**МІНІСТЕРСТВО ФІНАНСІВ УКРАЇНИ**
**ДЕРЖАВНИЙ ПОДАТКОВИЙ УНІВЕРСИТЕТ**

**ФАКУЛЬТЕТ ФІНАНСІВ ТА ЦИФРОВИХ ТЕХНОЛОГІЙ**
**КАФЕДРА КОМП'ЮТЕРНИХ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ І СИСТЕМ**

**КВАЛІФІКАЦІЙНА РОБОТА**
**на тему:**

# ДОДАТОК ДЛЯ РОЗПІЗНАВАННЯ МОВИ ЖЕСТІВ

Здобувача вищої освіти першого (бакалаврського) рівня,
4 курсу, групи КБ-19-1, спеціальності 122 «Комп'ютерні науки»,
освітня програма «Інформаційні управляючі системи і технології (за галузями)»

_____ Орлової Вікторії Андріївної
(підпис) (прізвище, ім'я, по батькові)

Науковий керівник _____ Котух Євген Володимирович, к.т.н., доцент
(підпис) (прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Керівник освітньо-професійної програми _____ Федотова - Півень Ірина Миколаївна, к.т.н, доцент
(підпис) (прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Допущено до захисту за рішенням кафедри
протокол від «26» травня 2023 № 13

Завідувач кафедри _____ Філоненко Михайло Миколайович, к.ф.-м.н., доцент
(підпис) (прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Ірпінь – 2023

Реєстрація № 12 05.06.2023 _____
(дата) (підпис відповідальної особи)

Результати перевірки до захисту 29.05.2023 _____
(до захисту, на доопрацювання) (дата) (підпис)

Результат захисту 5 (відмінно) 15.06.2023 _____
(оцінка) (дата) (підпис)

Голова комісії: _____ В.М. Франчук
(підпис)

Члени комісії: _____ М.М. Філоненко
(підпис)

_____ О.В. Редич
(підпис)

_____ Є.В. Котух
(підпис)

# БЛАНК ОЦІНЮВАННЯ ВИПУСКНОЇ РОБОТИ

| Об'єкти контролю | Діапазон оцінки в балах | Фактична кількість балів | Примітки | Дата, П.І.Б.: керівника, голови комісії членів комісії,секретаря, (підписи) |
|---|---|---|---|---|
| **ОЦІНЮВАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ КОМІСІЄЮ ПО ЗАХИСТУ** | | | | |
| 1. Зміст випускної роботи (якість теоретичного матеріалу, його зв'язок з практичним, вміння представляти матеріал в логічній послідовності, наявність висновків та пропозицій) | від 0 до 30 | 29 | | 15.06.2023 Франчук В.М. Філоненко М.М Редич О.В. Котух Є.В. Суховецька Л.П. |
| 2. Дотримання вимог щодо оформлення | від 0 до 10 | 8 | | |
| 3. Вчасне виконання роботи на всіх етапах у встановлені кафедрою терміни | від 0 до 5 | 4 | | |
| 4. Рецензент | від 0 до 5 | 5 | | |
| РАЗОМ БАЛІВ | від 0 до 50 | 46 | | |

**Оцінювання кваліфікаційної роботи науковим керівником - 50 балів.**

**ПЕРЕВЕДЕННЯ ОЦІНКИ ЗА 100-БАЛЬНОЮ ШКАЛОЮ ДО НАЦІОНАЛЬНОЇ ШКАЛИ ТА ШКАЛИ ЄКТС**

| Оцінка за 100-бальною шкалою | Оцінка за шкалоюЄКТС | Оцінка за національною шкалою | Дата | П.І.Б.: голови, членів комісії, секретаря, (підпис) |
|---|---|---|---|---|
| 96 | A | 5 (відмінно) | 15.06.2023 | Франчук В.М. Філоненко М.М Редич О.В. Котух Є.В. Суховецька Л.П. |

**ПРАВИЛА ПЕРЕВЕДЕННЯ ОЦІНОК ДО НАЦІОНАЛЬНОЇ ШКАЛИ ТА ШКАЛИ ЄКТС**

| Оцінка за 100-бальною шкалою | Оцінка за шкалою ЄКТС | Оцінка за національною шкалою |
|---|---|---|
| 90-100 | A | 5 (відмінно) |
| 80-89 | B | 4 (добре) |
| 70-79 | C | |
| 60-69 | D | 3 (задовільно) |
| 50-59 | E | |
| 35-49 | FX | 2 (незадовільно) |

акультет фінансів та цифрових технологій
афедра комп'ютерних та інформаційних технологій і систем
алузь знань: 12 «Інформаційні технології»
еціальність: 122 «Комп'ютерні науки»
світня програма: «Інформаційні управляючі системи і технології (за галузями)»

**Затверджую**
Завідувач кафедри
_____ Філоненко М.М.
«16» 11 2022

**Завдання на кваліфікаційну роботу**
**здобувачу вищої освіти**
Орловій Вікторії Андріївній
(прізвище, ім'я, по батькові)

Тема кваліфікаційної роботи
одаток для розпізнавання мови жестів
твердження наказом ректора від «15» листопада 2022 № 1205 (зі змінами наказ від 04» квітня 2023 № 361)
Строк здачі закінченої роботи «26» травня 2023
Цільова установка застосування набутих знань та практичних навичок в розробці програмного забезпечення для розпізнавання мови жестів
Мета роботи розробка системи для розпізнавання знаків АМЖ.
б'єкт дослідження Глухі та взаємодія з рішеннями розпізнавання мови жестів.
редмет дослідження розпізнавання жестів за допомогою машинного навчання.
аза дослідження доступні та створені набори даних, що містять записи жестів.
ихідні дані до роботи методологія розробки розпізнавання мови жестів
Перелік графічного матеріалу розподіл глухих та Глухих людей, робота датчиків ибини, переваги та мінуси рішень, ML рішення в MediaPipe, робота системи озпізнавання, схема обробки даних, тренування даних, цілісна обробка ключових очок, архітектура моделі екстракту орієнтирів руки, вікно Raspberry Pi Imager, атриця плутанини, результати тренування моделі, інтерфейс системи розпізнавання ови жестів, запуск команди, запуск потокового відео.
Зміст випускної кваліфікаційної роботи (перелік питань за кожним розділом): СТУП. РОЗДІЛ 1. ДОСЛІДЖЕННЯ ІСНУЮЧОГО СТАНУ ПРОБЛЕМНОЇ БЛАСТІ. 1.1 Особливості та труднощі при ідентифікації мови жестів. 1.2 Аналіз етодів та існуючих рішень у галузі розробки розпізнавання мови жестів.1.3. Вибір етодів та приладів для реалізації рішення розпізнавання жестів.РОЗДІЛ 2. ОЗРОБКА АЛГОРИТМУ ТА ПРОГРАМИ.2.1. Схема виявлення та розпізнавання естів в реальному часі за допомогою модулю MediaPipe.2.2. Підготовка Raspberry-(RPI) для запуску потокового відео.РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ТА НАЛІЗ ОТРИМАНИХ РЕЗУЛЬТАТІВ. 3.1. Результати дослідження. ВИСНОВКИ. ПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ. ДОДАТКИ

7. Календарний план виконання роботи

| № п/п | Назва етапів кваліфікаційноїроботи | Строк виконання етапів роботи | |
|---|---|---|---|
| | | За планом | Фактично |
| 1 | 2 | 3 | 4 |
| 1 | Вибір теми, підбір літературних джерел, формування структури кваліфікаційної роботи | До 15.11.2022 | 15.11.2022 |
| 2 | І розділ (розкрити сутність і стан дослідження наукової проблематики роботи, після чого навести її загальну характеристику; огляд літератури за темою і вибір конкретних напрямів досліджень) | 01.02.2023 | 01.02.2023 |
| 3 | ІІ розділ (аналіз експериментальної частини про методики досліджень) | 03.04.2023 | 04.04.2023 |
| 4 | ІІІ розділ (проведені теоретичні і (або) експериментальні дослідження та проектні рішення) | 10.05.2023 | 09.05.2023 |
| 5 | Висновки | 15.05.2022 | 15.05.2022 |
| 6 | Доопрацювання роботи з урахуванням зауважень наукового керівника | 26.05.2023 | 26.05.2023 |
| 7 | Отримання відгуку керівника | 06.06.2023 | 06.06.2023 |
| 8 | Отримання зовнішньої рецензії | 08.06.2023 | 08.06.2023 |

8. Дата видачі завдання «16» листопада 2022

9. Керівник випускної кваліфікаційної роботи _____ (підпис) Котух Є.В. (прізвище, ініціали)

10. Керівник освітньо-професійної програми _____ (підпис) Федотова-Пів (прізвище, ініціали)

11. Завдання до виконання отримав здобувач вищої освіти _____ (підпис) Орлова В.А. (прізвище, ініціали)

# АНОТАЦІЯ

«Додаток для розпізнавання мови жестів»

*Мета роботи полягає* в розробці програмного забезпечення по розпізнаванню знаків американської мови жестів з набором базових жестів

*Завдання роботи є*

- Вивчення особливостей мови жестів з метою збору тренувальних даних.
- Дослідження та порівняння можливих методів для розпізнавання жестів.
- Навчання нейронної мережі для розпізнавання мови жестів.
- Розробка базового інтерфейсу користувача.
- Валідація та оцінка ефективності розробленої системи розпізнавання мови жестів.
- Дослідження можливості розширення системи розпізнавання мови жестів.

*Предметом дослідження є* розпізнавання мови жестів глухих людей за допомогою машинного навчання.

*Об'єктом дослідження* є глухі люди та їх взаємодія з рішеннями розпізнавання мови жестів.

*Базою дослідження виступає* доступні та створені набори даних, що містять записи мови жестів глухих людей.

*За результатами дослідження сформульовані* методика та реалізація рішення для розпізнавання мови жестів.

*Одержані результати можуть бути використані* для подальших наукових робіт з ціллю оптимізації та для використання як додаток на веб-сайтах для перекладання при умові розширення бази даних жестів.

*Кваліфікаційна робота* містить 51 сторінок, 1 таблицю, 14 рисунків, список використаних джерел 43 найменувань, 1 додаток.

*Ключові слова*: Система для розпізнавання мови жестів, навчання нейронної мережі, американська мова жестів, довга короткочасна пам'ять(ДКЧП), бібліотека MediaPipe Holistic keypoints, OpenCV, Raspberry Pi.

*Рік виконання кваліфікаційної роботи* 2023

*Рік захисту роботи* 2023

**ANNOTATION**

«An application for recognizing sign language»

*The aim of this work is* in the development of software for recognizing American Sign Language signs with a set of basic gestures.

*The tasks of the work are as follows:*

- Researching the peculiarities of sign language in order to collect training data.
- Research and comparison of possible methods for gesture recognition.
- Training a neural network for sign language recognition.
- Basic user interface development.
- Validation and evaluation of the effectiveness of the developed sign language recognition system.
- Studying the possibility of expanding the sign language recognition system.

*The subject of the research is* recognizing the sign language of deaf people using machine learning.

*The object of the research is* there are deaf people and their interaction with sign language recognition solutions.

*The research is based on* datasets containing sign language recordings of deaf people are available and created.

*The results of the research include* methodology and implementation of a solution for sign language recognition.

*The obtained results can be used to* for further research with the aim of optimization and for use as an application on translation websites, subject to the expansion of the gesture database.

*The qualification work consists of* contains 51 pages, 1 table, 14 figures, a list of used sources of 43 items, 1 appendix.

*Keywords:* A system for recognizing sign language, neural network training, American Sign Language, long-short-term memory, MediaPipe Holistic keypoints library, OpenCV, Raspberry Pi.

*Year of qualification work:* 2023

*Year of work defense:* 2023

# ЗМІСТ

ВСТУП 3

РОЗДІЛ 1. ДОСЛІДЖЕННЯ ІСНУЮЧОГО СТАНУ ПРОБЛЕМНОЇ ОБЛАСТІ 6

1.1 Особливості та труднощі при ідентифікації мови жестів 6

1.2 Аналіз методів та існуючих рішень у галузі розробки розпізнавання мови жестів 8

1.2.1. Переваги та виклики існуючих методів 10

1.2.2. Економічний аналіз 13

1.3. Вибір методів та приладів для реалізації рішення розпізнавання жестів 14

РОЗДІЛ 2. РОЗРОБКА АЛГОРИТМУ ТА ПРОГРАМИ 19

2.1. Схема виявлення та розпізнавання жестів в реальному часі за допомогою модулю MediaPipe 19

2.1.1. Збір даних для моделі машинного навчання 26

2.1.2. Обробка відеосигналу 30

2.2. Підготовка Raspberry-pi(RPI) для запуску потокового відео 34

РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ТА АНАЛІЗ ОТРИМАНИХ РЕЗУЛЬТАТІВ 40

3.1. Результати дослідження 40

ВИСНОВКИ 45

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 46

ДОДАТКИ 51

# ВСТУП

Ніколи не зникаючи у світі зростає потреба в системах розпізнавання жестів для покращення взаємодії з комп'ютерними системами та підтримки людей з обмеженими можливостями. Більшість населення не володіє жестовою мовою. Розробка додатка який може розпізнавати американську жестову мову (АЖМ) для веб-сайтів чи інших ресурсів, стало би значним кроком до усунення комунікаційних бар’єрів, з якими щодня стикаються значна кількість глухих людей. Така програма забезпечить ефективнішу комунікацію між глухими та людьми без проблем зі слухом, що полегшить глухим людям взаємодію з навколишнім світом.

**Мета і завдання дослідження.**

Метою кваліфікаційної роботи було розробити програмне забезпечення по розпізнаванні знаків АЖМ з базовими жестами. Робота спрямована на навчання нейронної мережі, яка здатна автоматично вивчати та класифікувати жести мови з високою точністю. Ціль додатку ідентифікувати та розпізнати жест, після чого вивести у письмовому вигляді його значення.

1. Вивчення особливостей мови жестів, використовуваної глухими людьми, з метою збору базових жестів та їхнього семантичного аналізу.
2. Дослідження та порівняння можливих методів для розпізнавання мови жестів.
3. Навчання нейронної мережі для розпізнавання мови жестів, з використанням набору даних, що містить різноманітні жести.
4. Розробка базового інтерфейсу користувача, який забезпечує зручну та інтуїтивно зрозумілу взаємодію глухих користувачів з системою розпізнавання мови жестів.
5. Валідація та оцінка ефективності розробленої системи розпізнавання мови жестів, з використанням реальних тестових сценаріїв та порівняння з іншими методами розпізнавання.

6. Дослідження можливості розширення системи розпізнавання мови жестів для включення нових жестів та покращення адаптивності до індивідуальних потреб користувачів.

Наприклад, людина з вадами слуху може використовувати мову жестів, щоб камера передавала фрейми відео і в реальному часі нейрона система після обробки даних, як вихідний результат виводила розпізнанні жести в слова на екрані. Важливість даного проекту підкреслюється тим, що глухі люди часто стикаються з труднощами у спілкуванні, освіті та соціальному залученні через відсутність доступності та розуміння мови жестів. Зокрема дослідження Еріка Емерсона, професора охорони здоров'я людей з обмеженими можливостями показує, що люди з особливими потребами відчувають самотність набагато частіше, ніж люди без них. «Порівняно із населенням в цілому люди з інвалідністю мають менше друзів, менше соціальної підтримки та більш соціально ізольовані» [22]. Завдяки розпізнаванню та перекладу АЖМ додаток можна використовувати для створення субтитрів у реальному часі для лекцій та інших навчальних матеріалів. Це дозволить створити більш інклюзивне навчальне середовище, де глухі люди зможуть повною мірою брати участь у обговореннях і заходах у класі. Загалом розробка програми, яка може розпізнавати АЖМ, є важливою, оскільки вона має потенціал для покращення якості життя глухих людей шляхом.

**Об’єкт та предмет дослідження.** У контексті дослідження програмного забезпечення для розпізнавання мови жестів для глухих людей, можна визначити об'єкт і предмет дослідження наступним чином:

**Об'єктом дослідження** є глухі люди та їх взаємодія з рішеннями розпізнавання мови жестів. Об'єктом дослідження також рахуються особливості та потреби глухих людей у використанні мови жестів, їхні здібності та переваги у використанні такої системи, а також проблеми та виклики, з якими вони зіштовхуються.

**Предметом дослідження** являється програмне забезпечення для розпізнавання мови жестів та його компоненти. Відповідно до нього входять алгоритми та методи розпізнавання жестів, нейронні мережі та їх навчання, вибір та оптимізація вхідних даних для розпізнавання, розробка інтерфейсу користувача, технології передачі та обробки даних, а також оцінка та валідація ефективності системи розпізнавання мови жестів.

Отже, об'єкт і предмет дослідження взаємозв’язані між собою, де об'єкт - люди з дефектами слуху та мови, їх потреби та взаємодія з системою, а предмет - програмне забезпечення та його компоненти, які розробляються та адаптуються для відповідності вимог потребам цільової аудиторії користувачів і заради забезпечення ефективної взаємодії.

**Методи дослідження - теоретичні –**аналіз літератури та існуючих рішень, систематизація даних; **емпіричні –** збір і аналіз фактичних тренувальних даних, експерименти, тестування, використання досвіду в ланці для створення рішень.

**Базою дослідження виступає** доступні та створені набори даних, що містять записи мови жестів глухих людей.

**Наукова-практична -значимість –одержаних-результатів.** Наукова база для подальших досліджень та покращень результатів. забезпечить ефективнішу комунікацію між глухими та людьми без проблем зі слухом, що полегшить глухим людям взаємодію з навколишнім світом. Це також сприяло б більшому залученню та розумінню культури глухих, яка часто маргіналізована та неправильно зрозуміла. Крім того, додаток може мати значний вплив на освіту глухих людей.

# РОЗДІЛ 1. ДОСЛІДЖЕННЯ ІСНУЮЧОГО СТАНУ ПРОБЛЕМНОЇ ОБЛАСТІ

## 1.1 Особливості та труднощі при ідентифікації мови жестів

Створення перекладу для мови жестів з точки зору технічної реалізації може бути викликом через ряд особливостей і труднощів.

Однією з основних складнощів є відсутність стандартизованої граматики для мов жестів. Відсутність універсальних правил синтаксису та структури мовлення ускладнює розробку алгоритмів та моделей для перекладу. Для успішного перекладу потрібно враховувати варіації в граматиці мов жестів та їх зв'язок з контекстом.

Також варто зазначити, що мови жестів різноманітні та несуть ряд неоднозначностей. Це може бути особливо важким при розгляді різних культур та регіонів. «У світі є близько 150 національних жестових мов та ще стільки ж жестомовних діалектів» [5]. Система перекладу повинна визначати правильне значення та інтерпретацію жесту в конкретному контексті.

Ще одним викликом є забезпечення швидкості та реалізації перекладу в реальному часі. Це вимагає швидкої обробки та зворотного зв’язку від системи перекладу, а також ефективної оптимізації алгоритмів та моделей. Це особливо важливо для забезпечення плавної комунікації між глухими та неглухими людьми.

Недостатність доступних даних для навчання систем перекладу також може бути проблемою. Зібрання великого обсягу даних з різних мовних спільнот може бути складним завданням. Обмежена кількість даних може вплинути на точність та загальну ефективність системи перекладу.

Крім того, мови жестів мають просторові виміри, які потрібно враховувати при розробці системи перекладу. Розуміння та інтерпретація таких аспектів як позиція та рухи рук та інших частин тіла може бути складним завданням для технологій комп'ютерного зору та машинного навчання (Рис. 1.1).




Рис. 1.1 Розподіл глухих та Глухих людей

Зокрема, важливо пам’ятати що правильне значення терміну Глухих людей, щоб розрізнити їх потреби. «За цим підходом, термін, який визначає спільноту нечуючих людей, має написання тільки з великої літери (Глухі) на знак приналежності до відповідної лінгвістичної групи» [3].

«Глухий – Глухий з великої літери – це особа, яка ототожнює себе з культурною та мовною меншиною глухих. Дуже часто вона не сприймає себе як інваліда, а як представника меншини» [2].

«Глуха – це людина, яка не чує. Вона використовує PJM. Поняття ширше, ніж Глухий, оскільки воно також описує людей, які не ототожнюються з культурою глухих» [2]. Під PJM в даному контексті розуміється, як польську мову жестів для глухої людини.

З усіх цих причин вирішено розпочати проект з тренування невеликої кількості базових слів американської мови жестів, таких як «привіт», «так», «ні». Це дозволяє зосередитися на створенні системи перекладу з основними елементами та дослідженні методів, що найкраще відповідають вимогам розпізнавання мови жестів. Початок з невеликої кількості слів допомагає сконцентрувати увагу на вирішенні загальних проблем, пов'язаних з розпізнаванням та перекладом мови жестів, перш ніж розширювати асортимент слів та складність завдання.

## 1.2 Аналіз методів та існуючих рішень у галузі розробки розпізнавання мови жестів

У сфері розпізнавання мови жестів існує перечень можливостей та досліджень різноманітних технологій, які зокрема використовуються для розпізнавання та інтерпретації жестів користувача або знаходяться на стадії розробки та дослідження. Ось деякі з них:

1. Комп’ютерне бачення:

1.1. Методи комп’ютерного бачення передбачають аналіз даних відео або зображень для розпізнавання жестів. Ці методи використовують алгоритми для виявлення та відстеження рухів рук та інших відповідних частин тіла необхідних для визначення контексту. Аналізуючи просторові та часові моделі цих рухів, системи комп’ютерного зору можуть розпізнавати та інтерпретувати різні жести.

2. Машинне навчання.

2.1. Алгоритми машинного навчання відіграють вирішальну роль у розпізнаванні мови жестів. Ці алгоритми навчаються на великих наборах даних відеороликів мови жестів, де надаються позначені приклади різних знаків. Навчаючись на цих прикладах, моделі машинного навчання можуть ідентифікувати та класифікувати жести, які виконуються в нових, невидимих відео. Вони можуть вловлювати складні моделі та варіанти мови жестів і робити точні прогнози щодо значення різних знаків.

3. Сенсорні технології:

3.1. Сенсорні рукавички: сенсорні рукавички оснащені датчиками або пристроями для захоплення руху, які фіксують рухи рук і пальців користувача. Ці датчики визначають положення, орієнтацію та згинання пальців, а також загальний рух рук. Потім отримані дані обробляються та використовуються для розпізнавання мови жестів. Одним із прикладів сенсорних рукавичок є «CyberGlove», розроблений CyberGlove Systems LLC. Щоправда, як зазначено на веб-сайті компанії ціль рукавиць не є безпосередньо спрямована на розпізнавання мови жестів, а швидше на

машинно-людську взаємодію з комп'ютером. «CyberGrasp — це інноваційна система зворотного зв'язку за силою для ваших пальців і рук, яка дозволяє вам «дотягнутися до свого комп'ютера» та схопити створені комп'ютером об'єкти або об'єкти, керовані телескопічно» [17].

3.2. Електроміографія (ЕМГ): Електроміографія передбачає використання датчиків, розміщених на передпліччі або руці користувача, для вимірювання електричної активності м'язів, залучених до створення мови жестів. Датчики ЕМГ виявляють м'язові скорочення та рухи, пов'язані з певними ознаками. Потім ці сигнали аналізуються та інтерпретуються для розпізнавання намічених ознак. Прикладом комп'ютеризованих рукавичок є «Dexmo Haptic Glove», розроблена Dexta Robotics. «Dexmo — це найпортативніша у світі рукавичка зі зворотним зв'язком, створена для дослідників, підприємств і споживачів» [37]. Вони поєднують тактильний зворотний зв'язок із відстеженням рук і пропонують застосування у віртуальній реальності та взаємодії людини з машиною.

3.3. Комп'ютеризовані рукавички: Комп'ютеризовані рукавички поєднують датчики руху та датчики ЕМГ, щоб фіксувати рух рук і м'язів. Датчики руху відстежують рух і орієнтацію рук, а датчики ЕМГ вимірюють активність м'язів. Поєднуючи ці джерела даних, комп'ютеризовані рукавички забезпечують повний вхід для систем розпізнавання мови жестів. «На етапі збору даних ЕМГ отримують від датчиків ЕМГ, які можуть бути частиною саморобних або комерційних пристроїв» [30].

4. Системи на основі камери:

4.1. Комп'ютерне бачення на основі камер: системи на основі камер використовують алгоритми комп'ютерного бачення для аналізу відеоданих, знятих камерами або веб-камерами. Ці алгоритми відстежують рухи рук та інших частин тіла користувача, розпізнаючи певні жести та знаки. Захоплюючи та аналізуючи просторову та часову інформацію з відео, системи на основі камери можуть інтерпретувати слова мови жестів у реальному часі.

4.2. Kinect або датчики глибини: датчики глибини, такі як Microsoft Kinect або подібні пристрої, фіксують інформацію про глибину середовища. Ці датчики дозволяють відстежувати рухи рук і тіла в трьох вимірах. Використовуючи дані про глибину, ці системи можуть точно відстежувати позиції та жести рук і тіла користувача, полегшуючи розпізнавання мови жестів.

5. Електроенцефалографія (ЕЕГ):

5.1. Електроенцефалографія передбачає запис електричної активності мозку за допомогою датчиків, розміщених на шкірі голови. У контексті розпізнавання мови жестів ЕЕГ можна використовувати для виявлення та інтерпретації патернів мозку, пов’язаних з різними жестами мовою жестів. Алгоритми машинного навчання зазвичай використовуються для аналізу та класифікації цих сигналів ЕЕГ, що дозволяє розпізнавати передбачувані ознаки на основі активності мозку.

Ці технології мають свої переваги та обмеження. Деякі з них можуть бути обмежені в точності розпізнавання, особливо при складних жестах або в умовах обмеженого освітлення. Під час використання таких систем також важливо враховувати аспекти конфіденційності та безпеки.

1.2.1. Переваги та виклики існуючих методів

Розробляючи систему розпізнавання мови жестів, було враховувано конкретні потреби цільової аудиторії користувачів та контекст програми. Нижче наведено переваги і недоліки даних технологій.

1. Комп'ютерне зорове сприйняття:

- Переваги: Може фіксувати та аналізувати рухи та жести руки на основі відео чи зображень, що дозволяє розпізнавати їх у реальному часі. Забезпечує точні результати при тренуванні на великих наборах позначених даних.

- Виклики: Залежить від хороших умов освітлення та чіткої видимості рухів руки. Може мати проблеми з приховуванням чи складними фонами.

2. Машинне навчання:

- Переваги: Може навчитись складним моделям та варіаціям мови жестів за допомогою позначених наборів даних. Має потенціал для високої точності та узагальнення нових жестів.

- Виклики: Вимагає значних об'ємів позначених даних для тренування. Може потребувати частої перетренування для адаптації до нових варіацій жестів або жестів, специфічних для користувача.

3. Технології на основі датчиків:

3.1. Датчикові рукавички:

- Переваги: Забезпечують точне відстеження рухів рук і пальців. Здатні фіксувати широкий спектр рухів рук і артикуляцій пальців.

- Виклики: Вимагає носіння спеціалізованих рукавичок, що може бути не зручно. Може мати обмеження у фіксації незначних рухів м'язів або жестів, що включають інші частини тіла.

3.2. Електроміографія (ЕМГ):

- Переваги: Фіксує рухи м'язів, пов'язані з жестами . Надає пряме вимірювання активності м'язів, що дозволяє точніше розпізнавати жести. Датчик ЕМГ на основі поверхневих електродів дуже швидкий і простий у застосуванні. Не потребує медичного нагляду та не потребує сертифікації.

- Виклики: Вимагає правильного розміщення та калібрування сенсорів. Обмежена здатність фіксувати рухи м'язів та може не фіксувати інші аспекти рухів рук або тіла. Застосовується тільки для поверхневих м'язів. Немає стандартного розміщення електродів, що в свою чергу вимагає нестандартизованих дій для користувача. Область виявлення може не відображати весь м’яз, що призводить до неточності даних.

3.3. Комп'ютерні рукавички:

- Переваги: Забезпечують комплексний вхід, фіксуючи як рухи рук, так і м'язів. Надають більш детальне представлення жестів мови жестів.

- Виклики: Вимагає носіння спеціалізованих рукавичок. Калібрування та синхронізація декількох типів сенсорів можуть бути складними.

4. Системи на основі камери:

4.1. Камерна система комп'ютерного зору:

- Переваги: Аналізує рухи рук і жести на основі відеоданих. Може в реальному часі відстежувати рухи рук без використання сенсорів.

- Виклики: Вимагає налаштування камери або веб-камери. Може бути позначено умовами освітлення та приховуванням.

4.2. Датчики глибини:

- Переваги: Фіксують інформацію про глибину для відстеження рухів рук і тіла у трьох вимірах. Забезпечують точне відстеження позицій та жестів.

- Виклики: Вимагає специфічного апаратного забезпечення для сприйняття глибини. Обмежений діапазоном і полям зору датчиків. Нижче наглядно показано принцип роботи датчиків, а саме їхнє сприйняття осей, зокрема віртуальної глибини (Рис. 1.2) [10].

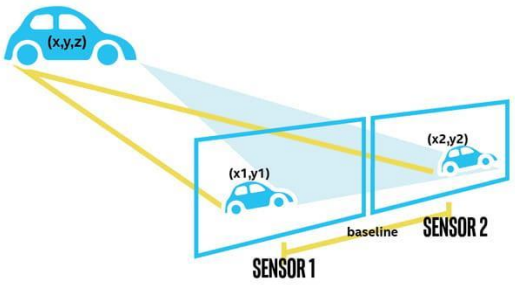



Рис. 1.2. Робота датчиків глибини

5. Електроенцефалографія (ЕЕГ):

- Переваги: Фіксує активність мозку, пов'язану з жестами. Може надати пряме нейрологічне вимірювання намірів.

- Виклики: Вимагає правильного розміщення та калібрування сенсорів. Може мати обмеження в точному фіксуванні складних жестів мови жестів. Дана галузь не є надто розвинутою і все ще знаходиться на стадії дослідження.

При виборі методу для розпізнавання жестів мови жестів, слід враховувати точність, обробку у реальному часі, комфорт користувача, простоту використання, наявність тренувальних даних та конкретні вимоги застосування. Важливо оцінити переваги та обмеження кожного методу та вибрати той, який найкраще підходить для бажаних цілей.

Важливо відзначити, що зазначені переваги та виклики є загальними і можуть варіюватися в залежності від реалізації, розвитку технологій та конкретних пропозицій

1.2.2. Економічний аналіз

У рамках дослідження розпізнавання жестів, методи були упорядковані за ціновими категоріями, що допоможе визначити оптимальне технічне обладнання. Нижче наведено приблизну класифікацію методів на основі вартості:

Висока цінова категорія:

1. Спеціалізовані системи для розпізнавання жестів, які використовують електроміографію (ЕМГ) та інші датчики. Вони забезпечують високу точність та деталізацію сигналів, але мають високу цінову вартість.

2. Професійні системи комп'ютерного зору, які включають в себе високоякісне обладнання, таке як високороздільні камери та датчики руху.

3. Комплексні рукавички зі вбудованими сенсорами для реєстрації рухів рук та пальців. Вони забезпечують докладний аналіз рухів, але також мають високу вартість.

Середня цінова категорія:

1. Компактні системи розпізнавання жестів, що використовують EMG-датчики та програмне забезпечення для обробки сигналів. Вони надають прийнятну точність за помірну ціну.

2. Камери середньої роздільної здатності та програмне забезпечення комп'ютерного зору для розпізнавання жестів. Вони забезпечують задовільну точність за доступну вартість.

3. Рукавички з обмеженим набором сенсорів та програмним забезпеченням для реєстрації рухів рук. Вони надають базовий аналіз рухів за помірну ціну.

Низька цінова категорія:

1. Використання загальноприйнятого обладнання, такого як веб-камери, ноутбуки або смартфони, з програмним забезпеченням комп'ютерного зору або спеціальними додатками для розпізнавання простих жестів. Вони мають низьку вартість, але також обмежену точність та функціональність.

Важливо зазначити, що це лише орієнтована класифікація, і фактичні ціни можуть суттєво варіюватися залежно від багатьох факторів, таких як виробник, постачальник, регіон та інші.

## 1.3. Вибір методів та приладів для реалізації рішення розпізнавання жестів

При виборі методів та приладів для розробки систем розпізнавання мови жестів важливо враховувати кілька факторів, які включають наступне. А саме цінову категорію, доступність продукту, тобто необхідно взяти до уваги наскільки він є доступний на ринку, його розповсюдженість чи потребує спеціального замовлення на виготовлення приладу. Також важливим моментом є ступінь готовності використання, тобто наскільки продукт ефективний чи сама технологія розвинута, мобільність технології та необхідність сторонньої допомоги експертів при її використані. Враховуючи

ці фактори, ми можемо зробити кращий вибір методів та приладів для розробки системи розпізнавання мови жестів. Рекомендовані методи та прилади включають датчикові рукавички, електроміографію, комп'ютерні рукавички, датчики глибини та камери комп'ютерного зору. Вибір конкретних технологій залежить від наших потреб, вимог проекту, часу та бюджету. На рисунку 1.3 зіставлена таблиця щодо плюсів та мінусів перерахованих вище технологій.

| | Доступність | Цінова категорія | Мобільність | Ступінь готовності використання | Необхідність асистента/допомоги експерта |
|---|---|---|---|---|---|
| Датчикові рукавички | Низька | Висока | Середня | Висока | +/- |
| Електроміографія (ЕМГ): | Низька | Висока | Низька | Низька | + |
| Комп'ютерні рукавички | Низька | Висока | Середня | Висока | +/- |
| Камерна система комп'ютерного зору | Висока | Низька | Висока | Висока | - |
| Датчики глибини | Середня | Низька | Середня | Висока | - |
| Електроенцефалографія (ЕЕГ): | Середня/Висока | Середня/Висока | Середня | Середня | +/- |

Рис. 1.3. Переваги та мінуси рішень

Доступність продукту: Перед вибором конкретного методу або приладу, слід переконатися, що вони доступні для придбання або використання. Деякі передові технології можуть бути доступні лише обмеженому колу дослідників або компаній. А саме продукти які відносяться до категорії датчикових та комп’ютерних рукавичок вимагають попереднього заповнення анкети та очікування виконання замовлення, те ж саме стосується деяких модель ЕЕГ, в той же час коли ЕМГ обмежений для вузькоспеціалізованого кола. Зате технології котрі підтримують рішення на основі комп’ютерної системи зору та датчиків глибини є досить легко доступними, камери або інтегровані в ноутбуки та телефони, або їх легко замовити при ураховані часу доставки.

Цінова категорія: Визначення бюджету, який ми готові витратити на розробку системи розпізнавання мови жестів, дуже важливо. Різні методи та

прилади мають різну цінову політику, і важливо знайти ті, які підходять під ваші фінансові можливості. «Середня вартість електроенцефалограми (ЕЕГ) у Сполучених Штатах без страхування коливається від 200 до 3000 доларів або більше. Звичайна ЕЕГ коштує дешевше, зазвичай від 200 до 700 доларів США, хоча витрати збільшуються, якщо вам потрібен посилений моніторинг» [19]. «Однак CyberGloveIII коштує близько 13 000 доларів США» [14]. Тому в таблиці вони були відведені до високої цінової категорії.

Ступінь готовності використання: При виборі технології варто з'ясувати, наскільки вона розвинута і які рівні точності та надійності можна очікувати. Якщо ви потребуєте високої точності в розпізнаванні жестів, то варто звернути увагу на ті методи та прилади, які демонструють високу якість роботи.

Мобільність: Якщо потрібно, щоб система була портативною і зручною для використання у різних умовах, то слід вибрати технології, які дозволяють рухатися вільно та не обмежувати користувача. Деякі прилади, такі як датчикові рукавички або комп'ютерні рукавички, можуть бути більш зручними для мобільного застосування порівняно з іншими методами, такими як ЕЕГ.

Необхідність сторонньої допомоги експертів: Врахуйте, чи потрібна допомога фахівців або експертів для використання обраного методу або приладу. Деякі технології можуть вимагати спеціалізованого навчання або допомоги від експертів для налагодження та коректної роботи.

Таким чином, доречним для подальшої роботи над проектом виявилася технологія глибинного зору. Виходячи з цього було необхідно виконати наступні технічні вимоги.

1.Камера глибини:

- Модель: Intel RealSense D435;
- Цінова категорія: Приблизно $150-200.

- Доречність використання: Камера глибини дозволить отримувати тривимірну інформацію про жести, що полегшить їхнє розпізнавання та покращить точність системи.

2.Nano камера:

- Модель: Raspberry Pi Camera Module V2
- Цінова категорія: Приблизно $25-30
- Доречність використання: Nano камера може використовуватися для захоплення відео жестів з ближньої відстані. Вона є доступною та компактною опцією для проекту.

3.Ноутбук:

- Модель: Наприклад, Dell XPS 13
- Цінова категорія: Від $1000 і вище
- Доречність використання: Ноутбук з потужним процесором та достатньою пам'яттю може виконувати обробку відео та алгоритми розпізнавання мови жестів. Він забезпечить швидку та ефективну обробку даних.

4.Raspberry Pi Zero:

- Модель: Raspberry Pi Zero W
- Цінова категорія: Приблизно $10-15
- Доречність використання: Raspberry Pi Zero є компактною та доступною платформою. Він може виконувати низькорівневі завдання та комунікувати з іншими компонентами проекту.

Загальні вимоги:

- Реалізація системи повинна забезпечувати високу точність розпізнавання мови жестів.

- Використання відкритих технологій та бібліотек рекомендується для забезпечення ширшої підтримки та спільноти розробників.

Даний список описує основні компоненти та їхні моделі для розробки системи розпізнавання мови жестів. Важливо врахувати, що моделі та цінова категорія можуть змінюватись з часом, тому дана інформація актуальна на момент 2023 року.

# РОЗДІЛ 2. РОЗРОБКА АЛГОРИТМУ ТА ПРОГРАМИ

## 2.1. Схема виявлення та розпізнавання жестів в реальному часі за допомогою модулю MediaPipe

Виявлення та розпізнавання жесту в реальному часі за допомогою MediaPipe є актуальною темою в галузі комп'ютерного зору та машинного навчання. «MediaPipe — це фреймоворк з відкритим вихідним кодом від Google для побудови мультимодальних (наприклад, відео, аудіо, будь-які дані часового ряду), міжплатформних (тобто Android, iOS, веб, периферійні пристрої) прикладних конвеєрів машинного навчання» [18].

Цей процес має великий потенціал для різноманітних застосувань, таких як інтерактивні інтерфейси, розумні системи контролю та аналізу жестів мови. Була створена зведена таблиця з аналізом які рішення з використанням бібліотеки в машинному навчанні можна побудувати, відповідно до модулів, завдань та мов програмування. Адже, для таких завдань, як орієнтири руки, райдужної оболонки ока чи обличчя, визначення долоні, визначення пози, зіставлення функцій на основі шаблону (KNIFT) або виявлення 3D-об’єктів (Objectron), більшість рішень MediaPipe використовують різні моделі TFLite (TensorFlow Lite). У табл. 2.1. наведено оптимальність використання модулів фреймоворку відповідно до мови та платформи.

Таблиця 2.1.

ML рішення в MediaPipe

| | C++ | JS | Python | Android | iOS | Coral |
|---|---|---|---|---|---|---|
| **Face Detection** | + | + | + | + | + | + |
| **Face Mesh** | + | + | + | + | + | - |
| **Iris Detection** | + | - | - | + | + | - |

Продовження табл. 2.1.

| | C++ | JS | Python | Android | iOS | Coral |
|---|---|---|---|---|---|---|
| **Face Detection** | + | + | + | + | + | + |
| **Face Mesh** | + | + | + | + | + | - |
| **Iris Detection** | + | - | - | + | + | - |
| **Hands Detection** | + | + | + | + | + | - |
| **Pose Detection** | + | + | + | + | + | - |
| **Holistic** | + | + | + | + | + | - |
| **Selfie Segmantation** | + | + | + | + | + | - |
| **Hair Segmentation** | + | - | - | + | - | - |
| **Object detection** | + | - | - | + | + | + |
| **Box Tracking** | + | - | - | + | + | - |
| **Instant Motion Tracking** | - | - | - | + | - | - |
| **Objectron(3D)** | + | + | + | + | - | - |
| **KNIFT** | - | - | - | + | - | - |
| **MediaSequence** | + | - | - | - | - | - |
| **YouTube 8M** | + | - | - | - | - | - |
| **AutoFlip** | + | - | - | - | - | - |

Наведена вище таблиця містить інформацію про різні функції та можливості рішень у сфері машинного навчання, доступних у модулі MediaPipe. Кожна колонка представляє одну з шести платформ або технологій: C++, JavaScript (JS), Python, Android, iOS та Coral. «Фреймворк MediaPipe в основному використовується для швидкого створення прототипів конвеєрів сприйняття з моделями штучного інтелекту для

логічного висновку та іншими повторно використовуваними компонентами» [25].

Кожен рядок описує конкретну функцію, таку як виявлення обличчя, мережу обличчя, виявлення ірисів, виявлення рук, виявлення пози, голістичне виявлення, сегментацію селфі, сегментацію волосся, виявлення об'єктів, відстеження об'єктів, миттєве відстеження руху, 3D-моделювання об'єктів (Objectron), KNIFT, MediaSequence, YouTube 8M, AutoFlip.

Знак «+» у кожній клітинці таблиці показує, що відповідна функція підтримується на відповідній платформі або технології. Наприклад, використаний у даній кваліфікаційні роботі голістичний модуль, підтримується C++, JavaScript (JS), Python та платформами Android з iOS. Не варто забувати і про ОС для персональних комп’ютерів, зокрема Windows і MacOs, але так як дана таблиця була націлена на зазначення доречних платформ для мобільних та компактних приладів, ОС для ПК не були вказані.

Всі модулі, зазначені в таблиці, пропонують різноманітні можливості для обробки даних. У нашому випадку окрім методу виявлення ключових точок, було використано ще три пов’язаних з лицем, руками та тілом. Інші модулі, характеристики яких коротко буде розібрано нище, наведені для наглядного розгляду з точки зору придатності для платформ або можливості реалізації на відповідній мові програмування у випадку вирішення підключення додаткових модулів до системи розпізнавання жестів щоб розширити функціонал.

1. Face Detection (виявлення обличчя): Ця функція дозволяє виявляти обличчя на вхідних зображеннях або відео. Вона підтримується на всіх платформах і технологіях (C++, JS, Python, Android, iOS та Coral). «Розпізнавання обличчя є першим і важливим кроком для розпізнавання обличчя, і воно використовується для виявлення обличь на зображеннях. Це частина виявлення об’єктів і може використовуватися в багатьох сферах, таких як безпека, біометрія, правоохоронні органи, розваги, особиста безпека тощо» [15].

2. Face Mesh (мережа обличчя): Ця функція дозволяє виявляти ключові точки обличчя та побудовувати мережу, яка описує форму обличчя. Вона також підтримується на всіх платформах і технологіях. «Модель розпізнавання обличчя: визначає наявність облич за кількома ключовими орієнтирами. Сітчаста модель обличчя: додає повне відображення обличчя. Модель видає оцінку 478 тривимірних орієнтирів обличчя» [16].

3. Iris Detection (виявлення ірисів): Ця функція дозволяє виявляти іриси очей на вхідних зображеннях або відео. Вона підтримується в Python, Android та iOS. «Розпізнавання райдужної оболонки або сканування райдужної оболонки ока — це процес використання видимого та ближнього інфрачервоного світла для отримання висококонтрастної фотографії райдужної оболонки ока людини» [35].

4. Hands Detection (виявлення рук): Ця функція дозволяє виявляти руки на вхідних зображеннях або відео. Вона підтримується на всіх платформах і технологіях. Принцип роботи моделі аналогічний до мережі обличчя та виявлення пози.

5. Pose Detection (виявлення пози): Ця функція дозволяє виявляти людську позу на вхідних зображеннях або відео. Вона підтримується на всіх платформах і технологіях.

6. Holistic (голістичне виявлення): Ця функція комбінує виявлення обличчя, ключових точок обличчя, виявлення рук та виявлення пози для забезпечення повного голістичного аналізу людського тіла. Вона підтримується на всіх платформах і технологіях. «MediaPipe Holistic забезпечує оцінку пози, рук і обличчя» [20].

7. Selfie Segmentation (сегментація селфі): Ця функція дозволяє виділяти основний об'єкт (людину) на задньому плані селфі-фотографій або відео. Вона підтримується на всіх платформах і технологіях. «API сегментації селфі ML Kit дозволяє розробникам легко відокремлювати фон від користувачів у межах сцени та зосереджуватися на тому, що має значення» [33].

8. Hair Segmentation (сегментація волосся): Ця функція дозволяє виділяти волосся людини на вхідних зображеннях або відео. Вона підтримується в C++ та Android.

9. Object Detection (виявлення об'єктів): Ця функція дозволяє виявляти різні типи об'єктів на вхідних зображеннях або відео. «Виявлення об’єктів - це область комп’ютерного зору, яка займається локалізацією та класифікацією об’єктів, що містяться на зображенні чи відео» [36].

10. Box Tracking (відстеження об'єктів): Ця функція дозволяє відстежувати об'єкти протягом часу на вхідних відео. «Відстеження об’єктів - це завдання взяття початкового набору виявлень об’єктів, створення унікального ідентифікатора для кожного з початкових виявлень, а потім відстеження кожного з об’єктів, коли вони рухаються по кадрах у відео, зберігаючи призначення ідентифікатора» [27].

11. Instant Motion Tracking (миттєве відстеження руху): Ця функція дозволяє відстежувати рух об'єктів в реальному часі на вхідних відео.

12. Objectron (3D-моделювання об'єктів): Ця функція дозволяє створювати 3D-моделі об'єктів на основі вхідних зображень або відео. Вона підтримується на всіх платформах і технологіях. «MediaPipe Objectron - це мобільне рішення для виявлення 3D-об’єктів у реальному часі для повсякденних об’єктів. Він виявляє об’єкти на двовимірних зображеннях і оцінює їхні пози за допомогою моделі машинного навчання (ML), навченої на наборі даних Objectron» [24].

13. KNIFT (Knowledge of Neighbouring Frames for the Interpolation of Time) є системою відстеження камери, яка дозволяє відстежувати рух камери та синхронізувати кадри. Це може бути корисно для створення плавного переходу між кадрами, синхронізації камер у багатокамерних системах тощо.

14. MediaSequence використовується для обробки послідовностей медіаданих, таких як відео або зображення. Цей компонент надає інструменти для маніпулювання послідовностями кадрів, їх обрізання, зміни розміру, збільшення швидкості, знімання зайвих кадрів тощо.

15. YouTube 8M є величезним набором даних, який містить відео з YouTube, анотовані метаданими, такими як класи об'єктів і етикетки. Цей набір даних можна використовуватись для навчання моделей машинного навчання, аналізу відео та розпізнавання об'єктів.

16. AutoFlip: AutoFlip - це інструмент автоматичного кадрування, який може виявляти та кадрувати відео або зображення автоматично. Він використовує алгоритми комп'ютерного зору для визначення найкращого кадрування, що допомагає поліпшити якість та естетику медіаконтенту.

Загалом ця таблиця є зведеною інформацією про доступні функції в проекті MediaPipe та показує, які з них підтримуються на різних платформах або технологіях. Як було сказано не всі компоненти таблиці використовуються, але вони є актуальними для аналізу та вибору додаткових методів у разі подальшої роботи над системою розпізнавання мови жестів для глухих людей.

Процес навчання нейронної мережі та повернення результатів включає імпортування модулів, підготовку вхідних даних, виявлення об'єктів, витягнення ознак, тренування моделі та розпізнавання жесту у реальному часі. Для досягнення поставленої цілі зокрема використовувалося бібліотеку OpenCV. «OpenCV (Open Source Computer Vision Library) - бібліотека комп’ютерного зору та машинного навчання з відкритим кодом. OpenCV було створено, щоб забезпечити загальну інфраструктуру для програм комп’ютерного зору та прискорити використання машинного сприйняття в комерційних продуктах» [9].

Схему роботи даного процесу наведено нижче (Рис. 2.1) [31]

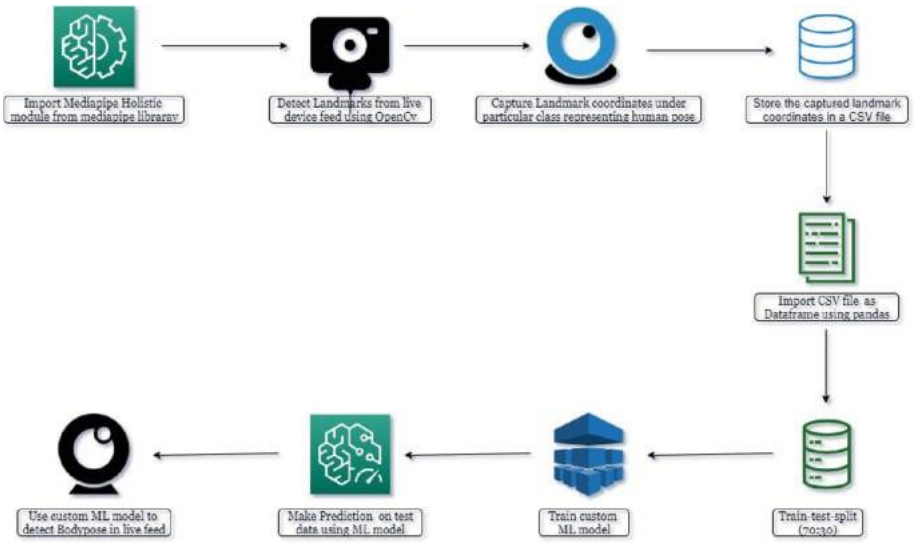



Рис. 2.1. Робота системи розпізнавання

Перший крок у процесі виявлення та розпізнавання жесту - імпортування необхідних модулів з MediaPipe та додаткових бібліотек, таких як OpenCV та TensorFlow. Ці модулі забезпечують основні функціональні можливості для обробки відео та машинного навчання.

Після імпорту модулів необхідно підготувати вхідні дані для обробки. Це може включати захоплення відеопотоку за допомогою веб-камери або завантаження відео з файлу. Перед обробкою вхідні кадри можуть були попередньо оброблені, тобто зменшені за розміром та нормалізовані.

Після підготовки вхідних даних використовуються модулі MediaPipe для виявлення об'єктів у відеопотоці. Ці модулі використовують глибинні нейронні мережі для точного виявлення обличь, рук та ключових точок. Результатом цього етапу є точне відстеження рухів та позицій об'єктів на кожному кадрі.

Після виявлення об'єктів можна витягнути ознаки, які відображають структуру та положення жестів. Це включає витягнення векторів ознак з ключових точок рук, обличчя та тіла під час руху та в стані спокою. Отримані ознаки слугують важливою інформацією для наступних етапів.

Після витягнення ознак можна було тренувати модель машинного навчання для розпізнавання жесту. У даній кваліфікаційній роботі була

використана рекурентна нейронна мережа (RNN) з LSTM-шарами, здатна аналізувати послідовність даних та здійснювати класифікацію. Під час тренування моделі використовувалися навчальні дані, які містили вхідні кадри відео та відповідні мітки жестів. «Шар виконує адитивні взаємодії, які можуть допомогти покращити градієнтний потік у довгих послідовностях під час навчання» [23].

Після тренування моделі можна використовувати її для розпізнавання жестів у реальному часі. Вхідні кадри подаються на вхід моделі, яка виконує передбачення класу або категорії жесту. Це може відбуватися шляхом порівняння витягнутих ознак зі заздалегідь вивченими патернами жестів або за допомогою глибоких нейронних мереж, що використовуються для класифікації.

Нарешті, коли жест розпізнаний, можна виконати відповідні дії або перекладати жест на мову або команду. Це може включати виконання дій в інтерактивному середовищі або спілкування з комп'ютерними системами за допомогою розпізнаних жестів.

2.1.1. Збір даних для моделі машинного навчання

Точність системи машинного навчання залежить напряму від характеристик та якості вхідних даних. У традиційній розробці програмного забезпечення, код має пріоритет перед навчальними даними. «Навчальні дані (або навчальний набір даних) — це вихідні дані, які використовуються для навчання моделей машинного навчання» [42]. Тепер, хоча програмування все ще є необхідним для розвитку машинного навчання, акцент зміщується на дані.

Наприклад, написання модульних тестів для перевірки коду є хорошою практикою при роботі над проектами традиційного програмного забезпечення. При роботі над проектами з машинним навчанням рахується хорошою практикою регулярно тестувати, підтверджувати та стежити за вхідними даними. Що й було здійснено при роботі з навчальною моделлю. При першій спробі при перевірці даних було виявлено некоректні ознаки

жестів для «ні», що призвело до помітно низької точності. Поведінка моделі швидко змінилися у негативний спосіб при раптовій змінні доданих вхідних даних.

Як було зазначено, точність моделі нейронної мережі зазвичай впливає кількість даних, доступних для навчання. Тому при подальшій роботі запису та пошуку даних для підвищення точності моделі, було враховано кілька ключових моментів щодо зв'язку між кількістю даних і точністю моделі:

✔ Загалом наявність більшої кількості різноманітних навчальних даних покращує здатність моделі добре узагальнювати невидимі приклади. Маючи більше даних, модель може вивчати ширший діапазон шаблонів і варіацій, що призводить до підвищення точності.

✔ Уникання перенавчання. Перетренування відбувається, коли модель стає занадто спеціалізованою на певному типі навчальних даних і погано працює з новими, невідомими даними. Наявність невеликої кількості навчальних даних збільшує ризик перенавчання. Надаючи більший набір даних, можна допомогти зменшити переобладнання та підвищити здатність моделі до узагальнення. Також існує і інша причина. «У перетренуванні модель намагається вивчити занадто багато деталей у навчальних даних разом із шумом від навчальних даних. Як наслідок, продуктивність моделі дуже низька на невидимих або тестових наборах даних. Таким чином, мережа не може узагальнити характеристики або шаблони, наявні в наборі навчальних даних» [13].

✔ Зменшення віддачі. Хоча збільшення обсягу навчальних даних зазвичай призводить до кращої продуктивності, віддача може бути меншою. Спочатку, коли набір даних зростає, точність моделі значно покращується. Однак після досягнення певної точки покращення стає менш значним, а віддача непомітна.

✔ «Більшість моделей машинного та глибокого навчання значною мірою покладаються на кількість і різноманітність даних, щоб добре

працювати. Обсяг і різноманітність даних, наданих під час навчання, мають значний вплив на точність прогнозування цих моделей» [4].

✔ Якість даних і маркування: якість даних і точність їх маркування також відіграють вирішальну роль. Важливо переконатися, що дані позначені правильно та послідовно. Шумні або неправильно позначені дані можуть негативно вплинути на точність моделі, навіть із великим набором даних. «Викиди – це дані, які суттєво відрізняються від інших даних у вашому наборі даних. Значна кількість викидів знижує продуктивність вашого алгоритму» [11].

✔ У ситуаціях, коли доступні обмежені позначені дані, передача навчання або попередньо підготовлені моделі можуть бути цінними. Використовуючи моделі, попередньо навчені на великомасштабних наборах даних, ви можете скористатися їхніми навченими функціями та підвищити точність своєї моделі з меншими обсягами даних.

Підводячи підсумок, хоча існує загальна тенденція, згідно з якою збільшення обсягів даних може підвищити точність моделі, важливо враховувати такі фактори, як якість даних, різноманітність, репрезентативність і уникнення перенавчання. Співвідношення між кількістю даних і точністю може відрізнятися залежно від конкретної проблеми, набору даних і архітектури моделі, що використовується. «Найпростіший спосіб запобігти перенавчанні — почати з простої моделі, моделі з меншою кількістю параметрів, які можна навчити. Іншими словами, місткість моделі повинна бути менше» [12]. Відповідно, на етапі збору даних спочатку ми виконуємо код для створення навчальної бази. Під час потокового відео він фіксує рух рук 30 кадрів на один файл даних. Таким чином, для однієї дії потрібно 30 кадрів. Таким чином, вхідні дані, які використовуються для цієї моделі виявлення дії, являють собою серію з 30 масивів, кожен з яких містить значення. Кожен із 30 масивів представляє значення орієнтирів з одного кадру.

Нижче представлено етап запису та маркування навчальних даних (Рис. 2.2) [38].

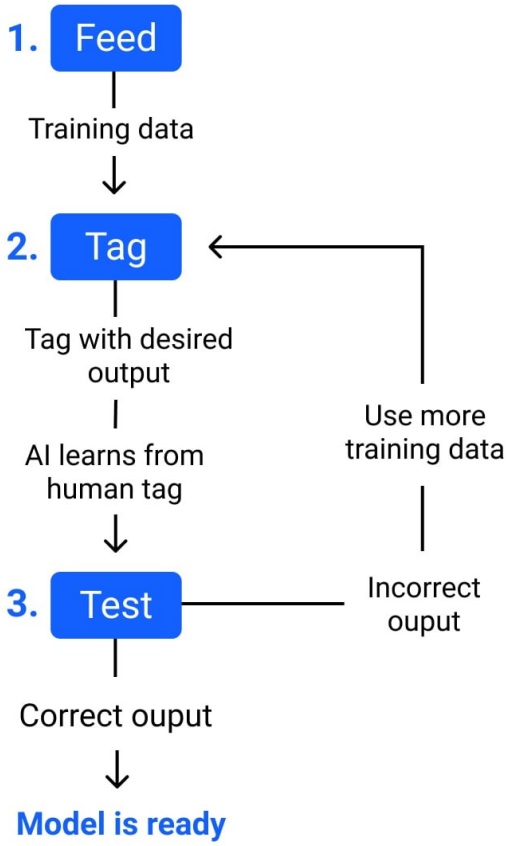



Рис. 2.2. Схема обробки даних

Був записаний власний набір даних через проблеми з пошуком готового та позначених навчальних даних для мого завдання та вибраного методу. Робота з немаркованим набором даних викликала би труднощі та ризики, котрі я мала враховувати. Причини для запису власного набору даних були наступні:

- Контроль маркування: було надано перевагу контролюванню процесу маркування, щоб забезпечити точне та послідовне маркування відповідно до конкретних вимог.

- Налаштування набору даних: запис власного набору даних дозволив адаптувати його до конкретних вимог мого проекту, зафіксувавши необхідні знаки жестової мови, а саме «так», «ні», «привіт».

- Якість та різноманітність даних: було звернено увагу на правильність записаних даних та забезпечення різноманітності, звертаючи увагу на такі фактори, як зміну освітлення, фону, кути камери та рук під час процесу запису.

Запис мого власного набору даних вимагав значних зусиль і інвестицій часу, але це забезпечило більший контроль над маркуванням, використанням знань предметної області, налаштуванням набору даних, якістю та узгодженістю даних, а також можливість раніше розпочати перевірку якості роботи навчальної моделі. Таким чином після зібрання понад 27 тисяч маркованих файлів для кожного з жестів, було приступлено до тренування моделі. На рисунку 2.3 зображено процес тренування з відображенням коливання точності.

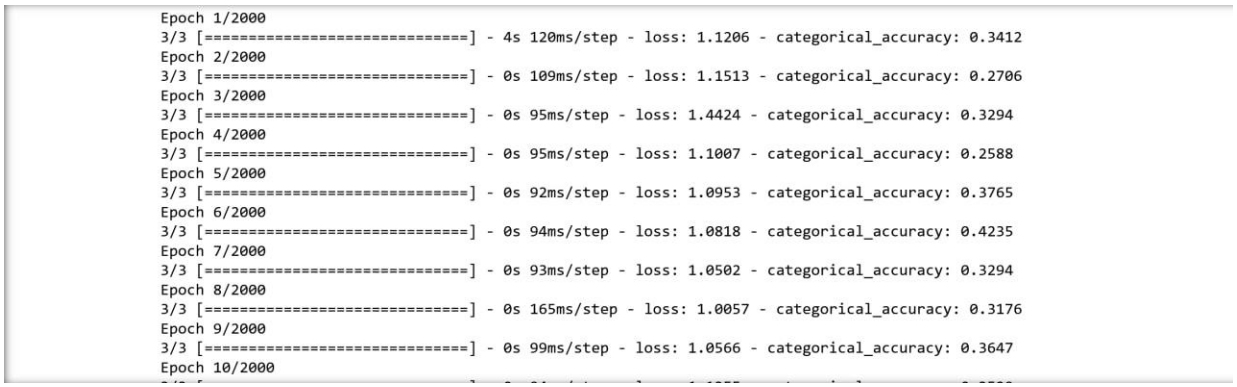



Рис. 2.3. Тренування даних

### 2.1.2. Обробка відеосигналу

Для обробки відеосигналу проект в основному покладається на бібліотеку Python MediaPipe та виконується в програмному середовищі Jupyter Notebook. Дана платформа, зокрема заточена під машинне навчання. «Jupyter Notebook дозволяє розробникам машинного навчання візуалізувати, а також ділитися кодом і наборами даних» [21].

У проекті бібліотека використовується для відстеження функцій тіла, таких як кістки, суглоби та м’язи, оскільки вони є ключем до виявлення жестів і пантоміми, необхідних для розпізнавання мови жестів. Ці

особливості тіла називають цілісними ключовими точками або орієнтирами. Нижче показано приклад зображення з нанесеними на нього цілісними ключовими точками (Рис. 2.4).

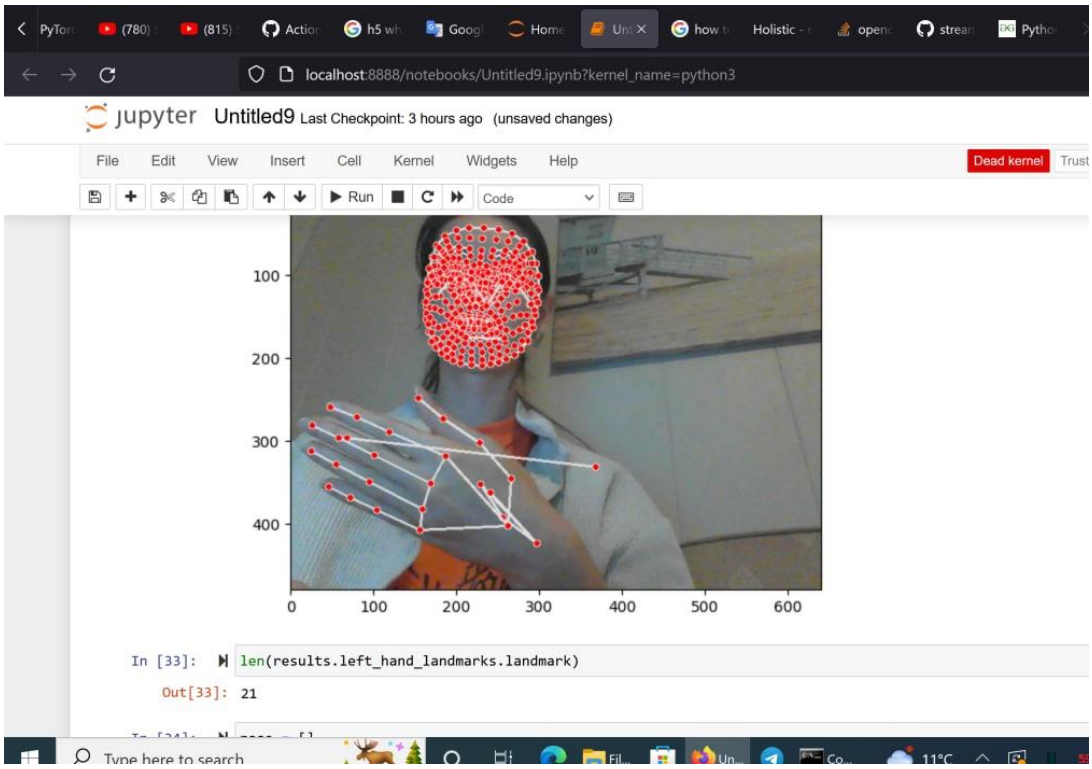



Рис. 2.4. Цілісна обробка ключових точок

Ключові моменти обчислюються за допомогою трьох різних моделей: одна для визначення орієнтирів обличчя, друга для визначення всіх суглобів рук і остання для визначення поз усередині нашого тіла. Перша модель збирає 21 точок даних на руку, друга – 468 точок для обличчя, а третя – 33 для тіла.

Ключові точки складаються з координат по трьох осях. Осі x і y — це ширина та висота зображення. Вісь z — це розрахункова віртуальна глибина. Ключові моменти зберігаються як масиви numpy і можуть бути збережені у файлі для подальших кроків. «NumPy (Numeric Python) надає базові методи для маніпулювання великими масивами і матрицями» [6]. Нижче подано принцип обробки вхідних даних (Рис. 2.5).

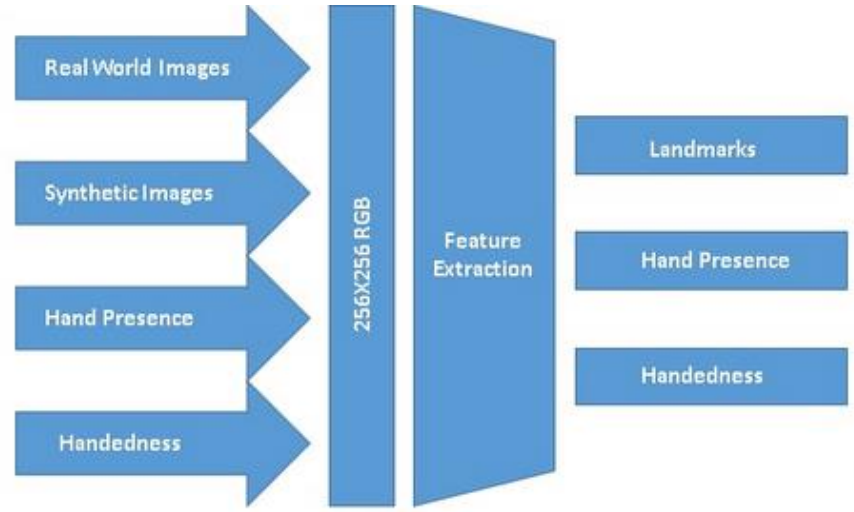



Рис. 2.5. Архітектура моделі екстракту орієнтирів руки

Однією з ключових оптимізацій MediaPipe є те, що детектор долоні запускається лише за потреби, заощаджуючи значний час обчислень. Це досягається шляхом визначення розташування руки в наступних кадрах відео з обчислених ключових точок руки в поточному кадрі, що усуває необхідність запускати детектор долоні в кожному кадрі. Щоб зменшити час обчислення, модель ручного відстеження підтверджує, чи присутня рука та достатньо вирівняна у вхідному кадруванні.

Після відстеження долоні по всьому зображенню модель орієнтира руки виконує точну локалізацію координат орієнтира в межах виявлених областей руки за допомогою регресії. Модель вивчає послідовне внутрішнє представлення пози рук і є надійною навіть для частково видимих рук і самоокклюзій.

Написаний код здійснює захоплення відео з веб-камери за допомогою OpenCV і обробку кадрів в реальному часі з використанням Mediapipe, що дозволяє виявляти ключові точки на обличчі, руках і тілі людини.

Основні кроки, які виконує код:

- Імпортує необхідні бібліотеки, включаючи NumPy, os, mediapipe та cv2 (OpenCV).

- Визначає класи mp_holistic та mp_drawing для використання функціоналу Mediapipe для виявлення ключових точок та візуалізації.
- Визначає допоміжну функцію mediapipe_detection, яка отримує кадр, перетворює його в RGB-простір кольорів і застосовує модель Mediapipe для виявлення ключових точок на зображенні.
- Визначає функцію draw_styled_landmarks, яка отримує зображення та результати виявлення ключових точок і візуалізує їх за допомогою функцій mp_drawing.draw_landmarks.
- Визначає функцію extract_keypoints, яка отримує результати виявлення ключових точок і повертає розгорнутий масив NumPy, що містить координати ключових точок для обличчя, рук і тіла.
- Визначає константи та змінні для шляху збереження даних, переліку дій (actions), кількості послідовностей (no_sequences) та довжини послідовності (sequence_length).
- Створює папки для збереження даних у відповідності зі структурою DATA_PATH/action/sequence.
- Ініціалізує відеозахоплення (cap) з веб-камери.
- У циклі for action in actions здійснюється ітерація по діях.
- У циклі for sequence in range(start_folder, start_folder+no_sequences) здійснюється ітерація по послідовностях.
- У циклі for frame_num in range(sequence_length) здійснюється ітерація по кадрах.
- Зчитується кадр з відеозахоплення.
- Застосовується функція mediapipe_detection для отримання результатів виявлення ключових точок на зчитаному кадрі.

- Застосовується функція draw_styled_landmarks для візуалізації ключових точок на зчитаному кадрі.
- Відображається зчитаний кадр з візуалізованими ключовими точками.
- Виконується експорт ключових точок за допомогою функції extract_keypoints. Збережені ключові точки зберігаються у форматі NumPy у відповідному шляху DATA_PATH/action/sequence/frame_num.
- Перевіряється, чи натиснута клавіша «q», яка дозволяє вийти з програми.
- Після завершення циклів звільняються ресурси камери та закриваються вікна.

Це дозволяє здійснювати захоплення відео з веб-камери, виявлення ключових точок на кадрах і збереження їх у відповідних папках для подальшого використання у тренуванні моделі.

Незважаючи на те, що цей метод досить складний, все ж можна обчислити орієнтири в реальному часі. Таким чином, вибір цілісного виявлення об'єктів видається розумним з точки зору здатності сприймати форму та рух частин тіла людини та розпізнавати АЖМ.

### 2.2. Підготовка Raspberry-pi(RPI) для запуску потокового відео

«Raspberry Pi — це недорогий комп’ютер розміром з кредитну картку, який під’єднується до комп’ютерного монітора чи телевізора та використовує стандартну клавіатуру та мишу. Це потужний маленький пристрій, який дозволяє людям будь-якого віку досліджувати комп’ютерну техніку та навчитися програмувати такими мовами, як Scratch і Python. Він здатний робити все, що ви очікуєте від настільного комп’ютера, від перегляду веб-сторінок і відтворення відео високої чіткості до створення електронних таблиць, обробки текстів та ігор» [40].

Raspberry Pi є вдалим вибором та часто використовується для створення невеликих проектів з кількох причин:

Доступна ціна. Плата Raspberry Pi відносно недорога порівняно з іншими обчислювальними платформами, що робить її доступною для хобістів, студентів і професіоналів з обмеженим бюджетом. «Якщо ви купуєте Raspberry Pi у схваленого торгового посередника, ви побачите, що вони все ще досить доступні: Raspberry Pi 4B на 8 ГБ коштує 75 доларів, а Raspberry Pi 3 – модель A+ (PLUS) можна придбати за 25 доларів» [43].

Універсальність: Плати Raspberry Pi дуже універсальні і можуть бути використані для різних проектів. Вони здатні запускати повноцінну операційну систему, таку як Linux, і підтримують різноманітні мови програмування, що дозволяє розробляти на них широкий спектр застосувань.

Компактний розмір: Плати Raspberry Pi являються маленькими і компактними, що робить їх ідеальними для проектів з обмеженим простором або вимогами до мобільності. Вони легко поміщаються в невеликі корпуси або можуть бути інтегровані в інші пристрої.

«GPIO (Інтерфейс введення / виводу загального призначення (англ. General-purpose input / output) це інтерфейс для спілкування з будь-якими зовнішніми пристроями і управління ними. Контакти GPIO можуть виступати як в ролі входу, так і в ролі виходу» [1]. Піни GPIO (General Purpose Input/Output): На платах Raspberry Pi є піни GPIO, що дозволяють підключати та керувати зовнішніми електронними компонентами, такими як сенсори, приводи, світлодіоди та двигуни. Це дозволяє легко взаємодіяти з фізичним світом і створювати інтерактивні проекти. «Кожна модель Raspberry Pi має набір контактів введення/виведення загального призначення (GPIO) уздовж верхнього краю плати. Вони можуть використовуватися для підключення та зв’язку з усіма видами електронних компонентів, діючи як фізичний інтерфейс між Raspberry Pi і зовнішнім світом» [28].

Спільнота та підтримка: Raspberry Pi має велику та активну спільноту користувачів та розробників. Це означає, що є багато онлайн-ресурсів,

посібників та проектів, завдяки чому легше навчитися, усунути проблеми та отримати допомогу у потребі.

Освітня цінність: Raspberry Pi широко використовується в освітніх закладах для викладання комп'ютерних наук, програмування та електроніки. Її доступність, легкість використання і універсальність роблять її чудовим інструментом для навчання та експериментування.

Енергоефективність: Плати Raspberry Pi споживають дуже мало енергії, що робить їх енергоефективними і підходящими для проектів з низьким споживанням енергії або роботи від батарей.

Загалом, Raspberry Pi є відмінним вибором для малих проектів завдяки своїй доступності, універсальності, компактному розміру, підтримці GPIO, спільноті та освітній цінності. Незалежно від того, чи ви початківець, чи досвідчений творець, Raspberry Pi надасть потужну та гнучку платформу для втілення ваших ідей у життя.

«Raspberry Pi чудово працює для повсякденних обчислювальних завдань, таких як веб-перегляд і офісні програми» [26].

Для того щоб розпочати роботу на наявному Raspberry Pi необхідно встановити операційну система. В нашому випадку з огляду на старшу модель приладу було використано Raspberry Pi OS (раніше називалася Raspbian), котра являється офіційно підтримуваною операційною системою.

Хоча важливо зазначити, що для Raspberry Pi доступно кілька операційних систем і це надзвичайно зручно з огляду на тестування проекту:

Raspbian, котра була вибрана для роботи є офіційною операційною системою, спеціально розробленою для Raspberry Pi. Вона базується на Debian Linux і має оптимізації для роботи з апаратним забезпеченням Raspberry Pi. «Найбільш часто використовуваним дистрибутивом у всіх raspberry є Raspberry Pi OS (спочатку відома як Raspbian), похідна від Debian» [29].

Ubuntu. Також доступний офіційний варіант операційної системи Ubuntu для Raspberry Pi. Він заснований на Ubuntu Linux і має велику

спільноту користувачів і підтримку. Зокрема, «Ubuntu MATE для Raspberry Pi надає повне знайоме робоче середовище, яке можна використовувати для базових настільних комп’ютерів» [39].

Windows 10 IoT Core Це спеціальна версія операційної системи Windows 10, розроблена для Інтернету речей (IoT) і підтримується Raspberry Pi. Вона дозволяє розробляти програми з використанням інструментів і мов програмування Microsoft.

Це лише кілька прикладів операційних систем, які можна використовувати на Raspberry Pi. Список може змінюватися, оскільки постійно розробляються нові дистрибутиви та варіанти операційних систем для Raspberry Pi.

Повертаючи нашу увагу до реалізації проекту на даному пристрої, щоб виконувати подальші кроки пов’язані з запуском розробленого програмного забезпечення необхідно налаштувати операційну систему (ОС). На цьому етапі для Raspberry Pi, слід дотримуватися таких загальних кроків:

Завантаження вибраної ОС для Raspberry Pi з офіційного веб-сайту або інших довірених джерел. Наприклад, у нашому випадку було обрано Raspbian, його можна завантажити з веб-сайту Raspberry Pi Foundation.

Наступним кроком буде підготовка SD-карти. Вставляємо SD-карту в комп'ютер і відформатовуємо її за допомогою програми для форматування (наприклад, SD Card Formatter). «SD Memory Card Formatter – програма, яка дозволяє форматувати різні SD карти. Вона надається для користування абсолютно безкоштовно і є найбільш якісною утилітою, яка працює з більшістю видів карт: SDHC/SDXC, SD, microSD тощо. За допомогою SD Memory Card Formatter ви зможете відформатувати будь-який подібний носій інформації і зробити наступний запис на нього надійнішим. Ця програма є універсальним засобом очищення пам'яті і повністю відповідає офіційним вимогам» [32]. Перед форматуванням перевіряємо, що була зроблена резервна копія важливих даних на SD-карті, оскільки цей процес стре всі

наявні дані. На рисунку 2.6 зображено вікно Raspberry Pi Imager, утиліти для вибору ОС, а наший варіант виділений синьою рамкою.

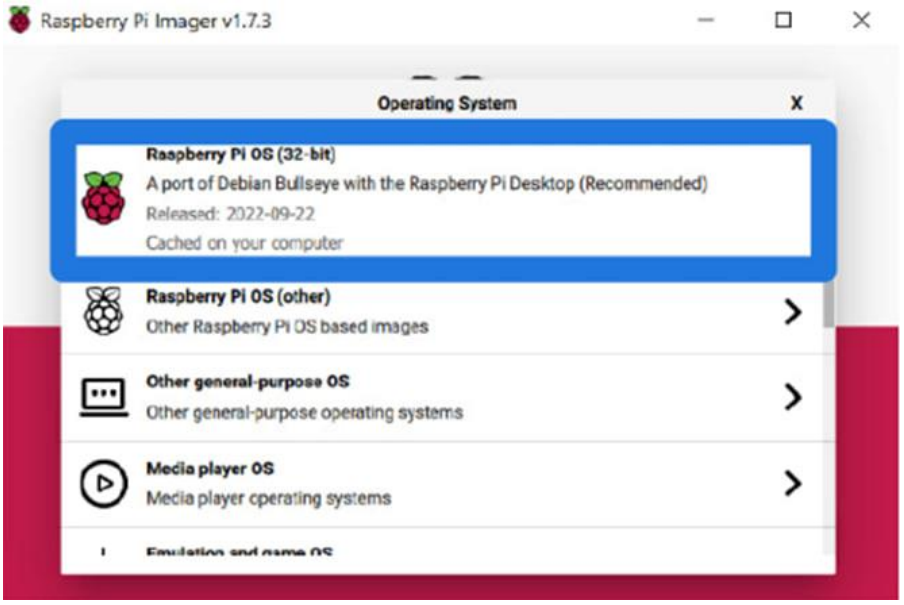



Рис. 2.6. Вікно Raspberry Pi Imager

«Raspberry Pi Imager - це інструмент, створений основою Raspberry Pi, який дозволяє нам встановлювати операційні системи (ОС) на нашу raspberry pi простим покроковим способом» [34].

Далі записуємо вибрану операційну систему на SD-карту. Після вибору файлу, а саме образу ОС і цільову SD-карту, розпочинаємо процес запису. Це створить завантажувальну SD-карту з ОС.

Після чого переходимо до налаштування Wi-Fi. При підключенні Raspberry Pi до Wi-Fi-мережі, потрібно налаштувати параметри Wi-Fi. Настройки можуть відрізнятися відповідно від моделі приладу, так як в нашому випадку ми використовуємо Raspberry Pi Zero то вибираємо наступні елементи. По-перше, встановлюємо своє унікальне ім’я хоста (ідентифікатор RPI замість IP-адреси), а саме «user04», далі вмикаємо SSH. «SSH або Secure Shell або Secure Socket Shell — це мережевий протокол, який допомагає нам безпечно отримувати доступ і спілкуватися з віддаленими машинами (переважно віддаленими серверами).

Він забезпечує надійне шифрування та широко використовується мережевими адміністраторами та розробниками для керування віддаленими системами та програмами, виконання команд, обміну файлами тощо» [7]. Встановлюємо ім'я користувача «group4» та пароль. Після чого ввімкнули і налаштували Wi-Fi, заповнивши SSID (ім’я Wi-Fi) і пароль Wi-Fi потрібної мережі. З даними параметрами ми натискаємо «Зберегти» та «Записати», щоб інсталювати ОС на картку MicroSD.

Далі було потрібно підключити модуль камери до Raspberry Pi і вставити SD-карту. Використовуючи Pi Zero, ми відкрили засув на кінці Raspberry Pi і вставили стрічку чорним кінцем, спрямованим до них, а потім під’єднали її до зарядного пристрою, що входить у комплект. Однак покриття для доступного елемента Raspberry Pi було відсутнє. З цієї причини після вставлення SD-карти потрібно було її підтримувати, заради щільного з’єднання між камерою та RPI, під час усього процесу налаштування додаткових модулів для апаратного компонента та запуску коду для потокового відео на веб-сторінці.

Коли Raspberry Pi було налаштовано, був отриманий до нього доступ із ноутбука, підключивши його до тієї ж мережі та відкривши вікно терміналу. Потім було введено команду «ssh user04@group04.local», щоб підключитися до Raspberry Pi та пароль.

# РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ТА АНАЛІЗ ОТРИМАНИХ РЕЗУЛЬТАТІВ

### 3.1. Результати дослідження

Для перевірки результатів та точності моделі використовувалося матрицю плутанини та епохальну точність. «Матриця плутанини - це тип побудови таблиці, яка відіграє специфічну роль у машинному навчанні та пов'язаній з ним техніці. Це допомагає показати прогнозування та відкликання в системі, де відомі значення тестових даних» [8].Теплова карта, на рисунку 3.1 є візуалізацією матриці плутанини.

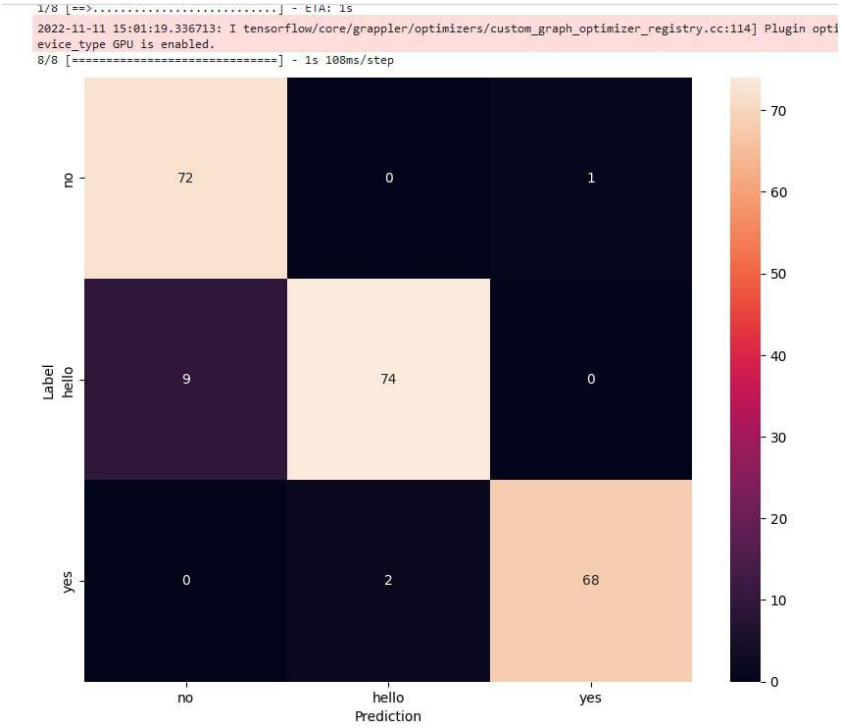



Рис. 3.1. Матриця плутанини

Вісь «y» ординат показує фактичний вид жесту, а вісь абсцисс «x» відображує вид жесту передбачений моделлю. Цифри по діагоналі означають кількість правильних прогнозів. Недіагоналі дають кількість невірних прогнозів. Наприклад, привіт (hello) неправильно передбачена як ні (no) 9 разів.

Для створення матриці плутанини зокрема використовуються наступні методи: 'sns.heatmap(confusion_mtx, xticklabels=commands, yticklabels=commands, annot=True, fmt='g'): Цей рядок створює теплову карту за допомогою бібліотеки seaborn (sns) і передає матрицю помилок

(confusion_mtx) як дані. Параметри xticklabels і yticklabels встановлюються на commands, що, ймовірно, представляє мітки класів. Параметр annot=True увімкнює підпис кожної комірки значенням, а fmt='g' вказує формат чисел для підписів як «загальний».

Використання матриці плутанини є важливим при перевірці результатів навчання моделі з декількома класами. Матриця плутанини надає детальну інформацію про те, як модель класифікує дані і які типи помилок вона робить. Це може підкреслити, де модель розпізнає не так і допоможе виявити які типи помилок вона робить найбільше. Таким чином це допомагає виявити її слабкі місця. Наприклад, ми бачимо, що модель найчастіше плутає жест привіт з жестом ні девять разів. Це має сенс, оскільки впорівняні з жестом так, котрий по суті виглядає як зжатий кулак, положення пальців в жесті «ні» у формі ножниць більш схоже на розкриту ладонь жесту «привіт».

Матриця плутанини та точність категорій за епохою (epoch categorical accuracy) є двома різними методами оцінки результатів навчання моделі. Епох це «міра того, скільки разів алгоритм бачив ті самі дані, і використовується для визначення точності моделі. Чим більше епох бачив алгоритм, тим точнішою буде модель. Епохи важливі в нейронних мережах, оскільки вони дозволяють моделі навчатися на основі даних і підвищувати свою точність» [41]. Тобто точність категорій за епох є метрикою, яка визначає загальну точність моделі на основі передбачених міток і справжніх міток. Вона вимірює відсоток правильно класифікованих прикладів загалом. Ця метрика дає загальне уявлення про точність моделі, але не надає докладної інформації про помилки та класифікацію для кожного окремого класу.

Спостереження росту та падіння хвиль на матриці епох дає змогу відслідковувати динаміку моделі під час навчання. Це допомагає визначити оптимальну кількість епох для навчання, контролювати перенавчання або недонавчання моделі та виявляти проблеми, які потребують втручання. Перевага полягає в тому, що спостереження росту хвиль може служити

показником прогресу та вказувати на покращення, тоді як падіння хвиль може вказувати на потенційні проблеми, які слід вирішити.

На рисунку 3.2 відображено коливання точності моделі під час її навчання на тренувальних даних.

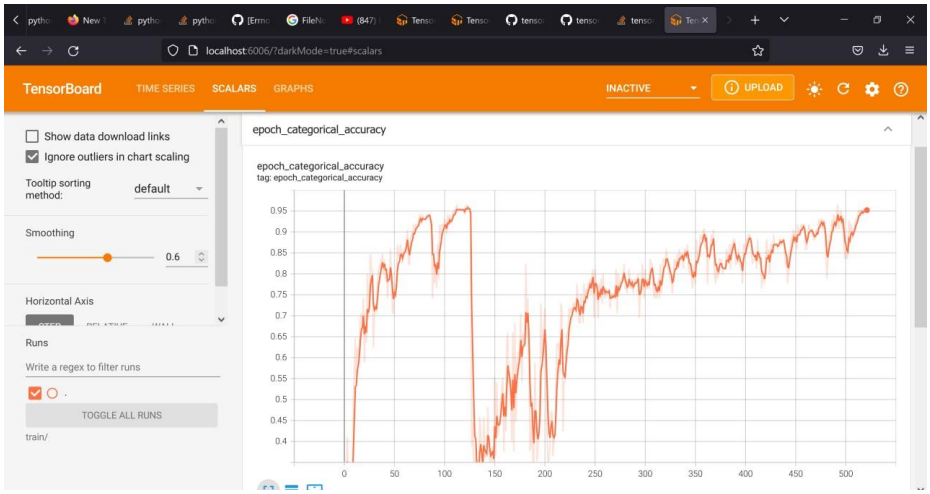



Рис. 3.2. Результати тренування моделі

Як можна помітити на рисунку вище, епох стверджує що точність коректного розпізнавання жестів моделі сягнула близькло 95%.

Отже, матриця плутанини і точність категорій за епохою є доповнюючими метриками, які можуть бути використані для аналізу результатів навчання моделі. Матриця плутанини надає більш детальну інформацію про класифікацію, в той час як точність категорій за епохою дає загальний огляд продуктивності моделі. Зазвичай, використання обох метрик разом допомагає отримати більш повне уявлення про якість і ефективність моделі.

Був створений інтуїтивно зрозумілий інтерфейс. Одразу після запуску програми вмикається камера та зокрема ідентифікує чи в полі зору знаходяться руки. Якщо так, відбувається процес лейблювання ключових точок та оброблення даної інформації щоб повернути результат розпізнавання. Так як для початку роботи було вибрано три базових та найчастіше використовуваних слова, вони відображаються на інтерфейсі в лівому верхньому кутку та підкреслені різними кольорами. Саме коливаня

цих полосок на фоні слів відображають процес обчислень нейромережі, тобто коливання процентної точності. Коли жест визначений він виводиться на синій полосі зверху, котра рухається при умові її повного заповнення словами (Рис. 3.3).

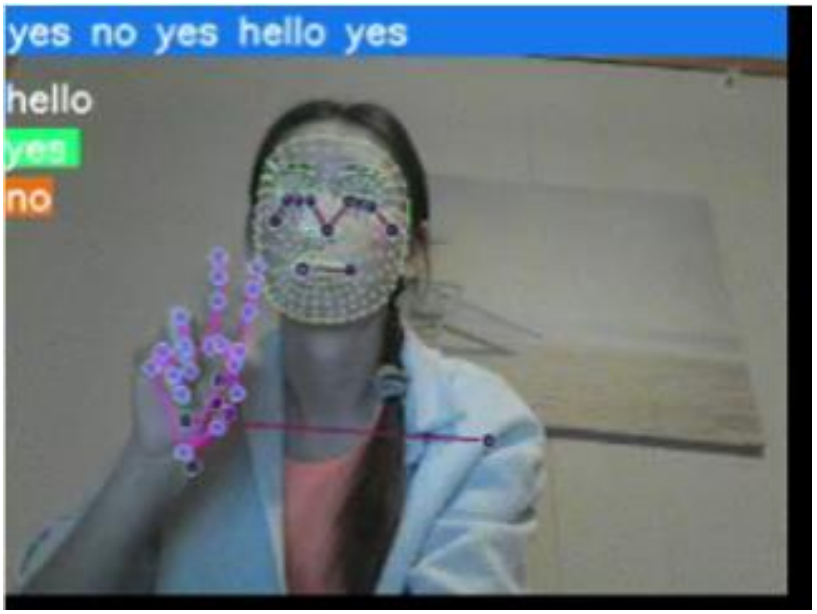



Рис. 3.3. Інтерфейс системи розпізнавання мови жестів

Таким чином жести розпізнаються в режимі реального часу. Також стрім розпізнавання мови жестів можна наприклад реалізувати на веб-сторінці. Для запуску потокового відео за допомогою Raspberry Pi та камери, нам потрібно використати команду sudo nano. На рисунку 3.4 зображено успішний запуск в терміналі.

```
Running NeuralMet Stream on: 192.168.50.185 Port: 8080
192.168.50.155 - - [23/Mar/2023 12:35:49] "GET /index.html HTTP/1.1" 200 -
192.168.50.155 - - [23/Mar/2023 12:35:49] "GET /stream.mjpg HTTP/1.1" 200 -
quit
end
```

Рис. 3.4. Запуск команди

Тепер ми знову підключенні до RPI через SSH, з створеним та відредагованим файлом з кодом у терміналі, ввівши: « sudo nano camera.py ». Запустивши сценарій та ввівши « python camera.py » ми відповідно

запускаємо код, котрий відповідає за потокове відео та вид веб-сторінки і це зображено нижче на рисунку 3.5.

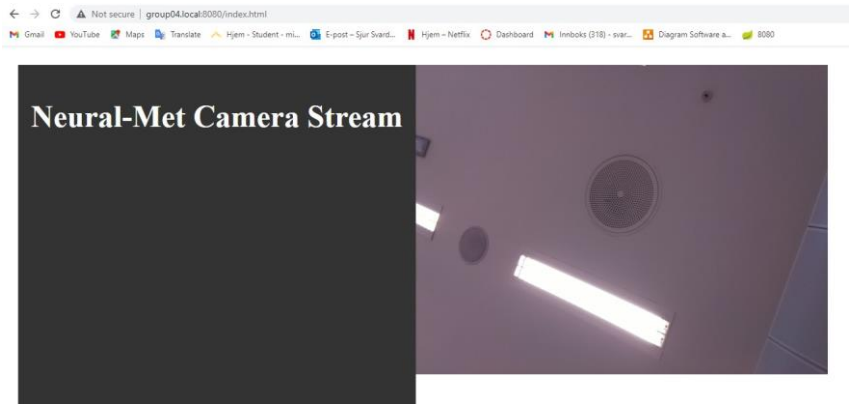



Рис. 3.5. Запуск потокового відео

Таким же чином, запускаючи файл з розпізнаванням мови жестів на веб-сайтах пов’язаних з ресурсами для глухих людей чи для навчання мови Глухих, після вдосконалення та розширення лексичного запасу в програмному забезпеченні, його можна реалізувати як додаток на подібних веб-сайтах.

## ВИСНОВКИ

Під час виконання кваліфікаційного проекту було розроблене програмне забезпечення для розпізнавання мови жестів з використанням ключових точок бібліотеки MediaPipe є потужним інструментом для реалізації рішення визначення та інтерпретації жестів. Вона дозволяє здійснювати точне визначення положення ключових точок на руках та враховувати їх взаємозв'язок для визначення значення жесту. Розробка та

подальше покращення подібного типу програмного забезпечення є надзвичайно важливим завданням. Адже це полегшить інтеграцію Глухих в суспільство, знизить витрати на перекладачі, підвищить загальний рівень якості життя за рахунок кращої та більш доступної освіти, вищих шансів працевлаштуватися та полегшення користування публічними сервісами. Результат продуктивності системи показує, що можна побудувати систему перекладу мови жестів, використовуючи лише камери та шари LSTM. Після навчання моделі таким словам, як «привіт», «так» і «ні», було помічено, що зі збільшенням словникового запасу вона схильна плутати кілька схожих знаків один з одним. Деякі покращення можна ввести, використовуючи більш багатий набір даних для навчання. Не лише за рахунок збільшення даних, але й за допомогою додаткових умов і форм рук.

Зараз дизайн відносно простий. Це може мати свої переваги, адже візуалізація деяких результатів на графічному інтерфейсі, включно з інформаційною панеллю про коливання вірогідності значень жесту зі спрощеним керуванням машиною та візуалізацією даних у реальному часі, додає значної цінності проекту.

Отже, в процесі роботи на кваліфікаційною роботою було вивчено різні аспекти машинного навчання та поглиблено знання щодо підготовки тренувальних даних і потреб цільової аудиторії - Глухих людей.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Лекція 3.2 Моноплатні ПК для систем керування. URL: https://elearn.nubip.edu.ua/mod/book/view.php?id=347836&chapterid=120690 (дата звернення: 13.05.2023).

2. Порадник Savoir Vivre по відношенню до людей з інвалідністю. URL: URL: https://bon.wspa.pl/wp-content/uploads/2020/11/Poradnik-UA-www.pdf (date of access: 30.05.2023).

3. Про неприпустимість вживання терміну «глухонімий» по відношенню до людей з інвалідністю. URL: https://ap.uu.edu.ua/article/293 (дата звернення: 12.03.2023).

4. Розширення даних: необхідне для моделей машинного навчання. URL: https://hashdork.com/uk/%D0%B7%D0%B1%D1%96%D0%BB%D1%8C%D1%88%D0%B5%D0%BD%D0%BD%D1%8F-%D0%B4%D0%B0%D0%BD%D0%B8%D1%85/ (дата звернення: 02.05.2023).

5. Супрун спростовує міф про єдину мову жестів. URL: https://www.ukrinform.ua/rubric-culture/2543744-suprun-sprostovue-mif-pro-edinu-movu-zestiv.html (дата звернення: 12.03.2023).

6. ТЕМА 1. NumPy. URL: https://kkite.pnu.edu.ua/wp-content/uploads/sites/50/2020/04/NumPy.pdf (дата звернення: 02.05.2023).

7. Що таке SSH і як він працює?. URL: https://techukraine.net/%D1%89%D0%BE-%D1%82%D0%B0%D0%BA%D0%B5-ssh-%D1%96-%D1%8F%D0%BA-%D0%B2%D1%96%D0%BD-%D0%BF%D1%80%D0%B0%D1%86%D1%8E%D1%94/ (date of access: 27.05.2023).

8. Що таке матриця плутанини? - визначення з техопедії. URL: https://uk.theastrologypage.com/confusion-matrix (дата звернення: 02.06.2023).

9. About. URL: https://opencv.org/about/ (date of access: 27.05.2023).

10. Beginner’s guide to depth (Updated). URL: https://www.intelrealsense.com/beginners-guide-to-depth/ (date of access: 10.05.2023).

11. Best practices for model training and serving in machine learning – Part 2. URL: https://kapernikov.com/best-practices-for-model-training-and-serving-in-machine-learning-part-2/ (date of access: 21.05.2023).

12. Best Practices for Training Deep Neural Networks in Deep Learning. URL: https://medium.com/swlh/best-practices-for-training-deep-neural-networks-in-deep-learning-8c69b6f8b140 (date of access: 21.05.2023).

13. Complete Guide to Prevent Overfitting in Neural Networks (Part-1). URL: https://www.analyticsvidhya.com/blog/2021/06/complete-guide-to-prevent-overfitting-in-neural-networks-part-1/ (date of access: 21.05.2023).

14. Evaluating the Effectiveness of Corrective Demonstrations and a Low-Cost Sensor for Dexterous Manipulation. URL: https://arxiv.org/pdf/2204.07631 (date of access: 02.06.2023).

15. Face Detection For Beginners. URL: https://towardsdatascience.com/face-detection-for-beginners-e58e8f21aad9 (date of access: 22.05.2023).

16. Face landmark detection guide. URL: https://developers.google.com/mediapipe/solutions/vision/face_landmarker (date of access: 22.05.2023).

17. Featured Hardware. URL: http://www.cyberglovesystems.com/ (date of access: 08.04.2023).

18. Getting Started with MediaPipe on reTerminal. URL: https://wiki.seeedstudio.com/reTerminal_ML_MediaPipe/ (date of access: 28.04.2023).

19. How Much Does an Electroencephalogram (EEG) Cost Without Insurance?. URL: https://www.talktomira.com/post/how-much-does-an-electroencephalogram-eeg-cost-without-insurance (date of access: 02.06.2023).

20. INDIAN SIGN LANGUAGE RECOGNITION USING MEDIAPIPE HOLISTIC. URL: https://arxiv.org/pdf/2304.10256 (date of access: 22.05.2023).

21. Jupyter Notebook for Machine Learning – A Gentle Introduction. URL: https://viso.ai/deep-learning/jupyter-notebook-for-machine-learning/ (date of access: 23.05.2023).

22. Loneliness, social support, social isolation and wellbeing among working age adults with and without disability: Cross-sectional study. PubMed

Central (PMC). URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7403030/ (date of access: 03.04.2023).

23. lstmLayer. URL: https://www.mathworks.com/help/deeplearning/ref/nnet.cnn.layer.lstmlayer.html (date of access: 23.05.2023).

24. MediaPipe Objectron. URL: https://github.com/google/mediapipe/blob/master/docs/solutions/objectron.md (date of access: 22.05.2023).

25. MediaPipe: Google's Open Source Framework for ML solutions (2023 Guide) . URL: https://viso.ai/computer-vision/mediapipe/ (date of access: 28.04.2023).

26. Mini PC vs. Raspberry Pi: What's Best for Your Desktop?. URL: https://www.makeuseof.com/mini-pc-vs-raspberry-pi-whats-best-for-desktop/ (date of access: 25.05.2023).

27. Object Tracking. URL: https://paperswithcode.com/task/object-tracking (date of access: 22.05.2023).

28. Raspberry Pi and General-Purpose Input/Output. URL: https://www.futurelearn.com/info/courses/robotics-with-raspberry-pi/0/steps/75878 (date of access: 25.05.2023).

29. RaspberryPi. URL: https://wiki.debian.org/RaspberryPi#Can_I_put_Debian_on_my_Raspberry_Pi.3F (date of access: 22.05.2023).

30. Real-Time Hand Gesture Recognition Using Surface Electromyography and Machine Learning: A Systematic Literature Review . URL: https://www.mdpi.com/1424-8220/20/9/2467 (date of access: 20.04.2023).

31. Real-Time Human Pose Detection and Recognition Using MediaPipe. URL: https://link.springer.com/chapter/10.1007/978-981-16-7088-6_12 (date of access: 10.05.2023).

32. SD Memory Card Formatter 5.0.2. URL: https://biblprog.org.ua/ua/sd-memory-card-formatter/ (date of access: 22.05.2023).

33. Selfie segmentation. URL: https://developers.google.com/ml-kit/vision/selfie-segmentation (date of access: 22.05.2023).

34. Setting up your raspberry pi from scratch using Raspberry Pi Imager. URL: https://www.youngwonks.com/blog/Raspberry-Pi-Imager (date of access: 22.05.2023).

35. STREET-LEVEL SURVEILLANCE. URL: https://www.eff.org/pages/iris-recognition (date of access: 22.05.2023).

36. The Ultimate Guide to Object Detection. URL: https://www.v7labs.com/blog/object-detection-guide (date of access: 22.05.2023).

37. Touch the Untouchable. URL: https://www.dextarobotics.com/ (date of access: 08.04.2023).

38. Train and Test datasets in Machine Learning. URL: https://www.javatpoint.com/train-and-test-datasets-in-machine-learning (date of access: 11.05.2023).

39. Ubuntu MATE for Raspberry Pi. URL: https://ubuntu-mate.org/raspberry-pi/ (date of access: 22.05.2023).

40. What is a Raspberry Pi?. URL: https://www.raspberrypi.org/help/what-%20is-a-raspberry-pi/ (date of access: 23.05.2023).

41. What Is Epoch In Neural Network. URL: https://www.alibabacloud.com/topic-center/network/ghcxip4b4v-what-is-epoch-in-neural-network (date of access: 30.05.2023).

42. What Is Training Data in Machine Learning?. URL: https://monkeylearn.com/blog/training-data/ (date of access: 23.05.2023).

43. Why Are Raspberry Pis So Expensive?. URL: https://www.makeuseof.com/why-are-raspberry-pis-so-expensive/ (date of access: 25.05.2023).

____08.06.2023_____ _____
(дата) (підпис)

# ДОДАТКИ

**Додаток А**

```
import numpy as np
import os
import mediapipe as mp
import cv2

mp_holistic = mp.solutions.holistic # Holistic model
mp_drawing = mp.solutions.drawing_utils # Drawing utilities

def mediapipe_detection(image, model):
    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB) # COLOR CONVERSION BGR 2 RGB
    image.flags.writeable = False                  # Image is no longer writeable
    results = model.process(image)                 # Make prediction
```

```
    return results


def draw_styled_landmarks(image, results):
    # Draw face connections
    mp_drawing.draw_landmarks(image, results.face_landmarks, mp_holistic.FACEMESH_CONTOURS,
                              mp_drawing.DrawingSpec(color=(80,110,10), thickness=1, circle_radius=1),
                              mp_drawing.DrawingSpec(color=(80,256,121), thickness=1, circle_radius=1)
                              )
    # Draw pose connections
    mp_drawing.draw_landmarks(image, results.pose_landmarks, mp_holistic.POSE_CONNECTIONS,
                              mp_drawing.DrawingSpec(color=(80,22,10), thickness=2, circle_radius=4),
                              mp_drawing.DrawingSpec(color=(80,44,121), thickness=2, circle_radius=2)
                              )
    # Draw left hand connections
    mp_drawing.draw_landmarks(image, results.left_hand_landmarks, mp_holistic.HAND_CONNECTIONS,
                              mp_drawing.DrawingSpec(color=(121,22,76), thickness=2, circle_radius=4),
                              mp_drawing.DrawingSpec(color=(121,44,250), thickness=2, circle_radius=2)
                              )
    # Draw right hand connections
```

```
    mp_drawing.draw_landmarks(image, results.right_hand_landmarks, mp_holistic.HAND_CONNECTIONS,
                              mp_drawing.DrawingSpec(color=(245,117,66), thickness=2, circle_radius=4),
                              mp_drawing.DrawingSpec(color=(245,66,230), thickness=2, circle_radius=2)
                              )


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


# Path for exported data, numpy arrays
DATA_PATH = os.path.join("video_data")

# Actions that we try to detect
actions = np.array(['hello', 'yes', 'no', 'noise'])
```

```
# Thirty videos worth of data
no_sequences = 30

# Videos are going to be 30 frames in length
sequence_length = 30

# Folder start
start_folder = 0
offset = max(map(lambda x: int(x), os.listdir(os.path.join(DATA_PATH, actions[0])))) + 1

for action in actions:
    for sequence in range(no_sequences):
        try:
            os.makedirs(os.path.join(DATA_PATH, action, str(sequence + offset)))
        except:
            pass

cap = cv2.VideoCapture(0)
# Set mediapipe model
with mp_holistic.Holistic(min_detection_confidence=0.5, min_tracking_confidence=0.5) as holistic:

    # NEW LOOP
    # Loop through actions
    for action in actions:
        # Loop through sequences aka videos
        for sequence in range(start_folder, start_folder+no_sequences):
            # Loop through video length aka sequence length
```

```
        for frame_num in range(sequence_length):

            # Read feed
            ret, image = cap.read()
# Import necessary modules
import io                       # provides a Pythonic interface for streaming binary data
import picamera                 # provides a Python interface to the Raspberry Pi's camera module
import logging                  # provides a flexible event logging system for Python
import socketserver             # provides classes for creating network servers
import subprocess               # provides a way to spawn new processes, connect to their input/output/error pipes, and obtain their return codes
import sys                      # provides access to some variables used or maintained by the interpreter
from threading import Condition # provides a simple way to manage access to shared resources between threads
from http import server         # provides classes for implementing HTTP servers that can be used as web servers
from subprocess import check_output # provides a way to run a command and capture its output as a byte string

# Define a function to install Python packages/modules
def install(package):
    subprocess.check_call([sys.executable, "-m", "pip", "install", package])

# Call the install() function to install the required packages
install('picamera')
```

```
# HTML code to be displayed on webpage
PAGE="""\
<html>
<head>
<title> Neural-Met Camera Stream </title>
   <!-- Define CSS styles for the elements on the page -->
        <style type="text/css">
                .container{
                        display: flex;
                        flex-direction: row;
                        width: 100%;
                        height: 100%;
                        margin: 0 auto;
                        padding: 20px;
                        box-sizing: border-box;
                }
        .col-1{
           flex-basis: 50%;
           padding: 20px;
        }

        .col-2{
           flex-basis: 50%;
           padding: 20px;
        }

        .header{
           background-color: #333;
           color: #fff;
           font-size: 24px;
```

```
            font-weight: bold;
            text-align: center;
            padding: 20px;
        }
    </style>
</head>
<body>
<div class="container">
<div class="header"><h1> Neural-Met Camera Stream </h1></div>
<img src="stream.mjpg" width="640" height="480">
</div>
</body>
</html>
"""

# Create a class to output the video stream from the camera
class StreamingOutput(object):
    def __init__(self):
        self.frame = None
        self.buffer = io.BytesIO()
        self.condition = Condition()

    # Write the data to buffer
    def write(self, buf):
        if buf.startswith(b'\xff\xd8'):
            # Reset the buffer to store the new frame
            self.buffer.truncate()
            with self.condition:
                self.frame = self.buffer.getvalue()
                self.condition.notify_all()
```

```
            self.buffer.seek(0)
        return self.buffer.write(buf)

# Define the handler to handle HTTP requests
class StreamingHandler(server.BaseHTTPRequestHandler):
    def do_GET(self):
        # Check if the request is for the root path and redirect to index.htm
        if self.path == '/':
            self.send_response(301)
            self.send_header('Location', '/index.html')
            self.end_headers()
            # Check if the request is for index.html and send the HTML content to the client
        elif self.path == '/index.html':
            content = PAGE.encode('utf-8')
            self.send_response(200)
            self.send_header('Content-Type', 'text/html')
            self.send_header('Content-Length', len(content))
            self.end_headers()
            self.wfile.write(content)
            # Check if the request is for the video stream resource and send the stream data to the client
        elif self.path == '/stream.mjpg':
            self.send_response(200)
            self.send_header('Age', 0)
            self.send_header('Cache-Control', 'no-cache, private')
            self.send_header('Pragma', 'no-cache')
            # Adds the 'Content-Type' header to the response, specifying that the response will contain multiple parts
```

```
            # where each part is a new JPEG image and specifies the boundary string that separates each part in the response
            self.send_header('Content-Type', 'multipart/x-mixed-replace; boundary=FRAME')
            self.end_headers()
            try:
                while True:
                    # Uses the condition variable to wait for a new frame to be available
                    with output.condition:
                        output.condition.wait()
                        frame = output.frame
                    # Sends the boundary frame to indicate the start of a new frame
                    self.wfile.write(b'--FRAME\r\n')
                    self.send_header('Content-Type', 'image/jpeg')
                    self.send_header('Content-Length', len(frame))
                    self.end_headers()
                    # Sends the actual frame
                    self.wfile.write(frame)
                    # Adds a newline character to indicate the end of the current frame
                    self.wfile.write(b'\r\n')
            # Logs a warning if an exception is raised while streaming the frames
            except Exception as e:
                logging.warning(
                    'Removed streaming client %s: %s', self.client_address, str(e))
        # If the request does not match any of the above, return a 404 error
        else:
            self.send_error(404)
            self.end_headers()

# Define class for the HTTP server
```

```
class StreamingServer(socketserver.ThreadingMixIn, server.HTTPServer):
    allow_reuse_address = True
    daemon_threads = True

# Create a new instance of the PiCamera
with picamera.PiCamera(resolution='640x480', framerate=24) as camera:
    output = StreamingOutput()
    #Uncomment the next line to change your Pi's Camera rotation (in degrees)
    #camera.rotation = 90
    camera.start_recording(output, format='mjpeg')

    # get and format the IP address and port to run the server
    port = 8080
    ip= check_output(['hostname', '-I'])
    clean_ip = ip.strip()
    clean_ip = str(clean_ip).replace("b", "").replace("b", "")
    clean_ip = str(clean_ip).replace("'", "").replace("'", "")
    clean_ip = str(clean_ip).replace(" ", "").replace(" ", "")

    # print the IP address and port
    print("Running NeuralMet Stream on:", clean_ip, "Port:", port)

    # create and start the server using the address and port
    try:
        address = ('', port)
        server = StreamingServer(address, StreamingHandler)
        server.serve_forever()

    finally:
        camera.stop_recording()
```

```
        # Make detections
                results = mediapipe_detection(image, holistic)
                keypoints = extract_keypoints(results)
                # Draw landmarks
                draw_styled_landmarks(image, results)

                # Apply wait logic
                if frame_num == 0:
                    cv2.putText(image, 'STARTING COLLECTION', (120,200), #in the middle of screen
                                cv2.FONT_HERSHEY_SIMPLEX, 1, (0,255, 0), 4, cv2.LINE_AA) #chose font for text
                    #specify actions, text on the top
                    cv2.putText(image, 'Collecting frames for {} Video Number {}'.format(action, sequence), (15,12),
                                cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 255), 1, cv2.LINE_AA)
                    # Show to screen
                    cv2.imshow('OpenCV Feed', image)
                    cv2.waitKey(600) #break between each video
                else:
                    cv2.putText(image, 'Collecting frames for {} Video Number {}'.format(action, sequence), (15,12),
                                cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 255), 1, cv2.LINE_AA)
                    # Show to screen
                    cv2.imshow('OpenCV Feed', image)

                # NEW Export keypoints
                keypoints = extract_keypoints(results)
```

```
                    npy_path = os.path.join(DATA_PATH, action, str(sequence
+ offset), str(frame_num))
                    np.save(npy_path, keypoints)

                    # Break gracefully
                    if cv2.waitKey(10) & 0xFF == ord('q'):
                        break

        cap.release()
    cv2.destroyAllWindows()
```