності інтелектуальної системи модуля «HandPose», було проведено тестування аналогів, а саме «GestureTek» і «TableFX». Розмір тестової вибірки дорівнює 200 жестів. GestureTek не вірно розпізнав 32 жести, а на вибірці з низькою роздільною здатністю – 60 жестів. TableFX допустив 35 помилок, на вибірці з низькою роздільною здатністю - кількість помилок сягає – 68 жестів. У HandPose кількість не вірно розпізнаних жестів дорівнює 25, а на вибірці з низькою роздільною здатністю – 58 жестів. Результати тестування продемонстровані в таблиці 4.1.

Таблиця 4.1 – Порівняння достовірності розпізнавання аналогів та інтелектуальної системи «HandPose»

Реалізація	К-сть жестів розпізнавання	К-сть вірно розпізнаних жестів	Достовірність(%)	К-сть вірно розпізнаних жестів(низька роз. здатність)	Достовірність (%) (низька роз. здатність)	Середнє значення
GestureTek	200	168	84%	140	70%	77%
TableFX	200	165	82.5%	132	66%	74.25%
HandPose	200	175	87.5%	142	71%	79.25%

З результатів таблиці можна побачити, що система «HandPose» показала результати достовірності розпізнавання в середньому на 4% краще в порівнянні з відомими аналогами. А також це значення підтверджується для тестувальної вибірки з низькою роздільною здатністю.

4.3 Висновок до розділу 4

В ході тестування програмного модулю трьома людьми було перевірено його роботу. Також було протестовано достовірність розпізнавання пристроєм з низькою роздільною здатністю. Інтелектуальна система показала достовірність 87.5%, а з веб-камерою з низькою роздільною здатністю – 71%. Веб-додаток «HandPose» показав результат достовірності від 3.5% до 5% краще серед аналогів.

Також було протестовано веб-додаток музичний плеєр, який використовував туж саму інформаційну систему, протестовано та одержано ідентичний результат розпізнавання, а саме близько 87.5% достовірності. Таким чином мета роботи досягнута, достовірність розпізнавання жестів підвищена, доведено однакового-ефективну модульність інтеграції системи з будь-якими іншими веб-додатками. Також під час тестування система реагувала на жести двох рук, що доводить, що програма може обробляти більше одної руки одночасно.

5 ЕКОНОМІЧНА ЧАСТИНА

5.1 Комерційний та технологічний аудит науково-технічної розробки

Метою даного розділу є проведення технологічного аудиту, в даному випадку нового програмного продукту інформаційна технологія жестового керування WEB-додатком на основі згорткових нейронних мереж «HandPose». Особливістю програми є те програмний продукт працює виключно у веб-середовищі, тобто на сайті та має доступ до веб-камери, з якої отримує і обробляє відео. Користувачу слід у цей час демонструвати, певний конкретний, заздалегідь визначений жест рукою, у полі видимості камери. Результатом обробки відео є виконання певної команди, яка відповідає розпізаному жесту. Розроблена ІТ підвищує достовірність розпізнавання на 85% у порівнянні з аналогами. Аналогами є GESTRACK 3D, який надає можливість грати в прості ігри або орієнтуватись в 3D ландшафтах за ціною 899\$ або 35900 грн. TableFX – технологія взаємодії із поверхнею інтерактивного столу із допомогою жестів, ціна столу 3499\$ або 126000 грн.

Для проведення комерційного та технологічного аудиту залучають не менше 3-х незалежних експертів. Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням п'ятибальної системи оцінювання за 12-ма критеріями, у відповідності із табл. 5.1.

Таблиця 5.1 – Рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можлива бальна оцінка

Бали (за 5-ти бальною шкалою)					
Кри- терій	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність в реальних умовах
Ринкові переваги					
2	Багато аналогів на малому ринку	Ринкові п Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкуренція немає

Продовження табл. 5.1

Практик на здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на ви-	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Усі дані по кожному параметру занесено в таблиці 5.2

Таблиця 5.2 – Результати оцінювання комерційного потенціалу розробки

Критерії оцінювання	ПІБ експертів		
	Експерт 1	Експерт 2	Експерт 3
	Бали		
Технічна здійсненність концепції	4	4	4
Наявність аналогів на ринку	4	4	4
Цінова політика	4	4	4
Технічні та споживчі властивості виробу	4	3	4
Експлуатаційні витрати	4	4	4
Ринок збуту	4	3	4
Конкурентоспроможність	3	4	3
Фахівці з технічної і комерційної реалізації	4	4	4
Фінансування	4	4	3
Матеріально-технічна база	3	3	3
Термін реалізації ідеї	4	3	3
Супровідна документація	3	3	4
Сума	45	43	44
Середньоарифметична сума балів	$(45+43+44) / 3 = 44$		

За даними таблиці 5.2 можна зробити висновок щодо рівня комерційного потенціалу даної розробки. Для цього доцільно скористатись рекомендаціями, наведеними в таблиці 5.3.

Таблиця 5.3 - Рівні комерційного потенціалу розробки

Середньоарифметична сума балів СБ , розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0 - 10	Низький
11 - 20	Нижче середнього
21 - 30	Середній
31 - 40	Вище середнього
41 - 48	Високий

Як видно з таблиці, рівень комерційного потенціалу розроблюваного нового програмного продукту є високим, що досягається за рахунок того, що програмний продукт відрізняється від існуючих тим, що дана технологія пропонує альтернативний підхід взаємодії з комп'ютером, а саме веб-сайтом. Реалізований програмний продукт у вигляді модуля, який буде легко інтегруватися із будь-яким сайтом. Розробка надає можливість одночасно розпізнавати до 10 рук користувачів та підвищує достовірність розпізнавання вище 85% у порівнянні з аналогами.

5.2 Прогнозування витрат на виконання науково-дослідної (дослідно-конструкторської) роботи

5.2.1 Основна заробітна плата розробників, яка розраховується за формулою:

$$Z_o = \frac{M}{T_p} \cdot t, \quad (5.1)$$

де M – місячний посадовий оклад конкретного розробника (дослідника), грн.;

T_p – число робочих днів в місяці, 20 днів;

t – число днів роботи розробника (дослідника).

Результати розрахунків зведемо до таблиці 5.4.

Таблиця 5.4 – Основна заробітна плата розробників

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник проекту	48500	2425,00	45	109125,000
Інженер	45200	2260,00	45	101700,000
Всього				210825,00

Так як в даному випадку розробляється програмний продукт, то розробник виступає одночасно і основним робітником, і тестувальником розроблюваного програмного продукту.

5.2.2 Додаткова заробітна плата розробників, які приймали участь в розробці обладнання.

Додаткова заробітна плата прийнято розраховувати як 15 % від основної заробітної плати розробників та робітників:

$$Z_d = Z_o \cdot 15 \% / 100 \% \quad (5.2)$$

$$Z_d = (210825,00 \cdot 15 \% / 100 \%) = 31623,75 \text{ (грн.)}$$

5.2.3 Нарахування на заробітну плату розробників.

Згідно діючого законодавства нарахування на заробітну плату складають 22 % від суми основної та додаткової заробітної плати.

$$H_z = (Z_o + Z_d) \cdot 22 \% / 100\% \quad (5.3)$$

$$H_z = (210825,00 + 31623,75) \cdot 22 \% / 100 \% = 53338,73 \text{ (грн.)}$$

5.2.4. Оскільки для розроблювального пристрою не потрібно витратити матеріали та комплектуючі, то витрати на матеріали і комплектуючі дорівнюють нулю.

5.2.5 Амортизація обладнання, яке використовувалось для проведення розробки.

Амортизація обладнання, що використовувалось для розробки в спрощеному вигляді амортизація обладнання, що використовувалась для розробки розраховується за формулою:

$$A = \frac{Ц}{T} \cdot \frac{t_{вик}}{12} \text{ [грн.]} \quad (5.4)$$

де Ц – балансова вартість обладнання, грн.;

T – термін корисного використання обладнання згідно податкового законодавства, років

$t_{вик}$ – термін використання під час розробки, місяців

Розрахуємо, для прикладу, амортизаційні витрати на комп'ютер балансова вартість якого становить 73628 грн., термін його корисного використання згідно податкового законодавства – 2 роки, а термін його фактичного використання – 2,25 міс.

$$A_{обл} = \frac{73628}{2} \times \frac{2,25}{12} = 6902,625 \text{ грн.}$$

Аналогічно визначаємо амортизаційні витрати на інше обладнання та приміщення. Розрахунки заносимо до таблиці 5.5. Для розрахунку амортизації нематеріальних ресурсів використовується формула:

$$A_{н.р.} = Ц_{н.р.} * H_a * \frac{t_{вик.}}{12} \quad (5.5)$$

Але, так як вартість ліцензійної ОС та спеціалізованих ліцензійних

нематеріальних ресурсів менше 20000 грн, то даний нематеріальний актив не амортизується, а його вартість включається у вартість розробки повністю, $V_{нем.ак.} = 6934$ грн.

Таблиця 5.5 – Амортизаційні відрахування матеріальних і нематеріальних ресурсів для розробників

Найменування обладнання	Балансова вартість, грн.	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн.
Комп'ютер та комп'ютерна периферія	73628	2	2,25	6902,625
Офісне обладнання	27000	4	2,25	1265,625
Приміщення	1500000	20	2,25	14062,500
Всього				22230,75

5.2.6 Тарифи на електроенергію для побутових споживачів (промислових підприємств) відрізняються від тарифів на електроенергію для населення. При цьому тарифи на розподіл електроенергії у різних постачальників (енергорозподільних компаній), будуть різними. Крім того, розмір тарифу залежить від класу напруги (1-й або 2-й клас). Тарифи на розподіл електроенергії для всіх енергорозподільних компаній встановлює Національна комісія з регулювання енергетики і комунальних послуг (НКРЕКП). Витрати на силову електроенергію розраховуються за формулою:

$$V_e = V \cdot \Pi \cdot \Phi \cdot K_{\Pi}, \quad (5.6)$$

де V – вартість 1 кВт-години електроенергії для 1 класу підприємства, $V = 6,2$ грн./кВт;

Π – встановлена потужність обладнання, кВт. $\Pi = 0,4$ кВт;

Φ – фактична кількість годин роботи обладнання, годин.

K_{π} – коефіцієнт використання потужності, $K_{\pi} = 0,9$.

$$B_e = 0,9 \cdot 0,4 \cdot 8 \cdot 45 \cdot 6,2 = 803,52 \text{ (грн.)}$$

5.2.7 Інші витрати та загальновиробничі витрати.

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками. Витрати за статтею «Інші витрати» розраховуються як 100...150% від суми основної заробітної плати дослідників:

$$I_e = (Z_o + Z_p) \cdot \frac{H_{ib}}{100\%}, \quad (5.7)$$

де H_{ib} – норма нарахування за статтею «Інші витрати».

$$I_e = 210825,00 \cdot 150\% / 100\% = 316237,5 \text{ (грн.)}$$

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін. Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуються як 100...150% від суми основної заробітної плати дослідників:

$$H_{нзв} = (Z_o + Z_p) \cdot \frac{H_{нзв}}{100\%}, \quad (5.8)$$

де $H_{нзв}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати».

$$H_{нзв} = 210825,00 * 150 \% / 100 \% = 316238 \text{ (грн.)}$$

5.2.8 Витрати на проведення науково-дослідної роботи.

Сума всіх попередніх статей витрат дає загальні витрати на проведення науково-дослідної роботи:

$$B_{заг} = 210825,00 + 31623,75 + 53338,73 + 22230,75 + 6934 + 803,52 + 316237,5 + 316238 = 958230,75 \text{ грн.}$$

5.2.9 Розрахунок загальних витрат на науково-дослідну (науково-технічну) роботу та оформлення її результатів.

Загальні витрати на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховуються ZB , визначається за формулою:

$$ZB = \frac{B_{заг}}{\eta} \text{ (грн)}, \quad (5.9)$$

де η – коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи.

Так, якщо науково-технічна розробка знаходиться на стадії: науково-дослідних робіт, то $\eta=0,1$; технічного проектування, то $\eta=0,2$; розробки конструкторської документації, то $\eta=0,3$; розробки технологій, то $\eta=0,4$; розробки дослідного зразка, то $\eta=0,5$; розробки промислового зразка, то $\eta=0,7$; впровадження, то $\eta=0,9$. Оберемо $\eta = 0,5$, так як розробка, на даний момент, знаходиться на стадії дослідного зразка:

$$ZB = 958230,75 / 0,5 = 1916461 \text{ грн.}$$

5.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором

В ринкових умовах узагальнювальним позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів тієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку. Саме зростання чистого прибутку забезпечить потенційному інвестору надходження додаткових коштів, дозволить покращити фінансові результати його діяльності, підвищить конкурентоспроможність та може позитивно вплинути на ухвалення рішення щодо комерціалізації цієї розробки.

Для того, щоб розрахувати можливе зростання чистого прибутку у потенційного інвестора від можливого впровадження науково-технічної розробки необхідно:

а) вказати, з якого часу можуть бути впроваджені результати науково-технічної розробки;

б) зазначити, протягом скількох років після впровадження цієї науково-технічної розробки очікуються основні позитивні результати для потенційного інвестора (наприклад, протягом 3-х років після її впровадження);

в) кількісно оцінити величину існуючого та майбутнього попиту на цю або аналогічні чи подібні науково-технічні розробки та назвати основних суб'єктів (зацікавлених осіб) цього попиту;

г) визначити ціну реалізації на ринку науково-технічних розробок з аналогічними чи подібними функціями.

При розрахунку економічної ефективності потрібно обов'язково враховувати зміну вартості грошей у часі, оскільки від вкладення інвестицій до отримання прибутку минає чимало часу. При оцінюванні ефективності інноваційних проектів передбачається розрахунок таких важливих показників:

- абсолютного економічного ефекту (чистого дисконтованого доходу);

- внутрішньої економічної дохідності (внутрішньої норми дохідності);
- терміну окупності (дисконтованого терміну окупності).

Аналізуючи напрямки проведення науково-технічних розробок, розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором можна об'єднати, враховуючи визначені ситуації з відповідними умовами.

5.3.1 Розробка чи суттєве вдосконалення програмного засобу (програмного забезпечення, програмного продукту) для використання масовим споживачем.

В цьому випадку майбутній економічний ефект буде формуватися на основі таких даних:

$$\Delta\Pi_i = (\pm\Delta\Pi_0 \cdot N + \Pi_0 \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\rho}{100}\right), \quad (5.10)$$

де $\pm\Delta\Pi_0$ – зміна вартості програмного продукту (зростання чи зниження) від впровадження результатів науково-технічної розробки в аналізовані періоди часу;

N – кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки;

Π_0 – основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки, $\Pi_0 = \Pi_0 \pm \Delta\Pi_0$;

Π_0 – вартість програмного продукту у році до впровадження результатів розробки;

ΔN – збільшення кількості споживачів продукту, в аналізовані періоди часу, від покращення його певних характеристик;

λ – коефіцієнт, який враховує сплату податку на додану вартість. Ставка податку на додану вартість дорівнює 20%, а коефіцієнт $\lambda = 0,8333$.

p – коефіцієнт, який враховує рентабельність продукту;

ϑ – ставка податку на прибуток, у 2022 році $\vartheta = 18\%$.

Припустимо, що при прогнозованій ціні 13000 грн. за одиницю виробу, термін збільшення прибутку складе 3 роки. Після завершення розробки і її вдосконалення, можна буде підняти її ціну на 300 грн. Кількість одиниць реалізованої продукції також збільшиться: протягом першого року – на 15000 шт., протягом другого року – на 5000 шт., протягом третього року на 3000 шт. До моменту впровадження результатів наукової розробки реалізації продукту не було:

$$\Delta\Pi_1 = (0*300 + (13000 + 300) * 15000) * 0,8333 * 0,18 * (1 - 0,18) = 23984999,041 \text{ грн.}$$

$$\Delta\Pi_2 = (0*300 + (13000 + 300) * (15000 + 5000)) * 0,8333 * 0,18 * (1 - 0,18) = 32717998,691 \text{ грн.}$$

$$\Delta\Pi_3 = (0*300 + (13000 + 300) * (15000 + 5000 + 3000)) * 0,8333 * 0,18 * (1 - 0,18) = 37625698,495 \text{ грн.}$$

Отже, комерційний ефект від реалізації результатів розробки за три роки складе 94328696,23 грн.

5.3.2 Розрахунок ефективності вкладених інвестицій та періоду їх окупності.

Розраховуємо приведену вартість збільшення всіх чистих прибутків $ПП$, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_1^T \frac{\Delta\Pi_i}{(1 + \tau)^t}, \quad (5.11)$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої науково-дослідної (науково-технічної) роботи, грн;

T – період часу, протягом якою виявляються результати впровадженої науково-дослідної (науково-технічної) роботи, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau = 0,05 \dots 0,15$;

t – період часу (в роках).

Збільшення прибутку ми отримаємо починаючи з першого року:

$$\text{ПП} = (23984999,041/(1+0,1)^1) + (32717998,691/(1+0,1)^2) + (37625698,495/(1+0,1)^3) = 21804544,58 + 27039668,34 + 28268744,17 = 77112957,1 \text{ грн.}$$

Далі розраховують величину початкових інвестицій PV , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки. Для цього можна використати формулу:

$$PV = k_{inv} * ZB, \quad (5.12)$$

де k_{inv} – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію. Це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай $k_{inv} = 2 \dots 5$, але може бути і більшим;

ZB – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

$$PV = 2 * 1916461 = 3832922,98 \text{ грн.}$$

Тоді абсолютний економічний ефект E_{abs} або чистий приведений дохід (NPV , *Net Present Value*) для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{abc} = III - PV, \quad (5.13)$$

$$E_{abc} = 77112957,1 - 3832922,98 = 73280034,12 \text{ грн.}$$

Оскільки $E_{abc} > 0$ то вкладання коштів на виконання та впровадження результатів даної науково-дослідної (науково-технічної) роботи може бути доцільним.

Для остаточного прийняття рішення з цього питання необхідно розрахувати внутрішню економічну дохідність або показник внутрішньої норми дохідності (*IRR, Internal Rate of Return*) вкладених інвестицій та порівняти її з так званою бар'єрною ставкою дисконтування, яка визначає ту мінімальну внутрішню економічну дохідність, нижче якої інвестиції в будь-яку науково-технічну розробку вкладати буде економічно недоцільно.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій E_e . Для цього використаємо формулу:

$$E_e = \sqrt[T_{жс}]{1 + \frac{E_{abc}}{PV}} - 1, \quad (5.14)$$

$T_{жс}$ – життєвий цикл наукової розробки, роки.

$$E_e = \sqrt[3]{(1 + 73280034,12/3832922,98) - 1} = 1,720$$

Визначимо мінімальну ставку дисконтування, яка у загальному вигляді визначається за формулою:

$$\tau = d + f, \quad (5.15)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2022 році в Україні $d = (0,09...0,14)$;

f – показник, що характеризує ризикованість вкладень; зазвичай, величина $f = (0,05...0,5)$.

$$\tau_{\min} = 0,14 + 0,05 = 0,19.$$

Так як $E_v > \tau_{\min}$, то інвестор може бути зацікавлений у фінансуванні даної наукової розробки.

Розрахуємо термін окупності вкладених у реалізацію наукового проекту інвестицій за формулою:

$$T_{ок} = \frac{1}{E_v}, \quad (5.16)$$

$$T_{ок} = 1 / 1,720 = 0,58 \text{ р.}$$

Оскільки $T_{ок} < 3$ -х років, а саме термін окупності рівний 0,58 роки, то фінансування даної наукової розробки є доцільним.

5.4 Висновок до розділу 5

У економічному розділі було виконано розрахунок витрат на розробку нового програмного продукту, сума яких складає 1916461 гривень. Було спрогнозовано орієнтовану величину витрат по кожній з статей витрат. Також розраховано чистий прибуток, який може отримати виробник від реалізації нового технічного рішення, розраховано період окупності витрат для інвестора та економічний ефект при використанні даної розробки. В результаті аналізу розрахунків можна зробити висновок, що розроблений програмний продукт за ціною дешевший за аналог і є висококонкурентоспроможним. Період окупності складе близько 0,58 роки.

ВИСНОВКИ

У ході виконання магістерської кваліфікаційної було запропоновано альтернативний підхід взаємодії з веб-додатком, а саме управління сайтом із допомогою жестів.

У першому розділі магістерської роботи було проаналізовано існуючі програмні продукти та реалізації, було поставлено за мету реалізувати інтелектуальний модуль, який зможе продемонструвати достовірність розпізнавання 85% і вище, в порівнянні з аналогами, а також розробити програмне забезпечення, яке дасть змогу інтегруватися з будь-яким WEB-додатком і кількість одночасно розпізнаваних рук буде від 2 до 10 .

У другому розділі магістерської кваліфікаційної роботи було розглянуто декілька видів нейронних мереж і обрано для програмної реалізації згорткову нейронну мережу, так як вона має найточніші механізми аналізу зображення, які допоможуть вирішити поставлену задачу. Також було проаналізовано структуру, математичну модель та принцип функціонування згорткової нейронної мережі. Було сформовано вимоги до вхідного сигналу нейронної мережі. Розроблено алгоритм функціонування інформаційної технології, а також запропоновано архітектуру, яка представляє собою ядро системи обгорнуте у вигляді пакету. Який у свою чергу надає інтерфейс за для зручної інтеграції інформаційної технології у будь-які інші системи.

У третьому розділі було проаналізовано три мови програмування. Для реалізації поставленої задачі було обрано мову програмування JavaScript, так як це єдина мова, яка надає гнучкі можливості реалізації цільового функціоналу у браузері. Також було обґрунтовано вибір середовища розробки WebStorm. На основі підбраної конфігурації технічних інструментів було оглянуто специфіку реалізації та програмно реалізовано інтелектуальну систему засобами бібліотек React.js та Tensorflow.js.

У четвертому розділі проводилось тестування веб-додатку трьома людьми, було перевірено його роботу. Веб-додаток «HandPose» показав результат достовірності від 3.5% до 5% краще серед аналогів. Також було протестовано веб-додаток музичний плеєр, який використовував теж саме ядро системи, одержано ідентичний результат розпізнавання, а саме близько 87.5% достовірності. Значення бажаної достовірності розпізнавання жестів було досягнуто та доведено модульність інтеграції системи з іншими додатками. Також під час тестування система реагувала на жести двох рук, що доводить, що програма може обробляти більше одної руки одночасно.

У п'ятому розділі було виконано розрахунок витрат на розробку та виготовлення нового технічного рішення, сума яких складає 1916461 гривень. Було спрогнозовано орієнтовану величину витрат по кожній з статей витрат. Також розраховано чистий прибуток, який потенційно може отримати виробник від реалізації розробленого технічного рішення, знайдено термін окупності витрат виробника та економічний ефект для споживача при використанні даної розробки. В результаті аналізу розрахунків можна зробити висновок, що розробка у виробництві та використання дешевша за аналог і є висококонкурентоспроможною. Період окупності складе близько 0,58 роки.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. В. Каліцінський. Розробка інформаційної системи жестового керування веб-додатком на основі згорткових нейронних мереж. Матеріали конференції «Молодь в науці: дослідження, проблеми та перспективи (МН-2021)», Вінниця-2021. [Електронний ресурс] – Режим доступу: <https://conferences.vntu.edu.ua/index.php/mn/mn2021/paper/view/12679>.
2. В. Каліцінський. Розробка інформаційної технології жестового керування веб-додатком на основі згорткових нейронних мереж. Матеріали конференції «Молодь в науці: дослідження, проблеми та перспективи (МН-2022)», Вінниця-2022. [Електронний ресурс] – Режим доступу: <https://conferences.vntu.edu.ua/index.php/mn/mn2022/paper/view/16199>.
3. Dix A., Finlay J., Abowd G.D., Beale R. Human-Computer Interaction. — Third Edition, Pearson Education Limited: 2004. — 857 с.
4. Bader T., Beyerer J. Natural Gaze Behavior as Input Modality for HumanComputer Interaction // Eye Gaze in Intelligent User Interfaces / Ed. by Y. I. Nakano, C. Conati, T. Bader. Springer London, 2013. 161– 183 с.
5. Leap Motion Controller. URL: <https://www.leapmotion.com/> (дата обращения: 22.10.2013)
6. Real-Time Hand Gesture Recognition Using Finger Segmentation [Електронний ресурс] – Режим доступу: <https://www.hindawi.com/journals/tswj/2014/267872/>
7. Zhi-hua Chen, Jung-Tae Kim, Jianning Liang, Jing Zhang, Yu-Bo Yuan. «Real-Time Hand Gesture Recognition Using Finger Segmentation». [Електронний ресурс] – Режим доступу: <https://www.hindawi.com/journals/tswj/2014/267872/>.
8. Невмержицький Р.В. «Згорткові нейронні мережі». [Електронний ресурс] – Режим доступу: <https://conf.ztu.edu.ua/wp-content/uploads/2019/02/45-1.pdf>

9. «Комп'ютерний зір». [Електронний ресурс] – Режим доступу: https://wiki.uk-ua.nina.az/Комп'ютерний_зір.html
10. Гонсалес Р., Вудс Р. Цифрова обробка зображень. М.: Техносфера, 2005. – 1072 с.
11. GestureTek [Електронний ресурс] – Режим доступу: <http://www.gesturetek.com>
12. TableFX [Електронний ресурс] – Режим доступу: <https://www.av-iq.com/avcat/ctl1642/index.cfm?manufacturer=gesturetek&product=tablefx>
13. ManoMotion [Електронний ресурс] – Режим доступу: <https://www.manomotion.com>
14. Нейронні мережі: Їх застосування, Робота [Електронний ресурс] – Режим доступу: <https://www.poznavayka.org/uk/nauka-i-tehnika-2/neyronni-merezhiih-zastosuvannya-robota/>.
15. Згорткова нейронна мережа – просте пояснення CNN та її застосування [Електронний ресурс] – Режим доступу: <https://evergreens.com.ua/ua/articles/cnn.html>.
16. Буц Вікторія Віталіївна, Магістерська дисертація, «Способи організації засобів нейромережевого розпізнавання об'єкта на зображенні» [Електронний ресурс] – Режим доступу: https://ela.kpi.ua/bitstream/123456789/23184/1/Buts_magistr.pdf.
17. Krizhevsky A. Imagenet classification with deep convolutional neural networks / A. Krizhevsky, I. Sutskever, G. E. Hinton. // Advances in neural information processing systems. – 2012.
18. Vedaldi A. Convolutional neural networks for matlab / A. Vedaldi, K. Lenc. // Proceedings of the 23rd Annual ACM Conference on Multimedia Conference. – 2015.
19. Dai J. Convolutional feature masking for joint object and stuff segmentation / J. Dai, K. He, J. Sun. // CVPR. – 2015.

20. Welcome to Python [Электронный ресурс] – Режим доступа:
<https://www.python.org>
21. C++ [Электронный ресурс] – Режим доступа:
<https://sites.google.com/site/korkholeh2/cppfund.pdf>
22. Д. Фленаган. «JavaScript. Полное руководство» 2018 с.358-362
23. К. Сімпсон «ES6 і не тільки» 2017 с.224-229
24. WebStorm: The Smartest JavaScript IDE by JetBrains [Электронный ресурс] –
Режим доступа: <https://www.jetbrains.com/webstorm/>
25. Npm [Электронный ресурс] – Режим доступа: <https://www.npmjs.com/>
26. React [Электронный ресурс] – Режим доступа: <https://uk.reactjs>

ДОДАТКИ

Додаток А (обов'язковий)

Результат перевірки на плагіат в онлайн-системі UNICHECK



Ім'я користувача:
Озеранський В. С. КН

ID перевірки:
1011461162

Дата перевірки:
04.06.2022 16:04:04 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
04.06.2022 16:06:57 EEST

ID користувача:
62038

Назва документа: 122МКР-СобчукКР2022

Кількість сторінок: 61 Кількість слів: 11574 Кількість символів: 88346 Розмір файлу: 1.23 MB ID файлу: 1011339465

4.36% Схожість

Найбільша схожість: 4.36% з джерелом з Бібліотеки (ID файлу: 1003991479)

2.01% Джерела з Інтернету 33 Сторінка 63

4.36% Джерела з Бібліотеки 9 Сторінка 63

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

5.72% Вилучень

Деякі джерела вилучено автоматично (фільтри вилучення: кількість знайдених слів є меншою за 8 слів та 5%)

1.98% Вилучення з Інтернету 69 Сторінка 64

4.24% Вилученого тексту з Бібліотеки 152 Сторінка 65

Додаток Б (обов'язковий)

Лістинг програми

```

import { createDetector, SupportedModels } from '@tensorflow-models/hand-pose-detection';
import fp from "fingerpose";
import _ from "lodash";
import RecognitionSettings from "./utils/RecognitionSettings";
import gestures from "./gestures";

import WebCamera from "./WebCamera";

const {
  highFiveGesture,
  fistGesture,
  okGesture,
  leftGesture,
  rightGesture,
  oneFingerUp,
  twoFingerUp,
  oneFingerDown,
  twoFingerDown,
} = gestures;

class Core {
  constructor({ handleGestureSubmit, onWebCamError }) {
    this.detector = null;
    this.gestureEstimator = null;

    this.tempGesture = null;

    this.recognitionInterval = null;
    this.confirmTimer = null;
    this.handleGestureSubmit = handleGestureSubmit;
    this.onWebCamError = onWebCamError;
    this.rafId = null;

    this.recognitionSettings = RecognitionSettings.getSettings();

    this.camera = new WebCamera();
    this.observingRecognition = false;
  }

  async createDetector(){

```

```

const model = SupportedModels.MediaPipeHands;
const config = {
  runtime: 'mediapipe', // or 'tfjs',
  solutionPath: 'https://cdn.jsdelivr.net/npm/@mediapipe/hands',
  modelType: 'full',
  maxHands: this.recognitionSettings.maxHands,
}

this.detector = await createDetector(model, config);
}

startObservingRecognition() {
  this.observingRecognition = true;
  return this.camera.returnLiveDetection();
}

stopObservingRecognition() {
  this.observingRecognition = false;
  this.camera.clearCtx();
}

async start() {
  try {
    await this.camera.setupCamera(
      {targetFPS: 60, sizeOption: 'fixed'},
      this.onWebCamError
    );
    await this.createDetector();
    this.gestureEstimator = new fp.GestureEstimator([
      highFiveGesture,
      fistGesture,
      okGesture,
      rightGesture,
      leftGesture,
      oneFingerUp,
      oneFingerDown,
      twoFingerUp,
      twoFingerDown,
    ]);
    this.detect();
    // this.recognitionInterval = setInterval(() => {
    //   this.detect();
    // }, detectInterval);
  } catch (error) {

```

```

        throw new Error({
            isOpen: true,
            title: "Recognition Error",
            message: "Something went wrong with recognition, check your connection and try
reload the page",
        });
    }
}

async reRunRecognition() {
    window.cancelAnimationFrame(this.rafId);
    await this.start();
}

stop() {
    window.cancelAnimationFrame(this.rafId);
    this.stopObservingRecognition();

    // this.detector.dispose();

    // clearInterval(this.recognitionInterval);
    // this.recognitionInterval = null;
}

async detect() {
    if (
        this.camera?.video?.readyState === 4
    ) {
        const video = this.camera.video;
        const { confidence } = this.recognitionSettings;

        const originalEstimation = await this.detector.estimateHands(video);
        // const originalEstimation = await this.detector.estimateHands(video,
{flipHorizontal: true});

        const adoptedEstimation = originalEstimation.map((item) => ({
            landmarks: item.keypoints?.map(({x, y}) => ([x, y, 0]))
        }));

        if (adoptedEstimation.length > 0) {
            const temp = adoptedEstimation.map((hand) => {
                if (_.isEmpty(hand?.landmarks)) {
                    return { confidence: null, gesture: null };
                }
            });
        }
    }
}

```

```

    }

    const result = this.gestureEstimator.estimate(hand.landmarks, confidence);

    if (_.isEmpty(result?.gestures)) {
      return { confidence: null, gesture: null };
    }

    const confidenceList = result.gestures.map(prediction =>
prediction.score);

    const maxConfidence = Math.max(...confidenceList);
    const maxConfidenceGestureIndex = confidenceList.indexOf(maxConfidence);
    const gesture = result.gestures[maxConfidenceGestureIndex].name;

    return { confidence: maxConfidence, gesture };
  });

  const newGesture = _.maxBy(temp, 'confidence)?.gesture;

  this.confirmGesture(newGesture || null);
} else {
  this.confirmGesture(null);
}

if (this.observingRecognition) {
  this.camera.setHands(originalEstimation);
}
}
this.rafId = requestAnimationFrame(this.detect.bind(this));
};

confirmGesture(gesture) {
  const { confirmTime } = this.recognitionSettings;

  if (gesture !== this.tempGesture) {
    clearTimeout(this.confirmTimer);
    this.tempGesture = gesture;

    if (gesture) {
      this.confirmTimer = setTimeout(() => {
        this.handleGestureSubmit(gesture);
        this.tempGesture = null;
      }, confirmTime);
    }
  }
}

```

```
        } else {
            this.confirmTimer = null;
        }
    }
}

getRecognitionSettings() {
    return this.recognitionSettings;
}

updateSetting(field, value) {
    this.recognitionSettings = {
        ...this.recognitionSettings,
        [field]: value,
    };
    RecognitionSettings.updateSetting(field, value);
    this.reRunRecognition();
}

setHandleGestureSubmit(newHandleGestureSubmit) {
    this.handleGestureSubmit = newHandleGestureSubmit;
}
}

export default Core;
```


Додаток В (обов'язковий)

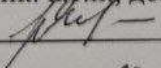
Додаток В (обов'язковий)

93

ІЛЮСТРАТИВНА ЧАСТИНА

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ЖЕСТОВОГО КЕРУВАННЯ
WEB-ДОДАТКОМ НА ОСНОВІ ЗГОРТКОВИХ НЕЙРОННИХ
МЕРЕЖ

Виконав: студент 2-го курсу,
групи 1КН-21м
спеціальності 122 «Комп'ютерні науки»
(шифр і назва напрямку підготовки, спеціальності)
 Каліцинський В.В.
(прізвище та ініціали)

Керівник: к.т.н., доцент каф. КН
 Колесницький О.К.
(прізвище та ініціали)
« 16 » 12 2022 р.

Вінниця ВНТУ - 2022 рік



Рисунок В.1 – Архітектура запропонованої згорткової нейронної мережі, що спроектована на основі SSD архітектури.

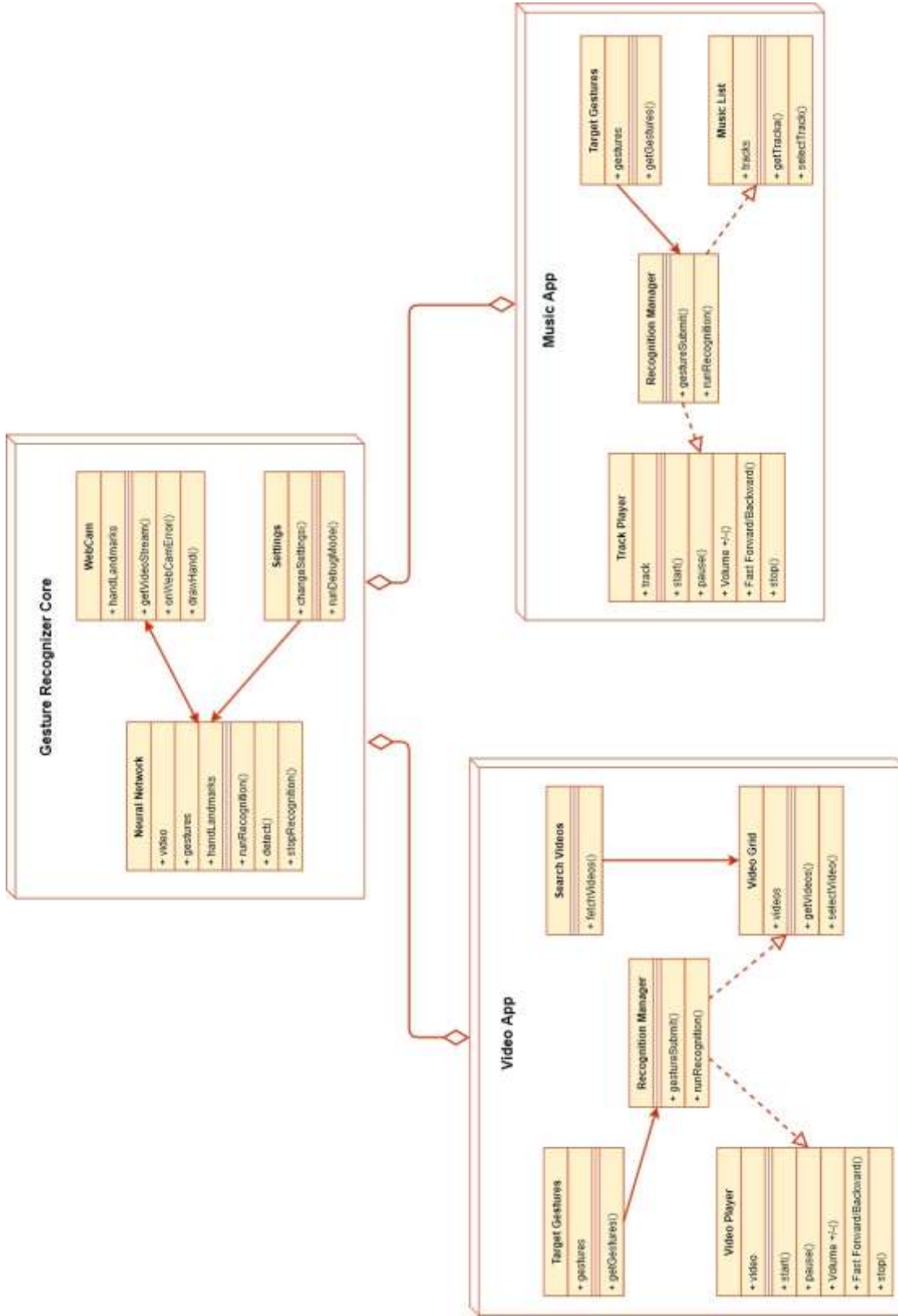


Рисунок В.2 – UML-діаграма класів інформаційної технології жестового керування WEB-додатком на основі згорткових нейронних мереж

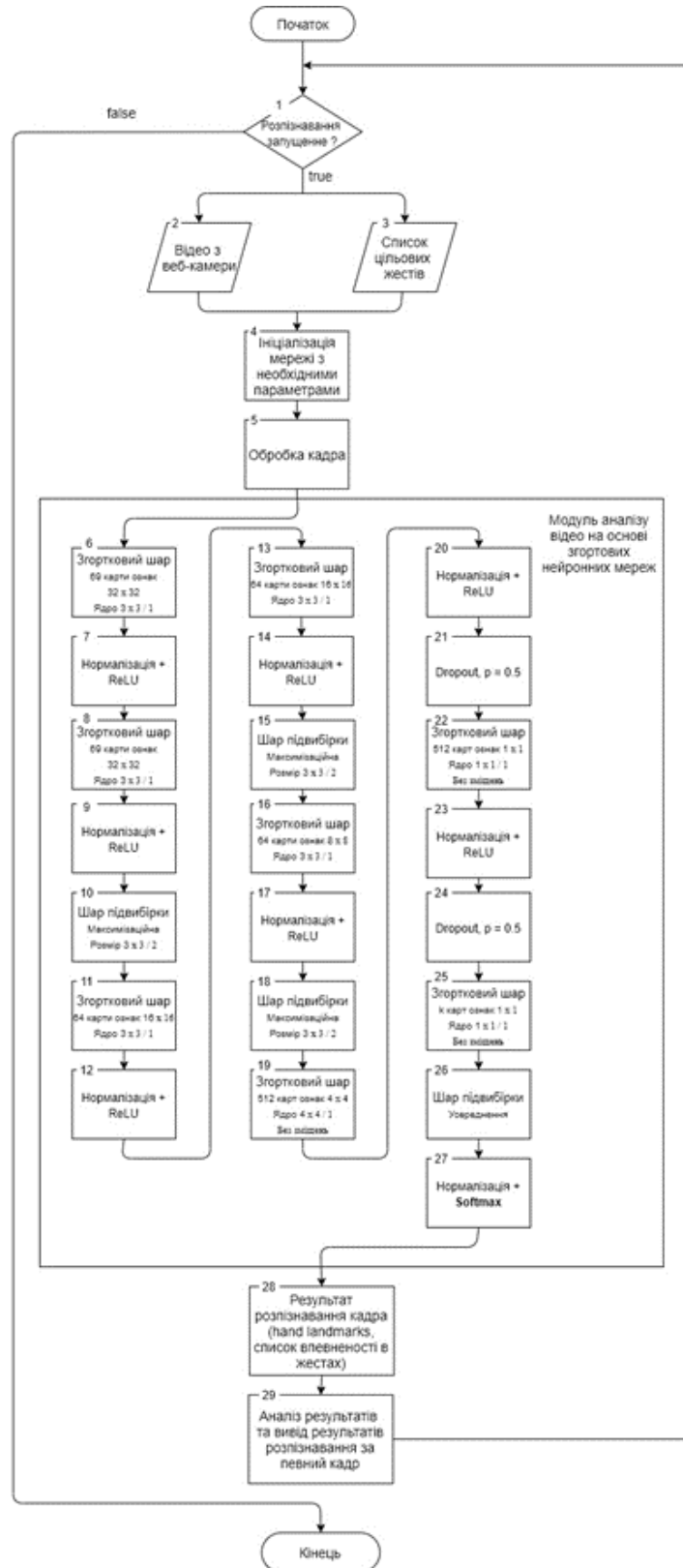


Рисунок В.3 – Загальний алгоритм роботи ядра інформаційної технології керування WEB-додатком на основі згорткових нейронних мереж

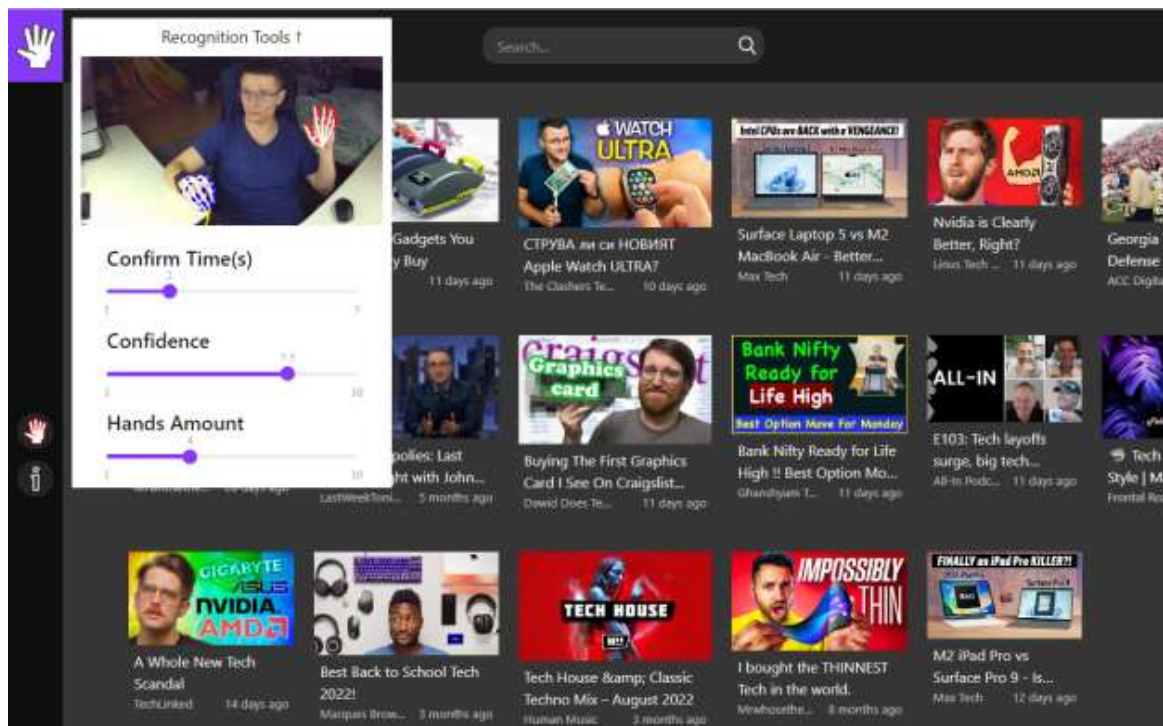


Рисунок В.4 – Робочі вікна програми керування WEB-додатком на основі згорткових нейронних мереж

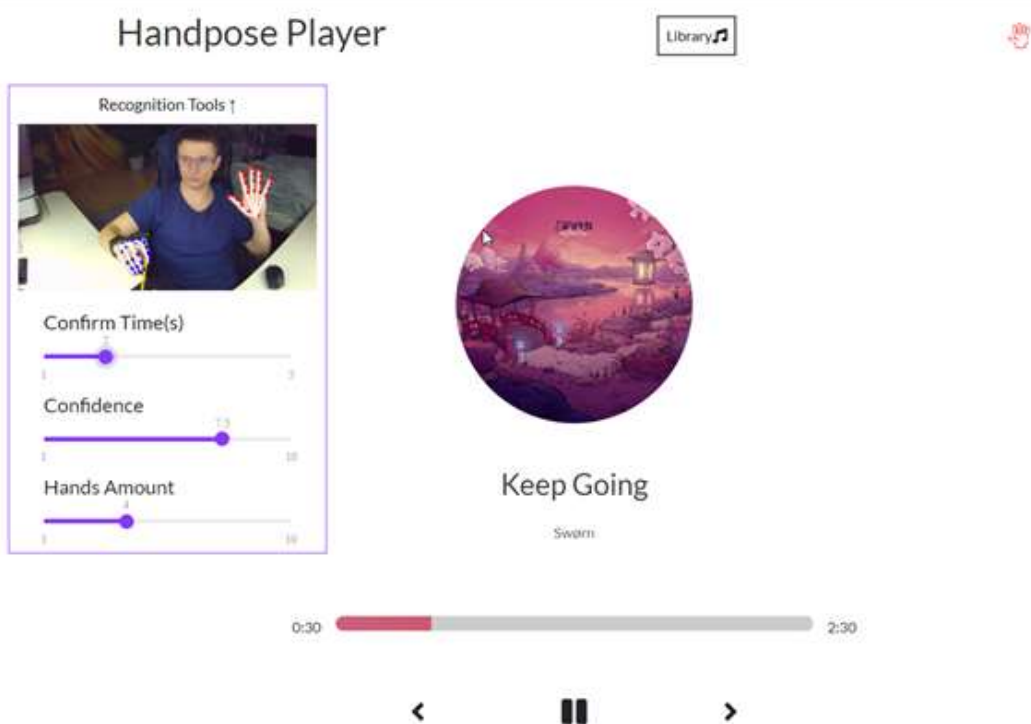
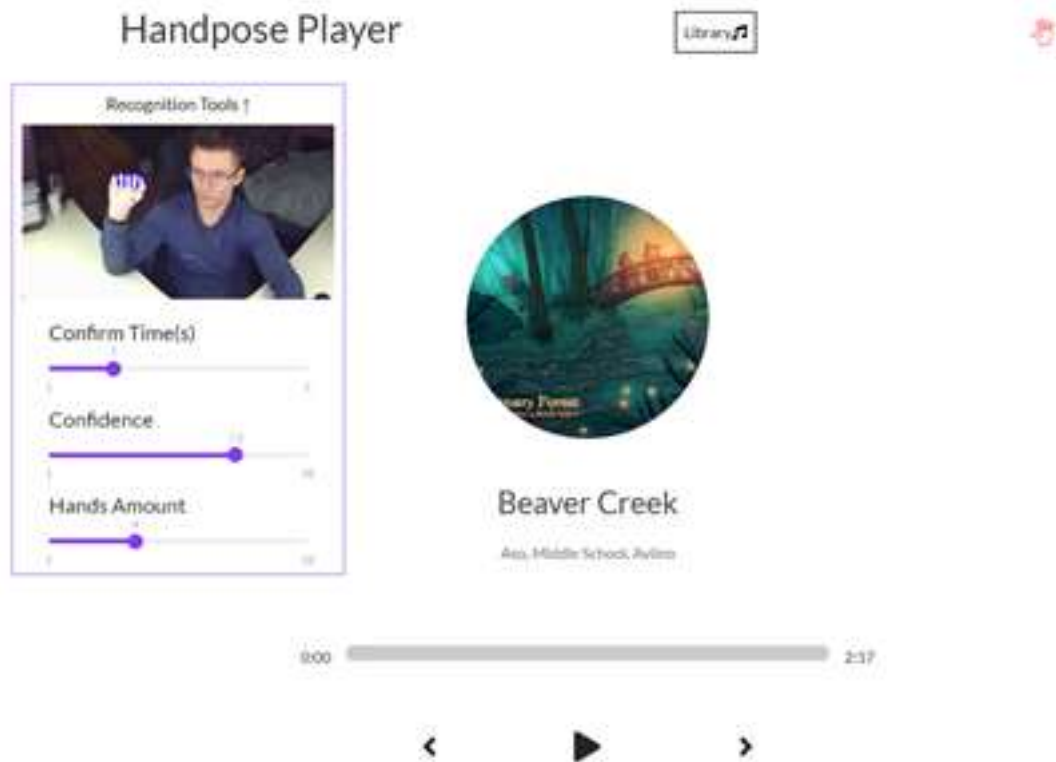


Рисунок В.5 – Робочі вікна програми керування WEB-додатком на основі згорткових нейронних мереж

Реалізація	К-сть жестів розпізнавання	К-сть вірно розпізнаних жестів	Достовірність(%)	К-сть вірно розпізнаних жестів(низька роз. здатність)	Достовірність (%) (низька роз. здатність)	Середнє значення
GestureTek	200	168	84%	140	70%	77%
TableFX	200	165	82.5%	132	66%	74.25%
HandPose	200	175	87.5%	142	71%	79.25%

Рисунок В.6 – Порівняння достовірності розпізнавання аналогів та інтелектуальної системи «HandPose»

Додаток Г (довідниковий) Інструкція користувача

Реалізована програма представляє собою веб-додаток, який розгорнутий у вільному доступі в мережі Інтернет за посиланням <https://handp0se.firebaseio.com>. Перейшовши за посиланням можна спостерігати головну сторінку додатку, представлену на рис. Г.1.

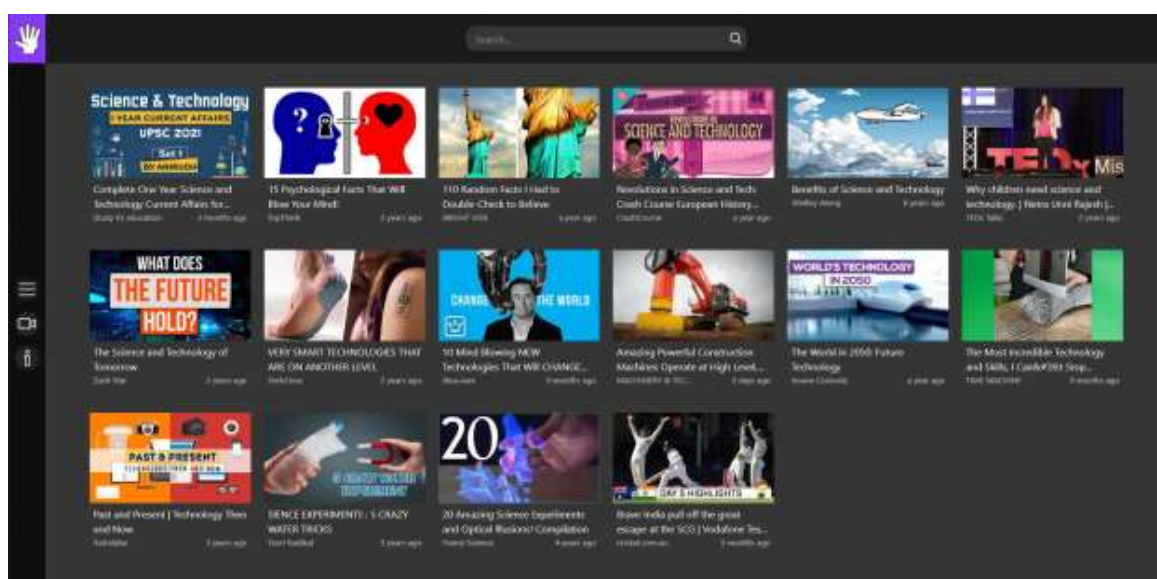


Рисунок Г.1 - Вид стартового вікно програми нейромережевого розпізнавання математичних формул

Додаток побудований на основі відомої відео-платформи Youtube, а саме з неї запозичено макет-шаблон сітки та програвач для відтворення відео. В звичайному режимі все працює, як на Youtube, тобто можна вмикати відео мишкою та керувати переглядом. Однією із функцій є пошук відео за певним ключовим словом або фразою, для цього потрібно скористуватись полем вводу, як показано на рисунку Г.2.



Рисунок Г.2 - Поле пошуку

Для увімкнення розпізнавання жестів необхідно натиснути спеціальну кнопку у вигляді камери, продемонстровану на рисунку Г.3.

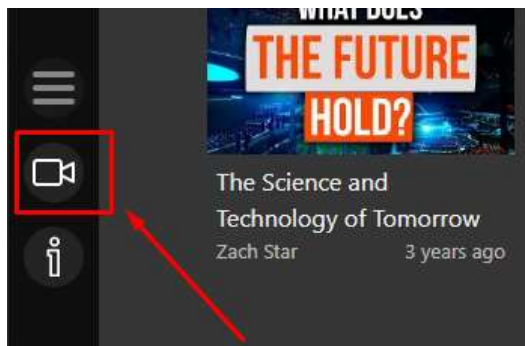


Рисунок Г.3 – Іконка увімкнення режиму розпізнавання

Після увімкнення режиму потрібно зачекати декілька секунд поки завантажатся модуль та загориться індикатор камери червоним кольором, що символізує про розпочатий процес розпізнавання жестів. Щоб розуміти краще, як працює модуль, можна відкрити ліву панель керування. На ній буде зображена трансляція в реальному часі з веб-камери. Також на панелі присутні налаштування для корегування точності та зручності процесу розпізнавання. Тобто можливо установити певну частоту оновлення кадрів, підкорегувати коефіцієнт достовірності або змінити час затримки підтвердження дії, як зображено на рисунку Г.4.



Рисунок Г.4 – Панель налаштувань

Переглянути список жестів можливо наводячи мишкою на іконку інформації, як показано на рисунку Г.5.

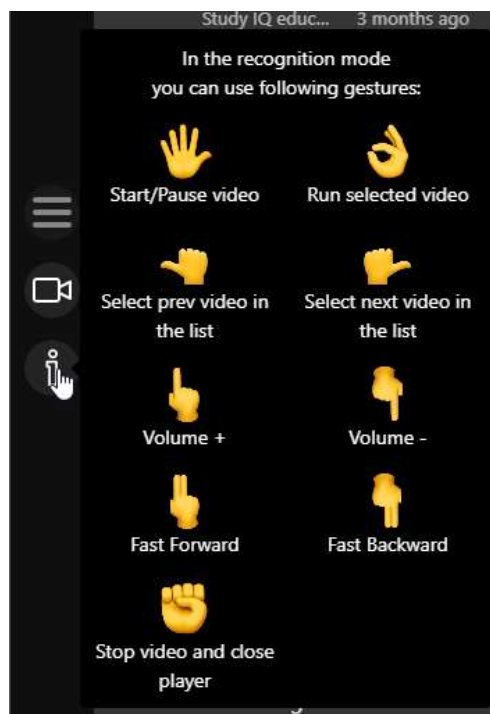


Рисунок Г.5 – Підказка із списком всіх доступних дій

Отже після освоєння описаного функціоналу веб-додатку можна розпочинати використовувати його за призначенням.

Додаток Д (довідниковий)
Довідка про впровадження

ТОВ «ДСофтвер»

Україна, 21009 м. Вінниця, вул. 600-річчя, 17 «Б» тел. 0432 55-03-99

«14» грудня 2022 року

Довідка дана студенту Вінницького національного технічного університету групи ІКН-21м Каліцінському Владиславу Вікторовичу в тому, що результати магістерської кваліфікаційної роботи «Інформаційна технологія жестового керування WEB-додатком на основі згорткових нейронних мереж» пройшли експериментальне дослідження на ТОВ «ДСофтвер». За результатами дослідження програмний продукт планується до впровадження.

Директор
ТОВ «ДСофтвер»



Н.П. Бондар