

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ШТУЧНОГО ІНТЕЛЕКТУ**

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Інтерактивний цифровий асистент із підтримкою
розпізнавання жестової мови та синтезу голосових відповідей»

на здобуття освітнього ступеня бакалавра
зі спеціальності 122 Комп'ютерні науки
освітньо-професійної програми Штучний інтелект

*Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання
на відповідне джерело*

_____ Марія ШИШУК
(підпис)

Виконала:
здобувачка вищої освіти
група ШІД-42

Марія ШИШУК

Керівник:
Д.т.н., доцент

Ольга ЗІНЧЕНКО

Рецензент:
науковий ступінь,
вчене звання

_____ Ім'я, ПРІЗВИЩЕ

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
Навчально-науковий інститут інформаційних технологій**

Кафедра Штучного інтелекту

Ступінь вищої освіти Бакалавр

Спеціальність 122 Комп'ютерні науки

Освітньо-професійна програма Штучний інтелект

ЗАТВЕРДЖУЮ

Завідувач кафедри Штучного інтелекту

_____ Ольга ЗІНЧЕНКО

«_____» _____ 2025 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Шишук Марії Олександрівни

1. Тема кваліфікаційної роботи: Інтерактивний цифровий асистент із підтримкою розпізнавання жестової мови та синтезу голосових відповідей
керівник кваліфікаційної роботи Ольга Зінченко д.т.н., доцент,
затверджені наказом Державного університету інформаційно-комунікаційних технологій від «24» лютого 2025 р. № 56
2. Строк подання кваліфікаційної роботи «23» травня 2025 р.
3. Вихідні дані до кваліфікаційної роботи: науково-технічна література, технічні бібліотеки мови програмування Python, документація Gemini API, спеціалізовані набори даних для розпізнавання жестової мови.
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
 1. Дослідження технологій розпізнавання мови жестів
 2. Побудова моделі машинного навчання для класифікації жестів
 3. Інтеграція з великою мовною моделлю Gemini API
 4. Імплементация системи синтезу мовлення
5. Перелік графічного матеріалу: *презентація*
 1. Використання бібліотеки mediapipe для розпізнавання жестів
 2. Архітектура RandomForestClassifier для класифікації жестів
 3. Архітектура інтеграції Gemini API
 4. Архітектура системи синтезу мовлення
6. Дата видачі завдання «25» лютого 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз наукових джерел за темою комп'ютерного зору та машинного навчання.	25.02-07.03.25	Виконано
2	Аналіз існуючих систем розпізнавання жестової мови та синтезу мовлення	10.03-13.03.25	Виконано
3	Дослідження методів машинного навчання для розпізнавання жестів	17.03-19.03.25	Виконано
4	Проектування архітектури та інтерфейсу розпізнавання жестів	20.03-28.03.25	Виконано
5	Розробка та реалізація системи	31.03-30.04.25	Виконано
6	Тестування та оцінка ефективності системи	01.05-02.05.25	Виконано
7	Оформлення роботи: вступ, висновки, реферат	05.05-14.05.25	Виконано
8	Розробка демонстраційних матеріалів	15.05-23.05.25	Виконано

Здобувачка вищої освіти

(підпис)

Марія ШИШУК

Керівник
кваліфікаційної роботи

(підпис)

Ольга ЗІНЧЕНКО

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра: 58 стор., 5 табл., 15 рис., 20 джерел.

Мета роботи – розробка та реалізація інтерактивного цифрового асистента, здатного розпізнавати жести рук на основі американської жестової мови (ASL) та надавати голосові відповіді, забезпечуючи інклюзивну двосторонню комунікацію.

Об'єкт дослідження – процес розпізнавання жестової мови та синтезу мовлення у взаємодії з цифровим асистентом.

Предмет дослідження – методи та алгоритми машинного навчання для ефективного розпізнавання жестів рук, а також інтеграція технологій синтезу мовлення для створення багатомодельного інтерфейсу цифрового асистента.

Короткий зміст роботи: У роботі проведено аналіз існуючих рішень та проблематики доступності технологій, а також специфіки американської жестової мови (ASL). Досліджено та обґрунтовано методи машинного навчання, зокрема RandomForestClassifier, для розпізнавання жестів. Для розробки системи було використано мову програмування Python та бібліотеку mediapipe. Розроблений інтерактивний цифровий асистент для двосторонньої комунікації між людиною та комп'ютером, що об'єднує розпізнавання жестової мови та синтез голосових відповідей.

КЛЮЧОВІ СЛОВА: РОЗПІЗНАВАННЯ ЖЕСТІВ, АМЕРИКАНСЬКА ЖЕСТОВА МОВА (ASL), СИНТЕЗ МОВЛЕННЯ (TEXT-TO-SPEECH), RANDOMFORESTCLASSIFIER, GEMINI API, MEDIAPIPE.

ABSTRACT

Text part of the bachelor's qualification work: 58 pages, 5 table, 15 pictures, 20 sources.

Purpose of the work – to develop and implement an interactive digital assistant capable of recognizing hand gestures based on American Sign Language (ASL) and providing voice responses, ensuring inclusive two-way communication.

Object of research – the process of sign language recognition and speech synthesis in interaction with a digital assistant.

Subject of research – machine learning methods and algorithms for effective hand gesture recognition, as well as the integration of speech synthesis technologies to create a multi-model digital assistant interface.

Summary of the work: The paper analyzes existing solutions and issues of technology accessibility, as well as the specifics of American Sign Language (ASL). Machine learning methods, in particular RandomForestClassifier, for gesture recognition are investigated and substantiated. The Python programming language and the mediapipe library were used to develop the system. An interactive digital assistant for two-way communication between humans and computers has been developed that combines sign language recognition and voice response synthesis.

KEYWORDS: GESTURE RECOGNITION, AMERICAN SIGN LANGUAGE (ASL), TEXT-TO-SPEECH SYNTHESIS, RANDOMFORESTCLASSIFIER, GEMINI API, MEDIAPIPE.

ЗМІСТ

ВСТУП.....	9
1 ТЕОРЕТИЧНІ ОСНОВИ СИСТЕМ РОЗПІЗНАВАННЯ ЖЕСТОВОЇ МОВИ	11
1.1 Аналіз проблематики області дослідження	11
1.2 Аналіз існуючих програмних рішень систем розпізнавання жестової мови.....	12
1.3 Специфіка роботи американської жестової мови	15
1.4 Основні підходи та методи розпізнавання жестів	17
2 ПРОЕКТУВАННЯ ТА АРХІТЕКТУРА СИСТЕМИ РОЗПІЗНАВАННЯ ЖЕСТОВОЇ МОВИ	21
2.1 Загальна архітектура системи розпізнавання жестів.....	21
2.2 Опис функціональних модулів системи розпізнавання жестової мови	22
2.3 Обґрунтування вибору класифікаційного алгоритму RandomForestClassifier	24
2.4 Алгоритм навчання моделі класифікації жестів.....	25
3 РЕАЛІЗАЦІЯ СИСТЕМИ РОЗПІЗНАВАННЯ ЖЕСТОВОЇ МОВИ.....	30
3.1. Середовище розробки та використовувані інструменти	30
3.2. Процес збору та підготовки даних для навчання моделі.....	33
3.3 Проектування графічного інтерфейсу системи розпізнавання мови жестів	37
3.4. Розробка та реалізація програмного інтерфейсу користувача.....	41
3.5 Експериментальні дослідження та аналіз результатів	45
ВИСНОВКИ	54
ПЕРЕЛІК ПОСИЛАНЬ.....	56
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ.....	58

ВСТУП

У сучасному світі цифрові технології дедалі більше інтегруються в повсякденне життя, забезпечуючи нові можливості для комунікації, доступу до інформації та взаємодії з електронними пристроями. Особливу роль у цьому процесі відіграють інтерактивні цифрові асистенти, які надають користувачам можливість взаємодії з технологіями за допомогою голосу, тексту або жестів. Проте для людей з порушеннями мовлення традиційні канали взаємодії з системами залишаються мало доступними або незручними.

Дана кваліфікаційна робота зосереджена на розробці інтерактивного цифрового асистента, який поєднує в собі розпізнавання жестової мови та синтез голосових відповідей. Такий підхід має на меті зробити технології більш інклюзивними, забезпечуючи зручний і природний інтерфейс для користувачів з особливими потребами. Розроблена система здатна розпізнавати жести, інтерпретувати їх у текстовий формат та озвучувати відповідь, що дозволяє забезпечити двосторонню комунікацію між користувачем і цифровим асистентом.

Актуальність теми зумовлена потребою у створенні доступних та інклюзивних технологій, які сприяють соціальній інтеграції осіб з обмеженими можливостями, а також загальним трендом на розвиток мультимедійних інтерфейсів користувача. Поєднання комп'ютерного зору, машинного навчання та технологій синтезу мовлення у межах однієї системи відкриває нові перспективи у галузі людської-комп'ютерної взаємодії.

У межах виконаної кваліфікаційної роботи було здійснено комплексний підхід до розробки застосунку для системи розпізнавання жестів рук, використовуючи мову програмування Python. Центральним елементом розробки стало проєктування та реалізація ефективного процесу класифікації, який дозволяє інтерпретувати жести рук у значущі дані. Окрім базової функції розпізнавання, застосунок було значно розширено для підвищення його функціональності та зручності використання. Зокрема, додано можливість створення власного набору даних (датасету) безпосередньо через інтерфейс програми. Це дозволяє

користувачам або дослідникам генерувати специфічні вибірки жестів, адаптовані під конкретні потреби або умови, що є критично важливим для навчання та адаптації моделі до різноманітних контекстів та стилів виконання жестів. Крім того, реалізовано функціонал перекладу розпізнаних жестів у простий текст, що є ключовим для візуалізації результатів та забезпечення зрозумілого зворотного зв'язку з користувачем.

1 ТЕОРЕТИЧНІ ОСНОВИ СИСТЕМ РОЗПІЗНАВАННЯ ЖЕСТОВОЇ МОВИ

1.1 Аналіз проблематики області дослідження

Сучасні інформаційні технології відіграють ключову роль у створенні інклюзивного середовища, що забезпечує рівні можливості для всіх членів суспільства. Особливої уваги потребують люди з порушеннями мовлення, для яких вербальна комунікація з оточенням або цифровими пристроями може бути істотно ускладненою або навіть повністю недоступною. Одним із перспективних напрямів вирішення цієї проблеми є використання мови жестів як альтернативного каналу взаємодії з комп'ютерними системами.

У рамках цієї кваліфікаційної роботи розглядається створення інноваційного інтерактивного цифрового асистента, здатного розпізнавати жести користувача за допомогою методів комп'ютерного зору, класифікувати їх із використанням алгоритмів машинного навчання та формувати голосову відповідь у відповідь на введений жест. Таким чином, система забезпечує двосторонній канал комунікації, в якому жести трансформуються у текст, а відповідь – в аудіоформат. Подібний підхід відкриває нові можливості для самостійного обслуговування, дистанційного навчання, професійної адаптації та соціалізації осіб із вадами мовлення.

Метою дослідження є розробка програмного забезпечення, яке виконує переклад жестової мови в текст із подальшою генерацією голосової відповіді. Для реалізації цієї мети були поставлені такі завдання: провести огляд існуючих підходів до розпізнавання жестів та синтезу мовлення; обґрунтувати вибір технологічного стека; спроектувати архітектуру системи; створити модель машинного навчання, здатну ідентифікувати жести на основі відеопотоку з веб-камери; реалізувати інтерфейс користувача; інтегрувати голосовий модуль та забезпечити обмін інформацією з великою мовною моделлю (Gemini API) для генерації змістовної текстової відповіді.

Актуальність цієї роботи обумовлена одночасно кількома чинниками: по-перше, соціальним запитом на доступні цифрові сервіси для людей з інвалідністю; по-друге, стрімким розвитком методів глибинного навчання, що відкриває нові

горизонти для розпізнавання складних патернів у поведінці людини; по-третє, широким поширенням мобільних та веб-технологій, що дає змогу впроваджувати подібні системи у повсякденне життя.

У результаті впровадження такого інтерактивного асистента з розпізнаванням жестів і голосовим зворотним зв'язком можна забезпечити нову якість взаємодії між людиною та машиною, зробити цифрове середовище більш гуманним, доступним і відкритим для кожного користувача, незалежно від його фізичних можливостей.

1.2 Аналіз програмних рішень систем розпізнавання жестової мови

На сучасному етапі розвитку технологій у галузі розпізнавання жестової мови існує низка програмних рішень, які реалізують окремі елементи систем комунікації для людей з порушенням мовлення. Ці рішення демонструють різні рівні складності, точності та функціональності. Водночас комплексні системи, що поєднують розпізнавання жестів та генерацію голосових відповідей, залишаються малорозвиненими або перебувають у стадії дослідних проєктів.

Одним із найбільш відомих проєктів є SignAll, який використовує декілька камер, глибинні сенсори та алгоритми комп'ютерного зору для автоматичного перекладу американської жестової мови (ASL) у текст. Ця система демонструє високу точність та підтримує контекстно-залежну обробку, однак вимагає складного апаратного забезпечення та дорогого програмного забезпечення, що обмежує її застосування в побуті чи для індивідуального використання.

Серед сучасних комерційних систем, що реалізують автоматичне розпізнавання жестової мови, варто виокремити KinTrans — технічне рішення, спрямоване на забезпечення комунікації між людьми з порушенням слуху та чуючими. Система використовує глибинні сенсори для фіксації положення рук, пальців, тіла та голови у тривимірному просторі, що дозволяє точно розпізнавати як статичні, так і динамічні жести, притаманні природному жестовому мовленню. Особливістю KinTrans є урахування повної тілесної експресії, а не лише рухів кисті, що наближає обробку до реальних комунікативних ситуацій. Вхідні дані

опрацьовуються спеціалізованими алгоритмами, які перетворюють жести у текст або голосові повідомлення, забезпечуючи двосторонній зв'язок у режимі реального часу. Хоча система демонструє високу точність розпізнавання та адаптивність до різних середовищ, її повноцінне функціонування залежить від наявності спеціалізованого апаратного забезпечення, що суттєво обмежує можливості її використання в особистому або побутовому контексті.

Також варто згадати мобільні застосунки, як-от Hand Talk, який забезпечує переклад тексту або голосу у жестову мову з використанням 3D-анімації. Цей застосунок орієнтований на зворотну функцію — від голосу до жесту — й не підтримує розпізнавання жестової мови з боку користувача. У цьому сенсі його застосування для глухих людей обмежене.

У рамках даної роботи було проведено порівняльний аналіз переваг і недоліків застосунків SignAll, KinTrans та Hand Talk (табл. 1.1).

Аналіз сучасних розробок у сфері розпізнавання жестової мови свідчить про наявність технічно потужних, але малодоступних рішень. Такі системи, як SignAll та KinTrans, демонструють високу точність, ефективне виявлення жестів та здатність до контекстної інтерпретації, проте мають суттєвий недолік — потребу в складному та дорогому апаратному забезпеченні. Цей фактор істотно обмежує їх використання у повсякденному житті, звужуючи сферу застосування переважно до спеціалізованих установ, навчальних центрів або комерційних середовищ.

З іншого боку, мобільні рішення на кшталт Hand Talk показують переваги у сфері зворотного перекладу — перетворення тексту чи голосу у жестову мову за допомогою віртуального 3D-аватара. Однак, через відсутність можливості розпізнавати жести користувача, ці застосунки не можуть повноцінно забезпечити комунікацію для людей з порушенням мовлення. Таким чином, їх функціональність є односторонньою та більше орієнтованою на потреби широкого загалу, які прагнуть зрозуміти жестову мову.

З урахуванням виявлених недоліків можна зробити висновок, що на ринку фактично відсутні доступні, універсальні системи, здатні забезпечити двосторонній обмін інформацією — розпізнавання жестів із наступною генерацією

голосової відповіді — без залучення дорогої або складної апаратної інфраструктури.

Таблиця 1.1 Порівняльна характеристика сучасних систем розпізнавання жестової мови

Назва системи	Тип пристрою / платформи	Основні функції	Методи та технології	Переваги	Обмеження
SignAll	Стационарна система з камерами та глибинними сенсорами	Автоматичний переклад жестової мови (ASL) у текст	Комп'ютерний зір, глибинне навчання, багатоекранна обробка	Висока точність, підтримка контекстної інтерпретації, розпізнавання у реальному часі	Висока вартість, потреба у складному обладнанні, складна установка
KinTrans	Стационарна інтерактивна система	Переклад жестів у текст або голос, підтримка двосторонньої комунікації	3D-аналіз положення тіла, глибинні сенсори, спеціалізовані алгоритми	Повне врахування тілесної експресії, реалістичне спілкування, точність розпізнавання	Вимоги до апаратного забезпечення, висока вартість, не персоналізована платформа
Hand Talk	Мобільний застосунок (Android, iOS)	Переклад голосу або тексту у жестову мову через 3D-анімацію	3D-анімація, синтез жестів, віртуальний аватар	Доступність на смартфонах, легкість у використанні, популярність серед освітніх програм	Не розпізнає жести користувача, одностороння комунікація, не підтримує повноцінний зворотний зв'язок

З урахуванням виявлених недоліків можна зробити висновок, що на ринку фактично відсутні доступні, універсальні системи, здатні забезпечити двосторонній обмін інформацією — розпізнавання жестів із наступною генерацією голосової відповіді — без залучення дорогої або складної апаратної інфраструктури.

Саме на подолання цієї прогалини спрямована дана кваліфікаційна робота. Її метою є створення простого у використанні цифрового асистента, що дозволить людям з порушенням мовлення ефективно комунікувати з навколишнім

середовищем за допомогою жестової мови, без бар'єрів, технічних обмежень та високих витрат.

Враховуючи обмеження сучасних програм у сфері розпізнавання жестової мови, розроблений у межах даної роботи застосунок покликаний усунути ключові недоліки наявних рішень та запропонувати більш доступну й функціональну альтернативу. На відміну від систем SignAll та KinTrans, які потребують складного й дорогого технічного оснащення, зокрема глибинних сенсорів і кількох камер, даний застосунок побудований з орієнтацією на використання стандартної веб-камери. Це значно спрощує умови впровадження технології, знижує витрати та відкриває можливість її використання у звичайних побутових умовах.

Крім того, якщо Hand Talk працює лише в одному напрямку — перетворення голосу або тексту в жестову мову за допомогою 3D-анімації — то в основі даного підходу лежить реалізація повноцінної двосторонньої комунікації. Застосунок не лише розпізнає жести користувача й трансформує їх у текст, але також генерує озвучену відповідь, створюючи повноцінну взаємодію між людиною з обмеженими можливостями та цифровим асистентом. Це забезпечує суттєву перевагу в плані функціональності та зручності використання.

Особливістю розробки є також розширення базового функціоналу стандартного жестового алфавіту (ASL) за рахунок можливості інтеграції власних жестів. Такий підхід дозволяє користувачам призначати унікальні жести для запуску певних функцій або команд, що значно підвищує гнучкість системи та адаптивність до індивідуальних потреб. У результаті розробка поєднує у собі технічну простоту, інклюзивність і розширений функціонал, що виводить її за межі обмежень наявних комерційних систем і робить придатною для щоденного використання широким колом користувачів.

1.3 Специфіка роботи американської жестової мови

Американська жестова мова (ASL) є не лише засобом комунікації, а й повноцінною природною мовою з власною граматиною, лексикою та структурою. Вона займає провідне місце серед жестових мов світу й виконує роль основного

інструмента спілкування для мільйонів глухих і слабочуючих людей, переважно у Сполучених Штатах Америки та англomовних частинах Канади. На відміну від допоміжних комунікативних засобів, ASL є самостійною мовною системою, сформованою під впливом історичних, культурних і соціальних чинників.

Як і будь-яка інша природна мова, ASL має регіональні діалекти та варіації, що виникли у результаті локальних освітніх підходів і культурних особливостей. Це ускладнює автоматичне розпізнавання жестів, оскільки навіть незначні відмінності у русі чи положенні рук можуть змінювати значення або ускладнювати інтерпретацію жесту. Попри це, ASL активно визнається на офіційному рівні в багатьох штатах США, широко використовується в системах освіти, медіа, державних установах і міжкультурній взаємодії між глухими спільнотами з різних країн, що підтверджує її статус повноцінної, соціально важливої мови.

Важливим компонентом ASL є дактилологія — система, у якій кожна літера латинського алфавіту відповідає певній конфігурації пальців однієї руки (рис. 1.1). Цей алфавіт використовується для передавання імен, назв, технічних термінів або слів, які не мають відповідника у жестовій лексиці. Оскільки алфавітні жести є статичними, вони добре підходять для навчання моделей розпізнавання на базовому рівні, особливо у рамках систем, що працюють зі стандартною камерою та обмеженим набором функцій.

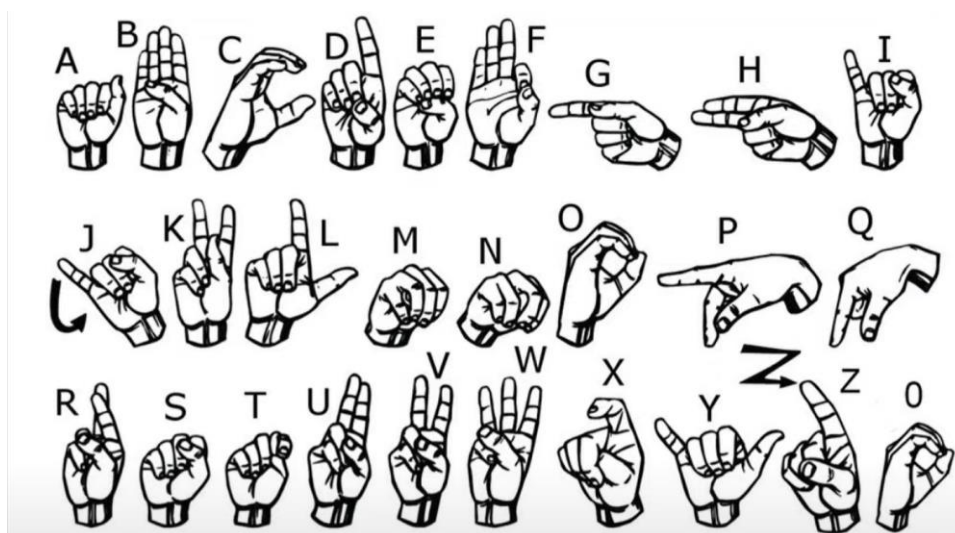


Рис. 1.1 Алфавіт американської мови жестів

Однак навіть при роботі з алфавітними жестами зберігається складність через дрібні відмінності у положенні пальців, можливі варіації між користувачами, а також зовнішні умови зйомки, які впливають на якість розпізнавання. Наприклад, деякі літери мають схожі конфігурації (як-от «М», «N» і «Т»), що потребує високої точності від алгоритмів обробки зображень і навчання моделі на великому наборі зразків.

Завдання автоматичного розпізнавання жестової мови загалом є складним і багатогранним. Воно потребує точного аналізу як статичних жестів, які ґрунтуються на фіксованому положенні рук, так і динамічних, де важливу роль відіграє рух у просторі та часі. Багато жестів подібні між собою, відрізняючись лише деталями — наприклад, положенням одного пальця або напрямком руху долоні, — що вимагає високої роздільної здатності аналізу та глибокого розуміння контексту.

Крім того, точність розпізнавання суттєво залежить від низки зовнішніх факторів, серед яких — анатомічні особливості користувачів (розмір і форма рук), швидкість виконання жестів, якість освітлення, фонові об'єкти та рівень шуму у відеопотоці. Усі ці параметри впливають на стабільність і точність інтерпретації зображення, що робить задачу ще більш складною.

У цьому контексті надзвичайно перспективним є використання методів штучного інтелекту, зокрема алгоритмів комп'ютерного зору та глибинного навчання. Завдяки здатності аналізувати великі обсяги відеоданих, виявляти приховані закономірності у траєкторіях рухів і створювати адаптивні моделі, ці технології відкривають нові можливості для розробки точних і гнучких систем розпізнавання жестової мови, здатних ефективно функціонувати в реальних, змінних умовах.

1.4 Основні підходи та методи розпізнавання жестів

У процесі виконання даної кваліфікаційної роботи було реалізовано низку сучасних методів та технологічних підходів, що дозволили створити ефективну систему інтерактивного цифрового асистента з можливістю розпізнавання

жестової мови та генерації голосової відповіді. На початковому етапі дослідження було здійснено ґрунтовний аналіз наукових публікацій, технічної документації та сучасних рішень у сфері комп'ютерного зору, глибокого навчання та мультимодальної взаємодії. Це дало змогу виявити найбільш перспективні технології, зокрема у контексті розпізнавання жестів, оцінити переваги й недоліки вже існуючих підходів, а також сформувану обґрунтовану методологічну базу для реалізації власної моделі.

Для отримання даних з камери та обробки відеопотоку було використано засоби комп'ютерного зору, що дали змогу виокремлювати зображення рук у кадрі, очищувати їх від шумів та приводити до формату, придатного для аналізу. Ключову роль у цьому відіграла бібліотека OpenCV, яка забезпечила попередню обробку вхідного відео, підвищення чіткості контурів та стабільне захоплення жестів. Отримані дані стали основою для створення власного навчального датасету, необхідного для побудови моделі класифікації.

Для задачі розпізнавання жестів було розглянуто різні алгоритми машинного навчання, серед яких метод опорних векторів (SVM), дерева рішень, метод k-ближчих сусідів (KNN) та ансамблеві методи (табл. 1.2). Найбільш доцільним з огляду на точність, швидкість обробки та стійкість до шумів виявився RandomForestClassifier, який і було обрано як основний класифікаційний алгоритм у межах проєкту. Такий вибір дозволив досягти прийняттого балансу між ефективністю моделі та простотою її реалізації.

Особливу увагу було приділено використанню гібридних рішень, що поєднують переваги кількох підходів. Так, модуль MediaPipe Hands, який інтегрує методи глибокого навчання для трекінгу рук та побудови ключових точок скелету, дозволив отримати якісні ознаки жестів, які згодом подавались на вхід класифікатору. Такий підхід забезпечив високу точність і стабільність розпізнавання навіть у складних умовах, зокрема при різному освітленні або незначних варіаціях у положенні рук.

Таблиця 1.2 Порівняльна характеристика алгоритмів машинного навчання для задачі розпізнавання жестів

Алгоритм	Точність	Швидкість обробки	Стійкість до шумів	Простота реалізації	Коментар
Метод опорних векторів (SVM)	Висока	Середня	Середня	Середня	Ефективний для малих та середніх наборів даних, чутливий до параметрів
Дерева рішень	Середня	Висока	Низька	Висока	Швидкий у навчанні, але схильний до перенавчання
Метод k-ближчих сусідів (KNN)	Середня–висока	Низька	Середня	Висока	Простий у реалізації, але повільний на великих даних
Ансамблеві методи (Random Forest)	Висока	Висока	Висока	Середня	Забезпечує хороший баланс точності й стійкості, рекомендований для практичного використання

В експериментальній частині було реалізовано функціональний прототип цифрового асистента, що, окрім розпізнавання жестів, включає можливість генерації голосової відповіді. Цей модуль став важливою складовою системи, оскільки надає зворотний зв'язок користувачеві в аудіоформаті, забезпечуючи тим самим інклюзивність і зручність у використанні. Реалізоване рішення демонструє переваги мультимодального підходу та підтверджує доцільність поєднання жестової мови з голосовими технологіями у межах єдиного програмного середовища.

У ході дослідження було також розглянуто сучасний стан розробки у цій галузі. Було встановлено, що більшість існуючих рішень або є надто складними у впровадженні, або обмеженими у функціональності. Наявні системи часто не підтримують одночасно розпізнавання жестів і генерацію голосової відповіді, що знижує їхню ефективність у реальних умовах використання. Запропонована в цій

роботі система вирізняється тим, що поєднує обидві функції у зручному й доступному для користувача форматі, забезпечуючи взаємодію з цифровим середовищем у реальному часі.

Модель, розроблена в межах проєкту, базується на комбінації модуля MediaPipe для детекції та трекінгу рук, алгоритму RandomForest для класифікації жестів та модуля синтезу мовлення для озвучення результатів. Такий інтегрований підхід дозволяє створити функціональний, надійний і водночас технічно нескладний прототип цифрового асистента, орієнтованого на потреби користувачів з порушеннями мовлення.

Таким чином, проведене дослідження підтвердило актуальність обраної тематики та продемонструвало практичну цінність розробленої системи. Запропоноване рішення має значний потенціал для подальшого розвитку, масштабування та впровадження у сфери, де необхідна доступна та інклюзивна комунікація між людиною та цифровими пристроями.

2 ПРОЕКТУВАННЯ ТА АРХІТЕКТУРА СИСТЕМИ РОЗПІЗНАВАННЯ ЖЕСТОВОЇ МОВИ

2.1 Загальна архітектура системи розпізнавання жестів

Загальна архітектура системи розпізнавання жестової мови являє собою комплексну структуру, що забезпечує послідовну та взаємопов'язану обробку візуальних даних для ідентифікації жестів у реальному часі (рис. 2.1). Ця архітектура спроектована таким чином, щоб кожен модуль виконував специфічну функцію, що дозволяє досягти високої ефективності та гнучкості системи.

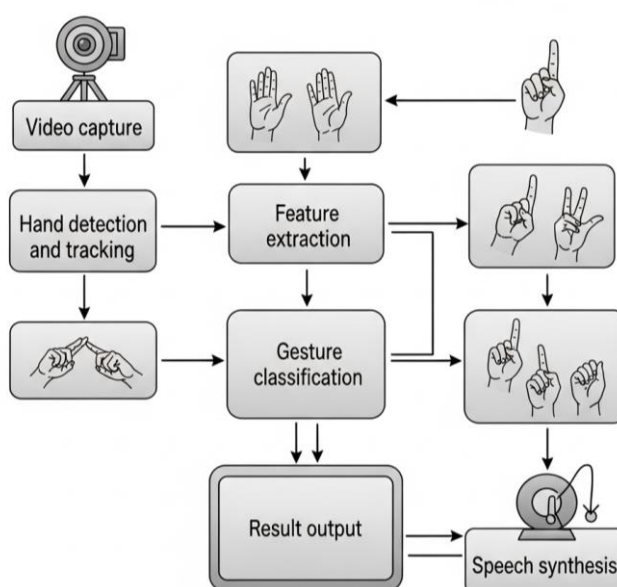


Рис. 2.1 Загальна архітектура системи розпізнавання жестів

Функціонування системи починається з модуля захоплення відео з камери. Це початковий етап, на якому відбувається отримання необробленого відеопотоку з веб-камери користувача. Захоплені окремі кадри є вхідними даними для подальших етапів обробки.

Наступним критично важливим етапом є модуль виявлення та трекінгу рук. Отримані кадри аналізуються для ідентифікації наявності однієї або обох рук у полі зору. Після виявлення, цей модуль безперервно відстежує рух рук, фіксуючи їхнє положення у просторі.

Відстежені дані про положення рук передаються до модуля вилучення ознак жестів. Цей модуль відповідає за трансформацію сирих координат рухів рук у

більш абстрактні та значущі ознаки. Ці ознаки можуть включати нормалізовані координати ключових точок, кути між пальцями, швидкості рухів та інші параметри, що є важливими для розрізнення різних жестів. Цей етап критично важливий для створення інваріантного до масштабу, положення та орієнтації жесту представлення.

Вилучені та підготовлені ознаки подаються на вхід модуля класифікації жестів. Це є центральним компонентом системи, де відбувається безпосереднє розпізнавання жесту. Модуль класифікації використовує попередньо навчену модель (у даному випадку, `RandomForestClassifier`), яка на основі отриманих ознак визначає, до якого класу жестів (наприклад, яка літера алфавіту) належить поточний рух руки.

Після успішної класифікації, розпізнаний жест може бути виведений до модуля виведення результатів на екран, а також до модуля запам'ятовування та виведення літер у спеціальне поле. З цього спеціального поля, дані можуть бути надіслані до Gemini API для додаткової обробки або розширення функціоналу. Після обробки Gemini API, результати можуть бути представлені у вигляді виведення текстової відповіді або генерації голосової відповіді.

Таким чином, загальна архітектура системи розпізнавання жестів базується на послідовному конвеєрі обробки даних, де кожен модуль виконує спеціалізовану функцію, від захоплення візуальних даних до їхнього перетворення у зрозумілий для людини формат. Ефективна взаємодія між цими модулями забезпечує стабільну, точну та своєчасну роботу всієї системи розпізнавання жестової мови.

2.2 Опис функціональних модулів системи розпізнавання жестової мови

Архітектура розробленої системи розпізнавання жестів базується на послідовному виконанні кількох функціональних модулів, які взаємодіють між собою для забезпечення стабільної та точної роботи в реальному часі (рис. 2.1). Кожен модуль виконує специфічну роль у загальному конвеєрі обробки даних, ілюструючи потокову обробку від вхідного кадру до кінцевого прогнозу.

Процес розпізнавання жестової мови в системі розпочинається з етапу захоплення відеопотоку за допомогою веб-камери. Цей модуль виконує первинне збирання вхідних даних у реальному часі та є вихідною точкою всього обчислювального процесу. Для організації відеозахоплення використовується бібліотека OpenCV, яка дозволяє отримувати послідовність кадрів із потоку в режимі реального часу. Зібрані кадри надходять до наступного етапу для обробки та аналізу.

На другому етапі система передає відеокадри до модуля виявлення та трекінгу рук, який відповідає за локалізацію рук у кожному кадрі. Для цього використовується бібліотека MediaPipe Hands — потужний інструмент від Google, призначений для високоточного визначення положення рук. Цей модуль виявляє одну або обидві руки в кадрі та визначає координати 21 ключової точки (орієнтиру) для кожної руки. Виявлені орієнтири відображають просторову позицію суглобів пальців, долоні, а також кінчиків пальців, що дозволяє сформувати тривимірну модель руки користувача.

Далі отримані координати передаються до модуля вилучення ознак. На цьому етапі здійснюється попередня обробка даних: координати нормалізуються та масштабуються для досягнення інваріантності до масштабу, позиції та орієнтації руки в кадрі. Це забезпечує стабільність роботи моделі в умовах різного освітлення, кута зйомки або фізичних відмінностей між користувачами. У процесі формування вектору ознак система може також обчислювати додаткові характеристики, зокрема: кути між фалангами пальців, відстані між орієнтирами та інші геометричні співвідношення. У результаті формується числовий вектор, який слугує вхідними даними для класифікаційної моделі.

Модуль класифікації жестів отримує на вхід сформований вектор ознак і виконує розпізнавання відповідного жесту. Для класифікації використовується ансамблевий метод — Random Forest Classifier, який складається з множини дерев рішень. Кожне дерево незалежно приймає рішення щодо класу жесту, після чого обирається той клас, що отримав більшість голосів. Модель була попередньо навчена на спеціально зібраному датасеті, що містить приклади жестів у форматі

координат, що забезпечує високу точність розпізнавання навіть за наявності варіацій у виконанні.

Після успішної класифікації результат передається до модуля виведення. Тут відбувається трансформація ідентифікованого класу у відповідне текстове повідомлення, яке демонструється на екрані користувача. Окрім цього, розпізнаний текст може бути озвучений за допомогою модуля синтезу мовлення. У проєкті реалізовано інтеграцію з бібліотекою pyttsx3, яка забезпечує автономну генерацію голосових відповідей без необхідності підключення до інтернету. Це дає змогу не лише надати додаткове підтвердження класифікованого жесту, але й зробити систему зручнішою для користувачів з різними потребами, зокрема для тих, хто краще сприймає інформацію на слух.

Таким чином, система реалізує повний цикл обробки вхідного відео: від захоплення сирих кадрів і виявлення рук до розпізнавання жесту, генерації текстової та голосової відповіді. Такий підхід поєднує в собі переваги комп'ютерного зору, машинного навчання та голосових технологій, забезпечуючи зручний і доступний спосіб комунікації між людиною та цифровим інтерфейсом.

2.3 Обґрунтування вибору класифікаційного алгоритму

RandomForestClassifier

У загальній структурній схемі системи розпізнавання жестів важливе місце займає модуль класифікації, для реалізації якого обрано алгоритм RandomForestClassifier — потужний ансамблевий метод машинного навчання, який демонструє високу точність при багатокласовій класифікації. Алгоритм належить до моделей із учителем і ґрунтується на ідеї поєднання великої кількості дерев рішень, кожне з яких навчається на випадковій вибірці навчальних даних та використовує випадкову підмножину ознак при розгалуженнях. Підсумковий результат формується на основі принципу "більшості голосів" — клас, що отримав найбільшу кількість голосів серед дерев, приймається за остаточне передбачення.

Такий підхід дозволяє досягти кількох важливих переваг: зменшення ризику перенавчання, підвищення стійкості до шумів у даних, а також загальне

покращення узагальнювальної здатності моделі. Завдяки випадковості у виборі підмножин і ознак, моделі в ансамблі залишаються різномірними, що сприяє зменшенню кореляції між помилками окремих дерев.

На концептуальному рівні кожне дерево в Random Forest — це незалежна модель, яка приймає рішення шляхом побудови розгалуженої структури з "вузлів" і "листіків". Кожен вузол відповідає за розділення даних за певним критерієм, наприклад, "якщо значення ознаки x менше за 0.5, перейти ліворуч, інакше — праворуч". Такий процес повторюється ітеративно, поки не буде досягнуто листка дерева, який відповідає певному класу. Модель приймає рішення шляхом "руху" від кореня дерева до листка на основі значень ознак у вхідному векторі.

Хоча окреме дерево рішень досить просте для інтерпретації, воно може мати низьку стійкість до змін у навчальних даних, тобто бути нестабільним і схильним до перенавчання. У свою чергу, Random Forest долає цей недолік, поєднуючи велику кількість дерев у єдиний ансамбль, тим самим створюючи більш стабільну і точну модель.

Random Forest може бути застосований як для класифікації, так і для регресійних завдань [2]. У межах цього проєкту алгоритм використовується для класифікації жестів, що представлені у вигляді числових векторів ознак, отриманих після обробки координат ключових точок руки. Його ефективність у багатовимірних просторах ознак, здатність працювати з обмежено очищеними даними та нечіткими межами між класами робить його особливо доречним вибором для задач розпізнавання жестової мови.

Таким чином, модуль класифікації на основі Random Forest є критично важливим компонентом системи, забезпечуючи надійне розпізнавання жестів у режимі реального часу, незалежно від варіативності виконання жестів користувачем, різниці в освітленні чи фоновому шумі у відеопотоці.

2.4 Алгоритм навчання моделі класифікації жестів

Алгоритм навчання моделі класифікації жестів є ключовим елементом у загальній архітектурі системи розпізнавання жестової мови. Його реалізація

побудована на взаємодії низки функціональних модулів, які працюють узгоджено, забезпечуючи обробку інформації в реальному часі. Центральною частиною цієї архітектури виступає модуль класифікації, що використовує алгоритм RandomForestClassifier. Цей метод, який належить до класу ансамблевих алгоритмів машинного навчання, виконує багатокласову класифікацію на основі аналізу координат ключових точок руки користувача (рис. 2.2).

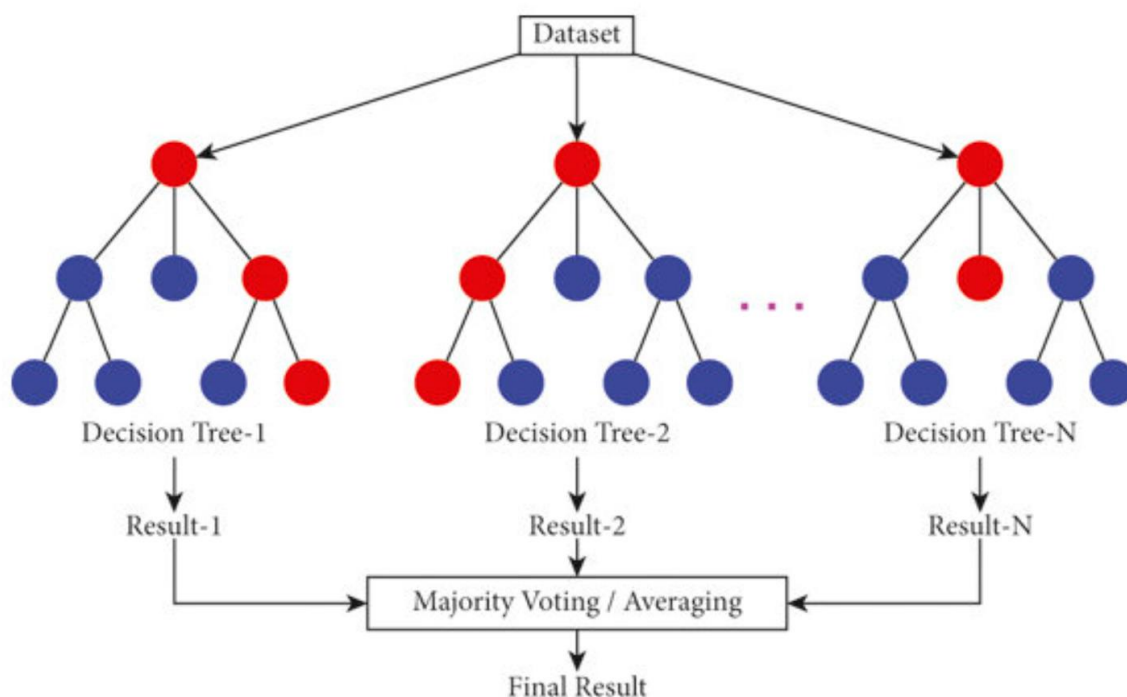


Рис. 2.2 Візуальне представлення класифікатора випадкового лісу

Робота системи починається з отримання відеопотоку за допомогою веб-камери. Захоплення та обробку відео виконує бібліотека OpenCV, яка дозволяє виділяти окремі кадри та передавати їх для подальшого аналізу. Кожен кадр надходить до модуля виявлення рук, у якому застосовується бібліотека MediaPipe Hands. Цей компонент визначає на зображенні одну або дві руки та обчислює координати 21 ключової точки для кожної з них. Отримані орієнтири відповідають розташуванню суглобів і кінчиків пальців у тривимірному просторі, що дозволяє точно відтворити положення руки.

Після виявлення ключових точок виконується вилучення ознак. Тривимірні координати кожного орієнтиру нормалізуються, що усуває залежність від розміру

руки, її положення у кадрі чи орієнтації. На основі нормалізованих даних формується вектор ознак, який слугує числовим представленням положення руки. Цей вектор може включати як координати (x, y, z), так і похідні характеристики, зокрема кути між сегментами пальців або відстані між вибраними точками.

Далі сформований вектор ознак подається на вхід до заздалегідь навченої моделі класифікації. Алгоритм `RandomForestClassifier`, який було натреновано на великому наборі прикладів жестів, аналізує ознаки та генерує передбачення. Кожне дерево в ансамблі приймає окреме рішення щодо класу, до якого може належати поточний жест. Остаточний результат визначається шляхом голосування: клас, який отримав найбільшу кількість голосів від дерев, вважається кінцевим передбаченням.

Після визначення класу розпізнаний жест передається у модуль відображення результату. Тут передбачений текстовий опис жесту виводиться на екран користувача. Крім того, текст може бути автоматично озвучений за допомогою модуля синтезу мовлення. Для реалізації цієї функціональності інтегровано бібліотеку `pyttsx3`, яка забезпечує конвертацію текстового повідомлення у природне мовлення.

З технічної точки зору, робота `Random Forest` ґрунтується на сукупності дерев рішень. Кожне дерево формує окреме рішення на основі функції розділення простору ознак, яка реалізується через правила виду «якщо... тоді...». Остаточне передбачення алгоритму визначається більшістю голосів дерев, які входять до складу ансамблю.

У процесі побудови кожного дерева використовується бутстреп-вибірка навчальних даних, а на кожному вузлі дерева обирається випадкова підмножина ознак для пошуку оптимального критерію розділення, наприклад, за допомогою мінімізації критерію індексу Джині (2.1):

$$Gini(t) = 1 - \sum_{i=1}^C p_i^2 \quad (2.1)$$

де p_i — частка об'єктів класу i у вузлі t , а C — загальна кількість класів.

Таким чином, реалізована система розпізнавання жестів є поєднанням комп'ютерного зору, методів машинного навчання та технологій синтезу мовлення. Завдяки використанню RandomForestClassifier вдається досягти високої точності класифікації, стійкості до шумів у вхідних даних та швидкої обробки, що особливо важливо для застосування в умовах реального часу.

У результаті, було всебічно розглянуто процес проектування системи розпізнавання жестової мови, що є одним із ключових етапів у створенні ефективного інтерфейсу взаємодії між людиною та комп'ютером. Встановлення архітектури системи дозволило визначити логічну послідовність обробки даних, від захоплення відеопотоку до остаточного представлення результатів, і заклало теоретичну та технічну основу для реалізації прототипу.

Розроблена архітектура системи включає низку послідовних функціональних модулів, кожен з яких виконує специфічне завдання та забезпечує узгоджену взаємодію з іншими компонентами системи. На першому етапі здійснюється захоплення відеопотоку за допомогою бібліотеки OpenCV, яка дозволяє проводити обробку зображень у режимі реального часу. Отримані кадри передаються до модуля виявлення рук, що реалізований на основі бібліотеки MediaPipe Hands. Цей модуль виконує просторове позиціонування кистей рук та визначає координати 21 ключової точки для кожної з виявлених рук, що є базисом для подальшого аналізу.

На наступному етапі координати ключових точок нормалізуються та перетворюються у вектори ознак. Така обробка забезпечує інваріантність до зміни масштабу, положення або повороту руки в кадрі, що є критично важливим для забезпечення стабільності результатів класифікації. Отримані вектори ознак є вхідними даними для класифікаційного модуля, в основі якого лежить алгоритм ансамблевого навчання — RandomForestClassifier.

У розділі обґрунтовано вибір саме цього алгоритму як основного методу класифікації. Такий вибір пояснюється низкою переваг Random Forest, серед яких — висока точність, стійкість до перенавчання, ефективна робота з багатовимірними ознаками та відносно низькі вимоги до обчислювальних ресурсів. Це робить його

придатним для використання в системах, які працюють у режимі реального часу, що є важливою умовою для інтерактивних застосунків.

Окрему увагу приділено розробці та опису алгоритму навчання моделі класифікації. Зокрема, охоплено етапи підготовки даних, що включають очищення, нормалізацію, розділення на тренувальну та тестову вибірки, а також безпосереднє навчання моделі на основі зібраного датасету. Ефективність роботи класифікатора оцінюється з використанням відповідних метрик якості, таких як точність, повнота, F1-міра, що дозволяє комплексно оцінити здатність моделі до узагальнення на нових прикладах.

Таким чином, у межах даного розділу була сформована цілісна, логічно структурована та методологічно обґрунтована архітектура системи розпізнавання жестової мови. Запропонована система має модульну побудову, що забезпечує її гнучкість, зручність у подальшому масштабуванні та адаптації до нових задач. Отримані результати створюють надійне підґрунтя для реалізації функціонального прототипу системи, її експериментального тестування та подальшої інтеграції в прикладні інформаційні системи, орієнтовані на підвищення доступності комунікацій для осіб з порушеннями мовлення.

3 РЕАЛІЗАЦІЯ СИСТЕМИ РОЗПІЗНАВАННЯ ЖЕСТОВОЇ МОВИ

3.1. Середовище розробки та використовувані інструменти

3.1.1 Вибір мови програмування та інструментів для реалізації системи розпізнавання жестової мови

Розробка системи розпізнавання жестової мови була виконана з використанням мови програмування Python, що стало ключовим фактором для досягнення гнучкості, масштабованості та ефективності при реалізації завдань комп'ютерного зору, обробки відео та побудови моделей машинного навчання. Python, завдяки своїй багатій екосистемі спеціалізованих бібліотек, дозволив інтегрувати всі необхідні компоненти системи в єдину функціональну архітектуру.

Розробка програмного забезпечення здійснювалася у середовищі PyCharm, яке забезпечує зручні засоби для написання, тестування, відлагодження та організації коду. Наявність розширень для Python, візуалізації результатів та інтеграції з системами керування версіями зробило цей інструмент ефективним середовищем для реалізації проєкту.

Для реалізації захоплення відеопотоку з веб-камери, попередньої обробки зображень та побудови графічних інтерфейсів було використано бібліотеку OpenCV, яка забезпечує доступ до потужного інструментарію комп'ютерного зору в реальному часі. Виявлення рук та відстеження ключових точок на кисті виконувалося за допомогою фреймворку MediaPipe, який дозволяє з високою точністю і швидкістю здійснювати трекінг руки, що є критично важливим для подальшої класифікації жестів.

Процес розпізнавання жестів реалізовано на основі моделі машинного навчання, побудованої за допомогою алгоритму RandomForestClassifier, що входить до складу бібліотеки Scikit-learn. Цей алгоритм було обрано завдяки його високій точності, стійкості до перенавчання та здатності ефективно працювати з табличними даними, які утворюються після вилучення координат ключових точок руки.

Уся система створювалася та проходила тестування в середовищі операційної системи Windows 11 із використанням стандартної вбудованої веб-камери ноутбука. Такий підхід дав змогу проводити випробування в умовах, максимально наближених до реального використання, що забезпечило зручність та оперативність під час налагодження функціоналу в реальному часі. Для організації ізольованого середовища розробки було створено віртуальне оточення, яке гарантувало контроль версій залежностей, уникнення конфліктів між бібліотеками та загальну стабільність роботи застосунку. Це рішення сприяло відтворюваності результатів та ефективному управлінню програмним середовищем у процесі розробки.

3.1.2. Архітектура та функціонал MediaPipe у системі розпізнавання жестової мови

У межах цієї кваліфікаційної роботи було застосовано фреймворк MediaPipe, розроблений компанією Google, що призначений для створення високопродуктивних мультимодальних систем обробки поточкових даних у реальному часі. Цей фреймворк широко використовується в задачах комп'ютерного зору, таких як виявлення обличч, поз тіла, об'єктів і, зокрема, жестів рук. Його основною перевагою є графова архітектура, яка дозволяє будувати гнучкі, масштабовані обчислювальні конвеєри, де окремі модулі легко комбінуються та повторно використовуються, що робить MediaPipe зручним для інтеграції у складні програмні рішення.

У контексті розпізнавання жестів MediaPipe забезпечує ефективне виявлення руки, а також визначення ключових точок кисті в тривимірному просторі. Завдяки високоточному відстеженню орієнтирів руки система працює стабільно навіть у складних умовах, таких як часткове перекриття кисті або вихід за межі кадру. Модель орієнтації кисті виконує пряме прогнозування координат на основі нейронної мережі, нетренованої з використанням великої кількості реальних та синтетичних 3D-зображень рук, що дозволяє охопити широкий спектр можливих положень кисті та покращити загальну адаптивність до різних сценаріїв.

Важливою особливістю MediaPipe є його здатність мінімізувати обчислювальні витрати. Система застосовує повторне використання даних попередніх кадрів: якщо положення руки вже було визначено раніше, поточна обробка базується на цій інформації. Лише у разі втрати точності або неможливості виявлення руки модель знову активує етап виявлення долоні. Це значно підвищує швидкодію та дозволяє підтримувати стабільну роботу в режимі реального часу навіть на пристроях з обмеженими ресурсами.

У процесі роботи MediaPipe видає набір із 21 тривимірної координати, що відповідають ключовим точкам руки — від зап'ястя до кінчиків пальців. Саме ці координати використовуються як вхідні дані для подальшої класифікації жестів. Завдяки поєднанню високої точності, швидкості обробки та ефективного використання ресурсів MediaPipe став одним з ключових інструментів, які забезпечують реалізацію функціональної та стабільної системи розпізнавання жестів у цьому проєкті (рис. 3.1).

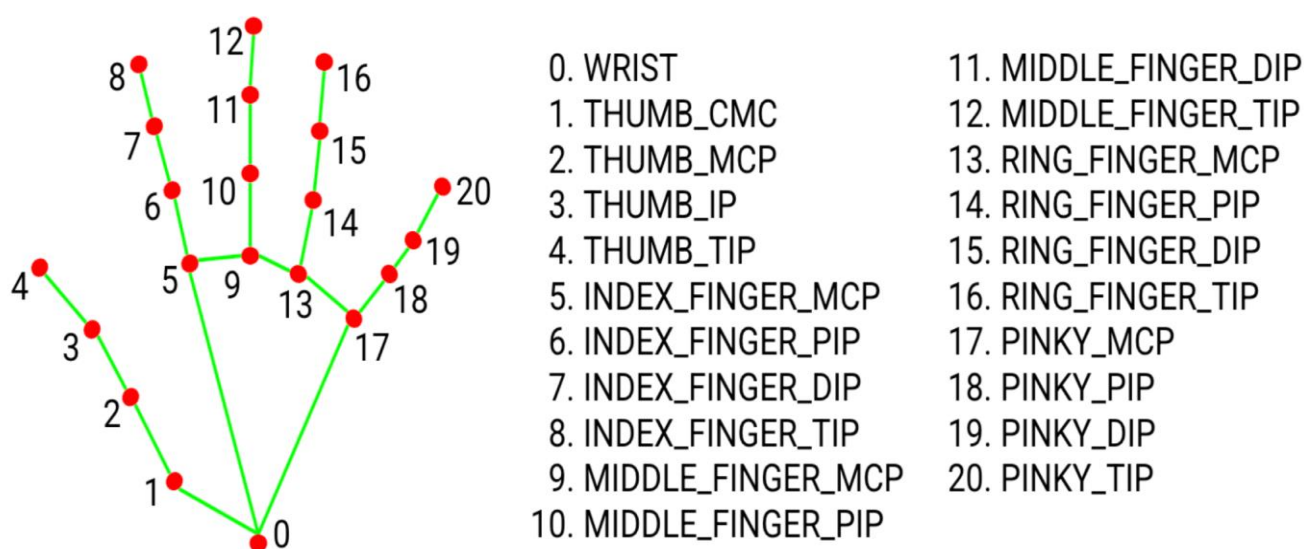


Рис. 3.1 Виявлення ключових точок на руці за допомогою MediaPipe [4]

Систематизація цих ключових точок, їхні числові індекси та відповідні анатомічні позначення, є фундаментальною для подальшого вилучення ознак та класифікації жестів (табл. 3.1).

Таблиця 3.1 Систематизація ключових точок кисті руки, використовуваних у моделі MediaPipe Hands

№	Назва ключової точки	Опис анатомічного положення	№	Назва ключової точки	Опис анатомічного положення
0	WRIST	Зап'ястя	10	MIDDLE_FINGER_PIP	Проксимальний міжфаланговий суглоб середнього пальця
1	THUMB_CMC	Основа п'ясткової кістки великого пальця	11	MIDDLE_FINGER_DIP	Дистальний міжфаланговий суглоб середнього пальця
2	THUMB_MCP	Проксимальний суглоб великого пальця	12	MIDDLE_FINGER_TIP	Кінчик середнього пальця
3	THUMB_IP	Дистальний суглоб великого пальця	13	RING_FINGER_MCP	Основа безіменного пальця
4	THUMB_TIP	Кінчик великого пальця	14	RING_FINGER_PIP	Проксимальний міжфаланговий суглоб безіменного пальця
5	INDEX_FINGER_MCP	Основа вказівного пальця	15	RING_FINGER_DIP	Дистальний міжфаланговий суглоб безіменного пальця
6	INDEX_FINGER_PIP	Проксимальний міжфаланговий суглоб вказівного пальця	16	RING_FINGER_TIP	Кінчик безіменного пальця
7	INDEX_FINGER_DIP	Дистальний міжфаланговий суглоб вказівного пальця	17	PINKY_MCP	Основа мізинця
8	INDEX_FINGER_TIP	Кінчик вказівного пальця	18	PINKY_PIP	Проксимальний міжфаланговий суглоб мізинця
9	MIDDLE_FINGER_MCP	Основа середнього пальця	19	PINKY_DIP	Дистальний міжфаланговий суглоб мізинця
			20	PINKY_TIP	Кінчик мізинця

3.2. Процес збору та підготовки даних для навчання моделі

Процес формування навчального набору даних здійснювався із застосуванням бібліотеки OpenCV на мові програмування Python, що забезпечило зручний механізм захоплення та обробки візуальної інформації в режимі реального часу. У результаті реалізованих процедур було створено власний датасет, який містить 5800 зразків зображень.

Для кожного символу англійського алфавіту, а також для низки спеціально визначених службових жестів — зокрема, "START" (рис. 3.2) (позначення запуску програми), "SPACE" (вставка пробілу між словами)(рис. 3.3) та "SEND" (відправка

згенерованого повідомлення) (рис. 3.4) — було зібрано по 200 зображень. Кожен жест фіксувався як у виконанні лівою, так і правою рукою, що сприяло підвищенню варіативності даних та забезпеченню більшої гнучкості і стійкості майбутньої моделі до різноманітних умов виконання.

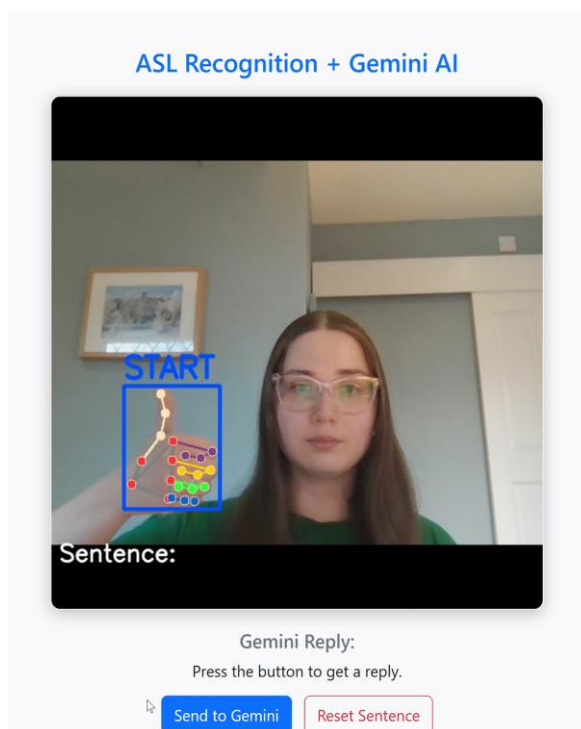


Рис. 3.2 Жест для запуску програми

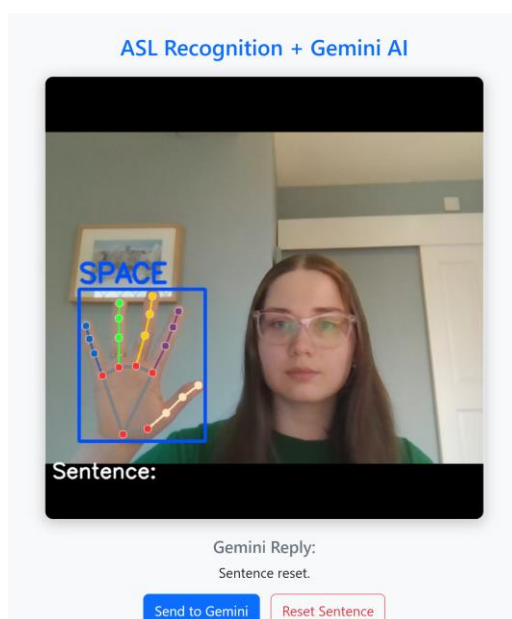


Рис. 3.3 Жест для вставки пробілу між словами

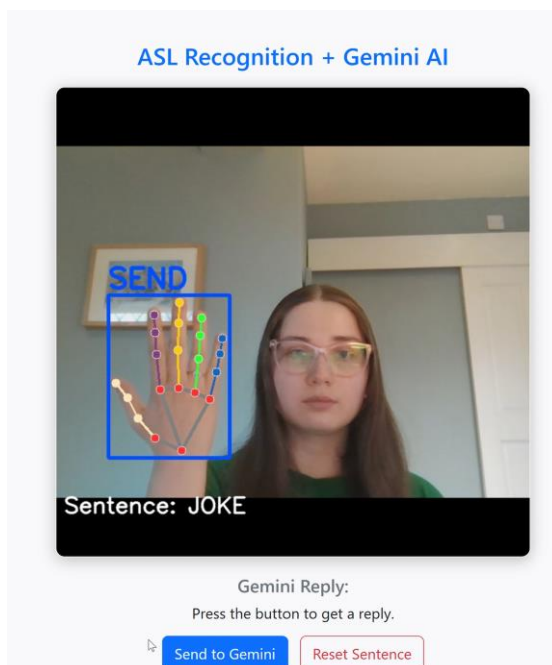


Рис. 3.4 Жест для відправки згенерованого повідомлення

Для збирання даних було попередньо опановано рухи американської жестової мови (ASL). Відтворення жестів перед веб-камерою ноутбука дозволяє здійснити захоплення відео-кадрів із подальшим перетворенням на зображення (рис. 3.5). Такий підхід забезпечує структуровану організацію даних і створив основу для ефективного навчання моделі розпізнавання жестової абетки. [11].

Виконавець жестів

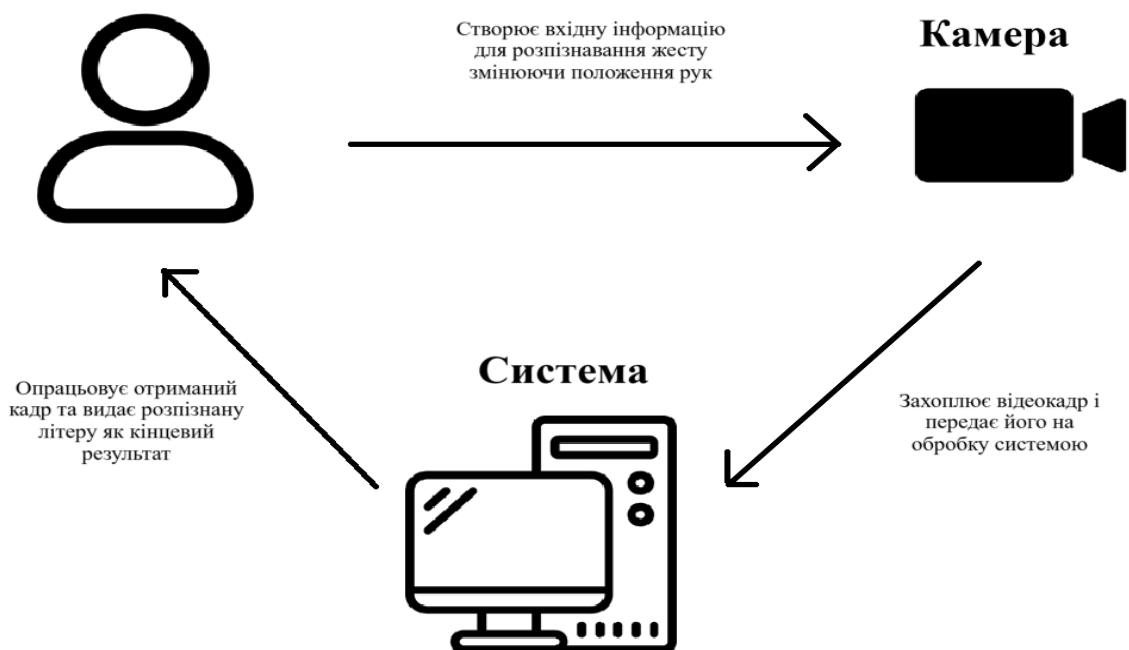


Рис. 3.5 Процес розпізнавання жестів з веб-камери

На початковому етапі виконавець жестів формує вхідні дані шляхом зміни положення рук у межах поля зору камери. Жестові конфігурації, які формуються за рахунок специфічного розташування пальців, відповідають певним літерам алфавіту або спеціальним керуючим командам. Геометричні характеристики положення та орієнтації пальців виступають критично важливими параметрами для точного виявлення та ідентифікації жесту.

Камера, інтегрована в систему, здійснює безперервне захоплення відеопотоку, в якому фіксуються рухи користувача. Кожен кадр цього потоку обробляється у реальному часі за допомогою бібліотеки OpenCV, яка забезпечує стабільне та високочастотне оновлення зображень. Ця функціональність є необхідною умовою для забезпечення належної якості вхідних даних та підтримки роботи системи в інтерактивному режимі. Після захоплення кадру зображення передається до обчислювального блоку, де виконується його подальший аналіз.

На першому етапі обробки даних, за допомогою бібліотеки MediaPipe, здійснюється визначення координат 21 ключового орієнтира кисті, які надалі використовуються як числові ознаки для ефективної класифікації жестів (рис. 3.1).

Далі здійснюється процес нормалізації координат, що дозволяє зменшити вплив масштабних і позиційних відмінностей. Отримані дані формуються у вигляді векторів ознак, які надходять на вхід класифікаційної моделі — у даному випадку, алгоритму RandomForestClassifier. Модель, проаналізувавши вхідний вектор, видає відповідну розпізнану літеру або команду, що інтерпретується як результат системної обробки жесту. Кінцевий результат виводиться користувачеві на екрані у вигляді розпізнаваної літери або службової команди.

Таким чином, було детально описано та реалізовано процес збору та підготовки даних, що є фундаментальним для навчання моделі розпізнавання жестів. Завдяки систематичному відтворенню як стандартних жестів американської жестової мови (ASL), так і спеціально розроблених власних команд перед веб-камерою, а також ефективному використанню MediaPipe для вилучення та нормалізації ключових точок, було сформовано репрезентативний та чистий датасет. Ця ретельно підготовлена вибірка даних забезпечила надійну основу для

подальшого навчання класифікаційної моделі, гарантуючи її здатність ефективно розпізнавати жести в інтерактивному асистенті.

3.3 Проектування графічного інтерфейсу системи розпізнавання мови жестів

Проектування графічного інтерфейсу системи розпізнавання мови жестів орієнтувалося на забезпечення максимальної зручності користувача при взаємодії із застосунком. Інтерфейс мав бути інтуїтивно зрозумілим, функціональним і простим у використанні, з урахуванням потреб користувачів, які можуть мати обмеження у традиційних способах взаємодії з комп'ютером.

Інтерфейс застосунку має лаконічний, але водночас функціонально насичений дизайн, орієнтований на інтуїтивну взаємодію користувача з інтерактивним цифровим асистентом. Структура та розташування ключових елементів були розроблені відповідно до макета, представленого на рисунку 3.6. Його реалізація здійснена за допомогою легкого та масштабованого веб-фреймворку Flask, що дозволяє ефективно поєднати серверну логіку з динамічним клієнтським інтерфейсом.

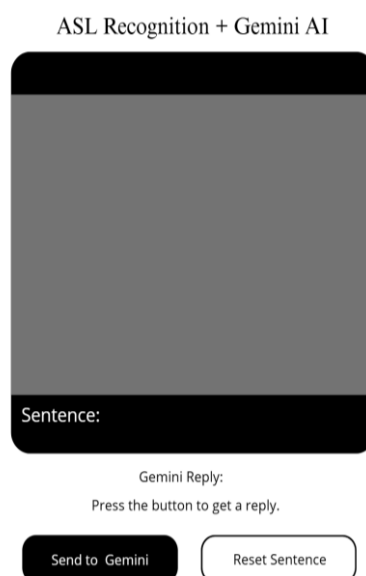


Рис. 3.6 Вид графічного інтерфейсу інтерактивного цифрового асистента

У процесі розробки було визначено основні функціональні елементи інтерфейсу. На головному екрані реалізовано відображення відеопотоку з веб-

камери, у якому користувач може бачити своє зображення та процес розпізнавання жестів у реальному часі (рис. 3.7).

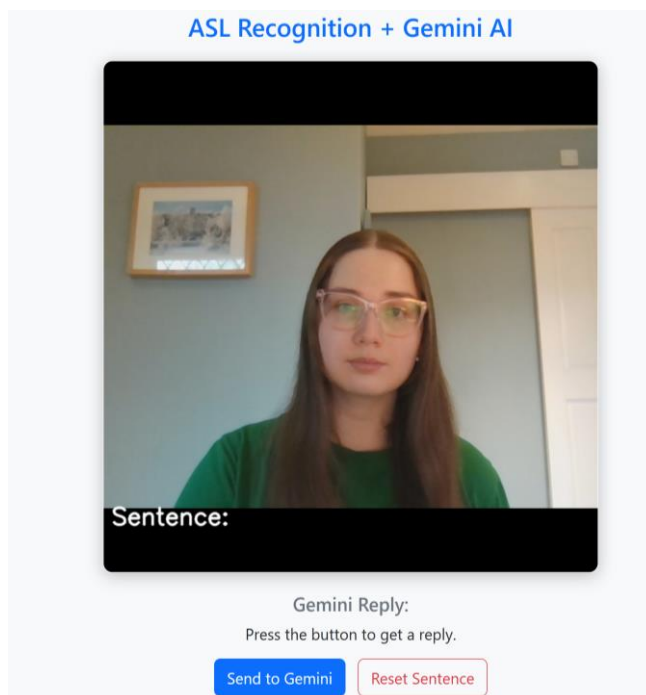


Рис. 3.7 Головний екран

У верхній частині сторінки розміщується заголовок “ASL Recognition + Gemini AI”, який відразу інформує користувача про ключову функціональність системи: розпізнавання американської жестової мови (ASL) у поєднанні з можливостями генеративного штучного інтелекту Gemini. Цей елемент виконує не лише навігаційну, а й комунікативну функцію, задаючи контекст взаємодії.

Центральний блок інтерфейсу займає прямокутна відеозона із заокругленими кутами, що транслює зображення з веб-камери в реальному часі. Вона служить основним візуальним каналом зчитування жестів користувача. У нижньому лівому куті відеоекрана знаходиться динамічне текстове поле з підписом “Sentence:”, яке в режимі реального часу відображає поточне речення, сформоване на основі розпізнаних жестів. Це дозволяє користувачу контролювати, наскільки точно система інтерпретує введену інформацію.

Нижче відеоблоку розташовано розділ, присвячений взаємодії з модулем штучного інтелекту Gemini. У ньому міститься текстовий заголовок “Gemini Reply:” та коротка інструкція “Press the button to get a reply.”, що орієнтує

користувача щодо подальших дій. Під цим блоком знаходяться дві чітко виділені кнопки керування: “Send to Gemini” (синього кольору) активує передачу сформованого речення на обробку AI-моделлю, тоді як “Reset Sentence” (червоного кольору) скидає вміст текстового поля для повторного введення.

Загальна візуальна структура інтерфейсу вирізняється простотою форм, зрозумілою композицією та естетично витриманим мінімалізмом. Всі елементи логічно згруповані й легко сприймаються візуально, що сприяє комфортній та безперешкодній роботі користувача з системою. Завдяки використанню Flask, інтерфейс забезпечує швидкий відгук, стабільність та можливість подальшого розширення функціоналу — зокрема, впровадження збереження історії, мультимовної підтримки або інтеграції голосового зворотного зв’язку.

У системі реалізовано спеціальний жест, який виконує функцію запуску основного процесу розпізнавання. Цей жест виступає як умовна команда “START”, яку система розпізнає перед початком обробки інших жестів (рис. 3.2). Його використання дозволяє уникнути помилкового розпізнавання випадкових рухів рук, забезпечуючи контрольований запуск програми користувачем.

Після фіксації цього стартового жесту активується модуль збору та аналізу подальших жестів. Завдяки цьому підходу забезпечується інтуїтивно зрозуміла взаємодія з інтерфейсом, а також покращується точність розпізнавання, оскільки система починає обробку лише після свідомого сигналу від користувача.

Поруч розміщено поле, що називається “Sentence”, у яке виводяться вже розпізнані літери, що поступово формують слова та речення (рис. 3.8). Цей блок дозволяє візуально контролювати правильність розпізнавання.

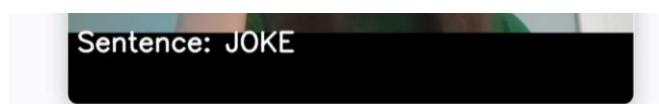


Рис. 3.8 Результати виведення поля розпізнаваних літер

Передбачено кнопку очищення тексту, а також кнопку відправки сформованого речення до зовнішнього сервісу – Gemini API (рис.3.4). Отриману відповідь система виводить нижче у спеціальному полі, супроводжуючи її

синтезованим голосом, що забезпечує зворотний зв'язок користувачу. Загальний стиль інтерфейсу виконаний у стриманій кольоровій гамі з великими, чіткими шрифтами та зручним розташуванням елементів.

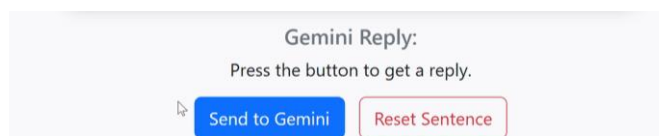


Рис. 3.9 Кнопки відправлення та очищення тексту

У проєктуванні графічного інтерфейсу також враховано інтеграцію з машинним навчанням, зокрема збереження та використання натренованої моделі класифікації жестів. Для цього застосовано бібліотеку *pickle*, яка дозволяє серіалізувати об'єкти Python, включно з навченою моделлю. Після завершення етапу навчання модель зберігається у вигляді файлу, що згодом завантажується безпосередньо у процесі роботи застосунку.

Це забезпечує швидкий доступ до класифікатора без необхідності повторного навчання моделі щоразу при запуску системи. Взаємодія інтерфейсу з *pickle* - моделлю здійснюється у фоновому режимі, що не впливає на зручність користувача, але забезпечує ефективність та стабільність роботи застосунку. Такий підхід дозволяє легко оновлювати або замінювати модель при покращенні алгоритмів розпізнавання без зміни структури інтерфейсу.

Після завершення формування речення з розпізнаних жестів, його можна відправити за допомогою спеціального жесту (рис. 3.4) або кнопки відправлення (рис. 3.9), що передається до моделі обробки природної мови через Gemini API. Цей інтерфейс забезпечує взаємодію з сучасною мовною моделлю, яка аналізує вхідний текст, формує осмислену відповідь і повертає її системі. Отримана відповідь виводиться на екран у текстовому полі, доступному користувачу для перегляду результату (рис. 3.10).

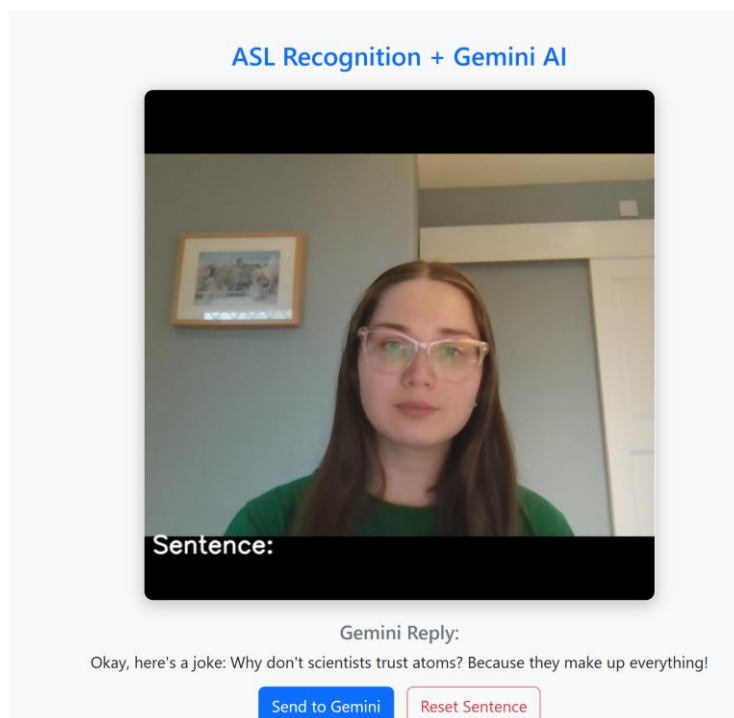


Рис. 3.10 Виведення відповіді від Gemini API на екран

Окрім текстового відображення, реалізовано додаткову функціональність — озвучення відповіді за допомогою модуля синтезу мовлення. Це дозволяє не лише візуально отримати відповідь, але й сприймати її аудіально, що значно розширює можливості комунікації для користувачів. Таким чином, система жестового інтерфейсу інтегрується з мовною моделлю, забезпечуючи повний цикл перетворення жестової мови у зрозумілу текстову відповідь із голосовим супроводом.

3.4. Розробка та реалізація програмного інтерфейсу користувача

3.4.1. Реалізація програмного інтерфейсу користувача на базі Flask

Розробка програмного інтерфейсу користувача стала ключовим етапом у створенні інтерактивної та доступної системи розпізнавання жестової мови. Зважаючи на вимоги до гнучкості, кросплатформенності та зручності розгортання, для реалізації інтерфейсу був обраний підхід на основі веб-технологій з використанням фреймворку Flask. Flask виступив у ролі бекенду веб-додатка, забезпечуючи логіку роботи системи та взаємодію з її основними модулями, такими як модуль класифікації та модулі захоплення й обробки відео. Його основні

функції охоплювали маршрутизацію запитів, прийом та обробку відеопотоку з веб-камери користувача та його подальшу передачу до модулів виявлення рук і класифікації жестів. Крім того, Flask керував логікою розпізнавання, інтегруючись з основним алгоритмом класифікації RandomForestClassifier, та відповідав за генерацію відповідей, формуючі та надсилаючи HTML-сторінки, а також передаючи результати розпізнавання (текстове представлення жесту) назад на фронтенд. Завдяки своїй легкості та розширюваності, Flask дозволив ефективно організувати взаємодію між користувацьким інтерфейсом та ядром системи розпізнавання.

3.4.2 Архітектура інтеграції Gemini API

Для розширення функціональних можливостей та забезпечення більш глибокої інтеракції з користувачем, у програмний інтерфейс була інтегрована технологія Gemini API. Gemini API є потужним інструментом для роботи з великими мовними моделями, що дозволяє виконувати додаткову інтерпретацію розпізнаних результатів, генерувати відповіді у природній мові та створювати більш складні сценарії діалогу. Інтеграція здійснювалася шляхом надсилання розпізнаних даних або запитів користувача до Gemini API та отримання відповіді для подальшого відображення або озвучення.

Інтеграція з Gemini API є одним із ключових елементів архітектури сучасного інтерактивного цифрового асистента, що забезпечує розширені можливості обробки природної мови, доступ до зовнішніх ресурсів і генерацію релевантних, контекстуально обґрунтованих відповідей. Однією з основних технологічних переваг Gemini є підтримка механізму функціональних викликів (Function Calling), який дозволяє моделі динамічно взаємодіяти з API інших сервісів. Загальна схема цього процесу наведена на рисунку 3.11.

Взаємодія розпочинається із формування запиту (Prompt), який надсилається до моделі Gemini. Такий запит може бути ініційований як кінцевим користувачем, так і внутрішніми модулями програмної системи. Зміст запиту зазвичай включає

інструкції або питання, які вимагають доступу до зовнішніх джерел інформації або виконання конкретних дій, що виходять за межі стандартної генерації тексту.

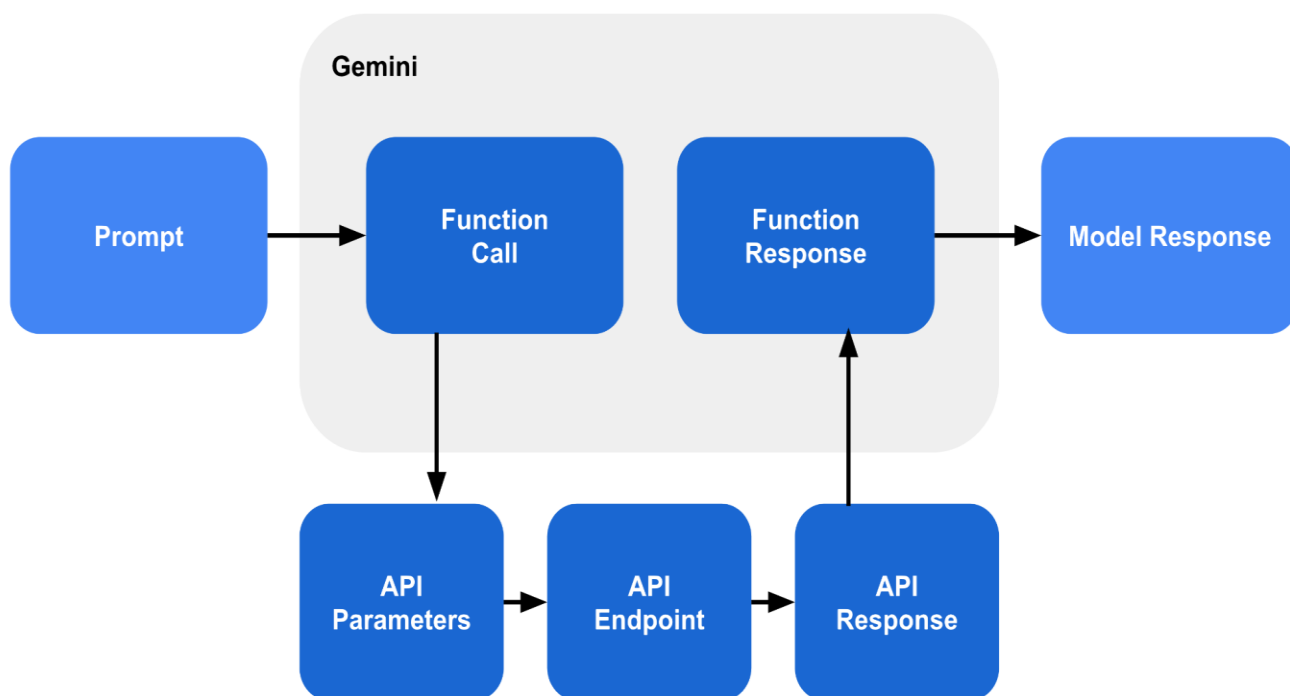


Рис. 3.11 Архітектура функціональних викликів Gemini API [3]

У разі виявлення такої потреби, модель активує механізм функціонального виклику (Function Call). На цьому етапі вона аналізує зміст запиту, визначає відповідну зовнішню функцію або сервіс, та формує набір параметрів (API Parameters), необхідних для її виклику. Ці параметри відображають суть запиту й адаптуються до вимог інтерфейсу обраного зовнішнього API.

Сформовані параметри надсилаються до певної кінцевої точки (API Endpoint) — програмного інтерфейсу зовнішньої системи або сервісу. На цьому етапі може здійснюватися звернення до баз даних, обчислення, обробка запитів до сторонніх сервісів тощо. Результатом успішного виконання функції є повернення відповіді API (API Response), яка містить оброблені дані або результати запитуваної операції.

Отримана відповідь надходить назад до моделі у вигляді Function Response. Модель інтегрує ці дані у свій внутрішній контекст, що дозволяє сформулювати фінальну відповідь (Model Response) для користувача або наступного етапу.

обробки в системі. Така відповідь може бути текстовою, або передаватися на синтез мовлення, створюючи повноцінну голосову взаємодію.

Таким чином, реалізація функціональних викликів у межах Gemini API забезпечує не лише більш гнучку поведінку моделі, а й розширює її здатність виконувати складні багатокрокові завдання в реальному часі. Це відкриває нові можливості для створення асистентів, здатних ефективно працювати в динамічному середовищі, інтегруючи інформацію з різних джерел та адаптуючись до контексту запиту. У контексті розробки цифрового асистента для людей з порушенням мовлення, такий механізм є надзвичайно важливим для забезпечення природної, багатоканальної та інклюзивної комунікації.

3.4.3 Синтез голосових відповідей

Для забезпечення голосового зворотного зв'язку та підвищення доступності інтерактивного цифрового асистента була реалізована функція перетворення тексту на мовлення (Text-to-Speech, TTS) із використанням бібліотеки pyttsx3. Ця бібліотека є офлайн-рішенням, що працює незалежно від Інтернет-з'єднання та базується на вбудованих у систему синтезаторах мовлення, таких як SAPI5 (для Windows), NSSpeechSynthesizer (для macOS) або eSpeak (для Linux). Такий підхід дозволяє забезпечити стабільну й автономну роботу голосового модуля навіть в умовах обмеженого доступу до мережі.

Процес синтезу розпочинається після того, як модуль розпізнавання жестів ідентифікує конкретний жест і перетворює його на текстове повідомлення. Це повідомлення передається до функції pyttsx3, яка генерує голосове відтворення – або у вигляді аудіофайлу, або у режимі реального часу через системні аудіовиходи. Крім базових можливостей, бібліотека також підтримує налаштування голосу, темпу мовлення та гучності, що дозволяє адаптувати звучання до потреб конкретного користувача чи ситуації використання.

Застосування pyttsx3 істотно розширює функціональність системи, роблячи її більш інклюзивною. Зокрема, така реалізація є корисною для осіб із порушеннями зору або в тих випадках, коли візуальний зворотний зв'язок є

недоступним або недостатнім. Голосове озвучення результатів жестового введення виконує також роль додаткового каналу підтвердження коректності розпізнавання, що підвищує зручність і надійність взаємодії з системою.

Окрім того, можливість генерування аудіофраз у реальному часі створює передумови для застосування цієї технології у ширшому спектрі сценаріїв: від освітніх застосунків до інтерфейсів для людей з порушенням мовлення, забезпечуючи природнішу комунікацію між користувачем і цифровим середовищем. У перспективі синтез мовлення може бути додатково оптимізований за рахунок впровадження сучасніших моделей, таких як Tacotron або VITS, що забезпечують більш природне та виразне звучання.

Таким чином, інтеграція модуля синтезу мовлення на базі pyttsx3 значно підвищує функціональну повноту цифрового асистента, сприяючи його адаптації до потреб різних категорій користувачів і забезпечуючи ефективну мультимодальну взаємодію.

3.5 Експериментальні дослідження та аналіз результатів

3.5.1 Методи оцінки точності та продуктивності моделі

Для об'єктивного оцінювання ефективності та надійності розробленої системи розпізнавання жестів використовуються стандартизовані метрики, які дозволяють кількісно проаналізувати її точність і виявити напрямки для покращення.

Одним із базових показників є `accuracy_score` — частка правильно класифікованих прикладів серед усіх. Проте в умовах незбалансованих даних ця метрика може бути неточною, оскільки висока точність може досягатися лише завдяки правильному передбаченню переважного класу.

Більш глибоке розуміння забезпечує `classification_report`, що містить ключові метрики — точність (`precision`), повноту (`recall`) і F1-міру — для кожного класу. Звіт також включає середні значення, які дають загальну оцінку роботи моделі, особливо актуальну при нерівномірному розподілі класів.

`Confusion_matrix` дає змогу побачити, які жести система плутає між собою, наочно демонструючи співвідношення правильних і хибних передбачень. Це дозволяє деталізувати аналіз помилок і краще зрозуміти слабкі сторони моделі.

Перед навчанням дані діляться за допомогою `train_test_split`, що забезпечує розділення на тренувальну та тестову вибірки. Це критично важливо для запобігання перенавчанню і забезпечення об'єктивної перевірки здатності моделі до узагальнення. Стратифіковане розділення допомагає зберегти пропорції класів у вибірках, що є важливим при роботі з незбалансованими даними.

Аналіз результатів класифікації на тестовій вибірці, проведений за допомогою `classification_report`, демонструє ефективність розробленої моделі ASL-розпізнавання та її здатність точно ідентифікувати жести різних категорій.

3.5.2 Аналіз результатів експериментів з розпізнавання жестів

Для всебічного та об'єктивного оцінювання ефективності розробленої моделі розпізнавання жестів було проведено серію експериментів із використанням окремого тестового набору даних. Цей набір не використовувався під час навчання моделі, що дозволило забезпечити неупереджену оцінку її здатності до узагальнення та уникнути ефекту перенавчання.

Результати класифікації (табл. 3.2) відображають продуктивність системи за кількома ключовими показниками: точністю, повнотою, F1-мірою та кількістю зразків для кожного класу жестів. Ці метрики дають можливість оцінити, наскільки правильно модель передбачає відповідний жест, чи здатна вона розпізнати всі правильні приклади кожного класу, а також наскільки збалансованою є її поведінка у складних або неоднозначних випадках. Зокрема, F1-міра як гармонійне середнє між точністю та повнотою надає уніфіковану оцінку якості класифікації для кожного окремого жесту.

Кожен жест у тестовому наборі був представлений однаковою кількістю прикладів — 40 зразками, що забезпечило високу збалансованість даних. Така рівномірність розподілу дозволяє уникнути домінування окремих класів і сприяє точнішій оцінці якості роботи моделі у загальному випадку. Високі значення F1-

міри для більшості класів свідчать про надійність обраного алгоритму класифікації — RandomForestClassifier — та його здатність ефективно відрізняти жести навіть при незначних варіаціях виконання.

Таблиця 3.2 Класифікації моделі розпізнавання жестів

Клас жесту	Точність (Precision)	Повнота (Recall)	F1-міра (F1-score)	Кількість зразків (Support)
A	1.00	1.00	1.00	40
B	1.00	1.00	1.00	40
C	1.00	1.00	1.00	40
D	1.00	1.00	1.00	40
E	1.00	1.00	1.00	40
F	1.00	1.00	1.00	40
G	1.00	1.00	1.00	40
H	1.00	1.00	1.00	40
I	1.00	1.00	1.00	40
J	1.00	0.97	0.99	40
K	1.00	1.00	1.00	40
L	1.00	1.00	1.00	40
M	1.00	1.00	1.00	40
N	1.00	1.00	1.00	40
O	1.00	1.00	1.00	40
P	1.00	1.00	1.00	40
Q	1.00	1.00	1.00	40
R	1.00	1.00	1.00	40
S	1.00	1.00	1.00	40
SEND	1.00	1.00	1.00	40
SPACE	1.00	1.00	1.00	40
START	1.00	1.00	1.00	40
T	1.00	1.00	1.00	40
U	1.00	1.00	1.00	40
V	1.00	1.00	1.00	40
W	1.00	1.00	1.00	40
X	1.00	1.00	1.00	40
Y	1.00	1.00	1.00	40
Z	0.98	1.00	0.99	40

Спостережуване незначне зниження продуктивності в окремих класах може бути пов'язане з візуальною схожістю жестів або нечітким виконанням рухів у тестових зразках. Проте загальна якість роботи моделі вказує на її стабільність, точність та доцільність інтеграції в реальні програмні рішення. Таким чином,

результати експериментального оцінювання підтверджують ефективність і практичну придатність створеної системи розпізнавання жестової мови.

Переважає більшість класів жестів демонструє виняткові показники точності, повноти та F1-міри, які дорівнюють 1.00. Це свідчить про те, що модель безпомилково ідентифікувала всі жести для таких класів, як літери від "A" до "I", від "K" до "S", а також команди "SEND", "SPACE", "START", "T", "U", "V", "W", "X" та "Y". Усі передбачення для зазначених жестів були точними, без жодного хибнопозитивного результату, що свідчить про абсолютну точність моделі. Повнота для цих класів також є ідеальною, оскільки модель успішно виявила всі наявні зразки у тестовому наборі, не втративши жодного з них. Як наслідок, F1-міра, що враховує обидва ці аспекти, досягла максимального рівня, що свідчить про ідеальну узгодженість між точністю та повнотою.

Незначні відхилення від ідеальних значень були зафіксовані лише для двох класів: "J" та "Z". Для класу "J" модель продемонструвала точність на рівні 1.00, що вказує на повну відсутність хибнопозитивних результатів — інші жести ніколи не були помилково класифіковані як "J". Проте повнота становить 0.97, що означає, що не всі фактичні зразки жесту "J" були розпізнані. Таким чином, імовірно, один або два з сорока зразків не були класифіковані як "J", що дещо знизило відповідну F1-міру до 0.99. У свою чергу, клас "Z" демонструє іншу ситуацію: повнота становить 1.00, тобто всі справжні зразки були виявлені, однак точність дорівнює 0.98, що вказує на незначну кількість хибнопозитивних спрацювань — у деяких випадках інші жести були помилково визначені як "Z". У підсумку, F1-міра для цього класу також дещо нижча за ідеальну, на рівні 0.99.

Загалом, результати, отримані під час тестування моделі, вказують на надзвичайно високу ефективність і надійність системи. Переважає більшість жестів розпізнається з ідеальною точністю та повнотою, що свідчить про стабільність і точність навченої моделі. Незначні розбіжності, виявлені для жестів "J" та "Z", не впливають суттєво на загальну якість розпізнавання, але можуть слугувати підґрунтям для подальшого вдосконалення системи. Загальна оцінка ефективності моделі (табл. 3.3), що ґрунтується на узагальнених метриках,

розрахованих на основі тестового набору даних, який містив 1160 зразків. Ці агреговані значення є вагомим доповненням до покласової статистики, підтверджуючи високу загальну якість роботи системи.

Таблиця 3.3 Метрики ефективності моделі класифікації

Метрика	Точність (Accuracy)	Precision (Макро/ Зважене)	Recall (Макро/ Зважене)	F1-міра (Макро/ Зважене)	Кількість зразків
Macro avg	1.0	1.00	1.00	1.00	1160
Weighted avg	1.0	1.00	1.00	1.00	1160

Загальний рівень точності (accuracy) моделі становить 1.00, що свідчить про абсолютну коректність класифікації усіх зразків, представлених у тестовому наборі. Інакше кажучи, з 1160 жестів, що входили до тестового датасету, жоден не був класифікований помилково. Це є ознакою не лише технічної досконалості реалізованої моделі, але й високої ступені її узагальнюючої здатності. Отриманий результат є надзвичайно позитивним, адже вказує на здатність моделі безпомилково оперувати інформацією за умов рівномірного представлення класів і в реальних сценаріях використання.

Макроусереднені значення основних метрик — точності (precision), повноти (recall) та F1-міри — також досягли максимального рівня 1.00. Оскільки макроусереднення передбачає обчислення середнього арифметичного кожної метрики без урахування кількості прикладів у класах, такий результат свідчить про стабільно високу ефективність моделі по всіх жестових категоріях. Модель не демонструє схильності до кращого розпізнавання лише окремих жестів — вона функціонує однаково ефективно для всіх класів, незалежно від складності або візуальної схожості між ними. Це особливо важливо в контексті реального застосування, де важлива надійність кожного елементу системи розпізнавання.

Зважені середні значення (weighted average) для всіх трьох ключових метрик також склали 1.00. На відміну від макроусереднення, у цьому випадку кожен клас враховується пропорційно до кількості своїх зразків. Проте, оскільки у даному

дослідженні всі класи мають однакову кількість представлень у тестовому наборі (по 40 зразків), зважені та макроусереднені значення повністю збігаються. Це ще раз підтверджує рівномірну якість класифікації на всіх категоріях і відсутність домінування або ігнорування будь-яких класів у процесі прийняття рішень.

Сукупність представлених агрегованих метрик — абсолютна точність, а також максимальні значення precision, recall та F1-score як у макро-, так і в зваженому середньому — засвідчує надзвичайно високу ефективність розробленої моделі розпізнавання жестової мови. Досягнення таких результатів вказує на успішне проходження етапу навчання, якісну підготовку вхідних даних і вдалий вибір алгоритмічної архітектури. Окрім того, такі метрики слугують обґрунтуванням готовності системи до подальшого розгортання, масштабування та впровадження в реальні практичні застосування, включаючи освітні, комунікаційні та інклюзивні технології.

Представлена матриця помилок (рис. 3.12) є візуальним інструментом для оцінки ефективності класифікації розробленою моделлю розпізнавання жестів на тестовому наборі даних. Кожен рядок цієї матриці відповідає фактичному класу жесту, тоді як кожен стовпець відображає клас, який модель спрогнозувала. Значення, що знаходяться на перетині рядка та стовпця, показують кількість зразків, які фактично належали до класу рядка, але були класифіковані моделлю як клас стовпця.

Аналізуючи дану матрицю, можна констатувати практично ідеальну класифікацію для переважної більшості жестів. Число 40 відображається на головній діагоналі для багатьох класів, включаючи A, B, C, D, E, F, G, H, I, K, L, M, N, O, P, Q, R, S, SEND, SPACE, START, T, U, V, W, X та Y. Це означає, що для кожного з цих жестів усі сорок тестових зразків були розпізнані моделлю абсолютно правильно, і жодного хибного спрацювання або пропуску не зафіксовано. Відсутність будь-яких інших значень у відповідних рядках і стовпцях підтверджує безпомилкову ідентифікацію цих жестів.

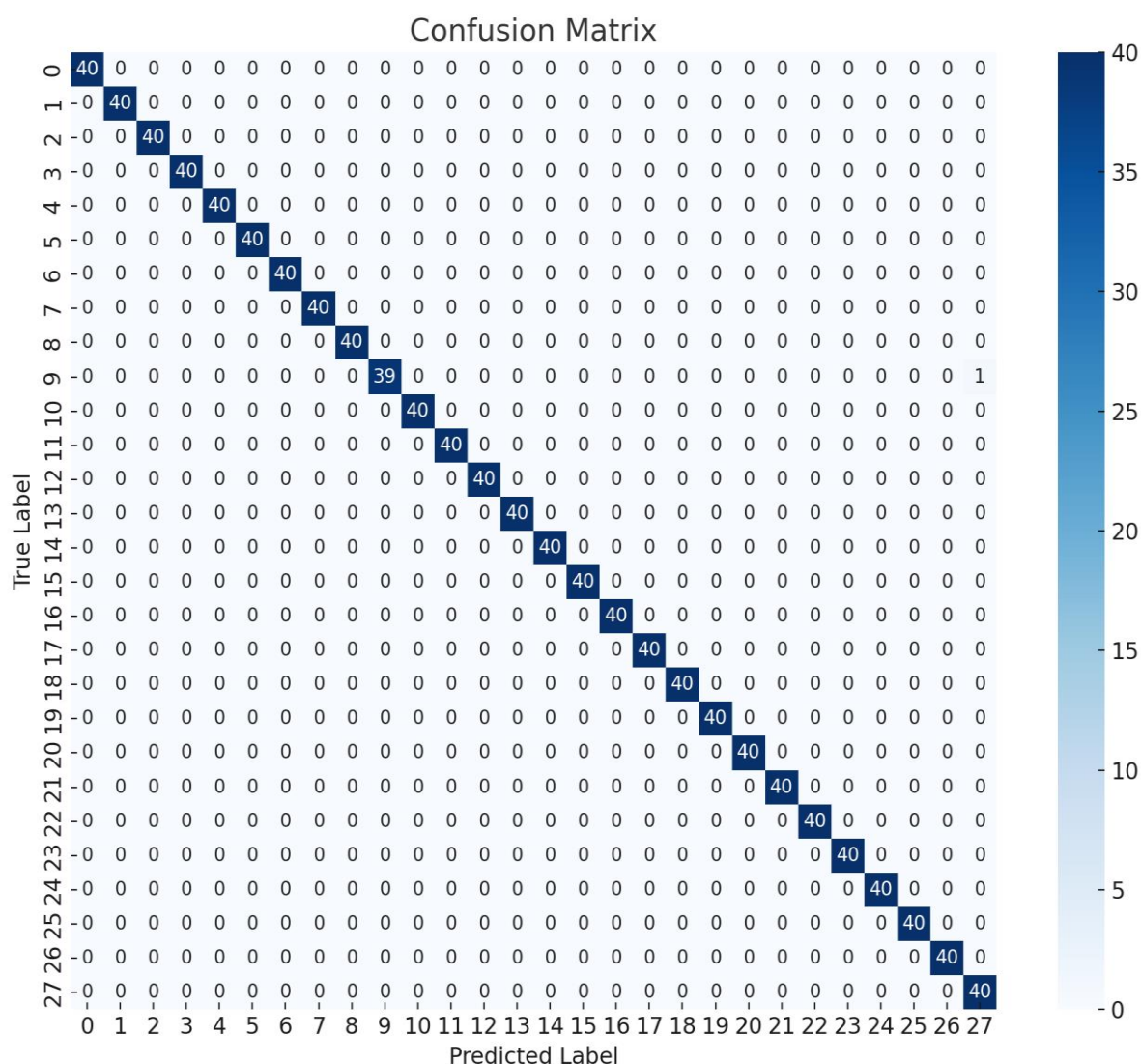


Рис. 3.12 Матриця помилок

Єдині відхилення від ідеальної класифікації спостерігаються для класу, умовно позначеного як десятий (відповідно до попередніх звітів, це клас "J"), та для останнього класу (умовно позначеного як "Z"). Для десятого класу значення на головній діагоналі становить 39, вказуючи на те, що 39 з 40 зразків були розпізнані вірно. Одиничне значення 1 зафіксовано в цьому ж рядку, але в останньому стовпці, що свідчить про одну помилку, коли жест, що фактично був "J", був помилково класифікований як "Z".

Щодо останнього класу ("Z"), значення на головній діагоналі становить 40, що означає правильне розпізнавання всіх 40 зразків цього жесту. Проте, згадане вище значення 1 у передостанньому рядку та останньому стовпці підтверджує, що

один зразок іншого жесту (а саме "J") був помилково ідентифікований як "Z", що є випадком хибнопозитивного спрацьовування для класу "Z".

Загалом, аналіз матриці помилок підтверджує високу ефективність розробленої моделі розпізнавання жестової мови. Переважна більшість жестів ідентифікується бездоганно, а поодинокі випадки помилкової класифікації для класів "J" та "Z" є незначними. Ці результати є свідченням успішного процесу навчання моделі та її здатності до достовірного розрізнення різних жестів Американської жестової мови на тестовому наборі даних.

У результаті, було докладно розглянуто процес практичної реалізації системи розпізнавання жестової мови, починаючи від вибору інструментарію і закінчуючи початковими експериментальними дослідженнями.

Було окреслено середовище розробки та обґрунтовано вибір використовуваних інструментів. Основною мовою програмування обрано Python, що дозволило ефективно інтегрувати низку спеціалізованих бібліотек та фреймворків. Для комп'ютерного зору та попередньої обробки відеоданих застосовувалися OpenCV та MediaPipe, які забезпечили надійне захоплення відеопотоку та високоточне виявлення ключових точок руки. Класифікація жестів здійснювалася за допомогою Scikit-learn (RandomForestClassifier), що підтвердило його ефективність для даної задачі.

Значну увагу приділено процесу збору та підготовки даних для навчання моделі. Було розроблено методологію збору відеоматеріалів, а також здійснено автоматичне вилучення 21 ключової точки руки з кожного кадру за допомогою MediaPipe. Отримані дані пройшли етапи очищення, нормалізації та були структуровані, що забезпечило високу якість навчальної вибірки та заклало основу для ефективного навчання моделі.

Розділ також охоплює проектування графічного інтерфейсу системи розпізнавання мови жестів. Оскільки йдеться про десктопний додаток або його частину, використання pickle для збереження моделей та їх швидкого завантаження є типовим для технічного застосування, що забезпечує швидкий доступ до навченої моделі.

Окремо описано розробку та реалізацію програмного інтерфейсу користувача, який, як з'ясовано, базується на веб-фреймворку Flask. Це дозволило створити зручний та доступний веб-інтерфейс для взаємодії з системою. Інтеграція Gemini API розширила можливості інтерфейсу, дозволивши використовувати потужні мовні моделі для обробки текстових даних, що може бути задіяно для покращення комунікації або додаткової інтерпретації розпізнаних жестів. Додатково, використання бібліотеки pytsx3 забезпечило голосовий вивід розпізнаних жестів, підвищивши інклюзивність та зручність використання системи.

Завершальна частина розділу присвячена експериментальним дослідженням та аналізу результатів. Проведене тестування та оцінка якості програмного забезпечення, що включає перевірку функціональності всіх модулів системи. Представлено результати, що демонструють високу точність та продуктивність розробленої моделі розпізнавання жестів на тестових даних, підтверджуючи ефективність обраного підходу та реалізованих рішень.

Таким чином, у цьому розділі було успішно здійснено практичну реалізацію всіх ключових компонентів системи розпізнавання жестової мови, підготовлено необхідну базу даних для навчання, розроблено зручний інтерфейс та проведено початкове тестування, яке підтвердило її функціональність та високу якість роботи.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було здійснено повноцінний цикл дослідження, проєктування, розробки та тестування інтерактивної системи розпізнавання жестової мови, орієнтованої на покращення доступності комунікації для осіб з порушенням мовлення. Робота охопила як теоретичні, так і практичні аспекти реалізації подібних систем, що дозволило отримати комплексне уявлення про сучасний стан галузі, а також розробити і протестувати власний прототип.

На основі вивчення теоретичних положень і аналізу сучасної літератури було досліджено ключові підходи до комп'ютерного розпізнавання жестів, включаючи методи машинного навчання, використання бібліотек комп'ютерного зору та систем трекінгу. Особливу увагу було приділено особливостям американської жестової мови (ASL), яка стала основою для навчання та тестування системи. Окрім цього, проведено порівняльний аналіз алгоритмів класифікації, що дозволив обґрунтовано обрати RandomForestClassifier як найефективніше рішення для задачі розпізнавання статичних жестів.

На етапі проєктування було побудовано архітектуру системи, яка включає окремі, взаємопов'язані модулі: захоплення відеопотоку, виявлення та трекінг рук (на основі MediaPipe), класифікацію жестів, генерацію текстового представлення, інтеграцію з мовною моделлю Gemini API для формування осмисленої відповіді та її озвучення за допомогою модуля TTS (Text-to-Speech) на базі бібліотеки pyttsx3. Цей підхід забезпечив гнучкість, модульність і можливість масштабування системи в майбутньому.

Особливої уваги заслуговує реалізація користувацького веб-інтерфейсу на базі фреймворку Flask, що забезпечив зручний доступ до функціоналу системи, а також інтеграцію з модулями розпізнавання в режимі реального часу. Така архітектура дозволяє легко адаптувати систему для використання як локально, так і в хмарному середовищі.

Проведені експериментальні дослідження показали, що створена система демонструє високу точність у розпізнаванні жестів, стабільну роботу у реальному

часі та надає зручний зворотний зв'язок у формі голосових відповідей. Практична реалізація проєкту підтвердила ефективність обраного технологічного стека, обґрунтованість архітектурних рішень і доцільність інтеграції мовних моделей для побудови осмисленої комунікації.

Результати даної роботи мають не лише навчальну та дослідницьку цінність, а й значний прикладний потенціал. Розроблений прототип може бути використаний як основа для створення повноцінного цифрового асистента для людей з вадами мовлення. Система має значні перспективи розвитку: розширення словника розпізнаваних жестів, підтримка динамічних жестів, адаптація під інші мови жестів (наприклад, українську жестову мову), удосконалення механізмів синтезу мовлення, а також створення мобільної або мультиплатформеної версії застосунку.

Таким чином, результати дослідження свідчать про актуальність і практичну значущість обраної теми, а також про готовність розробленого рішення до подальшої апробації, оптимізації та впровадження у реальних умовах.

ПЕРЕЛІК ПОСИЛАНЬ

1. Bhat, N. M. (2021). *pyttsx3* [Official documentation]. <https://pyttsx3.readthedocs.io/en/latest> (Accessed: June 1, 2025)
2. Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32. <https://doi.org/10.1023/A:1010933404324>
3. Google AI for Developers. (2025). *Gemini API* [Official documentation]. <https://ai.google.dev/gemini-api/docs> (Accessed: June 1, 2025)
4. Google Cloud Community. (2024). *Function calling – A native framework to connect Gemini to external systems, data, and APIs*. <https://www.googlecloudcommunity.com/gc/Community-Blogs/Function-calling-A-native-framework-to-connect-Gemini-to/ba-p/713612> (Accessed: June 1, 2025)
5. Google. (2023). *MediaPipe – AI Edge Solutions Guide*. <https://ai.google.dev/edge/mediapipe/solutions> (Accessed: June 1, 2025)
6. Govindarajan, V., Damodharan, S., Kesavan, M., & Akash, K. (2023). Virtual personal assistant by using hand gesture and voice assistant for disabilities. In *2023 Intelligent Computing and Control for Engineering and Business Systems (ICCEBS)* (pp. 1–6). IEEE. <https://doi.org/10.1109/ICCEBS60127.2023.00005>
7. Jairam, B. G., & Ponnappa, D. (2023). Gesture based virtual assistant for deaf-mutes using deep learning approach. In *2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS)* (Vol. 1). IEEE. <https://doi.org/10.1109/ICACCS60170.2023.10160936>
8. Joshi, H., Golhar, V., Gundawar, J., Gangurde, A., Yenikar, A., & Sable, N. P. (2024). Real-time sign language recognition and sentence generation. *SSRN*. <https://doi.org/10.2139/ssrn.4992818> (Accessed: June 11, 2025)
9. Lodha, C. (2021). *American sign language assistant* (Master's project, Project No. 991).
10. Sarma, C. V., Gojemthi, H. B., Prasad, B., & Kumar, P. P. (2025). Real-time recognition of American sign language using Mediapipe and machine learning techniques. *Journal of Information Systems Engineering and Management*, 10(3), 01–15. <https://doi.org/10.55262/jisem.2025.10.3.01>
11. Satardekar, S. (2023). *Sign language recognition using machine learning* (Master's thesis, California State University, Northridge).
12. Shanmugam, S., Mahalakshmi, P., Lakshmanan, S. A., Sharmila, A., & Dhanasekaran, P. (2021). Hand gesture recognition using convolutional neural network. In *2021 Innovations in Power and Advanced Computing Technologies (i-PACT)*. IEEE. <https://doi.org/10.1109/i-PACT52855.2021.9696463>

13. Shekar, S. M. (2023). Sign language recognition using Python and OpenCV. *International Journal for Research in Applied Science and Engineering Technology*, 11(11), 2393–2399. <https://doi.org/10.22214/ijraset.2023.52549>
14. Someshwar, D., Bhanushali, V., Chaudhari, S., & Nadkarni, S. (2020). Implementation of virtual assistant with sign language using deep learning and TensorFlow. In *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)* (pp. 595–600). IEEE. <https://doi.org/10.1109/ICIRCA48905.2020.9183179>
15. Vijay, M., Bondada, V., Chitturi, S., Dhanush, S. M., & Hemanth, C. (2024). A virtual assistant with real-time sign-to-sign language translation for users with disabilities. In *2024 International Conference on IoT Based Control Networks and Intelligent Systems (ICICNIS)* (pp. 828–833). IEEE. <https://doi.org/10.1109/ICICNIS60399.2024.00692>
16. Кисіль Т.М., Шишук М.О., СУЧАСНІ ТРЕНДИ E-LEARNING /II Всеукраїнська науково-технічна конференція «Сучасні інтелектуальні інформаційні технології в науці та освіті». Збірник тез. – К.: ДУТ, 2022 р., с.39-41
17. Редчук, О. І. (2023). *Дослідження здатності штучного інтелекту до розпізнавання жестової мови* (Кваліфікаційна робота магістра, Тернопільський національний технічний університет імені Івана Пулюя).
18. Стецюк, А. О. (2022). *Система розпізнавання та перекладу мови жестів у текстову форму в режимі реального часу* (Магістерська дисертація, НТУУ «КПІ імені Ігоря Сікорського»).
19. Шишук М.О., Єльченко С.В., ВИКОРИСТАННЯ CHATGPT В NPC / III ВСЕУКРАЇНСЬКА НАУКОВО-ПРАКТИЧНА КОНФЕРЕНЦІЯ «СУЧАСНІ ІНТЕЛЕКТУАЛЬНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ В НАУЦІ ТА ОСВІТІ». Збірник тез. – К.: ДУТ, 2023, с. 95-97
20. Шишук М.О., Зінченко О.В., ЗАПРОВАДЖЕННЯ CHAT-GPT В ПОШУКОВИХ СИСТЕМАХ / IV ВСЕУКРАЇНСЬКА НАУКОВО-ПРАКТИЧНА КОНФЕРЕНЦІЯ «СУЧАСНІ ІНТЕЛЕКТУАЛЬНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ В НАУЦІ ТА ОСВІТІ». Збірник тез. – К.: ДУІКТ, 2024 р., с. 279-281

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ
ТЕХНОЛОГІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА ШТУЧНОГО ІНТЕЛЕКТУ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: "Інтерактивний цифровий асистент із підтримкою розпізнавання
жестової мови та синтезу голосових відповідей"

Виконала: здобувач вищої освіти гр. ШІД-42

Шишук Марія Олександрівна

Керівник: Зінченко Ольга Валеріївна

2025

Постановка задачі

2

Мета дослідження

Метою даної кваліфікаційної роботи є покращити наявні розробки інтерактивного цифрового асистента, який здатний розпізнавати жести американської жестової мови (ASL) за допомогою методів комп'ютерного зору та машинного навчання, а також генерувати відповідні голосові відповіді для забезпечення інклюзивного спілкування.

Об'єкт дослідження

Об'єктом дослідження є процеси інтерпретації жестової мови та її інтеграція в системи людино-машинної взаємодії.

Предмет дослідження

Предметом дослідження є методи та засоби реалізації розпізнавання жестів з відеопотоку в реальному часі, машинне навчання для класифікації жестів, а також технології синтезу мовлення на основі текстових результатів розпізнавання.

Основні завдання кваліфікаційної роботи:

1. Проаналізувати існуючі технології та системи для розпізнавання жестової мови та генерації голосових повідомлень.
2. Реалізувати компонент розпізнавання жестів на основі координат орієнтирів руки, отриманих за допомогою бібліотеки MediaPipe.
3. Побудувати модель машинного навчання за допомогою RandomForestClassifier для класифікації жестів
4. Інтеграція з великою мовною моделлю Gemini API
5. Імплементація системи синтезу мовлення

Актуальність роботи

Розвиток інклюзивних технологій та сучасних систем взаємодії людини з комп'ютером, що відкриває нові способи взаємодії з комп'ютером без використання клавіатури.

Проблеми, які вирішує застосунок

Застосунок вирішує проблему обмеженого доступу до цифрових технологій для користувачів, які використовують жестову мову. Він дозволяє взаємодіяти з комп'ютером за допомогою жестів, що перетворюються на голосові повідомлення, полегшуючи керування системами без клавіатури.

Програми-конкуренти:



SignAll



KinTrans



Hand Talk

Переваги проектного застосунку:

Даний застосунок розпізнає жести та озвучує повідомлення у реальному часі, що дозволяє керувати цифровим асистентом без фізичного введення. Цей застосунок працює на звичайному комп'ютері з веб-камерою, не потребує спеціального обладнання та демонструє високу швидкість й точність завдяки використанню MediaPipe та RandomForestClassifier.

Опис процесу роботи програми

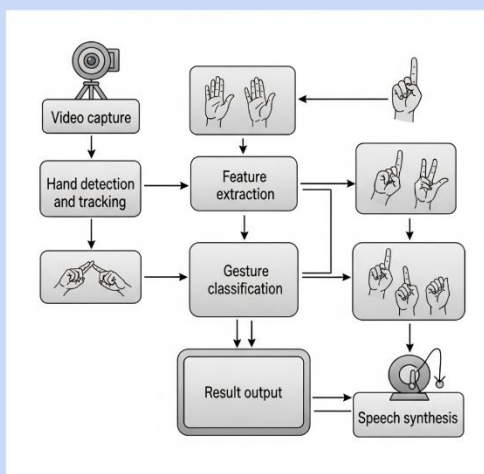


Рис. 1 Загальна архітектура системи розпізнавання жестів

Використані інструменти та технології

- MediaPipe
- OpenCV
- pickle
- Gemini API
- pyttsx3

Створення власного набору даних

- Було створено набір даних з 5800 зображень
- Кожний символ містить 200 зображень
- Було створено спеціальні службові символи: "START", "SPACE", "SEND".

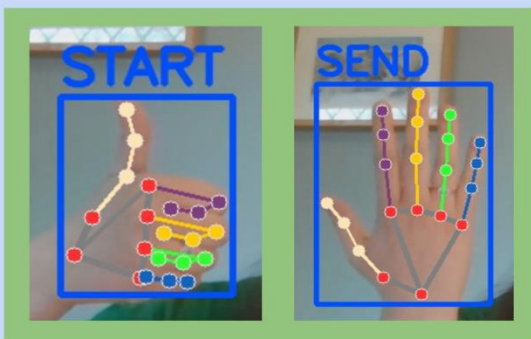


Рис. 2 Службові жести

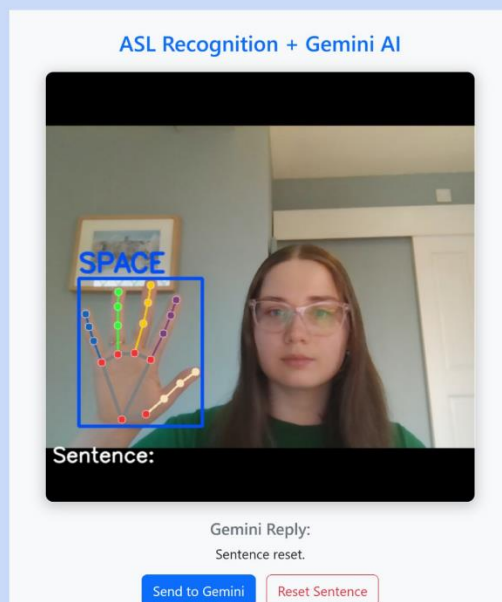


Рис. 3 Службовий жест

RandomForestClassifier

RandomForestClassifier — це метод машинного навчання, який використовується для класифікації даних. Його основна ідея полягає у створенні великої кількості дерев рішень, кожне з яких навчається на випадковій вибірці даних. Остаточне рішення приймається на основі голосування всіх дерев.

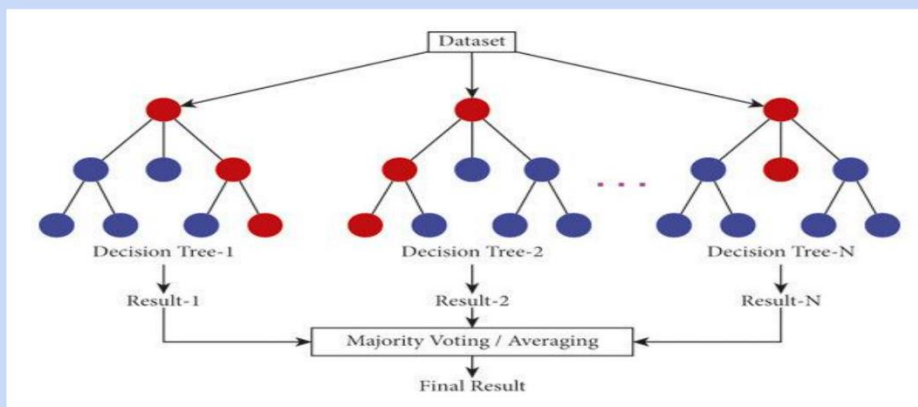


Рис. 4 Візуальне представлення класифікатора випадкового лісу

Приклад застосування розробленої програми

- сфера доступних технологій та інтерфейсів для взаємодії з комп'ютером.
- альтернативні методи керування цифровими системами
- у наукових дослідженнях, пов'язаних із інтерфейсами людина-комп'ютер.



Рис. 5 Виведення відповіді від Gemini API на екран

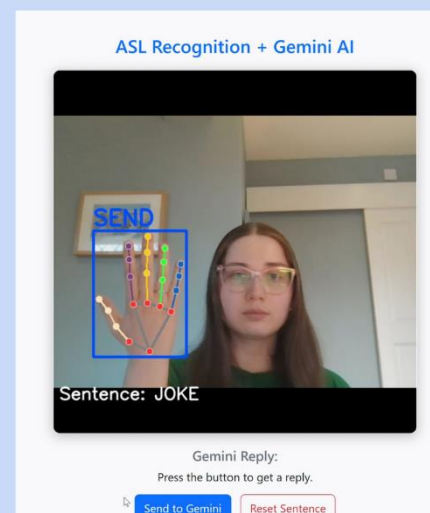
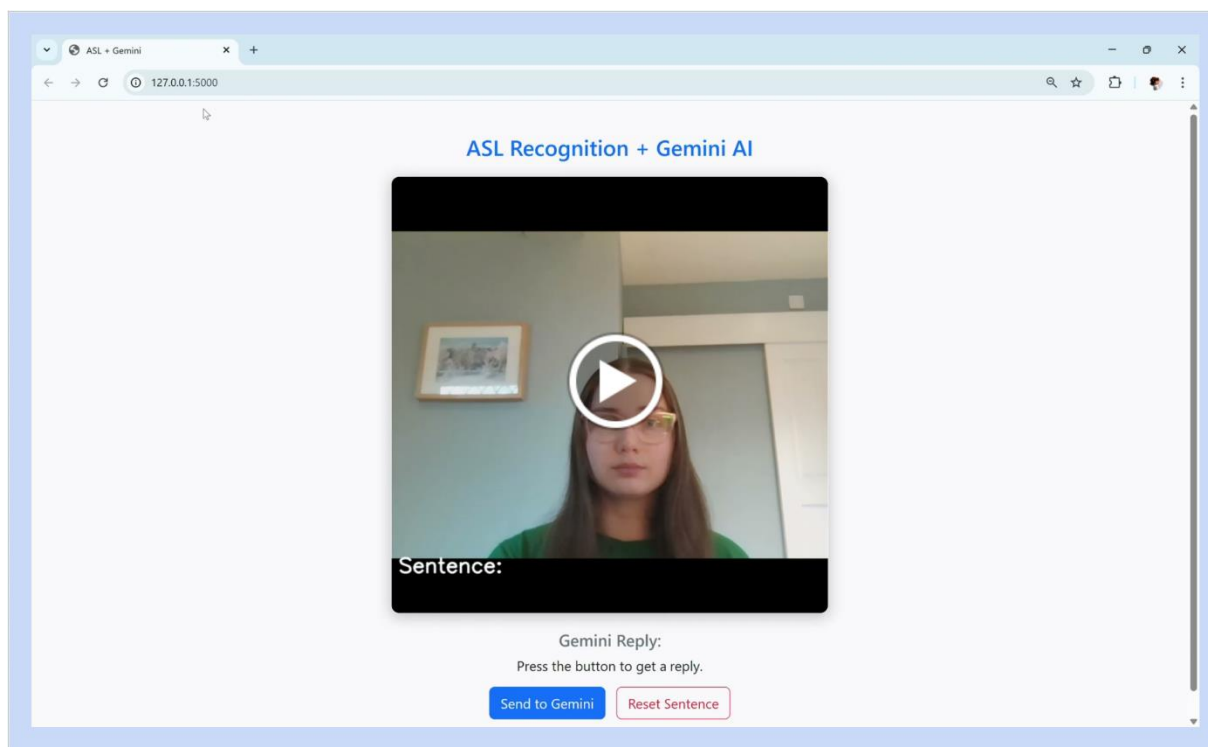


Рис. 6 Інтерфейс застосунку

Оцінка моделі

Клас жесту	Точність (Precision)	Повнота (Recall)	F1-міра (F1-score)	Кількість зразків (Support)	Клас жесту	Точність (Precision)	Повнота (Recall)	F1-міра (F1-score)	Кількість зразків (Support)
A	1.00	1.00	1.00	40	O	1.00	1.00	1.00	40
B	1.00	1.00	1.00	40	P	1.00	1.00	1.00	40
C	1.00	1.00	1.00	40	Q	1.00	1.00	1.00	40
D	1.00	1.00	1.00	40	R	1.00	1.00	1.00	40
E	1.00	1.00	1.00	40	S	1.00	1.00	1.00	40
F	1.00	1.00	1.00	40	SEND	1.00	1.00	1.00	40
G	1.00	1.00	1.00	40	SPACE	1.00	1.00	1.00	40
H	1.00	1.00	1.00	40	START	1.00	1.00	1.00	40
I	1.00	1.00	1.00	40	T	1.00	1.00	1.00	40
J	1.00	0.97	0.99	40	U	1.00	1.00	1.00	40
K	1.00	1.00	1.00	40	V	1.00	1.00	1.00	40
L	1.00	1.00	1.00	40	W	1.00	1.00	1.00	40
M	1.00	1.00	1.00	40	X	1.00	1.00	1.00	40
N	1.00	1.00	1.00	40	Y	1.00	1.00	1.00	40
					Z	0.98	1.00	0.99	40

Табл. 1 Класифікації моделі розпізнавання жестів



Висновки

Розроблено інтерактивний цифровий асистент, здатний розпізнавати жести американської жестової мови (ASL) та озвучувати повідомлення-відповіді. Для реалізації обрано MediaPipe, OpenCV, scikit-learn та pyttsx3.

Створено власний набір даних, виконано попередню обробку інформації та навчено модель класифікації на базі алгоритму RandomForestClassifier.

Створено інтерактивний інтерфейс для зручної взаємодії з програмою. Отриманий програмний продукт дозволяє ефективно здійснювати взаємодію між людиною та комп'ютером за допомогою мови жестів.