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 другий (магістерський)

ДОСЛІДЖЕННЯ МЕТОДІВ СТЕЖЕННЯ ЗА РУКАМИ ДЛЯ СУРДОПЕРЕКЛАДУ ЖЕСТОВИХ МОВ
(тема)

Виконав:
студент 2 курсу, групи ІНФм-23-1
Шовковий Є.І.
(прізвище, ініціали)

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

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

Керівник проф. Машталір В.П.
(посада, прізвище, ініціали)

Допускається до захисту

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

2025 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра Інформатики
(повна назва)

Рівень вищої освіти другий (магістерський)

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

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«» _____ 20 ___ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Шовковому Євгенію Ігоровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів стеження за руками для сурдоперекладу жестових мов

затверджена наказом по університету від 25 листопада 2024 року № 1246Ст

2. Термін подання студентом роботи до екзаменаційної комісії 29 грудня 2024 р.

3. Вихідні дані до роботи література та наукові джерела, що стосуються методів трекінгу рухів рук на основі комп'ютерного зору та 2D-аналізу, матеріали з розпізнавання жестової мови та побудови систем сурдоперекладу, ресурси з архітектури нейронних мереж (LSTM та гібридних моделей), документація з використання Python та бібліотек OpenCV, MediaPipe, TensorFlow і Keras для обробки відео та навчання моделей, дані щодо методів фільтрації та обробки сигналів.

4. Перелік питань, що потрібно опрацювати в роботі
1. Аналіз сучасних підходів до сурдоперекладу.
2. Огляд сучасних технологій стеження за руками.
3. Класифікація методів відстеження рухів рук.
4. Моделювання та вибір методу для системи сурдоперекладу.
5. Навчання моделей та оцінка результатів розпізнавання жестів.
6. Тестування оптимальної системи сурдоперекладу.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) опис роботи 2D та 3D трекінгу, принцип роботи ToF, схеми роботи датчиків, приклади використання контролерів, опис алгоритму Кенні, схеми роботи нейронних мереж, візуальні інтерфейси інсуючих програм для сурдоперекладу, етапи реалізації поставленої задачі, результати тестування.

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	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	25.11.2024	
2	Аналіз завдання, підбір літератури	25.11.24-26.11.24	
3	Аналіз літератури з досліджуваної проблеми	26.11.24-28.11.24	
4	Аналіз методів стеження за руками	28.11.24-03.12.24	
5	Розробка системи сурдоперекладу	03.12.24-10.12.24	
6	Програмна реалізація	10.12.24-13.12.24	
7	Оформлення пояснювальної записки	13.12.24-18.12.24	
8	Перевірка на плагіат	20.12.2024	
9	Рецензування	24.12.2024	
10	Підготовка презентації та доповіді	25.12.2024	
11	Занесення роботи в електронний архів	30.12.2025	
12	Попередній захист кваліфікаційної роботи	07.01.2025	

Дата видачі завдання 25 листопада 2024 р.

Студент _____
(підпис)

Керівник роботи _____ проф. Машталір В.П.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 87 с., 2 табл., 33 рис., 1 дод., 71 джерело.

АЛГОРИТМ КЕННІ, ФІЛЬТР КАЛМАНА, РОЗПІЗНАВАННЯ ЖЕСТІВ, КЛЮЧОВІ ТОЧКИ РУК, ТРЕКІНГ РУК, ЖЕСТОВА МОВА, КІНЕМАТИЧНА МОДЕЛЬ, НЕЙРОННІ МЕРЕЖІ, ОПЕРАТОР СОБЕЛЯ.

Об’єктом дослідження є системи стеження за рухами рук для забезпечення перекладу жестової мови, що є важливими в процесі комунікації з людьми, які мають порушення слуху.

Метою дослідження є аналіз методів відстеження рухів рук та створення ефективної системи сурдоперекладу жестової мови, яка гарантує точне розпізнавання жестів у реальному часі, водночас залишаючись економічно вигідною та доступною для широкої аудиторії.

Використано методи комп’ютерного зору, нейронних мереж та аналізу ключових точок рук. Проведено огляд і порівняння сучасних технологій, створено датасет, а також протестовано різні архітектури нейронних мереж.

У результаті дослідження здійснена програмна реалізація системи для сурдоперекладу.

KENNY ALGORITHM, KALMAN FILTER, GESTURE RECOGNITION, HAND KEYPOINTS, HAND TRACKING, SIGN LANGUAGE, KINEMATIC MODEL, NEURAL NETWORKS, SOBEL OPERATOR.

The object of research is hand movement tracking systems to provide sign language translation, which is important in the process of communication with people who have hearing impairments.

The aim of the research is to analyze methods for tracking hand movements and create an effective sign language interpretation system that guarantees accurate gesture recognition in real time, while remaining cost-effective and accessible to a wide audience.

Methods of computer vision, neural networks, and hand keypoint analysis were used. A review and comparison of modern technologies was conducted, a dataset was created, and various neural network architectures were tested.

As a result of the research, a software implementation of a sign language interpretation system was carried out.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

UASL – Ukrainian Sign Language (українська жестова мова)

ASL – American Sign Language (американська жестова мова)

CNN – Convolutional neural network (згорткова нейронна мережа)

RNN – Recurrent neural network (рекурентна нейронна мережа)

DNN – Deep neural network (глибока нейронна мережа)

GANs – Generative adversarial network (генеративна змагальна мережа)

LSTM – Long short-term memory (довга короткочасна пам’ять)

IR – Infrared (інфрачервоний)

IMU – Inertial Measurement Unit (Інерційний вимірювальний блок)

API – application programming interface (прикладни́й програмний інтерфейс)

MCP – metacarpophalangeal (метакарпофаланговий суглоб)

PIP – proximal interphalangeal (проксимальний міжфаланговий суглоб)

DIP – distal Interphalangeal (дистальний міжфаланговий суглоб)

AR – augmented reality (доповнена реальність)

VR – virtual reality (віртуальна реальність)

ВСТУП

Актуальність розробки автоматизованих систем сурдоперекладу обумовлена потребою створення інклюзивного середовища для людей з вадами слуху. За даними міністерства охорони здоров’я, понад 400 мільйонів людей у світі [1] страждають на порушення слуху, що створює суттєві бар’єри для їх соціальної інтеграції та ефективної комунікації в суспільстві. За прогнозами, до 2050 року ця кількість може майже подвоїтися, і кожен десятий мешканець планети матиме проблеми зі слухом [2].

Жестова мова є основним засобом спілкування для багатьох людей із вадами слуху, проте її використання залишається обмеженим через недостатню кількість кваліфікованих сурдоперекладачів, що ускладнює доступ до інформації, освіти та інших суспільних ресурсів.

Сучасні технології комп’ютерного зору та машинного навчання відкривають нові можливості для автоматизації процесу перекладу жестової мови, надаючи можливість створювати системи, що здатні розпізнавати жести та передавати їх у вигляді тексту чи голосового повідомлення.

Одним із ключових завдань таких систем є відстеження рухів рук, оскільки вони є основними елементами у формуванні жестів. Розробка точних і ефективних методів стеження за руками у реальному часі є критично важливою для досягнення високої якості розпізнавання жестової мови.

Враховуючи складність та різноманітність жестів, основною проблемою є забезпечення високої точності визначення положення рук та їх ключових точок у просторі. Це потребує застосування передових методів обробки зображень, нейронних мереж та алгоритмів глибокого навчання. Крім того, необхідно враховувати фактори навколишнього середовища, різні умови освітлення та індивідуальні особливості користувачів.

Метою цієї роботи є дослідження існуючих методів стеження за руками для автоматизованого сурдоперекладу жестової мови, їх аналіз та розробка

ефективної системи, що дозволить підвищити доступність комунікації для людей із вадами слуху. Завданням дослідження є детальний огляд сучасних технологій, розробка математичної моделі для відстеження рухів рук та проведення експериментів для оцінки ефективності запропонованих підходів.

Автоматизація перекладу жестової мови є важливим кроком до покращення якості життя людей із порушеннями слуху та розширення їх можливостей для активної участі у суспільному житті.

1 АНАЛІЗ СУЧАСНИХ ПІДХОДІВ ДО СУРДОПЕРЕКЛАДУ

1.1 Поняття сурдоперекладу та жестової мови

Сурдопереклад – це процес перекладу інформації з вербальної мови на жестову або з жестової мови на вербальну. Його основна мета – забезпечити повноцінну комунікацію між людьми з порушеннями слуху та тими, хто не володіє жестовою мовою [3]. Сурдопереклад використовується в різних сферах суспільного життя: освіта, публічні заходи, телетрансляції, юридичні справи, медицина, соціальні служби.

Це важливий інструмент для соціальної інтеграції осіб з вадами слуху, що забезпечує їм рівний доступ до інформації та можливість ефективної взаємодії.

Жестова мова – це природна візуально-моторна система спілкування, самостійна система комунікації, що складається з жестів, міміки, позицій тіла та інших невербальних елементів та використовується спільнотою глухих [4]. Вона складається з жестів, які формуються за допомогою рухів рук, положення пальців, міміки та інших невербальних елементів. Кожен жест є аналогом слова або фрази у вербальній мові.

Жестова мова має свою граматику, структуру та правила, які відрізняються від мов звукових. Важливо зазначити, що жестові мови є незалежними і не є простою візуальною інтерпретацією національних мов. Наприклад, українська жестова мова (UASL) є окремою від української звукової мови системою з власною лексикою та граматикою.

Кожна країна має свою унікальну жестову мову, яка може суттєво відрізнятися від усної мови, що використовується в цій країні. Наприклад, американська жестова мова (ASL) та UASL мають різні граматичні структури та словниковий запас [5], незважаючи на те, що обидві служать для спілкування людей з вадами слуху.

1.2 Актуальність технологій сурдоперекладу

Технології сурдоперекладу, що автоматизують процес розпізнавання жестів і їх перетворення на текст або звук, відіграють важливу роль у розвитку доступних і зручних засобів комунікації для людей з порушеннями слуху.

З розвитком сучасних технологій (таких як системи автоматичного розпізнавання жестової мови, відео та аудіо переклад, мобільні додатки) можливості для інтеграції людей з вадами слуху в суспільство значно розширюються.

Технології сурдоперекладу дозволяють автоматизувати процес перекладу жестової мови на текст або аудіосигнал, що надає можливість людям з вадами слуху взаємодіяти зі світом на рівних умовах [6]. Це особливо важливо в контексті швидкого розвитку цифрових платформ, де інформація поширюється через відео, вебінари, прямі трансляції тощо.

Традиційно сурдопереклад здійснюється кваліфікованими перекладачами жестової мови, які допомагають людям з вадами слуху отримувати інформацію в реальному часі під час публічних заходів, телевізійних програм або зустрічей.

Однак, кількість таких спеціалістів є обмеженою, і вони не завжди можуть бути доступними у потрібний момент. Технології ж дозволяють значно розширити можливості перекладу та забезпечити постійний доступ до комунікації та інформації, не залежачи від наявності перекладача.

Також автоматизовані системи перекладу жестової мови відіграють важливу роль у надзвичайних ситуаціях, де швидке отримання та розуміння інформації може бути життєво необхідним. Наприклад, під час природних катастроф, аварій чи екстрених повідомлень ці системи здатні негайно перекладати важливі повідомлення та надавати доступ до них людям з порушеннями слуху [7].

Таким чином, технології сурдоперекладу не лише підвищують якість життя людей з порушеннями слуху, але й сприяють їхній соціальній адаптації, розширюють можливості для освіти, професійної діяльності та активної участі у суспільному житті.

Впровадження таких рішень дозволяє зменшити комунікаційні бар'єри, забезпечуючи рівні права та можливості для всіх членів суспільства.

1.3 Соціальне значення автоматизації перекладу жестової мови

Автоматизація перекладу жестової мови має значний соціальний вплив, оскільки вона сприяє покращенню якості життя людей з вадами слуху та забезпечує їхню інтеграцію у суспільство.

Одним з основних соціальних аспектів автоматизації перекладу жестової мови є її вплив на інклюзію в освіті та професійній діяльності. Люди з порушеннями слуху часто стикаються з обмеженим доступом до навчальних ресурсів через відсутність перекладу або спеціалізованих програм [8]. Автоматизовані системи перекладу жестової мови дозволяють створювати інтерактивні освітні платформи, де інформація стає доступною безпосередньо в режимі реального часу. Це не тільки допомагає у навчанні, але й відкриває нові можливості для професійного розвитку та подальшої кар'єри.

Також важливим соціальним аспектом є доступ до громадських та соціальних послуг. Автоматизовані системи перекладу можуть бути використані у державних установах, медичних закладах, на громадських заходах та у сферах обслуговування, забезпечуючи людям з вадами слуху можливість отримати необхідну інформацію та брати участь у життєдіяльності на рівні з іншими громадянами. Це особливо важливо у

країнах та регіонах, де кількість кваліфікованих сурдоперекладачів є обмеженою.

Окрім цього, автоматизовані системи сурдоперекладу сприяють культурному збагаченню. Вони можуть бути застосовані під час культурних заходів, таких як театральні вистави, кіносеанси або концерти, де часто люди з вадами слуху стикаються з обмеженим доступом до інформації. Це дає можливість ширшому колу людей насолоджуватися культурними подіями і сприяє їх включенню в культурне життя суспільства.

Таким чином, автоматизація перекладу жестової мови є не лише технологічним досягненням, але й важливим кроком до створення інклюзивного суспільства, де кожен, незалежно від фізичних можливостей, має рівний доступ до інформації, освіти, професійної діяльності та культурних подій. Це значно підвищує якість життя людей з вадами слуху та сприяє їхній соціальній адаптації.

1.4 Виклики у впровадженні систем сурдоперекладу

Впровадження автоматизованих систем сурдоперекладу, попри їхню значну важливість для людей з вадами слуху, стикається з низкою проблем і викликів, які потребують ретельного дослідження та вирішення. Ці труднощі охоплюють як технологічні, так і соціальні аспекти, що впливають на ефективність та практичну реалізацію таких систем [9, 46].

Розробка таких систем є також складним завданням ще й через велику варіативність жестів, регіональні особливості жестових мов і складність перекладу невербальних компонентів комунікації, таких як міміка або рухи тіла, які є невід’ємною частиною жестової мови. Далі наведено детальний опис кожної із перелічених проблем.

Технологічні виклики:

– складність розпізнавання жестів;

– точність та реалістичність відстеження рухів;

– необхідність обробки невербальних компонентів;

– обмеження апаратного забезпечення.

Лінгвістичні та культурні виклики:

– різноманітність жестових мов;

– труднощі з перекладом абстрактних понять.

Соціальні та економічні виклики:

– доступність і впровадження;

– навчання користувачів.

Виклики розвитку штучного інтелекту:

– точність алгоритмів машинного навчання;

– проблема узгодження з реальними умовами.

Таким чином, впровадження систем сурдоперекладу супроводжується численними проблемами, які потребують вирішення для досягнення їх ефективного та масового використання. Проте подолання цих викликів створює значні можливості для покращення якості життя людей з вадами слуху та сприяє їхньому успішному включенню в суспільство.

1.5 Огляд сучасних технологій стеження за руками

Технології відстеження рук (hand tracking) є важливим напрямком в області комп’ютерного зору та взаємодії людини з комп’ютером, які використовуються в різних сферах, таких як віртуальна реальність, жестова мова, медичні програми, робототехніка, ігри тощо. Вони дозволяють системам розпізнавати рухи рук і пальців у реальному часі, перетворюючи ці рухи на команди чи інтерпретацію жестів. У контексті сурдоперекладу технології відстеження рук дозволяють автоматизувати жестовий переклад у

текст або мову, що значно підвищує доступність комунікації для людей з вадами слуху.

1.5.1 Камерні системи

Однією із найпоширеніших технологій стеження за руками є використання камерних систем, які фіксують положення та рухи рук у просторі. Ці системи можуть бути поділені на дві основні категорії: 2D та 3D технології стеження.

2D-трекінг на основі звичайних камер – це найбільш доступний варіант, що використовує звичайні RGB-камери для розпізнавання положення рук. Алгоритми комп’ютерного зору аналізують кадри відео, визначаючи форму та положення рук на площині. Такий підхід часто використовується в мобільних пристроях або веб камерах [10]. Однак він має обмеження у точності, оскільки 2D-зображення не дає змоги точно визначити глибину руху рук.

У процесі 2D-трекінгу система фіксує зображення рук за допомогою камери та використовує алгоритми комп’ютерного зору для визначення їхніх положень і контурів у кадрі. Основна мета – ідентифікувати ключові точки на руках (наприклад, кінчики пальців, зап’ястя), відстежити їхнє переміщення в кадрі та визначити жести і рухи. На рисунку 1.1 показано основі етапи роботи 2D-трекінгу [11, 12].

Рисунок 1.1 – Основі етапи роботи 2D-трекінгу

Для реалізації 2D-трекінгу рук застосовуються різноманітні алгоритми комп’ютерного зору, зокрема:

– сегментація за кольором шкіри. Один із найпростіших методів для виокремлення рук з фону, де аналізується колір шкіри [13]. Алгоритм визначає області з певним діапазоном кольорів, що відповідає кольору людської шкіри. Недоліком цього методу є чутливість до змін освітлення та до різноманітності кольорів шкіри;

– методи на основі контурів. Визначення контурів рук дозволяє відстежувати їхні контурні силуети. Алгоритми обробки зображень, такі як Canny edge detection, можуть бути використані для виділення контурів рук і пальців. Після цього система аналізує форму контуру для визначення положення пальців і долонь [14-16];

– методи на основі руху. Алгоритми, що використовують аналіз руху (оптичний потік), відстежують зміни в пікселях між кадрами відео [17]. Вони можуть бути використані для виявлення руху рук і їхньої траєкторії;

– глибокі нейронні мережі. Сучасні моделі глибокого навчання, такі як Convolutional Neural Networks (CNN), здатні автоматично виділяти ознаки зображення та виявляти положення рук із високою точністю [18]. Моделі машинного навчання можуть навчатися на великих наборах даних і з часом ставати більш адаптивними до різних умов.

2D-трекінг широко використовується у різних сферах, таких як мобільні додатки для перекладу жестової мови, інтерактивні системи (у рекламних або освітніх системах для взаємодії з користувачем через прості жести), дослідження та експерименти. Приклад використання 2D-трекінгу показано на рисунку 1.2.

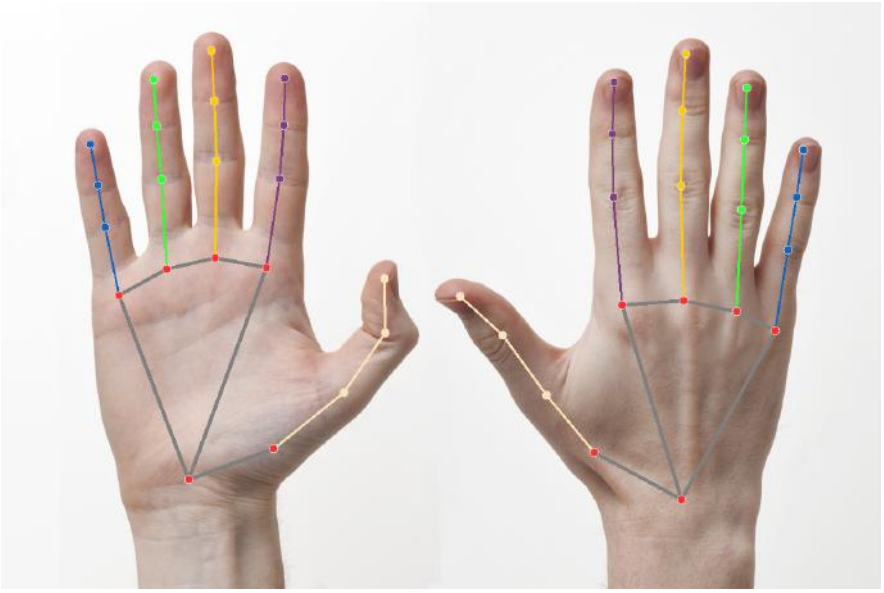

Рисунок 1.2 – Приклад використання 2D-трекінгу [19]

Більш точні системи, такі як Microsoft Kinect, Intel RealSense або Leap Motion, використовують 3D-трекінг із використанням глибоких камер, які здатні визначати не лише положення, але й глибину та просторову орієнтацію рук. Такі пристрої створюють тривимірну карту простору, що дозволяє

точніше розпізнавати рухи та позиції рук і пальців. Це особливо важливо для точного відстеження складних жестів жестової мови [20].

Глибокі камери (сенсори глибини), працюють за принципом активного або пасивного зчитування глибини, що дає змогу створювати тривимірну карту простору, включаючи положення рук і пальців користувача.

До найпоширеніших методів виявлення глибини можна віднести стереозображення, інфрачервоне сканування та техногогію Time of Flight (ToF) [21].

Стереозображення означає те, що деякі глибокі камери використовують дві або більше камер для отримання різних зображень одного об’єкта під різними кутами. За допомогою аналізу цих зображень система створює карту глибини, схожу на те, як людина бачить світ завдяки стереозору.

Інфрачервоне сканування полягає в тому, що камера випромінює інфрачервоне світло, яке, відбиваючись від об’єктів, фіксується спеціальними сенсорами. Ця технологія забезпечує надійне відстеження навіть за умов недостатнього освітлення, оскільки інфрачервоні промені не залежать від видимого світла.

ToF – технологія, принцип роботи якої полягає у вимірюванні часу, за який інфрачервоне (IR) або лазерне світло, випущене камерою [22], відбивається від об’єкта й повертається назад до сенсора (рис. 1.3). Камера випромінює світловий імпульс і фіксує час його відбиття від кожної точки сцени. Знаючи швидкість світла, система обчислює відстань до об’єкта для кожної точки, створюючи тривимірну карту глибини сцени.

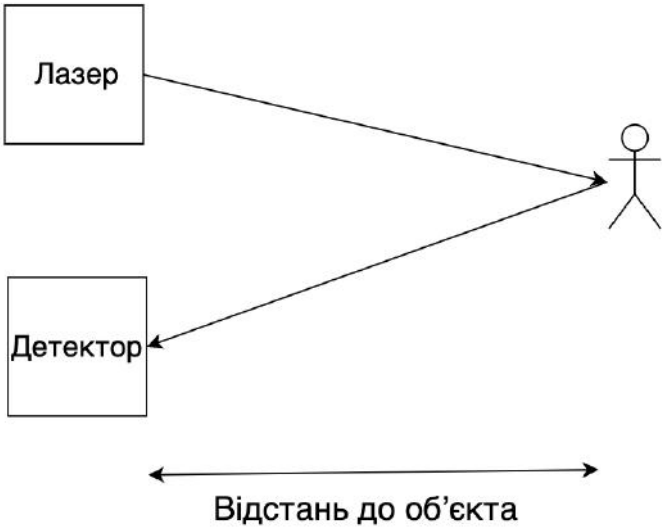

Рисунок 1.3 – Принцип роботи ToF

На рисунку 1.4 показано основі етапи роботи 3D-трекінгу.

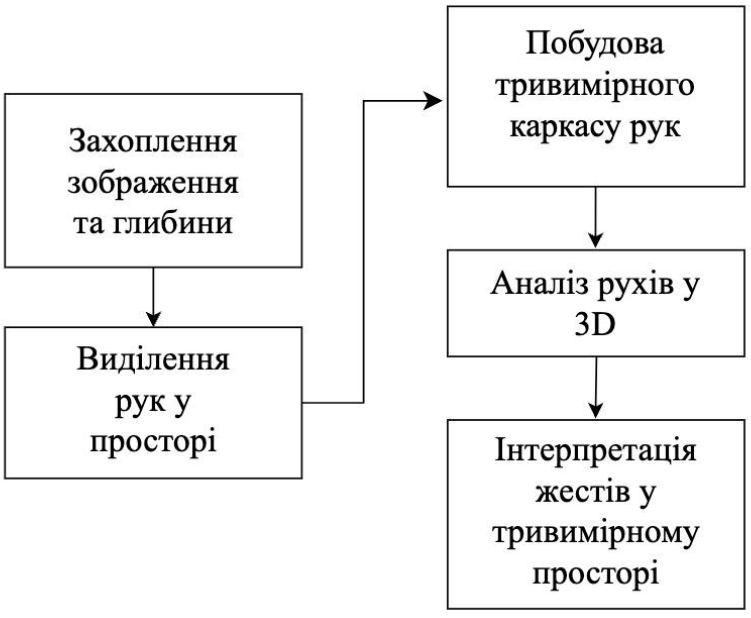

Рисунок 1.4 – Основі етапи роботи 3D-трекінгу

У 3D-трекінгу за допомогою глибоких камер використовуються передові методи обробки зображень та машинного навчання.

Використання алгоритмів обробки 3D-даних дозволяє точно визначати координати точок на руках і пальцях у просторі. Ці алгоритми обробляють карту глибини та побудовану модель рук, щоб відслідковувати навіть дрібні рухи, наприклад, згинання пальців.

Також, нейронні мережі, такі як CNN, використовуються для покращення точності розпізнавання жестів і рухів рук. Вони можуть навчатися на великих наборах даних, що дозволяє системі розпізнавати складні жести або адаптуватися до різних користувачів і умов.

Ще 3D-системи використовують методи класичного комп’ютерного зору для відстеження контурів і рухів рук. Глибокі камери дозволяють комбінувати ці методи з аналізом глибини, що забезпечує більш точне визначення рухів у трьох вимірах.

Врешті-решт, 3D-трекінг за допомогою глибоких камер є одним із найперспективніших напрямків для відстеження рухів рук завдяки високій точності та можливості аналізувати положення рук у тривимірному просторі. Він забезпечує складні та точні взаємодії у сфері віртуальної реальності, автоматичного перекладу жестової мови та інших застосунків, де необхідна точна фіксація рухів рук і пальців.

Хоча така технологія має значні переваги, вона також стикається з викликами, такими як висока вартість обладнання, складність налаштування та великі вимоги до обчислювальних ресурсів. Однак, попри ці недоліки, подальший розвиток і оптимізація 3D-трекінгу відкривають нові можливості для його застосування у різних галузях, сприяючи поліпшенню якості життя людей із вадами слуху та інтеграції інноваційних рішень у повсякденну діяльність.

1.5.2 Технології на основі сенсорів

Ще одним із найпоширеніших підходів для трекінгу рук, що забезпечують точне та швидке відстеження їхнього положення і рухів є технології на основі сенсорів. Ці технології використовують різноманітні датчики, які встановлюються на тіло або у пристрої, для збирання даних про положення, прискорення та інші параметри руху рук у просторі. Серед найбільш поширених сенсорів можна виділити інерційні датчики (IMU), датчики магнітного поля, а також ємнісні сенсори.

Інерційні датчики IMU (Inertial Measurement Unit) використовуються для вимірювання прискорення та кутових швидкостей. Вони складаються з кількох компонентів: акселерометра, гіроскопа і, часто, магнітометра.

Акселерометр вимірює лінійне прискорення об’єкта у трьох напрямках (X, Y, Z), що дозволяє визначати напрямок і швидкість руху рук.

Гіроскоп вимірює кутову швидкість, тобто швидкість обертання об’єкта навколо осей та фіксує нахил або повороти руки, що важливо для аналізу жестів.

Магнітометр (додатковий сенсор), що вимірює магнітне поле землі, дозволяє визначити абсолютну орієнтацію об’єкта у просторі, що допомагає коригувати дані акселерометра і гіроскопа, особливо при довготривалих вимірюваннях.

IMU часто використовуються у рукавичках або інших портативних пристроях.

На рисунку 1.5 показано схему роботи IMU датчику.

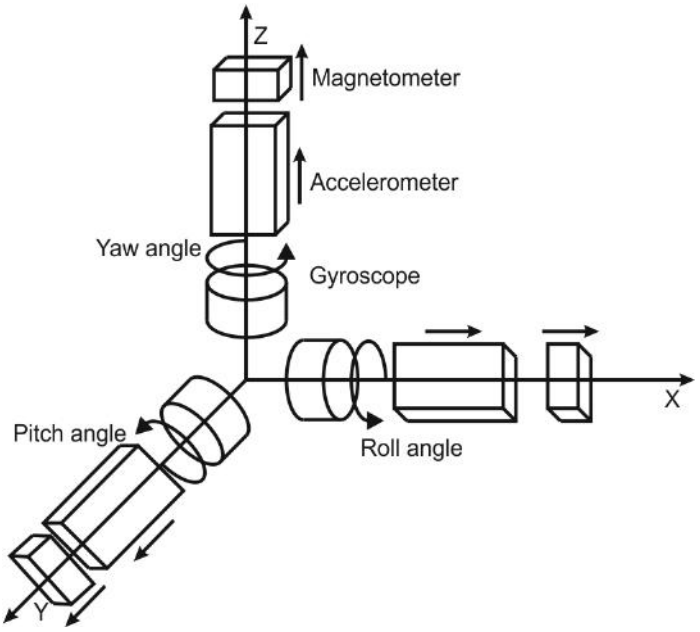

Рисунок 1.5 – Схема роботи IMU датчику [23]

IMU часто використовуються у рукавичках або інших портативних пристроях.

Ємнісні сенсори – це тип сенсорів, які використовують зміни електричного поля для виявлення наближення або дотику об’єкта. Вони працюють на основі принципу зміни ємності між електродами, коли до них наближається провідний об’єкт, такий як рука або палець. Ці сенсори широко використовуються в сенсорних екранах і пристроях для безконтактного управління.

Вони складаються з двох електродів, між якими утворюється електричне поле [24]. Коли до цього поля наближається об’єкт (наприклад, рука або палець), його електропровідність впливає на електричне поле, змінюючи ємність між електродами. Ця зміна фіксується сенсором, і на основі отриманих даних обчислюється відстань до об’єкта або визначається точка дотику.

На рисунку 1.6 представлено принцип роботи ємнісного датчику.

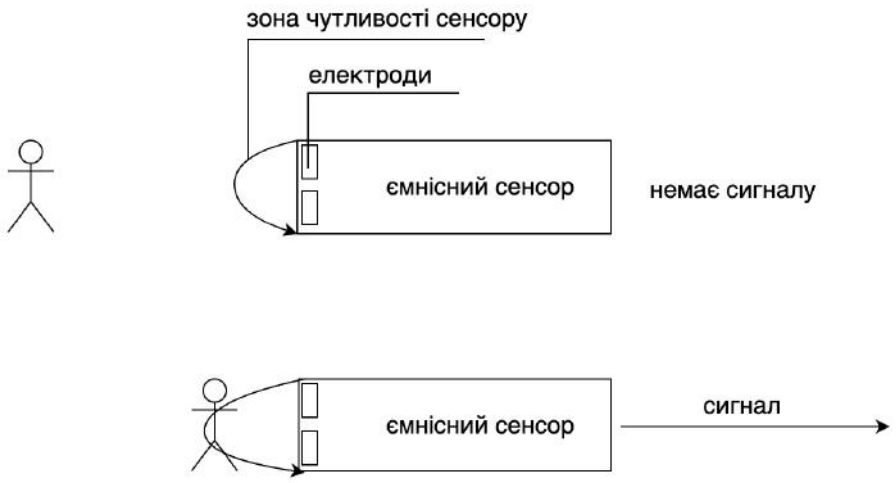

Рисунок 1.6 – Принцип роботи ємнісного датчику

Магнітні сенсори – це пристрої, які виявляють і вимірюють зміни магнітного поля. Вони використовуються для визначення положення або руху об’єктів, які містять магнітні матеріали, або для відстеження змін у магнітному середовищі [25]. У трекінгу рухів рук ці сенсори застосовуються для точного визначення просторових координат і напрямків руху, особливо у поєднанні з іншими сенсорами, такими як інерційні або ємнісні.

Магнітні сенсори реєструють магнітне поле навколо себе і виявляють його зміни. Це може бути природне магнітне поле Землі або штучно створене магнітне поле від магнітних міток, прикріплених до рук. Сенсори вимірюють інтенсивність та напрямок магнітного поля, що дозволяє визначати положення об’єкта в просторі з високою точністю.

Магнітні сенсори є надійним інструментом для трекінгу рухів рук, забезпечуючи високу точність навіть в умовах, де інші типи сенсорів можуть мати обмеження.

1.5.3 Технології доповненої та віртуальної реальності

Технології доповненої реальності (AR) та віртуальної реальності (VR) активно розвиваються і стають невід'ємною частиною багатьох сфер діяльності, включаючи ігрову індустрію, освіту, медицину та інженерію. В цих технологіях важливу роль відіграє трекінг рухів рук, який дозволяє користувачам взаємодіяти з віртуальними об'єктами та середовищами в реальному часі.

У VR користувач повністю занурюється у віртуальне середовище, яке створюється комп'ютером. Для забезпечення реалістичної взаємодії з цим середовищем необхідно точно відстежувати рухи рук та пальців. Трекінг рук у VR дозволяє користувачеві виконувати різні дії, такі як захоплення об'єктів, натискання кнопок або жестове управління інтерфейсом.

Для цього застосовуються кілька основних технологій:

– оптичний трекінг. Камери з високою роздільною здатністю використовуються для відстеження рухів рук у тривимірному просторі. Вони можуть працювати на основі 2D або 3D трекінгу. Такі системи, як Leap Motion (рис. 1.7), використовують інфрачервоні камери для високоточного відстеження положення рук та пальців;

Рисунок 1.7 – Використання Leap Motion Controller-а [26]

– сенсорні рукавички (рис. 1.8). Пристрої, які оснащені різними датчиками, такими як IMU або ємнісні сенсори. Вони дозволяють детально фіксувати рухи пальців і передавати їх у віртуальне середовище, що дає користувачам відчуття реального дотику і взаємодії з об’єктами;

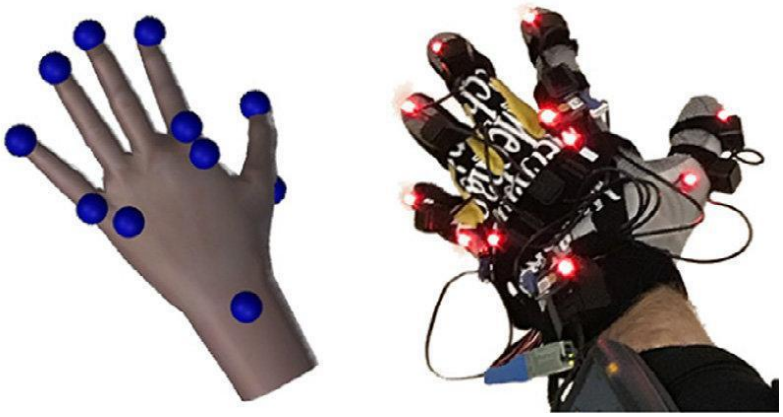

Рисунок 1.8 – Використання сенсорних рукавичок [27]

– контролери з трекінгом. Більшість сучасних VR-систем, таких як Oculus Rift або HTC Vive, використовують ручні контролери, що мають вбудовані трекінгові системи для відстеження положення рук.

AR накладає віртуальні елементи на реальний світ, надаючи користувачам додаткову інформацію або можливість взаємодіяти з реальними об’єктами через віртуальні інтерфейси [28]. У контексті AR трекінг рук також є ключовою технологією, оскільки він дозволяє користувачам маніпулювати віртуальними об’єктами, що проектуються на реальне середовище.

Важливі технології для AR включають камери мобільних пристроїв або AR-окулярів, таких як Microsoft HoloLens або Magic Leap, що використовують алгоритми комп’ютерного зору для відстеження рухів рук у реальному часі та системи з глибокими камерами. Це дозволяє користувачам взаємодіяти з віртуальними об’єктами, що «накладаються» на реальні сцени. Як у VR, у AR також використовуються камери з глибинною сенсорацією, що дозволяють точніше визначати положення рук та їхні взаємодії з реальним світом.

Зараз основними викликами у трекінгу рук для AR і VR є точність та швидкість, натуральність взаємодії та стійкість до зовнішніх перешкод. Камери або сенсори можуть бути чутливими до освітлення або фізичних перешкод, що впливає на якість трекінгу. А щоб взаємодія була максимально реалістичною, системи трекінгу повинні фіксувати навіть найдрібніші рухи пальців і кистей рук.

Таким чином перспективи розвитку технологій трекінгу рук у VR і AR включають поліпшення точності сенсорів, зниження затримок у відображенні рухів та інтеграцію штучного інтелекту для розпізнавання складних жестів і динамічних рухів. Це дозволить створювати ще більш захоплюючі й інтерактивні середовища для користувачів.

1.6 Постановка задачі дослідження

Метою даної роботи дослідження методів стеження за руками та розробка ефективної системи сурдоперекладу жестової мови, яка забезпечить високу точність розпізнавання жестів у реальному часі за умов низької вартості впровадження та доступності для широкого кола користувачів.

Для досягнення цієї мети передбачено:

– вивчення та систематизація існуючих підходів до відстеження рухів рук, включаючи методи, що базуються на комп’ютерному зорі, сенсорних технологіях та їхніх гібридних комбінаціях;

– формалізація критеріїв вибору, порівняння методів, проведення теоретичного аналізу та обґрунтування вибору конкретного рішення, яке дозволить ефективно розпізнавати жести за мінімальних ресурсів і забезпечить подальшу можливість вдосконалення;

– розробка набору даних, що включає відеоматеріали жестів із визначеними ключовими точками рук, і використання цього датасету для

побудови навчальних вибірок для якісного тренування нейронних мереж для розпізнавання жестів;

– створення кількох варіантів архітектур нейронних мереж для аналізу рухів рук, їхнє тестування для оцінки точності, швидкодії та адаптивності до різних умов роботи;

– перевірка роботи запропонованої системи в реальних умовах, включаючи тестування за різного освітлення, на змінених фонах та за наявності перешкод (тіней);

– проведення порівняльного аналізу ефективності розробленої системи з популярними аналогами для визначення її конкурентних переваг та виявлення напрямів для подальшого вдосконалення.

2 АНАЛІТИЧНИЙ ОГЛЯД МЕТОДІВ СТЕЖЕННЯ ЗА РУКАМИ

2.1 Класифікація методів відстеження рухів рук

2.1.1 Методи на основі комп’ютерного зору

Методи на основі комп’ютерного зору для відстеження рук використовують камери та алгоритми для аналізу зображень, щоб визначити положення та рухи рук у просторі. Ці методи дозволяють безконтактно відстежувати жести та рухи, що робить їх надзвичайно перспективними для застосувань у жестовому перекладі, взаємодії з інтерфейсами, доповненій та віртуальній реальності.

Комп’ютерний зір базується на аналізі послідовних зображень або відео потоків, які надходять від звичайних або спеціалізованих камер. Система обробляє ці зображення за допомогою алгоритмів, які здатні розпізнавати контури рук, відстежувати їхні положення та іноді навіть визначати окремі пальці . Процес трекінгу включає кілька ключових етапів [29, 30]:

Перший. Захоплення зображення, де камера фіксує зображення рук користувача в реальному часі.

Другий. Попередня обробка, де зображення очищається від шумів, покращується контрастність і виділяються ключові області для аналізу (руки та пальці).

Третій. Виявлення та відстеження контурів, де алгоритми аналізують форму руки і виділяють її контур. Це може включати виділення ключових точок, таких як суглоби пальців або долоні.

Четвертий. Розпізнавання рухів, де на основі трекінгу контурів та ключових точок система може визначати рухи рук і передавати їх в інтерфейс або використовувати для аналізу жестів.

Існує кілька підходів, які використовуються для трекінгу рук за допомогою комп’ютерного зору. Один із них – алгоритми на основі контурів.

Цей підхід використовує методи виділення країв і контурів на зображенні, щоб виявляти форми рук. Одним із найпопулярніших методів є алгоритм Кенні для виявлення країв, після чого система використовує інформацію про контури для трекінгу рухів.

Алгоритм Кенні (Canny edge detection) – популярний і широко використовуваний метод виявлення країв, метою якого є ідентифікація та виділення країв об’єктів на зображенні. Він був розроблений Джоном Ф. Кенні в 1986 році і з тих пір став основним інструментом комп’ютерного зору та аналізу зображень [15, 31].

Виявлення країв – це техніка, яка використовується в обробці зображень для визначення меж об’єктів на зображенні. Границя визначається як раптова зміна інтенсивності пікселів у зображенні. Краї представляють межі між окремими об’єктами або областями з різними рівнями інтенсивності.

Виділяють 6 основних кроків цього алгоритму:

Крок 1. Перетворення зображення у відтінках сірого.

Зображення у відтінках сірого мають один канал, який відображає інтенсивність кожного пікселя, що полегшує процес виявлення країв і знижує обчислювальну складність. Перетворення в градації сірого усуває колірну інформацію, залишаючи тільки яскравість пікселів.

Кольорові зображення зазвичай складаються з трьох каналів: червоного, зеленого та синього (RGB), де кожен канал містить інтенсивність відповідного кольору в кожному пікселі. На відміну від цього, зображення у відтінках сірого використовують лише один канал, де значення пікселів відповідають рівням яскравості.

Перетворення в сірі відтінки відбувається шляхом обчислення зваженої суми значень каналів RGB для кожного пікселя. Вагові коефіцієнти для такого перетворення можуть варіюватися залежно від вимог програми або умов зображення.

Крок 2. Знешумлення зображення.

Далі виконується знешумлення вхідного зображення, використовуючи Гаусове розмиття (Gaussian filter). Шум може створити хибні краї, що може поставити під загрозу точність процесу виявлення країв. Фільтр Гауса згладжує зображення, згортаючи його за допомогою ядра Гауса, ефективно зменшуючи високочастотний шум, зберігаючи чіткість країв. Математично ядро Гауса визначається як [32]:

$$G(x,\ y)\ =\ \frac{1}{2\pi\sigma^2}\cdot exp(-\frac{x^2+y^2}{2\sigma^2}), \tag{2.1}$$

де σ – стандартне відхилення, що контролює ширину розподілу Гауса;

π – математична константа Пі (приблизно 3,14159);

$x,\ y$ – просторові координати ядра.

Фільтр Гауса застосовується до кожного пікселя на зображенні шляхом ковзання ядра по всьому зображенню та отримання середнього зваженого значення інтенсивності сусідніх пікселів. Це означає, що він враховує яскравість навколишніх пікселів і надає більше значення ближчим. Отже, якщо ядро більше, воно включає більше пікселів у обчислення, що призводить до більш сильного ефекту розмиття зображення.

Крок 3. Визначення градієнта інтенсивності.

Після зменшення шуму алгоритм Кенні переходить до обчислення градієнта згладженого зображення. Градієнт вимірює, наскільки швидко змінюється інтенсивність у місці розташування кожного пікселя.

Алгоритм використовує концепцію похідних, як правило, оператор Собеля, щоб визначити як величину градієнта, так і орієнтацію для кожного пікселя. Величина градієнта вказує на силу зміни інтенсивності, а орієнтація градієнта визначає напрямок найкрутішої зміни.

Оператор Собеля – це техніка, яка використовується для визначення градієнта або швидкості зміни як у горизонтальному (зліва направо), так і у

вертикальному (зверху вниз) напрямках зображення. Він зазвичай використовується для виявлення країв під час обробки зображень [33].

Формули для операторів Собеля такі:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \tag{2.2}$$

де Gx – горизонтальний оператор Собеля;

$$G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}, \tag{2.3}$$

де Gy – вертикальний оператор Собеля.

Щоб обчислити градієнт зображення за допомогою оператора Собеля, ми проводимо ці маленькі матриці 3 × 3 (Gx та Gy) по зображенню піксель за пікселем. Для кожного пікселя ми виконуємо операцію згортки, яка передбачає множення значень навколишніх пікселів на відповідні значення в матрицях оператора Собеля, а потім підсумовуємо результати.

Застосовуючи оператор Собеля як у горизонтальному, так і у вертикальному напрямках, ми отримуємо два окремих градієнтних зображення. Поєднання цих двох градієнтних зображень допомагає визначити краї та межі вихідного зображення. Величина та напрямок градієнтів надають цінну інформацію про інтенсивність та напрямок країв зображення. Для розрахунку величини градієнта G та орієнтації θ для кожного пікселя використовуються такі формули [34]:

$$G = \sqrt{Gx^2 + Gy^2}, \quad (2.4)$$

$$\theta = atan2(Gy, Gx), \quad (2.5)$$

де Gx – горизонтальний оператор Собеля;

Gy – вертикальний оператор Собеля;

G – величина градієнту;

θ – орієнтація пікселя.

Крок 4. Заглушення немаксимумів (Non-maximum suppression).

Коли обчислено величину градієнта та орієнтацію кожного пікселя, можна перейти до критично важливого кроку заглушення немаксимумів. Цей крок ефективно стоншує краї та створює чіткіше представлення фактичних країв на зображенні.

Заглушення немаксимумів в алгоритмі виявлення країв Кенні працює, досліджуючи величину та орієнтацію градієнта кожного пікселя та порівнюючи його з сусідніми пікселями вздовж напрямку градієнта. Якщо величина градієнта центрального пікселя є найбільшою серед його сусідів, це означає, що цей піксель, ймовірно, є частиною краю, і ми зберігаємо його. Якщо ні, ми пригнічуємо його, встановлюючи його інтенсивність на нуль і видаляючи його, тим самим зробивши висновок, що це не крайовий піксель.

Крок 5. Подвійна порогова фільтрація.

На цьому етапі використовуються два порогові значення: нижній поріг і верхній поріг:

– пікселі з градієнтом, що перевищує верхній поріг, вважаються крайовими пікселями;

– пікселі, значення градієнта яких знаходиться між верхнім і нижнім порогами, вважаються потенційними крайовими пікселями, і вони включаються в результат тільки в тому випадку, якщо вони пов’язані з крайовими пікселями;

– пікселі з градієнтом нижче нижнього порогу вважаються шумом і відкидаються.

Крок 6 (останній). Трасування за порогом.

Останнім кроком алгоритму Кенні є трасування за порогом (відстеження краю за гістерезисом). Гістерезис означає «згадування минулого», щоб зробити наші краї більш точними та надійними. Цей крок має на меті зв’язати слабкі грані, які, ймовірно, є частиною справжніх ребер, із сильними.

Пікселі, які виявилися крайовими на попередньому етапі (після порогової фільтрації), зв’язуються разом для формування суцільних країв. Алгоритм з’єднує всі сусідні пікселі, якщо вони перевищують нижній поріг і пов’язані з пікселями, які перевищують верхній поріг. Це дозволяє уникнути пропуску важливих деталей і з’єднує фрагментовані краї.

Описавши роботу алгоритму Кенні можна зробити висновки щодо переваг та недоліків алгоритму (табл. 2.1). Результати роботи кожного з кроків представлено на рисунку 2.1 [35].

Таблиця 2.1 – Переваги та недоліки алгоритму Кенні

Переваги	**Недоліки**
Висока точність виявлення країв	Чутливість до параметрів
Мінімізація помилкових країв	Швидкість
З’єднання країв	

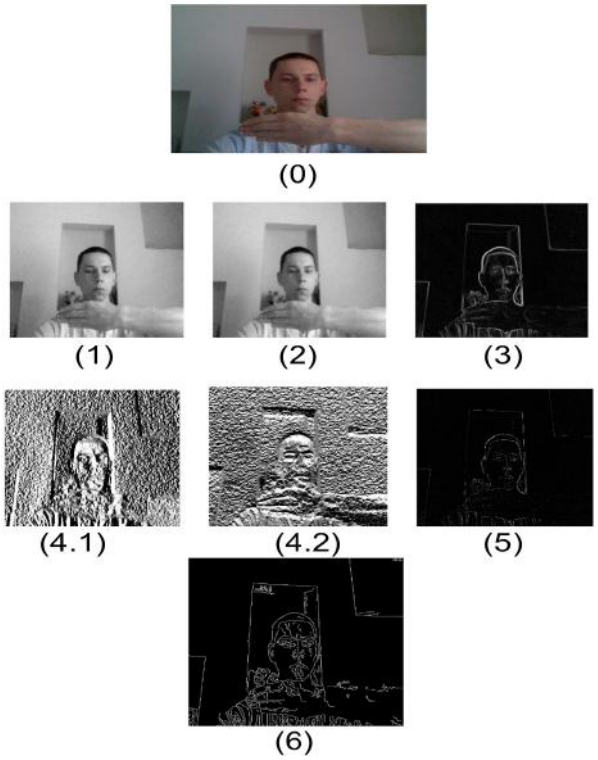

Рисунок 2.1 – Основні кроки алгоритму Кенні

Таким чином, алгоритм виявлення країв Кенні є широко використовуваним та потужним інструментом для ідентифікації країв на зображеннях. Його багатоступінчастий процес забезпечує точну локалізацію краю, низький рівень помилок і стійкість до шуму. З розвитком технологій важливість таких ефективних алгоритмів виявлення країв стає все більш очевидною, впливаючи на такі галузі, як розпізнавання об’єктів, медичне зображення, спостереження та доповнена реальність.

2.1.2 Методи з використанням нейронних мереж

Методи, що використовують нейронні мережі, відіграють ключову роль в сучасних системах відстеження рук, оскільки вони здатні вирішувати складні завдання комп’ютерного зору, включаючи розпізнавання жестів та точне відстеження рухів.

Нейронні мережі працюють на основі великої кількості навчальних прикладів, що дозволяє їм навчитися розпізнавати складні патерни, такі як положення рук або окремі жести. Основні етапи функціонування цих систем у трекінгу рук можна сформулювати наступним чином:

Перший. Збір і підготовка даних. Для навчання нейронних мереж необхідно зібрати великий набір зображень рук у різних положеннях і з різними жестами. Ці дані можуть бути як 2D, так і 3D, залежно від того, які завдання вирішує система.

Другий. Аналіз зображень. Нейронна мережа обробляє зображення, використовуючи шари нейронної мережі, які виявляють характерні особливості, такі як контури, кути, форми або інші важливі деталі рук.

Третій. Виявлення ключових точок. Для точного відстеження рук система виявляє ключові точки на руці, наприклад, суглоби пальців, долоню або зап’ястя. Це дозволяє визначати положення руки та її частин у просторі.

Четвертий. Класифікація жестів. Нейронна мережа класифікує рухи рук або статичні позиції в певні категорії (наприклад, алфавіт жестової мови або інші спеціальні рухи).

П’ятий. Інтерпретація і прогнозування. Мережі можуть прогнозувати траєкторію руху рук і відстежувати їхні зміни в реальному часі. Це дозволяє забезпечити точне взаємодію з користувачем в різних системах, таких як сурдопереклад, віртуальна реальність або інтерфейси управління жестами.

Існує багато типів нейронних мереж, але не всі можуть вирішити задачу розпізнавання жестів. Найкраще підходять CNN (Convolutional neural network), RNN (Recurrent neural network), DNN (Deep neural network) та GANs(Generative adversarial network).

Convolutional Neural Networks (CNN) – згорткові нейронні мережі, є одними з найефективніших архітектур глибокого навчання для аналізу зображень, включаючи трекінг рук і розпізнавання жестів. CNN здатні автоматично навчатися витягувати важливі ознаки з візуальних даних, що

робить їх зручними для розв’язання завдань, пов’язаних з комп’ютерним зором. Основні компоненти CNN: Convolutional Layer (згортковий шар), Activation Layer (активаційний шар), Pooling Layer (шар підсумовування), Fully Connected Layer (повнозв’язний шар) [36].

На рисунку 2.2 схематично представлено описані компоненти.

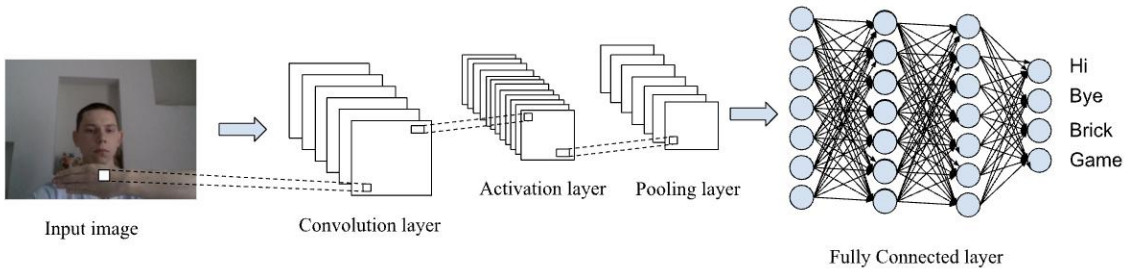

Рисунок 2.2 – Компоненти нейронної мережі CNN

Покроково принцип роботи CNN у трекінгу рук можна описати наступним чином:

– обробка вхідних зображень, де CNN отримує вхідні зображення рук або послідовності відеокадрів, на яких присутні руки в різних положеннях або рухах;

– вилучення ознак, де згорткові шари автоматично виявляють характерні ознаки, такі як контури пальців, долоні, форми рук і їхні відносні положення;

– зниження розмірності, де шари підсумовування зменшують розмір просторових даних, зберігаючи при цьому найважливіші ознаки;

– класифікація жестів, де після вилучення ознак повнозв’язні шари класифікують зображення, визначаючи, до якої категорії належить рука або жест.

Хоча цей тип нейронної мережі дуже гарно підходить для класифікування зображень, він не може допомогти із класифікацією саме жестів (тобто відео). Основним недоліком CNN для трекінгу рук є те, що вони в основному працюють із просторовими ознаками на окремих кадрах,

але мають труднощі з обробкою часових послідовностей або динамічних змін положення рук [37-39].

Recurrent neural network (RNN) – це клас нейронних мереж, які використовуються для обробки послідовностей даних. Основна особливість RNN полягає в їхній здатності зберігати інформацію про попередні етапи обробки та використовувати її для передбачення або класифікації поточних даних, що робить їх ефективними для роботи з часовими послідовностями, текстом, мовою, відео, рухами та іншими послідовними структурами [40].

Всі часові кроки в RNN використовують одні й ті самі параметри (ваги), що дозволяє зменшити кількість параметрів у порівнянні з іншими моделями. Це також допомагає зберігати послідовність у прогнозах на різних етапах.

Основні принципи роботи RNN:

– вхідні дані. Кожен крок часу отримує певний вхід, який передається до прихованого стану мережі(кадр відео);

– прихований стан. При кожному часовому кроці мережа оновлює свій прихований стан на основі поточного входу та попереднього стану;

– вихідні дані. На кожному кроці мережа може повертати вихід, який базується на поточному вході та прихованому стані.

На рисунку 2.3 показана частина рекурентної нейронної мережі A, що приймає вхідний сигнал x та виводить значення h. Цикл дозволяє передавати інформацію з одного кроку мережі на інший [41].

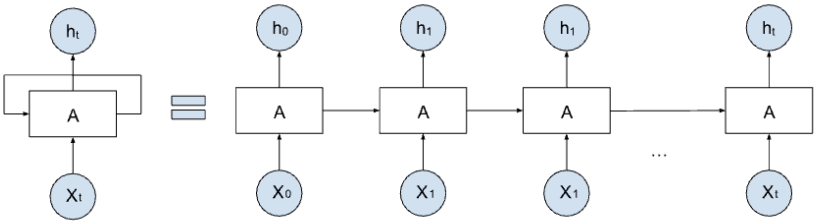

Рисунок 2.3 – Цикли рекурентної нейронної мережі

Однак цей клас нейронних мереж має декілька обмежень щодо застосування у задачі розпізнавання жестів і не тільки [42].

Перша проблема – «зникнення градієнтів». Оскільки під час навчання RNN інформація передається через багато часових кроків, існує проблема затухання градієнтів. Це означає, що коли градієнти, необхідні для оновлення вагів, стають дуже малими, навчання мережі стає неефективним, і вона не може запам’ятовувати далекі часові залежності.

Друга проблема – короткострокова пам’ять. Звичайні RNN добре працюють із короткими послідовностями, але коли потрібно обробляти тривалі часові ряди або послідовності з великою кількістю етапів, їхня здатність зберігати інформацію значно погіршується.

На щастя, ці проблеми було вирішено завдяки таким модифікаціям RNN, як LSTM (Long Short-Term Memory) та GRU (Gated Recurrent Unit).

LSTM-мережі здатні зберігати інформацію з попередніх кадрів або моментів часу і використовувати її для прогнозування та аналізу поточних станів.

LSTM відрізняється від звичайних RNN наявністю механізмів, що дозволяють зберігати та використовувати інформацію з минулих кроків навіть через тривалий час. Це вирішує проблему «зникнення градієнтів», яка виникає в базових RNN, коли дані з попередніх моментів часу стають «забутими».

Основними елементами LSTM є комірка пам’яті (memory cell) і система «воріт», які керують потоком інформації [43].

Комірка пам’яті (cell state) – відповідає за збереження інформації протягом довгих часових інтервалів. Комірка може «запам’ятовувати» важливі дані, необхідні для обробки на наступних кроках, і забувати не потрібну інформацію. Потік інформації через комірку контролюється спеціальними механізмами – «воротами».

Ворота. У LSTM використовуються три типи воріт, кожні з яких виконують певну функцію для керування інформацією:

– ворота забування (forget gate). Вони визначають, яку частину інформації з попереднього стану комірки слід «забути». Це критично для того, щоб уникати накопичення непотрібної або застарілої інформації. Ворота забування використовують сигмоїдну функцію активації, яка видає значення між 0 та 1. Чим ближче до 0 – тим більше інформації буде забуто;

$$f_t = \sigma\left(W_f \cdot \left[h_{t-1}, x_t\right] + b_f\right), \tag{2.6}$$

де f_t – значення воріт забування;

W_f – ваги;

h_{t-1} – попередній прихований стан;

x_t – поточний вхід;

σ – сигмоїдна функція.

– вхідні ворота (input gate). Вони контролюють, яку нову інформацію можна додати до комірки пам’яті. Як і в воротах забування, сигмоїдна функція визначає, яку частину нової інформації слід включити в комірку. Інформація, яка проходить через ці ворота, оновлює стан пам’яті;

$$i_t = \sigma\left(W_i \cdot \left[h_{t-1}, x_t\right] + b_i\right), \tag{2.7}$$

$$\mathrm{C}_t = tanh\left(W_c \cdot \left[h_{t-1}, x_t\right] + b_c\right), \tag{2.8}$$

де i_t – значення вхідних воріт;

C_t – нова інформація, яку можна додати до комірки пам’яті.

– вихідні ворота (output gate). Вони вирішують, яку частину інформації з комірки пам’яті потрібно передати на вихід і передати на наступний часовий крок. Вихідні ворота використовують комбінацію сигмоїдної активації та гіперболічної тангенс-функції для створення прихованого стану, який іде на вихід.

$$o_t = \sigma\left(W_o \cdot \left[h_{t-1}, x_t\right] + b_o\right), \tag{2.9}$$

$$h_t = o_t \cdot tanh\left(C_t\right), \tag{2.10}$$

де o_t – значення вихідних воріт;

h_t – прихований стан, який передається на наступний часовий крок;

C_t – оновлений стан пам’яті.

Таким чином процес роботи LSTM виглядає наступним чином [44]:

Крок 1. Обчислення воріт забування. Спочатку модель визначає, яку інформацію з попереднього стану потрібно забути.

Крок 2. Обчислення вхідних воріт. Модель вирішує, яку нову інформацію додати до стану комірки.

Крок 3. Оновлення стану комірки пам’яті. На цьому етапі відбувається оновлення комірки пам’яті на основі результатів роботи воріт забування та вхідних воріт. Стара інформація частково забувається, а нова – додається, що оновлює стан пам’яті.

Крок 4. Обчислення виходу. Нарешті, модель визначає, яку частину оновленого стану комірки потрібно передати на вихід для використання на наступному кроці або як поточний результат. Вихідні ворота контролюють, яку інформацію варто передати далі.

Deep neural network (DNN) – багатошарова штучна нейронна мережа, яка складається з кількох шарів нейронів і призначена для моделювання складних нелінійних залежностей у даних.

До основних компонентів CNN належать: Вхідний шар, Приховані шари та Вихідний шар. Кожен нейрон у прихованих і вихідних шарах отримує вхідні дані, обчислює зважену суму цих входів, додає до неї зміщення (bias) і застосовує функцію активації. Основна мета функції активації – додати нелінійність до моделі, що дозволяє мережі вирішувати складніші задачі [45].

Процес роботи нейрона можна описати таким чином:

$$z = W \cdot X + b, \tag{2.11}$$

$$a = f(z), \tag{2.12}$$

де W – матриця ваг нейрону;

X – вхідний вектор;

b – зміщення (bias);

$f(z)$ – функція активації, яка перетворює лінійну комбінацію входів z у кінцевий результат.

Навчання глибокої нейронної мережі здійснюється за допомогою алгоритму зворотного поширення помилки (backpropagation) у поєднанні з оптимізаційним алгоритмом, як правило, градієнтного спуску (gradient descent). До основних етапів навчання належать:

– прямий прохід (forward pass);

– обчислення втрат (loss calculation);

– зворотний прохід (backpropagation);

– оновлення ваг (weight update).

З одного боку, DNN може навчитися виділяти важливі ознаки для розпізнавання рук та їхнього руху без потреби вручну створювати фічі. Завдяки багатошаровій структурі, DNN може опрацьовувати складні взаємозв'язки між пікселями зображення і детально аналізувати руки та їхні пози.

З іншого боку, для навчання DNN необхідні великі й різноманітні набори даних для точного розпізнавання рухів рук у різних умовах (різні позиції рук, кути огляду, освітлення тощо). Якщо таких даних недостатньо, модель може бути недостатньо точною. Також DNN можуть бути занадто обчислювально інтенсивними. Це може вимагати потужних апаратних ресурсів, таких як GPU, особливо якщо мова йде про обробку відео у високій роздільній здатності.

Таким чином, DNN є ефективним інструментом для вирішення задачі стеження за руками, особливо якщо є достатньо ресурсів для тренування та інфраструктури для обробки даних. Однак для реальних додатків важливо враховувати обмеження, такі як обчислювальна складність і необхідність у великій кількості якісних навчальних даних.

2.1.3 Гібридні методи (глибина + нейронні мережі)

Гібридні методи, які поєднують технології обробки глибинних даних із нейронними мережами, є одним із найбільш перспективних підходів у задачах стеження за руками. Вони дозволяють поєднати переваги кожного підходу для покращення точності, швидкості та адаптивності системи [46].

Основна ідея гібридних систем полягає в тому, що вони використовують глибинні камери для збору тривимірної інформації про розташування рук і пальців, а нейронні мережі – для більш точного аналізу, класифікації та прогнозування рухів (рис 2.4). Глибинні дані забезпечують

додаткову інформацію про відстань до об’єктів, що дозволяє нейронним мережам працювати більш ефективно у тривимірному просторі.

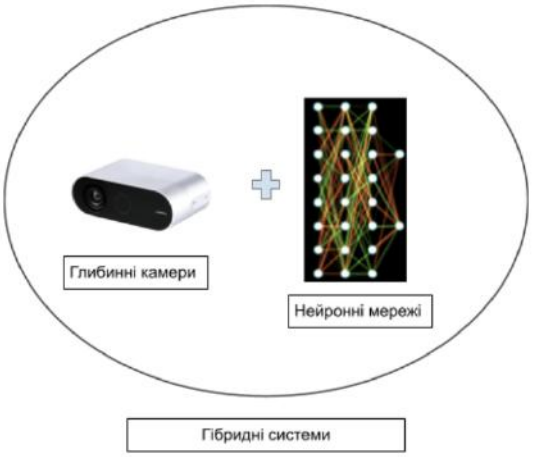

Рисунок 2.4 – Архітектура гібридних систем

До переваг гібридних методів методів можна віднести те, що в них глибинні камери надають дані про відстань до кожного пікселя, що допомагає з більшою точністю розпізнавати об’єкти, підвищену точність, покращену стійкість до шуму та роботу в умовах слабкого освітлення

Крім переваг, у даної системи є декілька недоліків, що можуть бути важливим фактором при реалізації такої задачі як відслідковування положення рук. Це обчислювальні ресурси, вартість та складність інтеграції.

Серед алгоритмів, що працюють за гібридним принципом, виділяють:

– 3D Convolutional Neural Networks (3D CNN);

– LSTM (Long Short-Term Memory) з глибинними даними;

– DenseNet (Dense Convolutional Network) з глибинними даними;

– Faster R-CNN з глибинними сенсорами.

Загалом, реалізація кожного із алгоритмів гібридних систем – це поєднання відслідковування знайдених ключових точок методами комп’ютерного зору та їхній аналіз методами з використанням нейронних

мереж. Тому такі системи, що поєднують 2 важливих завдання, мають дуже обіцяючий результат.

2.2 Існуючі системи жестової мови для сурдоперекладу

Задля того, щоб подолати проблему спілкування людей з порушеннями слуху та тими, хто не володіє жестовою мовою, було розроблено низку автоматизованих систем для перекладу жестової мови, які дозволяють забезпечити ефективний сурдопереклад у реальному часі.

SignAll – одна з найрозвиненіших систем автоматичного перекладу жестової мови, яка поєднує технології комп'ютерного зору, сенсорів і нейронних мереж для забезпечення точного та швидкого перекладу жестової мови на текст (рис. 2.5). Система була розроблена для того, щоб допомогти людям з порушеннями слуху спілкуватися з людьми, які не володіють жестовою мовою, шляхом автоматичного розпізнавання жестів та їхнього перекладу.

Рисунок 2.5 – Логотип та інтерфейс застосунку SignAll [47]

Ключові особливості SignAll:

– багатокамерна система. Застосунок використовує кілька камер для точного відстеження рухів рук, пальців і навіть обличчя користувача;

– комп’ютерний зір. Система застосовує алгоритми комп’ютерного зору для аналізу зображень і відео, що дозволяє виділяти ключові точки на руках і обличчі, необхідні для правильного розпізнавання жестів;

– нейронні мережі. SignAll використовує CNN для обробки зображень рук і жестів, а також RNN для розпізнавання послідовності жестів, що забезпечує точне визначення слів і речень у жестовій мові;

– реальний час. Система здатна миттєво перекладати жестові фрази на текст, що дозволяє використовувати її в живих діалогах або під час зустрічей;

– адаптація під різні мови жестів.

Одним із викликів для SignAll є необхідність індивідуального налаштування системи для кожного користувача. Точність перекладу може варіюватися залежно від унікальних особливостей жестів різних людей, тому система потребує певного часу для адаптації. Крім того, труднощі можуть виникати під час розпізнавання жестів у складних умовах, таких як недостатнє освітлення або наявність великої кількості людей, що ускладнює роботу системи.

Microsoft Kinect для жестової мови – система, що спочатку розроблена для ігрової індустрії, та згодом знайшла своє застосування в багатьох інших сферах, зокрема для автоматичного перекладу жестової мови. Технологія Kinect використовує глибинну камеру та інфрачервоні сенсори для захоплення рухів у тривимірному просторі, що дозволяє точно відстежувати положення рук, пальців і тіла користувача [48]. На рисунку 2.6 можна побачити як виглядає сам сенсор.

Рисунок 2.6 – Сенсор руху Microsoft Kinect

Microsoft Kinect використовує глибинну камеру, яка працює за принципом вимірювання часу польоту (ToF). Вона випромінює інфрачервоне світло, яке відбивається від об’єктів, і вимірює час, за який світло повертається до камери. Це дозволяє отримати точні дані про відстань до кожної точки в просторі, створюючи тривимірне зображення сцени.

Одна з основних переваг Kinect полягає в здатності відстежувати скелет людини в реальному часі. Система ідентифікує ключові точки на тілі (суглоби), що дозволяє точно аналізувати рухи користувача.

Завдяки поєднанню глибинної камери та скелетного трекінгу, Kinect може розпізнавати широкий діапазон жестів, що робить його ефективним інструментом для розпізнавання жестової мови.

Також Kinect може використовуватися не лише в локальних системах, а й у хмарних сервісах для обробки великої кількості даних у режимі реального часу. Це дозволяє інтегрувати систему в різні платформи для покращення доступності сурдоперекладу.

Однак існують декілька проблем, з якими може зіштовхнутися система Kinect.

Перша – робота в умовах з поганим освітленням або за наявності об’єктів, що заважають трекінгу. Друга – обмежена точність пальців. Третя – необхідність певного простору. Для ефективної роботи Kinect потребує достатнього простору для трекінгу рук і тіла, що може бути незручним у маленьких або обмежених просторах.

Інтерфейс застосунка, що використовує Microsoft Kinect можна побачити на рисунку 2.7.

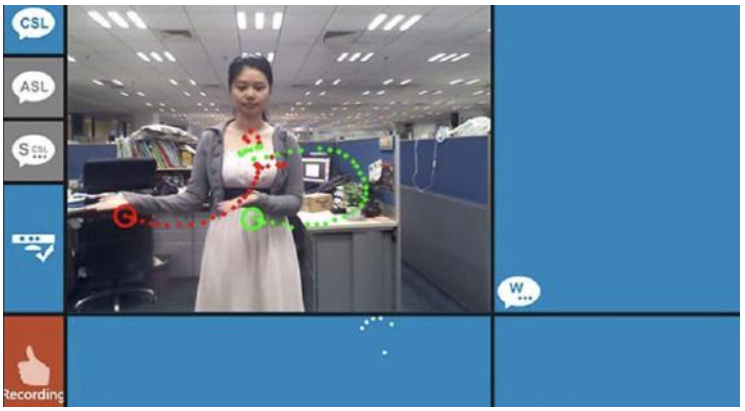

Рисунок 2.7 – Інтерфейс застосунку, що використовує Microsoft Kinect

Hand Talk – це додаток для мобільних пристроїв, що отримує дані та перекладає їх на Libras, мову жестів глухих спільнот міської Бразилії [49].

Hand Talk працює з трьома основними джерелами: аудіо, текстом та зображеннями. Коли голос вловлюється мобільним телефоном, програма перетворює звук на мову жестів за допомогою 3D-аватара (рис. 2.8), який з’являється на екрані мобільного телефону. Цю функцію можна використовувати у розмовах, лекціях та у багатьох інших ситуаціях.

Рисунок 2.8 – Використання 3D-аватара для відображення мови жестів

За допомогою функції тексту користувач з вадами слуху може скопіювати повідомлення, отримані від місцевого оператора або вебсайту, вставити його в програму та перекласти на Libras.

Функцію зображення можна використовувати для перекладу вмісту журналів, газет, книг або вивісок. Користувач фотографує фразу або слова, а програма перетворює їх на Libras.

API системи може використовуватися на інших операційних системах та платформах, роблячи контент доступним для людей, які не знають жодної розмовної мови. Потенційний контент включає все, що завгодно: від новинних або урядових вебсайтів, банківських терміналів та інших фінансових установ до супермаркетів і музеїв.

Застосунок використовує кілька основних технологій і алгоритмів, щоб забезпечити переклад тексту та голосу на жестову мову. Це алгоритми синтаксичного та семантичного аналізу для аналізу введенного тексту, виділяючи ключові частини мови, визначаючи контекст і розпізнаючи значення слів і фраз; моделі навчання з підкріпленням – для покращення точності перекладу (з часом модель навчається на основі введень користувача, що покращує її здатність до адаптації та інтерпретації різних типів введення); алгоритми розпізнавання мови (Speech-To-Text); алгоритми глибокого злиття даних (Data Fusion) тощо.

Таким чином, Hand Talk – це видатний та перспективний інструмент соціальної інтеграції, що дозволяє людям з обмеженими можливостями у Бразилії більше брати участь у повсякденному житті.

3 ДОСЛІДЖЕННЯ ОПТИМАЛЬНОГО МЕТОДУ СТЕЖЕННЯ ЗА РУКАМИ

3.1 Моделювання та вибір методу для системи сурдоперекладу

3.1.1 Математична модель руху руки

Математична модель руху руки базується на формалізації підходу до опису положення та руху руки у відкритому просторі за допомогою математичних рівнянь та обчислювальних методів. Рух руки – складний процес, який включає в себе рух не тільки всієї руки, але і окремо кожного пальця. Тому ефективна математична модель допомагає у відстеженні жестів, розпізнаванні мови та створенні віртуальних інтерфейсів.

Математичну модель руки можна побудувати за багатьма факторами: кінематика, інверсна кінематика, модель скелету руки, розпізнавання жестів та траєкторій.

Розглянемо кінематичну модель з використанням прямої кінематики. Вона використовує кінематичний ланцюг для опису положення та орієнтації сегментів руки. У цій моделі поділяється рука на сегменти (передпліччя, кисть, пальці) і вважається, що рух кожного сегмента задається обертанням у суглобах. Така модель часто використовується для відстеження положення кінцівок у системах трекінгу [50].

Долоня є центральним елементом руки, який забезпечує базу для руху пальців. Вона складається з кількох важливих елементів, що враховуються в кінематичній моделі:

– основна частина долоні (метакарпус) – фіксована основа для моделювання пальців;

– пальці – кожен палець складається з трьох фаланг і чотирьох суглобів: MCP – біля основи пальця, PIP, DIP, TIP (кінчик пальця);

– кінематична модель долоні включає 21 ступінь свободи (DoF): 5 для орієнтації кожного пальця (4 пальці + великий палець); 6 для положення та орієнтації долоні в просторі.

Для створення математичної моделі долоні необхідно врахувати такі аспекти:

– положення долоні у просторі, яка має 6 ступенів свободи (3 для положення: x, y, z, і 3 для орієнтації: α, β, γ);

– структура пальців, де палець моделюється як кінематичний ланцюг, який починається від метакарпальної частини (основи долоні);

– анатомічні обмеження, бо суглоби пальців мають обмеження на кути обертання. Наприклад, MCP суглоби можуть згинатися до ~90°, тоді як PIP і DIP суглоби мають більший діапазон руху (до 120°).

Палець складається з трьох основних сегментів:

– проксимальна фаланга (l1) – сегмент, що найближче до долоні;

– средняя фаланга (l2) – середній сегмент пальця;

– дистальна фаланга (l3) – кінцевий сегмент, на якому розташований кінчик пальця.

Кожен суглоб (MCP, PIP, DIP) визначається кутом обертання θ, що описує згинання і розгинання суглоба, та обмеженнями кута. Суми довжин всіх трьох фаланг (l1 + l2 + l3) визначають максимальну довжину пальця, але його реальне положення залежить від кутів у кожному суглобі.

Основна ідея моделювання долоні полягає в тому, що положення кожного пальця визначається положенням долоні та локальними параметрами, такими як кути обертання суглобів і довжини сегментів пальців.

Долоня моделюється як жорстке тіло, яке може рухатися в просторі. Всі пальці прикріплені до основи долоні й утворюють кінематичні ланцюги. Модель описується за допомогою матриць гомогенних перетворень T, які поєднують обертання і трансляції.

Загальна формула для гомогенної матриці виглядає наступним чином:

$$T_{joint} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & l\cos\theta \\ \sin\theta & \cos\theta & 0 & l\sin\theta \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (3.1)$$

де θ – кут обертання в суглобі;

l – довжина сегмента.

Для кожного пальця використовуються глобальні (що враховують положення долоні), та локальні перетворення (враховують обертання та зсуви в MCP, PIP і DIP суглобах). Пряма кінематика дозволяє розрахувати точні координати кінчика пальця (x,y,z) в просторі, що є важливим для задач сурдоперекладу, де кожен жест залежить від точного положення долоні та пальців [51].

Отже, головна формула для прямої кінематики має такий вигляд:

$$p_{finger\_tip} = T_{palm} \cdot T_{finger\_base} \cdot T_{MCP} \cdot T_{PIP} \cdot T_{DIP} \cdot p_{ref}, \quad (3.2)$$

де p_{finger_tip} – локальна позиція кінчика пальця в просторі;

T_{palm} – перетворення долоні;

T_{finger_base} – положення основи пальця відносно долоні;

T_{MCP}, T_{PIP}, T_{DIP} – перетворення для кожного суглоба пальця;

p_{ref} – референсна точка, зазвичай $[0, 0, 0, 1]^T$.

Для всієї долоні враховуються всі пальці, кожен з яких описується аналогічною моделлю. Положення кожного пальця визначається через множення відповідних матриць:

$$p_{hand} = T_{palm} \cdot \prod_{i=1}^{5} \left(T_{finger\_base} \cdot T_{MCP} \cdot T_{PIP} \cdot T_{DIP} \right) \cdot p_{ref}, \quad (3.3)$$

де p_{hand} – глобальна позиція всієї долоні.

Таким чином, знаючи phand у конкретний проміжок часу, можна точно визначити положення кожного пальця та його фаланг, використовуючи принципи прямої кінематики. Це дозволяє прогнозувати рухи руки, виявляти її позицію для подальшої обробки або розпізнавання жестів.

3.1.2 Фільтрування траєкторій та згладжування сигналів

Для покращення точності стеження за руками у системах сурдоперекладу жестових мов, фільтрування траєкторій і згладжування сигналів є важливим етапом для усунення шумів та неточностей, які можуть виникати під час рухів. Коли ми будуємо траєкторії на основі математично розрахованих точок положень рук, це може бути схильне до різноманітних помилок через багатий набір даних, таких як шум з сенсорів, обмеження апаратного забезпечення чи рухи, які неможливо передбачити.

Щоб подолати ці проблеми, застосовуються різні підходи фільтрації, щоб згладити рухи та отримати більш стабільні траєкторії. Один із них – фільтр Калмана, що дозволяє не лише зменшити шум, але й робить оцінки позиції руки в реальному часі більш точними, комбінуючи прогноз і вимірювання.

Враховуючи, що рухи руки можуть бути невизначеними або містити помилки, цей метод дозволяє отримати більш точну оцінку її позиції шляхом врахування як поточних вимірювань, так і прогнозованих значень на основі попереднього стану.

Фільтр Калмана працює в два етапи: прогнозування і коригування . Для прогнозування наступного стану системи (наприклад, позиція та швидкість руки) та обчислення прогнозованого коваріаційного значення використовується формула 3.4 та 3.5 відповідно [52]:

$$x_k = Ax_{k-1} + Bu_k, \tag{3.4}$$

$$P_k = AP_{k-1}A^t + Q, \tag{3.5}$$

де x_k – прогнозоване значення стану на поточному кроці k;

A – матриця стану;

x_{k-1} – оцінка стану попереднього кроку;

B – матриця керування;

u_k – вектор керування;

P_k – прогнозоване значення коваріації;

Q – матриця процесу шуму.

Для коригування – розрахунок фактичного стану на основі вимірювань та оновлення матриці коваріації розраховуються за формулами 3.6 та 3.7 відповідно:

$$x_k = x_k + K_k\left(y_k - Hx_k\right), \tag{3.6}$$

$$P_k = \left(I - K_kH\right)P_k, \tag{3.7}$$

де K_k – коефіцієнт Калмана;

y_k – вимірювання позиції руки;

I – одинична матриця;

H – матриця спостереження, що пов’язує вимірювання зі станом.

Використання фільтра Калмана для згладжування траєкторій руки дозволяє забезпечити стабільність і точність обчислень, навіть у присутності шуму та непередбачених змін [53]. Завдяки поєднанню прогнозів і фактичних вимірювань, система ефективно оцінює положення Phand у реальному часі. Це критично важливо для жестових мов, оскільки навіть незначні похибки можуть вплинути на точність розпізнавання жестів. Метод є адаптивним, універсальним і підходить для роботи з різними джерелами даних, що робить його ключовим інструментом у задачах стеження за руками.

3.1.3 Формалізація та вибір раціонального методу

Вибір методу для розпізнавання жестової мови базується на аналізі існуючих підходів, їх ефективності, вартості реалізації та можливості подальшого вдосконалення. У рамках дослідження було розглянуто три основні напрями: класичні алгоритми комп’ютерного зору, сучасні підходи з використанням глибоких нейронних мереж, а також комбінації цих методів.

Класичні підходи, такі як сегментація за кольором шкіри або контурний аналіз, мають низькі обчислювальні вимоги та добре працюють у контрольованих умовах. Проте вони є чутливими до змін освітлення, кольору шкіри та складності фону, що обмежує їхню практичну застосовність.

Глибокі нейронні мережі, зокрема CNN, показали високу ефективність у виявленні ознак з візуальних даних, тоді як RNN, і особливо їх модифікація LSTM, ефективно працюють із часовими послідовностями. Ці підходи забезпечують високу точність навіть у складних умовах, та вимагають

значних обчислювальних ресурсів і часу на навчання.

Комбінації методів, наприклад, використання CNN для виявлення ознак і LSTM для аналізу руху, об’єднують переваги обох підходів і демонструють відмінні результати у задачах розпізнавання жестів. Однак повноцінна реалізація таких систем може бути затратною по ресурсам.

У результаті аналізу було обрано підхід, який передбачає використання LSTM моделі для розпізнавання послідовностей жестів у поєднанні з 2D методами комп’ютерного зору для обробки вхідних даних. LSTM моделі краще всього підходять для задачі сурдоперекладу, оскільки дозволяють ефективно враховувати контекст руху та відстежувати часові залежності, а також наявність ресурсів. Ця особливість є ключовою для розуміння жестів, які складаються з плавних рухів і зміни положення рук у просторі.

2D методи комп’ютерного зору було обрано з огляду на їхню доступність і практичність. Зокрема, використання таких інструментів, як OpenCV або MediaPipe, дозволяє точно визначати положення рук без потреби в дорогих 3D-камерах або спеціалізованому обладнанні. Такі методи працюють на основі зображень зі звичайних камер, що робить їх дешевим і зручним рішенням для більшості застосувань. Крім того, вони легко інтегруються з існуючими бібліотеками для машинного навчання, такими як PyTorch або TensorFlow.

Ще одним важливим аргументом на користь цього підходу є його економічна ефективність і доступність. Що LSTM моделі, що 2D методи комп’ютерного зору – є безкоштовними завдяки широкодоступним відкритим бібліотекам та інструментам.

Таким чином, запропонований підхід є раціональним вибором для розпізнавання жестів у системах сурдоперекладу. Він забезпечує високу точність і надійність при мінімальних фінансових витратах, що робить його перспективним для практичної реалізації.

3.2 Постановка експериментальних завдань

У цьому розділі визначено ключові етапи та цілі експериментальної частини дослідження, спрямовані на розробку та вдосконалення системи розпізнавання жестів для сурдоперекладу. Основна мета експериментів – створення ефективної моделі для аналізу рухів рук та жестів із можливістю подальшого вдосконалення.

Перше завдання – реалізація системи, що здатна в реальному часі аналізувати положення рук у просторі. Ця система має виконувати попередню обробку візуальних даних, виділяти ключові ознаки, що характеризують положення та рухи рук, забезпечувати точну сегментацію та трекінг навіть у складних умовах, таких як неоднорідний фон або змінне освітлення.

Наступним завданням є розробка моделі, що працює із часовими послідовностями, зокрема з рухами рук, що формують жести. У процесі побудови моделі необхідно визначити оптимальну архітектуру для роботи з часовими залежностями, реалізувати можливість роботи з даними, що можуть містити шуми або бути неповними, а також забезпечити узгодженість між вхідними візуальними даними та вихідними категоріями жестів.

Для досягнення високої точності моделі провести ряд експериментів із різними параметрами. Серед них – кількість шарів у моделі (для визначення оптимальної глибини, яка забезпечує баланс між складністю та продуктивністю), вибір функцій активації для кожного шару, з метою виявлення найбільш ефективних варіантів для аналізу жестів, налаштування додаткових параметрів, таких як розмірність простору ознак та об’єм вхідних даних.

Останнім, та не менш важливим етапом є оцінка працездатності моделі та застосунку – вимірювання точності класифікації жестів на тестових наборах даних, аналіз швидкодії системи в реальному часі, оцінка стійкості до зовнішніх факторів (зміна освітлення, фонів, положення камери тощо).

За результатами тестування необхідно провести порівняння різних конфігурацій моделі, визначити їх переваги та недоліки. На основі отриманих даних запропонувати шляхи вдосконалення системи.

3.3 Обґрунтування вибору середовища програмної реалізації

Для розробки та тестування системи розпізнавання жестів було обрано низку технологій і інструментів, що забезпечують зручність, ефективність та високу продуктивність. Основні компоненти середовища програмної реалізації включають Jupyter Notebook, мову програмування Python, бібліотеки OpenCV, MediaPipe, Keras і TensorFlow.

Jupyter Notebook було обрано як основне середовище розробки завдяки його інтерактивності та зручності для досліджень у галузі машинного навчання та комп’ютерного зору [54]. Одними із найбільших його плюсів можна підкреслити можливість виконання коду поетапно з миттєвим отриманням результатів, інтеграцію текстових пояснень, графіків і коду, що забезпечує зручну документацію та аналіз експериментів, та підтримку численних бібліотек Python.

Python є основною мовою програмування, обраною для реалізації проекту. Вона ідеально підходить для задач комп’ютерного зору та машинного навчання завдяки простоті синтаксису, що сприяє швидкій розробці прототипів та великій екосистемі бібліотек, які охоплюють різноманітні задачі [55].

OpenCV використовується для обробки зображень і відео, що є основою для аналізу рухів рук. Бібліотека надає широкий набір функцій для обробки зображень: фільтрацію, сегментацію, трекінг об’єктів тощою. Має високу продуктивність і підтримка реального часу, та відкрите ліцензування,

що робить його безкоштовним для використання [56].

MediaPipe застосовується для трекінгу ключових точок рук. Ця бібліотека обрана через те, що надає можливість точного визначення положення рук і пальців у 2D-просторі та підтримку інтеграції з іншими бібліотеками Python для подальшого аналізу даних [57].

TensorFlow обрано як основну платформу для машинного навчання. Вона має високу продуктивність і підтримка GPU/CPU для прискорення навчання моделей, потужні засоби для оптимізації, відлагодження та візуалізації, підтримку розширених функцій для роботи з часовими послідовностями, що є критично важливим для задач розпізнавання жестів [58].

Keras використовується як високорівневий API для створення нейронних мереж [59]. Він має інтуїтивно зрозумілий інтерфейс, що дозволяє швидко будувати і тестувати моделі, підтримку як послідовних та функціональних API для створення складних архітектур, інтеграцію з TensorFlow, що забезпечує доступ до широкого спектра інструментів і можливостей.

Таким чином, вибір зазначених інструментів і технологій дозволяє ефективно реалізовувати й тестувати системи комп’ютерного зору та машинного навчання в контексті розпізнавання жестів. Кожен компонент виконує чітко визначену роль, а їх інтеграція забезпечує гнучкість і масштабованість розробленого рішення.

3.4 Налаштування системи для трекінгу рухів

Процес конфігурації системи для відстеження рухів рук можна розбити на такі ключові етапи – налаштування Mediapipe, знаходження та графічну

візуалізацію ключових точок рук на зображенні з камери.

Щоб налаштувати систему Mediapipe було використано модуль Holistic для відслідковування всього положення тіла. Перед тим як передати зображення на обробку до моделі Holistic, потрібно перевести зображення , що було отримано завдяки OpenCV із камери, з BGR до RGB формату для зменшення похибок при обробці даних (рис. 3.1). Після обробки необхідно повернути формат до BGR для подальшої роботи зображення із OpenCV API.

```
mp_holistic = mp.solutions.holistic
mp_drawing = mp.solutions.drawing_utils

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
```

Рисунок 3.1 – Налаштування Mediapipe середовища

На кожному кадрі проводиться аналіз для визначення ключових точок на руках. Ці точки є анатомічним орієнтиром, таким як кінчики пальців, суглоби або центр долоні. На рисунку 3.2 представлена функція, що визначає координати ключових точок у 2D-просторі та фільтрує результати для зменшення шуму й виключення помилкових спрацьовувань (що не відповідають ключовим точкам рук).

```
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)
    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, lh, rh])
```

Рисунок 3.2 – Виділення ключових точок рук на зображенні

Для візуалізації структури руки ключові точки з’єднуються лініями. Це допомагає представити руку як сукупність сегментів, що імітують кістки. На рисунку 3.3 представлена функція, що будує граф, де вузлами є ключові точки, а ребрами – лінії між ними. З’єднання ключових точок дозволяє в реальному часі відображати рухи рук у формі, зрозумілій як для людини, так і для подальших алгоритмів аналізу.

```
def draw_landmarks(image, results):
    mp_drawing.draw_landmarks(image, results.pose_landmarks, mp_holistic.POSE_CONNECTIONS,
                              mp_drawing.DrawingSpec(color=(255, 247, 0), thickness=1, circle_radius=3),
                              mp_drawing.DrawingSpec(color=(172, 0, 0), thickness=2, circle_radius=1))
    mp_drawing.draw_landmarks(image, results.left_hand_landmarks, mp_holistic.HAND_CONNECTIONS,
                              mp_drawing.DrawingSpec(color=(255, 247, 0), thickness=1, circle_radius=3),
                              mp_drawing.DrawingSpec(color=(172, 0, 0), thickness=2, circle_radius=1))
    mp_drawing.draw_landmarks(image, results.right_hand_landmarks, mp_holistic.HAND_CONNECTIONS,
                              mp_drawing.DrawingSpec(color=(255, 247, 0), thickness=1, circle_radius=3),
                              mp_drawing.DrawingSpec(color=(172, 0, 0), thickness=2, circle_radius=1))
```

Рисунок 3.3 – Побудова скелету руки

Таким чином, запропонований підхід до налаштування системи трекінгу рук розбиває задачу на три основні функції: налаштування Mediapipe, виділення ключових точок і з’єднання їх лініями. Такий модульний підхід дозволяє легко відлагоджувати кожен компонент окремо, а також забезпечує гнучкість і масштабованість системи. Результат налаштованої працюючої системи представлений на рисунку 3.4.

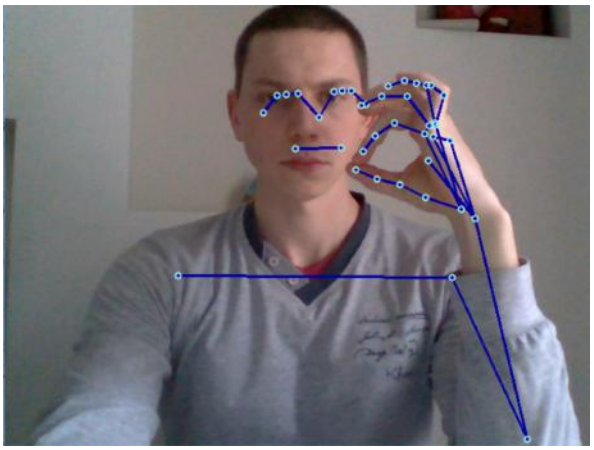

Рисунок 3.4 – Налаштована система для відслідковування рухів

3.5 Набір даних для тренування моделей сурдоперекладу

Для тренування моделей розпізнавання жестів було сформовано спеціалізований набір даних, що включає десять жестів (слів). Це слова: bicycle, brick, bye, casino, game, happy birthday, hi, house, rain, thunder. Кожному жесту відповідає 40 відео, які включають:

– відео з публічних датасетів жестів, що забезпечують різноманітність рухів і кутів зйомки (Microsoft Research Cambridge-12 Gesture Dataset, Chalearn Gesture Dataset, RWTH-PHOENIX-Weather 2014 T) (рис. 3.5) [60, 61];

– власноруч записані відео, що дозволяють врахувати унікальні умови освітлення та фону (рис. 3.6).

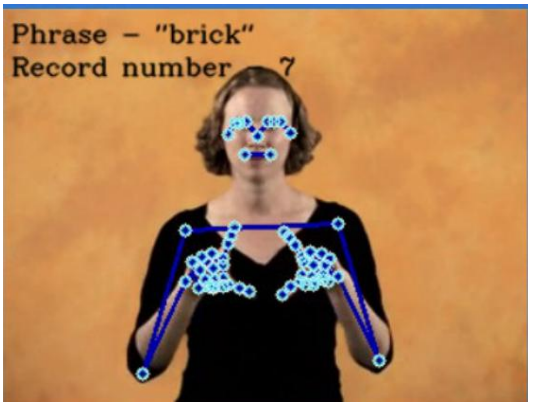

Рисунок 3.5 – Приклад виконання жесту «brick» із відкритого датасету

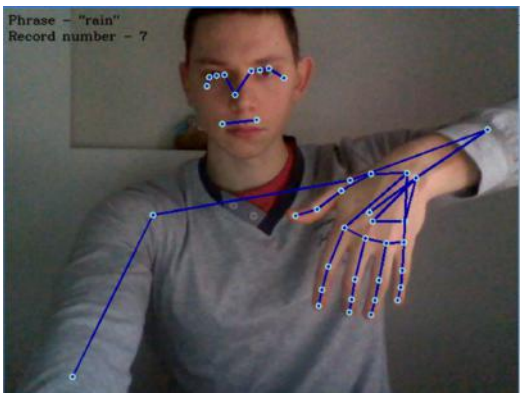

Рисунок 3.6 – Приклад виконання жесту «rain» із власного датасету

Кожне відео складається з 30 кадрів, які аналізуються системою трекінгу рук, описаною раніше. Для кожного кадру визначаються ключові точки рук (кінчики пальців, суглоби, центр долоні), і координати цих точок зберігаються у вигляді окремого файлу.

Таким чином, з одного відео створюється 30 файлів, які містять інформацію про просторове положення ключових точок на кожному кадрі.

Для кожного жесту, що складається із 40 відео, генерується 1200 файлів (40 відео × 30 кадрів). Загальний набір даних формується із координат, які чітко описують положення рук у кожний момент часу. Ці дані тепер можна використати для навчання нейронної мережі.

На виході отримано датасет, що складається з багатовимірних точок у просторі, які ідентифікують жести у часовому контексті. Тренувальна вибірка складає 80% датасету, а тестувальна – 20%. Такий формат забезпечує ефективну підготовку до навчання моделей, орієнтованих на розпізнавання жестів. Набір є гнучким і може бути розширений для додаткових жестів або включення нових відео, що підвищить його загальну точність і універсальність.

3.6 Створення різних моделей для розпізнавання жестів

Представлено чотири різні архітектури нейронних мереж для задачі розпізнавання жестів. Кожна з моделей має унікальну структуру, що дозволяє оцінити їх ефективність у навчанні та точності.

Модель 1 – базова LSTM модель (рис. 3.7). Вона включає три LSTM-шари зі зростаючою і спадною кількістю нейронів. Орієнтована на ефективне захоплення часових залежностей у послідовностях даних.

```
model = Sequential()
model.add(LSTM(64, return_sequences=True, activation='relu', input_shape=(30, 258)))
model.add(LSTM(128, return_sequences=True, activation='relu'))
model.add(LSTM(64, return_sequences=False, activation='relu'))
model.add(Dense(64, activation='relu'))
model.add(Dense(32, activation='relu'))
model.add(Dense(actions.shape[0], activation='softmax'))
```

```
model.compile(optimizer='Adam', loss='categorical_crossentropy', metrics=['categorical_accuracy'])
```

Рисунок 3.7 – Базова LSTM-модель

Вона налічує 3 LSTM-шари для глибокого аналізу часових залежностей, 2 щільних (Dense) шари для зменшення розмірності та генерації більш абстрактних ознак, і Softmax-активацію на вихідному шарі для багатокласової класифікації.

Для навчання моделі використано оптимізатор Adam, який забезпечує швидку та стабільну збіжність завдяки адаптивній швидкості навчання для кожного параметра. Також використано функцію втрат для багатокласової класифікації та додано метрику оцінки точності класифікації.

Модель 2 – спрощена LSTM-модель (рис. 3.8). Вона використовує меншу кількість нейронів і шарів для оцінки продуктивності на легших конфігураціях.

```
model2 = Sequential()
model2.add(LSTM(32, return_sequences=True, activation='tanh', input_shape=(30, 258)))
model2.add(LSTM(32, return_sequences=False, activation='tanh'))
model2.add(Dense(32, activation='relu'))
model2.add(Dense(actions.shape[0], activation='softmax'))
```

```
C:\Users\ADMIN\AppData\Local\Programs\Python\Python310\lib\site-packages\keras\src\layers\rnn\rnn.py:204
t_dim` argument to a layer. When using Sequential models, prefer using an `Input(shape)` object as the f
  super().__init__(**kwargs)
```

```
model2.compile(optimizer='RMSprop', loss='categorical_crossentropy', metrics=['categorical_accuracy'])
```

Рисунок 3.8 – Спрощена LSTM-модель

Ця модель має 2 LSTM-шари зі стабільною кількістю нейронів, 1 Dense-шар перед виходом для спрощення ознак, функцію активації tanh для стабільності градієнтів та оптимізатор RMSprop для більшої стабільності навчання на малих наборах даних.

Модель 3 – гібридна модель із GRU (рис. 3.9), що поєднує LSTM і GRU, щоб оцінити переваги різних рекурентних шарів.

```
model3 = Sequential()
model3.add(LSTM(128, return_sequences=True, activation='relu', input_shape=(30, 258)))
model3.add(GRU(64, return_sequences=False, activation='relu'))
model3.add(Dense(128, activation='relu'))
model3.add(Dense(64, activation='relu'))
model3.add(Dense(actions.shape[0], activation='softmax'))
```

```
model3.compile(optimizer='Adam', loss='categorical_crossentropy', metrics=['categorical_accuracy'])
```

Рисунок 3.9 – Гібридна модель із GRU

Перший шар моделі – LSTM, забезпечує аналіз довготривалих залежностей. GRU-шар додає ефективність у моделюванні короткотермінових залежностей. Більша кількість нейронів у Dense-шарах призначена для складніших ознак. Використовує Adam-оптимізатор для більш швидшої збіжності.

Модель 4 – Удосконалена багаторівнева LSTM-модель (рис. 3.10), побудована для глибшого аналізу часових залежностей із вбудованими методами запобігання перенавчанню, такими як Dropout і L2-регуляризація.

```
model4 = Sequential()

model4.add(LSTM(128, return_sequences=True, activation='relu',
               input_shape=(30, 258), kernel_regularizer=l2(0.01)))
model4.add(Dropout(0.2))

model4.add(LSTM(256, return_sequences=True, activation='relu', kernel_regularizer=l2(0.01)))
model4.add(Dropout(0.3))

model4.add(LSTM(128, return_sequences=False, activation='relu', kernel_regularizer=l2(0.01)))

model4.add(Dense(128, activation='relu'))
model4.add(Dropout(0.2))
model4.add(Dense(64, activation='relu'))

model4.add(Dense(actions.shape[0], activation='softmax'))
```

```
C:\Users\ADMIN\AppData\Local\Programs\Python\Python310\lib\site-packages\keras\src\layers\rnn\rnn.py
t_dim` argument to a layer. When using Sequential models, prefer using an `Input(shape)` object as t
  super().__init__(**kwargs)
```

```
model4.compile(optimizer='Adam', loss='categorical_crossentropy', metrics=['categorical_accuracy'])
```

Рисунок 3.10 – Удосконалена багаторівнева LSTM-модель

Запропонована модель має 3 LSTM-шари: перший і другий мають більшу кількість нейронів (128 і 256) для захоплення складних часових

залежностей, третій шар завершує часовий аналіз із 128 нейронами.

Dropout-регуляризація – додає випадкову «деактивацію» нейронів у кожному шарі, щоб зменшити ймовірність перенавчання.

L2-регуляризація – зменшує величину ваг для запобігання перенавчанню.

Щільні шари забезпечують поступове зменшення розмірності до вихідного простору.

Якщо порівнювати цю модель із базовою (модель 1), то тут більш глибока структура (окрім 3-х LSTM-шарів тут присутні додаткові Dropout-рівні), додано Dropout і L2 для запобігання перенавчанню, та в цілому – більше параметрів моделі. І таким чином, удосконалений підхід дозволяє краще аналізувати складні залежності, але вимагає більше ресурсів для навчання.

3.7 Навчання моделей та оцінка результатів розпізнавання жестів

Для правильності експерименту та забезпечення коректності порівняння, всі моделі були навчені на однаковому наборі даних.

Кількість епох було взято 100. Це число також обрано дослідницьким шляхом, адже майже всі моделі за цієї кількості не дуже піддавалися перенавчанню.

Частка даних для тренувального та тестового наборів склала 80% та 20% відповідно. Навчання проводилося на локальному комп’ютері із використанням CPU.

На рисунках 3.11 – 3.14 представлено графіки точності та втрат для кожної з епох навчання відповідної моделі.

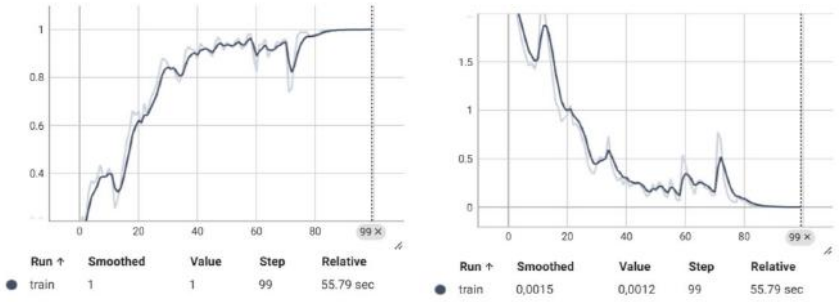

Рисунок 3.11 – Навчання базової LSTM-моделі

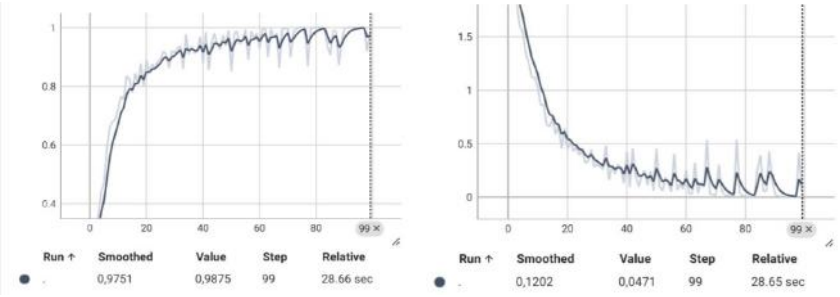

Рисунок 3.12 – Навчання спрощеної LSTM-моделі

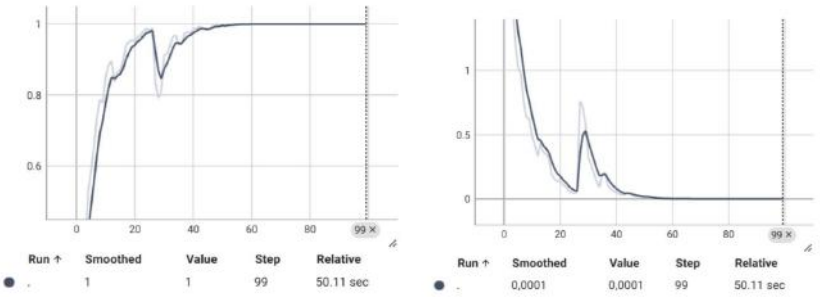

Рисунок 3.13 – Навчання гібридної моделі із GRU

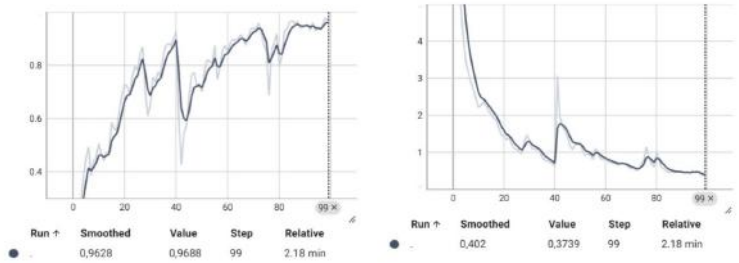

Рисунок 3.14 – Навчання удосконаленої багаторівневої LSTM-моделі

Грунтуючись на представлені графіки, було побудовано таблицю 3.1 порівняння результатів навчання.

Таблиця 3.1 – Порівняння результатів навчання моделей

	Точність	Втрати	Час	Кількість параметрів
Базова LSTM	100%	0,12%	55,79s	712 448
Спрощена LSTM	98,75%	4,71%	28,65s	93 910
Гібридна із GRU	100%	0,01%	50,11s	757 856
Уд. б-ва LSTM	96,88%	37,39%	2,18 m	2 444 768

Базова модель LSTM демонструє ідеальну точність і дуже низькі втрати, що свідчить про її ефективність. Однак час навчання трохи більший у порівнянні зі спрощеною версією. Це вказує на те, що модель є добре збалансованою за ефективністю та продуктивністю.

Спрощена модель LSTM має найкоротший час навчання завдяки меншій кількості шарів і параметрів. Однак точність трохи нижча, ніж у базової моделі, а втрати є вищими. Можна зробити висновок, що вона підходить для задач, де необхідна швидкість, але не критична максимальна точність.

Гібридна із GRU має ідеальну точність, як і базова LSTM, але демонструє ще менші втрати. Її час навчання трохи менший, ніж у базової моделі, що робить її оптимальним варіантом.

Удосконалена багаторівнева модель показує найнижчу точність і найвищі втрати попри складну архітектуру та використання регуляризації. Це може свідчити про перенавчання, або про те, що параметри не були оптимально налаштовані, але дивлячись на графік, точність на останньому

етапі є найвищою. Час навчання є найдовшим, що робить цю модель менш привабливою для задач, де важлива продуктивність.

Отже, найкращу точність мають базова LSTM і гібридна GRU (100%) моделі, найменші втрати – гібридна модель із GRU (0,01%), найшвидше навчання – спрощена LSTM (28,65с). Удосконалена LSTM модель не виправдовує себе через низьку точність і високі втрати, попри тривалий час навчання.

Таким чином, гібридна модель із GRU виглядає найбільш оптимальною завдяки найкращому балансу між точністю, втратами, часом навчання та кількістю параметрів.

3.8 Тестування оптимальної системи сурдоперекладу

Тестування системи сурдоперекладу проводилося в різних умовах, щоб оцінити її ефективність і стабільність. Зокрема, було перевірено роботу системи за трьох сценаріїв: при гарному освітленні, при поганому освітленні, а також на фоні зі зміненими умовами, включно з наявністю тіней.

У першому випадку, за умов гарного освітлення, система демонструє досить високу точність і швидкість розпізнавання жестів (рис. 3.15). Жести розпізнаються правильно, що свідчить про її стабільну роботу в сприятливих умовах.

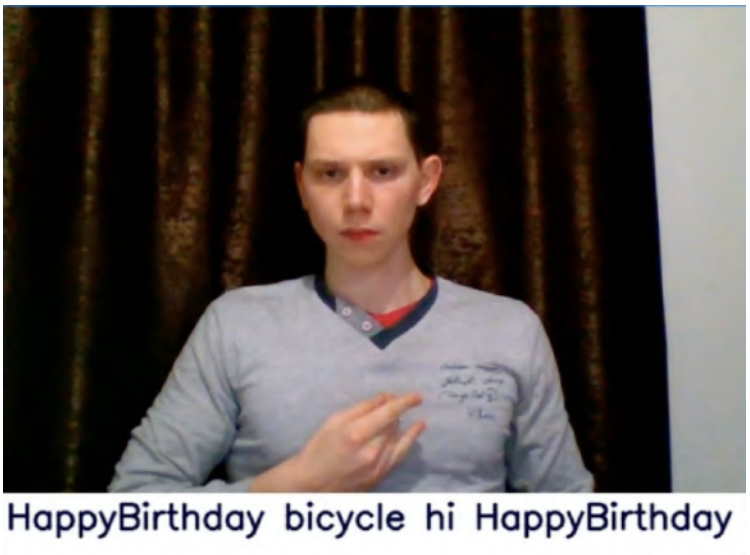

Рисунок 3.15 – Тестування жесту «Happy Birthday» за умов гарного освітлення

При поганому освітленні система також виявила здатність коректно розпізнавати жести, однак точність у таких умовах була дещо нижчою (рис. 3.16). Це вказує на залежність алгоритму від якості вхідних даних, зокрема освітлення, яке впливає на чіткість ключових точок рук. Незважаючи на це, результати залишилися прийнятними для більшості жестів.

Рисунок 3.16 – Тестування жесту «bicycle» за умов поганого освітлення

Також, у процесі тестування було помічено недолік: між виконанням жестів система іноді виводила результат для жесту, якого користувач не показував. Це створює перешкоди для безперервного перекладу і може бути пов’язане з шумами або недостатньою оптимізацією алгоритму інтерпретації жестів.

Третій сценарій включав тестування на зміненому фоні з наявністю тіней (рис. 3.17). У цих умовах система демонструвала трохи довший час обробки жестів, що може бути пов’язано зі складністю виділення ключових точок на більш контрастному фоні. Проте цікаво, що кількість хибнопозитивних результатів зменшилася. Ймовірно через те, що система стала обережніше інтерпретувати жести в умовах підвищеної складності.

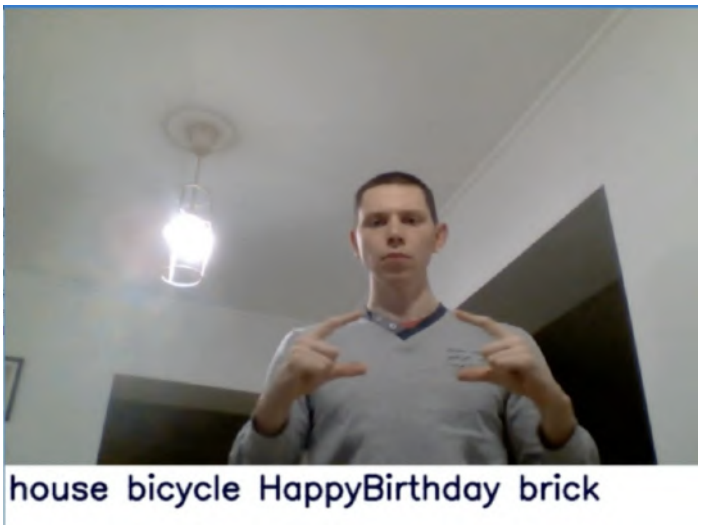

Рисунок 3.17 – Тестування жесту «brick» на зміненому фоні з наявністю тіней

Таким чином, тестування показало, що система загалом працює добре і здатна коректно розпізнавати жести за різних умов. Проте виявлені недоліки, зокрема поява жестів, яких користувач не виконував, а також затримки в умовах зміненого фону з тінями, потребують додаткової уваги для покращення стабільності та швидкодії системи.

Більше прикладів тестування системи додано до Додатку А.

3.9 Порівняння з існуючими системами

У процесі аналізу ефективності розробленої системи було проведено її порівняння з двома іншими існуючими рішеннями для сурдоперекладу – SignAll та HandTalk. Порівняння здійснювалося за кількома критеріями: точність, швидкість роботи, доступність, гнучкість і вартість впровадження.

Точність та ефективність.

Розроблена система демонструє високу точність розпізнавання жестів у базових умовах, таких як хороше освітлення та однорідний фон. У складних умовах, таких як змінене освітлення або наявність тіней, точність залишається прийнятною, але іноді з’являються некоректно розпізнані жести між правильними.

У порівнянні, інші системи краще справляються зі складними фонами та динамічними змінами у сцені завдяки використанню більш досконалих методів аналізу. Проте це часто пов’язано з потребою в більш потужному обладнанні або додаткових технологіях.

Швидкість роботи.

Запропонована система має хорошу швидкість обробки даних у реальному часі завдяки використанню оптимізованого підходу до аналізу відеопотоку. Проте в умовах складного фону чи великої кількості тіней швидкість обробки може дещо знижуватися.

SignAll має вищу швидкість роботи навіть у складних умовах завдяки спеціалізованому обладнанню, тоді як HandTalk демонструє приблизно схожі показники зі швидкістю розробленої системи за умови використання стандартного обладнання [62].

Доступність.

Створене рішення відзначається низькою вартістю впровадження та широкою доступністю, оскільки використовує лише стандартну камеру і не

потребує дорогого обладнання або постійного доступу до хмарних сервісів.

Інші системи менш доступні через високі вимоги до обладнання або залежність від інтернет-з’єднання для обчислень у хмарі, що обмежує їхнє використання в певних умовах.

Гнучкість.

Розроблена система має потенціал до адаптації, оскільки вона базується на відкритих інструментах і дозволяє модифікувати модель для нових мов чи специфічних умов.

Порівнювані системи, хоча й підтримують певну гнучкість, зосереджені на готових рішеннях і часто не передбачають можливості користувацьких модифікацій без доступу до внутрішніх алгоритмів чи API.

Отже, запропонована система забезпечує ефективне розпізнавання жестів за умов низької вартості та простоти впровадження. Вона вигідно відрізняється доступністю і можливістю локального використання без додаткових ресурсів. Проте за точністю та швидкістю роботи у складних умовах поступається більш технологічно розвинутим рішенням. Це свідчить про перспективність її подальшого вдосконалення для розширення функціональності та підвищення стійкості до зовнішніх факторів.

ВИСНОВКИ

У ході виконання роботи було розроблено систему сурдоперекладу жестової мови, яка забезпечує високу точність розпізнавання жестів у реальному часі та є доступною для широкого кола користувачів завдяки низькій вартості впровадження та використанню безкоштовних технологій.

Крім цього, у ході дослідження було досягнуто таких результатів:

– проведено аналіз сучасних методів відстеження рухів рук, включаючи методи на основі комп’ютерного зору, сенсорних технологій та їхні гібридні комбінації. На основі цього аналізу визначено переваги й недоліки різних підходів, що дозволило сформулювати критерії вибору оптимального методу;

– обґрунтовано використання методів комп’ютерного зору та рекурентних нейронних мереж типу LSTM для створення системи розпізнавання жестів. Обраний підхід забезпечує ефективне відстеження рухів рук із мінімальними витратами на реалізацію та відкриває можливості для подальшого вдосконалення;

– створено унікальний набір даних, який включає відеоматеріали жестів для десяти слів, отримані з відкритих джерел та власних записів. Проведено аналіз кожного відео, визначено ключові точки рук, які було використано для побудови навчальних вибірок;

– розроблено кілька архітектур нейронних мереж, включаючи базову, спрощену, гібридну та вдосконалену моделі. Моделі було навчені на створеному датасеті, і їх точність, швидкодія та адаптивність були проаналізовані;

– проведено тестування системи в реальних умовах експлуатації. Результати показали, що система демонструє стабільну роботу за різного освітлення, на змінених фонах і навіть у складних умовах, таких як наявність тіней. Однак було виявлено певні недоліки, наприклад, відображення жестів, яких користувач не показував, між зміною послідовних жестів;

– здійснено порівняння розробленої системи з існуючими продуктами, такими як HandTalk і SignAll. Визначено конкурентні переваги розробленої системи, включаючи її доступність, низьку вартість і можливість удосконалення. Також виявлено недоліки, що слугуватимуть основою для подальших досліджень і покращення.

Новизною роботи є аналіз методів стеження за руками та створення системи сурдоперекладу, що поєднує 2D-комп’ютерний зір та LSTM-нейронні мережі для високоточного розпізнавання жестів за мінімальних апаратних витрат.

Перспективи дослідження – вдосконалення запропонованої системи з розпізнавання жестів шляхом розширення датасету та впровадження нових методів обробки сигналів для підвищення точності в складних умовах, розвиток архітектур нейронних мереж (застосування більш складних гібридних моделей для кращої адаптивності та швидкодії системи). Крім того, перспектива інтеграції цієї системи в реальні програми для підтримки людей з вадами слуху та розширення її можливостей для різних мов жестів створює великий потенціал для соціального впливу та покращення комунікації між людьми.

Результати дослідження апробовано у вигляді 2 тез доповідей під час Міжнародного молодіжного форуму «Радіоелектроніка і молодь у XXI столітті» [63], VI Міжнародної студентської наукової конференції «Концепт науки XXI: стратегії, методи та наукові інструменти» [64].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Порушення слуху: як виявити і коли звернутися до лікаря? Міністерство охорони здоров'я України. URL: https://moz.gov.ua/uk/porushennja-sluhu-jak-vijaviti-i-koli-zvernutisja-do-likarja- (дата звернення: 25.11.2024)

2. Міжнародний день глухих. URL: https://zmdl5.zp.ua/mizhnarodnyj-den-gluhyh/ (дата звернення: 25.11.2024)

3. Горлачов, О. С. (2014). Психологічні особливості здійснення сурдоперекладу в навчальній діяльності та мікросоціумі осіб із вадами слуху. С. 147.

4. Жестова мова глухих: лексикографічний досвід / О. Тищенко // Лексикографічний бюлетень: Зб. наук. пр. – К.: Ін-т української мови НАН України, 2006. – Вип. 14. – С. 30-40. – Бібліогр.: 23 назв. – укр.

5. Кульбіда, С. В. (2009). Українська жестова мова як природна знакова система. Жестова мова й сучасність: збірник наукових праць, 1(4), 218-239.

6. Круглик, О.П., & Горлачов, О.С. (2022). Значення слухового сприймання в процесі здійснення сурдоперекладу для осіб з порушеннями слуху. Науковий часопис НПУ імені МП Драгоманова. Серія 19. Корекційна педагогіка та спеціальна психологія, (43), 39–48.

7. Данілін О.М. Цифрова дипломатія МЗС України: сучасність та перспективи. – Кваліфікаційна робота на здобуття освітнього ступеня магістра спеціальності 291 «Міжнародні відносини, суспільні комунікації та регіональні студії» освітньо-професійної програми «Зовнішня політика і дипломатія». – Національний авіаційний університет. – Київ, 2023. – 90с.

8. Поліщук, Н., & Пензай, С. (2024). Особи з обмеженими можливостями як об'єкт Соціально-реабілітаційної роботи. Перспективи та інновації науки, (2 (36)).

9. Кордіяка, Х. М. (2024). Соціальна інклюзія людей з інвалідністю у навчальному середовищі. Організація, від імені якої випущено видання, 405.

10. Luiten, J., Kopanas, G., Leibe, B., & Ramanan, D. (2024, March). Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In 2024 International Conference on 3D Vision (3DV) (pp. 800–809). IEEE.

11. 2D Object Tracking – Experiments. Lucas Dal'Col. URL: https://lucasrdalcol.github.io/posts/2d-object-tracking/ (дата звернення: 27.11.2024).

12. Bansal, M., Kumar, M. & Kumar, M. 2D Object Recognition Techniques: State-of-the-Art Work. Arch Computat Methods Eng 28, 1147–1161 (2021).

13. Оксанюк, М. С., Радюк, П. М., Скрипник, Т. К., & Пасічник, О. А. (2024). Метод віртуального примірювання одягу за зображеннями високої роздільної здатності з ефектами оклюзії.

14. Canny Edge Detection: Explained and Compared with OpenCV in Python. URL: https://medium.com/@abhisheksriram845/canny-edge-detection-explained-and-compared-with-opencv-in-python-57a161b4bd19 (дата звернення: 28.11.2024)

15. Edges: The Canny Edge Detector. School of Informatics, The University of Edinburgh. URL: https://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/MARBLE/low/edges/canny.htm (дата звернення: 28.11.2024)

16. What is Canny edge detection? URL: https://www.educative.io/answers/what-is-canny-edge-detection (дата звернення: 29.11.2024)

17. Karatay, R., Demir, B., Ergin, A. A., & Erkan, E. (2024). A real-time eye movement-based computer interface for people with disabilities. Smart Health, 34, 100521.

18. Al-Haddad, L. A., Alawee, W. H., & Basem, A. (2024). Advancing task recognition towards artificial limbs control with ReliefF-based deep neural network extreme learning. Computers in Biology and Medicine, 169, 107894.

19. Hand Tracking. Machine Learning for Engineers. https://apmonitor.com/pds/index.php/Main/HandTracking (дата звернення: 29.11.2024)

20. Kamble, T. U., & Mahajan, S. P. (2024). 3D vision using multiple structured light-based kinect depth cameras. International Journal of Image and Graphics, 24(01), 2450001.

21. Lahiri, S., Ren, J., & Lin, X. (2024). Deep learning-based stereopsis and monocular depth estimation techniques: a review. Vehicles, 6(1), 305-351.

22. Willomitzer, F. (2024). Synthetic Wavelength Imaging: Utilizing Spectral Correlations for High-Precision Time-of-Flight Sensing. Computational Imaging for Scene Understanding: Transient, Spectral, and Polarimetric Analysis, 187.

23. Tomaszewski, D., Rapiński, J., & Pelc-Mieczkowska, R. (2017). Concept of AHRS algorithm designed for platform independent IMU attitude alignment. Reports on Geodesy and Geoinformatics, 104(1), 33-47.

24. Матвіїв, Р. З., Онутчак, Т. А., Павленко, М. В., Живицький, І. Б., & Барило, Г. І. (2024, May). Програмно-керований вимірювальний перетворювач ємнісних сенсорів. In The 21st International scientific and practical conference «Innovative solutions in public communications and international relations» (May 28–31, 2024) Sofia, Bulgaria. International Science Group. 2024. 382 p. (p. 362).

25. Ковалінський, Б. І. (2024). Розумний паркінг на базі системи розпізнавання номерних знаків.

26. Leap motion controller overview. URL: https://www.ultraleap.com/leap-motion-controller-overview (дата звернення: 29.11.2024)

27. Pavllo, D., Delahaye, M., Porssut, T., Herbelin, B., & Boulic, R. (2019). Real-time neural network prediction for handling two-hands mutual occlusions. Computers & Graphics: X, 2, 100011.

28. Sharma, T., & He, Y. (2024). Design of a tracking controller for autonomous articulated heavy vehicles using a nonlinear model predictive control technique. Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi-body Dynamics, 14644193241232353.

29. Qi, J., Ma, L., Cui, Z., & Yu, Y. (2024). Computer vision-based hand gesture recognition for human-robot interaction: a review. Complex & Intelligent Systems, 10(1), 1581-1606.

30. Bilik, S., Zemcik, T., Kratochvila, L., Ricanek, D., Richter, M., Zambanini, S., & Horak, K. (2024). Machine learning and computer vision techniques in continuous beehive monitoring applications: A survey. Computers and Electronics in Agriculture, 217, 108560.

31. Zhang, H., Cheng, S., Zhao, Y., Jing, J., Su, Z., & Li, P. (2024). Measurement of yarn apparent evenness based on modified Canny edge detection. The Journal of The Textile Institute, 115(4), 600-606.

32. S. Bhattacharjee, P. Majumdar and Y. J. Singh, «An Effective Monitoring of Women Reproductive Organ Cancer using Mean based KPCA», 2018 Fourth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN), Kolkata, India, 2018, pp. 87-92.

33. Мандріков, А. Д. (2024). Аналіз алгоритмів сегментації пухлин за гібридними зображеннями.

34. Nguyen, T. D., Nguyen, T. H., Ene, A., & Nguyen, H. (2023). Improved convergence in high probability of clipped gradient methods with heavy tailed noise. Advances in Neural Information Processing Systems, 36, 24191-24222.

35. Sage, D., & Unser, M. (2003). Teaching image-processing programming in Java. IEEE Signal Processing Magazine, 20(6), 43-52.

36. Li, Z., Liu, F., Yang, W., Peng, S., & Zhou, J. (2021). A survey of convolutional neural networks: analysis, applications, and prospects. IEEE transactions on neural networks and learning systems, 33(12), 6999-7019.

37. O'Shea, K. (2015). An introduction to convolutional neural networks. arXiv preprint arXiv:1511.08458.

38. Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., ... & Chen, T. (2018). Recent advances in convolutional neural networks. Pattern recognition, 77, 354-377.

39. Yamashita, R., Nishio, M., Do, R. K. G., & Togashi, K. (2018). Convolutional neural networks: an overview and application in radiology. Insights into imaging, 9, 611-629.

40. Recurrent Neural Network (RNN) Architecture Explained. URL: https://medium.com/@poudelsushmita878/recurrent-neural-network-rnn-architecture-explained-1d69560541ef (дата звернення: 30.11.2024)

41. What is a recurrent neural network (RNN)? URL: https://www.ibm.com/topics/recurrent-neural-networks (дата звернення: 30.11.2024)

42. Li, Z., & Li, S. (2023). Recursive recurrent neural network: A novel model for manipulator control with different levels of physical constraints. CAAI Transactions on Intelligence Technology, 8(3), 622-634.

43. Mastering the LSTM Network: A Step-by-Step Guide in Easy Language.
URL:https://medium.com/@amiralizadeh1992/learn-the-lstm-step-by-step-and-utilise-this-model-for-two-nlp-and-time-series-prediction-projects-20e383b36c1a
(дата звернення: 30.11.2024)

44. Mienye, I. D., Swart, T. G., & Obaido, G. (2024). Recurrent neural networks: A comprehensive review of architectures, variants, and applications. Information, 15(9), 517.

45. Deep Neural Networks. URL: https://www.tutorialspoint.com/python_deep_learning/python_deep_learning_deep_neural_networks.htm (дата звернення: 01.12.2024)

46. Chakraborty, B. K., Sarma, D., Bhuyan, M. K., & MacDorman, K. F. (2018). Review of constraints on vision‐based gesture recognition for human–computer interaction. IET Computer Vision, 12(1), 3-15.

47. SignAll. URL: https://partner.microsoft.com/ko-kr/case-studies/signall (дата звернення: 01.12.2024)

48. Kinect for Windows. URL: https://learn.microsoft.com/en-us/windows/apps/design/devices/kinect-for-windows (дата звернення: 02.12.2024)

49. Hand Talk. URL: https://www.handtalk.me/en/ (дата звернення: 03.12.2024)

50. Daemi, P., Zhou, Y., Naish, M. D., Price, A. D., & Trejos, A. L. (2023). Comprehensive kinematic model of a tendon-driven wearable tremor suppression device. IEEE Transactions on Robotics.

51. Sharkawy, A. N., & Khairullah, S. S. (2023). Forward and Inverse Kinematics Solution of A 3-DOF Articulated Robotic Manipulator Using Artificial Neural Network. International Journal of Robotics & Control Systems, 3(2).

52. Bai, Y., Yan, B., Zhou, C., Su, T., & Jin, X. (2023). State of art on state estimation: Kalman filter driven by machine learning. Annual Reviews in Control, 56, 100909.

53. Kalman Filter. Prof. Giuseppe OrioloProf. Giuseppe Oriolo.URL: https://www.diag.uniroma1.it/oriolo/amr/slides/Localization2_Slides.pdf (дата звернення: 04.12.2024)

54. Jupyter Notebook. URL: https://jupyter.org/ (дата звернення: 04.12.2024)

55. Python. URL: https://www.python.org/ (дата звернення: 04.12.2024)

56. OpenCV. URL: https://opencv.org/ (дата звернення: 04.12.2024)

57. MediaPipe. URL: https://github.com/google-ai-edge/mediapipe (дата звернення: 04.12.2024)

58. Tensorflow. URL: https://www.tensorflow.org/ (дата звернення: 04.12.2024)

59. Keras. URL: https://keras.io/ (дата звернення: 04.12.2024)

60. Ghaleb, F. , Youness, E. , Elmezain, M. and Dewdar, F. (2015) Vision-Based Hand Gesture Spotting and Recognition Using CRF and SVM. Journal of Software Engineering and Applications, 8, 313-323.

61. Reed, M. P., Zhou, W., & Wegner, D. M. (2011). Automated grasp modeling in the human motion simulation framework. In Proceedings of the SAE Digital Human Modeling for Design and Engineering Conference.

62. Raj, D. R., & Anusha, M. M. (2024). A PROJECT REPORT ON SIGN LANGUAGE TRANSLATOR. International Journal of Engineering Research and Science & Technology, 20(4), 35-41.

63. Шовковий Є. І. Дослідження методів автоматизації сурдоперекладу жестових мов / Є. І. Шовковий ; наук. керівн. д. т. н., проф. В. П. Машталір // Радіоелектроніка та молодь у XXI столітті : матеріали 28-го Міжнар. молодіж. форуму, 16-18 квітня 2024 р. – Харків : ХНУРЕ, 2024. – Т. 7. – С. 148–149.

64. Шовковий Є.І. Способи автоматичного перекладу між різними жестовими мовами / Є. І. Шовковий ; наук. керівн. д. т. н., проф. В. П. Машталір // VI Міжнародна студентська наукова конференція «Концепт науки XXI: стратегії, методи та наукові інструменти», 13 вересня 2024р. – Харків. С.1 – С. 77-78.

65. Mashtalir, S., Mashtalir, V., & Stolbovyi, M. (2017). Video shot boundary detection via sequential clustering. International Journal «Information Theories and Applications», 24(1), 50-59.

66. D. Kinoshenko, S. Mashtalir, V. Shlyakhov Temporal video segmentation via spatial image segmentation International journal information technologies & knowledge 7(3), 2013, 212-219.

67. Mashtalir, V., Ruban, I., & Levashenko, V. (Eds.). (2020). Advances in Spatio-Temporal Segmentation of Visual Data. Springer.

68. Шовковий Є. І. Методи автоматичного сурдоперекладу із однієї жестової мови на іншу жестову мову / Є. І. Шовковий // Радіоелектроніка та молодь у XXI столітті : матеріали 27-го Міжнар. молодіж. форуму, 10–12 травня 2023 р. – Харків : ХНУРЕ, 2023. – Т. 6, ч.1. – С. 19–20.

69. Shovkovyi, Y., Grinyova, O., Udovenko, S., & Chala, L. (2023). Система автоматичного сурдоперекладу з використанням нейромережних технологій та 3D-анімації. Сучасний стан наукових досліджень та технологій в промисловості, (4 (26), 108–121).

70. Шовковий Є.І. Методи автоматизації сурдоперекладу: кваліфікаційна робота першого (бакалаврського) рівня вищої освіти: 122 Комп’ютерні науки. Харків, 2023. 75 с.

71. Liang, Z., Li, H., & Chai, J. (2023). Sign Language Translation: A Survey of Approaches and Techniques. Electronics, 12(12), 2678.