Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)
Кафедра Інформатики
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАСТОСУНКУ ДЛЯ РОЗПІЗНАВАННЯ ЖЕСТОВОЇ МОВИ У РЕАЛЬНОМУ ЧАСІ
(тема)

Виконав:
здобувач 4 року навчання,
групи ІТІНФ-21-1
Мармур О. М.
(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

Керівник доц. Творошенко І. С.
(посада, прізвище, ініціали)

Допускається до захисту

Завідувач кафедри інформатики _____ Кобилін О. А.
(підпис) (прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту

Кафедра Інформатики

Рівень вищої освіти перший (бакалаврський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«____»_____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Мармуру Олексію Миколайовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розроблення програмного застосунку для розпізнавання жестової мови у реальному часі

затверджена наказом університету від 19 травня 2025 року № 381Ст
2. Термін подання здобувачем роботи до екзаменаційної комісії 23 травня 2025 р.
3. Вихідні дані до роботи науково-методична та науково-технічна література, матеріали конференцій, мова програмування Python, бібліотека комп'ютерного зору з відкритим кодом OpenCV, бібліотека MediaPipe для трекінку рук, бібліотека TenserFlow для налаштування нейронних мереж, бібліотека Tkinter для створення користувацього інтерфейсу.

4. Перелік питань, що потрібно опрацювати в роботі
1. Аналіз сучасних рішень та технологій у сфері розпізнавання жестової мови.
2. Дослідити особливості передачі інформації жестовими засобами.
3. Формування методики розпізнавання української жестової мови у реальному часі.
4. Вибір інструметальних засобів реалізації програмного застосунку для розпізнавання жестової мови у реальному часі.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Актуальність проблеми розпізнавання української жестової мови, постановка задачі, схеми методів розпізнавання жестової мови у реальному часі, блок-схеми алгоритму розпізнавання жестової мови у реальному часі.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

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

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	07.04.2025	
2	Аналіз завдання, підбір літератури	08.04.25-10.04.25	
3	Аналіз літератури з досліджуваної проблеми	11.04.25-14.04.25	
4	Аналіз методів розпізнавання жестової мови	15.04.25-20.04.25	
5	Розробка методу розізнавання жестової мови у реальному часі	21.04.25-27.04.25	
6	Програмна реалізація	28.04.25-11.05.25	
7	Оформлення пояснювальної записки	12.05.25-20.05.25	
8	Перевірка на нормоконтроль	21.05.25-01.06.25	
9	Перевірка на плагіат	21.05.25-01.06.25	
10	Рецензування	21.05.25-01.06.25	
11	Підготовка презентації та доповіді	21.05.25-18.06.25	
12	Занесення роботи в електронний архів	02.06.25-18.06.25	
13	Попередній захист кваліфікаційної роботи	02.06.25-18.06.25	

Дата видачі завдання 7 квітня 2025 р.

Здобувач _____
(підпис)

Керівник роботи _____ _____ доц. Творошенко І. С.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 78 с., 44 рис., 30 джерел.

РОЗПІЗНАВАННЯ ЖЕСТОВОЇ МОВИ, КОМП'ЮТЕРНИЙ ЗІР, МАШИННЕ НАВЧАННЯ, НЕЙРОННІ МЕРЕЖІ, РЕКУРЕНТНІ НЕЙРОННІ МЕРЕЖІ, ЗГОРТКОВІ НЕЙРОННІ МЕРЕЖІ, РЕАЛЬНИЙ ЧАС.

Об'єктом роботи є розпізнавання жестової мови в реальному часі за допомогою програмного застосунку, який використовує методи комп'ютерного зору та машинного навчання.

Метою роботи є розроблення програмного застосунку, здатного розпізнавати жести української жестової мови та перетворювати їх у текстові повідомлення для покращення комунікації між людьми з порушеннями слуху та тими, хто не володіє жестовою мовою.

Використано методи згорткових нейронних мереж та рекурентних нейронних мереж для аналізу відеопотоку та класифікації жестів. Проведено дослідження методів детектування ключових точок рук, обробки послідовностей кадрів та навчання моделей на основі зібраних даних.

У результаті роботи здійснена програмна реалізація системи, яка дозволяє збирати дані для 108 базових українських жестів мовлення, навчати модель та розпізнавати жести в реальному часі.

SIGN LANGUAGE RECOGNITION, COMPUTER VISION, MACHINE LEARNING, NEURAL NETWORKS, RECURRENT NEURAL NETWORKS, CONVOLUTIONAL NEURAL NETWORKS, REAL TIME.

The object of the work is real-time sign language recognition using a software application that uses computer vision and machine learning methods.

The aim of the work is to develop a software application capable of recognizing Ukrainian sign language gestures and converting them into text messages to improve communication between people with hearing impairments and those who do not speak sign language.

Methods of convolutional neural networks and recurrent neural networks were used to analyze the video stream and classify gestures. Methods for detecting key points of hands, processing sequences of frames, and training models based on the collected data were studied.

As a result of the work, a software implementation of the system was carried out, which allows collecting data for 108 basic Ukrainian speech gestures, training the model and recognizing gestures in real time.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів 7
Вступ 9
1 Аналіз основних існуючих програмних застосунків для розпізнавання жестової мови 10
1.1 Аналіз сучасних застосунків для розпізнавання жестової мови в Україні та за кордоном 10
1.2 Особливості передачі букв, слів та словосполучень за допомогою жестів 22
1.3 Аналіз літературних джерел щодо апробації результатів розроблення програмних застосунків для розпізнавання жестової мови у реальному часі 24
1.4 Постановка задачі 27
2 Особливості розроблення логічної складової програмного застосунку для розпізнавання жестової мови у реальному часі 29
2.1 Детальний аналіз існуючих методів розпізнавання жестової мови 29
2.1.1 Метод на основі ключових точок руки 29
2.1.2 Розпізнавання на основі згорткових нейронних мереж 33
2.1.3 Метод із використанням рекурентних нейронних мереж 37
2.1.4 Аналіз положення тіла за допомогою системи OpenPose 39
2.2 Формування методики розпізнавання української жестової мови у реальному часі 42
3 Етапи розроблення програмного застосунку для розпізнавання жестової мови у реальному часі 49
3.1 Вибір інструментальних засобів реалізації програмного застосунку для розпізнавання жестової мови у реальному часі 49
3.2 Етапи програмної реалізації розпізнавання жестової мови у реальному часі 52
3.3 Тестування розробленого застосунку та аналіз результатів 66

3.4 Перспективи подальшої роботи 72
Висновки 74
Перелік джерел посилання 75

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

УЖМ – українська жестова мова

ASL – American Sign Language (американська жестова мова)

ASR – Automatic Speech Recognition (автоматичне розпізнавання мовлення)

TTS – Text-to-Speech (синтез мовлення з тексту)

Google Lens – технологія комп'ютерного зору від Google, яка дозволяє камері смартфона розпізнавати об'єкти, текст, QR-коди, перекладати надписи, ідентифікувати рослини, тварин, будівлі тощо

Pixel Motion Sense – система розпізнавання жестів у смартфонах Google Pixel

AI – Artificial Intelligence (штучний інтелект)

ML – Machine Learning (машинне навчання)

CV – Computer Vision (комп'ютерний зір)

YOLOv4 – You Only Look Once version 4 (алгоритм глибокого навчання для реального часу розпізнавання об'єктів на зображеннях)

LSTM – Long Short-Term Memory (тип рекурентної нейронної мережі, який ефективно працює з послідовностями, зокрема відео)

CNN – Convolutional Neural Network (згорткова нейронна мережа, що використовується для обробки зображень)

FPS – Frames Per Second (кількість кадрів за секунду при обробці відео)

IoT – Internet of Things (інтернет речей, взаємопов'язані пристрої, які збирають та обмінюються даними)

ReLU – Rectified Linear Unit (функція активації, що передає лише додатні значення)

Субдискретизація – зменшення розмірів зображення в CNN

TensorFlow – бібліотека машинного та глибокого навчання від Google

Keras – високорівнева бібліотека для нейронних мереж на Python

PyTorch – бібліотека глибокого навчання з відкритим кодом від Meta

RNN – Recurrent Neural Network (рекурентна нейронна мережа для послідовностей)

OpenPose – система визначення поз людини на зображенні

MoveNet – легка модель для визначення скелету людини в реальному часі

MobileNet – компактна згорткова нейронна мережа для мобільних пристроїв

SVM – Support Vector Machine (метод опорних векторів, алгоритм машинного навчання для класифікації)

Inception – архітектура CNN з паралельними фільтрами

EfficientNet – оптимізована архітектура CNN з високою точністю

Fine-tuning – донавчання попередньо натренованої моделі

Kinect – сенсорний пристрій Microsoft для захоплення глибинного зображення та відстеження рухів у тривимірному просторі

HOG – Histogram of Oriented Gradients (алгоритм виявлення об’єктів на зображеннях за напрямками градієнтів)

Random Forest – ансамблевий алгоритм машинного навчання, що поєднує множину дерев рішень для підвищення точності класифікації чи регресії

Dropout – це техніка регуляризації, яка використовується в нейронних мережах для запобігання перенавчанню

ВСТУП

Одним із важливих викликів для сучасного суспільства є створення умов, у яких кожен – незалежно від фізичних особливостей – мав би однаковий доступ до інформації та можливостей для спілкування. Для людей із порушеннями слуху головним засобом комунікації є жестова мова. Вона дозволяє передавати не лише зміст, а й емоції чи наміри. Проте виникає проблема: більшість людей, які чують, просто не розуміють жестову мову. Через це людям з вадами слуху часто складно повноцінно взаємодіяти з оточенням – і в роботі, і в навчанні, і в повсякденному житті.

На щастя, технології не стоять на місці. Сьогодні, завдяки розвитку комп'ютерного зору та машинного навчання, з'явилися рішення, які можуть «читати» жести в режимі реального часу [1]. Це відкриває нові можливості для створення інструментів, які перетворюють жести на текст або мову, роблячи спілкування простішим для всіх сторін.

Розпізнавання жестової мови – це не просто вловити рух руки. Важливо враховувати ще й міміку, положення тіла, інколи навіть контекст розмови. До того ж, кожна країна має свою унікальну жестову мову, тому не можна створити один універсальний перекладач «на всі випадки». Наприклад, українська жестова мова має свою структуру, і для її точного розпізнавання потрібна окрема адаптація алгоритмів.

Створення програми, яка розпізнає жести в реальному часі – це не просто цікавий технічний проєкт. Це важливий крок до більш доступного, відкритого і толерантного суспільства, де кожна людина має право на голос, навіть якщо вона не може говорити словами.

1 АНАЛІЗ ОСНОВНИХ ІСНУЮЧИХ ПРОГРАМНИХ ЗАСТОСУНКІВ ДЛЯ РОЗПІЗНАВАННЯ ЖЕСТОВОЇ МОВИ

1.1 Аналіз сучасних застосунків для розпізнавання жестової мови в Україні та за кордоном

У сучасному світі, де інформація є одним з основних ресурсів, забезпечення рівного доступу до неї для всіх верств населення набуває критичного значення. Особливо це стосується осіб із порушеннями слуху, для яких жестова мова є основним засобом комунікації. Часто вони стикаються з труднощами у повсякденному житті, коли необхідно взаємодіяти з оточуючими, які не володіють жестовою мовою. У цьому контексті технології для розпізнавання жестів у реальному часі стають ключовим фактором для покращення комунікації і забезпечення повноцінної інтеграції таких людей у суспільство.

Нові програмні платформи для розпізнавання жестової мови, що активно впроваджуються в Україні та за кордоном, є важливими інструментами для вирішення цієї проблеми [2]. Вони використовують камери смартфонів або інших пристроїв для відстеження рухів рук, які перетворюються спеціальними алгоритмами на текст або аудіо. Це не лише полегшує комунікацію в повсякденному житті, але й дає людям з порушеннями слуху можливість більш активно брати участь у професійній, освітній та соціальній діяльності. Одним із ключових аспектів таких програм є їх доступність і простота використання – завдяки цьому вони можуть бути застосовані у будь-який момент і в будь-якому місці.

Технології для розпізнавання жестової мови базуються на передових методах машинного навчання та комп’ютерного зору. Використовуючи глибоке навчання, ці системи здатні адаптуватися до різних жестових мов, оскільки в різних країнах існують власні національні варіанти жестової мови.

Такий підхід дозволяє створювати універсальні програми, які можуть бути ефективно застосовані на глобальному рівні. Сучасні алгоритми здатні не лише розпізнавати широкий спектр жестів, але й вчитися новим, що значно підвищує точність і ефективність таких рішень.

Одним із важливих проєктів у сфері розпізнавання жестової мови є розробка застосунку, над яким працюють фахівці та студенти Київського політехнічного інституту імені Ігоря Сікорського (КПІ). Зазначений інструментальний засіб має на меті стати зручним інструментом, який дозволить за допомогою камери смартфона або іншого мобільного пристрою розпізнавати жести української жестової мови (УЖМ) у реальному часі та перекладати їх на текст або голосове повідомлення. Це може значно полегшити спілкування між людьми з порушеннями слуху та тими, хто не володіє жестовою мовою (рис. 1.1) [3].

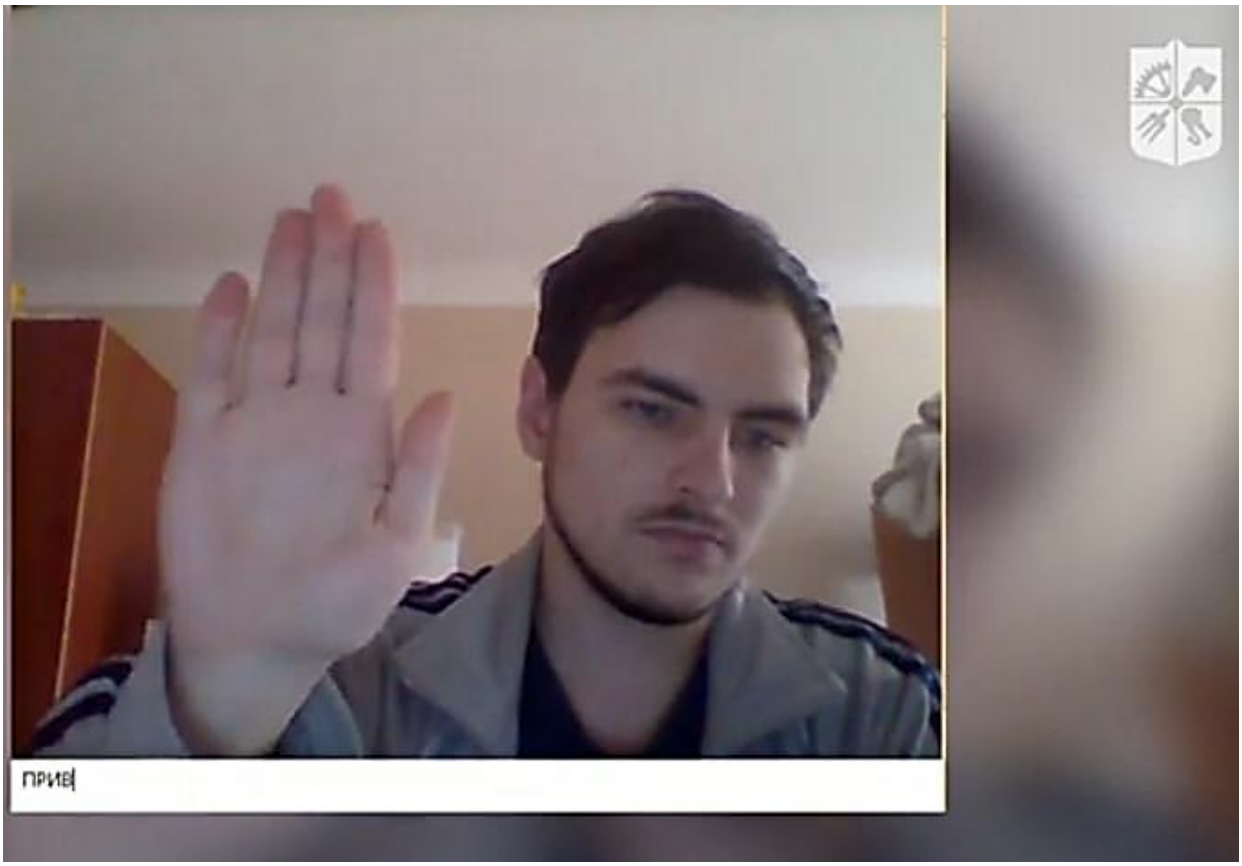

Рисунок 1.1 – Приклад роботи застосунку розпізнавання жестової мови Київського політехнічного інституту імені Ігоря Сікорського

Розробка платформи базується на сучасних методах машинного навчання та комп’ютерного зору. Команда використовує глибокі нейронні мережі, які навчаються на великій кількості зображень та відео з жестами УЖМ. Це дозволяє системі ефективно розпізнавати навіть складні жести та пристосовуватися до індивідуальних особливостей рухів різних людей.

Алгоритми застосунку також враховують міміку та положення тіла, оскільки в жестовій мові це має велике значення. Наприклад, один і той самий жест рук може мати різні значення залежно від виразу обличчя або нахилу голови. Тому система аналізує комплексний набір параметрів, щоб забезпечити максимально точний переклад.

Однією з головних цілей розробників було зробити інструментальний засіб максимально простим і зручним у використанні. Інтерфейс платформи інтуїтивно зрозумілий, з чітко позначеними кнопками та меню. Користувач може швидко розпочати роботу без необхідності довгого навчання чи налаштувань.

Застосунок також підтримує інтеграцію з популярними сервісами, такими як соціальні мережі та месенджери. Це означає, що користувачі можуть обмінюватися повідомленнями з друзями та колегами безпосередньо через інструментальний засіб, що робить спілкування ще більш зручним і доступним.

Розробники платформи зіткнулися з низкою викликів:

– технічні обмеження: забезпечення високої точності розпізнавання жестів у різних умовах освітлення та на фоні різних зображень є складним завданням. Наприклад, у темному приміщенні або при яскравому сонячному світлі камера може не точно фіксувати рухи, що впливає на якість розпізнавання;

– варіативність жестів: різні люди можуть виконувати один і той самий жест по-різному. Хтось рухає рукою швидше, хтось повільніше, у когось можуть бути індивідуальні особливості рухів. Це вимагає від алгоритмів високої гнучкості та здатності навчатися на різноманітних даних;

– розширення бази даних жестів: українська жестова мова є багатою та різноманітною. Щоб інструментальний засіб міг розпізнавати всі можливі жести, необхідно постійно поповнювати та оновлювати базу даних. Це трудомісткий процес, який вимагає співпраці з експертами та спільнотою людей з порушеннями слуху.

Перспективи розвитку застосунку є досить оптимістичними.

Розробники планують:

– вдосконалення алгоритмів: підвищення точності та швидкості розпізнавання жестів за допомогою новітніх методів машинного навчання;

– розширення функціоналу: додавання можливостей, таких як навчання новим жестам безпосередньо користувачами або підтримка інших мов жестів;

– інтеграція з іншими пристроями: використання розумних годинників або окулярів доповненої реальності для більш природної взаємодії;

– впровадження доповненої реальності: це може дозволити відображати переклад жестів безпосередньо в полі зору користувача, що зробить процес спілкування ще більш зручним.

Загалом, цей проєкт має великий потенціал і може зробити значний внесок у покращення комунікації та якості життя людей з порушеннями слуху в Україні.

Одним із провідних міжнародних проєктів у сфері розпізнавання жестової мови є компанія зі Сполучених Штатів Америки – SignAll. Вони розробили першу у світі систему, яка автоматично перекладає жестову мову, що використовується у Сполучених Штатах (ASL) на текст у реальному часі (рис. 1.2) [4].

Мета цього проєкту – подолати комунікаційний бар’єр між людьми з порушеннями слуху та тими, хто не володіє жестовою мовою.

SignAll використовує передові технології комп’ютерного зору та машинного навчання для точного розпізнавання жестів.

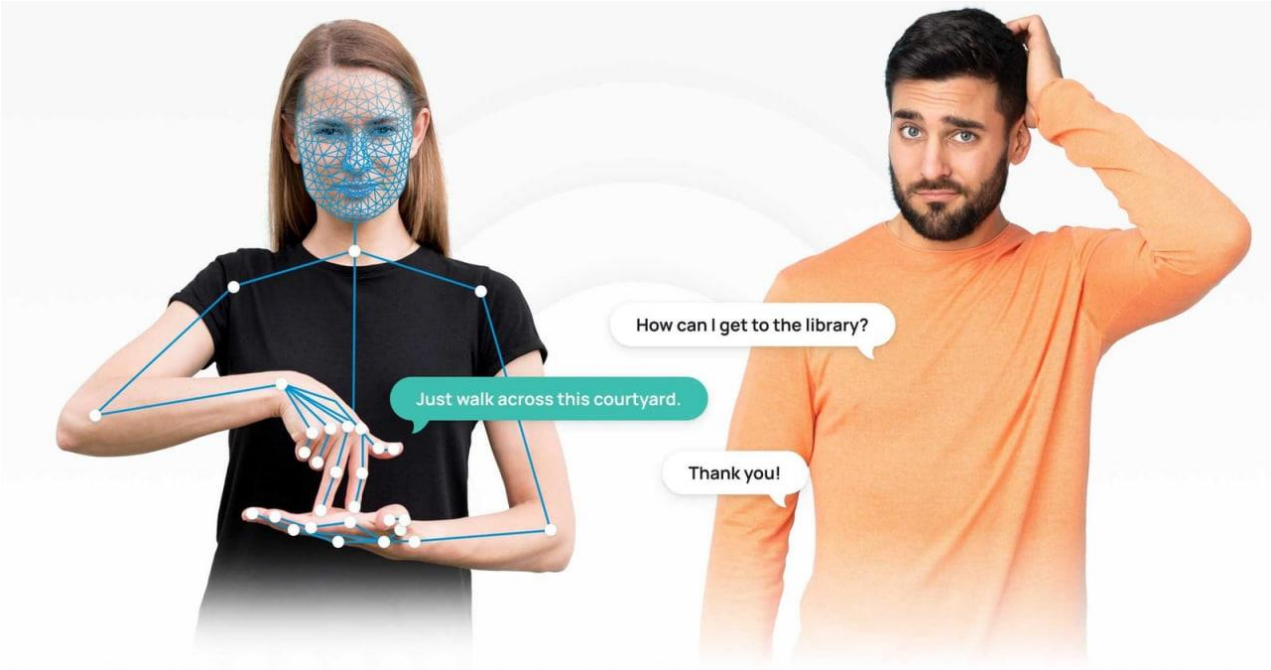

Рисунок 1.2 – Приклад реалізації застосунку розпізнавання жестової мови, розробленого компанією SignAll

Система складається з декількох ключових компонентів:

– мультимодальна система відстеження: вона включає кілька камер та сенсорів, які одночасно відстежують рухи рук, пальців, тіла та міміки обличчя. Це дозволяє системі захоплювати тривимірні рухи з високою точністю;

– алгоритми глибокого навчання: система навчена на великому обсязі даних ASL. Використовуючи нейронні мережі, вона здатна розпізнавати широкий спектр жестів та навіть їхні нюанси;

– база даних жестів та словник: містить тисячі жестів ASL та їхні еквіваленти в англійській мові, що дозволяє забезпечити точний та контекстуальний переклад.

SignAll прагне зробити свою технологію максимально доступною та зручною у використанні:

– кіоски SignAll: установлюються в громадських місцях, таких як бібліотеки, університети та офіси. Люди з порушеннями слуху можуть спілкуватися з персоналом або іншими відвідувачами, використовуючи жестову мову, яка автоматично перекладається на текст;

– програмне забезпечення для персональних комп’ютерів: компанія пропонує рішення, яке можна встановити на комп’ютер з підключеними камерами. Це особливо корисно для віддаленого спілкування або роботи;

– інтеграція з освітніми платформами: SignAll активно співпрацює з навчальними закладами для впровадження своєї технології в освітній процес.

Розробка та впровадження технології SignAll супроводжуються певними викликами:

– складність жестової мови: ASL має свою граматику та структуру, яка відрізняється від англійської мови. Переклад не тільки окремих слів, але й повних речень з урахуванням граматичних правил є складним завданням;

– технічні вимоги: для роботи системи потрібне спеціальне обладнання, включаючи кілька камер та сенсорів, що може бути дорогим та складним для встановлення;

– індивідуальні особливості жестів: різні люди можуть виконувати жести по-різному, що вимагає від системи високої гнучкості та адаптивності.

Перспективи розвитку SignAll включають:

– оптимізація обладнання: розробка більш компактних та доступних пристроїв для зниження вартості та спрощення впровадження;

– розширення на інші жестові мови: компанія планує адаптувати свою технологію для підтримки інших національних жестових мов, що розширить її глобальне значення;

– покращення алгоритмів: постійне вдосконалення алгоритмів машинного навчання для підвищення точності та швидкості розпізнавання.

SignAll є піонером у галузі автоматичного перекладу жестової мови на текст. Їхня технологія має потенціал суттєво покращити якість життя людей з порушеннями слуху, надаючи їм інструменти для більш вільного та незалежного спілкування. Досвід SignAll може бути цінним для розробників у інших країнах, включаючи Україну, де подібні рішення можуть адаптуватися для української жестової мови та сприяти соціальній інтеграції.

Розробка застосунків для розпізнавання жестової мови відкриває нові можливості у спілкуванні між людьми з порушеннями слуху та тими, хто не володіє жестовою мовою. Такі застосунки, як ті, над якими працюють у КПІ та компанія SignAll у США, демонструють потенціал сучасних технологій у цій галузі.

Переваги таких платформ полягають у тому, що вони дозволяють у реальному часі перекладати жести на текст або голосові повідомлення, роблячи спілкування більш доступним і зручним. Використання методів машинного навчання та комп'ютерного зору забезпечує високу точність розпізнавання, враховуючи навіть міміку та положення тіла. Інтеграція з соціальними мережами та месенджерами розширює можливості взаємодії, роблячи процес спілкування більш природним [5].

Однак, недоліки також присутні. Технічні обмеження, такі як різні умови освітлення або фон, можуть впливати на точність розпізнавання жестів. Варіативність виконання жестів різними людьми вимагає від алгоритмів високої гнучкості та здатності адаптуватися до індивідуальних особливостей. Крім того, постійне оновлення та розширення бази даних жестів є трудомістким процесом, що потребує співпраці з експертами та спільнотою.

Такий застосунок повинен мати інтуїтивно зрозумілий інтерфейс, щоб користувачі могли легко розпочати роботу без додаткового навчання. Важливо, щоб він був оптимізований для роботи на стандартних мобільних пристроях без потреби в додатковому обладнанні. Застосунок має враховувати комплексний набір параметрів – рухи рук, міміку, положення тіла – для забезпечення максимально точного перекладу. Інтеграція з різними платформами та можливість персоналізації під потреби користувача зробити його ще більш ефективним.

Прикладом інноваційної розробки у сфері розпізнавання альтернативних форм комунікації, зокрема невербального мовлення, є Project Relate – експериментальний застосунок компанії Google (рис. 1.3) [6].

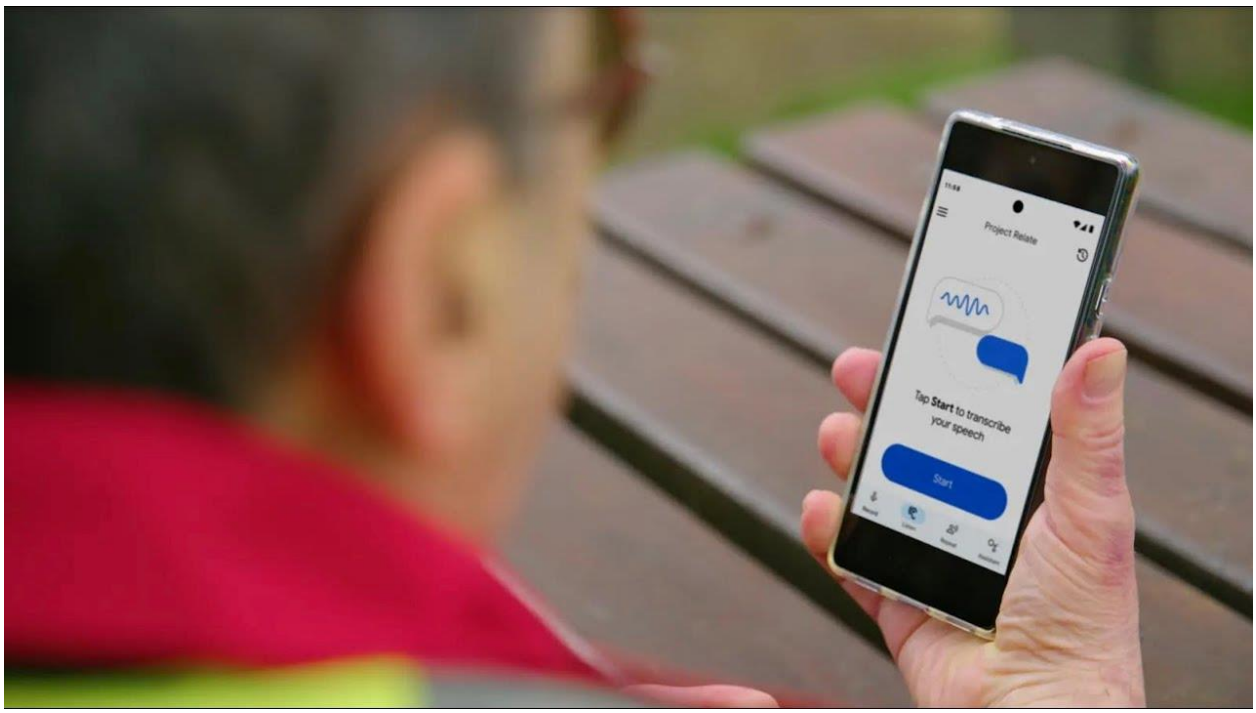

Рисунок 1.3 – Приклад роботи Project Relate

Ідея створення цієї платформи виникла як відповідь на потребу людей з порушеннями мовлення, вимова яких значно відрізняється від загальноприйнятих норм, зокрема внаслідок інсульту, церебрального паралічу або інших неврологічних станів. Хоча першочерговою метою проєкту є покращення доступу до вербальної комунікації для людей з унікальним мовним профілем, технологія, яка лежить в його основі, створює підґрунтя для подальшого розвитку у напрямку розпізнавання жестової мови як однієї з форм природної взаємодії.

Особливістю Project Relate є його персоналізований підхід: користувач проходить індивідуальне навчання системи, записуючи понад 500 типових фраз. На основі цих записів система створює мовну модель, яка дозволяє точно трансформувати унікальне мовлення або поведінкові сигнали користувача в текст або озвучене повідомлення. Це рішення базується на новітніх досягненнях у сфері штучного інтелекту, зокрема технологіях автоматичного мовного розпізнавання (ASR) та синтезу мовлення (TTS), які вже давно застосовуються в таких продуктах Google, як Assistant, Translate та Speech Services.

Проте Relate вирізняється глибокою адаптивністю та можливістю роботи із нестандартними зразками мовлення.

Застосунок має три основні функції:

– «Listen», яка дозволяє трансформувати мовлення на текст у реальному часі;

– «Repeat», що озвучує сказане чітким голосом для слухача;

– «Assistant», яка дає змогу взаємодіяти з віртуальним помічником Google, надсилаючи команди, наприклад, для керування пристроями чи пошуку інформації.

Завдяки цим функціям користувачі з унікальним мовленням можуть ефективно взаємодіяти з цифровими системами, що відкриває можливість застосування цієї технології також у поєднанні з іншими каналами комунікації – наприклад, жестами.

У контексті розвитку систем розпізнавання жестової мови, Project Relate викликає особливий інтерес через потенціал інтеграції з візуальними сенсорами Android-пристроїв. Уже зараз Google активно впроваджує в інші свої сервіси елементи комп’ютерного зору, здатні розпізнавати рухи, жести, міміку – наприклад, у Google Lens або Pixel Motion Sense. У майбутньому ці напрацювання можуть бути об’єднані із Relate в єдину екосистему, яка дозволить не лише розпізнавати мовлення, але й враховувати невербальні сигнали, включно з жестами.

Важливим аспектом є також фокус на приватності та етичності. Усі персональні мовні дані користувача зберігаються локально або передаються на сервери Google лише з дозволу користувача. Крім того, система навчається окремо для кожного користувача, не використовуючи ці дані для масового навчання, що дає змогу зберігати високу точність персоналізації без ризику для конфіденційності. Такий підхід є особливо важливим у контексті розпізнавання жестової мови, адже персоніфіковані рухи та жести можуть містити конфіденційну інформацію [7].

Попри те, що на момент написання цієї роботи Project Relate ще не підтримує повноцінне розпізнавання жестів як основного каналу комунікації, його технологічна база та відкритість до інтеграції створюють сприятливі умови для подальшого розвитку у цьому напрямі. Завдяки гнучкості архітектури застосунок може стати платформою для мультиканальної комунікації, де мовлення, текст, жести та міміка взаємодіятимуть у єдиній системі.

Google має достатній ресурсний потенціал, щоб масштабувати подібні рішення на глобальному рівні. Проєкти, подібні до Project Relate, демонструють зміну парадигми у сфері технологій доступності: від універсальних рішень до персоналізованих, гнучких інструментів, які пристосовуються до конкретної людини. Це особливо актуально для спільноти людей з порушеннями слуху або мовлення, адже саме вони часто потребують максимально індивідуального підходу у засобах комунікації.

Таким чином, навіть якщо Project Relate не є класичним прикладом застосунку для розпізнавання жестової мови, його концепція, технічна реалізація та потенціал інтеграції із засобами комп’ютерного зору дають усі підстави розглядати його як інноваційний аналог, що здатен трансформувати підходи до невербальної комунікації у найближчому майбутньому.

Міжнародним проєктом, який заслуговує на увагу в контексті розробки систем для розпізнавання жестової мови, є KinTrans (рис. 1.4) [8].

Платформа була створена у США як відповідь на потребу у забезпеченні безбар’єрної комунікації для людей з порушеннями слуху у громадських просторах, сервісних структурах та освітніх установах.

Основна ідея проєкту полягає у створенні системи, яка може у реальному часі зчитувати жести людини і трансформувати їх у текст або голосове повідомлення, забезпечуючи двостороннє розуміння між особами, які користуються різними способами мовлення.

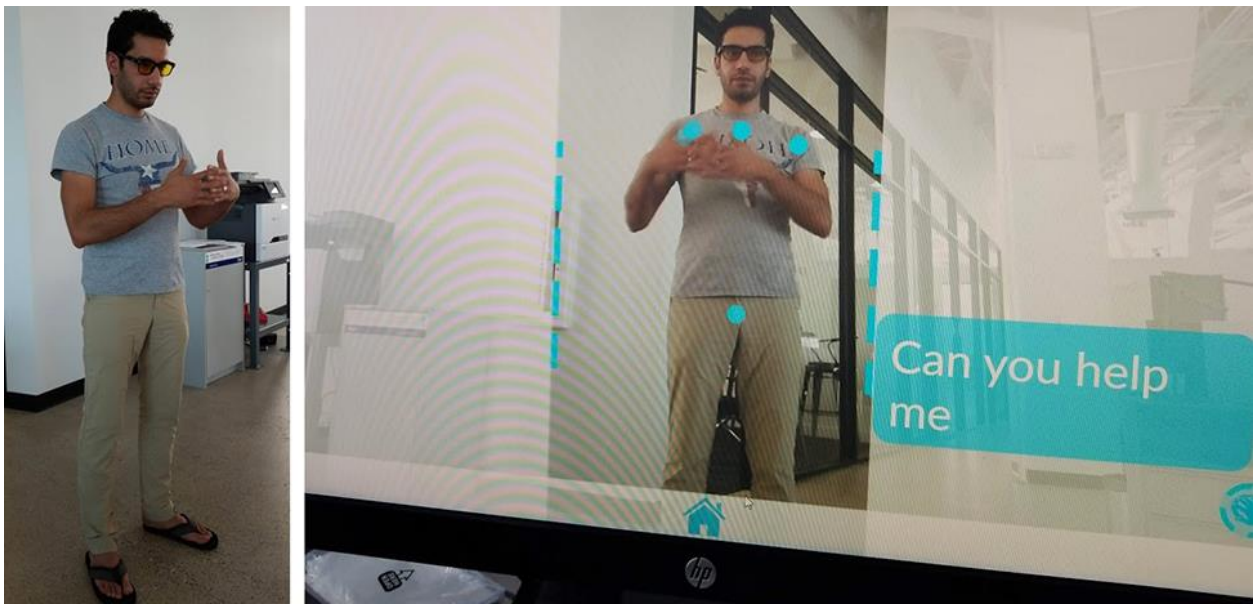

Рисунок 1.4 – Приклад роботи KinTrans

KinTrans використовує технології тривимірного аналізу руху за допомогою спеціальних камер і сенсорів, які здатні відстежувати положення тіла, рук, пальців і навіть міміку обличчя. Завдяки цьому система не просто аналізує окремі жести, а формує цілісний образ висловлювання, враховуючи контекст та емоційне забарвлення. Це особливо важливо, оскільки жестова мова, на відміну від вербальної, часто є синкретичною – в одному русі можуть поєднуватися лексичний зміст, граматичні конструкції та емоційна інтонація.

Платформа може бути встановлена як у вигляді стаціонарного пристрою (наприклад, у вигляді кіоску у банку або лікарні), так і на звичайному персональному комп’ютері з відповідним обладнанням. Розробники прагнуть зробити систему максимально зручною для користувача, інтерфейс інтуїтивно зрозумілий, запуск не потребує спеціальної підготовки, а адаптація до нових умов відбувається автоматично. Це дає змогу використовувати KinTrans у різноманітних середовищах, не обмежуючись лише спеціалізованими закладами.

Система підтримує кілька національних жестових мов, включаючи американську (ASL), арабську, французьку та іспанську. Це значно розширює її сферу застосування і робить її потенційно корисною для інтеграції в

мультинаціональних країнах, а також у міжнародних організаціях, де важливо забезпечити рівний доступ до інформації для всіх. Підтримка багатьох мов жестів здійснюється через гнучку архітектуру системи, яка дозволяє додавати нові мовні бази через оновлення.

Окрім цього, KinTrans має функцію двостороннього перекладу – система не лише «розуміє» жести, а й може перетворити голос або текст на візуалізований жест за допомогою аватара або екранної індикації. Це відкриває нові можливості для спілкування між чуючими та нечуючими людьми, особливо у випадках, коли один з учасників не володіє мовою жестів. Такий підхід сприяє реальній інклюзії, а не просто технічному дублюванню інформації.

У процесі розробки система пройшла масштабне навчання на тисячах відеозаписів реальних жестів, виконаних людьми з різними стилями та швидкістю рухів [9]. Завдяки цьому алгоритми KinTrans стали здатні не лише ідентифікувати типові жести, але й розпізнавати індивідуальні варіації. Команда активно співпрацює з лінгвістами, перекладачами жестової мови та представниками спільноти нечуючих для постійного вдосконалення точності розпізнавання.

Окремо варто відзначити стратегічне партнерство KinTrans з урядовими структурами та великими корпораціями. Система вже впроваджена у декількох міжнародних аеропортах, що свідчить про її надійність і здатність працювати в умовах з підвищеними вимогами до точності та безперебійної роботи. Це також є показником того, що технологія вже вийшла за межі лабораторій і стала реальною частиною соціальної інфраструктури.

Платформа KinTrans продовжує активно розвиватися, впроваджуючи нові функції, такі як розпізнавання емоцій, адаптація до діалектів жестових мов та підтримка мобільних версій. Таким чином, вона залишається однією з найбільш технологічно просунутих і функціонально гнучких систем у світі.

1.2 Особливості передачі букв, слів та словосполучень за допомогою жестів

Жестова мова є унікальною формою комунікації, в якій замість звуків використовується комплекс візуальних сигналів: рухи рук, положення пальців, міміка, положення тіла. В українській жестовій мові (УЖМ) передачу букв, слів та словосполучень можна умовно поділити на кілька ключових способів: дактилологію, лексичні жести та контекстно-ситуативне групування. Кожен із цих елементів має власну функцію в структурі мови, забезпечуючи точну та ефективну передачу інформації.

Дактилологія – це алфавіт жестової мови, що складається з ручних конфігурацій, кожна з яких відповідає окремій літері. Вона використовується переважно для передачі власних імен, абревіатур, термінів, а також слів, для яких немає окремих жестів. В УЖМ дактильна абетка є одноручною, на відміну від деяких інших жестових мов, де застосовується обидві руки. Дактилювання вимагає від мовця високої точності, адже навіть незначна зміна положення пальців може спотворити зміст. Водночас цей спосіб є повільнішим порівняно з лексичними жестами і використовується переважно як допоміжний.

Основу жестової мови становлять лексичні жести – усталені рухи рук або мімічні вирази, що відповідають окремим словам або поняттям. УЖМ має тисячі таких жестів, кожен із яких має власне значення та форму виконання. Вони охоплюють побутову лексику, абстрактні поняття, емоції, дії, стани тощо. Часто лексичні жести мають іконічний характер – тобто їх форма частково відображає зміст слова (наприклад, жест «книга» імітує рух розгортання сторінок). Але є й умовні знаки, які не мають візуального зв’язку з об’єктом, тому потребують попереднього вивчення.

Словосполучення та речення у жестовій мові будуються шляхом послідовного виконання жестів відповідно до граматичних правил мови. Варто зазначити, що структура речень у жестовій мові відрізняється від усної:

вона часто не використовує службових частин мови, натомість покладається на порядок слів, інтонацію жесту, а також на міміку та рухи голови. Наприклад, у питальних реченнях часто змінюється міміка (підняті брови, нахил голови), а не лише форма жесту. Саме це створює додаткову складність для автоматичного розпізнавання – адже мова є не лише жестом, а комбінацією руху, виразу обличчя та контексту.

Особливість передачі інформації в УЖМ полягає також у контекстності. Один і той самий жест може означати різні речі в залежності від супровідної міміки, положення тіла або попередніх жестів. Наприклад, жест «знати» може в певному контексті означати не лише «мати знання», але й «пам’ятати» чи «розуміти», що стає зрозумілим тільки з урахуванням попередньої комунікативної ситуації. У зв’язку з цим, при перекладі важливо враховувати не тільки окремі жести, а й цілісну структуру повідомлення [10].

За даними деяких досліджень, носії жестової мови здатні продукувати в середньому до 5 жестів за секунду, що є порівняним із швидкістю усного мовлення. При цьому мова не обмежується лише рухами рук – важливу роль відіграє так звана «неголовна артикуляція» (рухи плечей, тіла, міміка), яка несе граматичну та емоційну інформацію. Таким чином, жестова мова є багаторівневою системою, у якій кожен компонент має значення.

Варто також зазначити, що УЖМ має регіональні варіації, тобто один і той самий жест може виглядати по-різному в залежності від регіону. Це накладає додаткові вимоги до систем автоматичного розпізнавання, які повинні враховувати варіативність виконання і навчатися на достатньо великій та різноманітній вибірці даних.

Жестова мова передає зміст не лише через «жести-слова», а й через так звані просторові конструкції. Це коли значення виражається за допомогою розміщення рук у певному просторі, повторюваності рухів або напрямку. Наприклад, жест, виконаний в один бік, може означати одного суб’єкта, а той самий жест в іншому напрямку – іншого.

Такий принцип часто застосовується для позначення діалогу між двома персонажами або дії, спрямованої на когось.

Таким чином, жестова мова – це складна, багатокомпонентна система, в якій поєднуються лексичні, граматичні та просторові елементи. Її передача потребує не лише точного руху, але й розуміння контексту, міміки та емоційного забарвлення. Саме ці особливості створюють як унікальність жестової мови, так і виклики для її автоматичного розпізнавання.

При створенні програмного забезпечення для перекладу жестів у текст особливу увагу необхідно приділяти дактильній складовій, інтерпретації складених конструкцій, обліку просторових маркерів і адаптації до реального контексту спілкування. Без врахування цих аспектів забезпечити точність та ефективність перекладу неможливо.

1.3 Аналіз літературних джерел щодо апробації результатів розроблення програмних застосунків для розпізнавання жестової мови у реальному часі

Аналіз наукових джерел з розпізнавання жестової мови дозволяє визначити сучасні підходи до апробації та перевірки програмного забезпечення, яке використовується для цього процесу в реальному часі. Розпізнавання жестової мови є важливим елементом для спрощення комунікації між людьми з вадами слуху та звичайними співрозмовниками. Розвиток таких технологій вимагає ретельної перевірки ефективності, що полягає у тестуванні алгоритмів на різних етапах, зокрема використання нейромереж та систем комп’ютерного зору.

Різноманітні рішення для розпізнавання жестової мови вже були апробовані в наукових дослідженнях. Наприклад, у статті [11] аналізуються системи, які використовують інерційні датчики, встановлені на верхніх кінцівках людини, а також технології комп’ютерного зору, що дозволяють передавати відеопотік у реальному часі на сервери для подальшої обробки та

перекладу жестів. Ці рішення показали ефективність у розпізнаванні дактильних жестів, однак, потребують подальшого вдосконалення для забезпечення більшої точності та інтеграції у повсякденне використання.

З іншого боку, на практиці широко використовуються носимі пристрої, такі як рукавиці, що забезпечують збір даних про рухи пальців і долонь для подальшої інтерпретації жестів. Наприклад, система SignAloud [4], що була розроблена в Університеті Вашингтона, демонструє використання сенсорних рукавиць для перекладу жестової мови на текст або звукові сигнали. Втім, головним недоліком таких пристроїв є складність їхнього використання у повсякденному житті, що потребує альтернативних рішень на основі лише відеозображень.

У дослідженнях [12] значну увагу приділено розробці методів та засобів опрацювання зображень для ідентифікації елементів жестової мови у реальному часі. Дослідження охоплюють два основні аспекти: дактиль, який відображає букви алфавіту, та повну жестову мову. Автори застосували різні підходи до розпізнавання, серед яких використовували нейронні мережі, метод головних компонент, псевдодвовимірні марківські моделі та інші техніки. Ефективність цих алгоритмів була оцінена на основі здатності системи правильно розпізнавати кінці пальців та форму кисті руки, що дозволило розробити прототип тренажера для навчання жестової мови. Апробація алгоритмів розпізнавання жестів показала їхню здатність працювати в реальному часі, забезпечуючи до 90% точності при класифікації рухів пальців. Однак, дослідники зазначають, що навіть невеликі зміщення або спотворення зображення можуть негативно вплинути на результативність, що вимагає подальшого вдосконалення системи. Розроблений тренажер включає функції синхронізації з відео та інформує користувача про правильність виконання жестів, що значно спрощує процес навчання жестовій мові.

У роботі [13] було представлено підхід до багатоканального розпізнавання жестів із використанням нейронної мережі YOLOv4, що дозволило досягти високої швидкодії та точності в обробці потоку з камери.

Важливою особливістю було поєднання локалізації кисті руки з подальшою класифікацією жесту на основі позицій пальців, що значно підвищило якість розпізнавання. Система була протестована в умовах природного освітлення і показала точність понад 93%, що є вагомим кроком у напрямку практичного впровадження.

Також у [14] розглядалася система розпізнавання динамічних жестів із використанням згорткових нейронних мереж та рекурентних шарів LSTM. Було запропоновано метод об’єднання відеокадрів у послідовності, що дозволило моделі враховувати зміну жесту у часі. Апробація проводилась на корпусі ASL-відеозаписів, і результати свідчать про високу точність навіть при наявності шумів у фоновому середовищі.

У межах проєкту [6], розробленого компанією Google, апробація застосунку Project Relate здійснюється за участі людей із мовленнєвими та слуховими порушеннями, що дозволяє системі адаптуватися до індивідуального стилю комунікації користувача. Результати бета-тестування демонструють позитивну динаміку у сприйнятті персоналізованих моделей машинного навчання для розпізнавання жестів, зокрема у поєднанні з візуальними елементами.

Ще одним прикладом апробованої технології є система KinTrans [8], яка пройшла тестування в умовах реального використання у публічних установах. Її ефективність була підтверджена під час впровадження в аеропортах, лікарнях і державних офісах. Згідно з результатами дослідження, система здатна точно розпізнавати понад 400 жестів, підтримує багатомовність і має високу адаптивність до особливостей руху користувача.

Також у нещодавній роботі [15] розглядалися аспекти адаптації мобільних застосунків для жестової мови до українських реалій. Автор наголошує на необхідності урахування культурного контексту та специфіки УЖМ під час створення програмного забезпечення, яке планується апробувати у навчальних закладах.

У контексті практичного навчання варто згадати працю [16], у якій досліджувалася ефективність використання тренажерів для жестової мови на основі гейміфікованих інтерфейсів. Результати показали, що включення інтерактивних елементів покращує якість засвоєння матеріалу та пришвидшує навички жестового спілкування у дітей.

Таким чином, результати аналізу літературних джерел [11–16] підтверджують актуальність розробки нових інструментів для розпізнавання жестової мови, зокрема на базі технологій комп’ютерного зору та глибокого навчання.

Апробація подібних рішень на практиці вказує на високу ефективність таких систем у реальних умовах використання, але також підкреслює потребу в адаптації до конкретної мовної системи, включно з українською жестовою мовою.

1.4 Постановка задачі

Таким чином, розробка програмного забезпечення для розпізнавання жестової мови в реальному часі є актуальним завданням у сфері комп’ютерного зору та машинного навчання. Такий застосунок дозволяє забезпечити комунікацію між людьми з порушеннями слуху та тими, хто не володіє жестовою мовою, що сприяє соціальній інтеграції та інклюзії. У межах цієї роботи ставиться завдання створення програмного застосунку, здатного з високою точністю розпізнавати жести УЖМ та трансформувати їх у текстові повідомлення.

Об’єктом роботи є розпізнавання жестової мови в реальному часі за допомогою програмного застосунку, який використовує методи комп’ютерного зору та машинного навчання.

Метою роботи є розроблення програмного застосунку, здатного розпізнавати жести української жестової мови та перетворювати їх у текстові

повідомлення для покращення комунікації між людьми з порушеннями слуху та тими, хто не володіє жестовою мовою.

Для досягнення поставленої мети необхідно вирішити такі завдання:

– здійснити аналіз сучасних рішень та технологій у сфері розпізнавання жестової мови;

– дослідити особливості передачі інформації жестовими засобами, зокрема дактильних елементів та просторових конструкцій української жестової мови;

– розробити архітектуру системи розпізнавання на основі моделей глибокого навчання;

– реалізувати прототип програмного застосунку з можливістю обробки відеопотоку з камери в реальному часі;

– провести тестування та оцінити ефективність системи за ключовими показниками точності, швидкості та зручності використання.

2 ОСОБЛИВОСТІ РОЗРОБЛЕННЯ ЛОГІЧНОЇ СКЛАДОВОЇ ПРОГРАМНОГО ЗАСТОСУНКУ ДЛЯ РОЗПІЗНАВАННЯ ЖЕСТОВОЇ МОВИ У РЕАЛЬНОМУ ЧАСІ

2.1 Детальний аналіз існуючих методів розпізнавання жестової мови

У сучасних умовах активного розвитку технологій штучного інтелекту та комп’ютерного зору особливу увагу привертають задачі, пов’язані з інтерпретацією невербальної комунікації, зокрема жестової мови. Розпізнавання жестів це багатокомпонентна задача, що охоплює аналіз візуальної інформації, виділення характерних ознак, інтерпретацію рухів та, у деяких випадках, контекстуальних залежностей. У цьому процесі поєднуються методи комп’ютерного зору, алгоритми машинного навчання, а також, за потреби, сенсорні технології.

Серед численних підходів, які були запропоновані у науковій літературі та реалізовані в прикладних розробках, можна виокремити кілька напрямів, що демонструють найбільшу ефективність на практиці. Їх застосування залежить від поставлених цілей, характеру жестів (статичні чи динамічні), обмежень апаратного забезпечення та необхідного рівня точності [17]. У цьому розділі буде здійснено узагальнений огляд п’яти найбільш поширених методів, що використовуються для розпізнавання жестової мови в реальному часі, з урахуванням їх технічних особливостей, переваг, недоліків та можливостей інтеграції в програмні застосунки.

2.1.1 Метод на основі ключових точок руки

Одним із найбільш популярних та технологічно доступних підходів до розпізнавання жестової мови є використання координат ключових точок кисті руки, які зчитуються за допомогою бібліотеки MediaPipe, з подальшою

передачею цих координат у класифікаційний алгоритм. Цей метод умовно поділяється на два етапи: перший це отримання та попередня обробка просторової інформації про положення руки, другий це класифікація отриманих даних відповідно до навченої моделі.

Бібліотека MediaPipe, розроблена Google, надає набір високоточного інструментарію для виявлення ключових точок людської руки на основі зображень або відеопотоку з камери. У стандартному режимі вона ідентифікує 21 ключову точку (landmark) для кожної руки, охоплюючи всі основні суглоби та сегменти пальців, включно з основою долоні. Координати кожної точки зчитуються у вигляді тривимірних значень (x, y, z), де x та y – це координати на площині зображення, а z – глибина, яка інтерпретується умовно на основі позиції руки щодо площини камери.

Після зчитування координат ці дані проходять обробку та перетворюються у вектор ознак, який може бути використаний як вхідний сигнал для класифікатора. У ролі класифікаційної моделі зазвичай застосовують алгоритми машинного навчання. Серед них найпростішим є метод *K*-ближчих сусідів (*K*-Nearest Neighbors), який порівнює нові вектори з уже відомими прикладами. Іншим поширеним варіантом є багатошаровий персептрон (Multilayer Perceptron) – тип штучної нейронної мережі, здатної навчатися складніших залежностей. Метод опорних векторів (Support Vector Machine) також часто використовується, особливо коли кількість ознак є великою. Вибір конкретного алгоритму залежить від потреб завдання та якості вхідних даних.

Цей підхід передбачає наявність навчального датасету, який містить зразки жестів у вигляді векторів координат із мітками класу (наприклад, «A», «B», «C» для жестового алфавіту або конкретних слів). Навчання моделі здійснюється на основі цих зразків, після чого вона здатна розпізнавати нові жести у режимі реального часу.

Основною перевагою цього підходу є його висока швидкодія та ефективність. Оскільки класифікація відбувається на основі координат, а не

повноцінного зображення, об’єм даних, що підлягає обробці, є відносно невеликим. Це дозволяє реалізовувати застосунок навіть на малопотужних пристроях, таких як ноутбуки середнього рівня, планшети або смартфони.

Крім того, MediaPipe є кросплатформеним інструментом, що підтримується на різних операційних системах та інтегрується з популярними мовами програмування, зокрема Python та C++. Його відкритий код та велика спільнота розробників роблять цей інструмент привабливим для використання у наукових, навчальних та комерційних проєктах.

Ще однією позитивною рисою є гнучкість у налаштуванні. У залежності від складності жестів, розробник може обирати відповідну модель класифікації – від найпростіших (наприклад, для алфавітних символів) до складніших (наприклад, для розпізнавання окремих слів або команд).

Разом з тим, цей метод має й низку обмежень, які варто враховувати при його впровадженні.

По-перше, MediaPipe добре працює лише за наявності чіткого освітлення та контрастного фону. У складних умовах, наприклад, при слабкому освітленні або якщо рука частково перекрита – точність виявлення ключових точок значно знижується.

По-друге, метод має обмеження щодо динамічних жестів. Якщо жест включає складну послідовність рухів, що відбуваються у часі (наприклад, проведення рукою вліво чи вгору), простого векторного опису координат руки на одному кадрі буде недостатньо. У таких випадках краще працюють інші методи, що враховують часову динаміку.

Альтернативним підходом до розпізнавання жестів, який також базується на аналізі ключових точок руки, є метод частинно-орієнтованого представлення жесту, запропонований у дослідженні [18], присвяченому розпізнаванню жестів із використанням сенсора Kinect (рис. 2.1). У цьому випадку просторову структуру руки визначають не лише через координати суглобів, а передусім через глобальні особливості форми, зосереджуючись на виявленні та аналізі пальців як окремих функціональних частин.

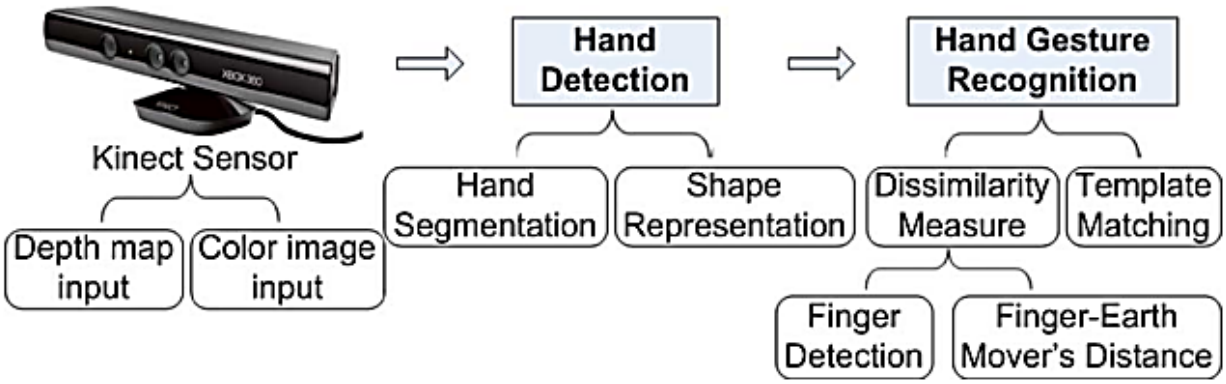

Рисунок 2.1 – Принцип роботи системи розпізнавання жестів рук на основі частин

Кожен палець у цьому методі розглядається як кластер (рис. 2.2), а вся рука – як сукупність таких кластерів, що дозволяє уникнути помилок, зумовлених локальними викривленнями або артефактами сегментації.

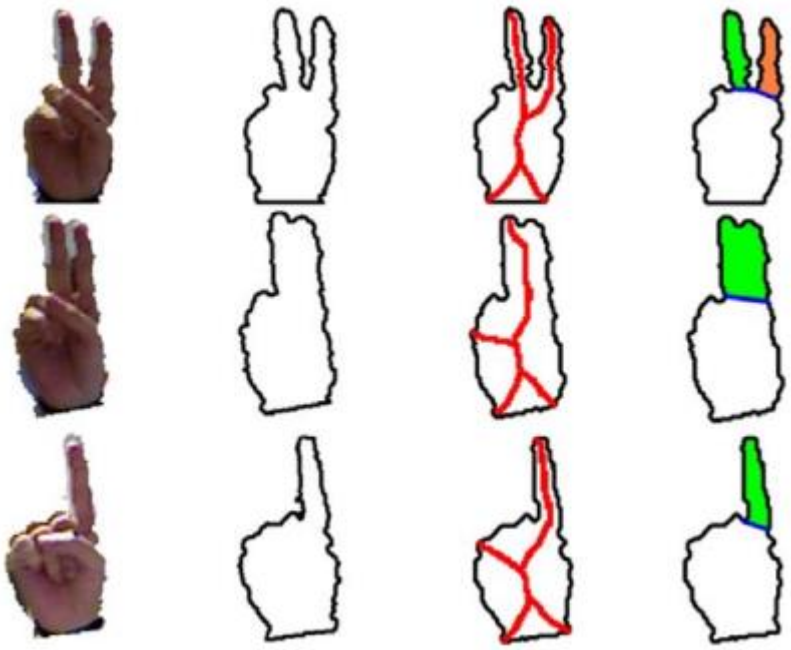

Рисунок 2.2 – Розпізнавання жестів рук за допомогою сенсора Kinect

Для обчислення схожості між жестами запропоновано спеціалізовану метрику під назвою Finger-Earth Mover’s Distance (FEMD), яка адаптує класичний алгоритм Earth Mover’s Distance до задачі розпізнавання жестів.

FEMD дозволяє точно визначати відстань між двома представленнями руки, фокусуючись на конфігурації пальців, а також враховує відсутність очікуваних елементів (наприклад, якщо частина пальців не ідентифікована) шляхом введення штрафної компоненти. Відтак система виявляє високу стійкість до варіацій масштабу, орієнтації руки, а також до артефактів, що виникають внаслідок обмежень глибини сенсора.

Залежно від вимог до точності та швидкодії, метод передбачає використання двох різних способів виявлення пальців: пороговий аналіз висот кривої контуру руки, який дозволяє досягти роботи в реальному часі, та більш точний, хоча й обчислювально складніший, підхід на основі майже опуклого розкладу форми. У ході тестування система продемонструвала середню точність до 93,9 % на спеціально зібраному наборі даних жестів, що підтверджує її придатність для використання в задачах людино-комп’ютерної взаємодії, зокрема в інтерактивних інтерфейсах та освітніх застосунках.

Такий підхід, як і метод з використанням MediaPipe, підтверджує ефективність використання ключових точок для розпізнавання жестів, проте відрізняється способом математичного представлення та оцінювання просторових характеристик руки, що дає змогу обирати оптимальний інструментарій залежно від поставленої задачі та ресурсних обмежень.

2.1.2 Розпізнавання на основі згорткових нейронних мереж

Згорткові нейронні мережі (Convolutional Neural Networks, CNN) є одним із найпотужніших інструментів у сфері комп’ютерного зору. Саме цей тип нейронних мереж забезпечив прорив у задачах класифікації зображень, розпізнавання об’єктів та інтерпретації візуальних патернів. У контексті розпізнавання жестової мови CNN застосовуються для аналізу кадрів, отриманих із камери, з метою класифікації зображень руки, що формує конкретний жест.

На відміну від методу, що ґрунтується на координатах ключових точок, як у випадку з MediaPipe, згорткові нейронні мережі (CNN) не вимагають попереднього ручного виділення ознак. Вони здатні самостійно навчатися розпізнавати візуальні закономірності на рівні пікселів, зокрема форму, контури, текстури та інші характеристики, характерні для кожного окремого жесту. Завдяки цьому модель може гнучко адаптуватися до різноманітних умов і потенційно розпізнавати ширший спектр жестів, у тому числі й ті, що не мають чітко визначеної пози.

Загальна архітектура CNN складається з кількох послідовних компонентів. Спочатку згорткові шари зчитують локальні фрагменти зображення за допомогою фільтрів або ядер [19]. Далі використовуються шари активації, зазвичай функція ReLU, які додають у модель нелінійність. Наступним етапом є шари субдискретизації (pooling), що зменшують розмірність даних і підвищують стійкість до шуму. Завершує структуру мережі блок повнозв’язних шарів, які виконують остаточну класифікацію зображення.

Для навчання такої мережі необхідно мати датасет з великою кількістю зображень рук у різних позах. Зображення зазвичай проходять попередню обробку – обрізання, нормалізацію, масштабування – перед подачею в мережу. Модель навчається шляхом зменшення функції втрат (loss function), порівнюючи передбачення з реальними мітками жестів.

Однією з ключових переваг CNN у задачах розпізнавання жестів є висока точність класифікації при належній кількості тренувальних даних. Завдяки здатності автоматично виявляти й узагальнювати складні патерни, CNN можуть успішно працювати навіть у випадках, коли жести мають мінімальні візуальні відмінності.

Ще однією перевагою є гнучкість у роботі з різними форматами вхідних даних: CNN однаково ефективно можуть працювати як із кольоровими зображеннями, так і з чорно-білими, як зі статичними кадрами, так і з відеопотоком.

Також CNN добре масштабується – існує безліч попередньо натренованих моделей (наприклад, MobileNet, ResNet, EfficientNet), які можна адаптувати до розпізнавання жестів за допомогою методів трансферного навчання (Transfer Learning). Це дозволяє скоротити час навчання та знизити вимоги до обчислювальних ресурсів.

По-друге, нейронні мережі зазвичай потребують потужного апаратного забезпечення, особливо на етапі навчання. Хоча inference (розпізнавання на вже натренованій моделі) може відбуватись на менш потужних пристроях, для початкового навчання часто необхідне використання GPU.

Ще одне обмеження – відсутність прозорості у процесі прийняття рішень. На відміну від класичних алгоритмів, CNN працюють як «чорна скринька», і складно пояснити, чому певне зображення було класифіковано як конкретний жест.

У межах сучасних досліджень значну увагу привертає підхід [20], що передбачає використання тривимірних згорткових нейронних мереж, які аналізують не лише просторові, а й часові характеристики жесту. На відміну від традиційних CNN, що працюють із окремими зображеннями, тривимірне CNN (рис. 2.3) обробляють відеофрагменти як тривимірні об’єми даних, що дозволяє моделі враховувати динаміку рухів руки протягом часу.

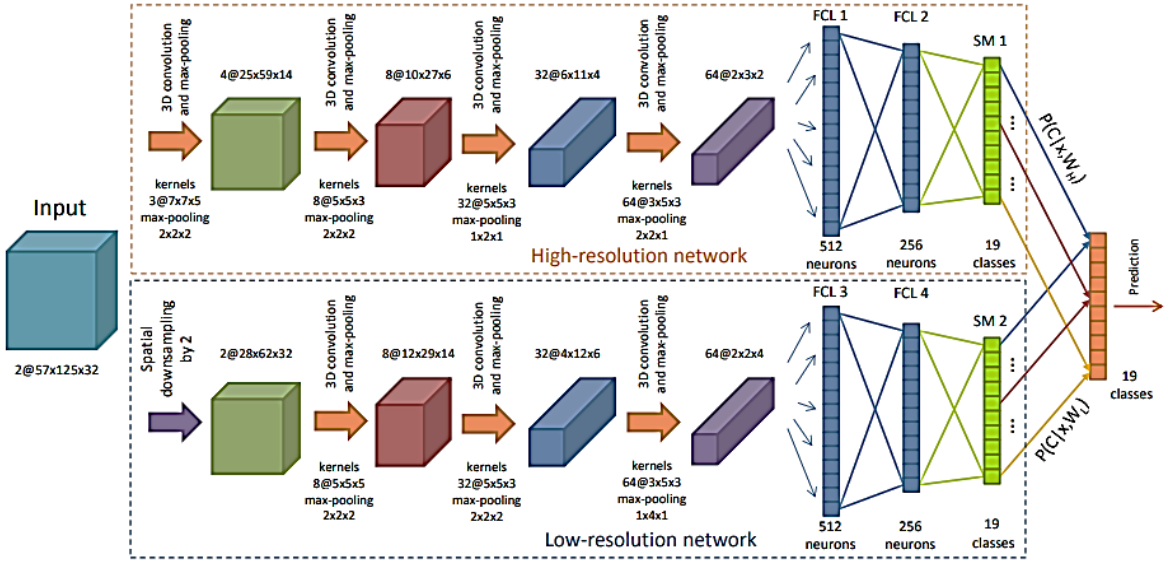

Рисунок 2.3 – Схема класифікатора CNN

Такий підхід особливо актуальний у задачах розпізнавання динамічних жестів, які складаються з послідовності положень руки, а не одного статичного кадру.

В одній з найбільш відомих реалізацій цього підходу, представленій компанією NVIDIA, було розроблено модель, яка використовує глибинну та інтенсивну (градієнтну) інформацію з відеопотоку для формування нормалізованого просторово-часового об’єму. Модель складається з двох незалежних підмереж: високої та низької роздільної здатності. Кожна з них складається з кількох тривимірних-згорткових і субдискретизувальних шарів, двох повнозв’язних шарів і шару softmax для визначення класу жесту. Остаточне рішення формується шляхом елементного множення ймовірностей, отриманих з обох підмереж, що підвищує надійність класифікації.

Особливу увагу в моделі приділено процедурі просторово-часового аугментування даних, яка реалізована як у формі попередньої обробки (наприклад, віддзеркалення, зміна порядку кадрів, масштабування), так і під час навчання (випадкові просторові зміщення, еластичні деформації, drop-out окремих пікселів). Такі техніки дозволяють ефективно запобігти перенавчанню та підвищити здатність моделі до узагальнення, навіть при обмеженому обсязі навчальних даних.

За результатами тестування на складному датасеті VIVA, що містить динамічні жести, виконані різними користувачами в умовах змінного освітлення, запропонована модель досягла точності 77,5 %, що суттєво перевищило результати традиційних методів, зокрема базової моделі, заснованої на гістограмах градієнтів. Такий результат свідчить про високу адаптивність тривимірного CNN до реальних умов і потенціал їх застосування в системах безконтактного керування, наприклад, в автомобільних інтерфейсах або розумних пристроях.

Таким чином, метод розпізнавання жестів на основі тривимірних згорткових нейронних мереж є перспективним напрямом, що дозволяє ефективно враховувати як просторову, так і часову інформацію про рухи руки.

Його реалізація потребує значних обчислювальних ресурсів на етапі навчання, проте забезпечує високу точність і гнучкість при роботі з відеопотоками в режимі реального часу.

2.1.3 Метод із використанням рекурентних нейронних мереж

Коли мова заходить про розпізнавання жестів, що складаються з послідовних рухів у часі (наприклад, провести рукою вперед або зробити характерний розмах), використання лише згорткових мереж (CNN) або координат ключових точок стає недостатнім. Адже ці підходи зазвичай фіксують лише окремий кадр – «моментальну фотографію» жесту. Для аналізу динамічних жестів, тобто тих, що мають часову структуру, логічним вибором є рекурентні нейронні мережі (RNN) та їх удосконалені варіації – LSTM (Long Short-Term Memory) та GRU (Gated Recurrent Unit).

Рекурентні мережі створені спеціально для роботи з послідовностями даних, які мають часову залежність. Це може бути текст, аудіо, відео – будь-яка інформація, яка змінюється у часі. У контексті розпізнавання жестової мови такими послідовностями є серії кадрів з координатами ключових точок руки або навіть послідовності зображень.

Основна ідея полягає в тому, що RNN «запам’ятовують» попередні стани, тобто вміють враховувати попередній контекст при обробці кожного нового елемента послідовності [21]. Наприклад, якщо людина показує жест, що складається з трьох фаз, то мережа буде аналізувати не кожен кадр окремо, а взаємозв’язок між ними – тобто, повну траєкторію руху.

Найголовнішою перевагою цього методу є його здатність враховувати час – тобто, аналізувати не лише «що», але й «коли і як». Це відкриває можливість розпізнавати складні динамічні жести, які важко або й неможливо описати одним кадром. Саме це робить рекурентні моделі незамінними у більшості сучасних систем розпізнавання повноцінної жестової мови,

особливо у задачах перекладу цілих речень або фраз, а не лише одиничних літер чи слів.

Як і всі глибокі нейронні архітектури, рекурентні моделі мають високі вимоги до ресурсів – і щодо обчислювальної потужності, і щодо кількості даних. Для того щоб навчити модель якісно відтворювати часову динаміку жесту, необхідно мати велику кількість відео або послідовностей координат, а також добре анотована колекція даних.

Також ці моделі мають високу складність налаштування: вибір довжини послідовності, кількості шарів, розмірності прихованого стану – все це вимагає експериментів і досвіду. Крім того, RNN гірше масштабуються, ніж CNN, особливо при роботі з великими датасетами.

Одним із перспективних напрямів розвитку систем розпізнавання жестів є інтеграція просторової та часової обробки даних у межах єдиної глибокої нейронної архітектури. У цьому контексті доцільним виявляється поєднання згорткових нейронних мереж (CNN), які ефективно вилучають ознаки з окремих кадрів, із рекурентними мережами типу LSTM, здатними аналізувати послідовності таких ознак у часі.

Такий підхід дозволяє моделі враховувати як форму руки в кожен момент часу, так і динаміку її руху, що є критично важливим для розпізнавання складних динамічних жестів.

У дослідженні [22], проведеному на основі відеозаписів із камери Kinect, було реалізовано двоетапну модель (рис. 2.4), у якій спершу за допомогою двох окремих CNN оброблялися яскравісні та глибинні дані кадрів, а потім їхні ознаки з’єднувалися та подавалися на вхід до двошарової LSTM.

Такий підхід дозволив моделі не лише точніше розпізнавати жести, але й залишатися стійкою до змін освітлення, положення руки або фону. Задля підвищення ефективності навчання з одного відео вибиралися лише ключові фрагменти тривалістю 8 кадрів, які охоплювали найхарактерніші фази жесту.

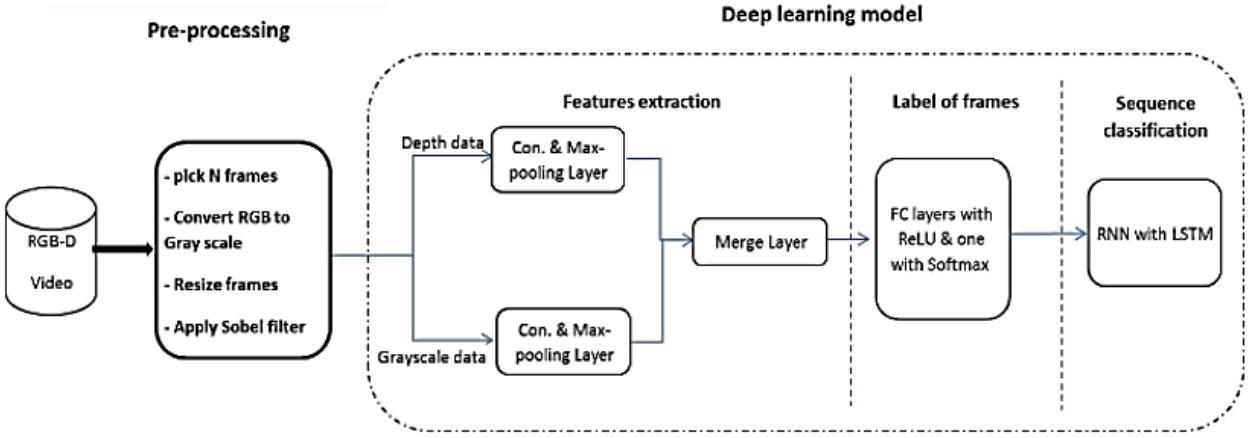

Рисунок 2.4 – Схема двоетапної моделі

Після проходження CNN, просторові ознаки кожного кадру перетворювалися у послідовність векторів, яку LSTM аналізувала як часову структуру. У підсумку, після проходження двох рекурентних шарів, фінальний шар softmax здійснював класифікацію жесту. Отримані результати засвідчили високу точність моделі: найкраще значення 93 % досягнуто при використанні глибини й інтенсивності одночасно. Це суттєво перевищило результати моделей, які покладалися виключно на CNN або традиційні алгоритми на кшталт HOG.

2.1.4 Аналіз положення тіла за допомогою системи OpenPose

Одним із найефективніших інструментів для відстеження рухів людини на відео є система OpenPose, розроблена дослідницькою лабораторією Університету Карнегі Меллона (Carnegie Mellon University). Це програмне забезпечення стало значним проривом у галузі комп’ютерного зору, оскільки вперше забезпечило можливість точного відстеження положення всього тіла, обличчя, рук і ніг людини в реальному часі за допомогою звичайної камери RGB. Система OpenPose функціонує на основі глибокої нейронної мережі, яка здатна виявляти ключові точки людського тіла на зображеннях або відеопотоках (рис. 2.5) [23].

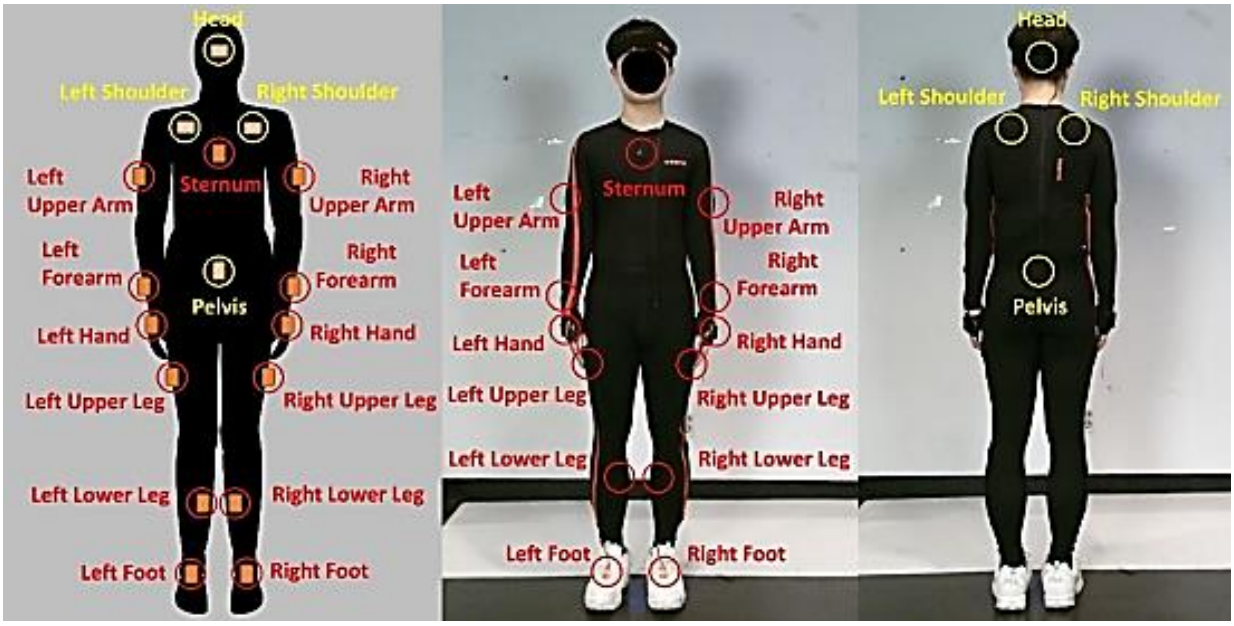

Рисунок 2.5 – Визначення координат основних точок

Для кожної особи, що потрапляє в кадр, OpenPose визначає координати основних точок:

– 18 точок тіла, зокрема голови, шиї, плечей, ліктів, зап’ясть, стегон, колін та щиколоток;

– 21 точки на кожній руці, включаючи фаланги пальців;

– близько 70 точок обличчя;

– точки ступнів, якщо увімкнено повний режим.

Після виявлення ключових точок OpenPose формує умовний «скелет» людини, що точно відображає пропорції та просторове положення частин тіла відносно одна одної (рис. 2.6) [23].

Завдяки цьому система здатна з високою точністю інтерпретувати пози, рухи та жести. У контексті розпізнавання жестової мови особливий інтерес становлять координати рук і пальців, адже саме вони несуть основне смислове навантаження.

OpenPose є однією з небагатьох систем, що забезпечують високу точність визначення положення пальців навіть за відсутності даних про глибину (тобто без використання тривимірних камер або глибинних сенсорів).

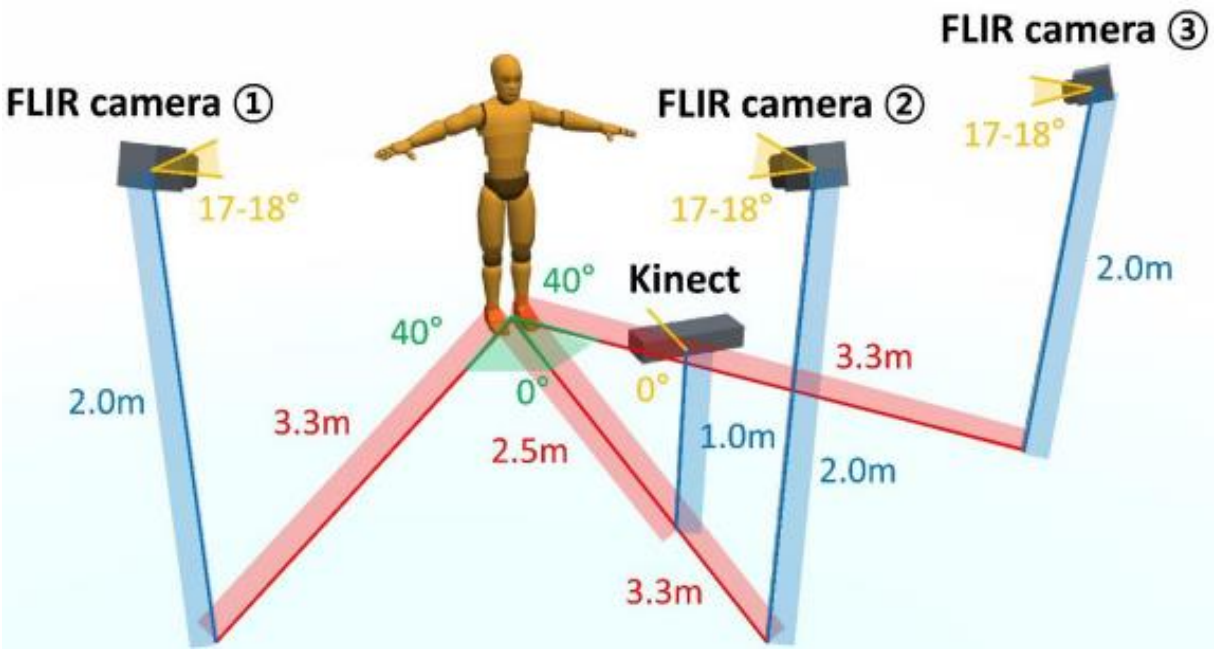

Рисунок 2.6 – Експериментальна установка

Серед ключових переваг цієї системи варто відзначити її високу точність виявлення ключових точок навіть у складних умовах, таких як перекриття частин тіла, складні пози або наявність сторонніх об’єктів у кадрі. Крім того, система здатна аналізувати рухи кількох осіб одночасно, що є корисним у контексті навчання або групової взаємодії. Завдяки підтримці повного тіла (від голови до ступнів) OpenPose дає змогу враховувати положення не лише рук, а й плечей, тулуба та голови, що забезпечує додатковий контекст для розпізнавання жестів. Іншою перевагою є гнучкість у використанні: система доступна як окремий застосунок, а також як бібліотека, що підтримує API для Python і C++.

Втім, попри свою ефективність, OpenPose має і певні недоліки. Серед них – високі вимоги до апаратного забезпечення: для обробки відео в реальному часі необхідна наявність потужного графічного процесора, інакше частота кадрів може знижуватись від 1 до 2 FPS. Крім того, велика вага моделі ускладнює її використання на мобільних пристроях або в браузерах без попередньої оптимізації. Якість роботи системи значною мірою залежить від умов зйомки: низьке освітлення або розмиті кадри можуть істотно знизити точність визначення координат.

Варто також зазначити, що OpenPose самостійно не здійснює класифікацію жестів – вона лише надає координати точок. Логіку розпізнавання (тобто визначення, який саме жест було показано) користувач має реалізовувати окремо, наприклад, із використанням моделей на основі LSTM або SVM.

2.2 Формування методики розпізнавання жестової мови у реальному часі

Методика розпізнавання жестової мови у реальному часі представляє собою структурований підхід до створення та використання системи, здатної інтерпретувати жестові сигнали та перетворювати їх у текстовий формат. Запропонована методика включає декілька взаємопов’язаних етапів, які забезпечують повний цикл розробки та функціонування системи розпізнавання жестів.

На першому етапі (рис. 2.7) відбувається підготовка системи до розпізнавання. Користувач активує відеопотік з камери, який є джерелом візуальних даних для подальшої обробки [24]. Система налаштовується на розпізнавання та відстеження рук у відеопотоці з використанням спеціалізованих методів комп’ютерного зору.

Особлива увага приділяється правильному налаштуванню параметрів відстеження, таких як мінімальний поріг впевненості детекції та відстеження, що забезпечує стабільність роботи системи в різних умовах освітлення та позиціонування.

Наступним ключовим етапом методики є формування навчальної бази жестів (рис. 2.8). Користувачу надається можливість вибору з визначеного переліку жестів, які він має намір використовувати.

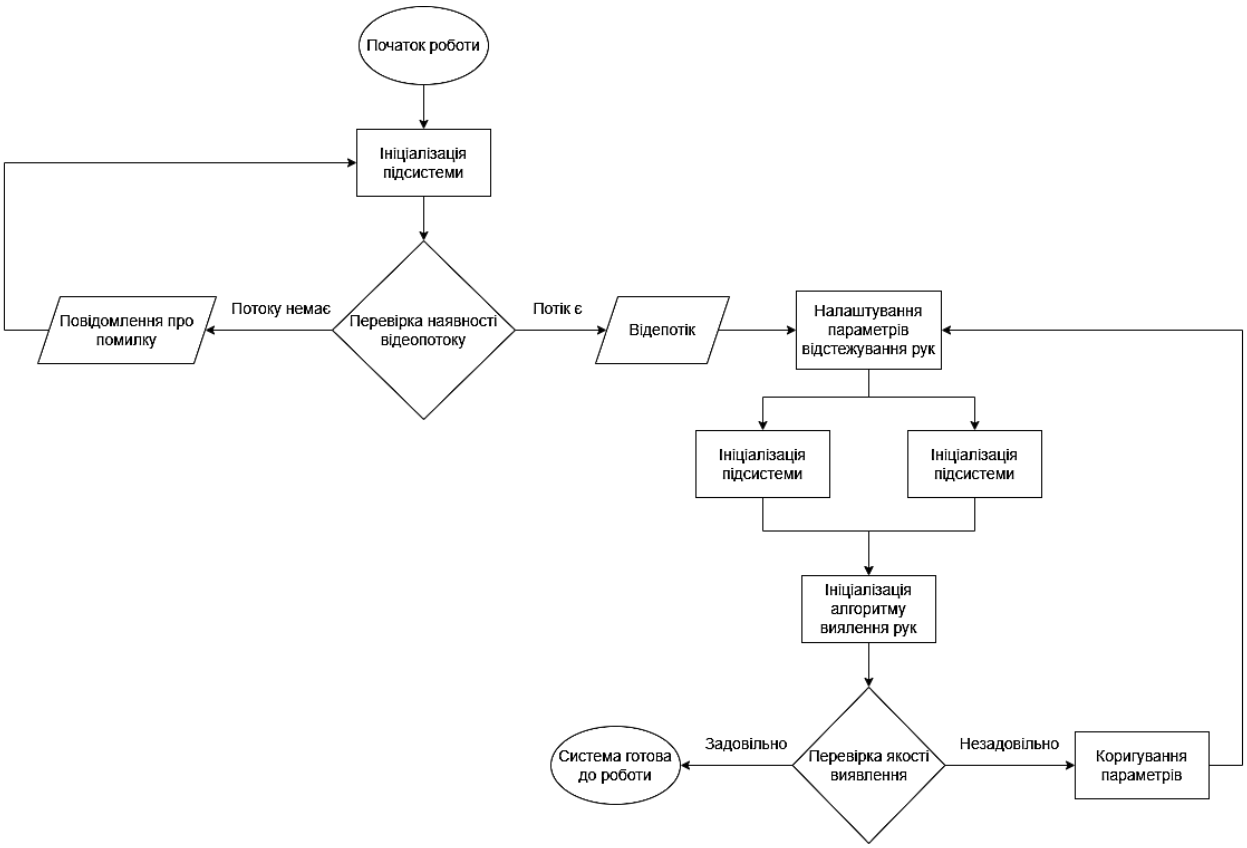

Рисунок 2.7 – Блок-схема діяльності системи

Для обраної системи передбачено 108 базових жестів української жестової мови, серед яких такі, як: «Привіт», «До побачення», «Дякую», «Будь ласка», «Вибачте», «Так», «Ні», «Допомога», «Туалет», «Вода», «Їсти», «Пити», «Стоп», «Більше», «Все», «Мене звати...», «Ім’я», «Друг», «Сім’я», «Люблю тебе».

Цей перелік охоплює базові комунікативні потреби та забезпечує основу для повсякденного спілкування. Для кожного жесту з переліку система проводить процес збору навчальних даних.

Користувач демонструє жест перед камерою, а система фіксує послідовність кадрів, на яких відображено різні фази виконання жесту.

Для забезпечення належної варіативності та репрезентативності навчальних даних передбачається збір значної кількості зразків кожного жесту – близько сотні послідовностей на кожен жест. Кожна послідовність складається з тридцяти кадрів, що дозволяє повністю охопити динаміку виконання жесту від початку до завершення (рис. 2.8).

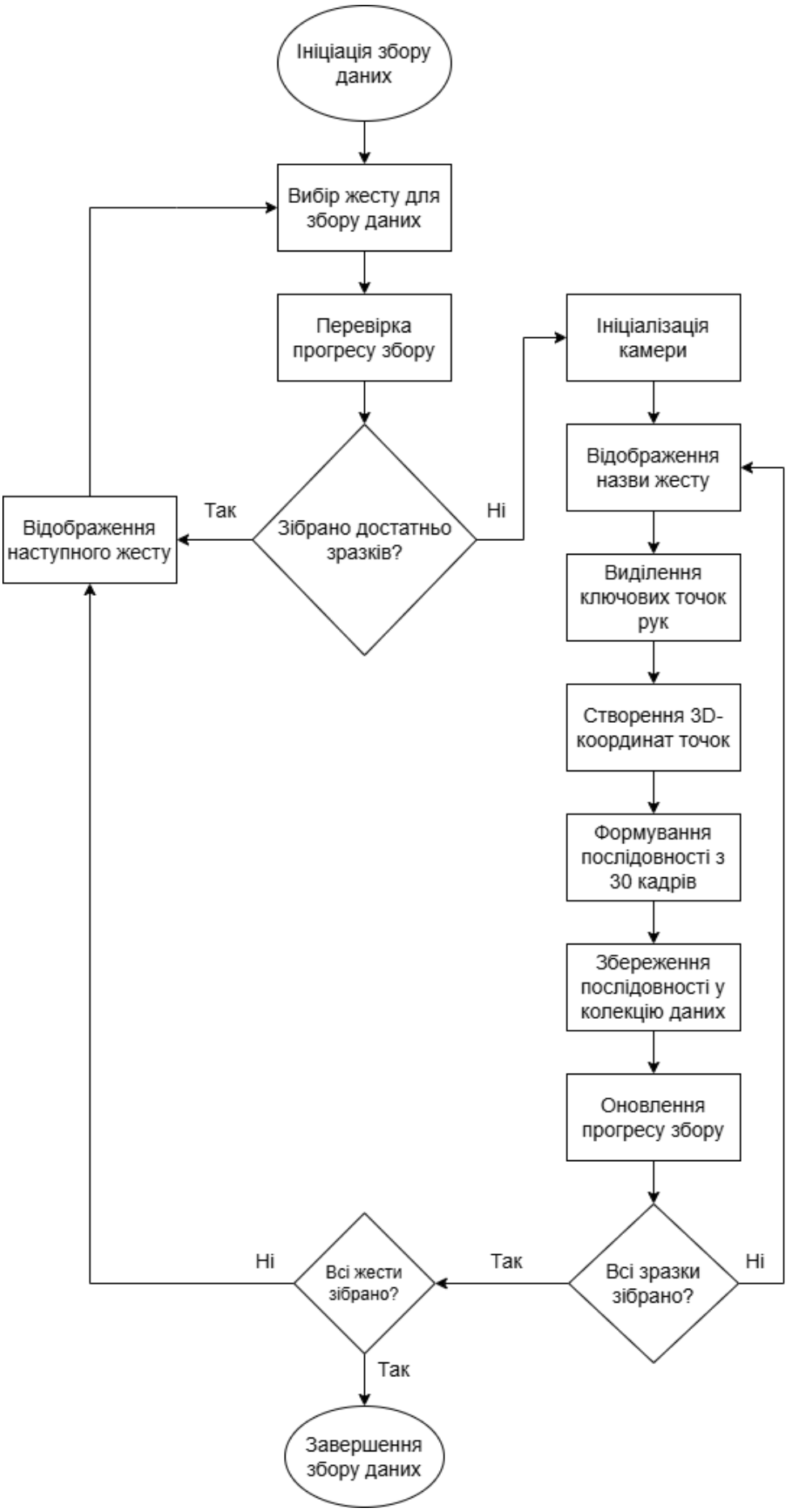

Рисунок 2.8 – Блок-схема формування навчальної колекції даних

У процесі збору даних система виділяє ключові точки руки (долоні та пальців) на кадрах, фіксуючи їх координати у тривимірному просторі.

Для кожної зафіксованої точки зберігаються значення координат *x, y* та *z*, що дозволяє системі враховувати не лише планарну проєкцію руки, але й глибину розташування її складових частин. Система обробляє обидві руки одночасно, що є важливим для жестів, які вимагають використання обох рук.

Зібрані дані зберігаються у спеціалізованій структурі файлової системи, де для кожного жесту створюється окрема директорія, а зразки послідовностей зберігаються у вигляді числових масивів. Інтерфейс системи забезпечує візуальний контроль за процесом збору даних, демонструючи користувачу поточний прогрес для кожного жесту та загальний статус наповнення бази даних.

Після завершення етапу збору даних відбувається навчання моделі машинного навчання (рис. 2.9). Для цього використовується рекурентна нейронна мережа з архітектурою LSTM (Long Short-Term Memory), яка добре підходить для обробки послідовностей, зокрема часових рядів координат точок руки.

Модель містить два шари LSTM по 64 нейрони, між якими розміщені шари Dropout для регуляризації та зниження ризику перенавчання. Після них іде повнозв’язний шар, який агрегує вихідну інформацію, а завершальний класифікаційний шар із функцією активації Softmax видає ймовірності належності до відповідного класу жестів.

Навчання моделі здійснюється шляхом оптимізації її параметрів з використанням категоріальної крос-ентропії як функції втрат.

Після навчання модель переходить у режим розпізнавання жестів у реальному часі (рис. 2.10). У цьому режимі система безперервно аналізує відеопотік з камери, виділяє на кожному кадрі положення рук, обчислює координати ключових точок та формує послідовність векторів ознак. Ця послідовність подається на вхід навченої моделі, яка генерує прогноз щодо приналежності послідовності до одного з відомих жестів.

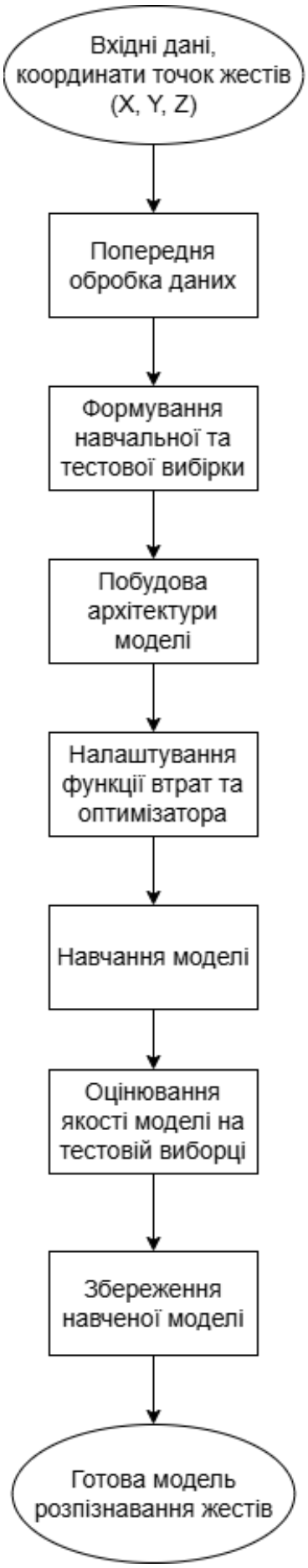

Рисунок 2.9 – Блок-схема процесу навчання моделі розпізнавання жестів

Рисунок 2.10 – Блок-схема процесу розпізнавання в реальному часі

Важливим аспектом методики є оцінка моделлю впевненості у власних прогнозах. Для кожного розпізнаного жесту система обчислює рівень впевненості як ймовірність відповідності поточної послідовності конкретному жесту.

Якщо цей рівень перевищує визначений поріг (наприклад, 0,3), система відображає назву розпізнаного жесту користувачеві. В іншому випадку система утримується від виведення результату, що запобігає появі помилкових інтерпретацій.

Інтерфейс системи у режимі розпізнавання забезпечує візуальний зворотний зв’язок, відображаючи не лише остаточний результат, але й список жестів з найвищими рівнями впевненості та відповідними відсотковими показниками [25]. Це дозволяє користувачеві оцінити якість розпізнавання та за потреби скоригувати виконання жесту для підвищення точності інтерпретації.

Таким чином, запропонована методика розпізнавання жестової мови у реальному часі являє собою комплексний підхід, що охоплює всі етапи від підготовки системи та збору даних до навчання моделі та її використання для розпізнавання жестів. Методика забезпечує високу адаптивність системи до індивідуальних особливостей користувача та створює основу для ефективної комунікації за допомогою жестової мови у різноманітних контекстах застосування.

3 ЕТАПИ РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАСТОСУНКУ ДЛЯ РОЗПІЗНАВАННЯ ЖЕСТОВОЇ МОВИ У РЕАЛЬНОМУ ЧАСІ

3.1 Вибір інструментальних засобів реалізації програмного застосунку для розпізнавання жестової мови у реальному часі

Розроблення застосунку для розпізнавання жестової мови у реальному часі на мові програмування Python передбачає реалізацію низки складних функцій: збір відеоданих із камери, оброблення та виділення ключових точок рук, побудову і навчання моделей машинного навчання для класифікації жестів, а також організацію зручного інтерфейсу для взаємодії з користувачем. Для забезпечення максимальної ефективності та продуктивності програмного комплексу важливо обрати належні інструментальні засоби, здатні працювати у режимі реального часу та забезпечувати високу якість розпізнавання [26].

Для отримання вхідних даних про просторові координати пальців і положення рук використовується система комп’ютерного зору, базована на алгоритмах трекінгу. Для цієї задачі оптимальним вибором є бібліотека MediaPipe від Google – відкритий, багатоплатформений інструментарій для розпізнавання широкого спектра візуальних патернів.

MediaPipe реалізує спеціалізовані нейронні мережі для виявлення детальних тривимірних координат 21 ключової точки на руці (рис. 3.1), що дає змогу точно уловлювати положення та жестикуляцію рук. Особливістю MediaPipe є висока швидкодія, низькі вимоги до апаратних ресурсів, простота інтеграції з бібліотеками для відображення, такі як OpenCV, та забезпечення реального часу без затримок. Цей інструментарій підтримує відстеження двох рук одночасно, що є критично важливим для повного охоплення мови жестів, де часто використовуються обидві кінцівки.

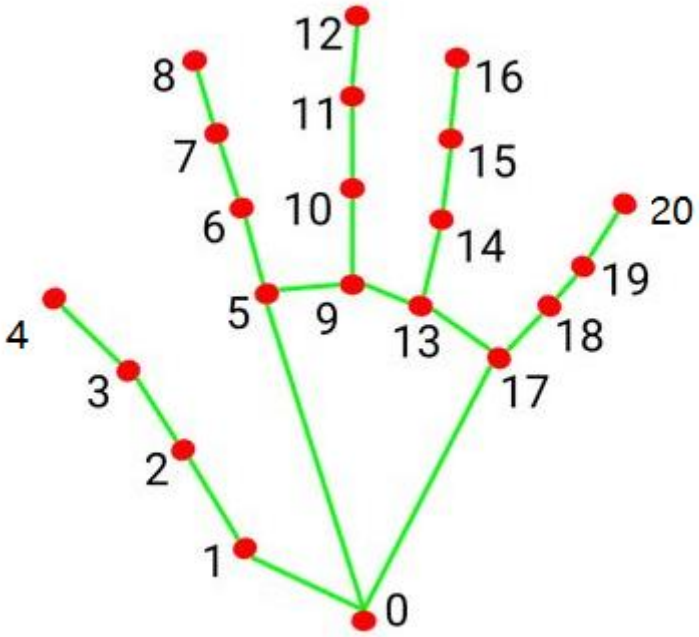

Рисунок 3.1 – Розміщення 21 ключової точки на руці

Можливості MediaPipe дозволяють обробляти відеопотік із камери користувача у реальному часі, передавати координати точок для аналізу, що робить її центральним компонентом.

Для завдань розпізнавання жестів потрібна ефективна модель машинного навчання, яка здатна врахувати часову послідовність рухів рук. Для цього доцільно застосувати рекурентні нейронні мережі, зокрема варіант Long Short-Term Memory, що ефективно захоплюють контекстні залежності у послідовностях.

Як платформа для створення, навчання та оптимізації таких моделей використовується TensorFlow – широко розповсюджений інструментарій з відкритим кодом, що підтримує налаштування глибоких нейронних мереж і має оптимізацію під різні апаратні пристрої.

Бібліотека Keras, що входить у TensorFlow, забезпечує високорівневий API для швидкої та зручної побудови моделей нейронних мереж, що особливо важливо для розробників, які хочуть гнучко налаштовувати архітектуру та експериментувати з різними параметрами.

У поєднанні з якісною колекцією даних, зібраним за допомогою MediaPipe, та обробкою відеопотоку моделі на основі LSTM демонструють високу точність розпізнавання динамічних жестів у реальному часі.

Для отримання відеопотоку з камери, відтворення результатів роботи та накладання додаткових графічних елементів (наприклад, відображення контурів рук, координат, інформації про розпізнаний жест) використовується бібліотека OpenCV. Вона надає можливості для обробки зображень та відео в реальному часі, працюючи з різними форматами камер і підтримуючи широкий спектр графічних операцій [27].

Для нанесення якісного тексту, зокрема з підтримкою української мови та спеціальних шрифтів, застосовується бібліотека PIL (Pillow). Вона дозволяє точно позиціонувати текст на відеокадрі, налаштовувати розмір та стиль шрифту, кольори, що підвищує зручність користування та інформативність інтерфейсу.

Для створення користувацького інтерфейсу автомати розробники застосовують стандартну бібліотеку Python – Tkinter. Вона забезпечує прості у використанні елементи керування, такі як кнопки, поля введення, вікна повідомлень, що дозволяє швидко створити інтерфейс з мінімальними затратами часу.

Tkinter сумісний з OpenCV, і його можна застосовувати для представлення додаткової інформації, вибору жестів, запуску режимів запису чи розпізнавання, а також контролю роботи системи користувачем.

Обрання MediaPipe, TensorFlow/Keras, OpenCV, Pillow та Tkinter як інструментальних засобів реалізації є оптимальним поєднанням сучасних технологій, що дозволяє створити ефективний, точний і чуйний програмний застосунок для розпізнавання жестової мови у реальному часі. Висока швидкодія, точність розпізнавання та зручний і зрозумілий інтерфейс надають змогу оптимально вирішувати задачі покращення комунікації для користувачів з порушеннями слуху.

3.2 Етапи програмної реалізації розпізнавання жестової мови у реальному часі

На першому етапі реалізації системи здійснюється ініціалізація програмного середовища та підключення пристрою захоплення відео. Зокрема, для отримання зображення в режимі реального часу використовується бібліотека OpenCV, яка забезпечує доступ до вебкамери комп'ютера. У реалізованій версії застосунку параметри роздільної здатності та частоти кадрів встановлюються за замовчуванням, однак архітектура коду дозволяє їх гнучке налаштування за потреби.

Після ініціалізації пристрою зображення з кожного кадру передається на обробку модулю MediaPipe Hands, який виконує виявлення кистей рук. Для забезпечення точності роботи алгоритму встановлено значення порогу в достовірності виявлення (*min_detection_confidence* = 0,7) та відстеження (*min_tracking_confidence* = 0,7). Це дозволяє стабільно фіксувати ключові точки руки на зображенні, що є критично важливим для подальшого розпізнавання жестів.

При захоплення зображення з камери кадри підлягають попередньому обробленні (рис. 3.2), що забезпечує коректну роботу алгоритмів розпізнавання жестів. Одним із перших кроків є перетворення зображення з колірного простору BGR, який використовується за замовчуванням у бібліотеці OpenCV, до RGB – формату, необхідного для коректного функціонування інструментарія MediaPipe.

```
def process_frame(self, frame):
    rgb_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    results = self.hands.process(rgb_frame)
    all_landmarks = []
    if results.multi_hand_landmarks:
        for hand_landmarks in results.multi_hand_landmarks:
            self.mp_draw.draw_landmarks(frame, hand_landmarks, self.mp_hands.HAND_CONNECTIONS)
            all_landmarks.append(hand_landmarks)
    return frame, all_landmarks
```

Рисунок 3.2 – Спосіб реалізації оброблення кадрів

Далі кадр передається до модуля MediaPipe Hands, який виконує детекцію кистей рук та побудову моделі їхньої структури на основі 21 ключової точки для кожної руки. Результатом обробки є список координат виявлених точок, що представляють просторове положення пальців та долоні.

Ключові точки автоматично з’єднуються лініями згідно з анатомічною структурою кисті, що візуалізується безпосередньо на відеопотоці для зручності користувача. Цей процес виконується за допомогою методу draw_landmarks(), який наносить контури та точки на кадр (рис. 3.2).

Окрім основної візуалізації, система реалізує відображення процесу розпізнавання жесту (рис. 3.3, рис. 3.4) (наприклад, назви жесту або поточних параметрів збору даних), що виконується за допомогою бібліотеки PIL. Графічні елементи накладаються на верхню частину зображення, а оброблений кадр повертається у форматі, придатному для показу користувачу.

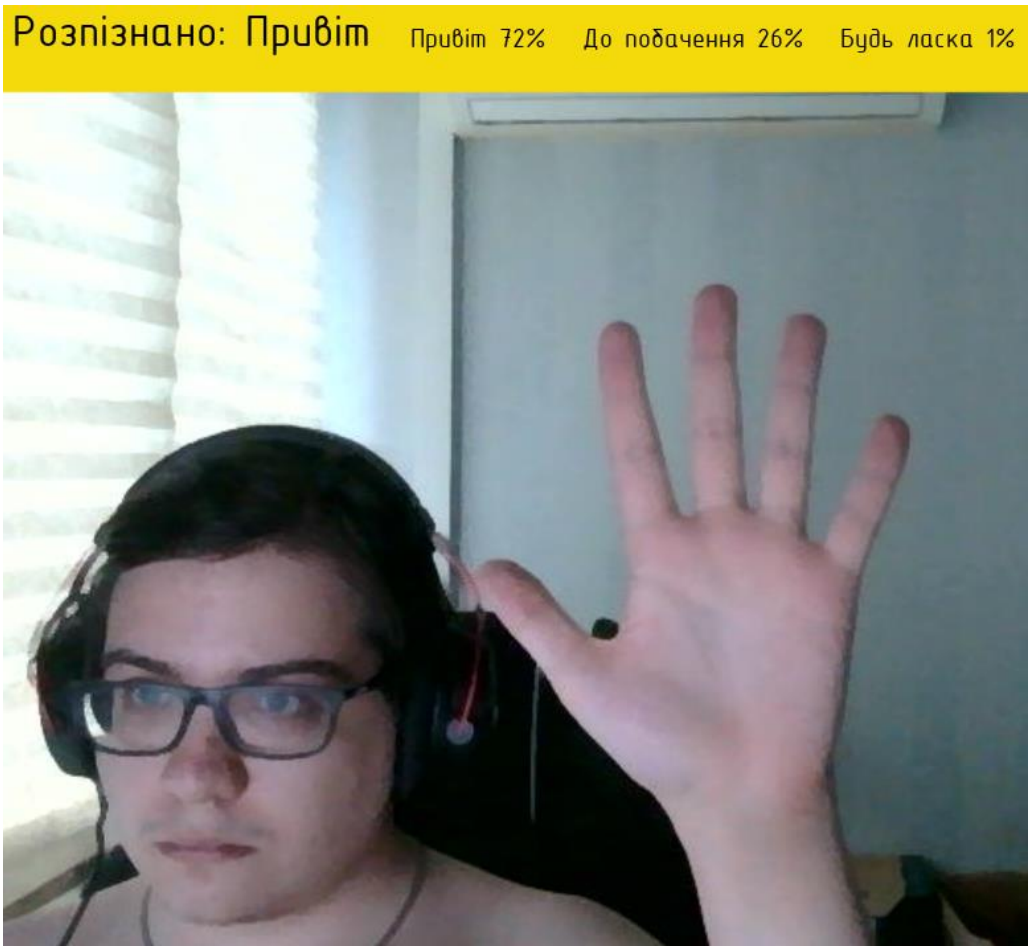

Рисунок 3.3 – Процес розпізнавання жесту (приклад № 1)

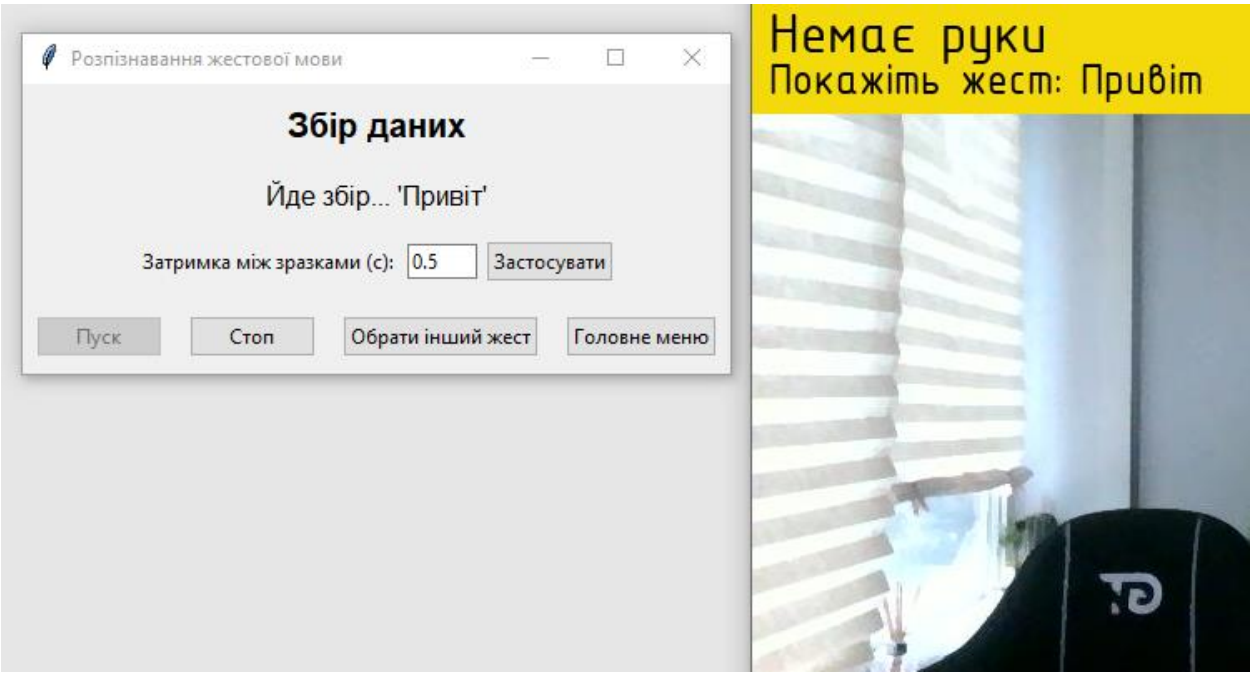

Рисунок 3.4 – Процес розпізнавання жесту (приклад № 2)

При потребі також змінюється масштаб зображення, що дозволяє адаптувати розміри вікна виводу до параметрів екрана користувача.

На етапі розпізнавання жестів одним із ключових процесів є формування векторів ознак, які описують положення рук у просторі. Для цього на кожному кадрі відео, що захоплюється з камери, спочатку виконується обробка зображення за допомогою функції *gr.process_frame(frame)* (рис. 3.5), яка повертає оброблений кадр та список знайдених рук. Кожна рука представлена у вигляді набору координат 21 ключової точки (*landmark*), які містять *x*, *y*, *z* координати у просторі. Тобто одна рука формує 63 значення, які й будуть ознаками для подальшого аналізу.

Метод *gr.extract_landmarks(hand_landmarks)* витягує координати всіх точок руки у вигляді плаского списку чисел. Вони додаються у вектор *vec*, який акумулює ознаки поточного кадру.

Однак у багатьох випадках користувач може показувати лише одну руку, тому передбачена обробка ситуацій, коли лише одна рука присутня в кадрі.

Щоб модель завжди отримувала на вхід однакову кількість ознак, до вектора додаються нулі для відсутньої руки (рис. 3.5).

```
vec = []
if hands_all:
    for hand_landmarks in hands_all:
        vec.extend(gr.extract_landmarks(hand_landmarks))
    if len(hands_all) == 1:
        vec.extend([0.] * 63)
else:
    vec = [0.] * 126
```

Рисунок 3.5 – Спосіб реалізації запису всіх точок координат рук

Це дозволяє уніфікувати розмірність вхідних даних та уникнути помилок при передачі даних у модель. Тобто просто додаємо 63 нульових значення до vec, які умовно представляють другу руку, якої насправді в кадрі немає.

Таким чином, загальна кількість ознак завжди дорівнює 126: по 63 для кожної руки. Якщо ж у кадрі не виявлено жодної руки, тобто список *hands_all* порожній, тоді весь вектор *vec* складається з нулів (рис. 3.5).

Після формування вектора ознак він додається до буфера *sequence_buffer*, який зберігає послідовність останніх 30 кадрів. Це необхідно, оскільки розпізнавання жесту ґрунтується не на одному кадрі, а на часовій послідовності руху рук. Саме така послідовність дозволяє врахувати динаміку жесту. Буфер реалізований за допомогою *deque* з обмеженням довжини *self.sequence_length*, яке зазвичай дорівнює 30. Це означає, що в кожен момент часу в буфері зберігаються ознаки останніх 30 кадрів. Як тільки буфер заповнюється, дані формуються у вигляді тривимірного масиву та передаються в модель для передбачення жесту.

Тут *data* набуває форми (1, 30, 126), що відповідає одній послідовності з 30 кадрів, кожен із яких представлений 126 ознаками. Така форма повністю відповідає очікуваному формату для RNN-моделі, яка навчається на часових рядах [28]. Цей підхід, який комбінує витяг просторових ознак (координати рук) з урахуванням часової структури (послідовність кадрів), забезпечує достатньо інформативне представлення жесту для точного його розпізнавання моделлю.

Розпізнавання жестів на основі машинного навчання, фокусує увагу на використанні рекурентної нейронної мережі типу LSTM (Long Short-Term Memory) для аналізу часових послідовностей. Такий підхід дозволяє ефективно обробляти серії ознак, які утворюються в процесі запису жесту у вигляді послідовних кадрів з камери. Передбачення виконується на основі вже натренованої моделі, створеної та збереженої за допомогою фреймворку TensorFlow/Keras. Усі жести представлені у вигляді фрагментів, кожен з яких містить набір числових ознак (наприклад, координати ключових точок скелету руки), які подаються на вхід моделі.

У процесі розпізнавання система накопичує набір кадрів у буфер (рис. 3.6), і як тільки кількість елементів у буфері досягає заданої довжини послідовності, вона перетворює цей буфер на вхідний масив, що відповідає формату, очікуваному LSTM-моделлю.

```
if len(sequence_buffer) == self.sequence_length:
    data = np.array(sequence_buffer).reshape(1, self.sequence_length, -1)
```

Рисунок 3.6 – Спосіб реалізації підготовки послідовності ознак для RNN-моделі

Цей код виконує підготовку перед фактичним розпізнаванням жесту:

– *sequence_buffer* це список, який постійно поповнюється новими кадрами;

– *self.sequence_length* це фіксована кількість кадрів, які потрібні для одного передбачення;

– коли буфер досягає необхідної довжини, він перетворюється у тривимірний масив *(reshape(1, N, F))*, і цей масив передається моделі.

Тривимірний набір чисел розміром (*1*, *N*, *F*), де *N* – довжина послідовності (кількість кадрів), а *F* – кількість ознак на один кадр. Такий формат є стандартним для LSTM-мереж, які обробляють послідовності вхідних векторів.

Модель, яка завантажується в систему, є заздалегідь навченим класифікатором, здатним розпізнавати жести на основі цих часових патернів. Як тільки модель робить передбачення, вона повертає не лише назву найімовірнішого жесту, а й рівень впевненості в цьому рішенні, що дозволяє застосунку фільтрувати випадкові або нечіткі жести.

Інтерфейс користувача (рис. 3.7, рис. 3.8) реалізовано за допомогою бібліотеки Tkinter, де окрема вкладка відповідає за запуск процесу розпізнавання. Це забезпечується вікном *RecognizeFrame*, який створює екран для візуального запуску модуля розпізнавання. Він дозволяє користувачу перемикатися між режимами, контролювати поточний стан програми та зупиняти/відновлювати розпізнавання за потреби.

```
class RecognizeFrame(tk.Frame):
    def __init__(self, parent, app):
        super().__init__(parent)
        self.app = app
        tk.Label(self, text="Розпізнавання жестів", font=("Arial", 15, "bold")).pack(pady=10)
        button_frame = tk.Frame(self)
        button_frame.pack(pady=10)
        ttk.Button(button_frame, text="Список жестів", command=self.app.show_gesture_list).pack(side=tk.LEFT, padx=8)
        ttk.Button(button_frame, text="Головне меню", command=lambda: [self.app.stop_recognize(), self.app.show_frame(MainMenu)]).pack(side=tk.LEFT, padx=8)
```

Рисунок 3.7 – Спосіб реалізації інтерфейсу користувача

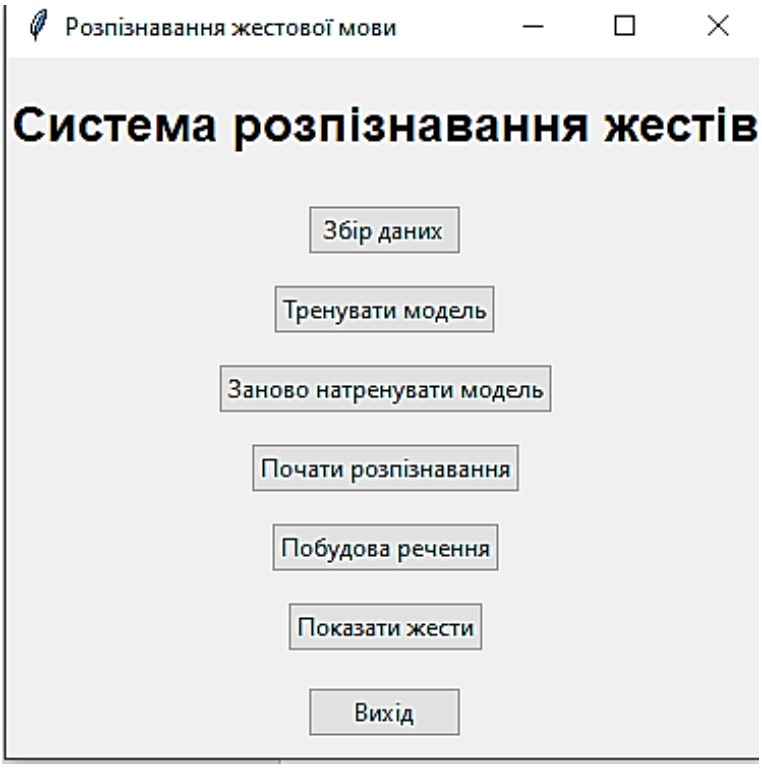

Рисунок 3.8 – Інтерфейс користувача

Важливо зазначити, що інтерфейс системи передбачає не лише розпізнавання окремих жестів, але і режим побудови речення, де розпізнані жести накопичуються у вигляді слів, що послідовно додаються до одного текстового рядка. Цей режим реалізовано у класі *SentenceFrame*, що також дозволяє редагувати побудоване речення – видаляти останнє слово або очищувати весь текст.

Після того, як модель розпізнавання жестів обробила послідовність даних і повернула свої прогнози, наступним важливим етапом є постобробка результатів (рис. 3.9). Цей процес необхідний для підвищення надійності системи та виключення випадкових або некоректних розпізнавань. Найперше, застосовується фільтрація результатів на основі встановленого порогового значення впевненості – *CONFIDENCE_THRESHOLD*. Це означає, що система враховує лише ті жести, які були розпізнані з впевненістю, що перевищує задану межу. Якщо жоден з варіантів не досяг цього порогу, програма вважає, що жест не розпізнано достовірно, і не видає результат.

```
if len(sequence_buffer) == self.sequence_length:
    data = np.array(sequence_buffer).reshape(1, self.sequence_length, -1)
    try:
        prediction_prob = self.model.predict(data, verbose=0)[0]
        pred_idx = np.argmax(prediction_prob)
        confidence = prediction_prob[pred_idx]
        if confidence < self.confidence_threshold:
            gesture_name = ""
        else:
            gesture_name = self.labels[pred_idx]

        top_indices = np.argsort(prediction_prob)[::-1]
        top_text_lines = []
        for i in top_indices[:3]:
            label = self.labels[i]
            conf_percent = prediction_prob[i] * 100
            top_text_lines.append(f"{label} {conf_percent:.0f}%")
        top_text = "\n".join(top_text_lines)
    except Exception:
        gesture_name = ""
        top_text = ""
```

Рисунок 3.9 – Спосіб реалізації передбачення
та «Топ-3» можливих жестів при розпізнанні

Далі з валідних результатів формується зведення: основним вважається той жест, який має найвищу оцінку ймовірності, а також додатково обчислюється список топ-3 найімовірніших варіантів. Кожен з них супроводжується відповідним відсотковим значенням, яке ілюструє рівень впевненості моделі [29]. Ця інформація виводиться у зручному для користувача форматі на екрані (рис. 3.10), що дозволяє краще зрозуміти, наскільки модель була впевнена у своєму виборі та які жести могли бути альтернативними.

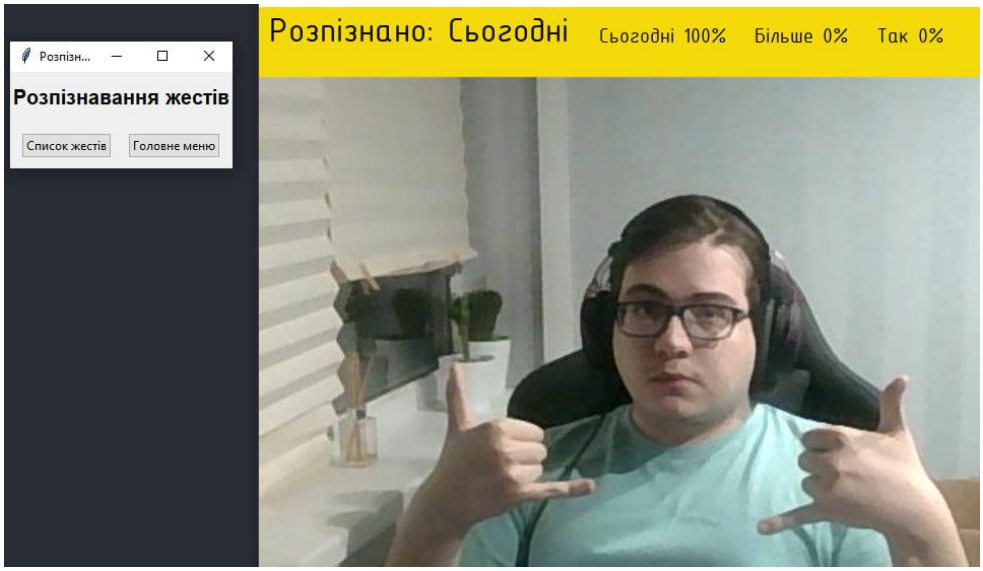

Рисунок 3.10 – Результат виведення інформації щодо розпізнавання

Для побудови інтерфейсу користувача використовувався графічний інтерфейс на базі Tkinter. Це дозволяє зручно взаємодіяти з програмою на різних етапах роботи.

Інтерфейс складається з кількох основних фреймів, кожен з яких відповідає за окремі функції системи.

Одним із таких вікон є *MainMenu* (рис. 3.11), що створює головне меню програми. У ньому користувач може вибирати різні опції, такі як збір даних, тренування моделі, розпізнавання жестів, побудова речень і вихід з програми. Всі ці кнопки ініціюють відповідні функції для перемикання між режимами програми.

```
class MainMenu(tk.Frame):
    def __init__(self, parent, app):
        super().__init__(parent)
        self.app = app
        tk.Label(self, text="Система розпізнавання жестів", font=("Arial", 18, "bold")).pack(pady=15)
        ttk.Button(self, text="Збір даних", command=lambda: app.show_frame(CollectFrame)).pack(pady=7)
        ttk.Button(self, text="Тренувати модель", command=app.train_model).pack(pady=7)
        ttk.Button(self, text="Почати розпізнавання", command=lambda: [app.show_frame(RecognizeFrame), app.start_recognize()]).pack(pady=7)
        ttk.Button(self, text="Побудова речення", command=lambda: [app.show_frame(SentenceFrame), app.start_sentence_mode()]).pack(pady=7)
        ttk.Button(self, text="Показати жести", command=app.show_gesture_list).pack(pady=7)
        ttk.Button(self, text="Вихід", command=app.root.destroy).pack(pady=10)
```

Рисунок 3.11 – Спосіб реалізації вікна головного меню

Інший важливий елемент інтерфейсу – це вікно *RecognizeFrame* (рис. 3.12, рис. 3.13), який відповідає за запуск і управління процесом розпізнавання жестів. Цей екран дозволяє користувачеві перемикатися між різними режимами програми і контролювати поточний стан, включаючи зупинку та відновлення процесу розпізнавання.

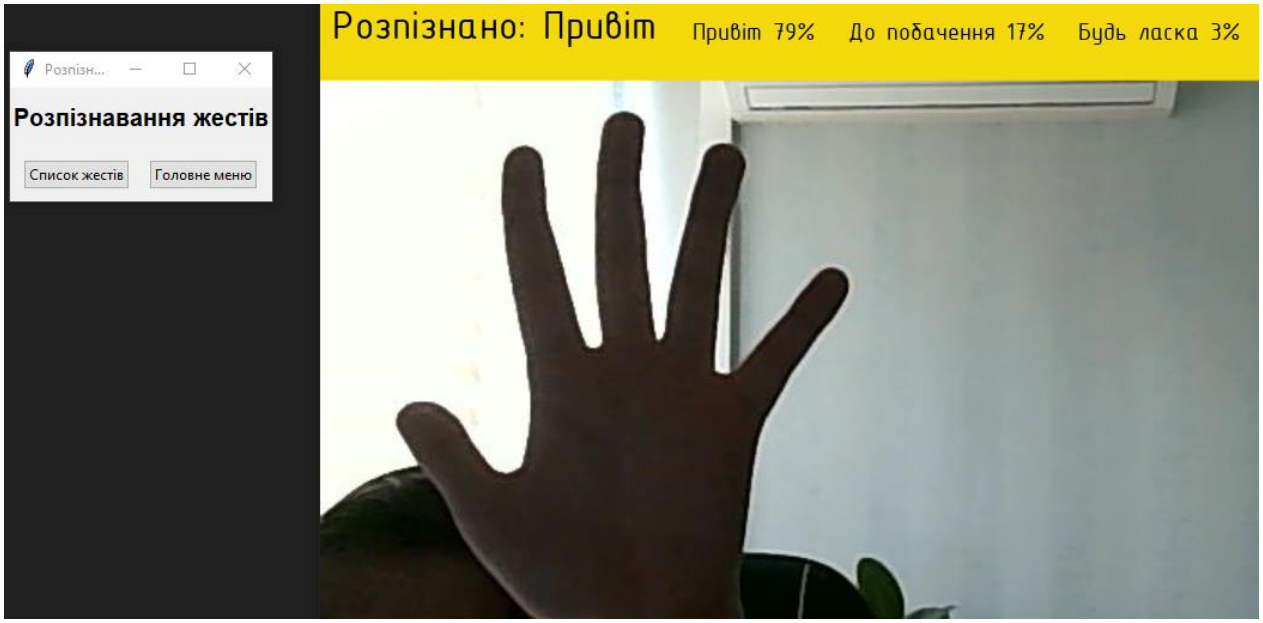

Рисунок 3.12 – Результат роботи вікна «Розпізнавання жестів»

```
class RecognizeFrame(tk.Frame):
    def __init__(self, parent, app):
        super().__init__(parent)
        self.app = app
        tk.Label(self, text="Розпізнавання жестів", font=("Arial", 15, "bold")).pack(pady=10)
        button_frame = tk.Frame(self)
        button_frame.pack(pady=10)
        ttk.Button(button_frame, text="Список жестів", command=self.app.show_gesture_list).pack(side=tk.LEFT, padx=8)
        ttk.Button(button_frame, text="Головне меню", command=lambda: [self.app.stop_recognize(), self.app.show_frame(MainMenu)]).pack(side=tk.LEFT, padx=8)
```

Рисунок 3.13 – Фрагмент реалізації роботи вікна «Розпізнавання жестів»

Відеопотік з накладенням графіки, таких як жовтий фон, текст і точки рук, також інтегровано в інтерфейс. Це дозволяє користувачеві в реальному часі бачити процес розпізнавання та спостерігати за позицією рук.

Для зручного управління збором даних були додані кнопки (рис 3.14, рис. 3.15) для запуску, зупинки збору, а також для вибору іншого жесту. Також було додано комфортний функціонал для побудови речень (рис. 3.16).

Для збору та збереження даних для навчання система використовує зручну структуру папок, де кожен «.npy» файл містить один запис послідовності кадрів, що відповідає конкретному жесту.

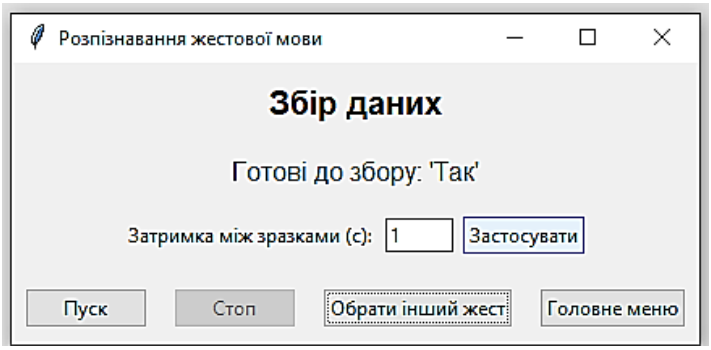

Рисунок 3.14 – Вікно «Збір даних»

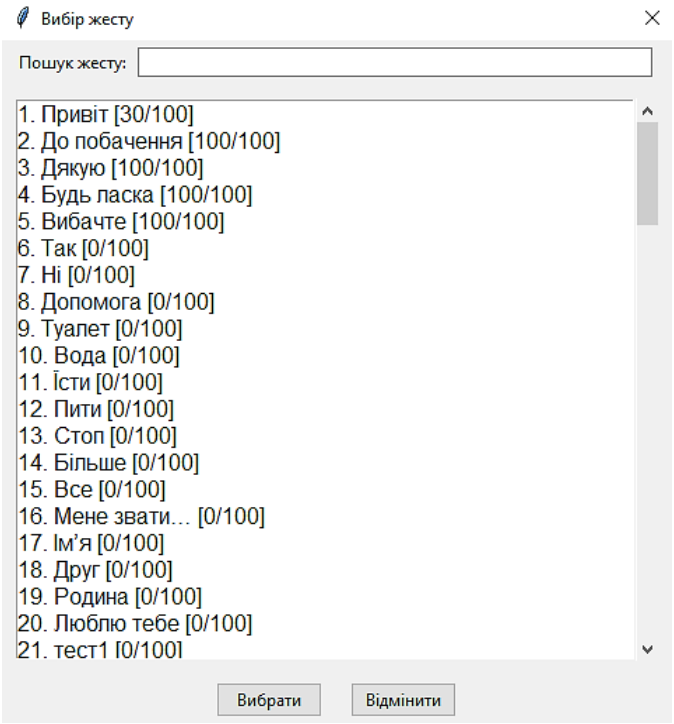

Рисунок 3.15 – Вікно «Вибір жесту» при натисканні «Вибір іншого жесту»

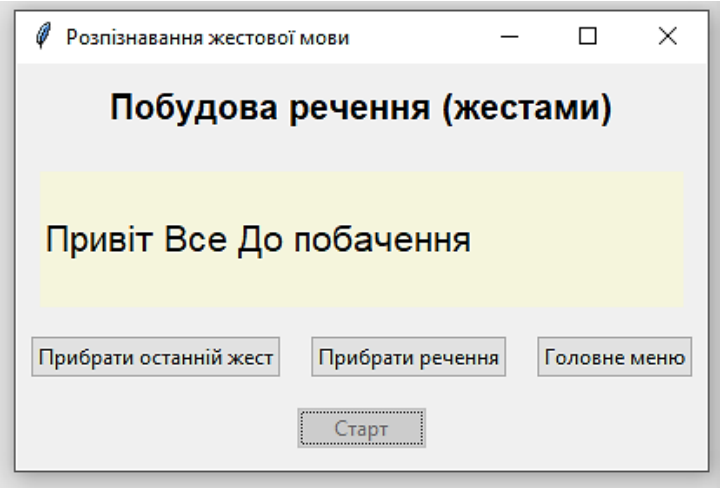

Рисунок 3.16 – Вікно «Побудова речення»

Ці файли зберігаються у відповідних папках для кожного жесту, що дозволяє ефективно організувати та зберігати великі об’єми даних для подальшого використання при тренуванні моделі.

Кожен жест зберігається в окремій папці у вигляді «.npy» файлів, що містять дані про кожен кадр послідовності. Зібрані дані автоматично підраховуються, і прогрес збору відображається на екрані (рис. 3.17), щоб користувач міг бачити, скільки кадрів було зібрано для конкретного жесту.

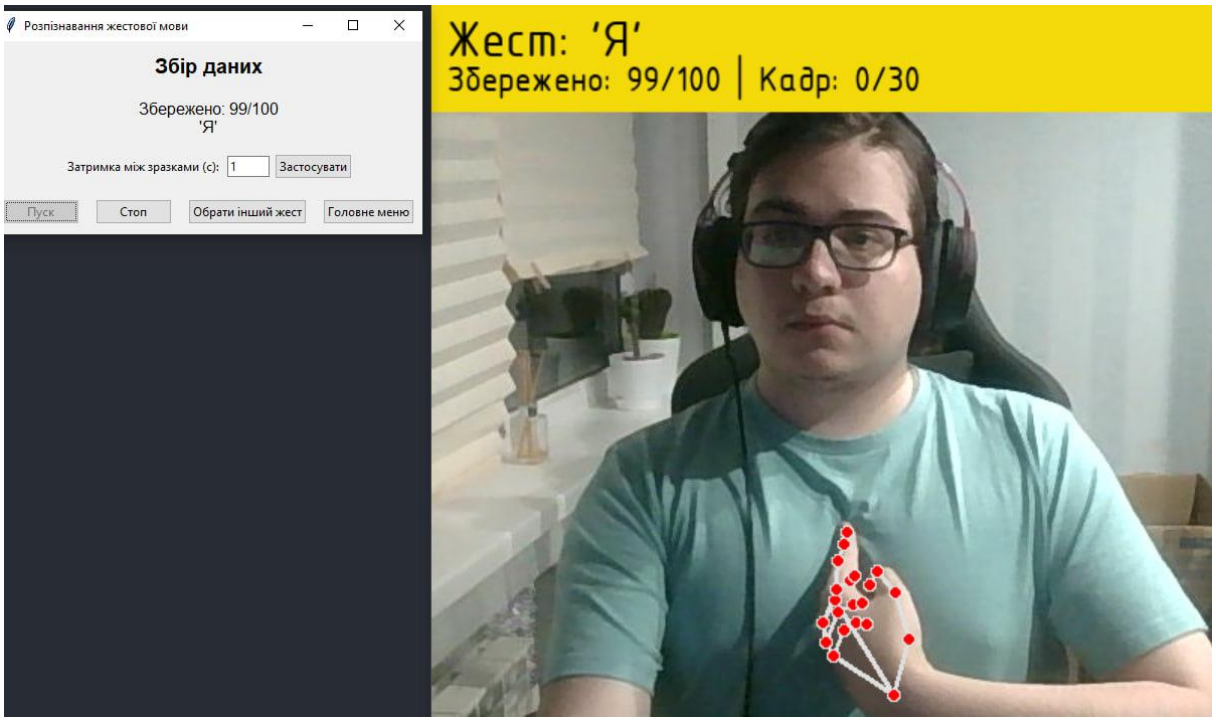

Рисунок 3.17 – Приклад збору даних з відображенням на екрані

У коді передбачено функцію, яка оновлює статус збору даних, показуючи, скільки кадрів зібрано для кожного жесту. Крім того, у меню збору є елемент для налаштування затримки між зразками, що дозволяє користувачу більш точно контролювати процес збору даних.

Кожен етап збору даних супроводжується відображенням прогресу, що дозволяє користувачу ефективно контролювати цей процес. Після збору достатньої кількості даних для кожного жесту настає етап тренування моделі.

У системі реалізована функціональність, що дозволяє автоматично запустити процес навчання натисканням однієї кнопки (рис. 3.18). Це здійснюється через головне вікно інтерфейсу – *MainMenu*, де передбачена відповідна кнопка «Тренувати модель».

```
Epoch 1: loss: 3.10, accuracy: 0.75
Epoch 2: loss: 1.78, accuracy: 0.75
Epoch 3: loss: 1.36, accuracy: 0.75
Epoch 4: loss: 1.15, accuracy: 0.76
Epoch 5: loss: 0.99, accuracy: 0.77
Epoch 6: loss: 0.84, accuracy: 0.78
Epoch 7: loss: 0.72, accuracy: 0.77
Epoch 8: loss: 0.62, accuracy: 0.80
Epoch 9: loss: 0.51, accuracy: 0.84
Epoch 10: loss: 0.43, accuracy: 0.87
Epoch 11: loss: 0.35, accuracy: 0.89
Epoch 12: loss: 0.31, accuracy: 0.91
Epoch 13: loss: 0.29, accuracy: 0.90
Epoch 14: loss: 0.25, accuracy: 0.91
Epoch 15: loss: 0.23, accuracy: 0.92
Epoch 16: loss: 0.22, accuracy: 0.91
Epoch 17: loss: 0.20, accuracy: 0.93
Epoch 18: loss: 0.18, accuracy: 0.92
Epoch 19: loss: 0.17, accuracy: 0.94
Epoch 20: loss: 0.16, accuracy: 0.94
Epoch 21: loss: 0.14, accuracy: 0.95
Epoch 22: loss: 0.13, accuracy: 0.95
Epoch 23: loss: 0.11, accuracy: 0.96
Epoch 24: loss: 0.12, accuracy: 0.95
Epoch 25: loss: 0.10, accuracy: 0.96
Epoch 26: loss: 0.11, accuracy: 0.95
Epoch 27: loss: 0.10, accuracy: 0.95
Epoch 28: loss: 0.08, accuracy: 0.96
Epoch 29: loss: 0.08, accuracy: 0.95
WARNING:absl:You are saving your model a
Epoch 30: loss: 0.08, accuracy: 0.96
Навчання завершено.
```

Рисунок 3.18 – Фрагмент процесу навчання

При її натисканні викликається метод *train_model*, який запускає повний цикл навчання на наявномій колекції даних. Цей процес є невід'ємною частиною логіки програми і не потребує від користувача спеціальних знань у сфері машинного навчання – все відбувається у фоновому режимі після натискання однієї кнопки та процес навчання виводиться у консоль (рис. 3.18).

Модель, що навчається, використовує інформацію з папок *dataset*, де кожен підкаталог відповідає певному жесту, а файли «.npy», які знаходяться у відповідних папках, це послідовності координат ключових точок рук зібраних з відеопотоку. Кожен файл – це окрема демонстрація жесту. Таким чином, модель отримує на вхід реальні часові послідовності рухів, які пізніше дозволяють їй ідентифікувати подібні шаблони у нових вхідних даних. Сам механізм збереження моделі реалізується у завершальній частині методу навчання: після того як всі епохи завершені, а параметри моделі скориговані, вона записується до окремого файлу «gesture_rnn_model.h5». Цей файл зберігається на диску і використовується в усіх подальших сесіях розпізнавання без необхідності повторного тренування. У такий спосіб зберігається результат інтелектуальної обробки, і вся система отримує доступ до нього через відповідний інтерфейс – зокрема, через *RecognizeFrame* та *SentenceFrame*, які викликають модель у реальному часі.

Модель навчається на основі раніше зібраних даних, зберігається у вигляді окремого файлу і автоматично завантажується у робочі модулі системи, що забезпечує безперервну і зручну роботу розпізнавання жестів при наступному запуску програми.

У програмі реалізовано режим розпізнавання жестів у реальному часі з можливістю формування речень із послідовностей розпізнаних жестів (рис. 3.16). Основна логіка цього процесу реалізується у вікні *SentenceFrame*, яке є одним із ключових елементів інтерфейсу користувача.

При запуску цього режиму активується потокове зчитування з камери, обробка відео, витяг ключових точок тіла, обличчя та рук за допомогою

MediaPipe, після чого формується послідовність кадрів, які подаються на вхід моделі для розпізнавання.

Результат розпізнавання – це передбачений жест, який додається до масиву *sentence*, якщо він відрізняється від попередніх та має достатній рівень впевненості. Завдяки цьому програма не дублює однакові жести поспіль і уникає випадкових помилок. Користувач бачить сформоване речення в окремому блоці інтерфейсу – воно формується з послідовних жестів у вигляді текстового рядка.

Для зручності передбачено функціонал корекції побудованого речення: зокрема, користувач може видалити останній доданий жест або повністю очистити речення, використовуючи відповідні кнопки на екрані («Прибрати останній жест», «Прибрати речення») (рис. 3.16). Таким чином, навіть у випадку помилкового розпізнавання є можливість швидко скоригувати результат без перезапуску системи.

Процес побудови речення є безперервним: поки активне вікно *SentenceFrame*, модель постійно приймає нові кадри з камери, здійснює передобробку, передбачення та оновлює речення у реальному часі. Це забезпечує плавну та природну взаємодію користувача з системою і дозволяє перетворювати серії жестів у цілісні текстові повідомлення, що особливо корисно для живої комунікації на основі жестової мови.

Для стабільної роботи системи та зручності користувача у коді реалізовано низку оптимізацій, спрямованих на підвищення продуктивності та усунення затримок під час виконання ресурсоємних процесів.

По-перше, активно використовується багатопоточність – один із ключових інструментів Python, який дозволяє виконувати кілька завдань одночасно. Наприклад, під час запуску режиму розпізнавання або збирання даних відеопотік обробляється в окремому потоці, що не блокує графічний інтерфейс користувача. Завдяки цьому взаємодія з кнопками, оновлення елементів GUI та виведення прогресу відбуваються без затримок навіть під час інтенсивної обробки кадрів.

По-друге, реалізовано масштабування інтерфейсу. Розміри вікон, у яких відображається відеопотік (наприклад, у *RecognizeFrame* або *CollectFrame*), адаптовано до типових розмірів екрану. Це забезпечує коректне відображення вмісту без обрізання, а також дає змогу чітко бачити накладені графічні елементи (наприклад, прямокутники, точки рук, текст розпізнавання). Також особливу увагу приділено обробці виняткових ситуацій. У всіх модулях, де здійснюється захоплення з камери, реалізовано перевірку на наявність потоку (*cap.isOpened()*), а також гарантоване вивільнення ресурсів у кінці сеансу (*cap.release()* та *cv2.destroyAllWindows()*). Це дозволяє уникнути помилок при повторному запуску камери та запобігає «зависанню» програми через невивільнені ресурси.

Таким чином, завдяки використанню багатопоточної обробки, гнучкого масштабу інтерфейсу та обережній роботі з зовнішніми пристроями, система забезпечує плавну й стабільну роботу як на етапах тренування та збору, так і під час реального розпізнавання.

3.3 Тестування розробленого застосунку та аналіз результатів

Щоб оцінити працездатність застосунку було проведено тестування з аналізом результатів. Після запуску відкривається вікно з головним меню (рис. 3.8) з відображенням назви програми та набором кнопок для взаємодії користувачем. Вікно можна редагувати за розміром, згорнути, поширити на весь екран та закрити за потреби. Далі перевіряється працездатність кнопок.

При натисканні на кнопку «Збір даних» зачинається початкове меню та відкривається нове вікно для колекції даних самим користувачем (рис. 3.14). На цьому екрані виводиться поточний стан програми, кнопки «Пуск» для початку роботи, «Стоп» для закічнення збору, «Обрати інший жест» для перегляду доступних жестів (рис. 3.19) та кнопка переходу до головного меню.

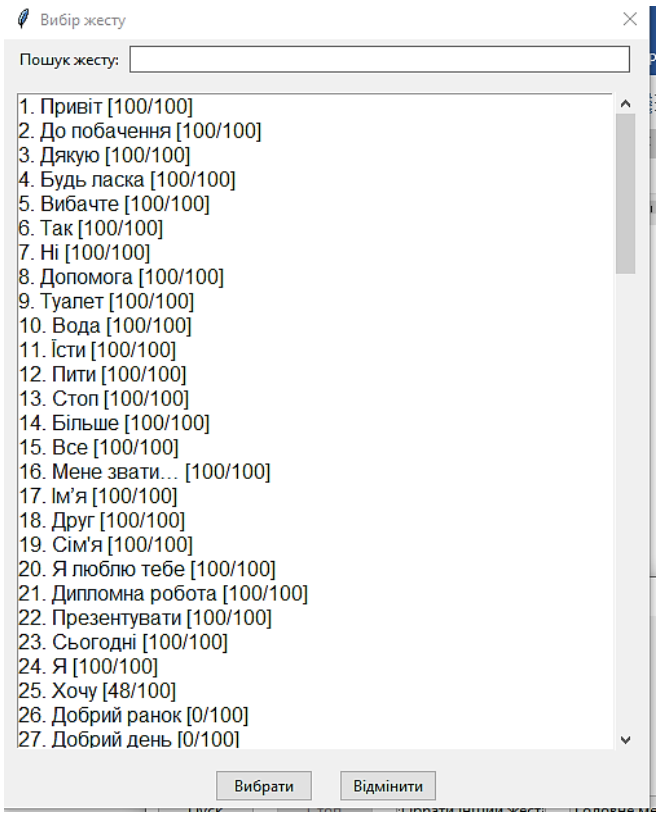

Рисунок 3.19 – Вікно перегляду доступних жестів

Якщо після запуску програми не показувати жест, то на екрані буде відповідне повідомлення (рис. 3.20).

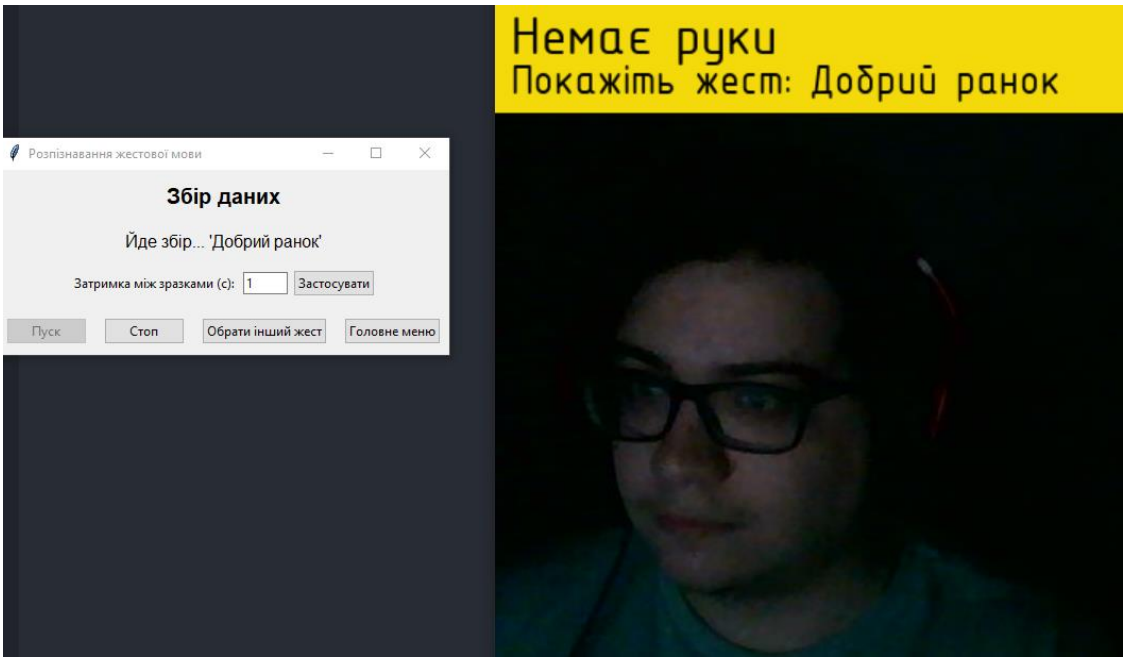

Рисунок 3.20 – Повідомлення про відсутність жесту/руки на екрані при зборі даних

Можна виставити затримку між зразками у секундах, для того, щоб було легше приготуватись для збору кожного зразку жесту, але, якщо ввести неправильно це значення, то буде видано помилку на екрані (рис. 3.21, рис. 3.22).

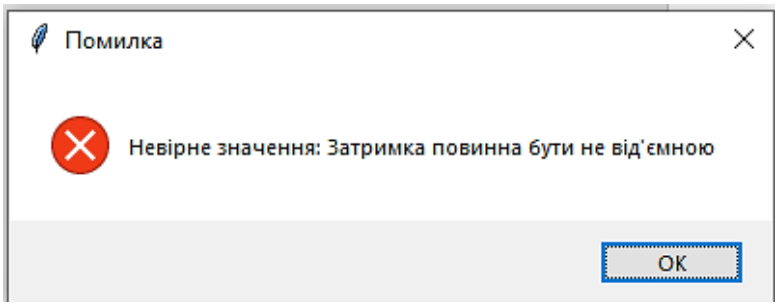

Рисунок 3.21 – Помилка при введені від’ємного числа

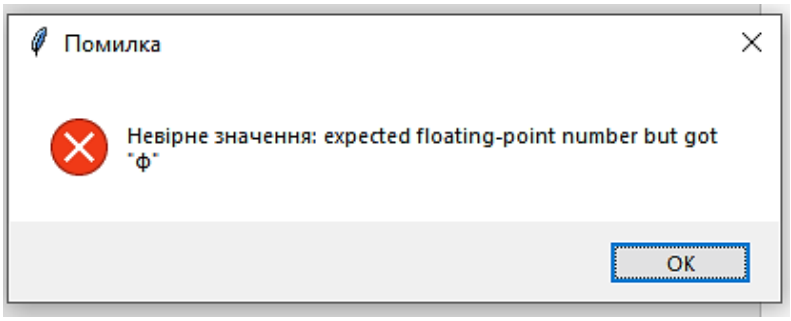

Рисунок 3.22 – Помилка при введені не числа

Якщо користувач хоче записати більше зразків для жесту, ніж доступно, то на екрані викликається повідомлення що збір цього жесту завершено (рис. 3.23).

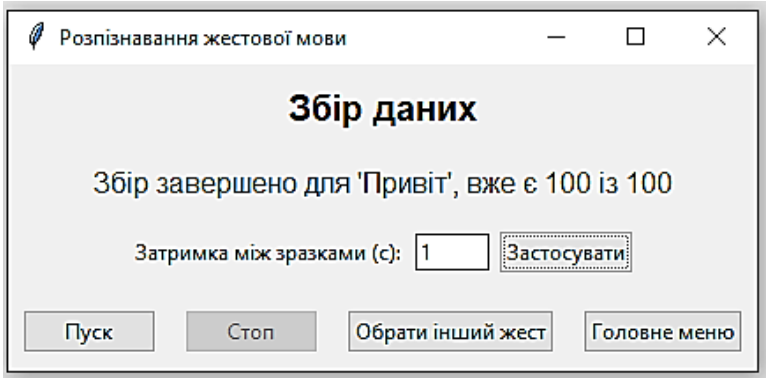

Рисунок 3.23 – Повідомлення про те, що жест вже був записаний повністю

Далі на головному меню йде кнопка «Тренувати модель». Якщо моделі в папці з програмою немає, то автоматично створюється нова модель та починається навчання (рис. 3.24). Після якого можна подивитися графік навчання (рис. 3.25).

```
Створена нова модель, починаємо навчання...
d:\anaconda\Lib\site-packages\keras\src\call
  current = self.get_monitor_value(logs)
Epoch 1: loss: 2.64, accuracy: 0.19
Epoch 2: loss: 1.94, accuracy: 0.32
Epoch 3: loss: 1.55, accuracy: 0.47
Epoch 4: loss: 1.30, accuracy: 0.56
Epoch 5: loss: 1.15, accuracy: 0.63
Epoch 6: loss: 0.89, accuracy: 0.72
Epoch 7: loss: 0.69, accuracy: 0.79
Epoch 8: loss: 0.53, accuracy: 0.84
Epoch 9: loss: 0.40, accuracy: 0.88
Epoch 10: loss: 0.40, accuracy: 0.88
Epoch 11: loss: 0.27, accuracy: 0.92
Epoch 12: loss: 0.37, accuracy: 0.89
Epoch 13: loss: 0.20, accuracy: 0.94
Epoch 14: loss: 0.19, accuracy: 0.94
Epoch 15: loss: 0.22, accuracy: 0.93
Epoch 16: loss: 0.22, accuracy: 0.93
Epoch 17: loss: 0.19, accuracy: 0.94
Epoch 18: loss: 0.19, accuracy: 0.94
Epoch 19: loss: 0.15, accuracy: 0.96
Epoch 20: loss: 0.08, accuracy: 0.97
Epoch 21: loss: 0.14, accuracy: 0.96
Epoch 22: loss: 0.10, accuracy: 0.97
Epoch 23: loss: 0.12, accuracy: 0.96
Epoch 24: loss: 0.19, accuracy: 0.95
Epoch 25: loss: 0.09, accuracy: 0.97
Epoch 26: loss: 0.07, accuracy: 0.98
Epoch 27: loss: 0.10, accuracy: 0.97
Epoch 28: loss: 0.11, accuracy: 0.96
Epoch 29: loss: 0.06, accuracy: 0.98
WARNING:absl:You are saving your model as an
Epoch 30: loss: 0.05, accuracy: 0.98
Навчання завершено.
```

Рисунок 3.24 – Повідомлення про створення нової моделі та процес її навчання

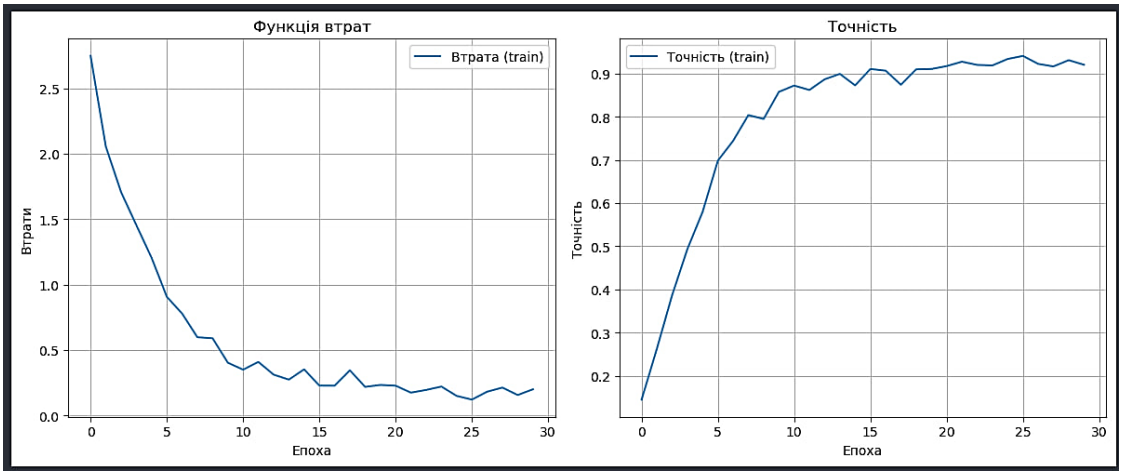

Рисунок 3.25 – Результат навчання у вигляді графіків

Наступна кнопка «Почати розпізнавання», котра відкриває нове вікно з списком жестів та можливістю повернення до головного меню (рис. 3.26).

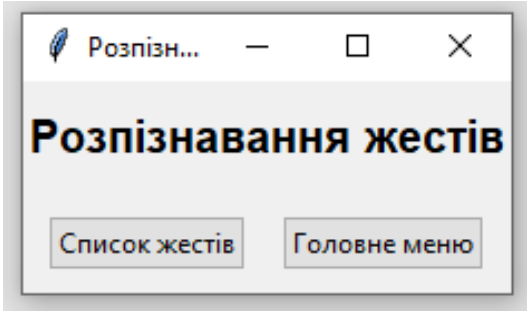

Рисунок 3.26 – Вікно меню розпізнавання жестів у реальному часі

Для отримання результату потрібно отримати більше або 90% відповідності жесту. Якщо нічого не показувати, то програма нічого і не видасть.

Далі перевіряємо жест у різних умовах: при поганому освітленні, при нормальному освітленні, в різних кутках камери показувати жест (рис. 3.27 – рис. 3.29). Програма добре себе поводить у ріних умовах, але результати можуть погіршуватися при складності їх відтворення (наприклад багато різких рухів).

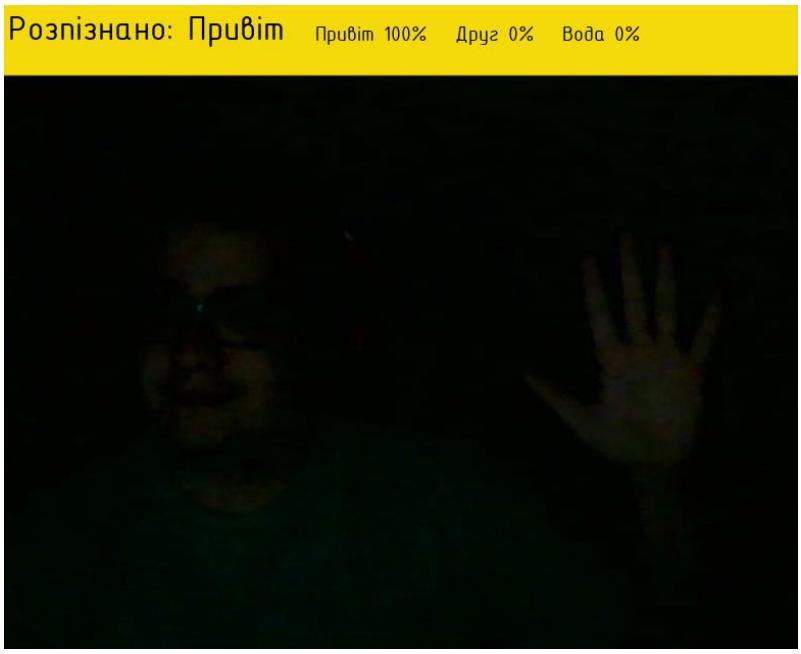

Рисунок 3.27 – Результат розпізнавання при поганому освітленні

Рисунок 3.28 – Результат при нормальному освітленні

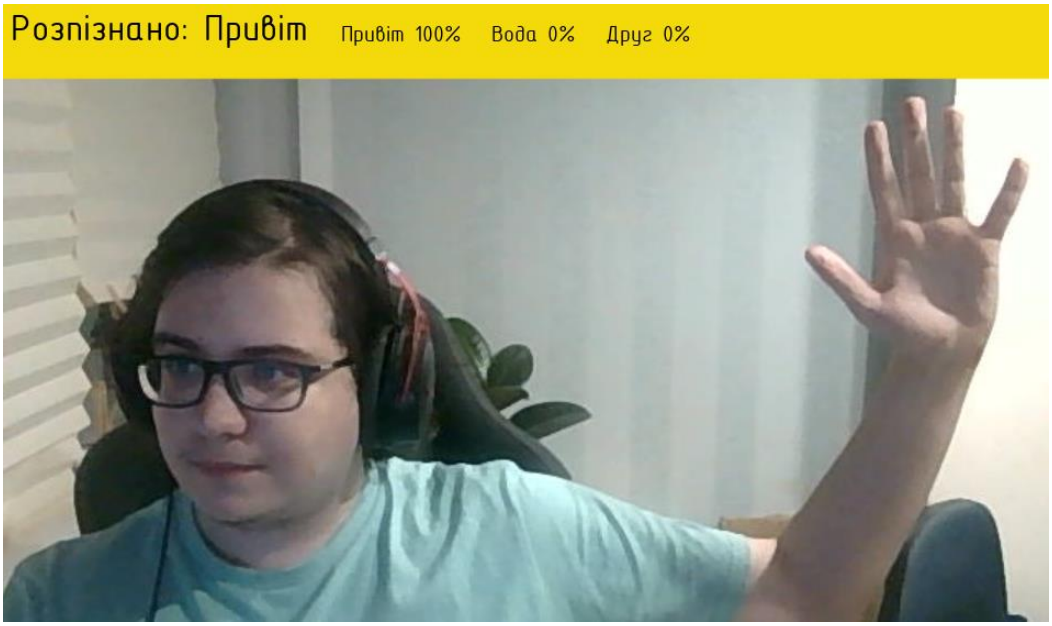

Рисунок 3.29 – Результат розпізнавання жесту при демонструванні в різних кутках камери

Інша кнопка для тестування це «Побудовання речення». При багаторазовому натисканні на «Прибрати останній жест» або «Прибрати речення» не виникає ніяких помилок. Після запуску програми якщо довго показувати один жест то він не буде дублюватися.

Останньою кнопкою є «Показати жести», де можна скористуватися пошуком доступних жестів. Якщо почати вводити назву жесту, на букву котрий він починається, але такого жесту нема, то видасть повідомлення про це (рис. 3.30).

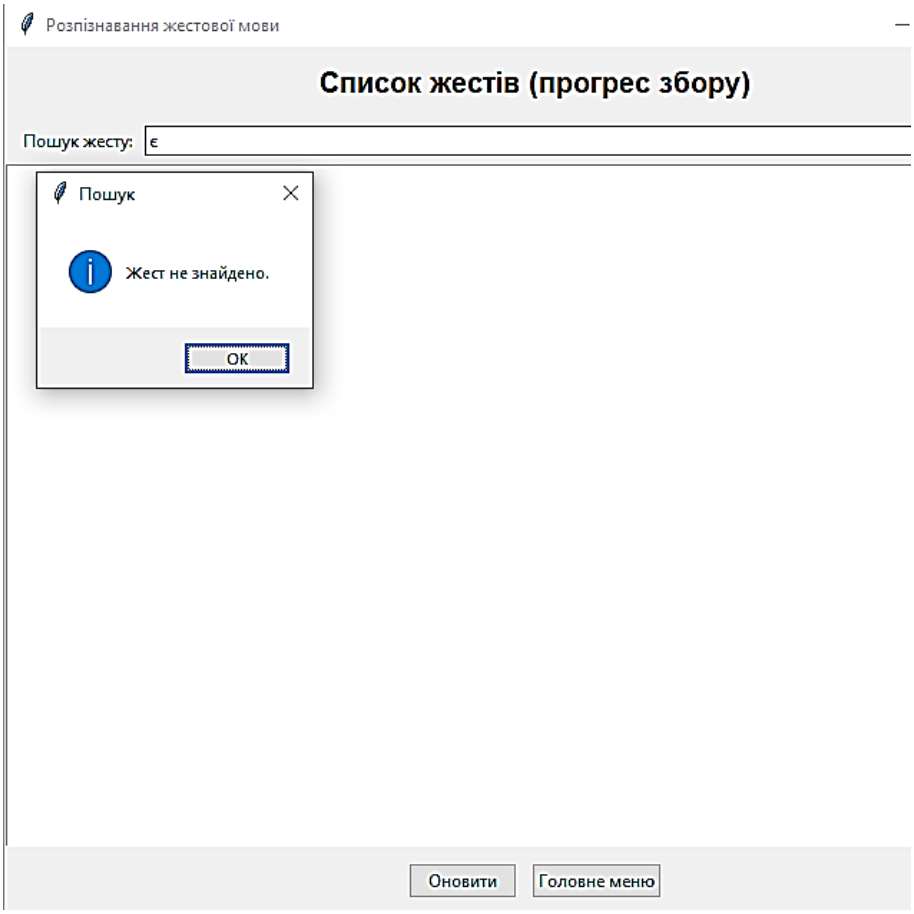

Рисунок 3.30 – Результат незнайденої або неправильної назви жесту

3.4 Перспективи подальшої роботи

Розроблений у рамках цього проєкту прототип системи розпізнавання жестової мови демонструє ефективність базового підходу із застосуванням нейронних мереж LSTM для класифікації жестів за послідовністю координат із відеопотоку. Водночас існує низка напрямків, що відкривають широкі можливості для подальшого розвитку, вдосконалення та розширення функціональності системи.

Для підвищення практичної користі системи планується збільшення кількості розпізнаваних жестів, зокрема включення складніших фраз та діалогових конструкцій. Це дозволить зробити спілкування з системою більш природним і багатогранним.

Можливе впровадження сучасних архітектур нейронних мереж, таких як трансформери або CNN-RNN гібриди, що забезпечують кращу обробку просторово-тимчасових ознак у відеопотоках.

Оптимізація системи розпізнавання жестової мови для роботи на мобільних та вбудованих пристроях є однією з головних перспектив подальшої роботи. Вже на початковому етапі розробки проекту передбачалося, що система повинна бути не лише ефективною на стаціонарних комп’ютерах, а й адаптованою до ресурсних обмежень мобільних платформ.

Однією з перспектив подальшого розвитку системи є створення навчальних та ігрових застосунків, які сприятимуть розвитку та поширенню української жестової мови. Інтерактивні платформи, що поєднують у собі навчання жестам та побудову речень, можуть значно підвищити зацікавленість користувачів, зробити процес оволодіння навичками більш доступним і захопливим.

Такі застосунки матимуть великий потенціал не лише у навчанні, а й у реабілітації людей з порушеннями слуху, допомагаючи їм ефективніше взаємодіяти з навколишнім світом та покращуючи якість їхнього життя.

ВИСНОВКИ

У даній кваліфікаційній роботі продемонстровано процес розроблення програмного застосунку для розпізнавання жестової мови у реальному часі. Робота включала в себе аналіз існуючих продуктів, які вже широко використовуються та мають схожі можливості, аналіз методів розпізнавання у різних варіаціях, розробку архітектури програми, а також її реалізація за допомогою сучасних технологій.

Розроблено програмний застосунок, що дозволяє збирати, тренувати та застосовувати модель для точного розпізнавання базових жестів у режимі реального часу. Для реалізації використані популярні бібліотеки комп'ютерного зору (MediaPipe, OpenCV) та глибокого навчання (TensorFlow, Keras) з застосуванням рекурентних нейронних мереж LSTM, що забезпечують обробку послідовностей кадрів. Програма підтримує інтерфейс для збору нових даних, тренування моделей, а також інтерактивне розпізнавання жестів із виводом результатів на екран.

Тестування системи проводилось як вручну, так і автоматизовано на заздалегідь зібраному наборі даних, що дозволило визначити оптимальні параметри для навчання та пороги впевненості моделі.

Розглянуті подальші напрямки покращення – це розширення словника жестів, застосування більш складних моделей нейронних мереж, оптимізація для мобільних платформ, розробка навчальних застосунків для популяризації жестової мови. Результати роботи мають велике значення для розвитку інклюзивних технологій та полегшення комунікації людей з порушеннями слуху.

Результати роботи апробовано у вигляді тез доповіді під час Міжнародного молодіжного форуму «РАДІОЕЛЕКТРОНІКА ТА МОЛОДЬ У XXI СТОЛІТТІ» [30].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Tvoroshenko I., and Gorokhovatskyi V. (2022) The Application of Hybrid Intelligence Systems for Dynamic Data Analysis, *International Journal of Engineering and Information Systems*, 6(2), pp. 40-48.

2. Pomazan V., Tvoroshenko I., and Gorokhovatskyi V. (2023) Development of an application for recognizing emotions using convolutional neural networks, *International Journal of Academic Information Systems Research*, 7(7), pp. 25-36.

3. КПІмедія. Застосунок для розпізнавання мови жестів. Електронний архів КПІ. URL: https://www.youtube.com/watch?v=fw27Ym 7f7tM (дата звернення 19.04.2025).

4. Leeson L., Sheikh H. (2009). SIGNALL: Developing online and blended deaf studies course content across EU borders. 1st International Conference on Education and New Learning Technologies.

5. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., and Zeghid M. (2024) Improving the effectiveness of image classification structural methods by compressing the description according to the information content criterion, *Computers, Materials & Continua*, vol. 80, no. 2, pp. 3085-3106.

6. Google Research. Project Relate: A personalized speech recognition app for people with speech impairments. URL: https://sites.research.google/relate/ (дата звернення 21.04.2025).

7. Gorokhovatskyi V., Tvoroshenko I., and Yakovleva O. (2024) Transforming image descriptions as a set of descriptors to construct classification features, *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 33, no. 1, pp. 113-125.

8. Dallas Innovates. KinTrans' Sign Language Tech Translates Movements Into Text and Voice. URL: https://dallasinnovates.com/kintrans-sign-language-tech-translates-movements-text-voice/ (дата звернення 05.04.2025).

9. Pomazan V., Tvoroshenko I., and Gorokhovatskyi V. (2023) Handwritten character recognition models based on convolutional neural networks, *International Journal of Academic Engineering Research*, 7(9), pp. 64-72.

10. Гороховатський В., Передрій О., Творошенко І., Марков Т. (2023) Матриця відстаней для множини компонентів структурного опису як інструмент для створення класифікатора зображень, *Сучасні інформаційні системи*, 7(1), С. 5-13.

11. Кульбіда, С. В. (2020). Linguistic personality in Ukrainian language education of deaf education direction. Education: Modern Discourses, 3, 131-142.

12. Давидов, М.В., Нікольський, Ю.В. (2008) Методи та засоби опрацювання зображень реального часу для ідентифікації елементів жестової мови, 131-138.

13. Zhou, Y., Huang, Z., & Wang, L. (2021). Real-time hand gesture recognition based on improved YOLOv4 algorithm. Journal of Intelligent & Fuzzy Systems, 41(4), 5217–5226.

14. Cihan, A., Demir, H., & Yildiz, M. (2020). Dynamic hand gesture recognition using CNN and LSTM networks. IEEE Access, 8, 142938–142947.

15. Яценко, О. В. (2022). Проблеми адаптації мобільних застосунків для української жестової мови. Сучасні технології в освіті, 4(2), 98–103.

16. Ільченко, М. Л. (2020). Використання ігрових технологій для навчання жестової мови у школах-інтернатах. Освітній простір, 2, 45–50.

17. Gorokhovatskyi, V., Tvoroshenko, I., Kobylin, O., & Vlasenko, N. (2023). Search for visual objects by request in the form of a cluster representation for the structural image description, *Advances in Electrical and Electronic Engineering*, 21(1), pp. 19-27.

18. Ren, Z., Yuan, J., Meng, J., & Zhang, Z. (2013). Robust part-based hand gesture recognition using kinect sensor. IEEE transactions on multimedia, 15(5), 1110-1120.

19. Gorokhovatskyi V., Chmutov Y., Tvoroshenko I., and Kobylin O. (2025) Reducing computational costs by compressing the structural description in image classification methods, *Advanced Information Systems*, vol. 9, no. 1, pp. 5–12.

20. Molchanov, P., Gupta, S., Kim, K., & Kautz, J. (2015). Hand gesture recognition with 3D convolutional neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops (pp. 1-7).

21. Tvoroshenko I., Gorokhovatskyi V., Kobylin O., and Tvoroshenko A. (2023) Application of deep learning methods for recognizing and classifying culinary dishes in images, *International Journal of Academic and Applied Research*, 7(9), pp. 57-70.

22. Obaid, F., Babadi, A., & Yoosofan, A. (2020). Hand Gesture Recognition in Video Sequences Using Deep Convolutional and Recurrent Neural Networks. Appl. Comput. Syst., 25(1), 57-61.

23. Kim, W., Sung, J., Saakes, D., Huang, C., & Xiong, S. (2021). Ergonomic postural assessment using a new open-source human pose estimation technology (OpenPose). International Journal of Industrial Ergonomics, 84, 103164.

24. Tvoroshenko I., Pomazan V., Gorokhovatskyi V., and Kobylin O. (2023) Application of video data classification models using convolutional neural networks, *International Journal of Academic and Applied Research*, 7(11), pp. 134-145.

25. Gorokhovatskyi V., Tvoroshenko I. (2023) Identification of visual objects by the search request. *International scientific symposium «INTELLIGENT SOLUTIONS-S». Computational intelligence (results, problems and perspectives). Decision making theory: proceedings of the international symposium, September 28, 2023, Kyiv-Uzhorod, Ukraine*, pp. 25-27.

26. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., Gadetska S., and Al-Dhaifallah M. (2023) Statistical data analysis models for determining the relevance of structural image descriptions, *IEEE Access*, vol. 11, pp. 126938-126949.

27. Gorokhovatskyi V., Tvoroshenko I., Yakovleva O., Hudáková M., and Gorokhovatskyi O. (2024) Application a committee of Kohonen neural networks to training of image classifier based on description of descriptors set, *IEEE Access*, vol. 12, pp. 73376-73385.

28. Gorokhovatskyi V., Tvoroshenko I., Yakovleva O., and Hudáková M. (2025) Image description compression in classification structural methods, *IEEE Access*, vol. 13, pp. 43631-43641.

29. Yakovleva O., Matúšová S., Tvoroshenko I., and Isaiev Y. (2024) Visitor counting based on video stream analysis from surveillance cameras to solve various business problems, *Verejná správa a regionálny rozvoj ekonómia, manažment a marketing*, XX(1), pp. 67-87.

30. Мармур О.М. (2025) Аналіз особливостей української жестової мови під час розроблення програмного застосунку. *Радіоелектроніка і молодь у XXI столітті: тези доповідей 29-го Міжнародного молодіжного форуму (Харків, 16–19 квітня 2025 р.)*. Харків: ХНУРЕ, 2025. Т. 7. С. 99-101.