МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп’ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління

МИХАЙЛИШИНА Марія Петрівна

Програмний модуль створення субтитрів мови жестів з використанням комп'ютерного зору і машинного навчання/Software module for generating sign language subtitles using computer vision and machine learning

Спеціальність 122 – Комп’ютерні науки
Освітньо-професійна програма – Комп’ютерні науки

Кваліфікаційна робота

Виконала студентка групи КН-41
М. П. Михайлишина
_____

Науковий керівник:
к.т.н., доцент І.В. Турченко
_____

Кваліфікаційну роботу допущено до захисту
«___»_____ 2025 р.

В.о. завідувача кафедри
_____ Н.М. Васильків

Тернопіль – 2025

Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління
Освітній ступінь «бакалавр»
спеціальність 122 – Комп'ютерні науки
освітньо-професійна програма – Комп'ютерні науки

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
_____ Н.М. Васильків
«____» _____2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
МИХАЙЛИШИНА Марія Петрівна
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи: Програмний модуль створення субтитрів мови жестів з використанням комп'ютерного зору і машинного навчання/Software module for generating sign language subtitles using computer vision and machine learning
керівник роботи Турченко Ірина Василівна к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом по університету від 20 грудня 2024 р. № 938.
2. Строк подання студентом закінченої кваліфікаційної роботи 25 травня 2025 р.
3. Вихідні дані до кваліфікаційної роботи: завдання на кваліфікаційну роботу студента, наукові статті, технічна література.
4. Основні питання, які потрібно розробити:

– проаналізувати предметну область розпізнавання жестової мови, дослідити сучасні підходи на основі комп'ютерного зору та машинного навчання, визначити їх переваги та обмеження;

– сформувати концепцію побудови програмного модуля, що дозволяє розпізнавати жестів у реальному часі з подальшим створенням субтитрів;

– спроектувати структуру та алгоритмічне забезпечення програмного модуля створення субтитрів до мови жестів;

– реалізувати програмний модуль, що забезпечує переведення жестів у текстові субтитри з використанням комп'ютерного зору та машинного навчання;

– провести тестування роботи програмного модуля, зокрема оцінку точності розпізнавання.

5. Перелік графічного матеріалу в роботі:

– схема алгоритму роботи програмного модуля переведення жестової мови в субтитри;

– схема алгоритму обробки відеозображення;

– схема алгоритму розпізнавання жестів;

– схема алгоритму перетворення жесту в текст.

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 20 грудня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1	Затвердження теми кваліфікаційної роботи, ознайомлення з літературними джерелами та складання плану роботи	до 01.01. 2025 р.	
2	Написання 1 розділу кваліфікаційної роботи	до 01.02. 2025 р.	
3	Написання 2 розділу кваліфікаційної роботи	до 01.04.2025 р.	
4	Написання 3 розділу кваліфікаційної роботи	до 01.05. 2025 р.	
5	Представлення попереднього варіанту кваліфікаційної роботи, перевірка та внесення змін керівником	до 15.05.2025 р.	
6	Опрацювання зауважень та представлення завершеного варіанту кваліфікаційної роботи. Підготовка супроводжуючих документів	до 25.05.2025 р.	
7	Перевірка кваліфікаційної роботи на оригінальність тексту	до 30.05.2025 р.	
8	Оформлення кваліфікаційної роботи та отримання допуску до захисту	до 10.06.2025 р.	
9	Подання кваліфікаційної роботи до захисту на засіданні атестаційної комісії	до 10.06. 2025 р.	

Студент_____ М.П. Михайлишина
(підпис) (прізвище та ініціали)

Керівник кваліфікаційної роботи_____ І.В. Турченко
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Кваліфікаційна робота на тему «Програмний модуль створення субтитрів мови жестів з використанням комп'ютерного зору і машинного навчання» на здобуття освітнього ступеня «бакалавр» зі спеціальності 122 «Комп’ютерні науки» освітньої програми «Комп’ютерні науки» написана обсягом в 56 сторінок і містить 21 ілюстрацію, 1 таблицю, 4 додатки та 29 використаних джерела.

Метою роботи є розроблення програмного модуля створення субтитрів мови жестів з використанням комп'ютерного зору і машинного навчання.

Методами дослідження обрано аналіз предметної області та існуючих рішень, методи комп’ютерного зору для обробки відеопотоку, метод глибокого навчання із застосуванням рекурентних нейронних мереж (LSTM) для класифікації жестів, а також метод експериментального тестування для перевірки ефективності розробленого модуля.

У результаті роботи спроєктовано та реалізовано програмний модуль, що забезпечує автоматичне розпізнавання жестів із відео й перетворення їх у текстові субтитри.

Результати можуть бути використані в системах онлайн-комунікації, відеотрансляцій, освітніх та інших цифрових платформах для забезпечення іклюзивного та безбар’єрного простору спілкування.

Ключові слова: ЖЕСТОВА МОВА, КОМП’ЮТЕРНИЙ ЗІР, МАШИННЕ НАВЧАННЯ, РОЗПІЗНАВАННЯ ЖЕСТІВ, LSTM, СУБТИТРИ.

ANNOTATION

Qualification work on the topic «Software module for generating sign language subtitles using computer vision and machine learning» for Bachelor's degree on speciality 122 «Computer Science» educational and professional program «Computer Science» is written on 56 pages and it contains 21 figures, 1 table, 4 annexes and 29 sources.

The purpose of the work is to develop a software module for generating subtitles from sign language using computer vision and machine learning techniques.

Research methods include domain and literature analysis, computer vision methods for video stream processing, deep learning techniques with recurrent neural networks (LSTM) for gesture classification, and experimental testing to evaluate the effectiveness of the developed module.

As a result of the work, a software module was designed and implemented to automatically recognize gestures from video and convert them into text subtitles.

The results can be applied in online communication systems, video broadcasting platforms, educational and other digital platforms to ensure an inclusive and barrier-free communication space.

Keywords: SIGN LANGUAGE, COMPUTER VISION, MACHINE LEARNING, GESTURE RECOGNITION, LSTM, SUBTITLES.

ЗМІСТ

Вступ 7
1 Аналіз предметної області та ростановка задачі проєктування 9
1.1 Опис предметної області 9
1.2 Огляд і аналіз існуючих рішень 13
1.3 Постановка завдання проєктування 16
2 Проєктування програмного модуля створення субтитрів мови жестів 19
2.1 Загальна структура проєктованого модуля 19
2.2 Алгоритмічне забезпечення 21
2.3 Інструменти комп'ютерного зору, машинного навчання та апаратне забезпечення, що застосовуються у задачах перекладу жестової мови у текст 25
3 Реалізація програмного модуля створення субтитрів мови жестів з використанням комп'ютерного зору і машинного навчання 30
3.1 Реалізація програмного забезпечення 30
3.2 Інтерфейс користувача 34
3.3 Тестування модуля 37
Висновки 41
Список використаних джерел 42
Додаток А Програмний код навчання моделі 44
Додаток Б Програмний код розпізнавання жестів у реальному часі 47
Додаток В Сертифікат участі у конференції 49
Додаток Г Копія опублікованих результатів 51

ВСТУП

У сучасних умовах стрімкого розвитку цифрових технологій особливого значення набуває створення інноваційних рішень, спрямованих на забезпечення інклюзивності та доступності інформаційного середовища для всіх категорій населення. Однією з вразливих груп залишаються особи з порушеннями слуху, для яких жестова мова є основним засобом комунікації. Проте, у повсякденному житті, професійному середовищі та сфері обслуговування часто виникають значні труднощі в комунікації між носіями жестової мови та тими, хто її не розуміє. Це створює серйозні бар'єри для соціалізації, доступу до освіти, медицини, правосуддя та інших важливих сфер.

У цьому контексті актуальним є застосування сучасних цифрових технологій, зокрема, комп'ютерного зору та методів машинного навчання для автоматичного розпізнавання жестів та трансформації жестової мови у текстовий формат або мовний супровід. Такі технології здатні не лише підвищити якість життя людей із порушеннями слуху, а й сприяти загальному прогресу в напрямку побудови інклюзивного суспільства, де технології адаптуються до потреб людини, а не навпаки.

Крім того, розвиток машинного навчання, збільшення обчислювальних потужностей і поява відкритих бібліотек комп'ютерного зору, таких як OpenPose, MediaPipe, відкривають нові можливості для ефективного вирішення задач реального часу, зокрема, автоматизованого створення субтитрів до відео або трансляцій на основі жестової мови.

Таким чином, створення програмного модуля для розпізнавання жестової мови та автоматичного генерування субтитрів є не лише актуальним технічним завданням, але й важливим кроком у напрямі цифрової інклюзії, що відповідає принципам сталого розвитку, прав людини та цифрової гуманності.

Метою кваліфікаційної роботи є створення програмного модуля створення субтитрів мови жестів з використанням комп'ютерного зору і машинного навчання.

Для досягнення поставленої мети потрібно виконати наступні завдання:

– проаналізувати предметну область розпізнавання жестової мови, дослідити сучасні підходи на основі комп’ютерного зору та машинного навчання, визначити їх переваги та обмеження;

– сформувати концепцію побудови програмного модуля, що дозволяє розпізнавати жестів у реальному часі з подальшим створенням субтитрів;

– спроектувати структуру та алгоритмічне забезпечення програмного модуля створення субтитрів до мови жестів;

– реалізувати програмний модуль, що забезпечує переведення жестів у текстові субтитри з використанням комп’ютерного зору та машинного навчання;

– провести тестування роботи програмного модуля, зокрема оцінку точності розпізнавання.

Об’єктом дослідження є процес автоматизованого розпізнавання та інтерпретації жестової мови за допомогою засобів комп’ютерного зору та методів машинного навчання.

Предметом дослідження є методи комп’ютерного зору та машинного навчання, що застосовуються для розпізнавання жестової мови та її трансформації у текстову форму в режимі реального часу.

Розроблений програмний модуль можуть бути використані у широкому колі практичних застосувань, пов’язаних з підвищенням доступності інформаційних технологій для осіб із порушеннями слуху. Зокрема, проєктований програмний модуль може бути інтегрований у системи автоматичного субтитрування для онлайн-комунікації, відеоконференцій, освітніх та інших цифрових платформах для забезпечення іклюзивного та безбар’єрного простору спілкування.

Результати дослідження опубліковано в матеріалах 4 Міжнародної науково-практичної конференції «Розвиток науки: теорії, відкриття та практичні результати», Цюріх, Швейцарія, 9-11 червня 2025 р. (додаток Г). Участь в конференцій підтверджує сертифікат (додаток В).

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ ПРОЄКТУВАННЯ

1.1 Опис предметної області

У сучасному світі інклюзивність стає не просто бажаною, а необхідною умовою повноцінного функціонування суспільства. Однією з ключових проблем на цьому шляху є мовний бар'єр, що виникає при спілкуванні між людьми з порушеннями слуху, які використовують жестову мову, та тими, хто нею не володіє. Жестова мова, будучи повноцінною системою комунікації зі своєю граматикою та лексикою, часто залишається незрозумілою для широкого загалу, що призводить до соціальної ізоляції, обмеженого доступу до інформації, освіти, працевлаштування та інших важливих сфер життєдіяльності осіб з вадами слуху.

У світі існує понад 200 різних жестових мов, які є рідними мовами понад 70 мільйонів осіб з порушеннями слуху [2]. Для багатьох з них жестова мова є рідною, особливо для дітей, що зростають у сім’ях глухих. Жестова мова є також важливою складовою культурної та мовної ідентичності національних спільнот глухих.

Є кальковане жестове мовлення (перекладається буквально кожне слово тексту з одночасним промовлянням), є літературна жестова мова (переклад за контекстом значення термінів з одночасним промовлянням) і розмовна жестова мова, якою люди з вадами слуху користуються при спілкуванні в побуті (один жест може означати словосполучення, речення, промовляють їх при цьому не завжди, навіть із закритими вустами) [12].

Жестова мова включає не лише рухи рук, але й міміку, положення тіла та навіть напрям погляду. Вона є повноцінною мовою зі своєю граматикою, синтаксисом і словниковим запасом. Діти з сімей глухих, як правило, вивчають жестову мову так само природно, як інші діти вивчають розмовну мову.

Попри те, що жестова мова є самодостатньою та ефективною формою спілкування, бар’єри у взаєморозумінні між людьми, які її використовують, та тими, хто не має відповідної підготовки, залишаються значними. Навіть досвідчені перекладачі жестової мови можуть стикатися з труднощами у тлумаченні окремих

жестів або варіантів їх використання. Таким чином, постає актуальне завдання – створення технічних засобів, здатних забезпечити автоматичне або напівавтоматичне розпізнавання жестів і переклад їх у текстову та усну форму.

Застосування сучасних інформаційних технологій комп'ютерного зору у поєднанні з машинним навчанням відкриває нові можливості у сфері автоматизованого перекладу жестової мови. Комп’ютерний зір дозволяє здійснювати захоплення, обробку та аналіз візуальної інформації, такої як положення рук, пальців, міміка обличчя та інші невербальні сигнали, за допомогою відео- або зображень у реальному часі. Такі можливості особливо важливі в умовах потреби у швидкому й точному розпізнаванні жестів у динамічному середовищі, де жести не лише статичні, а й мають складну послідовну природу [19].

Машинне навчання, зокрема глибокі нейронні мережі, використовується для класифікації та інтерпретації цих візуальних даних, що дозволяє системам адаптуватися до різноманітних варіацій жестів та контекстів. Такий підхід забезпечує гнучкість та масштабованість моделей, дозволяючи ефективно навчатися на основі обмежених або аугментованих наборів даних. Особливої популярності набули моделі на базі згорткових (Convolutional Neural Networks, CNN) та рекурентних (Recurrent Neural Networks, RNN) архітектур, які здатні розпізнавати як прості одиничні жести, так і складні фрази з урахуванням часової динаміки руху [20].

Інтеграція комп’ютерного зору з нейромережевими підходами дозволяє побудувати адаптивні системи, здатні самостійно покращувати точність розпізнавання шляхом самонавчання, а також забезпечує крос-платформену реалізацію – від мобільних застосунків до високопродуктивних десктопних рішень [21]. Крім того, системи комп’ютерного зору дають змогу проводити попередню обробку вхідного відео: нормалізацію освітлення, виділення контурів рук, зменшення шуму, що суттєво підвищує якість класифікації навіть в умовах складного фону [22].

В результаті, стає можливим розроблення інтелектуальних програмних модулів, здатних здійснювати трансформацію жестової мови у текстову або

звукову форму із затримкою в лічені мілісекунди. Такі системи можуть бути інтегровані у різноманітні сфери: освітні платформи, онлайн-сервіси, служби підтримки користувачів, медіа-продукцію, що дозволить зробити інформацію доступною для ширшого кола людей.

На сьогодні існують деякі рішення з розпізнавання рук, жестів, але основний на даний момент переклад все ще залежить від сім'ї особи з вадами слуху, спеціалізованих перекладачів або ручного вводу субтитрів. Є небагато додатків і програмних рішень з перекладу тексту у жестову форму, але прикладів додатків перекладу жестової мови в текстову форму ще менше.

Додатковою перешкодою для розробки якісних моделей є обмежена доступність корпусів анотованих відео з жестовою мовою. У більшості випадків існуючі дані є фрагментарними, неуніфікованими або захищеними авторськими правами, що ускладнює навчання глибоких нейронних мереж для розпізнавання жестів.

Окрім цього, важливо враховувати регіональні та діалектні відмінності у жестовій мові, які можуть суттєво впливати на точність автоматичного перекладу. У зв'язку з цим необхідно розробляти адаптивні системи, здатні налаштовуватись на конкретну жестову мову або навіть її діалект.

Попри суттєві науково-технічні досягнення, проблема автоматизованого перекладу жестової мови залишається відкритою та багатоаспектною. Існують різні методи перекладу: живий перекладач, технології та програми перекладачі. Одним із перспективних напрямів роботи є автоматизоване створення субтитрів мови жестів, оскільки це не лише технологічний виклик, але й важливий крок до створення більш інклюзивного та доступного суспільства.

Існують пристрої для перекладу мови жестів у текст. Наприклад, Enable Talk, який являє собою рукавички зі спеціальним пристроєм, а також гнучкими сенсорами, які відстежують згинання пальців і рух руки в просторі. Ця інформація передається на смартфон і обробляється на ньому, після чого зроблені жести озвучуються голосом [10].

Схожа ідея роботи є в рукавичок SignAloud, але на відміну від попереднього

прикладу, вони за допомогою Bluetooth передають дані на комп'ютер, який знаходить відповідності в базі жестів, після чого знайдене слово транслюється через вбудовані в рукавички динаміки [11].

Існують додатки, де текст автоматично перекладається у жестову мову. Цей процес здійснюється за допомогою спеціальних алгоритмів і технологій, які забезпечують візуальне представлення жестів. Результати перекладу можуть бути відображені у вигляді графічних зображень або тривимірних моделей людей, які виконують жестову мову. Такі системи застосовуються переважно в освітньому середовищі або для підвищення доступності контенту в Інтернеті. Найчастіше це односторонні перекладачі з тексту в жести, тоді як зворотний переклад із жестів у текст залишається значно складнішим завданням.

Також створюються програми, де в інтерактивній та ігровій формі відкривають нові можливості для навчання, поєднуючи ефективність і захопливість процесу.

Попри це, кількість рішень, які здійснюють переклад із жестової мови в текстову форму в режимі реального часу, залишається обмеженою. Більшість із них знаходяться на стадії експериментальних розробок або демонстраційних прототипів. Усе ще переважна частина перекладу залежить від участі перекладачів або ручного введення субтитрів, що істотно обмежує масштабованість та оперативність процесу спілкування. Це, актуалізує необхідність у створенні автоматизованих рішень, здатних забезпечити точне, швидке та безперервне розпізнавання жестової мови в реальному часі.

Отже, проблема автоматичного розпізнавання жестової мови набуває не лише технічного, а й соціального виміру. Створення програмного модуля, здатного розпізнавати жестову мову та автоматично створювати текстові субтитри, є не лише актуальним напрямом наукових досліджень, але й важливим практичним завданням. Такий підхід сприятиме зменшенню комунікаційного бар’єру між людьми з порушеннями слуху та іншими членами суспільства, розширить доступ до інформації, освіти та послуг, а також посилить інтеграцію людей з інвалідністю в усі сфери суспільного життя.

1.2 Огляд і аналіз існуючих рішень

Розробка рішень для полегшення комунікації людей з вадами слуху набуває все більшої уваги. У зв'язку з цим активізуються дослідження у сфері комп'ютерного зору, обробки природної мови та інтерактивних технологій, спрямовані на розпізнавання жестів і автоматичний переклад жестової мови. Сьогодні на ринку представлені різноманітні програмні та апаратні засоби, які вирішують завдання перекладу мовлення вербального в жести і зворотньо.

Було розглянуто наступні цифрові продукти перекладу жестової мови:

- Hand Talk;
- English to Sign Language Translator;
- Sign Translate;
- SignAll.

Hand Talk – безкоштовний мобільний додаток, який забезпечує переклад текстових та голосових даних на американську (ASL) та бразильську (LIBRAS) жестові мови. Дозволяє перекладати текстові та голосові дані в обрану мову жестів, за допомогою анімованого 3D-персонажа, що відтворює відповідні жести. Додаток також пропонує навчальні матеріали для вивчення жестових мов. Інтерфейс додатку представлено на рисунку 1.1.

Даний додаток не надає перекладу жестової мови на текст але дає можливості людям без вад слуху комунікувати з глухими людьми або навіть вивчати мову, що дає можливість збагачення і пізнання нового, а також зменшувати бар'єр в спілкуванні.

Застосунок працює на телефоні, що забезпечує його мобільність та доступність роблять його корисним інструментом для осіб, які бажають вивчати жестову мову або спілкуватися з людьми з порушеннями слуху.

English to Sign Language Translator від wecapable перекладає текст в жести за допомогою дактильної абетки, відображаючи кожну літеру окремо у вигляді відповідного жесту. Працює з американською та британською мовою жестів. Є також можливість вивчення літер названих жестових мов, а запропонований метод

перекладу дозволяє побачити значення кожного знаку та сприяє засвоєнню основ жестової абетки.

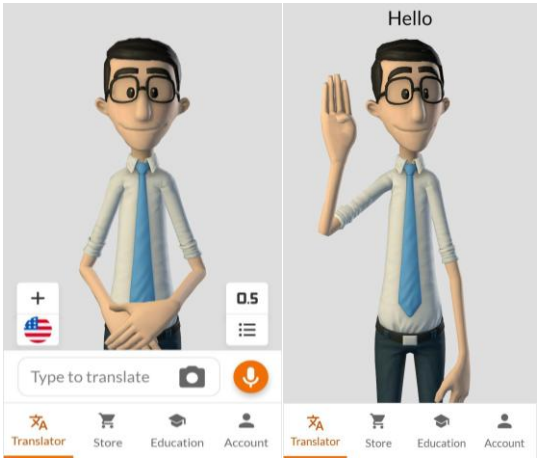

Рисунок 1.1 – Інтерфейс додатку Hand Talk

Хоча такий підхід не забезпечує повноцінного перекладу слів або речень, він є ефективним засобом для вивчення дактильної абетки та базових елементів жестової мови. Від попереднього рішення відрізняє те, що це веб-застосунок і з ним можна працювати як на телефоні, так і на комп'ютері. Робота даного інструменту представлена на рисунку 1.2.

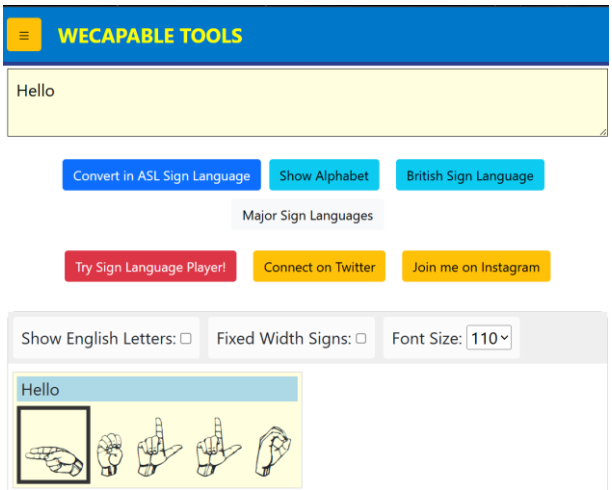

Рисунок 1.2 – Інтерфейс English to Sign Language Translator

Sign Translate [3] платформа перекладу мови жестів, яка використовує штучний інтелект для трансформації тексту та звуку у жестову мову за допомогою

анімованого 3D-персонажу. Підтримує 40 жестових та вербальних мов. Хоча заявлено про можливість перекладу жестів у текстовий та звуковий формат, ця функція наразі перебуває на стадії розробки та потребує подальшого вдосконалення.

Даний застосунок продовжує вдосконалюватися та збільшувати свої можливості, а також кількість підтримуваних мов. На рисунках 1.3 – 1.4 зображено роботу даної платформи.

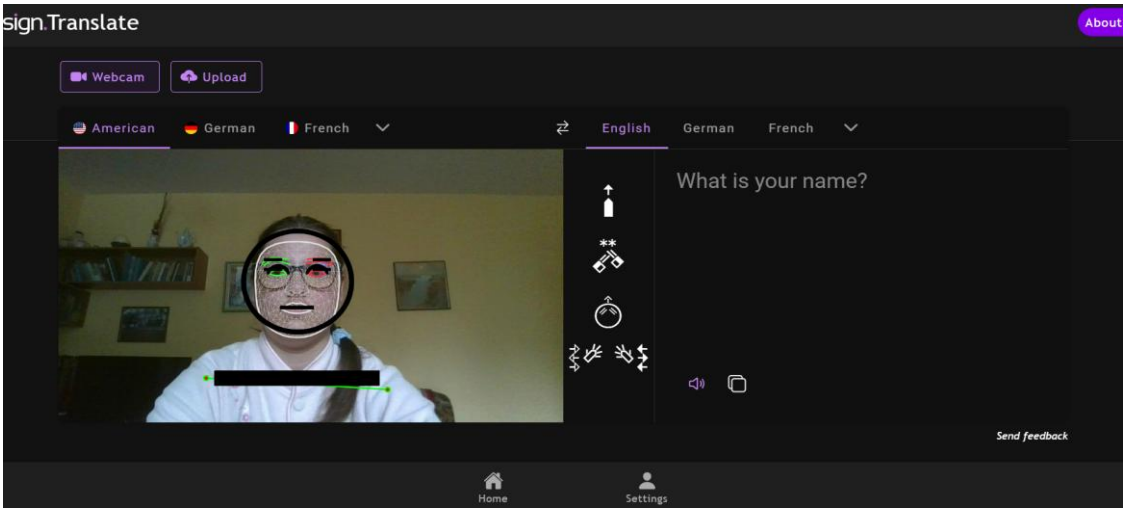

Рисунок 1.3 – Інтерфейс Sign Translate для розпізнавання мови жестів з вебкамери та перетворення її на текст

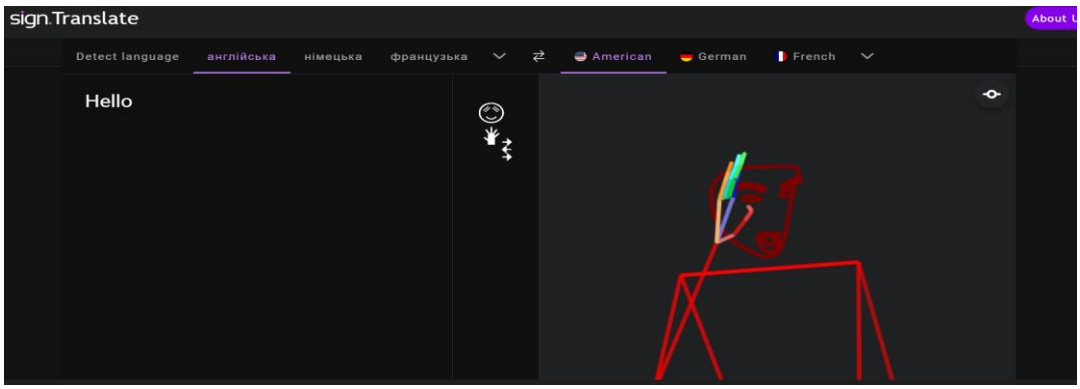

Рисунок 1.4 – Інтерфейс Sign Translate для перетворення тексту на мову жестів з візуальним відтворенням перекладу

SignAll платний автоматизований пристрій жестової мови [14]. Технологія використовує кілька камер та сенсорів глибини для відстеження рухів рук, міміки

та положення тіла. Зібрані дані обробляються за допомогою алгоритмів машинного навчання, які розпізнають жести та перетворюють їх у текст. Система враховує тривимірну природу жестової мови та її складну граматику. Проте висока вартість та необхідність спеціалізованого обладнання обмежують її широке впровадження.

Розглянуті існуючі рішення дають можливість комунікації, але в кожного є певні недоліки: Hand Talk та English to Sign Language Translator пропонують лише переклад на жестову мову, але не навпаки. Sign Translate має можливості перекладу тексту в жести та жестів у текст, але друга функція потребує подальшого розвитку. SignAll дає можливість переведення жестів у текст, але дане рішення є дороговартісним і потребує багато додаткових інструментів.

Таким чином, аналіз існуючих рішень демонструє значний прогрес у розвитку технологій перекладу жестової мови. Проте більшість з них мають певні обмеження: відсутність зворотного перекладу, обмежена підтримка мов, висока вартість або потреба у спеціалізованому обладнанні. Це підкреслює необхідність подальших досліджень та розробок у цій галузі для створення більш доступних та універсальних рішень, що сприятимуть інклюзивності та ефективній комунікації між людьми з порушеннями слуху та широким загалом.

1.3 Постановка завдання проєктування

Основною задачею, яку розглядає це дослідження є мовний бар’єр при комунікації з людьми, які спілкуються мовою жестів. Звичайні методи спілкування включають наявність живого перекладача, що не завжди доступний варіант, або всі люди розуміють потрібну мову жестів, чого неможливо гарантувати, враховуючи різноманіття жестових мов та обмеженість поширення відповідної освітньої бази серед населення. Існують додатки для перекладу вербальних мов у жестову, але мало шляхів перекладу жестів у вербальний формат.

У сучасному суспільстві простежується недостатній рівень технічної підтримки жестової мови, зокрема у сфері розпізнавання жестів та їхнього перекладу у вербальний (текстовий або голосовий) формат. Наявні цифрові

рішення переведення жестів у текст досі залишається обмеженим у функціональності та доступності. Це створює додаткові бар'єри у професійній, освітній, медійній та побутовій сферах для людей із порушеннями слуху.

Розробка програмного модуля, що поєднує технології комп'ютерного зору та машинного навчання, має на меті зробити інформацію більш доступною для людей із порушеннями слуху, а також сприяти ширшому прийняттю мови жестів у суспільстві. Використання алгоритмів машинного навчання дозволяє ефективно розпізнавати та аналізувати жести, перетворюючи їх у текстовий формат. Це рішення може бути інтегроване в системи перекладу, освітні платформи, мультимедійний контент та багато інших сфер.

Особливе значення має створення універсального, масштабованого та доступного інструменту, здатного працювати в реальному часі та підтримувати адаптацію під різні мови жестів. Такий підхід відповідає потребам інклюзивної освіти, цифрової трансформації та розбудови відкритого інформаційного простору.

Отже, метою цього дослідження є створення програмного модуля який, за допомогою машинного навчання та засобів комп’ютерного зору, буде розпізнавати жести та створювати переклад сказаного ними у текстовий формат.

Для досягнення даної мети потрібно вирішити наступні завдання:

– проаналізувати предметну область розпізнавання жестової мови, дослідити сучасні підходи на основі комп’ютерного зору та машинного навчання, визначити їх переваги та обмеження;

– сформувати концепцію побудови програмного модуля, що дозволяє розпізнавати жестів у реальному часі з подальшим створенням субтитрів;

– спроектувати структуру та алгоритмічне забезпечення програмного модуля створення субтитрів до мови жестів;

– реалізувати програмний модуль, що забезпечує переведення жестів у текстові субтитри з використанням комп’ютерного зору та машинного навчання;

– провести тестування роботи програмного модуля, зокрема оцінку точності розпізнавання.

Результатом повинне стати створення програмного модуля, який буде

здатний розпізнавати жести мовця та створювати переклад мови жестів у текстовий формат. Також важливим є забезпечення роботи модуля в реальному часі та виведення субтитрів у заданому інтерфейсі. Це дозволить зменшити мовний бар'єр та створити більш доступне та інклюзивне середовище для спілкування з людьми з вадами слуху, а також, розширені можливості для їхнього соціального та професійного росту.

Системи автоматичного створення субтитрів на основі жестової мови можуть бути інтегровані у різноманітні інформаційні та навчальні сервіси: онлайн-курси, відеолекції, інструктажі, телевізійні передачі тощо. Це сприятиме кращій інтеграції людей з порушеннями слуху у цифровий простір та доступу до інформації, що транслюється жестовим способом.

2 ПРОЄКТУВАННЯ ПРОГРАМНОГО МОДУЛЯ СТВОРЕННЯ СУБТИТРІВ МОВИ ЖЕСТІВ

2.1 Загальна структура проєктованого модуля

Метою побудови модуля, що переводить жести в субтитри є забезпечення доступності комунікації для осіб з порушенням слуху шляхом інтеграції засобів комп'ютерного зору та машинного навчання. Реалізація даного підходу дозволяє розпізнавати жести користувача в реальному часі та перетворювати їх у текстову форму, придатну для сприйняття.

Для представлення загальної структури програмного модуля використано діаграму, де представлено взаємозв'язки між функціями системи, вхідними та вихідними даними, засобами реалізації (механізмами) та елементами контролюдані що входять та виходять, а також механізми і контроль, що впливають на роботу модуля (рисунок 2.1).

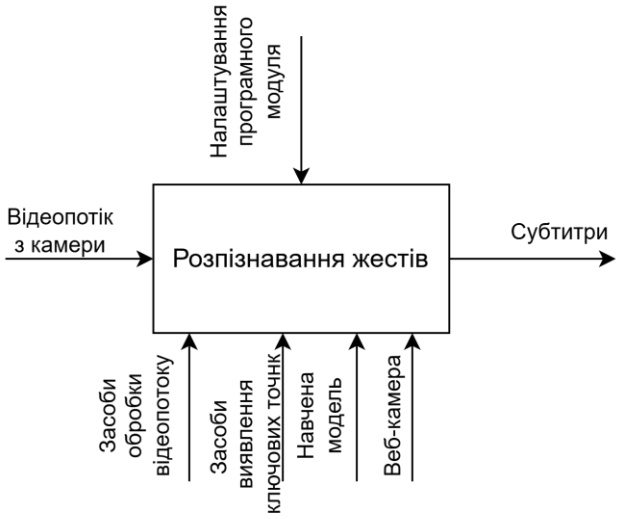

Рисунок 2.1 – IDEF0-діаграма для програмного модуля

"Розпізнавання жестів" – центральний блок, який виконує основну функцію розпізнавання жестів та формування субтитрів у реальному часі. На вході система отримує відеопотік, що надходить із камери, а на виході формуються субтитри, які можуть бути виведені на екран або збережені.

На всі процеси впливають налаштування програмного модуля, що є контролем роботи, оскільки включає налаштування параметрів обробки відео, гіперпараметри моделі, правила сегментації жестів, інтервал часу для обробки тощо.

З використаних механізмів ключовими є:

- веб-камера користувача як джерело відеопотоку;
- засоби обробки відеопотоку, наприклад OpenCV;
- засоби виявлення ключових точок, наприклад MediaPipe;
- навчена модель.

На рисунку 2.2 зображено декомпозицію діаграми “Розпізнавання жестів” з рисунку 2.1.

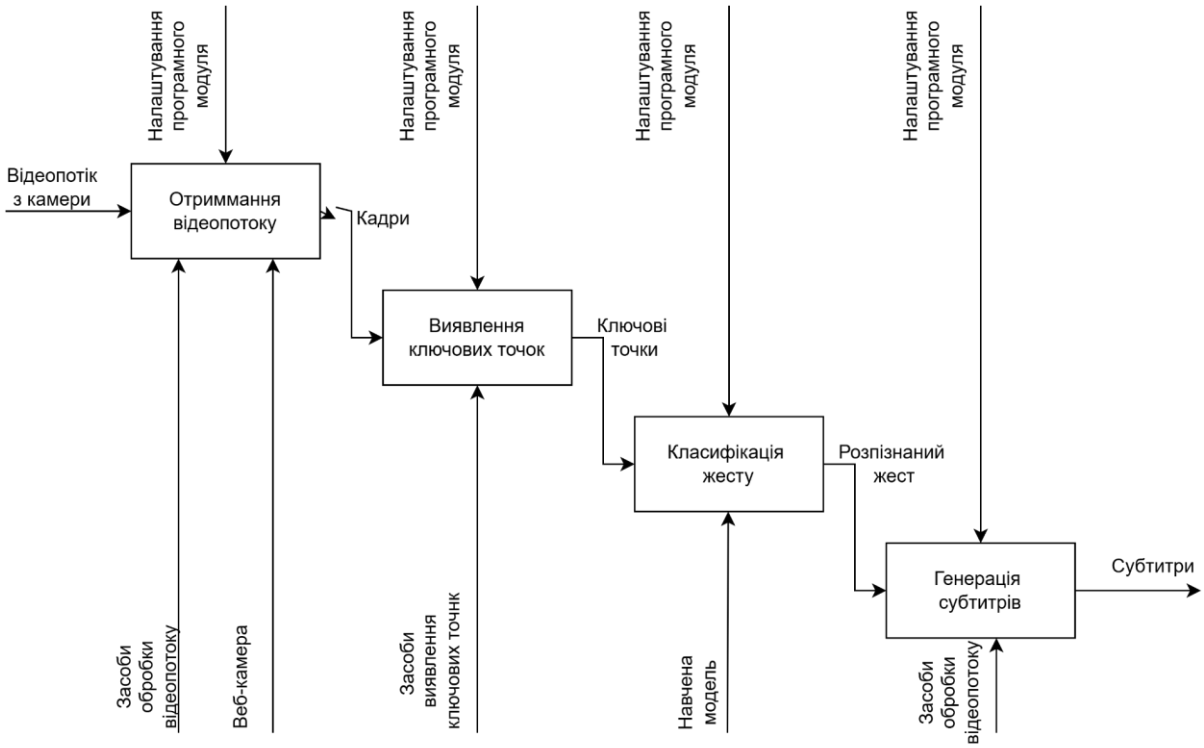

Рисунок 2.2 – Декомпозиція “Розпізнавання жестів” програмного модуля

У ході декомпозиції було виділено наступні процеси:

а) отримання відеопотоку, де виконується захоплення відео з камери користувача та його попередня обробка для подальшого аналізу. Забезпечується стабільність частоти кадрів та нормалізація зображення;

б) виявлення ключових точок, де на основі кадрів, отриманих у попередньому блоці, за допомогою певних засобів виявляються ключові точки руки (та, за потреби, інших частин тіла), що описують її положення в просторі;

в) класифікація жесту, де послідовність координат ключових точок подається на вхід навченої моделі, яка аналізує часовий контекст та визначає, який саме жест був здійснений користувачем;

г) генерація субтитрів, де визначений жест інтерпретується як відповідна текстова фраза, яка оформлюється у вигляді субтитру та передається до інтерфейсу користувача.

Інформаційні потоки між функціями є послідовними та відображають логіку обробки вхідного відеосигналу: від захоплення – до розпізнавання та виводу результату. Усі підфункції керуються спільними параметрами контролю (налаштуваннями модуля) та спільно використовують однакові механізми (обладнання й бібліотеки).

Отже, сформована функціональна модель забезпечує логічну й структурну основу для розробки та реалізації програмного модуля, дозволяючи уніфікувати процеси, мінімізувати похибки на етапах обробки даних та забезпечити цілісність системної архітектури.

2.2 Алгоритмічне забезпечення

Розробка програмного модуля, здатного розпізнавати мову жестів на основі комп’ютерного зору та машинного навчання і створювати їх переклад в текстовому форматі, дозволяє розв’язати проблему комунікації між людьми, що спілкуються вербально і жестами. Це рішення дозволяє людині, яка не розуміє жестів, отримати переклад сказаного, використовуючи алгоритми комп'ютерного зору та машинного навчання для аналізу отриманих зображень у реальному часі.

Схема алгоритму роботи програмного модуля створення субтитрів мови жестів з використанням комп’ютерного зору та машинного навчання представлена на рисунку 2.3.

Рисунок 2.3 – Схема алгоритму роботи програмного модуля переведення жестової мови в субтитри

Із схеми алгоритму роботи програмного модуля (див. рисунок 2.3) можна побачити таку послідовність дій роботи застосунку:

а) обробка відеозображення, для пошуку рук;

б) розпізнавання жестів за допомогою навченої моделі;

в) перетворення жесту в текст, для створення субтитрів;

г) вивід перекладу у вигляді тексту або субтитрів.

Розглянемо детальніше крок обробки відеозображення. На рисунку 2.4 зображено детальний алгоритм обробки відеозображення.

Рисунок 2.4 – Схема алгоритму обробки відеозображення

При виконанні обробки відеозображення виконуються наступні кроки:

а) захоплення кадру з потоку камери;

б) виявлення рук;

в)сегментація і обробка результату, де витягуються координати для наступного етапу – розпізнавання жестів.

Для розпізнавання жестів виконується наступний алгоритм дій (рисунок 2.5):

а) вилучення ознак, де на основі ключових точок обчислюються ознаки, що описують жест;

б) класифікація жестів за допомогою навченої моделі;

в) визначення відповідного жесту.

Рисунок 2.5 – Схема алгоритму розпізнавання жестів

Алгоритм перетворення жесту в текст представлено на рисунку 2.6.

Рисунок 2.6 – Схема алгоритму перетворення жесту в текст

Нижче розглянуто представлені кроки для перетворення жесту в текст (рисунок 2.6):

а) пошук відповідного жестового еквівалента, де підбирається відповідне слово для визначення жесту;

б) формування тексту, де текстові відповідники жесту об’єднуються в рядок, який представляє собою переклад.

Підсумовуючи, ідея програмного модуля полягає в тому, що при розмові у відео форматі, технологія комп’ютерного зору буде обробляти зображення для розпізнавання рук і жестів, модель машинного навчання розпізнаватиме жест, потім програмний компонент переведе розпізнаний жест у текстовий формат, що буде відображатися іншій стороні розмови, яка потребуватиме даного перекладу.

2.3 Інструменти комп’ютерного зору, машинного навчання та апаратне забезпечення, що застосовуються у задачах перекладу жестової мови у текст

Задача перекладу жестової мови у текст є багаторівневою і передбачає інтеграцію апаратних та програмних компонентів. Процес включає захоплення візуальних сигналів, їхню обробку, аналіз та перетворення у текстову форму. Ключовими у цьому процесі є засоби комп’ютерного зору, методи машинного навчання та відповідне технічне забезпечення.

Комп'ютерний зір є міждисциплінарною галуззю, що поєднує методи цифрової обробки зображень, аналітичної геометрії, математики та інформатики з метою автоматичного отримання, аналізу та інтерпретації візуальної інформації з реального світу [13, 1]. Розвиток цієї галузі став можливим завдяки значному прогресу в обчислювальній техніці, алгоритмах машинного навчання та наявності великих масивів даних. У контексті перекладу жестової мови комп’ютерний зір використовується для виявлення та відстеження рухів рук, визначення конфігурації пальців, пози користувача, що надалі аналізуються засобами штучного інтелекту.

Основні завдання комп'ютерного зору включають сегментацію зображень, детектування контурів, відстеження об'єктів, реконструкцію сцени, а також

відновлення тривимірної структури об'єктів за серією двовимірних проєкцій [23]. У практичних задачах перекладу жестової мови комп’ютерний зір використовується на етапах попередньої обробки вхідного відео, нормалізації координат та підготовки ознак для моделі.

Серед найпоширеніших програмних бібліотек, які використовуються в розробці систем комп’ютерного зору виділяються такі бібліотеки OpenCV, TensorFlow, MediaPipe [28].

OpenCV (Open Source Computer Vision Library) – бібліотека комп’ютерного зору та машинного навчання з відкритим кодом. Бібліотека містить понад 2500 оптимізованих алгоритмів, що включає повний набір як класичних, так і найсучасніших алгоритмів комп’ютерного зору та машинного навчання [4]. У системах перекладу жестів вона використовується для попередньої обробки відео, зокрема: фільтрації, бінаризації, виявлення контурів та інтеграції з іншими програмними компонентами.

Дана бібліотека добре оптимізована для роботи в режимі реального часу, що важливо в задачі розпізнавання жестів. Також вона легко інтегрується з іншими бібліотеками, що дозволяє розширити можливості системи [9].

TensorFlow це – потужний інструмент для створення і навчання різноманітних моделей, починаючи від простих лінійних регресій і закінчуючи складними нейронними мережами. Використовується в широкому спектрі застосунків, від розпізнавання зображень і мови до автономної їзди. Це відкрита програмна бібліотека для машинного навчання [7].

Фреймворк MediaPipe дозволяє розробнику брати вже написані іншими програмістами базові алгоритми комп’ютерного зору і простим чином об’єднувати їх у потрібний йому конвеєр. Крім того, фреймворк спочатку створювався з розрахунком на те, що програми повинні без проблем розгортатися на всі основні платформи, без необхідності в адаптації під кожну з них [5]. MediaPipe пропонує рішення для рук, поз і навіть цілісного відстеження [6]. Перевагою є кросплатформність, оптимізація під мобільні пристрої та інтеграція з TensorFlow.

TensorFlow підтримує обчислення на GPU, що дозволяє значно прискорити

навчання моделей і є важливим при роботі з великими наборами даних. Легко поєднується з з іншими бібліотеками, такими як OpenCV або MediaPipe, для створення комплексних рішень.

Ефективність і точність моделей машинного навчання значною мірою залежать від якості та обсягу навчального набору даних. У задачах перекладу жестової мови зазвичай використовуються спеціалізовані набори даних, що містять відеозаписи або послідовності зображень жестів у поєднанні з анотаціями. Серед найбільш поширених можна виокремити American Sign Language (ASL) Dataset, RWTH-PHOENIX-Weather, LSA64, а також користувацькі набори даних, сформовані за допомогою камер з високою роздільною здатністю.

Для роботи над заданим модулем використано набір даних American Sign Language Dataset [8]. Цей набір містить зображення рук, які відображають літери американської абетки жестової мови, а також цифри від 0 до 9. Для кожного символу подано 70 зразків зображень, що дає змогу ефективно навчати модель класифікації. Приклади зображень, поданих у цьому наборі даних, наведено на рисунку 2.7.

Рисунок 2.7 – Приклад зображень у American Sign Language Dataset

Для навчання моделі розпізнавання жестів було розглянуто рекурентну нейронну мережу типу LSTM (Long Short-Term Memory), що добре зарекомендувала себе у роботі з послідовнісними даними, зокрема такими, що характеризуються часовою динамікою. LSTM-мережі є модифікацією класичних рекурентних нейронних мереж (RNN), яка дозволяє зберігати довготривалі

залежності у даних шляхом використання спеціальних блоків пам’яті з механізмом «затворів» (gates), що регулюють потік інформації через нейрон.

Архітектура LSTM включає чотири основні компоненти: вхідний вентиль (input gate), забувальний вентиль (forget gate), кандидат на оновлення пам’яті (candidate memory) та вихідний вентиль(output gate), які відповідають за оновлення та збереження інформації в комірках пам’яті (рисунок 2.8).

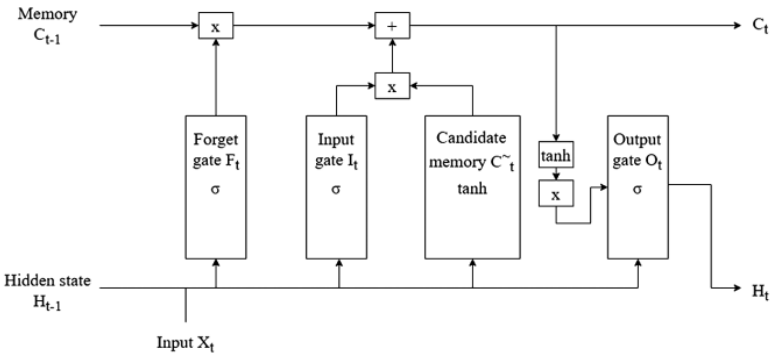

Рисунок 2.8 – Архітектура LSTM [27]

Забувальний вентиль визначають, яку частину попередньої інформації слід зберегти, а яку відкинути. Вхідний вентиль контролюють, яка нова інформація буде додана до стану пам’яті. Кандидат на оновлення пам’яті генерує нові дані, які можуть бути записані у пам’ять. Вихідний вентиль відповідає за те, яка частина оновленого стану пам’яті буде передана далі як вихід.

Такий підхід дозволяє ефективно обробляти послідовності ознак, отримані на основі жестів у відео, зокрема координати ключових точок кисті руки, що генеруються за допомогою бібліотеки MediaPipe.

Завдяки здатності LSTM-моделі враховувати контекст попередніх кадрів, підвищується точність розпізнавання динамічних жестів, які складаються з кількох послідовних рухів [25; 26].

Якість перекладу жестів значною мірою залежить від характеристик апаратного забезпечення. Особливо важливими є параметри камер, які забезпечують вхідний відеопотік. Для розпізнавання алфавіту жестової мови достатньо звичайної HD-камери з частотою 30 FPS вбудованої в ноутбук. Для

складніших динамічних жестів рекомендується використовувати камери з частотою 60 FPS або глибинні камери, наприклад, Intel RealSense чи Microsoft Kinect [24]

Також, для обробки великих обсягів зображень і тренування глибоких нейронних мереж важливо забезпечити відповідну обчислювальну потужність системи. Для цього використовують високопродуктивні графічні процесори (GPU), такі як NVIDIA RTX або Tesla, а також спеціалізовані прискорювачі на базі TPU (Tensor Processing Unit) чи NPU (Neural Processing Unit), які оптимізовані для задач штучного інтелекту.

Таким чином, ефективна реалізація програмного модуля перекладу жестової мови у текстову форму потребує комплексного підходу, що поєднує сучасні програмні засоби комп’ютерного зору, потужні інструменти машинного навчання та відповідне апаратне забезпечення. Від узгодженості цих компонентів залежить не лише продуктивність і стабільність системи, а й точність розпізнавання жестів у реальному часі, що є критично важливим для забезпечення якісної комунікації між людьми з порушеннями слуху та іншими учасниками соціальної взаємодії.

3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО МОДУЛЯ СТВОРЕННЯ СУБТИТРІВ МОВИ ЖЕСТІВ З ВИКОРИСТАННЯМ КОМП'ЮТЕРНОГО ЗОРУ І МАШИННОГО НАВЧАННЯ

3.1 Реалізація програмного забезпечення

Для роботи було обрано Python [15, 17, 18], оскільки в неї велика кількість застосувань у сферах комп’ютерного зору та машинного навчання. Дана мова вирізняється широким спектром бібліотек та інструментів, які спрощують розробку завдяки наявності численних готових рішень. Крім того, Python підтримується великою спільнотою розробників, що забезпечує оперативне вирішення проблем та наявність численних відкритих ресурсів і навчальних матеріалів.

Зокрема для побудови і навчання моделі корисні TensorFlow [7], а для роботи з комп’ютерним зором – OpenCV [9] та MediaPipe [6].TensorFlow забезпечує гнучкі можливості для створення як простих, так і складних архітектур глибинного навчання, зокрема рекурентних мереж, а MediaPipe надає модулі для попередньої обробки зображень з високою точністю виявлення ключових точок. Остання, зокрема, має готове рішення для розпізнавання рук в реальному часі.

Завдяки використанню зазначених бібліотек реалізацію системи можна умовно поділити на два основні етапи:

- навчання моделі розпізнавання жестів;

- компонент реального часу для розпізнавання жестів і виведення результатів у вигляді субтитрів.

Першим є навчання моделі, яке реалізує повний цикл підготовки даних і побудови нейронної мережі. На етапі попередньої обробки здійснюється автоматичне вилучення просторових координат (keypoints) для обох рук із заздалегідь підготовлених зображень за допомогою інструментів MediaPipe Holistic. Цей процес дозволяє перейти від роботи з «сирими» зображеннями до більш компактного та інформативного представлення, що суттєво знижує обчислювальну складність подальшого аналізу.

Вилучення координат реалізується через функцію, яка здійснює перевірку

наявності виявлених ключових точок для лівої та правої руки, після чого формує єдиний вектор ознак. Код цієї функції представлено на рисунку 3.1.

```
def extract_keypoints(results):
    pose = np.array([[res.x, res.y, res.z, res.visibility]
                     for res in results.pose_landmarks.landmark]).flatten()
                     if results.pose_landmarks else np.zeros(33*4)
    face = np.array([[res.x, res.y, res.z]
                     for res in results.face_landmarks.landmark]).flatten()
                     if results.face_landmarks else np.zeros(468*3)
    lh = np.array([[res.x, res.y, res.z]
                   for res in results.left_hand_landmarks.landmark]).flatten()
                   if results.left_hand_landmarks else np.zeros(21*3)
    rh = np.array([[res.x, res.y, res.z]
                   for res in results.right_hand_landmarks.landmark]).flatten()
                   if results.right_hand_landmarks else np.zeros(21*3)
    return np.concatenate([pose, face, lh, rh])
```

Рисунок 3.1 – Код вилучення координат

Розпізнавання зображення руки та її положення у просторі реалізується з використанням бібліотек OpenCV (для обробки відеопотоку) та MediaPipe (для виявлення ключових точок тіла й рук). Отримані координати ключових точок слугують вхідними даними для класифікаційної моделі, що інтерпретує жести згідно з наперед визначеним словником символів. Результати роботи моделі використовуються для формування текстових субтитрів відповідно до розпізнаних жестів, що дає змогу автоматично перетворювати жести на зрозумілий текстовий формат. Такий підхід дозволяє виконувати не лише розпізнавання окремих знаків, але й забезпечує можливість формування послідовностей, які складаються у слова та речення.

Для навчання моделі вибрано американську жестову мову, зокрема її алфавіт, що складається з 26 літер (a–z) та 10 цифр (0–9). Для цього було використано набір зображень, що містить відповідні пози рук, які відтворюють кожен із символів.

На основі обраної мови жестів та використаного набору даних побудовано словник, що закодовує в собі символи англійського алфавіту та арабських цифр, що представлене на рисунку 3.2. Кожному символу надається унікальне ціле значення, що дозволяє здійснювати навчання та передбачення в категоріальному форматі.

Побудова, конфігурація та навчання моделі здійснюються за допомогою бібліотеки TensorFlow, яка дозволяє створювати гнучкі багаторівневі рекурентні

нейронні мережі [16] (зокрема LSTM) для задач класифікації послідовностей ознак.

```
label_map = {
    'a': 0,  'b': 1,  'c': 2,  'd': 3,  'e': 4,
    'f': 5,  'g': 6,  'h': 7,  'i': 8,  'j': 9,
    'k': 10, 'l': 11, 'm': 12, 'n': 13, 'o': 14,
    'p': 15, 'q': 16, 'r': 17, 's': 18, 't': 19,
    'u': 20, 'v': 21, 'w': 22, 'x': 23, 'y': 24,
    'z': 25,
    '0': 26, '1': 27, '2': 28, '3': 29, '4': 30,
    '5': 31, '6': 32, '7': 33, '8': 34, '9': 35
}
```

Рисунок 3.2 – Словник символів, використаних для моделі

На основі сформованого набору ознак будується модель LSTM . Архітектура моделі включає три послідовні LSTM-шари з різною кількістю нейронів, шари Dropout для регуляризації, а також фінальний повнозв'язний шар із функцією активації softmax, що дозволяє здійснювати багатокласову класифікацію 36 жестів (латинські літери та цифри).

Застосування саме рекурентної архітектури є обґрунтованим, оскільки LSTM-мережі здатні ефективно працювати з часовими послідовностями та зберігати інформацію про попередні стани, що є критичним при розпізнаванні жестів, які розгортаються в часі.

На рисунку 3.3 зображено фрагмент коду, який відповідає за створення, конфігурацію та тренування нейронної мережі на основі LSTM.

Після завершення навчання модель зберігається у форматі .h5. Це зроблено з метою використання програмного модулю без необхідності постійного перенавчання моделі. Збереження моделі у зазначеному форматі забезпечує її швидке завантаження при подальшому розгортанні та інтеграції у систему реального часу. Повний програмний код навчання моделі розпізнавати символ жестової мови знаходиться в додатку А.

Наступним етапом є обробка відеопотоку з камери в реальному часі та відображення результатів на екрані. На кожному кадрі здійснюється передобробка зображення, після чого виділяються ключові точки, аналогічно процедурі, застосованій на етапі навчання. Послідовність із 30 останніх кадрів передається на вхід збереженої LSTM-моделі для передбачення поточного жесту. Такий підхід дає

змогу моделі враховувати динаміку жесту та підвищує точність розпізнавання в умовах реального часу.

```
def build_model():
    model = Sequential()
    model.add(LSTM(64, return_sequences=True, activation='relu',
                   input_shape=(30, 1662)))
    model.add(Dropout(0.2))
    model.add(LSTM(128, return_sequences=True, activation='relu'))
    model.add(Dropout(0.3))
    model.add(LSTM(64, return_sequences=False, activation='relu'))
    model.add(Dense(64, activation='relu'))
    model.add(Dropout(0.2))
    model.add(Dense(36, activation='softmax'))
    model.compile(optimizer='adam', loss='categorical_crossentropy',
                  metrics=['accuracy'])
    return model
```

Рисунок 3.3 – Код налаштування шарів моделі

Рисунок 3.4 демонструє реалізацію модуля, який відповідає за захоплення відео з камери, попередню обробку кадрів, витягнення ключових ознак та прогнозування значення жесту за допомогою попередньо навченої моделі.

Даний код реалізує розпізнавання жестів у реальному часі. Після захоплення кадру з камери, він обробляється за допомогою модулю MediaPipe для виявлення ключових точок рук. Далі, з отриманих координат формується послідовність ознак, яка передається на вхід моделі. Модель виконує прогноз, який візуалізується у вигляді підпису на екрані. Також реалізовано логіку збирання слів у речення з метою подальшого виводу субтитрів.

Результат передбачення відображається на екрані у вигляді текстового рядка з найвірогіднішими розпізнаними жестами. Для уникнення шуму вивід обмежується лише тими символами, які досягають порогового значення впевненості та не повторюють попередній розпізнаний символ. Таким чином, забезпечується як точність, так і читабельність результату розпізнавання.

Реалізоване програмне забезпечення поєднує засоби комп’ютерного зору, обробки відео та машинне навчання, що дозволяє ефективно вирішувати задачу розпізнавання жестів у реальному часі. Отримані результати демонструють

потенціал подальшого розвитку системи для підтримки інклюзивних технологій, зокрема автоматизованого перекладу жестової мови в текстову форму.

Програмний код розпізнавання жестів у реальному часі знаходиться в додатку Б.

```
cap = cv2.VideoCapture(0)
with mp_holistic.Holistic(min_detection_confidence=0.5,
                          min_tracking_confidence=0.5) as holistic:
    while cap.isOpened():
        ret, frame = cap.read()
        image, results = mediapipe_detection(frame, holistic)
        # print(results)
        keypoints = extract_keypoints(results)
        sequence.append(keypoints)
        sequence = sequence[-30:]
        if len(sequence) == 30:
            res = model.predict(np.expand_dims(sequence, axis=0))[0]
            print(actions[np.argmax(res)])
            if res[np.argmax(res)] > threshold:
                if len(sentence) > 0:
                    if actions[np.argmax(res)] != sentence[-1]:
                        sentence.append(actions[np.argmax(res)])
                else:
                    sentence.append(actions[np.argmax(res)])
            if len(sentence) > 5:
                sentence = sentence[-5:]
        cv2.rectangle(image, (0,0), (640, 40), (245, 117, 16), -1)
        cv2.putText(image, ' '.join(sentence), (3,30),
                       cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 255, 255), 2,
                       cv2.LINE_AA)
        cv2.imshow('OpenCV Feed', image)
        if cv2.waitKey(10) & 0xFF == ord('q'):
            break
    cap.release()
    cv2.destroyAllWindows()
```

Рисунок 3.4 – Фрагмент коду для захоплення відео та прогнозування жестів

3.2 Інтерфейс користувача

Інтерфейс представлений у даній реалізації, виконує демонстраційну функцію та не є остаточним зразком зовнішнього вигляду програмного продукту. Його основне призначення полягає в ілюстрації функціональних можливостей розробленого модуля, а не для оцінки дизайну або користувацького досвіду.

Для демонстрації роботи забезпечується обробка відеозображень у реальному часі та виведення відповідного текстового супроводу. На рисунку 3.5 представлено інтерфейс за допомогою заданої реалізації.

Слід зазначити, що наведений інтерфейс не є фінальною версією продукту, а

слугує лише ілюстративною реалізацією його функцій. Даний програмний модуль передбачається як додаток до вже існуючих сервісів. Передбачається, що програмний модуль буде інтегруватися до існуючих цифрових платформ, таких як відеохостинги (наприклад, YouTube) або сервіси відеозв'язку (Zoom, Google Meet).

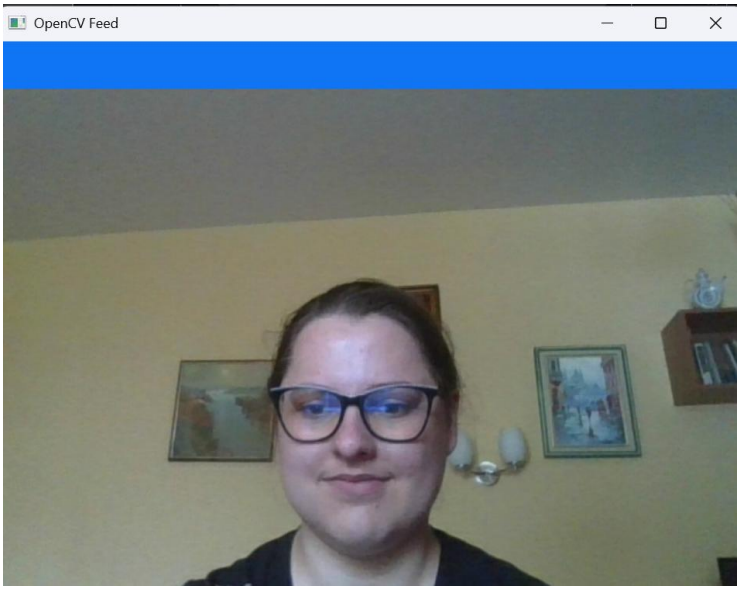

Рисунок 3.5 – Інтерфейс використаний у демонстраційній реалізації

Основна функціональність полягає у можливості увімкнення субтитрів, які автоматично генеруються у реальному часі на основі жестів мовця. Розпізнавання здійснюється за допомогою засобів комп'ютерного зору та алгоритмів машинного навчання, що аналізують положення руки та інтерпретують жести згідно з попередньо навченими моделями.

На рисунку 3.6 представлено можливий вигляд інтерфейсу програми Zoom з функцією субтитрування мови жестів. Така візуалізація імітує потенційну інтеграцію модуля у реальне середовище користувача, демонструючи можливість сумісності з популярними програмами відеозв'язку та сценаріями реального використання, наприклад, під час онлайн-лекцій, конференцій чи відеозустрічей.

Субтитри відображаються у заданому користувачем місці на екрані, що забезпечує гнучкість та зручність використання. Користувач має змогу керувати параметрами відображення субтитрів за допомогою вікна налаштувань, представленого на рисунку 3.7. Інтерфейс налаштувань є інтуїтивно зрозумілим і

дозволяє персоналізувати вигляд субтитрів відповідно до індивідуальних вподобань.

Рисунок 3.6 – Можливий варіант інтерфейсу користувача з функцією субтитрування

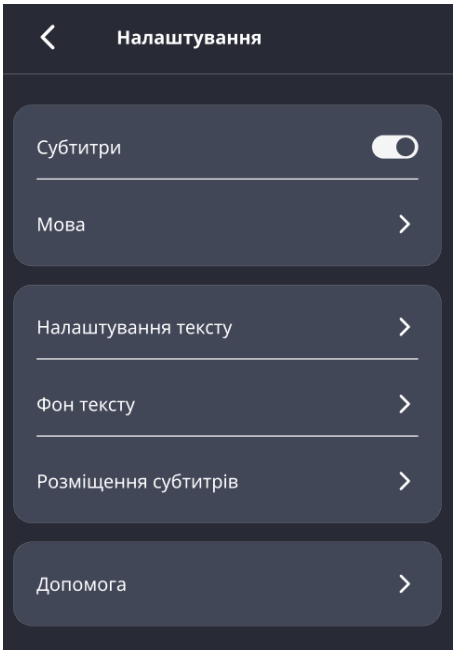

Рисунок 3.7 – Інтерфейс вікна налаштувань параметрів субтитрів

У даному вікні є наступні налаштування, що дозволяють здійснювати персоналізацію відображення субтитрів відповідно до індивідуальних потреб користувача, а саме:

- “Субтитри” – дозволяє вимкнути або увімкнути субтитри;
- “Мова” – передбачає вибір мови жестів та субтитрів;

- “Налаштування тесту” – вибір шрифту тексту, його розміру та кольору;
- “Фон тексту” – налаштування фону тексту, його кольору та прозорості;
- “Розміщення субтитрів” – дозволяє обрати положення субтитрів в залежності від уподобань користувача;
- “Допомога” – надає поради використання та рішення стосовно певних питань.

Таким чином, користувач отримує повний контроль над візуальними параметрами субтитрів, може адаптувати їх до власних потреб і забезпечити комфортне сприйняття інформації. Розроблений модуль орієнтований на інтеграцію з існуючими платформами, що розширює можливості доступу до відеоконтенту для осіб з порушеннями слуху та сприяє інклюзивності цифрового середовища.

3.3 Тестування модуля

Для перевірки працездатності розробленого програмного модуля було проведено тестування його основних функціональних компонентів. Тестування здійснювалося з метою визначення відповідності реалізації поставленим вимогам та оцінки роботи кожного з етапів обробки вхідних даних – від захоплення відео до виведення текстових субтитрів.

Тестування проводилося в умовах реального часу із застосуванням вебкамери, а також із використанням попередньо підготовлених жестів, які входили до навчального набору даних.

Перевірялися наступний функціонал:

- захоплення відео з відеокамери;
- розпізнавання рук і визначення їх ключових точок;
- виведення тексту;
- коректність виведеного перекладу.

Першим етапом тестування було перевірено працездатність захоплення відеопотоку. З рисунка 3.8 можна побачити, що модуль відображає відео, та

передає його для подальшої обробки. Отже, можна зробити висновок про функціональну придатність цього компонента.

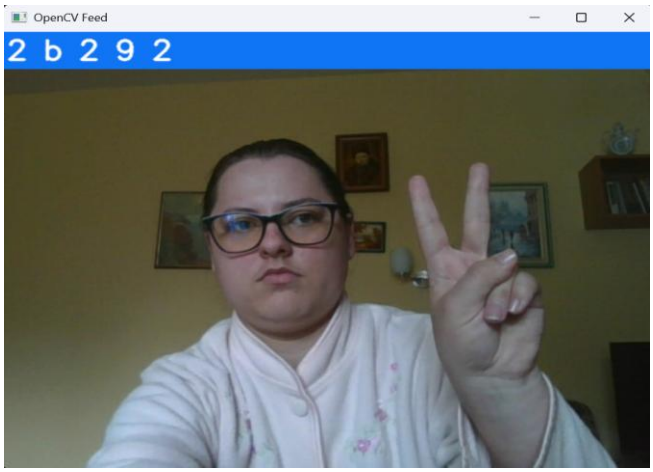

Рисунок 3.8 – Тестування роботи програмного модуля

Розпізнавання рук здійснюється на основі використання бібліотеки MediaPipe. Результати тестування засвідчили стабільну роботу модуля в частині визначення положення кистей рук у кадрі, незалежно від варіацій фону, освітлення та орієнтації руки відносно камери. Виявлення координат ключових точок відбувається з достатньою точністю для подальшого використання в процемі розпізнавання жестів. На рисунку 3.9 зображено визначені ключові точки руки в реальному часі.

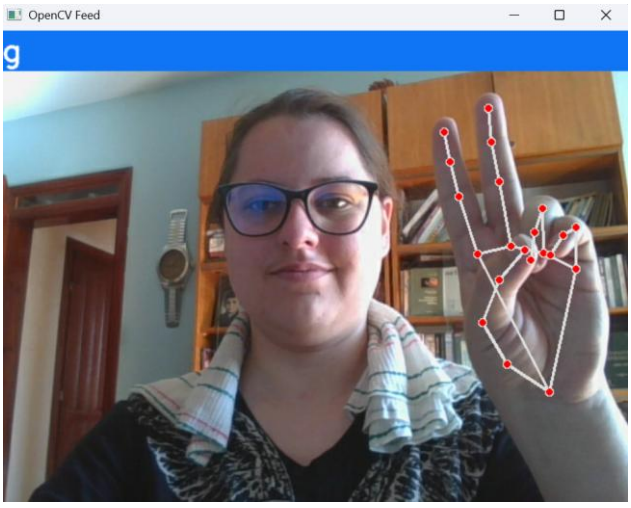

Рисунок 3.9 – Робота програмного модуля з відображенням ключових точок руки

Програмний модуль передбачає виведення результатів розпізнавання у вигляді текстових субтитрів у реальному часі. У ході тестування підтверджено, що текстова інформація з’являється після розпізнання жесту, хоча зафіксовано певні затримки між введенням жесту та відображенням результату. Крім того, певний вплив може чинити і затримка, зумовлена обробкою відеопотоку та передачею даних через програмні компоненти, що також потребує подальшої оптимізації.

В ході тестування виявлено, що програмний модуль має певні обмеження у точності розпізнавання жестів. Було зафіксовано випадки некоректного розпізнавання, коли один і той самий жест інтерпретувався по-різному або ж видавався за інший жест. В окремих ситуаціях модуль генерує текстову відповідь на відповідний жест, тоді як в інших – відсутня реакція на вхідні дані. Також спостерігалися випадки виведення тексту навіть за умови відсутності руки в кадрі, що свідчить про наявність помилкових спрацьовувань у процесі аналізу відеопотоку.

Такий результат, ймовірно, пов'язаний з декількома факторами, такими як недостатня кількість або якість навчальних даних, використанням обмеженої кількості ознак для класифікації, або ж із невідповідністю обраної моделі. З цього випливає необхідність удосконалення підготовки навчальних даних, розгляду альтернативних моделей машинного навчання , здатних підвищити ефективність та точність розпізнавання.

Узагальнені результати тестування представлені у таблиці 3.1.

Таблиця 3.1 – Узагальнені результати тестування програмного модуля

Тестовані функції	Мета	Очікуваний результат	Фактичний результат
Захоплення відео	Перевірити здатність системи здійснювати захоплення відео з камери	Передача зображення з камери	Здійсгюється захоплення відео та передається для наступної роботи
Розпізнавання рук	Визначити, чи здатна система виявляти присутність рук у кадрі	Система точно визначає контури та положення рук у кадрі	Руки розпізнаються коректно, незалежно від положення у кадрі

Продовження таблиці 3.1

Тестовані функції	Мета	Очікуваний результат	Фактичний результат
Виведення тексту	Перевірити здатність системи виводити текстовий результат розпізнаного жесту	Відображення тексту на екрані після кожного розпізнаного жесту	Виведення тексту функціонує, система демонструє текст на екрані
Коректність розпізнавання жестів	Оцінити точність відповідності між введеним жестом і текстовим перекладом	Жест розпізнається відповідно до навчальних даних	Рівень коректності низький; жести часто розпізнаються помилково або не ідентифікуються

Таким чином, попри успішну реалізацію ключових етапів обробки візуальних даних, модуль потребує подальшого вдосконалення у частині розпізнавання жестів для досягнення функціональної повноцінності та відповідності практичним потребам користувачів. Зокрема, доцільним є покращення якості та розширення обсягу навчальних даних, а також оптимізація архітектури моделі шляхом застосування сучасних методів машинного навчання і глибинного навчання для підвищення точності та стабільності розпізнавання.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи:

1. Проаналізовано предметну область, обгрунтовано актуальність розробки програмного модуля створення субтитрів мови жестів з використанням комп'ютерного зору і машинного навчання, проаналізовано соціальне значення та технологічну складність задачі формування субтитрів на основі візуальних сигналів.

2. Проведено огляд існуючих рішень, який виявив розвиток теми перекладу жестової мови в текст та визначено їхні переваги та недоліки.

3. Представлено загальну структуру програмного модуля та його алгоритмічне забезпечення.

4. Реалізовано функціональний прототип програмного модуля для перекладу жестової мови у текстові субтитри в реальному часі. Для вирішення поставленої задачі застосовано засоби комп’ютерного зору (MediaPipe), методи глибокого навчання (LSTM) та бібліотеки OpenCV і TensorFlow.

5. У ході тестування було підтверджено, що розроблений модуль захоплює відеопотік, визначає ключові точки рук, розпізнає жести та виводить субтитри. Проте було виявлено деякі неточності при розпізнаванні жеста, що створює неточності в перекладі.

6. У майбутньому можна розширити набір даних для підвищення точності перекладу, а також збільшення можливостей самого перекладу, що дасть можливості перекладу цілого слова. Також можливе інтегрування з платформами, таких як відеохостинги або сервіси відеозв’язку, для забезпечення автоматичного субтитрування жестової мови в реальному часі, що значно розширить доступність відеоконтенту для осіб з порушенням слуху.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Шевченко В.В. Комп’ютерний зір: підручник. Київ: Кондор, 2019. 384 с.
2. Офіційний сайт УТОГ. URL: https://utog.org/ (дата звернення: 22.03.2025).
3. Sign Translate. URL: https://sign.mt/about (дата звернення: 22.03.2025).
4. OpenCV About. URL: https://opencv.org/about/ (дата звернення: 22.03.2025).
5. MediaPipe Solutions guide. URL: https://ai.google.dev/edge/mediapipe/solutions/guide (дата звернення: 22.03.2025).
6. Gomase K., Dhanawade A., Gurav P., Lokare S. Sign Language Recognition using Mediapipe. International Research Journal of Engineering and Technology (IRJET). 2022. Vol. 09, №. 01. P.744-746
7. TensorFlow і машинне навчання. URL: https://foxminded.ua/tensorflow-shcho-tse/ (дата звернення: 22.03.2025).
8. American Sign Language Dataset. URL: https://www.kaggle.com/datasets/ayuraj/asl-dataset/data (дата звернення: 28.05.2025).
9. Exploring OpenCV-Python: Features, Benefits, and Usage. URL: https://www.linkedin.com/pulse/exploring-opencv-python-features-benefits-usage-aditya-mishra-ljvjc (дата звернення: 22.03.2025).
10. EnableTalk. URL: https://enabletalk.com/ (дата звернення: 22.03.2025).
11. SignAloud gloves. URL: http://multilingual.com/signaloud-gloves/ (дата звернення: 22.03.2025).
12. Біланова О. А., Іващенко Л. Д., Дяків С. М. Українська жестова мова : навч. посіб. для осіб з особл. освіт. потребами (Н90, Н91): 5 кл. Ч. 1. Київ: ТОВ «Видавництво Атлант», 2024. 112 с.
13. What is computer vision?. URL: https://www.ibm.com/think/topics/computer-vision/ (дата звернення: 28.05.2025).
14. SignAll. URL: https://signall.world/ (дата звернення: 28.05.2025).
15. Python. URL: https://www.python.org/ (дата звернення: 28.05.2025).
16. Working with RNNs. URL: https://www.tensorflow.org/guide/keras/working_with_rnns (дата звернення: 28.05.2025).
17. Das U., Lawson A., Mayfield C., Norouzi N. Introduction to Python Programming. Houston: OpenStax, 2024. 406 p.
18. Lubanovic B. FastAPI: Modern Python Web Development. Sebastopol: O’Reilly Media, 2023. 277 p.
19. Lugaresi L., Tang J., Nash H. та ін. MediaPipe: A Framework for Building Perception Pipelines. URL: https://arxiv.org/abs/1906.08172 (дата звернення: 28.05.2025).

20. Huang J., Zhou W., Wu H., Li H. Video-based Sign Language Recognition without Temporal Segmentation. Proceedings of the AAAI Conference on Artificial Intelligence. 2018. Т. 32, № 1. С. 2257-2264.

21. Cui R., Liu H., Zhang C. A Deep Neural Framework for Continuous Sign Language Recognition by Iterative Training. IEEE Transactions on Multimedia. 2019. Т. 21, № 7. С. 1880-1891.

22. Moeslund T. B., Hilton A., Krüger V. A survey of advances in vision-based human motion capture and analysis. Computer Vision and Image Understanding. 2006. Т. 104, № 2-3. С. 90-126.

23. Гонсалес Р. Цифрова обробка зображень. пер. з англ. 3-тє вид. Київ : Діалектика, 2014. 976 с.

24. Zhang, Z. Microsoft Kinect Sensor and Its Effect. IEEE Multimedia. 2012. Vol. 19, No. 2. P. 4-10.

25. Hochreiter S., Schmidhuber J. Long Short-Term Memory. Neural Computation. 1997. Vol. 9, No. 8. P. 1735-1780.

26. Brownlee J. Long Short-Term Memory Networks with Python. Machine Learning Mastery, 2017. 251 p.

27. М'якенький, О. В., Алексєєв М. О., Мацюк С.М. Дослідження моделей прогнозування часових рядів на основі рекурентних нейронних мереж типу LSTM. Науковий вісник Національного гірничого університету. 2025. № 1. С. 5-14.

28. Михайлишина М. П., Турченко І. В. Інтеграція комп’ютерного зору та глибинного навчання для розпізнавання жестової мови. Розвиток науки: теорії, відкриття та практичні результати: збірник 4-ї Міжнар. наук.-практ. конф., Цюріх, Швейцарія, 9-11 черв. 2025 р. Цюріх, 2025. С.86-88.

29. Комар М.П., Саченко А.О., Васильків Н.М., Гладій Г.М., Коваль В.С., Ліп’яніна-Гончаренко Х.В. Методичні рекомендації до виконання кваліфікаційної роботи з освітньо-професійної програми «Комп’ютерні науки» спеціальності 122 «Комп’ютерні науки» за першим (бакалаврським) рівнем вищої освіти. Тернопіль: ЗУНУ, 2024. 52 с.

Додаток А

Програмний код навчання моделі

```
import os
import cv2
import numpy as np
from tqdm import tqdm
import mediapipe as mp
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, Dropout

mp_holistic = mp.solutions.holistic

def mediapipe_detection(image, model):
    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
    image.flags.writeable = False
    results = model.process(image)
    image.flags.writeable = True
    image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)
    return image, results

label_map = {
    'a': 0,  'b': 1,  'c': 2,  'd': 3,  'e': 4,
    'f': 5,  'g': 6,  'h': 7,  'i': 8,  'j': 9,
    'k': 10, 'l': 11, 'm': 12, 'n': 13, 'o': 14,
    'p': 15, 'q': 16, 'r': 17, 's': 18, 't': 19,
    'u': 20, 'v': 21, 'w': 22, 'x': 23, 'y': 24,
    'z': 25,
    '0': 26, '1': 27, '2': 28, '3': 29, '4': 30,
    '5': 31, '6': 32, '7': 33, '8': 34, '9': 35
}
actions = list(label_map.keys())


def extract_keypoints(results):
    lh = np.array([[res.x, res.y, res.z] for res in
results.left_hand_landmarks.landmark]).flatten() if results.left_hand_landmarks else
np.zeros(21*3)
    rh = np.array([[res.x, res.y, res.z] for res in
results.right_hand_landmarks.landmark]).flatten() if results.right_hand_landmarks else
np.zeros(21*3)
    return np.concatenate([lh, rh])
```

```
def create_dataset_from_images(data_dir, sequence_length=30):
    X, y = [], []
    with mp_holistic.Holistic(static_image_mode=True) as holistic:
        for label in tqdm(os.listdir(data_dir)):
            if label not in label_map:
                continue
            label_path = os.path.join(data_dir, label)
            if not os.path.isdir(label_path):
                continue

            all_keypoints = []
            for img_file in sorted(os.listdir(label_path)):
                img_path = os.path.join(label_path, img_file)
                image = cv2.imread(img_path)
                if image is None:
                    continue
                image_rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
                results = holistic.process(image_rgb)
                keypoints = extract_keypoints(results)
                all_keypoints.append(keypoints)

            for i in range(0, len(all_keypoints) - sequence_length + 1, sequence_length):
                sequence = all_keypoints[i:i+sequence_length]
                if len(sequence) == sequence_length:
                    X.append(sequence)
                    y.append(label_map[label])

    X = np.array(X)
    y = to_categorical(y, num_classes=len(label_map))
    return X, y

def build_model():
    model = Sequential()
    model.add(LSTM(64, return_sequences=True, activation='relu', input_shape=(30,
1662)))
    model.add(Dropout(0.2))
    model.add(LSTM(128, return_sequences=True, activation='relu'))
    model.add(Dropout(0.3))
    model.add(LSTM(64, return_sequences=False, activation='relu'))
    model.add(Dense(64, activation='relu'))
    model.add(Dropout(0.2))
    model.add(Dense(36, activation='softmax'))
    model.compile(optimizer='adam', loss='categorical_crossentropy',
metrics=['accuracy'])
    return model
```

```
if __name__ == "__main__":
    DATASET_PATH = '/content/ASL_Dataset/Train'
    print("[INFO] Creating dataset...")
    X_train, y_train = create_dataset_from_images(DATASET_PATH)
    print(f"[INFO] Dataset shape: X={X_train.shape}, y={y_train.shape}")

    print("[INFO] Building model...")
    model = build_model()

    print("[INFO] Training model...")
    model.fit(X_train, y_train, epochs=30, batch_size=32, validation_split=0.1)

    print("[INFO] Saving model...")
    model.save("asl_lstm_model.h5")
```

Додаток Б

Програмний код розпізнавання жестів у реальному часі

```
import cv2
import numpy as np
import mediapipe as mp
from tensorflow.keras.models import load_model

mp_holistic = mp.solutions.holistic
model = load_model("asl_lstm_model.h5")

label_map = {
    'a': 0, 'b': 1, 'c': 2, 'd': 3, 'e': 4,
    'f': 5, 'g': 6, 'h': 7, 'i': 8, 'j': 9,
    'k': 10, 'l': 11, 'm': 12, 'n': 13, 'o': 14,
    'p': 15, 'q': 16, 'r': 17, 's': 18, 't': 19,
    'u': 20, 'v': 21, 'w': 22, 'x': 23, 'y': 24,
    'z': 25,
    '0': 26, '1': 27, '2': 28, '3': 29, '4': 30,
    '5': 31, '6': 32, '7': 33, '8': 34, '9': 35
}
actions = list(label_map.keys())

def extract_keypoints(results):
    lh = np.array([[res.x, res.y, res.z] for res in
results.left_hand_landmarks.landmark]).flatten() if results.left_hand_landmarks else
np.zeros(21*3)
    rh = np.array([[res.x, res.y, res.z] for res in
results.right_hand_landmarks.landmark]).flatten() if results.right_hand_landmarks else
np.zeros(21*3)
    return np.concatenate([lh, rh])

def mediapipe_detection(image, model):
    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
    image.flags.writeable = False
    results = model.process(image)
    image.flags.writeable = True
    image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)
    return image, results


sequence = []
sentence = []
threshold = 0.8
```

```
cap = cv2.VideoCapture(0)
with mp_holistic.Holistic(min_detection_confidence=0.5,
                          min_tracking_confidence=0.5) as holistic:
    while cap.isOpened():
        ret, frame = cap.read()
        image, results = mediapipe_detection(frame, holistic)
        keypoints = extract_keypoints(results)
        sequence.append(keypoints)
        sequence = sequence[-30:]
        if len(sequence) == 30:
            res = model.predict(np.expand_dims(sequence, axis=0))[0]
            print(actions[np.argmax(res)])
            if res[np.argmax(res)] > threshold:
                if len(sentence) > 0:
                    if actions[np.argmax(res)] != sentence[-1]:
                        sentence.append(actions[np.argmax(res)])
                else:
                    sentence.append(actions[np.argmax(res)])
            if len(sentence) > 5:
                sentence = sentence[-5:]
        cv2.rectangle(image, (0,0), (640, 40), (245, 117, 16), -1)
        cv2.putText(image, ' '.join(sentence), (3,30),
                      cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 255, 255), 2,
                    cv2.LINE_AA)
        cv2.imshow('OpenCV Feed', image)
        if cv2.waitKey(10) & 0xFF == ord('q'):
            break
    cap.release()
    cv2.destroyAllWindows()
```

Додаток В

Сертифікат участі у конференції

CERTIFICATE
of participation

EUROPEAN OPEN SCIENCE SPACE

Mariia Mykhailyshyna

took part in the 4th International Scientific and Practical Conference

«EVOLVING SCIENCE: THEORIES, DISCOVERIES AND PRACTICAL OUTCOMES»

12 Hours of Participation
(0,4 ECTS credits)

Head of the
organizing committee
Helen Volokitina

EUROPEAN OPEN SCIENCE SPACE 3382508349

EOSS-25/0609-054

June 9-11, 2025, Zurich, Switzerland

eoss-conf.com

Додаток Г

Копія опублікованих результатів

eoss-conf.com

ISSUE
№39

EUROPEAN OPEN
SCIENCE SPACE

COLLECTION OF SCIENTIFIC PAPERS

4TH INTERNATIONAL
SCIENTIFIC
AND PRACTICAL
CONFERENCE

EVOLVING SCIENCE: THEORIES, DISCOVERIES AND PRACTICAL OUTCOMES

JUNE 9-11, 2025, ZURICH, SWITZERLAND

Section: Geography, Geology and Geodesy

Балабак О.
РОЛЬ ДРОНІВ ТА ЛІДАРІВ У МОДЕРНІЗАЦІЇ УПРАВЛІННЯ ЗЕМЕЛЬНИМИ РЕСУРСАМИ……... 81

Section: Information Technology, Cyber Security and Computer Engineering

Клівак В.
ЗАХИЩЕНІ МЕРЕЖЕВІ ПРОТОКОЛИ ДЛЯ VR: АДАПТАЦІЯ СТАНДАРТНИХ РІШЕНЬ ПІД НИЗЬКІ ЗАТРИМКИ……. 84

Михайлишина М.П., Турченко І.В.
ІНТЕГРАЦІЯ КОМП'ЮТЕРНОГО ЗОРУ ТА ГЛИБИННОГО НАВЧАННЯ ДЛЯ РОЗПІЗНАВАННЯ ЖЕСТОВОЇ МОВИ…… 86

Тимонін Ю.О., Савелко А.
АДАПТИВНІ АЛГОРИТМИ ІНФОРМАЦІЙНОГО АНАЛІЗУ ДЛЯ УПРАВЛІННЯ КРИПТОПОРТФЕЛЕМ……. 88

Січкар М., Веретюк С.
ВПЛИВ ПРОГРАМНИХ ЗАСТОСУНКІВ НА ЕФЕКТИВНІСТЬ КЕРУВАННЯ ДІЯЛЬНОСТІ СТАРТАП ПРОЄКТІВ…….. 92

Федорчук Д.Ю.
ДОСЛІДЖЕННЯ МОЖЛИВОСТЕЙ АВТОМАТИЗОВАНОГО ПРОГРАМНОГО НАЛАШТУВАННЯ ДЛЯ ІНТЕГРАЦІЇ РІЗНОРІДНИХ ТЕХНОЛОГІЙ У РОЗУМНИХ СИСТЕМАХ……. 95

Бацула М., Николюк О.
ІНФОРМАЦІЙНА СИСТЕМА ЯК ІНСТРУМЕНТ ПЕРСОНАЛЬНОГО ТАЙМ-МЕНЕДЖМЕНТУ ТА ДОСЯГНЕННЯ ЦІЛЕЙ……. 99

Рябчук Я.В., Николюк О.М.
ОГЛЯД НЕЙРОМЕРЕЖІ QWEETS ДЛЯ ПРОГНОЗУВАННЯ РЕЗУЛЬТАТІВ ЗМАГАНЬ У КІБЕРСПОРТІ……. 102

Бойко М.В., Топольницький П.П.
ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ДЛЯ ЗБОРУ ТА ОБРОБКИ ДАНИХ ЗА ДОПОМОГОЮ МУЛЬТРАСПЕКТРАЛЬНОГО СКАНЕРА TERRA.. 105

WebRTC із локалізованими серверами сигналізації. Легковагі UDP-центровані протоколи з апаратним прискоренням шифрування створюють додаткові можливості для подальшого зниження затримки та підвищення продуктивності VR-систем.

Список використаних джерел

1. Sheffer Y., Holz R., Saint-Andre P. Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS) [Електронний ресурс] / Internet Engineering Task Force. – Режим доступу: https://www.rfc-editor.org/info/rfc7525. – Дата звернення: 04.06.2025.
2. QUIC [Електронний ресурс] / Wikipedia: The Free Encyclopedia. – URL: https://en.wikipedia.org/wiki/QUIC. – Дата звернення: 04.06.2025.
3. Di Francesco M., Kazantzaki K., Frossard P. A systematic review on WebRTC for potential applications [Електронний ресурс] / Multimedia Tools and Applications (Springer). – DOI: 10.1007/s11042-024-20448-9. – Дата звернення: 04.06.2025.
4. Datagram Transport Layer Security (DTLS) [Електронний ресурс] / GetStream.io. – URL: https://getstream.io/glossary/datagram-transport-layer-security. – Дата звернення: 04.06.2025.

ІНТЕГРАЦІЯ КОМП'ЮТЕРНОГО ЗОРУ ТА ГЛИБИННОГО НАВЧАННЯ ДЛЯ РОЗПІЗНАВАННЯ ЖЕСТОВОЇ МОВИ

Михайлишина Марія Петрівна
здобувач вищої освіти бакалаврського рівня
Науковий керівник:
Турченко Ірина Василівна
к. т. н., доцент
Кафедра інформаційно-обчислювальних систем і управління
Західноукраїнський національний університет, Україна

Анотація. У роботі розглядається підхід до розпізнавання жестової мови з використанням сучасних інструментів комп'ютерного зору та методів глибинного навчання. Акцент зроблено на інтеграції бібліотек OpenCV, MediaPipe та OpenPose для визначення ключових точок скелетної моделі людини, а також застосуванні згорткових і рекурентних нейронних мереж для класифікації жестів. Проаналізовано актуальні підходи до попередньої обробки даних, а також переваги використання трансформерів для аналізу часових послідовностей. Представлено огляд інструментальних засобів і методів, що забезпечують ефективність розпізнавання навіть за умов обмежених ресурсів.

Ключові слова: жестова мова, комп'ютерний зір, MediaPipe, OpenPose, OpenCV, глибинне навчання, CNN, RNN, трансформери, розпізнавання жестів.

Введення. Жестова мова є важливим засобом комунікації для людей з порушеннями слуху та мовлення. Автоматичне розпізнавання жестів дозволяє створювати технології, що полегшують їхню інтеграцію у цифрове середовище та повсякденне життя. Сучасні комп'ютерні системи здатні аналізувати жести завдяки поєднанню методів комп'ютерного зору, машинного навчання та обробки часових послідовностей.

Мета та задачі дослідження. Метою дослідження є аналіз та узагальнення сучасних методів і технологій, що застосовуються для розпізнавання жестової мови на основі відеопотоку, з акцентом на попередню обробку зображень, визначення ключових точок тіла та моделювання часових послідовностей. Завдання дослідження: проаналізувати сучасні інструменти для визначення ключових точок тіла у відеопотоці; оцінити підходи до обробки візуальних даних та класифікації жестів за допомогою моделей глибинного навчання; дослідити перспективи використання трансформерів і сучасних бібліотек в умовах обмежених ресурсів.

Результати дослідження і їх обговорення. Розпізнавання жестової мови базується на аналізі відеопотоку або послідовностей зображень, де ключову роль відіграють положення рук, пальців, обличчя та тіла. Ефективна реалізація таких систем потребує інтеграції методів комп'ютерного зору, глибинного навчання та аналізу часових послідовностей.

Серед найпоширеніших інструментів для аналізу положення тіла виділяють OpenPose, MediaPipe та BlazePose. Наприклад, OpenPose дозволяє визначати до 135 ключових точок скелетної моделі людини у відео в реальному часі [1].

MediaPipe, розроблений Google, забезпечує високу швидкодію при обробці зображень з мобільних пристроїв та вебкамер [2].

Значну роль у попередній обробці зображень відіграє бібліотека OpenCV, яка застосовується для фільтрації шуму, виявлення контурів та ключових точок. У поєднанні з MediaPipe, що забезпечує визначення ключових точок, стає можливим формування точного вектору ознак для подальшої класифікації [4].

На етапі моделювання та розпізнавання жестів застосовуються глибокі нейронні мережі, зокрема згорткові (CNN) та рекурентні мережі (RNN, LSTM). CNN використовуються для витягання ознак із зображень або відеокадрів, тоді як RNN і трансформери дозволяють аналізувати часову послідовність жестів [3].

Останні дослідження демонструють зростання інтересу до використання трансформерів, які здатні моделювати контекст у складних комунікативних структурах та забезпечувати високу точність розпізнавання [5].

Невід'ємною складовою є попередня обробка вхідних даних, що включає нормалізацію координат скелетної моделі, фільтрацію шуму та сегментацію кадрів. Реалізація відповідних алгоритмів можлива за допомогою програмних

засобів на зразок TensorFlow, PyTorch та OpenCV, які забезпечують гнучкість та відносну ефективність навіть за умов обмежених обчислювальних ресурсів.

Висновки. Розпізнавання жестової мови є міждисциплінарною задачею, що поєднує комп'ютерний зір, машинне навчання та обробку часових послідовностей. Аналіз сучасних інструментів, таких як MediaPipe, OpenPose та OpenCV, показує їхню здатність ефективно визначати ключові точки скелетної моделі людини у реальному часі. Застосування згорткових нейронних мереж та рекурентних моделей, а також перспективних трансформерів, дозволяє підвищити точність класифікації жестів. Комплексне використання цих технологій створює основу для побудови адаптивних і продуктивних систем розпізнавання жестів, здатних працювати в реальному часі.

Список використаних джерел

1. Cao Z., Hidalgo G., Simon T. et al. OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2021. – Vol. 43, No. 1. – P. 172–186.
2. Google. MediaPipe [Електронний ресурс]. – Режим доступу: https://mediapipe.dev/ (дата звернення: 07.06.2025).
3. Zhang X., Liu C., Yu H. A review of sign language recognition methods // Pattern Recognition Letters. – 2021. – Vol. 148. – P. 3–11.
4. Bradski G. The OpenCV Library. Dr. Dobb's Journal of Software Tools, 2000. URL: https://opencv.org/
5. Vaswani, A. et al. Attention Is All You Need. Advances in Neural Information Processing Systems, 2017, vol. 30. URL: https://papers.nips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

АДАПТИВНІ АЛГОРИТМИ ІНФОРМАЦІЙНОГО АНАЛІЗУ ДЛЯ УПРАВЛІННЯ КРИПТОПОРТФЕЛЕМ

Тимонін Ю.О.
к.т.н., доцент кафедри КТіМС
Савелко Артем
здобувач вищої освіти бакалаврського рівня
ОП «Інформаційні системи та технології»
Поліський національний університет

Вступ.

Управління криптовалютними активами є складним завданням, особливо для інвесторів, які не мають глибоких знань у галузі фінансового аналізу [1, 3]. Ринок криптовалют характеризується високою волатильністю, нестабільністю