

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

магістр

(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ
ЗАСОБІВ РОЗПІЗНАВАННЯ ЖЕСТОВОЇ МОВИ
RESEARCH AND SOFTWARE IMPLEMENTATION OF
SIGN LANGUAGE RECOGNITION TOOLS

Виконав(ла): студент(ка) 2 курсу, групи КНТ-214М
Спеціальності 122 Комп'ютерні науки
(код і найменування спеціальності)

Освітня програма (спеціалізація)
Системи штучного інтелекту

ОМ О.Г.

(ПРІЗВИЩЕ та ініціали)

Керівник ІЛЬЯШЕНКО М.Б.

(ПРІЗВИЩЕ та ініціали)

Рецензент КОЦУР М.І.

(ПРІЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет КНТ
Кафедра програмних засобів
Ступінь вищої освіти магістр
Спеціальність 122 Комп'ютерні науки
(код і найменування)

Освітня програма (спеціалізація) Системи штучного інтелекту
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
“ ” _____ 2025 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

ОМА Олександра Германовича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Дослідження та програмна реалізація засобів розпізнавання жестової мови. Research and Software Implementation of Sign Language Recognition Tools

керівник проєкту (роботи) к.т.н., доцент, ІЛЬЯШЕНКО Матвій Борисович,
(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “30” вересня 2025 року № 447

2. Строк подання студентом проєкту (роботи) 02 грудня 2025 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Матеріали і методи. 3. Опис програми. 4. Експлуатація, тестування та експериментальне дослідження програми.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів) _____

Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	ІЛЪЯШЕНКО М.Б., доцент		
Нормоконтроль	ДЕЙНЕГА Л.Ю., ст. викладач		

7. Дата видачі завдання “30” вересня 2025 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій розробки.	2 тиждень	Розділ 2
4	Розробка структури програми.	3 тиждень	Розділ 3
5	Розробка програми.	4-7 тижні	Розділи 3,4
6	Тестування та експериментальне дослідження програмного забезпечення.	8 тиждень	Розділ 4
7	Оформлення пояснювальної записки та документів до неї.	9-10 тижні	Додатки
8	Нормоконтроль та рецензування.	11 тиждень	
9	Захист роботи.	12 тиждень	

Студент(ка)

_____ Олександр ОМ
(підпис) (Імя ПРИЗВИЩЕ)

Керівник проєкту (роботи)

_____ Матвій ІЛЪЯШЕНКО
(підпис) (Імя ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи магістра:
80 с., 3 табл., 28 рис., 3 дод., 14 джерел.

PYTHON, SUBLIME TEXT, ЖЕСТОВА МОВА, МОВА
ПРОГРАМУВАННЯ, РОЗПІЗНАВАННЯ ЖЕСТІВ.

Об'єкт дослідження – процеси обчислень, пов'язані з розпізнаванням мови жестів.

Предмет дослідження – засоби для розпізнавання жестової мови.

Мета роботи – дослідження та розробка програмного забезпечення для розпізнавання жестової мови.

Матеріали, методи та технічні засоби: мова програмування Python, середовище розробки Sublime Text.

Результати. Розроблено програмне забезпечення для розпізнавання жестової мови за допомогою мови програмування Python та середовища розробки Sublime Text.

Практична цінність роботи полягає у розробці програмного забезпечення для розпізнавання жестової мови.

Висновки. Виконано проектування програмного забезпечення для розпізнавання жестової мови. Розроблено програмне забезпечення для розпізнавання жестової мови. Здійснено тестування розробленого програмного забезпечення для розпізнавання жестової мови.

Галузь використання – розпізнавання жестів.

ABSTRACT

Explanatory note to the diploma qualifying work of the master: 80 pages, 3 tables, 28 figures, 3 appendixes, 14 sources.

PYTHON, SUBLIME TEXT, SIGN LANGUAGE, PROGRAMMING LANGUAGE, GESTURE RECOGNITION.

The object of research is the computational processes associated with sign language recognition.

The subject of the research is tools for sign language recognition.

The purpose of this work is to develop the software for sign language recognition.

Materials, methods and technical tools: Python programming language, Sublime Text development environment.

Results. The software tools for sign language recognition using the Python programming language and the Sublime Text development environment have been developed.

The practical value of the work lies in the development of software for sign language recognition.

Conclusions. The software for sign language recognition has been designed. The software for sign language recognition has been developed. The developed software for sign language recognition has been tested.

Scope of use – sign language recognition.

ЗМІСТ

	С.
Перелік скорочень та умовних позначок	8
Вступ.....	9
1 Аналіз предметної області.....	11
1.1 Розпізнавання жестової мови	11
1.2 Аналіз програмного забезпечення для розпізнавання жестової мови	20
1.3 Висновки за розділом 1	29
2 Матеріали і методи	31
2.1 Вибір мови програмування	31
2.2 Вибір середовища розробки для створення програмного забезпечення для розпізнавання жестової мови.....	34
2.3 Висновки за розділом 2	37
3 Опис програми	38
3.1 Структура програмного забезпечення для розпізнавання жестової мови	38
3.2 Функціонування програмного забезпечення для розпізнавання жестової мови	40
3.3 Розробка нейромережевої моделі для розпізнавання жестової мови.....	42
3.4 Навчання нейромережевої моделі для розпізнавання жестової мови.....	45
3.5 Висновки за розділом 3	48
4 Експлуатація, тестування та експериментальне дослідження програми.....	49
4.1 Призначення й умови застосування програми	49
4.2 Характеристики програми для розпізнавання жестової мови	50
4.3 Інструкція по експлуатації програми.....	51
4.3.1 Звернення до програми	51
4.3.2 Вхідні й вихідні дані	51
4.3.3 Повідомлення.....	51
4.4 Виконання програмного забезпечення для розпізнавання жестової мови	52

4.5 Тестування програмного забезпечення для розпізнавання жестової мови	56
4.6 Висновки за розділом 4	56
Висновки	57
Перелік джерел посилання	60
Додаток А Технічне завдання	62
Додаток Б Фрагмент тексту програми	66
Додаток В Слайди презентації	74

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

API – Application Programming Interface;

ЖМ – жестова мова;

НМ – нейронна мережа;

ПЗ – програмне забезпечення;

ШІ – штучний інтелект.

ВСТУП

Методи та засоби для розпізнавання жестової мови є надзвичайно актуальними, оскільки забезпечують можливість подолання комунікаційних бар'єрів між людьми з порушеннями слуху та ширшою спільнотою. В умовах зростаючого впровадження цифрових технологій і штучного інтелекту підвищується потреба в програмних системах, здатних точно та оперативно інтерпретувати жести в реальному часі. Це стає критично важливим не лише для забезпечення соціальної інклюзії, але й для інтеграції людей із обмеженими можливостями у сферу освіти, праці та повсякденного життя [1], [2].

Використання методів комп'ютерного зору та штучного інтелекту дало змогу досягти істотного підвищення точності інтерпретації жестів і забезпечило можливість функціонування відповідних програмних засобів у режимі реального часу. Наукові дослідження поступово переходять від експериментальних моделей до комплексних архітектур, здатних інтегрувати кілька каналів даних, таких як зображення, просторово-часові характеристики чи сигнали від сенсорних пристроїв. Це відкриває перспективи для створення універсальних програмних систем, які здатні адаптуватися до різних жестових мов і умов використання [3], [4].

Програмне забезпечення (ПЗ) для розпізнавання жестової мови дедалі частіше застосовується як засіб комунікації між людьми з порушеннями слуху, що сприяє підвищенню рівня інклюзії. Програмні засоби у вигляді мобільних застосунків та апаратних пристроїв уже демонструють реальне використання в різних сферах суспільного життя, включаючи освіту, транспорт, сферу обслуговування та державні послуги [5], [6].

Проте деякі програми для розпізнавання жестової мови висувають високі вимоги до якості вхідних даних. Крім того, на ефективність розпізнавання за допомогою таких програмних систем суттєво впливає індивідуальна різниця у виконанні жестів, швидкість рухів та фізіологічні

особливості користувачів, що створює складності при навчанні та подальшому використанні розпізнавальних моделей у ПЗ. Ще одним недоліком таких програмних засобів є обмеженість мовного та культурного покриття. Більшість сучасних програмних систем орієнтовані на відносно поширені жестові мови, у той час як менш вивчені або регіональні мови залишаються недостатньо представленими в базах даних, що обмежує їх практичне застосування. Суттєвим викликом є також висока обчислювальна складність алгоритмів, що ускладнює реалізацію таких систем у мобільних пристроях або у середовищах з обмеженими ресурсами [1]-[6].

Тому актуальною є розробка програмного забезпечення для розпізнавання жестової мови. У дипломній кваліфікаційній роботі магістра розв'язується актуальне завдання розробки та дослідження програмного забезпечення для розпізнавання жестової мови.

Для досягнення поставленої мети у кваліфікаційній роботі магістра необхідно розв'язати такі задачі:

- виконати аналіз предметної області та програмних засобів для розпізнавання жестової мови;
- розробити нейромережеву модель для розпізнавання жестової мови;
- здійснити проектування програмного забезпечення для розпізнавання жестової мови;
- створити програмне забезпечення для розпізнавання жестової мови;
- виконати тестування розробленого програмного забезпечення для розпізнавання жестової мови.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Розпізнавання жестової мови

Жестова мова розглядається як самостійна форма комунікації, що має ключове значення для багатьох людей у суспільстві. Вона формується завдяки поєднанню рухів рук, положень тіла, міміки та інших невербальних ознак, які разом створюють знаки зі змістовим навантаженням. Саме через таку складність побудови висловлювань завдання автоматичного розпізнавання жестів належить до найскладніших напрямів комп'ютерного зору. Протягом останніх років у цій сфері з'явилися численні підходи, у тому числі на основі глибинних нейронних мереж, що значно підвищили ефективність розпізнавання та сприяли формуванню систем з високим рівнем точності. Водночас низка проблем залишається невирішеною, зокрема питання відсутності достатньо великих та якісно анотованих баз даних, складність обробки безперервних висловлювань і потреба у створенні універсальних алгоритмів для різних жестових мов [1].

Сучасні програмні засоби та технології штучного інтелекту відкривають можливості для усунення бар'єрів у спілкуванні між людьми з порушеннями слуху та іншими соціальними групами. Використання вдосконалених сенсорних пристроїв у поєднанні з алгоритмами машинного навчання створює передумови для появи програм, здатних автоматично перекладати жести у текстову чи звукову форму. Дослідження у цьому напрямку спрямовані на створення систем, які можуть не лише розпізнавати ізольовані знаки, але й аналізувати їх у контексті повноцінних висловлювань. Такі розробки сприяють інтеграції людей з вадами слуху у суспільство та відкривають перспективи для більш широкого використання технологій у навчанні, професійній діяльності та повсякденному спілкуванні [1].

Розробка подібних програмних систем вимагає комплексного підходу, що охоплює етапи збору даних, створення моделей для розпізнавання та розробки інтерфейсів для зручної взаємодії. Точність таких рішень залежить

не лише від архітектури нейромереж, а й від можливостей сенсорного обладнання, яке використовується для фіксації рухів. Проблеми із захопленням швидких або перекритих рухів залишаються суттєвим викликом, що зумовлює потребу у створенні більш витривалих методів обробки сигналів. Незважаючи на це, тенденція розвитку свідчить про поступове зростання ефективності систем розпізнавання жестової мови, що відкриває шлях до широкого застосування технологій у різних сферах [1].

Переклад жестової мови у відповідному програмному забезпеченні охоплює процеси як між різними жестовими мовами, так і між жестовою та усною мовами. У таких системах застосовуються методи обробки послідовностей у рамках машинного навчання, що дає змогу зменшити бар'єри у спілкуванні між людьми, які користуються жестами, та носіями розмовних мов. Основні труднощі виникають через відсутність достатніх багатомовних корпусів, а також через похибки, що накопичуються на етапі автоматичного розпізнавання глос, адже саме ця стадія виступає основою для подальшого перекладу. Ще одним важливим напрямом є відтворення жестів, що здійснюється за допомогою віртуальних аватарів чи відеогенерації. Попри це, створювані моделі ще не мають належної природності рухів і часто сприймаються нечітко, оскільки точність фіксації поз та деталізація міміки залишаються обмеженими [1], [2].

Усі технології, пов'язані з опрацюванням жестів, перебувають у взаємозалежності. Від того, наскільки якісно захоплюється інформація про положення рук, тіла та вирази обличчя, залежить результативність алгоритмів, які здійснюють класифікацію та розпізнавання. А вже точність розпізнавання, у свою чергу, визначає рівень ефективності перекладу та достовірність подальшого відтворення жестів. Прогрес у сфері сенсорних технологій разом із швидким розвитком штучного інтелекту відкриває нові можливості для створення систем, здатних полегшити комунікацію людям із порушеннями слуху та зробити її більш природною у повсякденному житті [1], [2].

Процес запису жестової мови спрямований на те, щоб за допомогою спеціального обладнання та програмних засобів зібрати дані, придатні для подальшої обробки. Зібрані матеріали служать основою для аналізу, навчання алгоритмів, формування наборів даних та створення тривимірних моделей відображення жестів. Використання відеопотоків та датчиків різних типів дозволяє формувати багатокomпонентні бази, що охоплюють як ізольовані знаки, так і безперервні висловлювання [1], [2].

У цьому контексті активно застосовуються різноманітні сенсорні системи з використанням програмного забезпечення. Камери глибинного бачення здатні одночасно реєструвати кольорове зображення, просторову структуру та скелетні параметри. Такі підходи дають змогу отримувати мультимодальні записи жестової мови з високим рівнем деталізації. Удосконалені датчики, наприклад ті, що відстежують просторове положення кистей та пальців у тривимірному просторі, забезпечують більш точне захоплення навіть без фізичного контакту з пристроєм. Поряд із цим досліджуються й альтернативні методи, наприклад використання сигналів радіочастотної ідентифікації, де фіксується зміна хвильових характеристик під час виконання жестів. Такі експерименти підтверджують, що спектр технічних рішень у сфері захоплення та розпізнавання жестів є дуже широким, і кожен підхід має власні переваги та обмеження, що впливають на якість кінцевої системи [1], [2].

Ефективність програмного забезпечення для роботи з жестовою мовою значною мірою залежить від наявності якісних наборів даних. Від того, наскільки точно було зафіксовано рухи та виконано анотацію, безпосередньо залежить успіх алгоритмів розпізнавання, перекладу й синтезу. Найбільш поширеним джерелом інформації для формування таких корпусів виступають візуальні сенсори, які забезпечують детальне відтворення жестів у відеопослідовностях. Саме вони становлять основу переважної більшості відкритих ресурсів. Особливе значення мають набори, орієнтовані на безперервне розпізнавання, оскільки саме вони дають змогу створювати

системи, близькі до реальних сценаріїв комунікації. Відомим прикладом є Phoenix-2014, у якому представлено записи німецької жестової мови з тематикою прогнозів погоди, що було отримано від кількох мовців із використанням високої частоти кадрування [1], [2].

Завдання автоматичного розпізнавання жестової мови з відеопотоків розглядається як важливий напрям досліджень у сфері розробки програмного забезпечення для комп'ютерного зору. Його значення полягає у можливості подолання мовного бар'єру між людьми з порушеннями слуху та тими, хто користується усною мовою. У практиці прийнято виокремлювати два основних підходи: ізольоване розпізнавання, коли алгоритм працює з окремими глосами, та безперервне, що орієнтоване на класифікацію повних висловлювань [2], [3].

Безперервне розпізнавання жестів вважається більш складним, оскільки у цьому випадку модель повинна виявляти значення без наявності явних анотацій, які позначають початок і кінець глоси. Для обробки відео послідовностей зазвичай застосовуються архітектури на основі дво- чи тривимірних згорткових мереж, які вилучають просторові ознаки, після чого до справи долучаються рекурентні мережі або часові згортки, що відповідають за моделювання динаміки рухів. Для оцінки ефективності моделей традиційно використовується метрика, яка відображає, скільки змін необхідно внести до прогнозованої послідовності, щоб вона відповідала еталонній [2], [3].

Останнім часом спостерігається активне впровадження механізмів самоуваги, зокрема архітектур на основі Transformer, у програмне забезпечення для безперервного розпізнавання. У дослідженнях було запропоновано комбінування кількох джерел даних, наприклад окремих зображень рук та повних кадрів, що оброблялися незалежними CNN-модулями, а згодом синхронізувалися за допомогою механізму самоуваги, що забезпечувало точніший контекст. Також застосовуються модифікації CNN-шарів у поєднанні з самоувагою, що дозволяє підвищити ефективність початкового вилучення ознак [2], [3].

Поряд із цим з'являються спроби використання методів навчання з підкріпленням у поєднанні з трансформерними мережами. Такий підхід передбачає оптимізацію моделей безпосередньо відносно цільової метрики, використовуючи її як функцію винагороди. Експерименти продемонстрували, що поєднання тривимірних згорткових мереж із механізмом Transformer дозволяє ефективно обробляти довгі послідовності жестів, тоді як алгоритми підкріплення допомагають точніше визначати часові межі глос. Основною перешкодою у цьому випадку залишається необхідність обмеження простору дій агента, що може впливати на результативність моделі [3], [4].

Розпізнавання ізольованих жестів передбачає визначення окремих одиниць жестової мови на основі відеопотоків. Це завдання за своєю природою близьке до проблем класифікації дій чи рухових патернів, де ключову роль відіграє вилучення інформативних ознак та їх подальше навчання моделлю. Найчастіше робота з такими даними починається з відсікання другорядних ділянок кадру, зокрема шумного фону, щоб зосередити увагу на руках і артикуляції губ. Такий підхід значно полегшує аналіз і підвищує точність розпізнавання [3], [4].

Задля збільшення продуктивності програмних засобів для розпізнавання жестових символів було запропоновано розширювати набір характеристик шляхом включення динамічних параметрів, таких як оптичний потік, або використання даних про положення суглобів. У таких багатопотокових архітектурах відбувається інтеграція кількох типів сигналів, що дозволяє моделі краще справлятися з неоднозначними ситуаціями, хоча вартість обчислень у цьому випадку зростає. Ефективність таких підходів доведено на прикладах двопотокових мереж, де вхідними даними виступали як звичайні відео, так і карти руху, отримані за допомогою оптичного потоку [3], [4].

Суттєвий вплив на розвиток програмних систем розпізнавання жестів мали новітні досягнення у сфері глибинного навчання. У сучасних моделях все частіше використовуються механізми уваги, графові згорткові мережі та

трансформерні архітектури. Завдяки їм стає можливим фокусування на найбільш релевантних елементах відео, що безпосередньо впливає на якість класифікації [3], [4].

Паралельно відзначається інтерес до застосування сенсорів, які спочатку активно впроваджувалися у завданнях розпізнавання дій. Їх використання у жестовій мові забезпечує мультимодальний підхід, поєднуючи колірну та глибинну інформацію. Проте ефективність таких систем обмежується відсутністю масштабних відкритих корпусів даних та тим, що карти глибини нерідко містять похибки [3], [4].

Завдання перекладу жестової мови орієнтується на відтворення змісту відеопослідовності у вигляді тексту чи усної мови. Його складність полягає у необхідності врахування граматичних правил і структурних особливостей мов, на які здійснюється переклад. У таких дослідженнях традиційно використовується метрика, що оцінює точність результату шляхом порівняння з еталонними варіантами перекладу. Розвиток напрямку значною мірою пов'язаний із застосуванням архітектур Transformer, які довели свою ефективність у природньомовних завданнях [3], [4].

Окрему нішу становить проблема представлення жестової мови. Створення зрозумілих і природних візуальних форм є обов'язковою умовою для систем, що покликані забезпечувати зручність користувачів з порушеннями слуху. Найбільш поширеним підходом вважається використання 3D-аватарів, здатних демонструвати рухи тіла та міміку. Проте завдання генерації таких відтворень пов'язане зі складністю моделювання швидких та дрібних рухів рук. У наукових роботах були запропоновані різні варіанти, зокрема інтеграція LSTM-мереж для перекладу відео у текст і подальший синтез жестів або використання словникових баз для створення жестових рухів в анімаційних редакторах. Водночас недоліками таких рішень залишаються відсутність багатой виразності та обмеженість у розширенні на великі корпуси даних [3], [4].

Програмні засоби для синтезу жестової мови останнім часом привертають дедалі більше уваги завдяки активному розвитку технологій глибинного навчання. Використання сучасних моделей дозволяє формувати відеоряд, у якому рухи та міміка виглядають значно природніше, ніж у випадку з класичними аватарами. У таких системах зазвичай використовуються як вихідні дані послідовності глос та опорне зображення людини, на основі яких створюється відео, де відтворюються відповідні жести з високим рівнем реалістичності. Це дає можливість уникати неприродної жорсткості рухів, властивої синтетичним моделям [4], [5].

Паралельно з цим активно розвивається практичне застосування програмних систем роботи з жестовою мовою. Розвиток методів захоплення, інтерпретації та генерації жестів дав поштовх створенню широкого спектра програмних рішень. Такі системи можуть функціонувати як на персональних комп'ютерах, так і на мобільних пристроях різних платформ, що робить їх доступними для широкого кола користувачів. Важливим аспектом залишається вибір технічних засобів для збору інформації: дорогі носимі сенсори забезпечують точне фіксування рухів рук, але накладають обмеження на користувача, тоді як камери мобільних телефонів або вебкамери пропонують дешевший варіант і дозволяють додатково отримувати дані про вираз обличчя та загальну позу [4], [5].

У завданнях безперервного розпізнавання жестової мови переважає використання двовимірних згорткових мереж у поєднанні з часовими або рекурентними архітектурами, що забезпечує прийнятний баланс між точністю й обчислювальними витратами. При цьому моделі з тривимірними згортками потребують значно більших ресурсів. Експериментальні дані свідчать, що інтеграція кількох модальностей, зокрема інформації про оптичний потік або положення тіла, дозволяє досягати більшої точності у порівнянні з використанням лише одного джерела даних. Попри це, результати на наборах даних з великими словниками чи невідомими словами все ще залишаються далекими від оптимальних. Методи розпізнавання ізольованих жестів

демонструють високу ефективність на спеціалізованих корпусах, однак їх застосування у реальних умовах обмежене [4], [5].

Дослідження у сфері перекладу жестової мови лише набирають обертів, хоча вже показують багатообіцяючі результати. Основою таких підходів стають технології нейронного машинного перекладу, які дозволяють трансформувати відеопослідовність у текстові або усні аналоги. Така робота має важливе соціальне значення, оскільки забезпечує додаткові можливості для комунікації між людьми з порушеннями слуху та іншими спільнотами. Особливий напрям розвитку становить представлення жестової мови. Традиційні рішення із застосуванням тривимірних анімованих моделей вимагають трудомісткого проектування й не завжди здатні відтворити дрібні рухи з належною плавністю. Створення реалістичної анімації без використання складних систем захоплення руху практично неможливе. Водночас сучасні підходи на основі глибинного навчання дозволяють автоматично синтезувати відео із залученням зображень реальних людей. Такі результати оцінюються значно вище з боку спільноти глухих, ніж статичні аватари [4], [5].

Сучасні програмні засоби у сфері розпізнавання жестової мови часто інтегруються у мобільні платформи, де виконують переклад або розпізнавання жестів у реальному часі. Окрему увагу привертають навчальні програми, орієнтовані на користувачів, які лише починають знайомитися з жестовою мовою. Подальший розвиток цієї галузі пов'язаний із потребою створювати моделі, здатні працювати на обмежених ресурсах, а також із вдосконаленням методів інтеграції таких моделей у портативні пристрої для масового використання [4], [5].

Програмне забезпечення для розпізнавання жестової мови являє собою комплекс засобів, здатних інтерпретувати рухи рук, положення пальців і міміку обличчя з метою перетворення їх у зрозумілий для користувача текстовий або звуковий формат. Його створення ґрунтується на поєднанні

комп'ютерного зору, машинного навчання та технологій обробки сигналів, що дає змогу автоматизувати процес перекладу жестів [5], [6].

Розробка таких програмних засобів починається з формування якісної бази даних, що включає відео або сенсорні записи жестів відповідної мови. Зібрані дані підлягають ретельному маркуванню, адже саме від цього залежить здатність алгоритму коректно розрізняти схожі рухи. Далі застосовуються методи попередньої обробки: усуваються шуми, нормалізуються зображення, виділяються ключові точки тіла, які формують структуровані вектори ознак [5], [6].

На наступному етапі розробки програмного забезпечення для розпізнавання жестової мови будується модель машинного навчання, що має здатність аналізувати динаміку рухів у часі. Для цього використовуються нейронні мережі, які враховують як просторові, так і часові характеристики. Модель тренується на великій кількості прикладів, після чого відбувається її тестування та оптимізація для підвищення точності [5], [6].

Фінальною частиною процесу розробки програмного забезпечення для розпізнавання жестової мови є інтеграція алгоритмів у зручний інтерфейс. Користувач програмного забезпечення для розпізнавання жестової мови отримує змогу взаємодіяти з програмою в режимі реального часу, а програмна система, сприймаючи жести через камеру або сенсорні пристрої, миттєво відтворює їх у текстовому чи голосовому форматі. Такий підхід забезпечує не лише технічну функціональність, а й практичну цінність у повсякденному спілкуванні [5], [6].

Визначено, що засоби для розпізнавання жестової мови являють собою комплекс інструментів, здатних інтерпретувати рухи рук та положення пальців з метою перетворення їх у зрозумілий для користувача текстовий або звуковий формат. Створення таких засобів ґрунтується на поєднанні інструментів комп'ютерного зору, машинного навчання та технологій обробки сигналів, що дає змогу автоматизувати процес перекладу жестів.

1.2 Аналіз програмного забезпечення для розпізнавання жестової мови

Проаналізуємо програмне забезпечення для розпізнавання жестової мови [7]-[9].

Програмне забезпечення SignLive (рис. 1.1) створено для живої або віртуальної трансляції, коли аватар жестової мови з'являється на екрані під час подій, забезпечуючи автоматичний переклад жестів у реальному часі без участі людини перекладача. Програма функціонує як постійний інтерпретатор, інтегрований у програмне забезпечення для трансляцій, й надає можливість миттєвого відображення жестових повідомлень для глядачів. Серед особливостей SignLive варто відзначити його здатність працювати у контексті онлайн зустрічей – підключення до платформ на кшталт Zoom, Google Meet чи Microsoft Teams відкриває можливість двостороннього жестового спілкування через аватар. Така інтеграція забезпечує безшовний досвід взаємодії для людей з порушенням слуху [7].

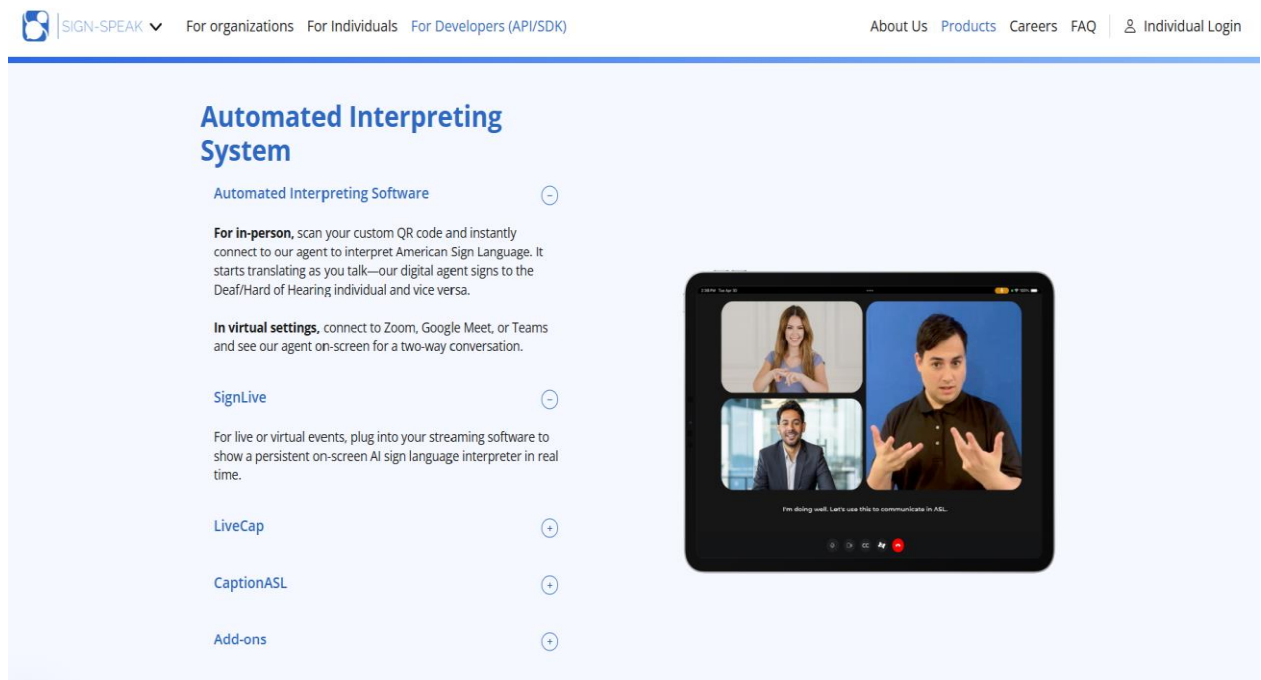


Рисунок 1.1 – Програмне забезпечення SignLive [7]

Програмна реалізація SignLive базується на інтелектуальному інтерпретаторі жестової мови, який використовує інтерфейс прикладного програмування (Application Programming Interface, API), що дозволяє вбудовувати функції розпізнавання жестів та генерації жестового аватара прямо в продукт користувача. Завдяки цим інтерфейсам можна передавати відео у відповідному форматі та отримувати візуальну інтерпретацію у вигляді жестів або тексту [7].

Розробка SignLive здійснювалася у тісній взаємодії з людьми, які користуються жестовою мовою, а також націлена на забезпечення доступності, інклюзивності і справжнього двостороннього спілкування. Крім того, вся архітектура побудована з урахуванням використання хмарної інфраструктури, що забезпечує масштабованість, ефективність обчислень та простоту розгортання системи. Таким чином, SignLive виступає як інтегроване, AI-кероване середовище для автоматичного перетворення жестів у візуального аватара на екрані, що забезпечує доступність заходів і спілкування для глухих і людей з вадами слуху. Воно поєднує плавну інтеграцію зі стрімінговими або конференцплатформами, AI-аналіз жестів і хмарну архітектуру із фокусом на співпраці зі спільнотою [7].

Основними характеристиками програмного забезпечення SignLive є такі [7]:

- своєчасна відеоінтерпретація: програмне забезпечення SignLive забезпечує можливість миттєвого підключення до живого перекладача жестової мови під час відеодзвінків, без потреби попереднього запису чи бронювання, що робить взаємодію максимально оперативною для користувачів [7];

- універсальність доступу: програмне забезпечення SignLive дозволяє використовувати сервіс як через мобільні додатки, так і у браузері, що створює комфортні умови для широкого кола користувачів, включаючи тих, хто не має спеціального обладнання [7];

- підтримка відеорелейної служби: програмне забезпечення SignLive працює на базі технологій Video Relay Service і Video Remote Interpreting, що дозволяє забезпечити якісний переклад між глухими користувачами та співробітниками організацій через інтернет-дзвінки [7];

- пріоритет безпеці та конфіденційності: практики програмного забезпечення SignLive дотримуються високих стандартів захисту даних, забезпечуючи шифрування та відповідність регламентам конфіденційності [7];

- застосування у медичних та публічних сервісах: програмне забезпечення SignLive визнається корисним у ключових галузях, таких як охорона здоров'я, дозволяючи глухим користувачам отримувати доступ до консультацій та послуг без бар'єрів [7];

- партнерські інтеграції: програмне забезпечення SignLive застосовується у співпраці з організаціями, зокрема авіакомпаніями, для забезпечення доступного обслуговування клієнтів із порушеннями слуху, з можливістю звернення через вебсайт або додаток [7].

Серед недоліків програмного забезпечення SignLive можна відзначити обмежену доступність у деяких регіонах, що ускладнює його використання для міжнародних користувачів; залежність від стабільного Інтернет з'єднання, без якого якість відеозв'язку значно знижується; можливі труднощі з інтеграцією у внутрішні системи організацій; відсутність широкого вибору мов жестів, що може обмежувати комунікацію для користувачів з різних країн; а також потребу у певному навчанні для нових користувачів, які вперше стикаються з таким типом сервісу [7].

Програмне забезпечення Nuskel (рис. 1.2) створено з метою автоматичного розпізнавання окремих жестів, що позначають букви чи цифри, на статичних зображеннях. Досягати цього дозволяє інтелектуальна модель, яка аналізує кадр, де видно руку, і визначає, яка саме літера або цифра зображена, надаючи при цьому рівень впевненості у класифікації. Така модель вбудовується через інтерфейс прикладного програмування, що дає змогу інтегрувати цей функціонал у готові додатки, сервіси або сайт без

необхідності розробки власних нейромереж, достатньо передати зображення у відповідному форматі, а сервіс поверне результат класифікації [8].

Nysckel CUSTOMERS DOCS INDUSTRIES RESOURCES LOG-IN BOOK DEMO GET STARTED

Identify sign language using AI

Below is a free classifier to identify sign language. Just upload your image, and our AI will predict which letter or number it is - in just seconds.

Drop image or click to select
JPG, PNG, BMP, or WEBP

Рисунок 1.2 – Програмне забезпечення Nysckel [8]

Значущою особливістю даного сервісу є його орієнтація на роботу з жестами, які представляють окремі символи — букви чи цифри — без підтримки цілих фраз або динаміки рухів. У роботі використовувалися готові попередньо навчені моделі, що охоплюють близько трьох десятків класів жестів. Однією з переваг вважається можливість миттєвого отримання результатів із показом коефіцієнта впевненості, що суттєво спрощує інтеграцію розпізнавання у навчальні платформи, системи доступності, онлайн зустрічі, відеомонтаж, дослідницькі продукти чи служби екстреного інформування [8].

З точки зору програмної реалізації, Nysckel надає простий у використанні інтерфейс, доступний через клієнтські бібліотеки для популярних мов, таких як Python чи JavaScript. Це дає змогу викликати

функції класифікації без глибоких знань у машинному навчанні, винятково через передачу зображення та прийняття відповіді з ймовірнісною оцінкою. Платформа також дозволяє створювати власні класифікатори на основі користувацьких даних із мінімальною конфігурацією й автоматичним донавчанням моделей [8]. Загалом, Nyskel вирізняється як гнучке, швидкодіюче рішення для тих, хто потребує швидке і точне класифікування жестів, орієнтованих на букви чи цифри, з можливістю легко інтегрувати його в існуючий цифровий продукт без складного втручання у модель чи інфраструктуру машинного навчання [8].

Програмне забезпечення Nyskel має такі особливості [8]:

- інтуїтивна інтеграція: програмне забезпечення Nyskel забезпечує просту інтеграцію з підтримкою популярних мов програмування, що дозволяє вмикати функції розпізнавання жестів без глибоких знань у машинному навчанні [8];

- миттєва класифікація: програмне забезпечення Nyskel реагує на статичні зображення з рукою, визначаючи символ жестової мови та повертаючи показник впевненості, що сприяє швидкій обробці кадрів [8];

- ефективний результат: програмне забезпечення Nyskel використовує попередньо навчені моделі, що позбавляє необхідності у власному процесі навчання та пришвидшує застосунок [8];

- модифікована гнучкість: програмне забезпечення Nyskel дозволяє створювати класифікатори на основі власних даних, змінюючи зразки й мітки під специфічні потреби, без складного налаштування [8];

- прозорі відповіді: програмне забезпечення Nyskel повертає не лише результат класифікації, а й рівень впевненості, що допомагає зрозуміти надійність результату при застосуванні у реальному часі [8];

- універсальна доступність: програмне забезпечення Nyskel надає можливість впровадження у навчальні системи, сервіси доступності, відеоплатформи чи інші додатки, розширюючи простір застосування та масштаб [8].

Серед недоліків програмного забезпечення Nuskel можна відзначити обмежену кількість готових інтеграцій із популярними інструментами, що може ускладнити його використання в комплексних бізнес-системах. Також варто враховувати, що сервіс орієнтований переважно на англійськомовних користувачів, тому українськомовна чи багатомовна підтримка наразі є недостатньою. Крім того, при роботі з великими масивами даних продуктивність може знижуватися, що впливає на швидкість обробки запитів. Ще однією слабкою стороною є відносно висока залежність від стабільного Інтернет з'єднання, оскільки більшість функцій реалізовано у хмарному середовищі [8].

Програмне забезпечення GoSign (рис. 1.3) являє собою інструмент, призначений для перетворення жестів у відображення букв або символів, насамперед для роботи із ручним алфавітом жестової мови. Основна мета цього ПЗ полягає у підтримці ідентифікації окремих знаків, що складають повідомлення, шляхом аналізу статичних або ізольованих жестів, що подаються на вхід системі [9].

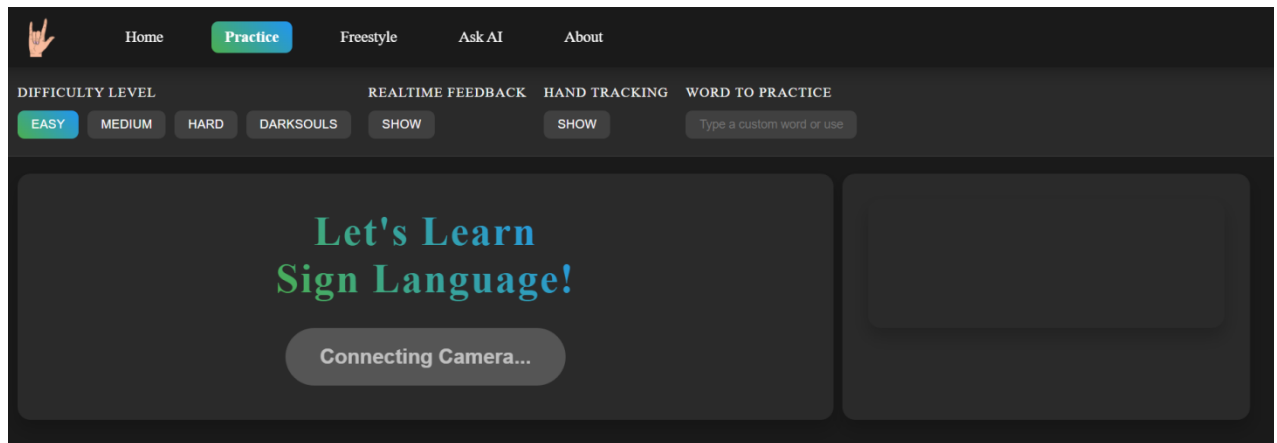


Рисунок 1.3 – Програмне забезпечення GoSign [9]

Програмне забезпечення GoSign вирізняється інтеграцією з технологіями обробки поз і ключових точок, зокрема зручна робота на основі MediaPipe, що дозволяє виділяти ознаки рук. Це створює основу для ефективного розпізнавання форм пальців і позицій руки. Платформа

GoSign.AI, до складу якої належить GoSign Spell, забезпечує збір навчальних даних через ігровий досвід, що дає змогу розширити репрезентативність вибірки жестів та поліпшити якість моделі [9].

Програмна реалізація опирається на легковаговий формат розпізнавання, здатний працювати швидко й з мінімальною затримкою. Оптимізовані моделі, попередньо навчені на даних із різноманітними стилями жестів, фонами й освітленням, розгортаються як ядро інтерактивних додатків. Адаптація моделі під конкретне середовище забезпечується тонким налаштуванням параметрів тренування, архітектури нейронної мережі та форматів експорту. GoSign Spell зосереджується на простоті використання й швидкості реагування — його реалізація дозволяє запускати класифікацію жестів на пристроях без потужних обчислювальних ресурсів, що робить її придатною для мобільних ігрових застосунків, освітніх інструментів або низькорівневих платформ взаємодії [9].

Система також акцентує увагу на інклюзивності: збір даних із різним освітленням, кольором шкіри та фонами гарантує, що модель залишатиметься стійкою в реальних умовах. Усе це спрямовано на створення інтерфейсів, інтуїтивно призначених для користувачів жестової мови, а не лише адаптованих під них, щось подібне до інтерфейсів для жестової комунікації. Загалом, GoSign Spell спроектовано як легший, але ефективний компонент екосистеми GoSign.AI, що фокусується на виявленні та інтерпретації алфавітних жестів із потужною підтримкою через обробку і оптимізацію моделей [9].

Програмне забезпечення GoSign має такі особливості [9]:

- інтерактивне навчання: у програмному забезпеченні GoSign реалізована гейміфікація, яка залучає користувачів до освітнього процесу через ігри на основі розпізнавання жестів і спонукає їх сприяти вдосконаленню моделей шляхом активної участі в діяльності спільноти [9];

- чат-асистент на основі штучного інтелекту: інтегроване середовище, в якому можна взаємодіяти з чатом, інтерпретуючи запити у жестовій мові і

отримуючи візуальні відповіді або відповіді у вигляді тексту, що створює зручний спосіб комунікації без необхідності введення вручну [9];

- аватари жестової мови: розроблена технологія, яка відтворює жестові повідомлення через анімованих персонажів, що дозволяє створювати візуально привабливі і доступні інтерпретації жестів [9];

- глобальна підтримка жестових мов: програмне забезпечення GoSign підтримує охоплення значного числа жестових систем із різних регіонів світу, що робить цей інструмент зручною платформою для міжмовної доступності та інклюзивного навчання [9];

- відкриті ресурси: надання доступу до супровідних датасетів, інструментів і документації, що полегшує спільний розвиток технологій жестової мови і дозволяє дослідникам і розробникам швидше впроваджувати інновації [9];

- медіапайп-інтеграція: використовується система MediaPipe для захоплення поз рук та передачі ключових точок, що забезпечує плавне та точне розпізнавання жестів у реальному часі [9];

- орієнтація на спільноту: команда розробників складається з людей, які особисто використовують жестову мову, завдяки чому платформа розробляється з глибоким розумінням потреб глухих і слабочуючих користувачів [9];

- інтелектуальні сервіси для творчості: окрім навчальних і комунікаційних можливостей, програмне забезпечення GoSign дозволяє перетворювати жести на генеровані мистецькі вирази або створювати контент на їхній основі для творчих застосувань [9];

- масштабована архітектура: використовуються хмарні технології та інтелектуальна інфраструктура, що забезпечує стабільне розгортання, швидкість обробки і можливості для росту у відповідь на потреби користувачів [9].

Серед недоліків програмного забезпечення GoSign можна відзначити обмежений інтерфейс, який може здаватися застарілим і незручним для нових

користувачів, а також можливі труднощі з інтеграцією у складні корпоративні системи. Деякі функції вимагають додаткового налаштування або встановлення сторонніх компонентів, що ускладнює процес використання. Крім того, підтримка різних операційних систем може бути неоднаковою, що призводить до несумісності та зниження продуктивності на окремих платформах [9].

Порівняльну характеристику програмного забезпечення для розпізнавання жестової мови наведено у таблиці 1.1.

Таблиця 1.1 – Порівняння програмного забезпечення для розпізнавання жестової мови

Критерій порівняння	SignLive [7]	Nyckel [8]	GoSign [9]
Легкість інтеграції через API	+–	+	+–
Підтримка розпізнавання у реальному часі	+	+–	+
Орієнтація на широке охоплення жестових мов	+–	–	+
Здатність до настроювання моделей	+–	+	+–
Інтеграція з різними платформами	+	+–	+–
Анімоване відтворення жестів	+	–	+

За результатами проведеного аналізу можна зробити висновок, що у наш час існує досить багато програмних засобів для розпізнавання жестової мови. Проте деякі програмні засоби для розпізнавання жестової мови висувають високі вимоги до якості вхідних даних. Крім того, на ефективність розпізнавання за допомогою таких програмних систем суттєво впливає індивідуальна різниця у виконанні жестів, швидкість рухів та фізіологічні

особливості користувачів, що створює складності при навчанні та подальшому використанні розпізнавальних моделей у ПЗ. Ще одним недоліком таких програмних засобів є обмеженість мовного та культурного покриття. Більшість сучасних програмних систем орієнтовані на відносно поширені жестові мови, у той час як менш вивчені або регіональні мови залишаються недостатньо представленими в базах даних, що обмежує їх практичне застосування. Суттєвим викликом є також висока обчислювальна складність алгоритмів, що ускладнює реалізацію таких систем у мобільних пристроях або у середовищах з обмеженими ресурсами. Тому актуальною є розробка програмного забезпечення для розпізнавання жестової мови.

При розробці програмного забезпечення для розпізнавання жестової мови необхідно забезпечити такі функціональні вимоги:

- підтримка можливості отримання відеопотоку з камери;
- підтримка можливості виявлення регіону інтересу, де розташовано мовний жест;
- можливість нормалізації зображення;
- можливість синтезу нейромережових моделей для розпізнавання мови жестів;
- підтримка можливості розпізнавання мовних жестів за допомогою навченої нейромережової моделі;
- підтримка можливості виведення розпізнаного жесту;
- можливість інформування щодо ймовірності правильного розпізнавання мовного жесту.

1.3 Висновки за розділом 1

Визначено, що засоби для розпізнавання жестової мови являють собою комплекс інструментів, здатних інтерпретувати рухи рук та положення пальців з метою перетворення їх у зрозумілий для користувача текстовий або звуковий формат. Створення таких засобів ґрунтується на поєднанні

інструментів комп'ютерного зору, машинного навчання та технологій обробки сигналів, що дає змогу автоматизувати процес перекладу жестів.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато програмних засобів для розпізнавання жестової мови. Проте деякі програмні засоби для розпізнавання жестової мови висувають високі вимоги до якості вхідних даних. Крім того, на ефективність розпізнавання за допомогою таких програмних систем суттєво впливає індивідуальна різниця у виконанні жестів, швидкість рухів та фізіологічні особливості користувачів, що створює складності при навчанні та подальшому використанні розпізнавальних моделей у ПЗ. Ще одним недоліком таких програмних засобів є обмеженість мовного та культурного покриття. Більшість сучасних програмних систем орієнтовані на відносно поширені жестові мови, у той час як менш вивчені або регіональні мови залишаються недостатньо представленими в базах даних, що обмежує їх практичне застосування. Суттєвим викликом є також висока обчислювальна складність алгоритмів, що ускладнює реалізацію таких систем у мобільних пристроях або у середовищах з обмеженими ресурсами. Тому актуальною є розробка програмного забезпечення для розпізнавання жестової мови.

Сформульовано функціональні вимоги до програмного забезпечення для розпізнавання жестової мови.

2 МАТЕРІАЛИ І МЕТОДИ

2.1 Вибір мови програмування

При розробці програмного забезпечення для розпізнавання жестової мови було використано мову програмування Python [10], [11].

Мова програмування Python відзначається значною абстрактністю та адаптивністю, що дозволяє зосередитися на вирішенні завдань без надмірного залучення до технічних деталей реалізації. Ця мова сприяє безперешкодній взаємодії з іншими мовами та платформами, забезпечуючи швидке поєднання різних компонентів програмного забезпечення. Наявність розгалуженої екосистеми наукових бібліотек і програмних модулів відкриває можливості для виконання складних обчислень, проведення статистичних досліджень та візуалізації даних, що особливо важливо для застосування у сфері штучного інтелекту (ШІ) [10], [11].

Особливістю мови програмування Python є її кросплатформеність, що дозволяє запускати розроблені програми на різних операційних системах без значних модифікацій коду. Мова використовується як у наукових експериментах, так і в промислових проєктах, оскільки забезпечує швидке створення прототипів, тестування алгоритмів та інтеграцію рішень машинного навчання у практичні застосунки. Мова програмування Python забезпечує високу швидкість прототипування завдяки простому та зрозумілому синтаксису, що дозволяє швидко реалізовувати і тестувати різноманітні моделі без необхідності заглиблюватися у низькорівневі деталі. Крім того, Python активно підтримує розвиток відкритих проєктів, що сприяє спільній роботі та обміну напрацюваннями у професійній спільноті [10], [11].

Завдяки поєднанню простоти, універсальності та багатофункціональності Python забезпечує швидке впровадження новітніх технологій, зокрема у галузях комп'ютерного зору, обробки природної мови та автоматизації комплексних процесів. Це робить цю мову однією з найбільш

ефективних мов програмування для досліджень та практичних розробок сучасного рівня [10], [11].

Для створення програмного забезпечення, призначеного для розпізнавання жестової мови, доцільним є застосування мови Python, оскільки її особливості та розвинена екосистема значно спрощують роботу з алгоритмами штучного інтелекту та обробки зображень. Простота синтаксису та зрозуміла структура коду дозволяють швидко створювати прототипи моделей розпізнавання жестів і тестувати їхню ефективність без необхідності глибокого занурення у низькорівневі аспекти програмування [10], [11].

Важливе значення мають численні бібліотеки та фреймворки, що орієнтовані на обробку відеопотоків та зображень, а також на побудову нейронних мереж. Програмні інструменти, як OpenCV, забезпечують можливість виділення ключових точок та обробки відео в реальному часі, тоді як TensorFlow і PyTorch дозволяють реалізовувати складні інтелектуальні моделі для класифікації та прогнозування жестів. Додатково мова програмування Python підтримує інтеграцію з датчиками руху та апаратними пристроями, що особливо важливо для систем із використанням камер або розумних рукавичок [10], [11].

Активна спільнота мови програмування Python забезпечує її безперервний розвиток, надаючи нові модулі, приклади та документацію, що прискорює розробку та підвищує надійність програмного забезпечення. Це дає змогу розробникам швидко адаптувати алгоритми до різних жестових мов і умов експлуатації [10], [11].

У результаті мова програмування Python виявляється найбільш доцільним вибором для створення програмних систем розпізнавання жестової мови, поєднуючи простоту використання, широкий спектр програмних інструментів для роботи зі ШІ та комп'ютерним зором і підтримку науково-технічної спільноти [10], [11].

Обґрунтування вибору мови програмування для реалізації програмного забезпечення для розпізнавання жестової мови наведено у таблиці 2.1.

Таблиця 2.1 – Обґрунтування вибору мови програмування для реалізації програмного забезпечення для розпізнавання жестової мови

Критерій порівняння мов програмування	Мова програмування		
	Python	C#	JavaScript
Простота та швидкість прототипування	+	+–	+
Підтримка бібліотек для роботи з засобами штучного інтелекту	+	+–	+–
Обробка зображень та відео	+	+–	–
Інтеграція з апаратними пристроями	+	+	–
Активна науково-технічна спільнота	+	+–	+–

Отже, для реалізації програмного забезпечення для розпізнавання жестової мови обрано мову програмування Python через її особливості та екосистему, які значно полегшують роботу з алгоритмами машинного навчання та комп'ютерного зору. Python забезпечує високу швидкість прототипування завдяки простому та зрозумілому синтаксису, що дозволяє швидко реалізовувати і тестувати різноманітні моделі розпізнавання жестів без необхідності заглиблюватися у низькорівневі деталі. Особливе значення має наявність численних бібліотек і фреймворків, орієнтованих на обробку зображень і відео, а також на глибоке навчання. Крім технічних переваг, Python має активну спільноту, що постійно розробляє нові модулі, надає приклади, документацію та готові рішення для задач розпізнавання жестів різними мовами. Це сприяє скороченню часу розробки, підвищенню надійності програмного забезпечення та можливості швидко адаптувати систему під нові умови чи жестові мови.

2.2 Вибір середовища розробки для створення програмного забезпечення для розпізнавання жестової мови

В якості середовища розробки для створення програмного забезпечення для розпізнавання жестової мови було обрано середовище Sublime Text [12], [13].

Середовище розробки Sublime Text вирізняється гнучкістю та здатністю адаптуватися до різних підходів у програмуванні, що забезпечує зручну роботу як із невеликими скриптами, так і з масштабними проектами. Інтерфейс редактора Sublime Text спроектований таким чином, що навігація між файлами та каталогами здійснюється швидко та інтуїтивно, що значно полегшує організацію коду. Можливість одночасної роботи з кількома вікнами та панелями дозволяє розробникам паралельно відслідковувати різні модулі або компоненти, підвищуючи ефективність налагодження та тестування алгоритмів функціонування програмного забезпечення [12], [13].

Редактор Sublime Text відзначається широкими можливостями персоналізації, що досягається через налаштування та плагіни, які дозволяють додавати інструменти для перевірки синтаксису, аналізу коду та автоматизації повторюваних операцій. Висока швидкодія при обробці великих файлів і програмних проєктів робить його придатним для створення складних програмних систем, включно з обробкою зображень, відеопотоків та сигналів від сенсорних пристроїв [12], [13].

Підтримка кросплатформенності забезпечує перенесення робочих проєктів між різними операційними системами без втрати продуктивності або налаштувань, що робить середовище зручним для командної розробки, де важливо зберігати єдину конфігурацію для всіх учасників. Завдяки поєднанню швидкості, стабільності та широких можливостей налаштувань під потреби користувача, середовище Sublime Text стає ефективним інструментом для організації процесу розробки програмного забезпечення у сфері розпізнавання жестової мови [12], [13].

Для створення програмного забезпечення, призначеного для розпізнавання жестової мови, доцільним є застосування середовища розробки Sublime Text, оскільки воно відзначається швидкістю роботи, легкістю та високим рівнем налаштовуваності. Редактор Sublime Text забезпечує комфортну роботу з кодом завдяки підсвічуванню синтаксису, автодоповненню та підтримці багатовіконного режиму, що значно підвищує ефективність при реалізації складних алгоритмів обробки відеопотоків і сигналів від сенсорних пристроїв [12], [13].

Важливим аспектом є здатність інтегрувати середовище Sublime Text із різними мовами програмування, бібліотеками та фреймворками, які застосовуються у сфері машинного навчання та комп'ютерного зору. Плагіни та програмні розширення дозволяють адаптувати робоче середовище під конкретні завдання розробника, автоматизувати рутинні операції та оптимізувати процес розробки [12], [13].

Середовище розробки Sublime Text вирізняється стабільністю та невисокими системними вимогами, що забезпечує комфортну роботу навіть на менш продуктивних комп'ютерах. Можливість одночасного опрацювання великої кількості файлів і легка кастомізація підтримують ефективну організацію програмного проекту, що є критично важливим при створенні комплексних програмних систем з численними модулями та компонентами, зокрема, з використанням засобів штучного інтелекту [12], [13].

У результаті застосування Sublime Text сприяє підвищенню продуктивності розробки, забезпечує гнучкість роботи з кодом і дозволяє швидко прототипувати алгоритми для систем автоматичного розпізнавання жестової мови [12], [13].

Обґрунтування вибору середовища розробки для створення програмного забезпечення для розпізнавання жестової мови наведено у таблиці 2.2.

Таблиця 2.2 – Обґрунтування вибору середовища розробки для створення програмного забезпечення для розпізнавання жестової мови

Критерій порівняння середовищ розробки	Середовища розробки		
	Sublime Text	VS Code	PyCharm
Швидкість роботи та легкість запуску	+	+–	+–
Простота налаштування	+	+	+–
Зручність інтеграції з бібліотеками Python	+	+–	+–
Продуктивність при налагодженні коду	+	+	+–
Низькі системні вимоги	+	–	+–

Таким чином, для створення програмного забезпечення для розпізнавання жестової мови обрано середовище розробки Sublime Text завдяки його легкості, швидкості роботи та гнучкості налаштування. Середовище Sublime Text забезпечує зручне редагування коду з підсвічуванням синтаксису, автодоповненням та багатовіконною роботою, що значно підвищує ефективність роботи при написанні складних алгоритмів обробки відео та сигналів від сенсорних пристроїв. Наявність плагінів та розширень дозволяє швидко адаптувати середовище під конкретні потреби проєкту, автоматизувати рутинні процеси та оптимізувати робочий процес. Крім того, Sublime Text характеризується високою стабільністю та низькими системними вимогами, що дозволяє працювати навіть на менш потужних комп'ютерах без втрати продуктивності. Легке налаштування і можливість роботи з великою кількістю файлів одночасно забезпечують ефективну організацію проєкту, що особливо важливо для розробки комплексних систем розпізнавання жестової мови з численними модулями і компонентами.

2.3 Висновки за розділом 2

Для реалізації програмного забезпечення для розпізнавання жестової мови обрано мову програмування Python через її особливості та екосистему, які значно полегшують роботу з алгоритмами машинного навчання та комп'ютерного зору. Python забезпечує високу швидкість прототипування завдяки простому та зрозумілому синтаксису, що дозволяє швидко реалізовувати і тестувати різноманітні моделі розпізнавання жестів без необхідності заглиблюватися у низькорівневі деталі.

Для створення програмного забезпечення для розпізнавання жестової мови обрано середовище розробки Sublime Text завдяки його легкості, швидкості роботи та гнучкості налаштування. Середовище Sublime Text забезпечує зручне редагування коду з підсвічуванням синтаксису, автодоповненням та багатовіконною роботою, що значно підвищує ефективність роботи при написанні складних алгоритмів обробки відео та сигналів від сенсорних пристроїв.

3 ОПИС ПРОГРАМИ

3.1 Структура програмного забезпечення для розпізнавання жестової мови

Структуру програмного забезпечення для розпізнавання жестової мови наведено на рис. 3.1.



Рисунок 3.1 – Структура програмного забезпечення для розпізнавання жестової мови

Структура програмного забезпечення для розпізнавання жестової мови може бути представлена у вигляді п'яти взаємопов'язаних модулів, кожен з яких виконує окрему функцію та забезпечує послідовне перетворення даних

від моменту отримання відеопотоку до відображення результату розпізнавання. Основними модулями є: модуль захоплення відео, модуль виділення регіону інтересу, модуль попередньої обробки, модуль створення та навчання нейромережових моделей, модуль розпізнавання жестів.

Модуль захоплення відео відповідає за отримання потоку з камери в режимі реального часу та передачу його в систему для подальшої обробки. У ньому виконується зчитування відеопотоку, кадрування області з руками або тілом, а також первинна обробка, наприклад нормалізація яскравості, зменшення шумів чи приведення зображення до потрібного формату. У цьому блоці формується база вхідних даних, яка надалі використовується для навчання та тестування системи.

Модуль виділення регіону інтересу визначає зону кадру, де знаходиться жест, і передає лише необхідні фрагменти зображення для зменшення обчислювальних витрат. Тут здійснюється виділення контурів, сегментація зображення, відокремлення жесту від фону та визначення ключових точок. Застосовуються методи комп'ютерного зору для покращення якості даних і зменшення впливу сторонніх факторів.

Модуль попередньої обробки забезпечує нормалізацію зображення, усунення шумів, зміну роздільної здатності та приведення вхідних даних до уніфікованого формату.

Модуль створення та навчання нейромережових моделей забезпечує побудову архітектури штучних нейронних мереж, їх тренування на навчальних вибірках та збереження вагових коефіцієнтів для подальшого використання.

Модуль розпізнавання жестів за допомогою навченої нейромережової моделі повертає ідентифікований жест разом з інформацією про його вірогідність. Модуль розпізнавання виконує класифікацію отриманих даних. Тут застосовуються синтезовані нейронні мережі, які навчені на попередньо зібраній базі жестів, вхідні ознаки порівнюються з еталонними, визначається найбільш ймовірний варіант, і система видає результат. Важливим аспектом є

оптимізація цього процесу для досягнення високої точності навіть у реальному часі.

Крім того, додатковий модуль відповідає за представлення результатів, який виводить розпізнаний жест у зрозумілому для користувача форматі та додатково інформує про рівень ймовірності правильності класифікації.

Така послідовна модульна організація дозволяє ефективно реалізувати всі вимоги до програмного забезпечення для розпізнавання жестової мови, забезпечуючи як технічну коректність роботи системи, так і зручність взаємодії користувача з результатами розпізнавання.

3.2 Функціонування програмного забезпечення для розпізнавання жестової мови

Програмне забезпечення для розпізнавання жестової мови функціонує на основі поєднання комп'ютерного зору, машинного навчання та лінгвістичного аналізу. На початковому етапі відбувається зчитування рухів користувача за допомогою камери або сенсорних пристроїв, що дозволяє зафіксувати положення рук, пальців та міміку обличчя. Зібрані дані перетворюються у цифровий формат, де кожен рух розкладається на ключові характеристики, такі як траєкторія, швидкість і форма.

Подальша обробка передбачає використання спеціалізованих алгоритмів, які визначають відповідність між зафіксованими жестами та мовними одиницями. Програмне забезпечення для розпізнавання жестової мови застосовує нейронні мережі для розпізнавання закономірностей і відокремлення випадкових рухів від тих, що мають мовне значення. Після цього програмне забезпечення для розпізнавання жестової мови трансформує жести у зрозумілий текстовий або голосовий формат, що дозволяє здійснювати повноцінну комунікацію між людьми, які користуються жестовою мовою, та тими, хто нею не володіє.

Функціонування програмного забезпечення для розпізнавання жестової мови подамо за допомогою схеми, зображеної на рис. 3.2.



Рисунок 3.2 – Схема функціонування програмного забезпечення для розпізнавання жестової мови

Функціонування програмного забезпечення для розпізнавання жестової мови відбувається поетапно, коли кожний крок забезпечує поступовий перехід

від реального руху до зрозумілого тексту. Спочатку здійснюється захоплення зображення чи відеопотоку за допомогою камери.

Отримані дані передаються у систему для попередньої обробки, де зображення очищується від шумів, нормалізується яскравість і контрастність, а також виділяються ключові області, що містять рухи рук чи положення пальців.

Після цього відбувається сегментація зображення з відокремленням області жесту від фону. На наступному етапі проводиться виявлення та відстеження характерних точок або контурів, які відображають форму та динаміку рухів. Далі здійснюється екстракція ознак, коли виділяються числові характеристики, що описують положення, траєкторію та форму жесту.

Ці ознаки передаються в модуль розпізнавання, який використовує алгоритми машинного навчання або нейронні мережі для зіставлення отриманих даних із відомими шаблонами жестів.

У разі успішного визначення жест перетворюється на відповідний текстовий опис. Завершальним етапом є відображення результату, що дозволяє користувачеві отримати зрозумілий переклад жестової мови.

3.3 Розробка нейромережевої моделі для розпізнавання жестової мови

У розробленому програмному забезпеченні для розпізнавання жестової мови використовується нейромережева модель. Процес побудови нейромережевої моделі для системи розпізнавання жестової мови починається з визначення принципів опрацювання даних і вибору архітектури, здатної відтворити складність жестових рухів. Щоб алгоритм міг правильно інтерпретувати жести, з вхідних зображень виділяються характерні ознаки, які стають основою для подальшої класифікації. Для цього потрібна велика кількість інформативних характеристик, що дозволяє системі враховувати як форму, так і динаміку жесту.

Для розпізнавання жестової мови доцільно використовувати згорткові нейронні мережі, що довели свою ефективність у роботі з візуальними даними. Їхня структура (рис. 3.3) складається з кількох функціональних рівнів: згортковий шар виявляє ознаки, пулінговий шар виконує скорочення розмірів зображення для зменшення обчислювальної складності, а повнозв'язний шар реалізує класифікацію отриманих результатів. На відміну від класичних нейромереж, ця архітектура працює з даними у трьох вимірах, що забезпечує більш глибоке відображення просторових зв'язків.

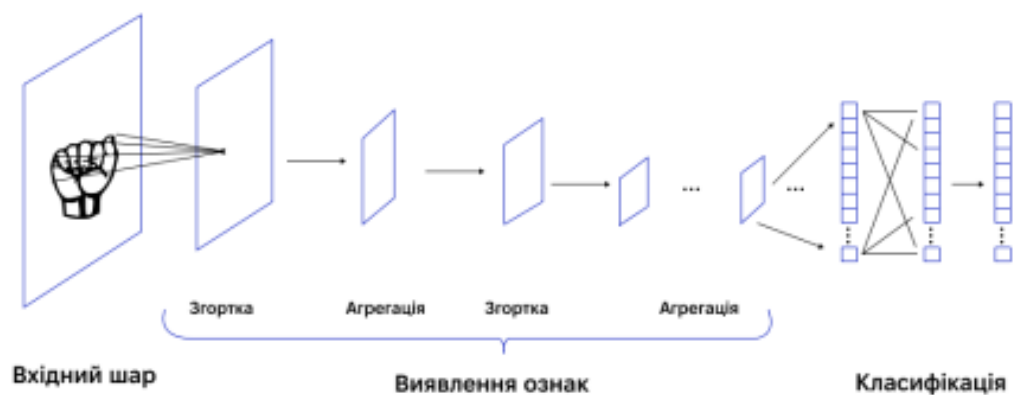


Рисунок 3.3 – Графічне подання нейромережевої моделі для розпізнавання жестової мови

Особливістю таких нейромережевих моделей є відсутність потреби у ручному визначенні ознак, оскільки мережа самостійно навчається розпізнавати суттєві елементи зображення. Крім того, такі моделі виявляють здатність розрізняти користувачів, які не брали участі у навчанні, що створює додаткову можливість ідентифікації. Висока стійкість до варіацій освітлення, масштабу та часткових перешкод робить згорткову нейромережеву модель для розпізнавання жестової мови придатною для роботи у реальних умовах, де жести можуть відрізнятися залежно від контексту.

Структуру згорткової нейромережевої моделі для розпізнавання жестової мови наведено на рис. 3.4.

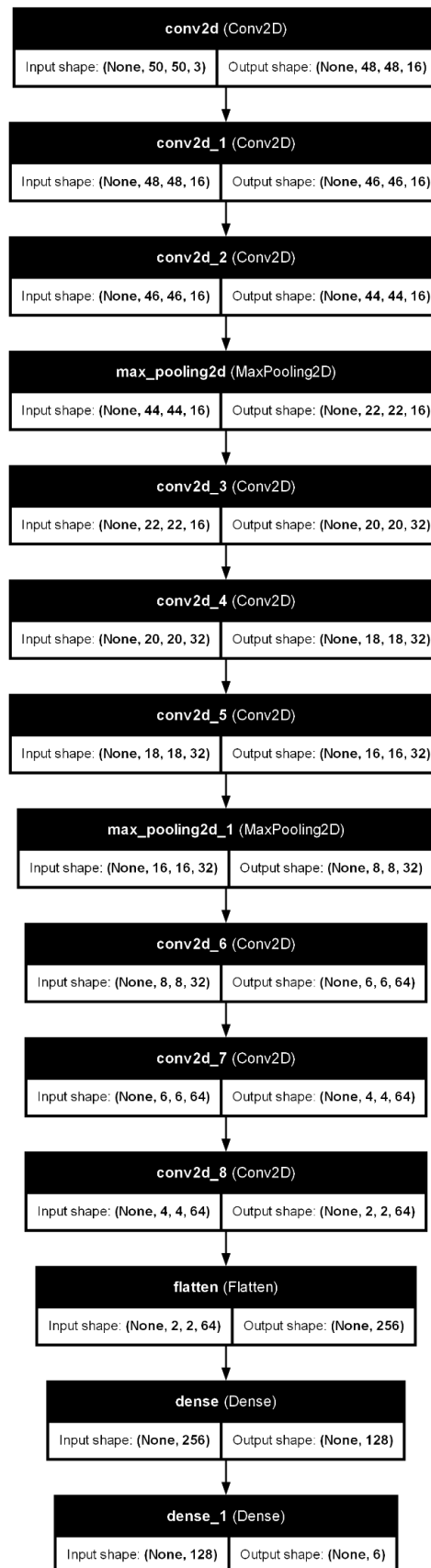


Рисунок 3.4 – Структура згорткової нейромережевої моделі для розпізнавання жестової мови

3.4 Навчання нейромережевої моделі для розпізнавання жестової мови

У процесі створення нейромережевої моделі для розпізнавання жестової мови важливим є етап навчання, у процесі якого здійснюється пошук оптимальних параметрів. Це завдання формулюється як глобальна оптимізація, що передбачає вибір найкращого рішення з-поміж великої кількості можливих варіантів. Ключову роль у цьому процесі відіграє оптимізація функції втрат, яка дозволяє поступово наближатися до ефективного стану моделі. Одним із найбільш поширених методів такого налаштування є стохастичний градієнтний спуск, що оновлює параметри мережі на основі обчислених градієнтів, забезпечуючи поступове покращення якості класифікації.

На рис. 3.5 наведено структуру шарів згорткової нейромережевої моделі для розпізнавання жестової мови, де міститься інформація про кількість нейронів на кожному шарі та кількість налаштовуваних параметрів.

```
Model: "sequential"
```

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 48, 48, 16)	448
conv2d_1 (Conv2D)	(None, 46, 46, 16)	2320
conv2d_2 (Conv2D)	(None, 44, 44, 16)	2320
max_pooling2d (MaxPooling2D)	(None, 22, 22, 16)	0
conv2d_3 (Conv2D)	(None, 20, 20, 32)	4640
conv2d_4 (Conv2D)	(None, 18, 18, 32)	9248
conv2d_5 (Conv2D)	(None, 16, 16, 32)	9248
max_pooling2d_1 (MaxPooling2D)	(None, 8, 8, 32)	0
conv2d_6 (Conv2D)	(None, 6, 6, 64)	18496
conv2d_7 (Conv2D)	(None, 4, 4, 64)	36928
conv2d_8 (Conv2D)	(None, 2, 2, 64)	36928
flatten (Flatten)	(None, 256)	0
dense (Dense)	(None, 128)	32896
dense_1 (Dense)	(None, 6)	774

```

-----
Total params: 154246 (602.52 KB)
Trainable params: 154246 (602.52 KB)
Non-trainable params: 0 (0.00 Byte)

```

Рисунок 3.5 – Структура згорткової НМ для розпізнавання жестової мови

У процесі побудови моделі для розпізнавання жестової мови застосовується архітектура згорткової нейронної мережі, що складається з кількох послідовних блоків, які поступово трансформують зображення у набір компактних та інформативних ознак. На початковому етапі зображення вводиться у мережу у вигляді невеликої матриці, після чого відбувається його обробка фільтрами. Перші згорткові шари виконують виділення найпростіших характеристик, таких як контури та текстури, а подальші рівні ускладнюють ці ознаки, виявляючи все більш абстрактні патерни. Застосування операцій пулінгу забезпечує зменшення розмірів проміжних представлень, що знижує обчислювальні витрати та зменшує ризик перенавчання.

Подальше нарощування кількості фільтрів дозволяє нейронній мережі аналізувати складніші структури зображення з відповідним мовним жестом, фокусуючись на більш високорівневих деталях. На завершальних етапах обробки тривимірні тензори перетворюються у вектори ознак, які надходять до повнозв'язних шарів нейромережевої моделі. Ці шари виконують інтеграцію всієї попередньо зібраної інформації та здійснюють класифікацію жестів відповідно до заданих категорій.

У згортковій нейромережевій моделі для розпізнавання жестової мови використовується активаційна функція ReLU, яка забезпечує нелінійність процесу навчання та дозволяє мережі коректно відтворювати складні залежності між ознаками. Вона активує нейрон лише у випадку, коли вхідне значення є додатним, що значно прискорює навчання і зменшує ризик зникнення градієнта.

Завдяки такій організації згорткової нейромережевої моделі для розпізнавання жестової мови досягається поступове узагальнення інформації: від найпростіших ознак на початкових рівнях до високорівневих представлень на виході. Це дає змогу моделі точно класифікувати жести навіть за умов різного освітлення, варіацій масштабу чи часткових перешкод, зберігаючи стабільність і високу точність роботи.

Для навчання нейромережевої моделі для розпізнавання жестової мови використовувався набір даних SL-MNIST [14], який містить жести американської жестової мови (рис. 3.6). Його основу становлять зображення символів алфавіту, представлені у вигляді статичних кадрів, що дозволяє застосовувати методи згорткових нейронних мереж для класифікації. Структура даних є спрощеною, оскільки кожне зображення має невеликий розмір у відтінках сірого, що полегшує обчислення та пришвидшує процес навчання. При цьому візуальна інформація зберігає ключові риси жестів, достатні для ідентифікації.



Рисунок 3.6 – Графічне подання фрагменту набору даних SL-MNIST [14]

Набір даних SL-MNIST побудований на основі класичного MNIST, який свого часу використовувався для класифікації рукописних цифр, проте SL-MNIST є його адаптацією до сфери жестової мови. Він охоплює двадцять чотири класи, кожен з яких відповідає певній літері, за винятком символів, що вимагають руху. Кожен клас представлений великою кількістю зображень, що створює збалансовану структуру даних і забезпечує рівномірність під час навчання моделі [14].

Матеріали доступні для вільного використання, що робить набір одним з найбільш популярних у дослідженнях і прикладних рішеннях у сфері розпізнавання жестів. Завдяки компактності та добре організованій структурі SL-MNIST дозволяє перевіряти якість алгоритмів комп'ютерного зору, тестувати архітектури нейронних мереж і проводити порівняння результатів. Наявність великої кількості прикладів жестів різних користувачів сприяє підвищенню узагальнюючої здатності моделей і створює основу для практичного використання таких систем у реальних умовах.

3.5 Висновки за розділом 3

Розроблено програмне забезпечення для розпізнавання жестової мови.

Запропоновано структуру програмного забезпечення для розпізнавання жестової мови. Визначено, що основними модулями є: модуль захоплення відео, модуль виділення регіону інтересу, модуль попередньої обробки, модуль створення та навчання нейромережевих моделей, модуль розпізнавання жестів.

Описано функціонування програмного забезпечення для розпізнавання жестової мови. Визначено, що функціонування програмного забезпечення для розпізнавання жестової мови відбувається поетапно, коли кожний крок забезпечує поступовий перехід від реального руху до зрозумілого тексту.

Описано особливості реалізації програмного забезпечення для розпізнавання жестової мови. Виконано проектування інтерфейсу взаємодії користувача з програмним забезпеченням для розпізнавання жестової мови.

Розроблено згорткову нейромережеву модель, яка являє собою набір взаємопов'язаних нейронів та дозволяє виконувати розпізнавання жестової мови. Згорткова НМ модель забезпечує високу стійкість до варіацій освітлення, масштабу та часткових перешкод, що дозволяє її використовувати у реальних умовах, де жести можуть відрізнятися залежно від контексту.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

4.1 Призначення й умови застосування програми

Програма призначена для розпізнавання жестової мови. Програма написана на мові Python через її особливості та екосистему, які значно полегшують роботу з алгоритмами машинного навчання та комп'ютерного зору. В якості середовища розробки для створення програмного забезпечення для розпізнавання жестової мови обрано Sublime Text завдяки його легкості, швидкості роботи та гнучкості налаштування.

Програмне забезпечення для розпізнавання жестової мови забезпечує виконання таких функцій:

- підтримка можливості отримання відеопотоку з камери;
- підтримка можливості виявлення регіону інтересу, де розташовано мовний жест;
- можливість нормалізації зображення;
- можливість синтезу нейромережових моделей для розпізнавання мови жестів;
- підтримка можливості розпізнавання мовних жестів за допомогою навченої нейромережової моделі;
- підтримка можливості виведення розпізнаного жесту;
- можливість інформування щодо ймовірності правильного розпізнавання мовного жесту.

Для експлуатації програмного забезпечення для розпізнавання жестової мови необхідно таке апаратне та програмне забезпечення:

- процесор із багатоядерною архітектурою та тактовою частотою не нижче 2,5 ГГц для забезпечення швидкої обробки відеопотоку та роботи алгоритмів машинного навчання;
- оперативна пам'ять обсягом не менше 8 ГБ для стабільної роботи системи та завантаження моделей розпізнавання;

- графічний процесор для прискорення обробки нейронних мереж;
- вільний простір на жорсткому диску не менше 5 ГБ для встановлення програмного забезпечення та збереження тимчасових файлів;
- вебкамера з роздільною здатністю від 720p і вище, яка здатна працювати в режимі реального часу.

Функціональні характеристики розробленого програмного забезпечення для розпізнавання жестової мови наведено у технічному завданні (додаток А). Фрагмент тексту програмного забезпечення для розпізнавання жестової мови наведено у додатку Б.

4.2 Характеристики програми для розпізнавання жестової мови

Програмне забезпечення для розпізнавання жестової мови дозволяє забезпечити взаємодію між людьми, які спілкуються жестами, та тими, хто використовує усну мову. Його функціонування базується на застосуванні технологій комп'ютерного зору, машинного навчання та штучного інтелекту, що дозволяють у реальному часі аналізувати рухи рук, міміку та положення тіла. Таке програмне забезпечення здатне розпізнавати окремі жести, та перетворювати на зрозумілий текст.

Основна характеристика програмного забезпечення для розпізнавання жестової мови полягає у високій чутливості до деталей руху, що забезпечується завдяки використанню камер високої роздільності. Для досягнення точності аналізу застосовуються згорткові нейромереві моделі, здатні враховувати не лише форму руки чи пальців, а й динаміку руху, швидкість та контекст, у якому використовується той чи інший жест. Це дає змогу відрізнити схожі рухи, які можуть мати різне значення у жестовій мові.

Важливим аспектом роботи програмного забезпечення для розпізнавання жестової мови є адаптивність. Воно завдяки можливості синтезу різних розпізнавальних моделей, може враховувати індивідуальні відмінності користувачів, адже жести можуть виконуватися з різною точністю, темпом або

стилем. Для цього застосовуються моделі навчання, які можуть поступово вдосконалюватися під час використання програмної системи. Це дозволяє підвищувати рівень коректності розпізнавання та робить програму більш зручною для щоденного застосування.

Програмне забезпечення для розпізнавання жестової мови може обробляти дані у режимі реального часу. Важливою властивістю програм є здатність інтегруватися з іншими програмними сервісами та платформами.

Таким чином, програмне забезпечення для розпізнавання жестової мови характеризується комплексністю, поєднанням точності, швидкодії та зручності використання.

4.3 Інструкція по експлуатації програми

4.3.1 Звернення до програми

Для запуску програмного забезпечення для розпізнавання жестової мови на серверній частині необхідно запустити відповідний ру-файл.

4.3.2 Вхідні й вихідні дані

Вхідними даними до програмного забезпечення для розпізнавання жестової мови є відеопотік з зображеннями користувачів та їх діями, які відображають жести жестової мови.

Вихідними даними програмного забезпечення для розпізнавання жестової мови є розпізнані жести.

4.3.3 Повідомлення

Користувач програмного забезпечення для розпізнавання жестової мови може отримати такі повідомлення: повідомлення про відсутність з'єднання з камерою.

4.4 Виконання програмного забезпечення для розпізнавання жестової мови

Програмне забезпечення для розпізнавання жестової мови реалізовано у вигляді набору програмних модулів, що дозволяють виконувати розпізнавання жестів людини та можуть бути імплементовані до більш складних програмних систем.

У складі програми реалізовано демонстраційний модуль, після запуску якого забезпечується підключення до камери пристрою та отримання відеопотоку з камери. Після отримання відеопотоку з камери програма виконує виявлення регіону інтересу, де розташовано мовний жест. Регіон інтересу виділяється зеленим прямокутником. Після цього на основі синтезованої нейромережевої моделі виконується розпізнавання мовних жестів. Розпізнані жести виводяться у текстовому вигляді трохи вище рамки з відповідним регіоном інтересу.

Результати виконання програми при розпізнаванні різних символів, які користувач відображає за допомогою жестової мови, наведено на рис. 4.1-4.6.

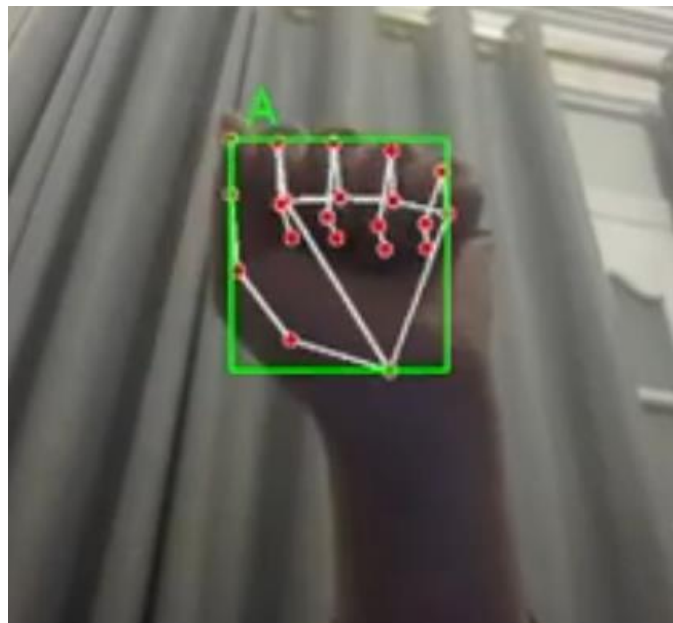


Рисунок 4.1 – Відображення результату розпізнавання жестового символу А

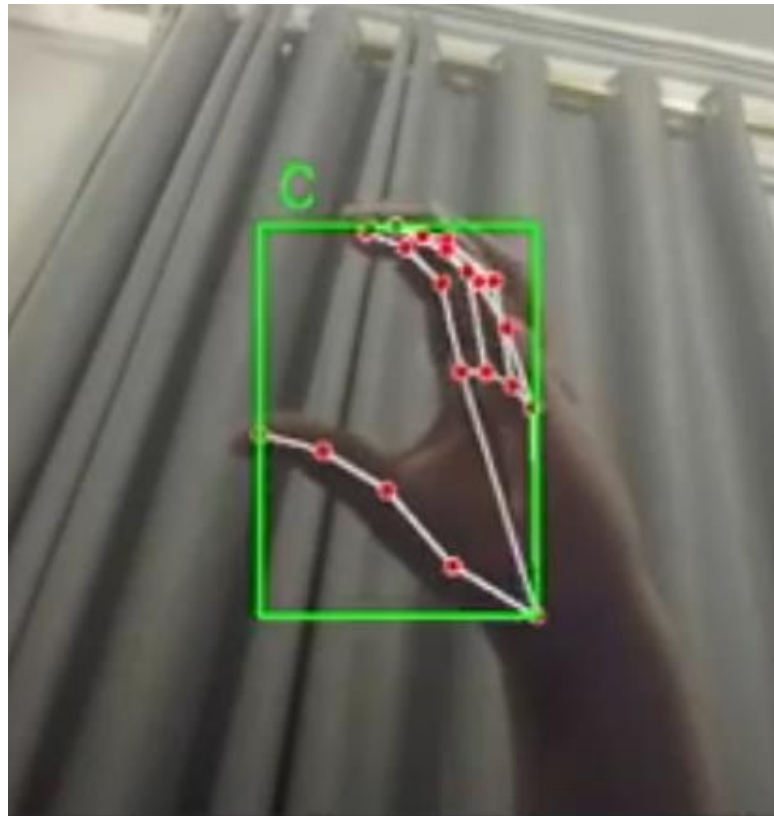


Рисунок 4.2 – Відображення результату розпізнавання жестового символу С

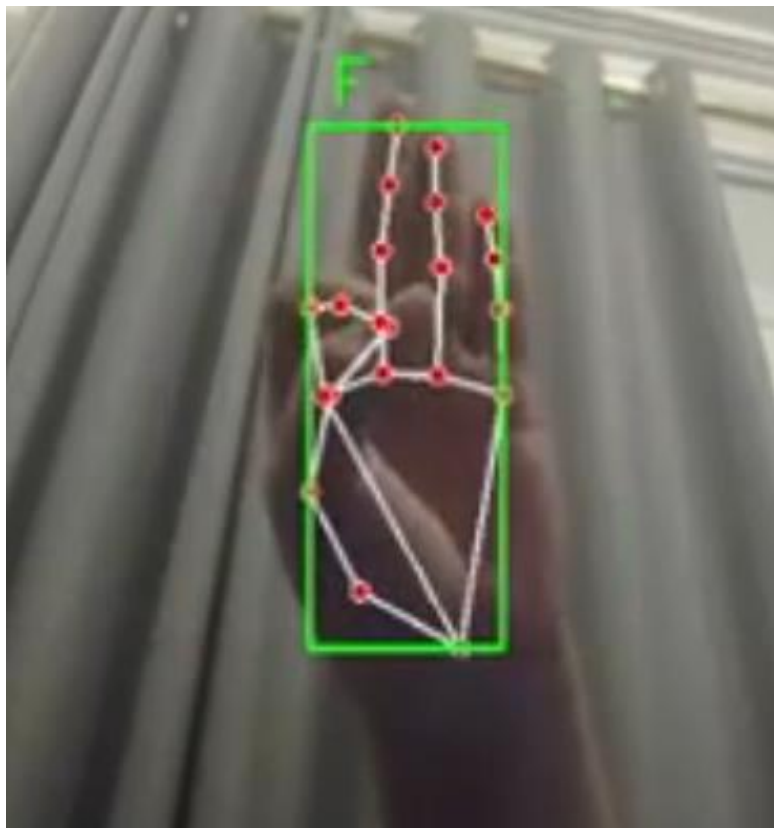


Рисунок 4.3 – Відображення результату розпізнавання жестового символу F

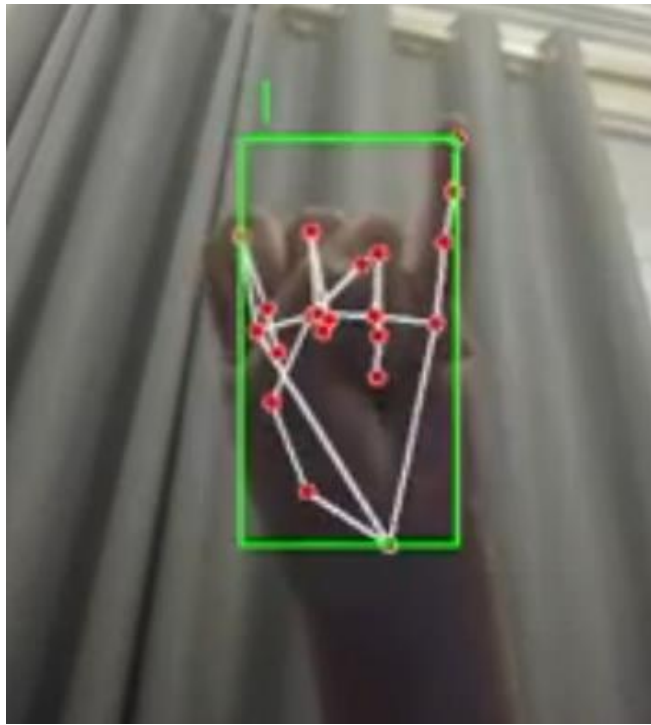


Рисунок 4.4 – Відображення результату розпізнавання жестового символу І

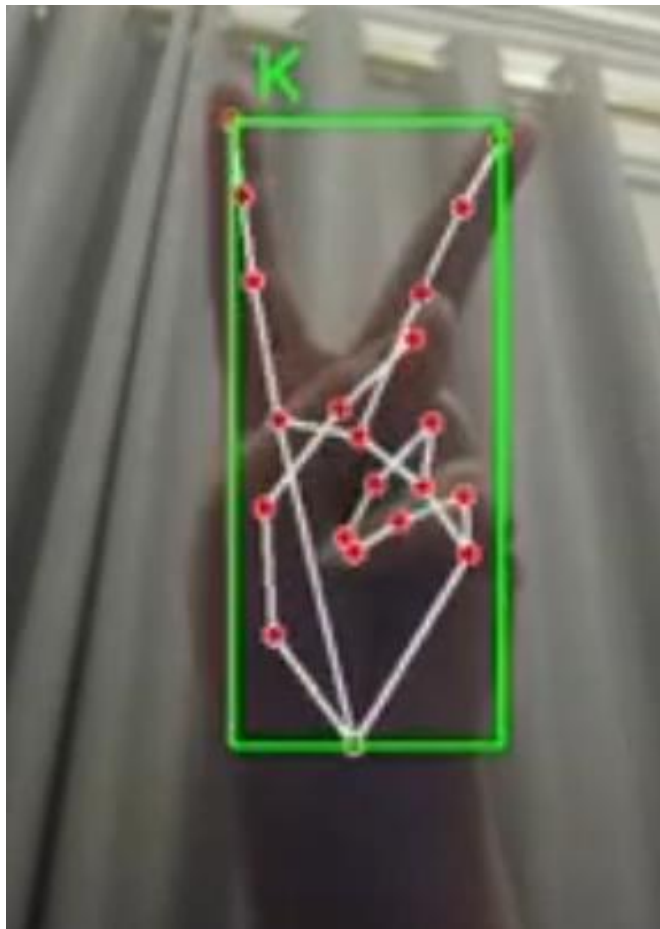


Рисунок 4.5 – Відображення результату розпізнавання жестового символу К

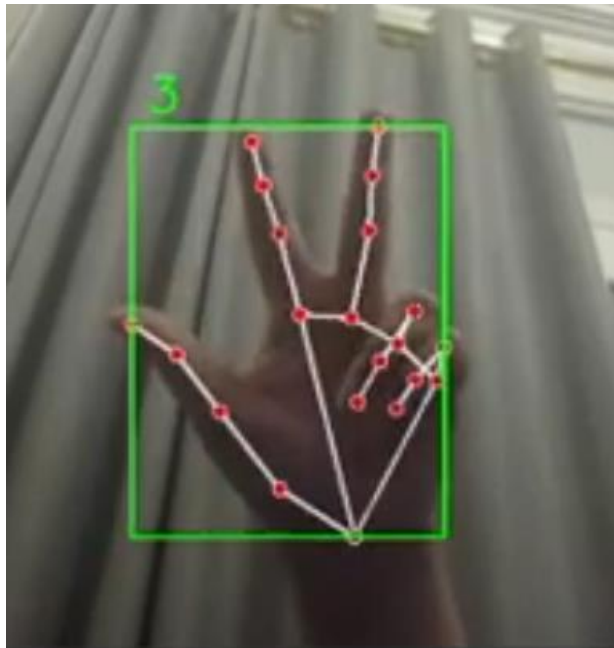


Рисунок 4.6 – Відображення результату розпізнавання жестового символу 3

За необхідності користувач може увімкнути (шляхом натискання великої літери І) можливість інформування щодо ймовірності правильного розпізнавання мовного жесту (рис. 4.7).

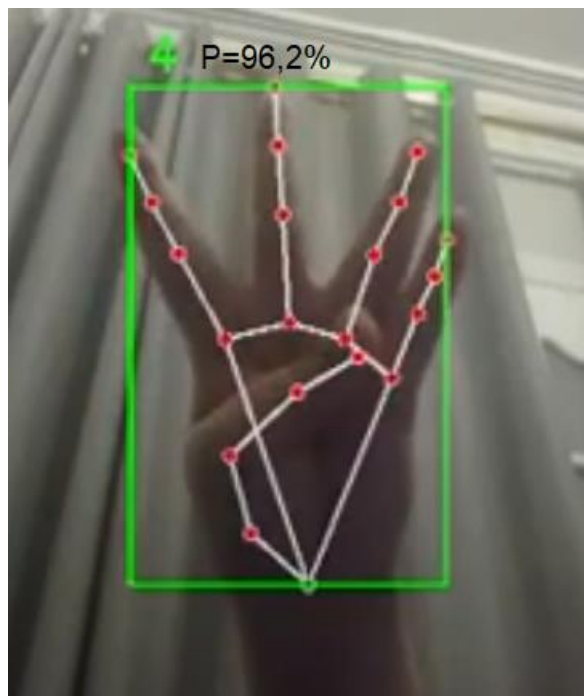


Рисунок 4.7 – Відображення результату розпізнавання жестового символу 4 з позначенням ймовірності правильного розпізнавання

4.5 Тестування програмного забезпечення для розпізнавання жестової мови

Виконано тестування програмного забезпечення для розпізнавання жестової мови.

Виявлено, що програма для розпізнавання жестової мови функціонує правильно та злагоджено. Вона реалізує всі функціональні вимоги та успішно виконує свою основну задачу підтримки процесу розпізнавання жестової мови.

Розроблене програмне забезпечення дозволяє забезпечувати автоматизацію процесів, пов'язаних з підтримкою розпізнавання жестової мови.

4.6 Висновки за розділом 4

Описано програмне забезпечення для розпізнавання жестової мови. Виконано тестування розробленого програмного забезпечення для розпізнавання жестової мови. Результати тестування програмного забезпечення показали, що розроблена програма дозволяє забезпечити автоматизацію процесів, пов'язаних з розпізнаванням жестової мови.

ВИСНОВКИ

В ході виконання дипломної кваліфікаційної роботи магістра було проаналізовано та досліджено процеси обчислень, пов'язані з розпізнаванням мови жестів.

Визначено, що засоби для розпізнавання жестової мови являють собою комплекс інструментів, здатних інтерпретувати рухи рук та положення пальців з метою перетворення їх у зрозумілий для користувача текстовий або звуковий формат. Створення таких засобів ґрунтується на поєднанні інструментів комп'ютерного зору, машинного навчання та технологій обробки сигналів, що дає змогу автоматизувати процес перекладу жестів.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато програмних засобів для розпізнавання жестової мови. Проте деякі програмні засоби для розпізнавання жестової мови висувають високі вимоги до якості вхідних даних. Крім того, на ефективність розпізнавання за допомогою таких програмних систем суттєво впливає індивідуальна різниця у виконанні жестів, швидкість рухів та фізіологічні особливості користувачів, що створює складності при навчанні та подальшому використанні розпізнавальних моделей у ПЗ. Ще одним недоліком таких програмних засобів є обмеженість мовного та культурного покриття. Більшість сучасних програмних систем орієнтовані на відносно поширені жестові мови, у той час як менш вивчені або регіональні мови залишаються недостатньо представленими в базах даних, що обмежує їх практичне застосування. Суттєвим викликом є також висока обчислювальна складність алгоритмів, що ускладнює реалізацію таких систем у мобільних пристроях або у середовищах з обмеженими ресурсами. Тому актуальною є розробка програмного забезпечення для розпізнавання жестової мови.

Сформульовано функціональні вимоги до програмного забезпечення для розпізнавання жестової мови.

Для реалізації програмного забезпечення для розпізнавання жестової мови обрано мову програмування Python через її особливості та екосистему, які значно полегшують роботу з алгоритмами машинного навчання та комп'ютерного зору. Python забезпечує високу швидкість прототипування завдяки простому та зрозумілому синтаксису, що дозволяє швидко реалізовувати і тестувати різноманітні моделі розпізнавання жестів без необхідності заглиблюватися у низькорівневі деталі.

Для створення програмного забезпечення для розпізнавання жестової мови обрано середовище розробки Sublime Text завдяки його легкості, швидкості роботи та гнучкості налаштування. Середовище Sublime Text забезпечує зручне редагування коду з підсвічуванням синтаксису, автодоповненням та багатовіконною роботою, що значно підвищує ефективність роботи при написанні складних алгоритмів обробки відео та сигналів від сенсорних пристроїв.

Новизна роботи полягає у тому, що розроблено згорткову нейромережеву модель, яка являє собою набір взаємопов'язаних нейронів та дозволяє виконувати розпізнавання жестової мови. Згорткова НМ модель забезпечує високу стійкість до варіацій освітлення, масштабу та часткових перешкод, що дозволяє її використовувати у реальних умовах, де жести можуть відрізнятися залежно від контексту.

Практичне значення роботи полягає у тому, що створено програмне забезпечення для розпізнавання жестової мови.

Запропоновано структуру програмного забезпечення для розпізнавання жестової мови. Визначено, що основними модулями є: модуль захоплення відео, модуль виділення регіону інтересу, модуль попередньої обробки, модуль створення та навчання нейромережевих моделей, модуль розпізнавання жестів.

Описано функціонування програмного забезпечення для розпізнавання жестової мови. Визначено, що функціонування програмного забезпечення для

розпізнавання жестової мови відбувається поетапно, коли кожний крок забезпечує поступовий перехід від реального руху до зрозумілого тексту.

Описано особливості реалізації програмного забезпечення для розпізнавання жестової мови. Виконано проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для розпізнавання жестової мови.

Виконано тестування розробленого програмного забезпечення для розпізнавання жестової мови. Результати тестування програмного забезпечення показали, що розроблена програма дозволяє забезпечити автоматизацію процесів, пов'язаних з розпізнаванням жестової мови.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Sign Language Recognition: A Deep Survey [Electronic resource]. – Access mode: <https://www.sciencedirect.com/science/article/abs/pii/S095741742030614X>.
2. Artificial Intelligence Technologies for Sign Language [Electronic resource]. – Access mode: <https://pmc.ncbi.nlm.nih.gov/articles/PMC8434597/>.
3. Exploration of Sign Language Recognition Methods Based on Improved YOLOv5s [Electronic resource]. – Access mode: <https://www.mdpi.com/2079-3197/13/3/59>.
4. A review of deep learning-based approaches to sign language processing [Electronic resource]. – Access mode: <https://www.tandfonline.com/doi/full/10.1080/01691864.2024.2442721>.
5. Sign Language Recognition with Advanced Computer Vision [Electronic resource]. – Access mode: <https://towardsdatascience.com/sign-language-recognition-with-advanced-computer-vision-7b74f20f3442/>.
6. Recognizing American Sign Language gestures efficiently and accurately using a hybrid transformer model [Electronic resource]. – Access mode: <https://www.nature.com/articles/s41598-025-06344-8>.
7. SignLive [Electronic resource]. – Access mode: <https://sign-speak.com/solution>.
8. Nyckel [Electronic resource]. – Access mode: <https://www.nyckel.com/pretrained-classifiers/sign-language-identifier/>.
9. GoSign [Electronic resource]. – Access mode: <https://www.gosignthat.com/spell>.
10. The Python Tutorial [Electronic resource]. – Access mode: <https://docs.python.org/3/tutorial/index.html>.
11. What is Python [Electronic resource]. – Access mode: <https://www.pythontutorial.net/getting-started/what-is-python/>.

12. Sublime Text Unofficial Documentation [Electronic resource]. – Access mode: <https://docs.sublimetext.io/>.

13. Sublime Text [Electronic resource]. – Access mode: <https://www.techspot.com/downloads/5546-sublime-text.html>.

14. Sign Language MNIST [Electronic resource]. – Access mode: <https://www.kaggle.com/datasets/datamunge/sign-language-mnist>.

ДОДАТОК А
Технічне завдання

Вступ

Програмне забезпечення може використовуватися для розпізнавання жестової мови.

A.1 Підстава для розробки

Підставою для розробки є завдання на дипломну кваліфікаційну роботу на тему «Дослідження та програмна реалізація засобів розпізнавання жестової мови», затверджене наказом Національного університету «Запорізька політехніка» № 447 від 30 вересня 2025 р.

A.2 Призначення розробки

Програмний продукт призначений для розпізнавання жестової мови.

A.3 Основні вимоги до програми, що розробляється

A.3.1 Вимоги до функціональних характеристик

Програмне забезпечення для розпізнавання жестової мови забезпечує виконання таких функцій:

- підтримка можливості отримання відеопотоку з камери;
- підтримка можливості виявлення регіону інтересу, де розташовано мовний жест;
- можливість нормалізації зображення;
- можливість синтезу нейромережових моделей для розпізнавання мови жестів;
- підтримка можливості розпізнавання мовних жестів за допомогою навченої нейромережової моделі;
- підтримка можливості виведення розпізнаного жесту;

– можливість інформування щодо ймовірності правильного розпізнавання мовного жесту.

А.3.2 Вимоги до інтерфейсу програми

Інтерфейс програмного забезпечення для розпізнавання жестової мови повинен бути зручним для користувачів. Повинна бути забезпечена можливість роботи в візуальному режимі.

А.3.3 Вимоги до надійності

Програмне забезпечення для розпізнавання жестової мови повинно забезпечити надійне функціонування.

А.3.4 Умови експлуатації

Для експлуатації програмного забезпечення для розпізнавання жестової мови необхідна наявність персонального комп'ютера.

А.3.5 Вимоги до складу та параметрів технічних засобів

Для експлуатації програмного забезпечення для розпізнавання жестової мови необхідно таке апаратне та програмне забезпечення:

– процесор із багатоядерною архітектурою та тактовою частотою не нижче 2,5 ГГц для забезпечення швидкої обробки відеопотоку та роботи алгоритмів машинного навчання;

– оперативна пам'ять обсягом не менше 8 ГБ для стабільної роботи системи та завантаження моделей розпізнавання;

- графічний процесор для прискорення обробки нейронних мереж;
- вільний простір на жорсткому диску не менше 5 ГБ для встановлення програмного забезпечення та збереження тимчасових файлів;
- вебкамера з роздільною здатністю від 720p і вище, яка здатна працювати в режимі реального часу.

A.3.6 Вимоги до маркування і пакування

Програма для розпізнавання жестової мови може бути записана на будь-якому носії інформації.

На пакуванні повинна бути назва програми – «Програмне забезпечення для розпізнавання жестової мови».

A.4 Вхідні дані до роботи

Вхідними даними до програмного забезпечення для розпізнавання жестової мови є відеопотік з зображеннями користувачів та їх діями, які відображають жести жестової мови.

Вихідними даними програмного забезпечення для розпізнавання жестової мови є розпізнані жести.

ДОДАТОК Б
Фрагмент тексту програми

```

def rddt_dts():
    dts = np.load('sl_mnst_dts.npy')
    lbls = np.load('sl_mnst_lbls.npy')
    dts = dts.reshape(-1, 28, 28, 1).astype('float32') / 255.0
    lbls = tf.keras.utils.to_categorical(lbls, 24)
    return dts, lbls

def cr_mdl():
    mdl = Sequential()
    mdl.add(Conv2D(16, (3,3), activation='relu', padding='same',
input_shape=(28,28,1)))
    mdl.add(Conv2D(16, (3,3), activation='relu'))
    mdl.add(MaxPooling2D((2,2)))
    mdl.add(Conv2D(32, (3,3), activation='relu', padding='same'))
    mdl.add(Conv2D(32, (3,3), activation='relu'))
    mdl.add(MaxPooling2D((2,2)))
    mdl.add(Conv2D(64, (3,3), activation='relu', padding='same'))
    mdl.add(Conv2D(64, (3,3), activation='relu'))
    mdl.add(MaxPooling2D((2,2)))
    mdl.add(Flatten())
    mdl.add(Dense(128, activation='relu'))
    mdl.add(Dense(24, activation='softmax'))
    mdl.compile(optimizer='adam',
loss='categorical_crossentropy', metrics=['accuracy'])
    return mdl

def trn_mdl(mdl, dts, lbls):
    mdl.fit(dts, lbls, batch_size=64, epochs=10,
validation_split=0.1)
    return mdl

def prdct_jst(mdl, frm):
    frm_gray = cv2.cvtColor(frm, cv2.COLOR_BGR2GRAY)
    frm_rs = cv2.resize(frm_gray, (28,28))
    frm_rs = frm_rs.reshape(1,28,28,1).astype('float32') / 255.0
    prdct = mdl.predict(frm_rs)
    return np.argmax(prdct), np.max(prdct)

def cpt_roi(frm):
    frm_hsv = cv2.cvtColor(frm, cv2.COLOR_BGR2HSV)

```

```

lwr = np.array([0, 30, 60])
upr = np.array([20, 150, 255])
msk = cv2.inRange(frm_hsv, lwr, upr)
res = cv2.bitwise_and(frm, frm, mask=msk)
return res

def stt_cpt(mdl):
    cpt = cv2.VideoCapture(0)
    while True:
        rtd, frm = cpt.read()
        if not rtd:
            break
        roi = cpt_roi(frm)
        lbl, prb = prdct_jst(mdl, roi)
        cv2.putText(frm, f'Lbl: {lbl} Prb: {prb:.2f}', (10,30),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0,255,0), 2)
        cv2.imshow('SL RCGN', frm)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break
    cpt.release()
    cv2.destroyAllWindows()

def m_in():
    dts, lbls = rddt_dts()
    mdl = cr_mdl()
    mdl = trn_mdl(mdl, dts, lbls)
    stt_cpt(mdl)

def prpr_frm(frm):
    frm_clr = cv2.cvtColor(frm, cv2.COLOR_BGR2GRAY)
    frm_blr = cv2.GaussianBlur(frm_clr, (5,5), 0)
    frm_thr = cv2.adaptiveThreshold(frm_blr, 255,
cv2.ADAPTIVE_THRESH_GAUSSIAN_C,
                                cv2.THRESH_BINARY_INV, 11, 2)
    frm_edge = cv2.Canny(frm_thr, 50, 150)
    frm_dil = cv2.dilate(frm_edge, np.ones((3,3), np.uint8),
iterations=1)
    frm_er = cv2.erode(frm_dil, np.ones((2,2), np.uint8),
iterations=1)
    frm_rsz = cv2.resize(frm_er, (28,28))

```

```

frm_nrm = frm_rsz / 255.0
frm_nrm = np.expand_dims(frm_nrm, axis=-1)
frm_nrm = np.expand_dims(frm_nrm, axis=0)
return frm_nrm

def prdct_jst(mdl, frm):
    frm_prpr = prpr_frm(frm)
    prd = mdl.predict(frm_prpr)
    lbl_idx = np.argmax(prd)
    prb = prd[0][lbl_idx]
    lbl_map = {0:'A', 1:'B', 2:'C', 3:'D', 4:'E', 5:'F', 6:'G',
7:'H', 8:'I',
              9:'K', 10:'L', 11:'M', 12:'N', 13:'O', 14:'P',
15:'Q', 16:'R',
              17:'S', 18:'T', 19:'U', 20:'V', 21:'W', 22:'X',
23:'Y'}
    lbl = lbl_map.get(lbl_idx, 'Unknown')
    return lbl, prb

def prcs_vd_stream(mdl, cam_idx=0):
    cap = cv2.VideoCapture(cam_idx)
    while cap.isOpened():
        ret, frm = cap.read()
        if not ret:
            break
        frm_flip = cv2.flip(frm, 1)
        lbl, prb = prdct_jst(mdl, frm_flip)
        cv2.putText(frm_flip, f'{lbl} ({prb:.2f})', (10,30),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0,255,0), 2)
        cv2.imshow('Gesture Recognition', frm_flip)
        key = cv2.waitKey(1)
        if key == 27:
            break
    cap.release()
    cv2.destroyAllWindows()

def bld_trn_batch(dts, lbls, sz=(28,28)):
    X, Y = [], []
    for i, frm in enumerate(dts):
        frm_gray = cv2.cvtColor(frm, cv2.COLOR_BGR2GRAY)
        frm_rsz = cv2.resize(frm_gray, sz)

```

```

        frm_nrm = frm_rsz / 255.0
        X.append(frm_nrm)
        Y.append(lbls[i])
X = np.array(X).reshape(-1, sz[0], sz[1], 1)
Y = np.array(Y)
return X, Y

def vslz_frm_prdct(frm, lbl, prb):
    frm_cp = frm.copy()
    h, w = frm_cp.shape[:2]
    cv2.rectangle(frm_cp, (5, h-40), (w-5, h-5), (0, 0, 0), -1)
    cv2.putText(frm_cp, f'Gesture: {lbl} Prob: {prb:.2f}', (10, h-
10),
                cv2.FONT_HERSHEY_SIMPLEX, 0.7, (255, 255, 255), 2)
    cv2.imshow('Frame Visualization', frm_cp)
    cv2.waitKey(1)

def smth_prdcts(prdcts, alpha=0.6):
    smth_arr = []
    for i, prd in enumerate(prdcts):
        if i == 0:
            smth_arr.append(prd)
        else:
            smth_val = alpha*prd + (1-alpha)*smth_arr[-1]
            smth_arr.append(smth_val)
    return np.array(smth_arr)

def lg_gst_hstry(lbl, tm, fl='gst_hstry.txt'):
    with open(fl, 'a') as f:
        f.write(f'{tm}: {lbl}\n')

def clp_rgn_of_intr(frm):
    h, w = frm.shape[:2]
    x1, y1 = int(w*0.3), int(h*0.3)
    x2, y2 = int(w*0.7), int(h*0.7)
    rgn = frm[y1:y2, x1:x2]
    rgn = cv2.resize(rgn, (28, 28))
    return rgn

def prdct_w_rgn mdl, frm):
    rgn = clp_rgn_of_intr(frm)

```

```

lbl, prb = prdct_jst mdl, rgn
smth_prb = prb * 0.9 + 0.1
vslz_frm_prdct frm, lbl, smth_prb
lg_gst_hstry lbl, time.time()
return lbl

def dnn_trnsfrm_frm frm):
    frm_clr = cv2.cvtColor frm, cv2.COLOR_BGR2RGB)
    frm_rsz = cv2.resize frm_clr, (64, 64)
    frm_nrm = frm_rsz / 255.0
    frm_exp = np.expand_dims frm_nrm, axis=0)
    frm_flip = np.flip frm_exp, axis=2)
    frm_shft = np.roll frm_flip, shift=1, axis=1)
    frm_gs = cv2.cvtColor ((frm_shft[0]*255).astype(np.uint8),
cv2.COLOR_RGB2GRAY)
    frm_blr = cv2.GaussianBlur frm_gs, (5,5), 0)
    frm_thr = cv2.adaptiveThreshold frm_blr, 255,
cv2.ADAPTIVE_THRESH_MEAN_C,
                                cv2.THRESH_BINARY, 11, 2)
    frm_edge = cv2.Canny frm_thr, 30, 120)
    frm_dil = cv2.dilate frm_edge, np.ones((3,3), np.uint8),
iterations=2)
    frm_er = cv2.erode frm_dil, np.ones((2,2), np.uint8),
iterations=1)
    frm_norm = frm_er / 255.0
    frm_norm = np.expand_dims frm_norm, axis=-1)
    frm_norm = np.expand_dims frm_norm, axis=0)
    return frm_norm

def cmp_gst_prdcts prd1, prd2):
    max_len = max(len(prd1), len(prd2))
    prd1_ext = np.pad(prd1, (0, max_len - len(prd1)),
mode='edge')
    prd2_ext = np.pad(prd2, (0, max_len - len(prd2)),
mode='edge')
    diff = np.abs(prd1_ext - prd2_ext)
    smth = np.convolve(diff, np.ones(5)/5, mode='valid')
    avg_diff = np.mean(smth)
    if avg_diff < 0.1:
        corr_flag = True

```

```

else:
    corr_flag = False
    return corr_flag, avg_diff

def bld_frm_seq(vd_path, frm_step=1):
    cap = cv2.VideoCapture(vd_path)
    frms = []
    idx = 0
    while cap.isOpened():
        ret, frm = cap.read()
        if not ret:
            break
        if idx % frm_step == 0:
            frm_rsz = cv2.resize(frm, (64,64))
            frms.append(frm_rsz)
            idx += 1
    cap.release()
    frms_np = np.array(frms)
    frms_np = frms_np / 255.0
    return frms_np

def fltr_noise_prdcts(prdcts, threshold=0.5):
    fltrd = []
    for p in prdcts:
        if p[1] > threshold:
            fltrd.append(p)
        else:
            fltrd.append(('Unknown', p[1]))
    return fltrd

def mrg_rgn_frms(frm_lst):
    merged = np.zeros_like(frm_lst[0])
    cnt = np.zeros_like(frm_lst[0], dtype=np.float32)
    for frm in frm_lst:
        mask = (frm > 0).astype(np.float32)
        merged += frm * mask
        cnt += mask
    cnt[cnt == 0] = 1
    merged = (merged / cnt).astype(np.uint8)
    return merged

```

```

def bld_heatmap(frm_lst):
    heat = np.zeros(frm_lst[0].shape[:2], dtype=np.float32)
    for frm in frm_lst:
        gray = cv2.cvtColor(frm, cv2.COLOR_BGR2GRAY) / 255.0
        heat += gray
    heat /= len(frm_lst)
    heat_map = np.uint8(255 * heat)
    heat_map_color = cv2.applyColorMap(heat_map,
cv2.COLORMAP_JET)
    return heat_map_color

def bld_rgn_mask(frm):
    frm_hsv = cv2.cvtColor(frm, cv2.COLOR_BGR2HSV)
    lower = np.array([0, 30, 60])
    upper = np.array([20, 150, 255])
    mask = cv2.inRange(frm_hsv, lower, upper)
    mask_blr = cv2.GaussianBlur(mask, (5,5), 0)
    mask_er = cv2.erode(mask_blr, np.ones((3,3), np.uint8),
iterations=1)
    mask_dl = cv2.dilate(mask_er, np.ones((3,3), np.uint8),
iterations=2)
    return mask_dl

def bld_seq_prdcts mdl, frm_lst):
    prdcts = []
    for frm in frm_lst:
        frm_prpr = dnn_trnsfrm_frm(frm)
        lbl, prb = prdct_jst(mdl, frm_prpr)
        prdcts.append((lbl, prb))
    prdcts_smth = smth_prdcts([p[1] for p in prdcts])
    prdcts_final = [(prdcts[i][0], prdcts_smth[i]) for i in
range(len(prdcts))]
    return prdcts_final

```

ДОДАТОК В
Слайди презентації

Міністерство освіти і науки України
Національний університет «Запорізька політехніка»
Кафедра програмних засобів

Дослідження та програмна реалізація засобів розпізнавання жестової мови

Виконав: ст. гр. КНТ-214м

Олександр ОМ

Керівник:
к.т.н., доцент

Матвій ІЛ'ЯШЕНКО

Рисунок В.1 – Слайд 1

Об'єкт, предмет та мета роботи

Об'єкт дослідження – процеси обчислень, пов'язані з розпізнаванням мови жестів.

Предмет дослідження – засоби для розпізнавання жестової мови.

Метою роботи є дослідження та розробка програмного забезпечення для розпізнавання жестової мови.

Рисунок В.2 – Слайд 2

Завдання роботи

Для досягнення поставленої мети у кваліфікаційній роботі магістра необхідно розв'язати такі задачі:

- виконати аналіз предметної області та програмних засобів для розпізнавання жестової мови;
- розробити неймережеву модель для розпізнавання жестової мови;
- здійснити проектування програмного забезпечення для розпізнавання жестової мови;
- створити програмне забезпечення для розпізнавання жестової мови;
- виконати тестування розробленого програмного забезпечення для розпізнавання жестової мови.

3

Рисунок В.3 – Слайд 3

Порівняння ПЗ для розпізнавання жестової мови

Критерій порівняння	SignLive	Nyckel	GoSign
Легкість інтеграції через API	+–	+	+–
Підтримка розпізнавання у реальному часі	+	+–	+
Орієнтація на широке охоплення жестових мов	+–	–	+
Здатність до настроювання моделей	+–	+	+–
Інтеграція з різними платформами	+	+–	+–
Анімоване відтворення жестів	+	–	+

4

Рисунок В.4 – Слайд 4

Завдання роботи

При розробці програмного забезпечення для розпізнавання жестової мови необхідно забезпечити такі функціональні вимоги:

- підтримка можливості отримання відеопотоку з камери;
- підтримка можливості виявлення регіону інтересу, де розташовано мовний жест;
- можливість нормалізації зображення;
- можливість синтезу нейромережевих моделей для розпізнавання мови жестів;
- підтримка можливості розпізнавання мовних жестів за допомогою навченої нейромережевої моделі;
- підтримка можливості виведення розпізнаного жесту;
- можливість інформування щодо ймовірності правильного розпізнавання мовного жесту.

5

Рисунок В.5 – Слайд 5

Вибір мов програмування

Критерій порівняння мов програмування	Мова програмування		
	Python	C#	JavaScript
Простота та швидкість прототипування	+	+–	+
Підтримка бібліотек для роботи з засобами штучного інтелекту	+	+–	+–
Обробка зображень та відео	+	+–	–
Інтеграція з апаратними пристроями	+	+	–
Активна науково-технічна спільнота	+	+–	+–

6

Рисунок В.6 – Слайд 6

Порівняння середовищ розробки

Критерій порівняння середовищ розробки	Середовища розробки		
	Sublime Text	VS Code	PyCharm
Швидкість роботи та легкість запуску	+	+–	+–
Простота налаштування	+	+	+–
Зручність інтеграції з бібліотеками Python	+	+–	+–
Продуктивність при налагодженні коду	+	+	+–
Низькі системні вимоги	+	–	+–

7

Рисунок В.7 – Слайд 7

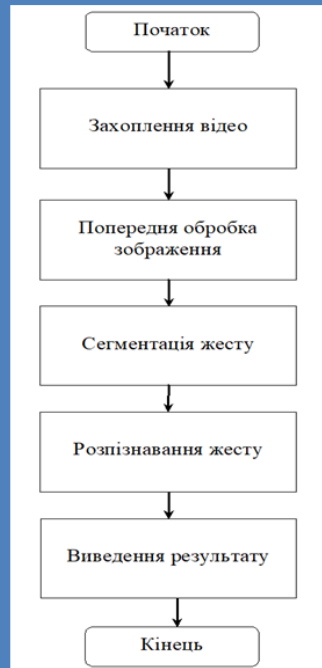
Структура програмного забезпечення для розпізнавання жестової мови



8

Рисунок В.8 – Слайд 8

Функціонування ПЗ для розпізнавання жестової мови



9

Рисунок В.9 – Слайд 9

Синтез нейромережевої моделі для розпізнавання жестової мови



Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 48, 48, 16)	448
conv2d_1 (Conv2D)	(None, 46, 46, 16)	2320
conv2d_2 (Conv2D)	(None, 44, 44, 16)	2320
max_pooling2d (MaxPooling2D)	(None, 22, 22, 16)	0
conv2d_3 (Conv2D)	(None, 20, 20, 32)	4640
conv2d_4 (Conv2D)	(None, 18, 18, 32)	9248
conv2d_5 (Conv2D)	(None, 16, 16, 32)	9248
max_pooling2d_1 (MaxPooling2D)	(None, 8, 8, 32)	0
conv2d_6 (Conv2D)	(None, 6, 6, 64)	18496
conv2d_7 (Conv2D)	(None, 4, 4, 64)	36928
conv2d_8 (Conv2D)	(None, 2, 2, 64)	36928
flatten (Flatten)	(None, 256)	0
dense (Dense)	(None, 128)	32896
dense_1 (Dense)	(None, 6)	774

Total params: 154246 (602.52 KB)
 Trainable params: 154246 (602.52 KB)
 Non-trainable params: 0 (0.00 Byte)



10

Рисунок В.10 – Слайд 10

Розпізнавання жестів за допомогою ПЗ

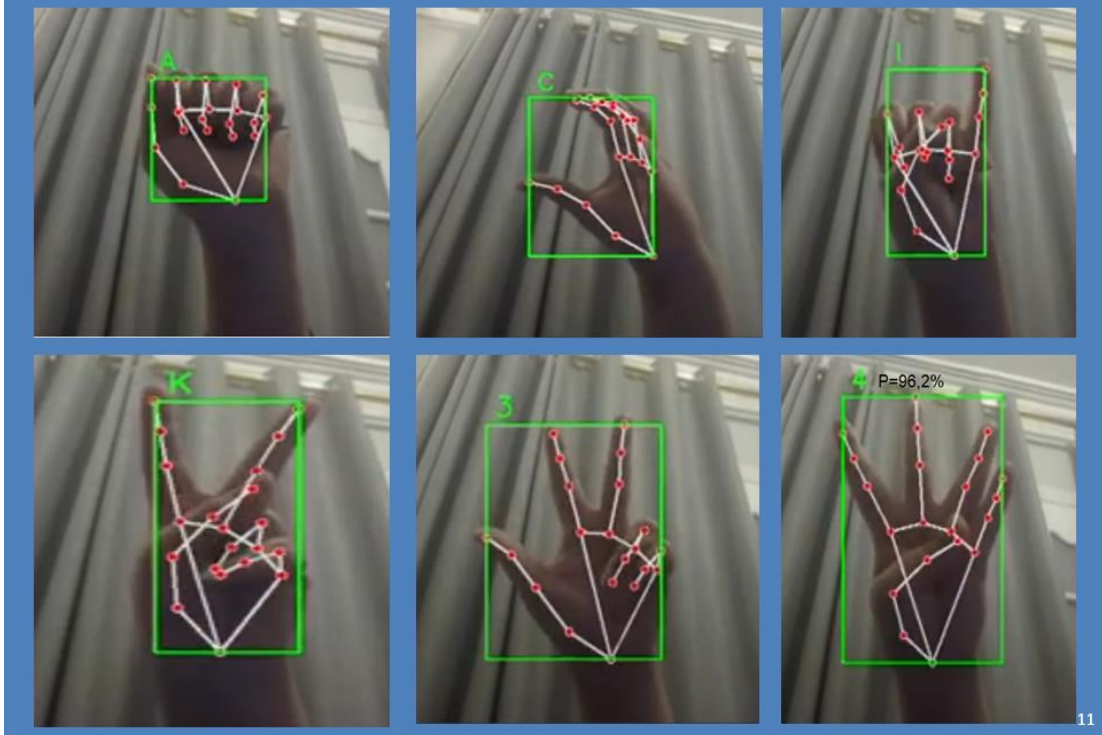


Рисунок В.11 – Слайд 11

Висновки

В результаті виконання дипломної кваліфікаційної роботи магістра було проаналізовано та досліджено процеси обчислень, пов'язані з розпізнаванням мови жестів.

Новизна роботи полягає у тому, що розроблено згорткову нейромережеву модель, яка являє собою набір взаємопов'язаних нейронів та дозволяє виконувати розпізнавання жестової мови.

Практичне значення роботи полягає у тому, що створено програмне забезпечення для розпізнавання жестової мови.

Результати тестування програмного забезпечення підтвердили його працездатність.

Усі завдання випускної дипломної роботи магістра повністю виконано.

Рисунок В.12 – Слайд 12