МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет “Києво-Могилянська академія”

Факультет інформатики

Кафедра інформатики

КВАЛІФІКАЦІЙНА РОБОТА

освітній ступінь – бакалавр

на тему: **“МУЛЬТИМОДАЛЬНІ СИСТЕМИ: ДОСЛІДЖЕННЯ ІНТЕГРАЦІЇ РІЗНОМАНІТНИХ СЕНСОРНИХ МОДАЛЬНОСТЕЙ, ТАКИХ ЯК ЗВУКОВА ТА ВІЗУАЛЬНА, ДЛЯ МОЖЛИВОГО ПОКРАЩЕННЯ ВЗАЄМОДІЇ КОРИСТУВАЧІВ У МУЛЬТИМЕДІЙНИХ ПРОГРАМАХ”**

Виконала: студентка 4-го року навчання,

Освітньої програми «Комп’ютерні науки», 122

Станіславська Катерина Євгеніївна

Керівник Афонін А.О.,_____

кандидат фізико-математичних наук

Рецензент _____

(прізвище та ініціали)

Курсова робота захищена

з оцінкою _____

Секретар ЕК _____

«____» _____ 2024 р.

Київ – 2024

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра інформатики факультету інформатики

ЗАТВЕРДЖУЮ
Зав.кафедри інформатики,
проф., д.ф-м.н.
_____С. С. Гороховський (підпис)
„____”_____2024 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на кваліфікаційну роботу

студентці 4-го курсу факультету інформатики Станіславській Катерині Євгеніївні
ТЕМА: Мультимодальні системи: дослідження інтеграції різноманітних сенсорних модальностей, таких як звукова та візуальна, для можливого покращення взаємодії користувачів у мультимедійних програмах

Зміст текстової частини до кваліфікаційної роботи:

Зміст

Анотація

Вступ

Аналіз предметної області

Дизайн мультимодальних систем

Реалізація мультимодальної взаємодії

Висновки по роботі

Список використаних джерел

Дата видачі „___” _____ 2024 р.

Керівник _____ (підпис)

Завдання отримав _____ (підпис)

КАЛЕНДАРНИЙ ПЛАН ВИКОНАННЯ РОБОТИ

№ п/п	Назва етапу курсової роботи	Термін виконання етапу	Примітка
1.	Отримання завдання на кваліфікаційну роботу	жовтень 2023	
2.	Пошук та огляд літератури за темою роботи.	листопад 2023	
3.	Проведення дослідження	грудень 2023	
4.	Реалізація застосунку з мінімальним функціоналом	січень 2024	
5.	Написання текстової частини	лютий 2024	
6.	Аналіз отриманих результатів з науковим керівником	березень 2024	
7.	Фіналізація реалізованого застосунку	квітень 2024	
8.	Фіналізація текстової частини	травень 2024	
9.	Створення презентації до захисту	травень 2024	
10.	Внесення фінальних корективів у роботу	травень 2024	

АНОТАЦІЯ

Метою роботи є дослідження мультимодальних систем та інтеграції різних сенсорних модальностей, таких як аудіо та візуальна, для покращення взаємодії користувачів з мультимедійними додатками. Проведено аналіз предметної області, описано характеристики мультимодальних систем, включаючи приклади управління комп'ютером за допомогою когнітивної модальності. Зазначено основні аспекти дизайну мультимодальних систем, зокрема використання дизайн фреймворків, а також принципи орієнтованого на користувача дизайну.

Програмний застосунок задіює візуальну та звукову модальності, забезпечуючи зменшення когнітивного навантаження та покращуючи користувацький досвід. Робота підкреслює важливість інтеграції модальностей для створення інклюзивного цифрового середовища, що дозволяє користувачам ефективно взаємодіяти з комп'ютером.

Ключові слова: мультимодальні системи, модальність, інтеграція, взаємодія, користувацький досвід, зручність, доступність.

ЗМІСТ

ВСТУП

Актуальність. Мультимодальні системи зворотного зв'язку відіграють ключову роль у покращенні взаємодії з користувачами, активізуючи декілька органів чуття. Відповідно до теорій когнітивної психології, одночасна стимуляція різних сенсорних каналів підвищує ефективність запам'ятовування інформації, прискорює реакцію та зменшує когнітивне навантаження.

На практиці, системи з мультимодальним зворотним зв'язком забезпечують більш інтуїтивно зрозумілий досвід для користувача, особливо у сферах віртуальної (VR) та доповненої реальності (AR), де тактильний, аудіо- та візуальний зворотний зв'язок створюють захоплююче середовище. Так, VR симулятори для пілотів чи хірургів використовують мультимедійні системи для імітації реальних сценаріїв, покращуючи навчання за допомогою реалістичного мультисенсорного впливу. Сучасні смартфони використовують вібрацію та голосові підказки для полегшення навігації, а у постпандемічний період зросла популярність безконтактних інтерфейсів, які дозволяють зменшити фізичний контакт завдяки жестам і голосовому управлінню.

Актуальність даної теми з часом тільки зростає та відкриває нові можливості для користувачів. Спостерігаючи тенденції передових компанії, таких як Neuralink, можна зробити висновок, що наступною місією мультимодальної взаємодії є компенсація сенсорних порушень у людей.

Метою роботи є розробка програмного застосунку для розширення можливостей користувачів взаємодіяти з комп’ютером через мультимодальні системи.

Об’єктом дослідження є мультимодальна взаємодія з інтерфейсом комп’ютера. **Предметом дослідження** є швидкість, зручність та доцільність такої взаємодії.

Робота розділена на 3 частини. У першому розділі проведено аналіз предметної області з визначенням та описом характеристик мультимодальних систем. У другому розділі зазначено основні аспекти дизайну мультимодальних систем, включаючи узгодженість, зворотний зв'язок і запобігання помилок. У третьому розділі пояснюється процес розробки програми, яка розширює можливості взаємодії користувача з комп'ютером за допомогою візуальної, звукової та тактильної модальностей.

Постановка задачі:

1. Дослідити модальності доступні для взаємодії з персональним комп’ютером.
2. Проаналізувати характеристики та інтеграцію мультимодальних систем.
3. Розробити програму для мультимодальної взаємодії користувача з комп’ютером:
   a. Інтегрувати розпізнавання жестів
   b. Інтегрувати розпізнавання голосу
   c. Додати звуковий зворотній зв’язок від комп’ютера
   d. Імплементувати додаткові застосування модальностей
   e. Проаналізувати отримані результати

РОЗДІЛ 1: АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Огляд взаємодії з інтерфейсом комп’ютера, як приклад мультимодальної системи

Люди взаємодіють із навколишнім середовищем за допомогою п’яти основних органів чуття: зору, слуху, дотику, нюху та смаку. Сприйняття модальності стосується того, як навколишні стимули передаються через одне з чуттів. Таким чином, мультимодальні системи – це ті, що використовують комбінацію сенсорних модальностей, включаючи візуальну, звукову, тактильну, нюхову, смакову і навіть більше, про що йтиме мова далі.

Мультимодальна взаємодія в комп'ютерних системах використовує інтеграцію комунікативних каналів або модальностей для надання вхідних даних і отримання відповідей від системи. Як зазначає Тюрк, мультимодальні інтерфейси прагнуть використовувати природні людські здібності до спілкування через мову, жести, дотики та вирази обличчя, що збагачує процес взаємодії з комп'ютером складними методами розпізнавання образів і класифікації [1].

У своїй праці Тюрк висвітлив типи модальностей та приклади взаємодії в контексті кожної з них. Зі стрімкою появою нових технологій список каналів комунікації з персональним комп’ютером тільки зростає, а пристрої введення та виведення інформації стають дедалі більш інтегрованими в людське життя.

У розширеній таблиці нижче (таблиця 1.1) способами введення є ті, якими користувачі можуть надавати дані або команди комп’ютерній системі, а способи виведення відповідно передбачають, як комп’ютерна система обмінюється даними з користувачем.

Модальність	Введення		Виведення	
	Дія людини, приклади	Пристрій на вхід, приклади	Зворотний зв’язок, приклади	Пристрій зворотного зв’язку, приклади
Візуальна	Вираз обличчя, рух очима, рух губ, рух тіла, включаючи жести, тощо	Камера	Інформація у формі тексту, графіки, відео тощо	Дисплей
Звукова	Мова, немовний звук (наприклад, наспівування мелодії)	Мікрофон	Записаний звук	Динаміки вбудовані в систему або гарнітуру
Тактильна	Натиснення клавіш, сенсорних панелей або екранів, трекпаду; використання цифрового стилусу	Клавітура, мишка, цифровий стилус, трекпад, джойстік, сенсорний дисплей або панель і тд	Різного виду вібрації, електростимуляція	Вібраційні мотори, тактильний одяг, електро тактильні девайси, тактильні девайси (наприклад, дисплей Брайля)
Когнітивна	Думка	Нейрокомп'ю-терний інтерфейс	Передача інформації назад в мозок через закритий цикл нейрокомп'ю-терного інтерфейсу	Нейрокомп'ю-терний інтерфейс

Таблиця 1.1 Модальності введення та виведення при взаємодії за комп'ютером

Як показав Neuralink у своєму нещодавньому звіті про тестування мозкового імпланту (нейрокомп'ютерного інтерфейсу) вперше на людині, управління комп'ютером можливе силою думки [2]. Конкретно, Ноланд Арбо, 29-річний чоловік, повністю паралізований від плечей донизу протягом восьми років після нещасного випадку під час дайвінгу, за допомогою імпланту може управляти курсором миші, а також виконувати натискання правої та лівої кнопок миші за допомогою думки. Таким чином, можна спостерігати за появою нового типу модальності – когнітивного.

Іншим прикладом нетрадиційного пристроєм вводу інформації у тактильній модальності є MouthPad від Augmental. Це ротова пластина, яка дозволяє керувати персональними електронними пристроями за допомогою рухів язика, голови та дихання. Як заявляють розробники, натискання язиком використовуються для клацання лівої кнопки миші або клацання та перетягування. Ковток використовуються для клацання правою кнопкою миші. Розробка також має за мету допомогти людям з моторною дисфункцією взаємодіяти з інтерфейсом комп'ютера [3].

1.2. Характеристики мультимодальних систем

Мультимодальні системи, які об'єднують різні типи вхідних і вихідних модальностей для покращення взаємодії користувача з комп'ютером, демонструють кілька відмінних характеристик. Розуміння важливе для розробки зручних для користувача систем.

1. *Паралельність:* очевидно, мультимодальні системи можуть обробляти кілька модальностей введення одночасно. Наприклад, користувачі можуть

одночасно взаємодіяти за допомогою мови, жестів і дотиків, що робить взаємодію більш природною порівняно з унімодальними системами.

2. *Надмірність:* мультимодальні системи часто надають надлишкову інформацію в різних модальностях. Ця надлишковість може допомогти підтвердити дані, отримані одним каналом за допомогою іншої модальності. Така особливість підвищує надійність взаємодії та зменшує ймовірність помилок при розпізнаванні інформації на вхід.

3. *Взаємодоповнюваність* дозволяє різним модальностям забезпечити більш повне розуміння вхідних даних шляхом поєднання різноманітних типів інформації та швидшу обробку інформації користувачем, яка надається на вихід у вигляді декількох модальностей. Мультимодальне злиття має вирішальне значення для інтеграції вхідних даних для вирішення неоднозначностей і створення єдиної інтерпретації. Джеймес і Себе наголошують на необхідності складних методів синтезу, виділяючи історичні та сучасні підходи, які полегшують спільну обробку модальностей. Вони відзначають важливість контекстно-залежної обробки в спільному просторі функцій, яка вирішує такі проблеми, як велика розмірність і різні формати функцій [4].

4. *Адаптивність:* Мультимодальні системи, як правило, розроблені з урахуванням контексту, здатні інтерпретувати та адаптуватися до ситуації, середовища та потреб користувача. Ця характеристика дозволяє системі забезпечувати відповідний зворотний зв'язок і адаптувати своє функціонування до змін у реальному часі, а також робить її гнучкою в різних сценаріях користувача.

5. *Гнучкість:* релеватно до попереднього пункту, Тюрк зазначає, що адаптивні інтерфейси характеризуються своєю здатністю навчатися на основі взаємодії користувача [1]. Вони постійно оновлюють свої бази знань за допомогою неявного чи явного зворотного зв’язку від користувачів, а значить системи є гнучкими до їх потреб й уподобань.

1.3. Інтеграція мультимодальної взаємодії

1.3.1 Когнітивне навантаження

Ключовою теорією, яка має велике значення для розробки та ефективності мультимодальних систем, є теорія когнітивного навантаження (ТКН). Когнітивне навантаження стосується обсягу інформації, яку може одночасно вмістити оперативна пам’ять людини. Джон Свелер, що розробив дану теорію зазначає, що оскільки оперативна пам’ять має обмежену ємність, навчальні методи повинні уникати її перевантаження додатковими видами діяльності, які безпосередньо не сприяють навчанню [5].

ТКН виділяє три типи когнітивного навантаження:

1. *Внутрішнє навантаження* – це інформаційна складність, що включає кількість елементів, труднощі їх розуміння та їх взаємодію один з одним [6].

2. *Зовнішнє навантаження* – когнітивне навантаження, пов’язане не зі складністю завдання чи наданої інформації, а зі способом її надання та типами навчальних матеріалів, які її супроводжують [6].

3. *Доречне навантаження (Germane load)* – когнітивне навантаження, пов’язане з процесом обробки, конструювання та автоматизація схем, тобто моделей мислення чи поведінки, які організовують категорії інформації та зв'язки між ними.

Мультимодальні системи в свою чергу можуть допомогти керувати когнітивним навантаженням шляхом зменшення зовнішнього навантаження через представлення інформації за допомогою кількох модальностей (наприклад, візуальної та слухової) [7]. Де Йонг наголошує, що ця тактика є найбільш ефективною для коротких термінів навчання та при наявній системі контролю. Мультимодальні системи також оптимізують внутрішнє навантаження,

розподіляючи когнітивну обробку між різними сенсорними каналами. Наприклад, використання візуального, слухового та тактильного зворотного зв’язку одночасно може запобігти перевантаженню окремої сенсорної системи та сприяти швидшому навчанню або виконанню завдань [8]. Представляючи інформацію за допомогою кількох модальностей, підтримується створення надійних ментальних схем, що сприяє покращенню доречного навантаження. Користувачі можуть пов’язувати нову інформацію з наявними знаннями за допомогою сенсорних даних.

Таким чином, теорія когнітивного навантаження забезпечує міцну теоретичну основу для використання мультимодальних систем, оскільки вона підкреслює важливість управління когнітивним навантаженням для полегшення навчання та обробки інформації.

1.3.2 Техніки інтеграції

Інтеграція кількох модальностей передбачає кілька технічних методів, які покращують функціональність системи та взаємодію з користувачем. Основні методи включають: злиття даних, злиття сенсорів та алгоритмічна інтеграція.

Злиття даних передбачає об’єднання даних із кількох джерел для отримання більш узгодженої та точної інформації, ніж та, яку надає будь-яке окреме джерело [9]. У мультимодальних інтерфейсах методи злиття даних використовуються для об’єднання даних із різних модальностей введення, таких як мова, жест, погляд тощо. Поширені підходи злиття даних включають об’єднання:

- На рівні ознак: ознаки, отримані з різних модальностей, об’єднуються в один вектор ознак, який потім обробляється алгоритмом розпізнавання патернів.
- На рівні прийняття рішень: модальності обробляються окремо, а їхні результати об’єднуються за допомогою таких методів, як голосування, усереднення або моделі машинного навчання.

Злиття сенсорів – це тип злиття даних, який об’єднує дані з декількох датчиків для досягнення підвищеної точності та більш конкретних висновків, ніж можна було б досягти, використовуючи лише один датчик [10]. Методи злиття сенсорів включають:

- Фільтри Калмана, що використовуються для оцінки стану динамічної системи на основі серії неповних і “шумних” вимірювань від кількох датчиків.
- Фільтри частинок – це послідовні методи Монте-Карло, що використовуються для нелінійних, негаусових задач оцінки стану.

Алгоритмічна інтеграція відноситься до комбінації різних алгоритмів або моделей для обробки та інтерпретації мультимодального введення [10]. Вона може включати такі методи, як:

- Підходи на основі уніфікації, що характеризуються поєднанням даних за допомогою операції уніфікації над структурами ознак.
- Статистичні підходи, серед яких ймовірнісні моделі, наприклад приховані марковські моделі, байєсівські мережі або нейронні мережі, що використовуються для об’єднання мультимодальних даних.
- Гібридні підходи, що поєднують символічні і статистичні методи для мультимодальної інтеграції.

Вибір техніки інтеграції залежить від таких факторів, як задіяні модальності, область застосування, вимоги в режимі реального часу та доступність навчальних даних.

1.3.3 Міркування щодо доступності

Мультимодальність відіграє важливу роль у сприянні доступності та інклюзивності для користувачів з різними здібностями та потребами.

Дослідження Параторе та Лепоріні вказує, як мультимодальні взаємодії можуть пристосуватись до користувачів з вадами зору [11]. Наприклад, інтеграція тактильного зворотного зв’язку та звукових підказок у програмі для навігації може покращити когнітивне картографування та підвищити обізнаність про навколишнє середовище.

Подібним чином в огляді Ватаву та Унгур обговорюється, як поєднання розпізнавання голосу та взаємодії на основі жестів може принести значну користь користувачам з порушеннями моторики [12]. Ці модальності дозволяють контролювати комп’ютерними системами без використання рук, зменшуючи фізичні вимоги до користувача та надаючи альтернативу традиційним методам введення, таким як клавіатури та миші.

Іншим важливим аспектом доступності мультимодальних інтерфейсів є когнітивні та навчальні відмінності. Шнейдерман та ін. наголошують на необхідності враховувати індивідуальні особливості та потенціал когнітивного навантаження, коли користувачеві одночасно пропонують декілька модальностей [13]. Інтерфейс повинен забезпечувати гнучкість, щоб дозволити користувачам налаштувати рівень інтеграції певної модальності на основі своїх потреб.

Постійне тестування на користувачах є ключовим для виявлення та вирішення проблем із доступністю, які можуть виникнути під час використання мультимодальних систем. Надаючи пріоритет доступності під час проектування та реалізації мультимодальних взаємодій, розробники мають можливість створити інклюзивне середовище, що розширює можливості та відповідає різноманітним потребам користувачів, у тому числі людей з обмеженими можливостями.

Отже, у розділі розглянуто основи взаємодії з інтерфейсом комп'ютера як приклад мультимодальної системи. Люди використовують п'ять основних органів чуття для взаємодії з навколишнім середовищем. Мультимодальні системи поєднують відповідні модальності для покращення користувацького досвіду, використовуючи методи розпізнавання.

З розвитком технологій зростає кількість каналів комунікації з комп'ютером. Наприклад, Neuralink продемонстрував управління комп'ютером силою думки за допомогою мозкового імпланту, а MouthPad від Augmental дозволяє керувати комп’ютером за допомогою рухів язика та ковтків, що доречно для людей з порушеннями моторики.

Мультимодальні системи мають такі характеристики: обробка кількох модальностей одночасно (паралельність), надання надлишкової інформації для підвищення надійності (надмірність), доповнення різних типів інформації для кращого розуміння (взаємодоповнюваність), адаптація до контексту і потреб користувача (адаптивність) та навчання на основі взаємодії (гнучкість).

Теорія когнітивного навантаження підкреслює необхідність управління інформаційним навантаженням для полегшення навчання та обробки даних людиною. Мультимодальні системи зменшують зовнішнє навантаження через представлення інформації у кількох модальностях.

Інтеграція модальностей включає злиття даних, сенсорів та алгоритмічну інтеграцію, що забезпечують безперебійну взаємодію.

Мультимодальність також сприяє доступності, дозволяючи користувачам з різними потребами взаємодіяти з системою. Наприклад, поєднання тактильного зворотного зв'язку та звукових підказок покращує навігацію для людей з вадами зору, а розпізнавання голосу допомагають користувачам з порушеннями

моторики. Постійне тестування забезпечує інклюзивність та розширює можливості всіх користувачів.

РОЗДІЛ 2: ДИЗАЙН МУЛЬТИМОДАЛЬНИХ СИСТЕМ

2.1 Принципи дизайну для мультимодальних систем

2.1.1 Дизайн фреймворки

Коли йдеться про проектування мультимодальних систем, слід розрізняти структуровану методологію інтеграції багатьох модальностей. Існують певні теорії, яких дотримуються під час проектування мультимодальних систем, серед них дві найпопулярніші це теорія КПНЕ і W3C Multimodal Interaction Framework.

Структура КПНЕ (комплементарність, присвоєння, надмірність та еквівалентність), запропонована Кутазом та ін., наголошує на можливості інтеграції модальностей на основі їхньої взаємодоповнюваності та надмірності [14]. Модель допомагає розробникам вирішити, як модальності повинні доповнювати одна одну, де вони можуть надмірно накладатися, як їх можна еквівалентно використовувати та які функції слід присвоїти кожній модальності.

Консорціум Всесвітньої Веб Павутини розробив структуру *W3C Multimodal Interaction Framework,* щоб стандартизувати розробку мультимодальних програм [15]. Теорія складається з кількох компонентів, зокрема:

- *Компоненти модальності (введення та виведення)*, що відповідають за обробку модальностей на вхід та вихід, наприклад розпізнавання мовлення, жестів і перетворення тексту в звук.
- *Менеджер взаємодії* допомагає в інтеграції компонентів на основі модальності разом із логікою програми для координації потоку інформації.

- *Компонент сеансу* призначений для керування робочими процесами, пов’язаними з обробкою сеансу та стану, щоб допомогти мультимодальним програмам підтримувати роботу на різних пристроях та у різних режимах.
- *Компонент системи та середовища* використовується для сповіщення конкретної системи про параметри пристрою і уподобання користувача, щоб адаптувати інтерфейс до відповідних запитів.

Важливо пам’ятати, що обидва підходи корисні для розробників мультимодальних систем. Враховуючи такі фактори, як взаємодоповнюваність модальностей, когнітивне навантаження та використання стандартизованої архітектури, фреймворки допомагають забезпечити інтуїтивно зрозумілі мультимодальні інтерфейси та взаємодію з користувачем.

2.1.2 Дизайн, орієнтований на користувача та спільний дизайн

Ефективність мультимодальних систем багато в чому залежить від того, наскільки ці системи задовольняють реальні потреби людей. Для досягнення даної мети важливо звертати увагу на принципи *дизайну, орієнтованого на користувача.* Цей підхід до проектування означає, що користувачі залучені протягом усього процесу проектування, щоб переконатися, що система є зручною та зрозумілою [13].

У мультимодальних системах ця теорія включає кілька ключових практик:

- *Дослідження користувачів:* для визначення цілей, завдань і проблем цільових користувачів проводяться інтерв'ю, анкетування та спостереження.
- *Тестування зручності використання:* перевірка простоти використання системи шляхом спостереження за користувачами, які намагаються використовувати прототипи або ранні моделі системи.
- *Інтеграція зворотного зв’язку:* постійний збір та інтеграція відгуків користувачів у проект для вдосконалення системи.

Спільне проектування доповнює підхід, орієнтований на користувача, залучаючи зацікавлених сторін, наприклад самих кінцевих користувачів до процесу проектування [16]. Ключовий аспект спільного дизайну – це надання користувачам права голосу у прийнятті дизайн-рішень, та постійне залучення користувачів у інших аспектах. Таким чином, під час розробки системи буде охоплено потреби та уподобання користувачів.

2.1.3 Адаптивність та персоналізація

Адаптивність і персоналізація є надважливими при проектуванні мультимодальних систем, щоб гарантувати, що програма задовольняє потреби користувачів у різних контекстах і середовищах.

Адаптивні мультимодальні системи динамічно коригують свою поведінку на основі взаємодії користувача, його уподобань і умов середовища. Вони реагують на поточне завдання, навколишнє середовище та когнітивне навантаження користувача. Приклади адаптивних мультимодальних систем включають:

- *Контекстно-залежне обчислення:* мультимодальні системи, розроблені з використанням контекстно-залежних технологій, можуть виявляти зміни в середовищі користувача та відповідним чином змінювати свою поведінку. Наприклад, система може перемикатися з візуальних на звукові методи виведення, коли виявляє, що користувач керує автомобілем.
- *Динамічні налаштування інтерфейсу:* системи можуть налаштовувати елементи інтерфейсу на основі вподобань або потреб користувача, наприклад змінювати розмір тексту, модальність або її складність на основі попередньої історії взаємодії користувача або налаштувань.

Персоналізація в мультимодальних системах передбачає попередню конфігурацію системи для задоволення індивідуальних потреб користувача,

надання користувачам можливості встановлювати свої налаштування стосовно того, як подається інформація або як виконуються команди. Створення детальних профілів користувачів на основі попередніх взаємодій допомагає передбачати наступні уподобання користувачів. Алгоритми машинного навчання можуть аналізувати поведінку користувача з часом, щоб передбачити потреби та автоматизувати персоналізацію без введення даних про себе користувачем.

2.2 Принципи людино-машинної взаємодії в мультимодальних системах

2.2.1 Практики людино-машинної взаємодії

Практики людино-машинної взаємодії гарантують, що системи є доступними та надійними при інтеграції кількох модальностей взаємодії. Найпопулярнішими з них є узгодженість, зворотний зв’язок, запобігання помилок і відновлення та контроль над взаємодією, що необхідні для ефективної мультимодальної взаємодії.

Узгодженість є фундаментом для проектування мультимодальної системи для зменшення плутанини та підвищення зручності використання. [17] Ця практика передбачає збереження одноманітності візуальних елементів та патернів взаємодії у різних модальностях. Оскільки мультимодальні системи часто охоплюють кілька пристроїв, підтримання узгодженості дизайну та функціональності на різних платформах також покращує взаємодію з користувачем.

Зворотний зв'язок допомагає підтверджувати дії користувача та органічно оновлювати стан системи. Зворотний зв’язок, що відповідає конкретному типові модальності, має бути своєчасним та інформативним, аби уникнути будь-яких непорозумінь з боку користувача. Поєднання слухових сигналів із візуальним зворотним зв’язком може покращити реакцію системи, особливо у

складних завданнях, [18] а значить покращує досвід користування. Зворотний зв’язок також має регулюватися відповідно до контексту використання, щоб уникнути когнітивного перевантаження користувача.

Запобігання помилкам і відновлення: добре спроектовані мультимодальні системи повинні не тільки мінімізувати ймовірність помилки користувача, але й забезпечувати надійні механізми для відновлення від помилок. Наприклад, додання обмежень та підтверджень, щоб запобігти помилкам, може допомогти користувачам впевненіше взаємодіяти із програмою. Або, ж як показують деякі дослідження, користувачі "природно" перемикають модальності для виправленні помилок [19], тому їм необхідно забезпечити цю можливість. Система також повинна пропонувати зрозумілі шляхи відновлення після помилки, включаючи детальні повідомлення для відповідних модальностей та параметри скасування [20].

Крім вище перелічених практик, користувачі повинні відчувати *контроль над взаємодією*. Цього можна досягти за допомогою настроюваних інтерфейсів і можливості легко скасовувати операції або переходити між різними станами системи. Необхідно дозволяти користувачам вибирати бажані способи взаємодії та перемикатися між ними за потреби, те ж саме і стосується і перемикання між модальностями.

2.2.2 Практики ефективного навчання

Як і будь-якої технології, для мультимодальних систем є важливим забезпечення швидкого навчання як правильно взаємодіяти з системою.

Однією з практик є *скаффолдинг і прогресивне розкриття,* коли знайомство користувачів із функціями системи відбувається поступово, щоб запобігти перевантаженню інформацією. Дослідження Бакки-Акости та ін. висвітлили, що природно вбудовані у середовище віртуальної реальності

скаффолди покращують навчання за принципом просторової суміжності [21]. Подібної користі в інших мультимодальних системах можна досягти за допомогою методів каркасу, які забезпечують тимчасові структури підтримки для користувачів, наприклад контрольовані завдання при перших взаємодіях з програмою, які поступово видаляються в міру набуття знань. Контекстна довідка також є прикладом доречного навчання; можна, наприклад використовувати для пояснення підказок для жестів або голосових команд відповідно до контексту (рис. 2.1).

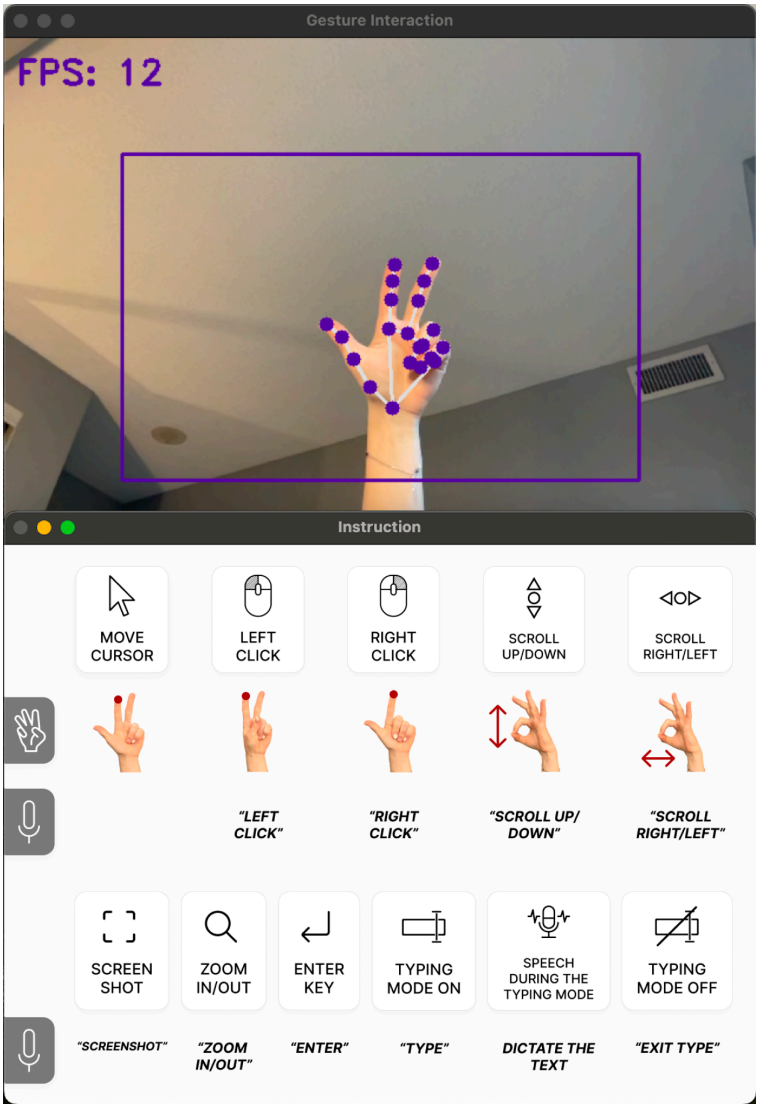


Рис 2.1 Виведення підказок у мультимодальній програмі для жестової та голосової взаємодії

Навчання користувачів за допомогою симуляції є схожою практикою, втіленою за допомогою симуляцій або інтерактивних навчальних сеансів, які імітують випадки реального використання, з метою навчання користувачів навігації по системі. Більш того, як показало дослідження Росс та ін., навчання в симуляції може адаптовуватись до досвіду користувача через використані сенсори, наприклад біо-датчики, як вказано у статті, де біосигнали ЕКГ класифікують досвід для адаптивного моделювання [22]. Важливим компонентом навчання за допомогою симуляції є запровадження своєчасного зворотного зв'язку під час навчання, щоб дати змогу користувачу виправити помилки в режимі реального часу.

Використання знайомих метафор дизайну (наприклад, розведення пальців для збільшення і зведення для зменшення) та проектування взаємодії на основі існуючого досвіду користувача, щоб скоротити криву навчання запроваджує таку практику як *узгодженість і звичність***,** про яку вже було згадано раніше. Узгодженість модальностей також може допомогти користувачам передбачити моделі взаємодії. Важливо пам'ятати, що командні структури в різних модальностях, наприклад в сенсорному та голосовому інтерфейсах, мають бути узгодженими.

2.3 Складності дизайну мультимодальної системи

2.3.1 Технічні складності

Розробка мультимодальних систем передбачає ряд технічних проблем, що випливають із складності інтеграції кількох модальностей. Серед них проблеми синхронізації, складність об'єднання даних, високе обчислювальне навантаження.

Однією з головних технічних проблем у мультимодальних системах є *синхронізація* різних модальностей введення та виведення. Щоб забезпечити

роботу аудіо, візуальних та тактильних компонентів в унісон і загалом цілісну взаємодію з користувачем, необхідних точний таймінг та координація. Для цього модальності мають бути синхронізовані в режимі реального часу, щоб уникнути плутанини у користувачів. Наприклад, для ефективного виконання команди потрібно одночасно обробляти введення голосом і жестами [9].

Об'єднання даних у мультимодальних системах включає об'єднання інформації отриманої з різних джерел і модальностей для створення уніфікованої матриці рішень. Цей процес складний через різнорідність типів даних і різний час обробки [23]. Синхронізація, що була вже згадана раніше, є однією зі складностей об'єднання даних через керування різницею в часі між модальностями. Наприклад, потоки аудіо- та візуальних даних можуть бути неправильно узгоджені між собою, що може спричинити неправильне тлумачення намірів користувача системою. Також, різні сенсори або методи введення надають дані в різних форматах, роздільній здатності та деталізації, що спричиняє неоднорідність даних. Наприклад, поєднання візуальних даних високої роздільної здатності з низькоякісним аудіо вимагає алгоритмів, здатних обробляти значні розбіжності в якості даних. Семантична інтеграція даних для надання певних висновків системою передбачає розуміння семантичних зв'язків між різними модальностями. Наприклад, слова, сказані в аудіопотоці, можливо, потрібно буде зіставити з рухом губ у відео, щоб перевірити точність.

Мультимодальні системи часто вимагають значних обчислювальних ресурсів для обробки та аналізу одночасних вхідних даних від кількох модальностей, що може збільшувати *обчислювальне навантаження.* Ефективне розподілення обчислювальними ресурсами дуже важливе, особливо для додатків, що працюють на пристроях з обмеженою ємністю, по типу мобільних телефонів. Динамічний розподіл ресурсів передбачає розподіл

обчислювальних ресурсів у режимі реального часу на основі поточних потреб програми. Методи балансування навантаження та планування ресурсів особливо в середовищах, де вимоги до ресурсів постійно змінюються можуть допомогти в цьому. Апаратне прискорення, що заключається у використанні спеціалізованих апаратних компонентів, таких як GPU для завдань обробки зображень і відео або DSP для обробки аудіо, може значно зменшити навантаження на головний процесор, підвищуючи загальну продуктивність системи [24]. Використання методів оптимізації допоможе керувати обчислювальним навантаженням. Зокрема за допомогою скорочення даних можна зменшити навантаження на обробку шляхом раннього видалення нерелевантної або надлишкової інформації. Крім того, завдання, що можуть бути оброблені паралельно, такі як розпізнавання зображень і мови, виконуються ефективніше за використання багатоядерних процесорів або розподілених систем [25]. Хмарні обчислення ще більше полегшують навантаження на локальний пристрій, перекладаючи важкі завдання на динамічно масштабовані хмарні ресурси [26]. Периферійні обчислення обробляють дані ближче до їх джерела, наприклад, на локальних серверах або пристроях Інтернету речей, зменшуючи затримку, важливу для додатків у реальному часі, підвищуючи ефективність системи та швидкість реакції [26].

2.3.2 Складності зручності користування

Інтеграція багатьох модальностей взаємодії в мультимодальних системах може створювати проблеми зручності використання.

Мультимодальні системи, пропонуючи кілька модальностей та режимів взаємодії, можуть призвести до *когнітивного перевантаження*, якщо не спроектувати їх ретельно. Користувачам може бути важко обробити одночасне виведення від декількох джерел, якщо інформація представлена у незручний для сприйняття спосіб. Одним з рішень буде спрощення взаємодії користувача,

узгоджуючи її з когнітивними можливостями людини. Наприклад, можна обмежити кількість активних модальностей у будь-який момент часу та надати чіткі підказки щодо того, як перемикатися між модальностями за необхідності. Інтерфейси, які можуть адаптуватися до поточного когнітивного навантаження користувача, можливо, шляхом моніторингу відповідей користувача або фізіологічних показників, як уже було згадано у дослідженні Росс та ін. [22], щоб налаштувати складність представленої інформації також можуть допомогти вирішити проблему зручності користування.

Оскільки мультимодальні системи поєднують різні типи введення і виведення, *інтерфейс може стати заскладним*, потенційно призводячи до не дуже приємного досвіду користування. Використання принципів узгодженого дизайну інтерфейсу в усіх модальностях, скоротить час навчання та покращить адаптації і зручність користування. Варто перевірити, що кожна модальність використовується для завдань, для яких вона найбільше підходить. Наприклад, голосові команди можуть бути ідеальними для операцій без використання рук, тоді як візуальні введення можуть бути кращими для детальних завдань навігації.

Забезпечення того, щоб мультимодальні системи були *доступними* для користувачів із різними потребами, у тому числі з обмеженими можливостями, є серйозною проблемою. Застосування принципів універсального дизайну дозволяє користувачам з різними здібностями ефективно взаємодіяти з інтерфейсом та забезпечує доступ до інформації через різні сенсорні канали [27]. Надання користувачам опції для налаштування інтерфейсу відповідно до їхніх конкретних потреб, наприклад налаштування розміру тексту, колірних контрастів або уподобань модальності також допоможе полегшити проблему доступності.

Також, користувачі з обмеженим контактом із передовими технологіями можуть вважати мультимодальні системи особливо складними, що впливає на їхнє початкове впровадження та подальше використання.

Для подолання труднощів із зручністю використання необхідно проводити широке тестування із різними групами користувачів, щоб отримати уявлення про різний досвід і виявити проблеми в різних модальностях. А ітеративний процес проектування має включати результати тестування користувачів для вдосконалення інтерфейсу та модальностей взаємодії.

Отже, у розділі розглянуто основні аспекти дизайну мультимодальних систем. Фреймворки КПНЕ та W3C Multimodal Interaction Framework допомагають уникати надмірного накладання модальностей і адаптувати системи до різних потреб користувачів. Дизайн, орієнтований на користувача, забезпечує активну участь користувачів у процесі проектування, що підвищує доступність і зручність використання. Адаптивність і персоналізація дозволяють системам динамічно реагувати на запити користувачів і умови середовища.

Практики людино-машинної взаємодії, такі як узгодженість, зворотний зв’язок і запобігання помилок, важливі для створення мультимодальних систем. Узгодженість зменшує плутанину, зворотний зв’язок підтверджує дії користувача, а запобігання помилок підвищує надійність системи. Методики скаффолдингу та прогресивного розкриття допомагають користувачам швидко навчитися взаємодіяти з системою.

Розробка мультимодальних систем стикається з технічними проблемами, такими як синхронізація модальностей, складність об'єднання даних і управління обчислювальними навантаженнями. Синхронізація аудіо,

візуальних і тактильних компонентів потребує точного часу та координації. Ефективне управління ресурсами і використання хмарних обчислень допомагають справлятися з високими вимогами. Інтеграція багатьох модальностей може створювати й проблеми зручності використання, які вирішуються через тестування та узгоджений дизайн.

РОЗДІЛ 3: РЕАЛІЗАЦІЯ МУЛЬТИМОДАЛЬНОЇ ВЗАЄМОДІЇ

3.1 Вимоги та загальні відомості

Основною вимогою до програми є забезпечення користувачів можливості більш глибоко взаємодіяти з комп'ютерним інтерфейсом без необхідності кардинально змінювати його зовнішній вигляд. Замість цього покращення інтерактивності досягається шляхом інтеграції додаткових модальностей. Такий підхід робить програму універсальним інструментом, який може покращувати користувацький досвід незалежно від особливостей вже існуючого інтерфейсу.

У програмі реалізовано такі додаткові модальності як візуальна та звукова, як на введення так і на виведення. Для реалізації використано мову програмування Python. Ключовими задіяними бібліотеками є:

- OpenCV – це бібліотека функцій програмування, головним чином для комп’ютерного зору в реальному часі [28].
- MediaPipe – це платформа для побудови комунікаційної лінії машинного навчання для обробки даних, таких як відео, аудіо тощо [29].
- PyAutoGui – це кросплатформний модуль Python, призначений для програмного керування мишею та клавіатурою [30].
- playsound – кросплатформний однофункціональний модуль без залежностей для відтворення звуків [31].
- PyAudio – надає прив’язки Python для PortAudio v19, міжплатформенної бібліотеки аудіо введення/виведення [32].
- Vosk – офлайн-набір інструментів розпізнавання мовлення з відкритим кодом [33].

3.2 Модуль відслідковування рук HandModule

Для розпізнавання та відслідковування рук реалізовано допоміжний модуль HandModule, написаний за використання набору бібліотек MediaPipe, і містить клас MyHands.

У класі MyHands є 5 ключових функцій. Вони зосереджені на розпізнаванні точок-лендмарків (рис. 3.1) на долоні та прописанні умов відповідно до цього.

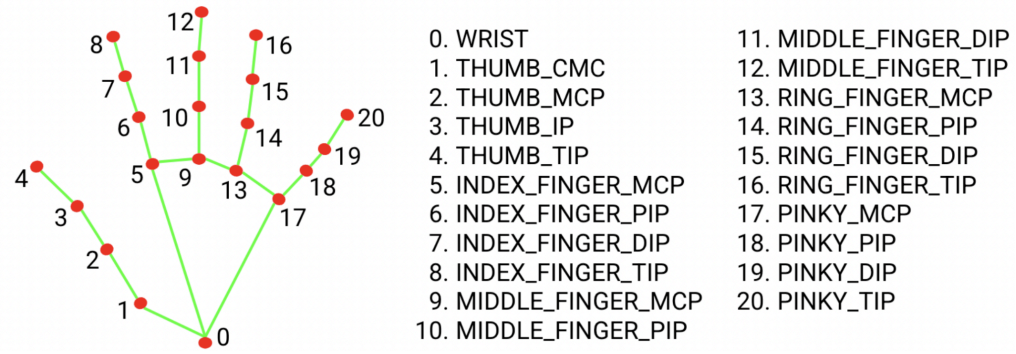


Рис 3.1 Лендмарки-точки долоні

- process_frame – обробляє отримане зображення з камери та повертає системні координати *x, y, z* кожної лендмарки долоні та “handedness”, тенденцію використовувати ту чи іншу руку.
- draw_hands – додає схематичні ліній та точки за замовченням поверх зображення рук та повертає оновлене зображення.
- get_landmarks_list – повертає координати *x, y, z* на основі піксельних значень кожної лендмарки долоні та додає кастомне зображення точок до кожної з лендмарки поверх зображення руки.
- landmarks_distance – знаходить відстань між двома заданими лендмарками.

- fingers_up – повертає масив із п’яти значень 0 або 1, відповідно до кожного з пальців. 1 є сигналом про те, що відповідний палець піднятий, 0 - не піднятий.

3.3 Виконання команд за допомогою жестів

Для зчитування жестів очевидно необхідною умовою є доступ до зображення з камери протягом усього часу роботи програми. Крім того, щоб користувачу було легше орієнтувався у системі, надається контекстна інструкція управління жестами та голосовим вводом, про який йтиме мова далі (рис. 3.2).

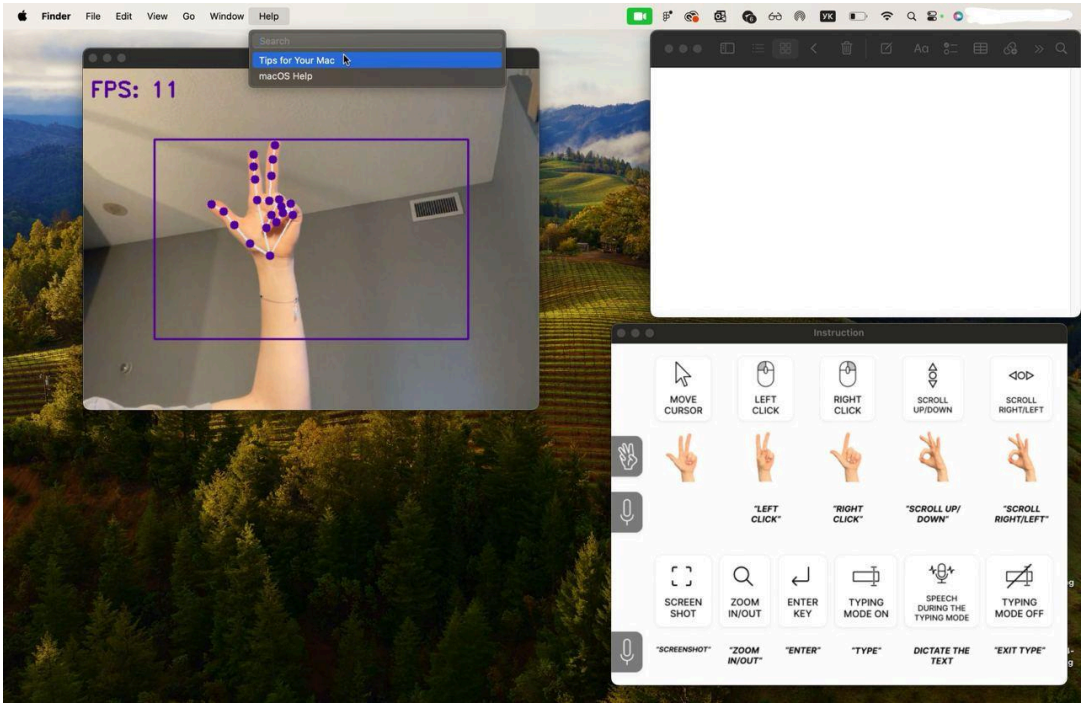


Рис 3.2 Інтерфейс програми з інструкцією

Для неперервності у циклі *while True:* першочергово відбувається зчитування картинки з камери (рис. 3.3, рис. 3.4)

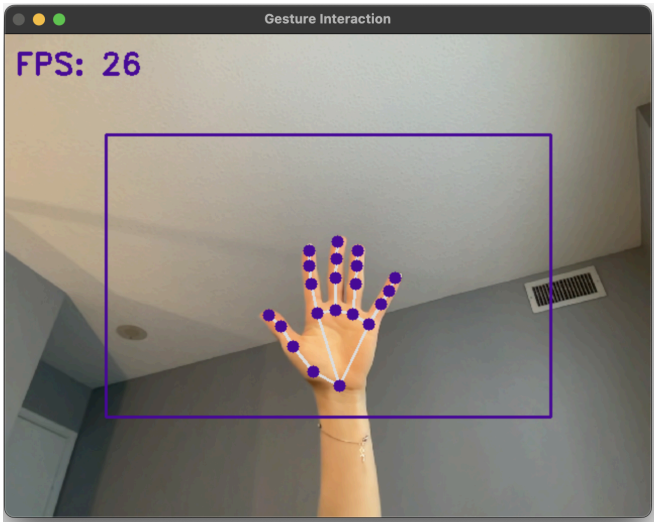


Рис 3.3 Зображення руки з камери

```
cap = cv2.VideoCapture(0)
while True:
    _, img = cap.read()
```

Рис 3.4 Безперервний захват кадру з веб-камери за допомогою OpenCV

Використовуючи функції зазначені в HandModule, а також бібліотеку PyAutoGui, можемо задати команди для управління курсором за допомогою відповідних жестів.

Додана нова функція ***move_cursor***, яка головним чином відповідає за рух курсора. Камера відстежує координати *x* та *y* вказівного пальця, та відповідно до руху пальця виконується рух курсора по екрану. Як можна побачити на рис. 3.3 та рис. 3.4, область керуванням курсора на зображенні з камери обмежена фіолетовим прямокутником, що пропорційно відповідає зменшеному розміру всього екрану інтерфейсу. Таким чином рух всередині прямокутника відповідатиме руху на екрані у відповідно пропорційних координатах.

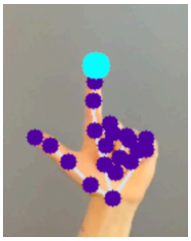
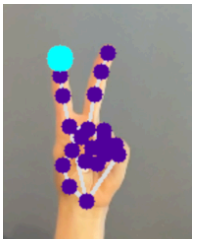
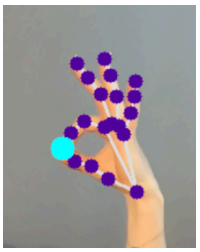
Цикл *while True,* згаданий раніше, розпочинає головну частину програми, в якій задаються умови для виконання команд за допомогою жестів. Конкретно, виклик тої чи іншої команди залежить від комбінації піднятих

пальців, що дозволяє визначити функція *fingers_up* з HandModule, та інших метрик, наприклад відстані між певними пальцями, що визначається за допомогою функції *landmarks_distance* (рис. 3.4)

```
# [SCROLL]: middle, ring, pinky are up
if fingers[2] and fingers[3] and fingers[4]:
    length_thumb_index, img, lineInfoTI = myhand.landmarks_distance( lmId1: 4,  lmId2: 8, img, lmList)

    if length_thumb_index < 13:
        cv2.circle(img,  center: (lineInfoTI[4],lineInfoTI[5]),  radius: 10,  color: (255, 255, 0), cv2.FILLED)
        if not was_pinching_v:
            initial_y_TI = lineInfoTI[5]
            was_pinching_v = True
        if not was_pinching_h:
            initial_x_TI = lineInfoTI[4]
            was_pinching_h = True

        x_TI = lineInfoTI[4]
        y_TI = lineInfoTI[5]

        # vertical scroll
        if initial_y_TI is not None:
            displacement_v = initial_y_TI - y_TI
            # Check if the displacement is at least 10 pixels
            if abs(displacement_v) >= 10:
                pyautogui.vscroll(-displacement_v)
                playsound(scrollSound, block=False)
                initial_y_TI = y_TI

        # horizontal scroll
        if initial_x_TI is not None:
            displacement_h = initial_x_TI - x_TI
            # Check if the displacement is at least 10 pixels
            if abs(displacement_h) >= 10:
                pyautogui.hscroll(-displacement_h)
                playsound(scrollSound, block=False)
                initial_x_TI = x_TI
    else:
        was_pinching_v = False
        was_pinching_h = False
```

Рисунок 3.4 Приклад реалізації команди для прокручування елементів інтерфейсу

У таблиці 3.1 наведено перелік команд та відповідні жести, які викликають кожну з них. Команди реалізовані за використання бібліотеки PyAutoGui.

Комбінація на введення		Жест, який розпізнає камера	PyAutoGui процедура	Результат
Підняті пальці	Інші метрики			
Великий + Вказівний + Середній			moveTo	Рух курсору за вказівним пальцем
Великий + Вказівний			click	Клік лівою кнопкою миші
Вказівний + Середній			rightClick	Клік правою кнопкою миші
Середній + Безіменний + Мізинець	Відстань між вказівним і великим пальцями менше 13 одиниць		vscroll hscroll	Горизонтальне та вертикальне прокручування в залежності від орієнтації руху руки – по вертикалі чи горизонталі

Таблиця 3.1 Перелік команд та відповідні жести, що їх викликають

Вибір жестів обгрунтований двома аспектами: інтуїтивність застосування, тобто чи має користувач асоціацію певної дії з відповідним жестом, та обмеженням розпізнавання схожих жестів.

Лімітація стосується неможливості приписання однакової комбінації піднятих пальців до різних жестів, при умові, що немає оцінки іншої метрики, наприклад відстані між певними пальцями. Важливим аспектом є також те, що результати вимірювання відстані між пальцями або показник того, чи піднятий певний палець мають бути однозначними, аби бути надійною умовою для диференціації команд. Це значить, що навіть невелика схожість команд за метриками може призвести до плутанини.

Інтуїтивність застосування зумовлена використанням найбільш природних жестів відповідно до попереднього досвіду користування інтерфейсів, особливо сенсорних; низькою витратою фізичних зусиль для жесту; а також урахуванням обмежень зазначених вище. До прикладу, жест піднятого вказівного пальця є найбільш інтуїтивно зрозумілим для переміщення курсора в інтерфейсі користувача завдяки імітації реальної поведінки вказівки напряму або місця. Також, як зазнає Гіл та ін., вказівний або великий палець – основний жест для виконання рухів та натиснення об’єктів на сенсорних екранах [34]. Дійсно, в програмі саме вказівний палець відповідає за надання координат для руху курсора. Разом з тим, дві інші дії, клік лівою та правою кнопкою миші, мають бути суміжними до жесту руху курсора (піднятого вказівного пальця), тобто необхідна мінімальна зміна положення вказівного пальця при кліку, аби не збити мішень кліку. Тому команда руху курсора активізується при одночасно піднятому великому, вказівному та середньому пальців, де згин середнього пальця сигналізувати про ЛКМ, а великого – ПКМ.

3.4 Розпізнавання введення голосом

Синхронно з введенням команд за допомогою жестів, або ж традиційно за використання трекпаду або миші та клавіатури, реалізовано розпізнавання голосу. PyAudio використана для налаштування та керування аудіо потоками, а

KaldiRecognizer (VOSK API) застосовано у ролі бібліотеки для аналізу мовлення, моделювання, декодування та інших завдань, пов’язаних із автоматичним розпізнаванням мовлення.

Функція speech_recognition() постійно прослуховує аудіопотік. Коли аудіодані надходять, вони надсилаються до розпізнавача мовлення. Коли розпізнавач (рис. 3.5) виявляє повну фразу з поточного аудіопотоку, він перевіряє, чи відповідає ця фраза певним командам, таким як «клацнути лівою кнопкою миші», «прокрутити вгору» або «збільшити масштаб».

```
def speech_recognition():
    model = Model("../vosk-model-small-en-us-0.15")

    # Stream to read audio data
    p = pyaudio.PyAudio()
    stream = p.open(format=pyaudio.paInt16, channels=1, rate=16000, input=True, frames_per_buffer=8000)
    stream.start_stream()

    recognizer = KaldiRecognizer( *args: model, 16000)

    while True:
        data = stream.read( num_frames: 4000, exception_on_overflow=False)
        if recognizer.AcceptWaveform(data):
            result = json.loads(recognizer.Result())
            print(result)
            recognized_text = result.get('text', '').lower()
```

Рис 3.5 Ініціалізація розпізнавача

Якщо збіг знайдено, програма виконує відповідну команду за допомогою PyAutoGui та інших функцій (рис. 3.6)

```
if recognized_text == 'left click':
    print("Performing a left click")
    pyautogui.click()
```

Рис 3.6 Приклад умови розпізнавання фрази та ініціації команди

У таблиці 3.2 наведено команди для голосового розпізнавання та команди, що надходять у відповідь від системи.

Команда для розпізнавання	**Команда, що виконується**
left click	Клік лівою кнопкою миші
right click	Клік правою кнопкою миші
scroll up	Прокручування вгору на 80 одиниць
scroll down	Прокручування вниз на 80 одиниць
scroll left	Прокручування вліво на 80 одиниць
scroll right	Прокручування вправо на 80 одиниць
screenshot	Скріншот та збереження файлу скріншота
zoom in	Збільшення масштабу
zoom out	Зменшення масштабу
enter	Натиснення клавіші “enter”
type	Увімкнення режиму друкування, який дозволяє надиктовувати текст для друку
будь-яке мовлення, крім фраз, що співзвучні з іншими голосовими командами, під час режиму друкування тексту	Друк надиктованого тексту
exit type	Вимкнення режиму друкування

Таблиця 3.2 Команда для голосового розпізнавання та відповідний зворотний зв’язок

Використання потоків (threads) дозволяє програмі керувати одночасними операціями. Один потік обробляє розпізнавання мовлення, а інший керує відстеженням жестів і оновленнями GUI (рис. 3.7). Це запобігає зависанню програми, якщо один із процесів блокується або виконується довго. Розподіляючи завдання між декількома потоками, програма може виконувати

інтенсивні операції, такі як обробка мови та зображень у реальному часі, без зупинки основного інтерфейсу користувача.

```
thread = threading.Thread(target=speech_recognition)
thread.daemon = True
thread.start()
```

Рис 3.7 Використання потоків для оптимізації програми

3.5 Додавання звукового зворотного зв'язку

Звукова модальність на виведення реалізована за допомогою бібліотеки playsound (рис. 3.8). Одночасно з виконанням жестових команд або введення голосових команд надається звуковий зворотний зв'язок для деяких команд. Серед них клік лівою кнопкою миші, клік правою кнопкою миші, прокручування та скріншот.

```
playsound(leftClickSound, block=False)
```

Рис 3.8 Приклад реалізації звукового зворотного зв'язку у відповідь на клік ЛКМ

Використання різних звуків для різних команд в інтерфейсі сприяє когнітивному картографуванню користувача, що допомагає швидко пов'язувати певні дії зі звуковим відгуком, зменшуючи помилки [35]. Такий підхід також збільшує доступність для користувачів із вадами зору, пропонуючи чіткі звукові підказки, які окреслюють функціональні можливості. Крім того, звукові підказки також можуть допомогти навчитися та запам'ятати, як користуватися системою швидше, що є особливо актуальним у поєднанні роботи з жестами, як видом нетрадиційної взаємодії, що потребує освоєння програми [35].

3.6 Інші теоретичні можливості контролю інтерфейсу

Керування жестами, як приклад візуальної модальності на введення, дійсно розширює можливості користувача, проте все ще є доволі фізично затратною взаємодією. Спостерігаючи за тенденціями розвитку VR та AR та прагненням використання найбільш природних рухів для взаємодії, як наприклад робить Apple у їх продукті VisionPro, хедсеті віртуальної реальності, що дозволяє контролювати інтерфейс за допомогою погляду та жестів [36], звичайна взаємодія з комп’ютером також стає цікавим місцем для подібних інновацій.

Бібліотека MediaPipe, використана раніше для розпізнавання жестів, також може розпізнавати лендмарки обличчя (рис. 3.9), включно з очима, а отже надає можливість запрограмувати керування інтерфейсу поглядом.

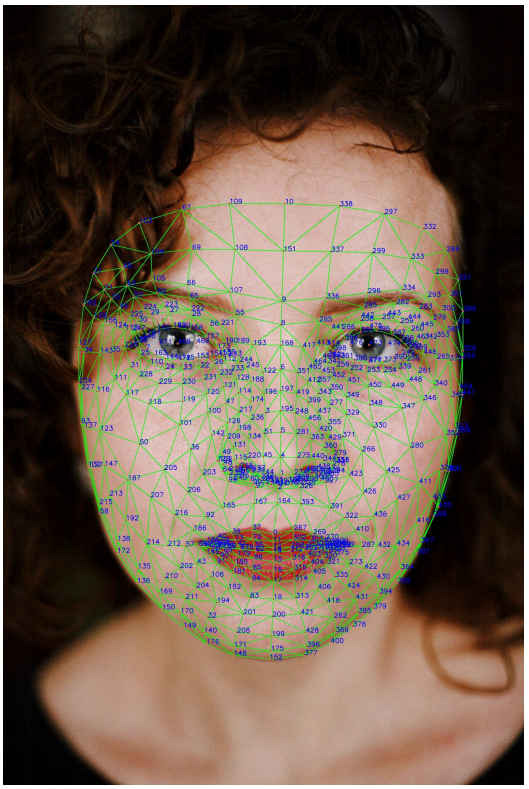


Рис 3.9 Лендмарки обличчя

Подібно зчитуванню лендмарки вказівного пальця для руху курсору, програма може зчитувати лендмарки правого ока та використовує згадану раніше функцію *move_cursor* для керуванням курсору. Ліве око відповідає за клік ЛКМ. Для полегшення навігації аналогічно модулю керування жестами, для керуванням погляду також є “контейнер”, який є областю взаємодії і менше за вікно зображення з камери. Такий підхід мінімізує рухи користувача, адже контейнер зменшує допустиму область руху курсору (рис. 3.10).

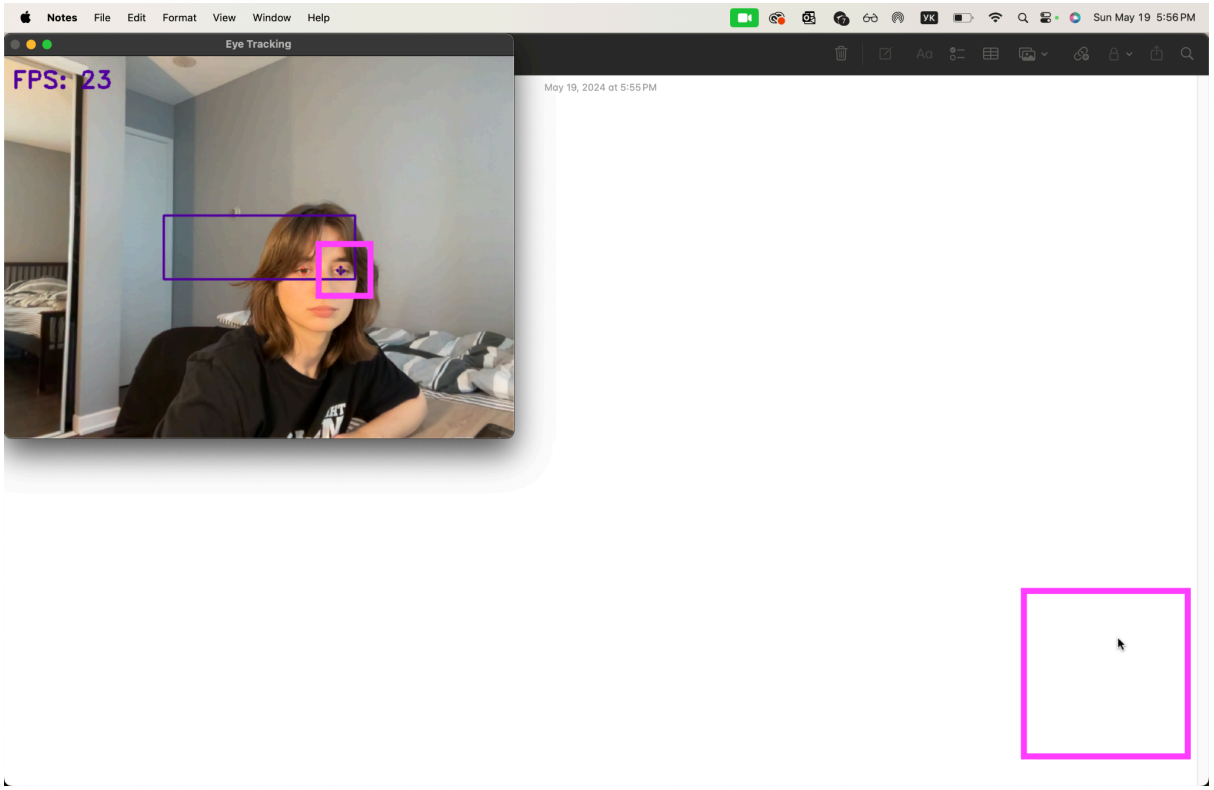


Рис 3.10 Відповідність положенню ока у контейнері положенню курсора на екрані

На жаль, наразі не існує точних технологій для керування курсором лише за допомогою погляду при статичному положенні голови або мінімальних рухах, використовуючи лише веб-камеру персонального комп'ютера. Відсутність додаткових засобів, таких як спеціалізовані камери та сенсори для

зчитування міміки, обмежує точність безконтактної взаємодії при мінімальних рухах голови.

3.7 Результати, недоліки та перспективи

Підсумовуючи, програма є прикладом розширення можливостей взаємодії користувача з комп’ютером. А саме, вона задіює візуальну модальність на вхід у вигляді розпізнавання жестів або, альтернативно, погляду людини і звукову у вигляді розпізнавання мовлення на додачу до традиційної тактильної модальності за використання миші та клавіатури. У свою чергу на дії користувача програма надає звуковий зворотний зв’язок, доповнюючи звичний візуальний у вигляді зміни елементів в інтерфейсі та тактильний.

Програма, модальності якої поєднані за допомогою алгоритмічної інтеграції різних моделей, суттєво впливає на когнітивне навантаження користувачів, об’єднуючи різні способи взаємодії та надаючи двомодальний зворотний зв’язок. Хоча вона може збільшити внутрішнє навантаження через додаткову складність навчання використанню нових модальностей, програма разом з тим потенційно зменшує зовнішнє навантаження, пропонуючи більш природні методи взаємодії, аніж традиційні клавіатура та мишка. Найважливіше те, що програма може підвищити доречне навантаження, сприяючи глибшому залученню користувача, що зрештою призводить до більш інтуїтивно зрозумілого досвіду користувача. Цей вплив підкреслює баланс, необхідний при проектуванні мультимодальних систем, щоб максимізувати зручність використання та ефективно керувати когнітивними вимогами.

Одним із недоліків програми є точність управління інтерфейсом, що охоплює керування за допомогою жестів та погляду, а також точне розпізнавання команд. Використання веб-камери як пристрою для зчитування візуальної інформації обмежує точність рухів курсору через технічні

обмеження камери. Помилки в звуковій модальності можуть виникати через недоліки в моделі розпізнавання та через зовнішні шуми, що потрапляють у аудіо потік. Важливо покращувати аспекти безконтактної взаємодії, мінімізуючи фізичні зусилля і рухи, необхідні для виконання команд, щоб забезпечити максимальну доступність інтерфейсу, особливо для користувачів з порушеннями моторних функцій.

Розширення можливостей взаємодії з комп'ютером через розвиток і впровадження мультимодальних систем сприятиме створенню більш інклюзивного цифрового середовища, що дозволяє користувачам виконувати команди з мінімальними фізичними зусиллями, тим самим сприяючи їхньому комфорту.

ВИСНОВКИ ПО РОБОТІ

У роботі було розглянуто основи взаємодії з інтерфейсом комп'ютера через мультимодальні системи, які використовують п'ять основних органів чуття для покращення користувацького досвіду. Інноваційні приклади, такі як мозковий імплантат Neuralink та MouthPad від Augmental, демонструють можливості управління комп'ютером через різні модальності, що особливо корисно для людей з обмеженими можливостями. Мультимодальні системи мають важливі характеристики, такі як паралельність, надмірність, взаємодоповнюваність, адаптивність та гнучкість, що дозволяють покращувати взаємодію користувача з системою. Теорія когнітивного навантаження підкреслює важливість управління інформаційним навантаженням, щоб полегшити навчання та обробку даних. Інтеграція модальностей, злиття даних та алгоритмічна інтеграція є критичними для забезпечення безперебійної взаємодії.

Основні аспекти дизайну мультимодальних систем включають використання фреймворків КПНЕ та W3C Multimodal Interaction Framework, які допомагають адаптувати системи до різних потреб користувачів. Дизайн, орієнтований на користувача, забезпечує активну участь користувачів у процесі проектування, що підвищує доступність і зручність використання. Адаптивність і персоналізація дозволяють системам динамічно реагувати на запити користувачів і умови середовища. Використання практик людино-машинної взаємодії, такі як узгодженість, зворотний зв’язок і запобігання помилок, важливі для створення мультимодальних систем, адже вони покращують досвід користування.

Програма є прикладом розширення можливостей взаємодії користувача з комп'ютером через візуальну модальність (розпізнавання жестів або погляду) та звукову модальність (розпізнавання мовлення) на додачу до традиційної

тактильної модальності (мишки та клавіатури). Звуковий зворотний зв'язок доповнює візуальний і тактильний, зменшуючи зовнішнє когнітивне навантаження та сприяючи глибшому залученню користувача через доречне навантаження, що зрештою призводить до більш інтуїтивного досвіду користувача. Однак точність управління інтерфейсом та розпізнавання команд потребують подальшого вдосконалення, особливо для користувачів з порушеннями моторних функцій. Таким чином, розширення можливостей взаємодії з комп'ютером через розвиток мультимодальних систем сприятиме створенню більш інклюзивного цифрового середовища.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Turk, M. (2014). Multimodal interaction: A review. *Pattern Recognition Letters*, *36*, 189–195. https://doi.org/10.1016/j.patrec.2013.07.003

2. *X.com*. (n.d.). X (Formerly Twitter). https://twitter.com/neuralink/status/1770563939413496146?ref_src=twsrc%5Etfw%7Ctwcamp%5Etweetembed%7Ctwterm%5E1770565942168420750%7Ctwgr%5Eed3593fb7fa9133c7ae83369df79373237ca448b%7Ctwcon%5Es3_&ref_url=https%3A%2F%2Fm.economictimes.com%2Ftech%2Ftechnology%2Fdemonstrating-telepathy-neuralinks-first-brain-chip-patient-plays-chess-with-his-mind%2Farticleshow%2F108669245.cms

3. *Augmental*. (n.d.). Home. https://www.augmental.tech/

4. Jaimes, A., & Sebe, N. (2007). Multimodal human–computer interaction: A survey. *Computer Vision and Image Understanding*, *108*(1–2), 116–134. https://doi.org/10.1016/j.cviu.2006.10.019

5. Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. *Cognitive Science*, *12*(2), 257–285. https://doi.org/10.1207/s15516709cog1202_4

6. Shaffer, E. L. (2022). Cognitive Load Theory and Instructional Message Design. In M. Ramlatchan & C. Kohler (Eds.), Instructional Message Design: Theory, Research, and Practice (Vol. 2). Kindle Direct Publishing.

7. De Jong, T. (2009). Cognitive load theory, educational research, and instructional design: some food for thought. *Instructional Science*, *38*(2), 105–134. https://doi.org/10.1007/s11251-009-9110-0

8. Mayer, R. E., & Moreno, R. (2003). Nine ways to Reduce Cognitive Load in Multimedia Learning. In University of California, Santa Barbara &

University of New Mexico, *EDUCATIONAL PSYCHOLOGIST* (Vols. 38–1, pp. 43–52). Lawrence Erlbaum Associates, Inc. https://www.uky.edu/~gmswan3/544/9_ways_to_reduce_CL.pdf

9. Multimodal interfaces. (2008). In *Springer eBooks* (pp. 651–652). https://doi.org/10.1007/978-0-387-78414-4_159

10. Oviatt, S., Center for Human-Computer Communication, Computer Science Dept., & Oregon Graduate Institute of Science & Technology. (n.d.). Multimodal interfaces. In *Handbook of Human-Computer Interaction* (2002nd ed.). Lawrence Erlbaum. https://www.cogsci.msu.edu/DSS/2004-2005/Oviatt/Multimodel%20Interfaces.pdf

11. Paratore, M. T., & Leporini, B. (2023). Exploiting the haptic and audio channels to improve orientation and mobility apps for the visually impaired. *Universal Access in the Information Society*. https://doi.org/10.1007/s10209-023-00973-4

12. Vatavu, R., & Ungurean, O. (2022). Understanding Gesture Input Articulation with Upper-Body Wearables for Users with Upper-Body Motor Impairments. *CHI Conference on Human Factors in Computing Systems*. https://doi.org/10.1145/3491102.3501964

13. Shneiderman, B., & Plaisant, C. (2005). *Designing the user interface: Strategies for Effective Human-computer Interaction*. Allyn & Bacon. https://thuvienso.hcmute.edu.vn/tailieu/2013/20130505/huanltgc00061/textbook_ben_shneiderman_designing_the_user_interface_772.pdf

14. Nigay, L., & Coutaz, J. (2000). Multifeature Systems: the CARE properties and their impact on software design. *ResearchGate*. https://www.researchgate.net/publication/2468437_Multifeature_Systems_The_CARE_Properties_and_Their_Impact_on_Software_Design

15. *W3C Multimodal Interaction Framework*. (n.d.). https://www.w3.org/TR/mmi-framework/

16. Participatory design. (2017). In *CRC Press eBooks*. https://doi.org/10.1201/9780203744338

17. Principle of consistency and standards in user interface design. *The Interaction Design Foundation*. https://www.interaction-design.org/literature/article/principle-of-consistency-and-standards-in-user-interface-design

18. Chauhan, D. S., Chauhan, Akhtar, M. S., Akhtar, Ekbal, A., Bhattacharyya, P., Bhattacharyya, & Indian Institute of Technology Patna. (n.d.). Context-aware interactive attention for multi-modal sentiment and emotion analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing* (pp. 5647–5657). https://aclanthology.org/D19-1566.pdf

19. Cohen, P. R., Johnston, M., McGee, D., Oviatt, S. L., Clow, J., & Smith, I. (1998). The efficiency of multimodal interaction: a case study. *ISCAArchive*. https://doi.org/10.21437/icslp.1998-260

20. Suhm, B., BBN Technologies, Myers, B., Human-Computer Interaction Institute, Waibel, A., & Interactive Systems Laboratories. (2000). *Multimodal error correction for speech user interfaces*. https://www.cs.cmu.edu/~cpof/papers/suhm_tochi.pdf

21. Bacca-Acosta, J., Tejada, J., Fabregat, R., Kinshuk, N., & Guevara, J. (2021). Scaffolding in immersive virtual reality environments for learning English: an eye tracking study. *Educational Technology Research and Development*, *70*(1), 339–362. https://doi.org/10.1007/s11423-021-10068-7

22. Ross, K., Sarkar, P., Rodenburg, D., Ruberto, A. J., Hungler, P., Szulewski, A., Howes, D., & Etemad, A. (2019). Toward dynamically adaptive simulation: multimodal classification of user expertise using wearable devices. *Sensors*, *19*(19), 4270. https://doi.org/10.3390/s19194270

23. Wu, S. (2012). Data fusion in information retrieval. In *Adaptation, learning, and optimization*. https://doi.org/10.1007/978-3-642-28866-1

24. Merelli, I., Liò, P., Kotenko, I., & D'Agostino, D. (2020). Latest advances in parallel, distributed, and network‐based processing. *Concurrency and Computation*, *32*(10). https://doi.org/10.1002/cpe.5683

25. Ayed, F., & Hayou, S. (2023). Data pruning and neural scaling laws: fundamental limitations of score-based algorithms. *Transactions on Machine Learning Research*. https://openreview.net/pdf/a2433de6bb53464732f7a73cc1ea19c7a92f729f.pdf

26. Yang, S., Xie, Z., Peng, H., Xu, M., Sun, M., & Li, P. (2022, May 19). *Dataset Pruning: Reducing Training Data by Examining Generalization Influence*. arXiv.org. https://arxiv.org/abs/2205.09329

27. Wentzel, J., Velleman, E., & Van Der Geest, T. (2016). Developing Accessibility design Guidelines for wearables: Accessibility Standards for Multimodal wearable devices. In *Lecture notes in computer science* (pp. 109–119). https://doi.org/10.1007/978-3-319-40250-5_11

28. Pulli, K., Baksheev, A., Kornyakov, K., & Eruhimov, V. (2012). Realtime Computer Vision with OpenCV. *ACM Queue*, *10*(4), 40–56. https://doi.org/10.1145/2181796.2206309

29. *MediaPipe Solutions guide*. (n.d.). Google for Developers. https://ai.google.dev/edge/mediapipe/solutions/guide

30. *Welcome to PyAutoGUI's documentation! — PyAutoGUI documentation*. (n.d.). https://pyautogui.readthedocs.io/en/latest/

31. *playsound*. (2021, July 24). PyPI. https://pypi.org/project/playsound/

32. *PyAudio*. (2023, November 7). PyPI. https://pypi.org/project/PyAudio/

33. Alphacep. (n.d.). *GitHub - alphacep/vosk-api: Offline speech recognition API for Android, iOS, Raspberry Pi and servers with Python, Java, C# and Node*. GitHub. https://github.com/alphacep/vosk-api

34. Gil, H., Kim, H., & Oakley, I. (2018). Fingers and angles. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, *2*(4), 1–21. https://doi.org/10.1145/3287042

35. Gaver, William. (1989). The SonicFinder: An Interface that Uses Auditory Icons. Human-Computer Interaction. 4. 67-94. 10.1207/s15327051hci0401_3.

36. Apple. (n.d.). *Apple Vision Pro*. https://apple.com/apple-vision-pro/