

DOI: <https://doi.org/10.36910/6775-2524-0560-2024-57-13>

УДК 004.5

Міронов Нікіта Олександрович, здобувач вищої освіти

Самчук Людмила Михайлівна, канд. техн. наук, доцент

<https://orcid.org/0000-0003-2516-045X>

Луцький національний технічний університет, м. Луцьк, Україна

ДОСЛІДЖЕННЯ ХАРАКТЕРИСТИК ТА МЕТОДОЛОГІЇ СИСТЕМИ РОЗПІЗНАВАННЯ ЖЕСТИВ ДЛЯ БЕЗКОНТАКТНОЇ ВЗАЄМОДІЇ З КОМП'ЮТЕРОМ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЙ COMPUTER VISION

Міронов Н.О., Самчук Л.М. Дослідження характеристик та методології системи розпізнавання жестів для безконтактної взаємодії з комп'ютером з використанням технологій Computer Vision. В роботі досліджено методи розпізнавання жестів для безконтактної взаємодії з комп'ютером, що використовують технології комп'ютерного зору та машинного навчання. Наведено приклади застосування в різних сферах діяльності людини, де вони забезпечують більшу безпеку, ефективність та зручність у використанні. Запропоновано методологію розпізнавання жестів рук у реальному часі. Система базується на алгоритмах, що використовують сегментацію зображень, виявлення контурів та класифікацію рухів рук. Основний акцент зроблено на застосуванні бібліотек OpenCV для обробки відеопотоків та MediaPipe для точного відстеження позицій пальців і рук. Мовою програмування для розробки системи є Python, який використовує дані та додаткові бібліотеки, що дозволяють інтегрувати різні функції. Розроблено діаграму прецедентів системи, визначено взаємодії компонентів, візуалізації сценаріїв використання та запропоновано шляхи застосування даної діаграми. Результатом дослідження є програмний продукт, який дозволяє керувати курсором миші за допомогою жестів рук. Система підтримує такі функції, як переміщення курсора, лівий та правий кліки, подвійний лівий клік, перетягування елементів, прокручування та регулювання гучності.

Ключові слова: розпізнавання жестів, комп'ютерний зір, віртуальна миша, машинне навчання, Python, MediaPipe, OpenCV, безконтактна взаємодія, обробка зображень, управління курсором.

Mironov N., Samchuk L. Study of characteristics and methodology of gesture recognition system for contactless interaction with a computer using Computer Vision technologies. The paper investigates methods of gesture recognition for contactless interaction with a computer using computer vision and machine learning technologies. Examples of application in various fields of human activity are given, where they provide greater safety, efficiency and ease of use. A methodology for recognizing hand gestures in real time is proposed. The system is based on algorithms that use image segmentation, contour detection, and hand motion classification. The main emphasis is placed on the use of OpenCV libraries for processing video streams and MediaPipe for precise tracking of finger and hand positions. The programming language used to develop the system is Python, which uses data and additional libraries to integrate various functions. A system precedent diagram was developed, the interactions of components were identified, use cases were visualized, and ways to use this diagram were proposed. The result of the study is a software product that allows you to control the mouse cursor using hand gestures. The system supports such functions as moving the cursor, left and right clicks, double left click, dragging elements, scrolling, and volume control.

Keywords: gesture recognition, computer vision, virtual mouse, machine learning, Python, MediaPipe, OpenCV, contactless interaction, image processing, cursor control.

Постановка наукової проблеми. Жести відіграють важливу роль у повсякденному житті людини як природний засіб спілкування. Вони використовуються для передачі емоцій, вираження думок і команд у взаємодії з іншими людьми [1]. Простота та універсальність жестів роблять їх ефективним засобом комунікації, зрозумілим незалежно від мови чи культурного контексту. Ця природна здатність до невербального спілкування відкриває можливості для її застосування і в технічних системах, зокрема для взаємодії з комп'ютерами.

Системи розпізнавання жестів для безконтактної взаємодії з комп'ютером набувають все більшої популярності завдяки можливості використання комп'ютерного зору та методів машинного навчання. Важливим аспектом таких систем є можливість зменшити потребу в фізичних пристроях, таких як миші та клавіатури, що особливо корисно для людей з обмеженими можливостями. Окрім того, такі технології відкривають нові перспективи для застосування в ігрових та віртуальних середовищах, де користувачі можуть управляти віртуальними об'єктами за допомогою рухів рук, подібно до взаємодії з фізичними предметами.

Аналіз досліджень. Системи розпізнавання жестів базуються на використанні технологій комп'ютерного зору, що дозволяє виявляти та аналізувати рухи користувача в реальному часі. Однією з найбільш використовуваних технологій є OpenCV, яка надає інструменти для обробки зображень та відеопотоків. Наприклад, у системах на основі OpenCV фіксуються рухи пальців та рук для симуляції натискання кнопок миші або для управління курсором [2].

Через OpenCV та обробку зображень можна використовувати систему маркерів, яка дозволяє контролювати комп'ютер за допомогою кольорових точок на кінчиках пальців, що забезпечує більш

стабільний контроль курсора і виконання команд. Цей метод використовує обробку зображень для виявлення відмінностей у кольорі та положенні пальців, що дозволяє системі визначати різні команди, такі як лівий або правий клік.

Дослідження показують, що використання методів глибокого навчання, таких як згорткові нейронні мережі (CNN), дозволяє досягати високої точності в розпізнаванні жестів. В одній із робіт, що дослідили дане питання [3], система на основі CNN дозволила здійснювати реальний контроль над комп'ютером за допомогою жестів рук, забезпечуючи точне відстеження та аналіз рухів. Такі моделі дозволяють системам адаптуватися до різних типів жестів і зменшують вплив зовнішніх факторів, таких як зміни освітлення або позицій користувача.

У багатьох дослідженнях підкреслюється перевага систем розпізнавання жестів, особливо в умовах обмеженого простору або під час пандемій, коли важливо мінімізувати фізичний контакт із пристроями. Наприклад, під час розпау COVID-19 розроблено систему безконтактної взаємодії з комп'ютером для зменшення поширення вірусу [4]. Це підтверджує актуальність та практичність застосування таких технологій у реальному житті.

Мета роботи. Метою та одним із напрямків дослідження є розробка ефективних алгоритмів для точного розпізнавання жестів, що використовують методи обробки зображень для перетворення рухів рук у команди. Важливим питанням є також забезпечення достатньої швидкодії системи при мінімальних вимогах до апаратного забезпечення.

Виклад основного матеріалу й обґрунтування отриманих результатів дослідження. Однією з ключових проблем, яка потребує вирішення, є точність розпізнавання жестів у складних умовах, наприклад, при змінному освітленні або у випадку часткового перекриття рук. Деякі системи пропонують використовувати електроміографічні (EMG) сигнали [5], що дозволяє підвищити точність розпізнавання жестів, аналізуючи сигнали від м'язів передпліччя. Це особливо корисно в середовищах, де камера не може забезпечити достатньо точне зчитування.

Різні системи розпізнавання жестів використовують різноманітну техніку, включаючи звичайні веб-камери, інфрачервоні сенсори та спеціалізовані пристрої, такі як Leap Motion. Останній забезпечує точне виявлення рухів пальців та рук людини [6]. Leap Motion дозволяє використовувати інтуїтивні жести для виконання різних завдань, таких як переміщення курсора, прокручування сторінок, регулювання гучності на комп'ютері та інших команд.

Системи розпізнавання жестів також знаходять застосування у спеціалізованих середовищах, таких як програми для моделювання в CAD-системах. Наприклад, використання EMG-сигналів для керування програмами, такими як Solidworks, дозволяє інженерам взаємодіяти з інтерфейсом безпосередньо за допомогою жестів.

Технології розпізнавання жестів також використовуються в комп'ютерних іграх і системах доповненої та віртуальної реальності для більш природної взаємодії з віртуальними об'єктами. Kinect, Leap Motion, VR-шоломи та інші подібні системи дозволяють користувачам маніпулювати віртуальними об'єктами за допомогою рухів не лише рук, а й, в окремих випадках, усього тіла, що створює більш інтерактивний досвід. Це дозволяє користувачам діяти у віртуальних середовищах так, як вони б взаємодіяли з реальними об'єктами.

Системи розпізнавання жестів все дедалі частіше використовуються і в інших сферах життя людини. До прикладу, в сучасних автомобілях вони дозволяють знизити відволікання водія під час керування, підвищують безпеку і комфорт. Вони дозволяють керувати мультимедійними та навігаційними пристроями, що забезпечує безконтактну взаємодію під час руху. Прикладом є компанії BMW та Mercedes-Benz, які впроваджують ці технології у свої преміальні автомобілі, дозволяючи водіям керувати основними функціями автомобіля за допомогою жестів. Ці інновації не тільки підвищують зручність водіння, але й знижують ризики аварій.

У військових операціях, особливо в інженерно-саперній справі, розпізнавання жестів стає ефективним інструментом безпеки. Віддалене керування роботами з маніпуляторами дає змогу виконувати небезпечні операції з мінімальною загрозою для життя [7]. Наприклад, рукавички DataGlove передають положення рук у реальному часі, дозволяючи керувати роботами з високою точністю. Це знижує ризики при розмінуванні та інших небезпечних завданнях.

Для навчання та розпізнавання команд у військовій справі використовуються системи на базі Kinect, які визначають рухи рук і передають команди. Ці системи дозволяють виявляти команди, що робить їх ефективними у військових операціях. Наприклад, логістична модель на базі Kinect продемонструвала ефективність розпізнавання жестів з точністю 96,75% [8].

Військові дослідження зосереджуються на створенні систем, здатних не тільки розпізнавати навчальні команди, але й вивчати нові за допомогою глибокого навчання. Такі системи використовують нейронні мережі, що дозволяє роботизованим партнерам адаптуватися до нових ситуацій та команд. Цей підхід має потенціал значно розширити можливості автоматизованих систем у бойових умовах, що, своєю чергою, в умовах військової ситуації на території України, є край необхідним та дуже актуальним під час виконання операцій.

Хоча багато сучасних систем демонструють високу точність, вони ще мають низку обмежень, таких як обмежена кількість розпізнаваних жестів та складність адаптації до нових сценаріїв. Для подальшого розвитку необхідно зосередитись на створенні адаптивних систем, здатних вивчати нові жести та пристосовуватися до різних умов використання.

Пропонована система розпізнавання жестів використовує технологію комп'ютерного зору, що дозволяє керувати курсором миші та виконувати основні дії, такі як кліки, прокручування та перетягування об'єктів, без фізичного дотику до пристрою. Система базується на використанні веб-камери для захоплення рухів рук, які обробляються та інтерпретуються програмним забезпеченням для виконання відповідних команд.

Технологія комп'ютерного зору (Computer Vision, CV) є основою системи розпізнавання жестів і дозволяє комп'ютеру «бачити» та інтерпретувати візуальну інформацію з камер або інших сенсорів. Комп'ютерний зір має в собі набір алгоритмів та методів, що обробляють та аналізують зображення чи відео для розпізнавання об'єктів, рухів та інших характеристик сцени [9]. У контексті системи розпізнавання жестів, Computer Vision використовується для виявлення рук користувача, визначення їхнього положення у просторі та розпізнавання специфічних жестів.

Процес включає кілька етапів:

- захоплення зображення;
- попередня обробка (фільтрація, покращення якості зображення);
- сегментація (виділення області рук);
- виявлення ключових точок (пальців та суглобів);
- інтерпретація жестів для перетворення їх у команди для управління комп'ютером.

Основною мовою програмування для розробки цієї системи є Python, що надає широкий спектр бібліотек для реалізації алгоритмів CV, обробки зображень та машинного навчання. Python був обраний завдяки його гнучкості, великій кількості доступних інструментів для обробки даних та простоті інтеграції з різними бібліотеками. У проекті використовуються ключові бібліотеки, такі як OpenCV та MediaPipe, які забезпечують розпізнавання жестів у реальному часі.

OpenCV (Open Source Computer Vision Library) є однією з найважливіших бібліотек у системі. Вона дозволяє захоплювати та обробляти зображення з камери, а також виконувати різноманітні операції, пов'язані з обробкою зображень, такі як виявлення контурів, сегментація та розпізнавання об'єктів. У даній системі OpenCV відповідає за зачитування відеопотоку з камери, обробку кадрів та підготовку даних для подальшого аналізу.

MediaPipe – це бібліотека для обробки мультимедійних потоків у реальному часі, яка широко використовується для розпізнавання рухів тіла, обличчя та рук [10]. У даній розробці MediaPipe виконує функцію трекінгу рухів рук та пальців, що дозволяє системі точно розпізнавати жести користувача. Використання бібліотеки MediaPipe дозволяє досягти високої точності та швидкості у розпізнаванні жестів без необхідності використання спеціалізованого апаратного забезпечення. Даний фреймворк дозволяє використати попередньо навчену модель для точної ідентифікації 21 різних точок на руці, включаючи пальці та суглоби долоні (рис. 1).

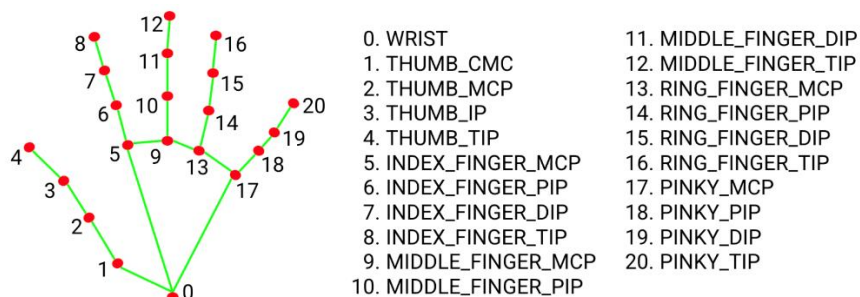


Рисунок 1 – Ідентифікація точок на руці навченої моделі бібліотеки MediaPipe

Окрім OpenCV та MediaPipe, у системі також застосовуються NumPy, ruyprut та інші бібліотеки. NumPy використовується для роботи з масивами даних та виконання математичних операцій. У контексті системи розпізнавання жестів ця бібліотека дозволяє ефективно обробляти числові дані, отримані з веб-камери, та виконувати обчислення, необхідні для аналізу рухів рук. Ruyprut відповідає за взаємодію з інтерфейсом користувача. Вона дозволяє програмно керувати мишею, виконуючи такі дії, як кліки, прокручування сторінок, переміщення курсора тощо.

Запустивши виконувану програму, система використовує функцію cv2.VideoCapture() з бібліотеки OpenCV для підключення до веб-камери. Ця функція забезпечує постійний потік кадрів у реальному часі, які будуть оброблятися програмою для виявлення жестів. Кожен кадр є окремим зображенням, яке програма аналізує. Кожен кадр з відеопотоку проходить попередню обробку. В Зображення може бути перетворене в інший колірний простір для точнішого виявлення об'єктів, таких як шкіра рук. Це дозволяє ефективніше розрізнити фон та руки користувача.

Для розпізнавання рук використовується навчена модель MediaPipe, яка дозволяє визначити необхідні точки на руці. Фреймворк забезпечує функцію виявлення та відстеження положення рук у кадрі незалежно від їхнього руху.

Коли система ідентифікує руку, вона починає аналізувати положення пальців для визначення жестів. Наприклад, якщо всі пальці розтягнуті, а долоня рухається по вертикальній осі, у цьому випадку, це інтерпретується як команда для регулювання гучності (рис. 2).

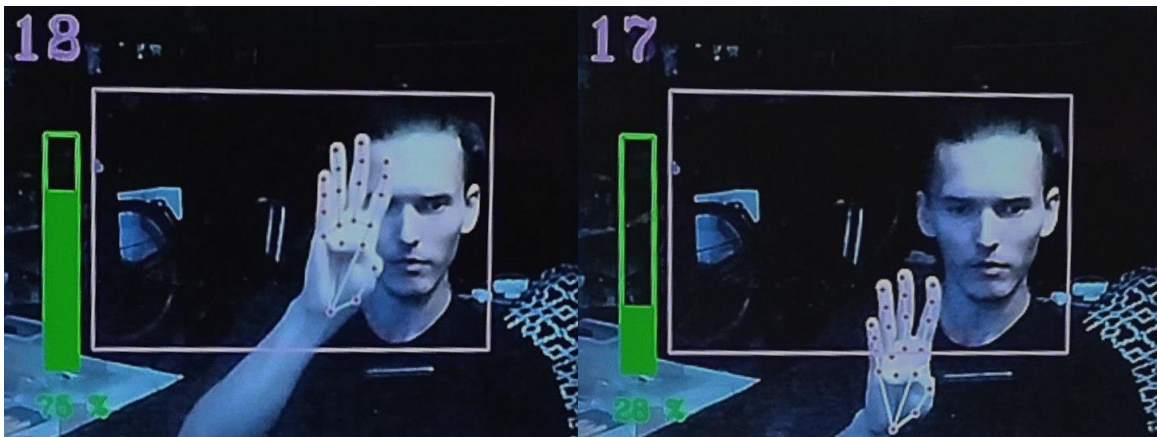


Рисунок 2 – Регулювання гучності

Якщо ж лише один вказівний палець піднятий, це інтерпретується як жест для руху курсора миші (рис. 3). Важливою частиною є розпізнавання просторових відносин між ключовими точками.

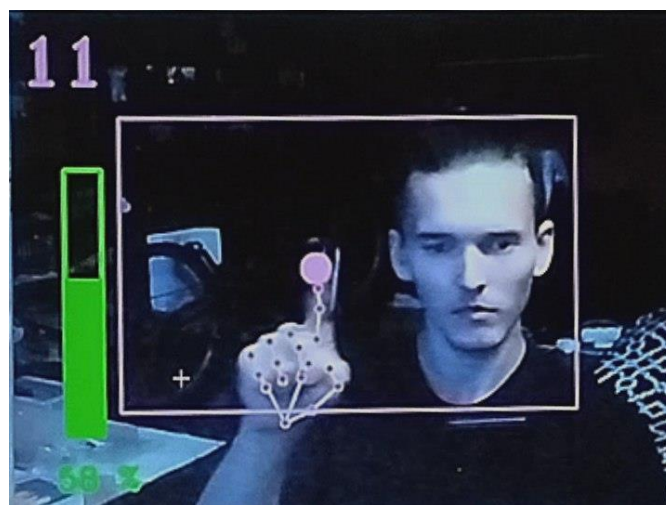


Рисунок 3 – Рух курсора миші

Використовуючи дані про позиції рук і пальців, програма передає ці координати бібліотеці ruprut для керування курсором миші. Ruprut може виконувати різні дії, такі як переміщення курсора на екрані залежно від положення долоні або виконання кліків при виявленні специфічних жестів (наприклад, розмикання пальців для кліку).

Для виконання дій, таких як кліки або прокручування сторінки, програма використовує додаткові функції ruprut. Наприклад, при виявленні жесту розмикання вказівного і середнього пальців, програма виконує лівий клік миші (рис. 4).

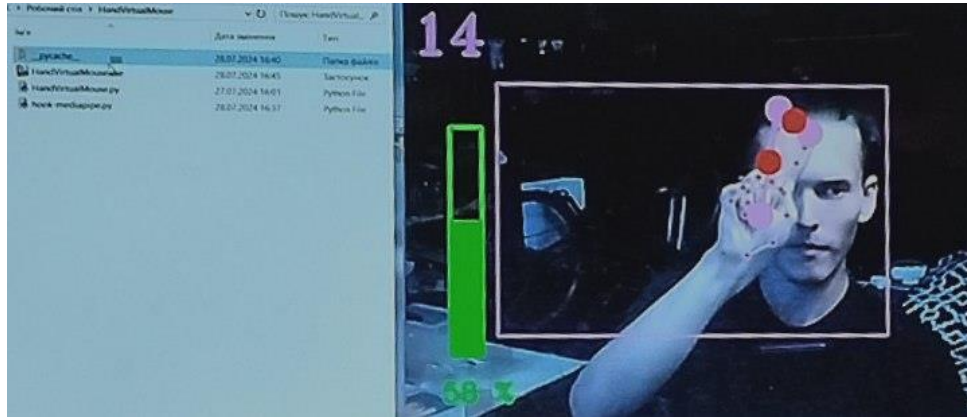


Рисунок 4 – Лівий клік миші

Прокручування реалізоване через положення великого пальця (рис. 5). Кожен жест вимагає специфічного обчислення на основі координат ключових точок на руці, що дозволяє програмі визначити точну дію.

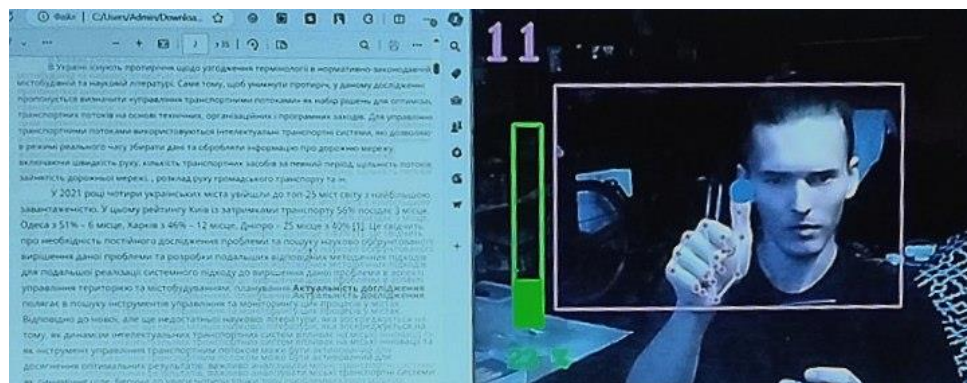


Рисунок 5 – Прокручування сторінок документа

Щоб уникнути «дрижання» курсору через незначні коливання рук, програма використовує методи фільтрації. Це включає згладжування координат або використання алгоритмів, що запобігають надто чутливій реакції системи на незначні рухи користувача. Плавне та згладжене пересування курсора миші досягається завдяки використанню функцій бібліотеки ruprut. Це дозволяє системі реагувати на жести користувача з більшим рівнем стабільності та плавності під час переміщення курсора.

Після обробки кожного кадру програма виводить результат на екран, показуючи положення руки або виконані дії. Коли програма завершена або камера більше не потрібна, використовується функція для закриття відеопотоку та звільнення ресурсів системи.

У процесі розробки системи розпізнавання жестів важливим етапом є моделювання основних сценаріїв використання. Для цього ефективним інструментом є UML діаграми прецедентів (Use Case Diagrams), які дозволяють візуалізувати взаємодію користувача із системою та визначити ключові функції, які вона підтримує. Ці діаграми є особливо корисними на етапі проектування, оскільки дають змогу краще зрозуміти, як користувачі будуть взаємодіяти з програмою та які компоненти відповідають за реалізацію певних дій.

UML діаграма прецедентів демонструє взаємодію між актором (користувачем або іншим зовнішнім елементом) та системою через конкретні прецеденти використання (use cases), що описують дії, які може виконати користувач [11]. У випадку даної системи, прецедентами можуть бути сценарії виконання певних команд за допомогою жестів, таких як переміщення курсора, виконання кліку тощо. Кожен з цих сценаріїв показує, як система відповідає на конкретні команди користувача.

Діаграми прецедентів також допомагають виявити основні компоненти системи, що беруть участь у виконанні цих дій. Наприклад, компонент, що відповідає за розпізнавання рухів рук, отримує дані з камери, а потім передає їх для обробки модулем керування інтерфейсом, який взаємодіє з операційною системою для виконання дій, таких як рух курсора або кліки.

Таким чином, UML діаграми дозволяють не лише відобразити взаємодію користувача з системою, але й деталізувати її внутрішні компоненти, показуючи, які модулі відповідають за конкретні функції. Для роз'яснення сценаріїв використання та компонентів системи розпізнавання жестів, використано UML діаграму прецедентів, що допомагає візуалізувати основні функціональні можливості системи та взаємодію користувача з нею (рис. 6). Опис ролі акторів та прецедентів системи наведено в табл. 1 і табл. 2 відповідно.

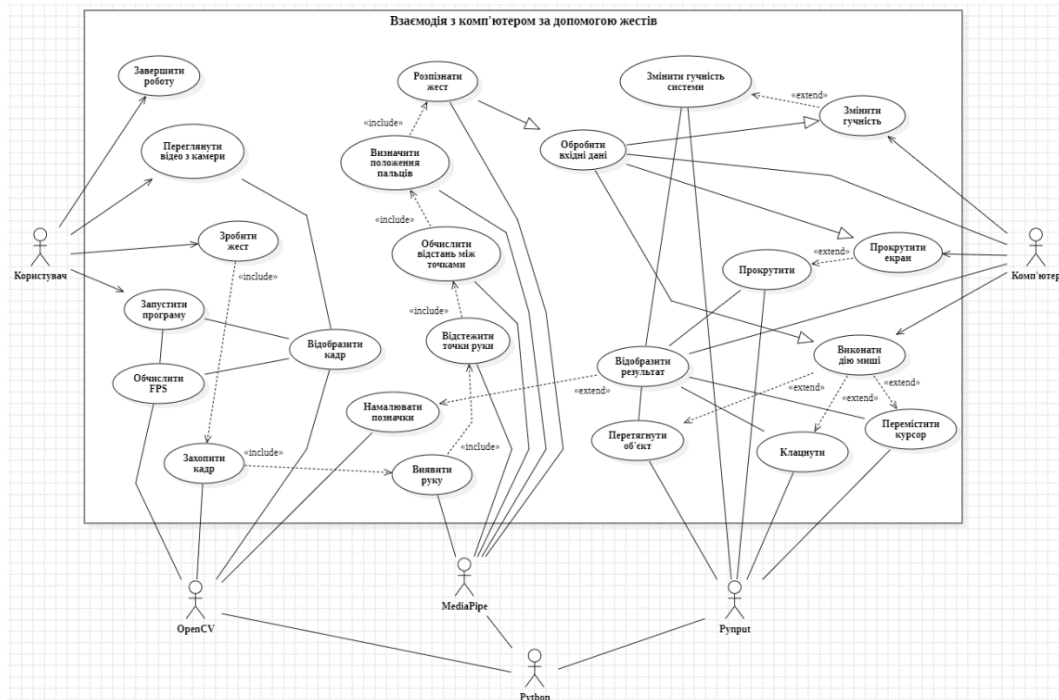


Рисунок 6 – Діаграма прецедентів процесу взаємодії з комп'ютером за допомогою жестів

Таблиця 1 – Опис ролі акторів системи

Актор	Опис ролі
Користувач	Особа, яка взаємодіє з системою. Користувач виконує жести перед камерою, які система інтерпретує як команди для керування комп'ютером. Він може запускати та завершувати роботу програми, а також виконувати різні дії на комп'ютері за допомогою жестів.
Комп'ютер	Пристрій, яким керує користувач за допомогою жестів. Комп'ютер виконує команди, такі як рух курсора, кліки, прокрутка, зміна гучності тощо. Він також надає зворотний зв'язок користувачеві через візуальний інтерфейс.
Python	Мова програмування, яка використовується для розробки та виконання системи. Python забезпечує основу для інтеграції різних компонентів системи, обробки даних та реалізації логіки управління. Відповідає за координацію роботи всіх компонентів та забезпечує загальну функціональність програми. У цій

	системі Python розділений на три основні бібліотеки: MediaPipe, OpenCV та ruyprut.
MediaPipe	Бібліотека, яка відповідає за розпізнавання та відстеження рук користувача. MediaPipe аналізує відеопотік з камери, виявляє руки, визначає положення ключових точок руки та інтерпретує жести.
OpenCV	Бібліотека обробки зображень, яка використовується для захоплення та обробки відео з камери. OpenCV забезпечує функціональність для перетворення кольорового простору, малювання на кадрах та відображення результатів користувачеві.
Ruyprut	Бібліотека для автоматизації взаємодії з графічним інтерфейсом користувача. Ruyprut виконує фактичні дії на комп'ютері, такі як переміщення курсора, кліки, прокрутка та зміна гучності, на основі розпізнаних жестів.

Таблиця 2 – Опис прецедентів системи

Прецедент	Опис прецеденту
Запустити програму	Ініціалізація програмного забезпечення, що включає активацію камери, завантаження необхідних бібліотек та підготовку системи до розпізнавання жестів.
Зробити жест	Процес виконання користувачем певного руху рукою перед камерою. Цей жест буде захоплено камерою та проаналізовано системою для подальшої інтерпретації як команди для комп'ютера.
Переглянути відео з камери	Відображення користувачеві відео в реальному часі з камери, що дозволяє йому бачити, як система «бачить» його руки та жести.
Завершити роботу	Коректне завершення роботи системи, що включає вивільнення ресурсів, закриття відеопотоку тощо.
Обробити вхідні дані	Процес аналізу відеопотоку з камери, виділення кадрів та їх попередня обробка для подальшого аналізу жестів.
Виконати дію миші	Реалізація команд, пов'язаних з керуванням курсором миші, таких як переміщення курсора, клік, подвійний клік або перетягування об'єктів.
Змінити гучність	Процес регулювання системної гучності комп'ютера на основі розпізнаних жестів користувача.
Прокрутити екран	Виконання прокрутки вмісту активного вікна вгору або вниз на основі жестів користувача.
Відобразити результат	Візуалізація результатів розпізнавання жестів та виконаних дій на екрані для надання користувачеві зворотного зв'язку.
Виявити руку	Процес ідентифікації наявності руки в кадрі відеопотоку з камери.
Відстежити точки руки	Визначення положення ключових точок руки (таких як суглоби пальців) у тривимірному просторі.
Визначити положення пальців	Аналіз взаємного розташування точок руки для визначення, які пальці підняті, а які опущені.
Розпізнати жест	Інтерпретація конфігурації руки та руху пальців як конкретного жесту, що відповідає певній команді.
Обчислити відстань між точками	Розрахунок відстаней між ключовими точками руки для визначення специфічних жестів, наприклад, для розрізнення кліків миші.
Захопити кадр	Отримання поточного кадру з відеопотоку камери для подальшої обробки.
Намалювати позначки	Додавання візуальних елементів на кадр для відображення

	розпізнаних рук, ключових точок тощо.
Відобразити кадр	Показ кадру з візуальними позначками користувачеві для зворотного зв'язку.
Обчислити FPS	Розрахунок кількості кадрів за секунду (FPS) для оцінки продуктивності системи та відображення цієї інформації користувачеві.
Перемістити курсор	Зміна позиції курсора на екрані відповідно до руху руки користувача.
Клацнути	Виконання кліку (лівою або правою кнопкою миші) на основі розпізнаного жесту.
Перетягнути об'єкт	Імітація натискання та утримання кнопки миші для переміщення об'єктів на екрані.
Прокрутити	Виконання вертикальної або горизонтальної прокрутки вмісту активного вікна.
Змінити гучність системи	Регулювання рівня системної гучності комп'ютера.

Як видно на рисунку 6, з UML діаграми прецедентів системи розпізнавання жестів для взаємодії з комп'ютером, цей вид моделювання дозволяє ефективно візуалізувати ключові етапи взаємодії користувача з системою. У діаграмі відображено, як користувач ініціює дії за допомогою жестів, і як система розпізнає та обробляє ці команди. Операції, такі як переміщення курсора, зміна гучності, прокручування екрана і так далі, деталізовано через взаємодію з конкретними функціями програми. Це дає чітке уявлення про те, як різні компоненти програми, зокрема бібліотеки OpenCV, MediaPipe та ruyprut, об'єднуються для досягнення поставленої мети.

Діаграма прецедентів також допомагає визначити всі ключові компоненти системи, які відповідають за обробку жестів та взаємодію з комп'ютером. Завдяки візуалізації процесу, розробники можуть краще розуміти, як реалізуються основні функції системи. Крім того, це сприяє фіксації вимог до системи та покращенню її надійності. Діаграма дозволяє бачити взаємозв'язки між компонентами, зокрема, як система розпізнає положення пальців, обчислює відстань між точками та відображає результати у вигляді дій миші.

Застосування UML діаграми для даної системи також відкриває можливість для подальшого її розвитку та вдосконалення. Використовуючи такі діаграми, можна покращити комунікацію між розробниками й тестувальниками, а також задокументувати всі процеси для полегшення майбутніх змін у системі. UML діаграми можуть бути застосовані для моделювання нових сценаріїв використання, таких як інтеграція нових жестів або підключення додаткових пристроїв вводу, що дозволить адаптувати систему до нових викликів та потреб користувачів.

Висновки. Розробка віртуальної миші на основі розпізнавання жестів демонструє значний прогрес у сфері безконтактної взаємодії з комп'ютером. Застосування комп'ютерного зору та алгоритмів машинного навчання забезпечує високу точність розпізнавання жестів у реальному часі.

Створено програмний продукт, який дозволяє керувати курсором за допомогою жестів рук, використовуючи веб-камеру. Система підтримує базові функції миші, такі як лівий і правий клік, переміщення курсора, прокручування та перетягування елементів, а також функцію регулювання гучності на комп'ютері.

Для покращення системи доцільно впровадити вдосконалені алгоритми виявлення кінчиків пальців для підвищення точності та швидкості взаємодії. Також варто звернути увагу на забезпечення стабільної роботи системи в різних умовах освітлення, що дозволить використовувати її в ширшому діапазоні ситуацій. Додатково можна інтегрувати підтримку багатопальцевих жестів та динамічних рухів для розширення функціональних можливостей.

Майбутні вдосконалення можуть включати розробку адаптивних систем, які зможуть динамічно підлаштовуватись під умови користувача та середовища. Інтеграція голосових команд та відстеження рухів очей разом з жестами рук також може підвищити зручність використання. Крім того, важливо забезпечити кросплатформну сумісність, що дозволить системі бути інтегрованою в різні операційні системи та пристрої.

Список бібліографічного опису

1. Roshnee Matlani, Roshan Dadlani, Sharv Dumbre, Shruti Mishra, Abha Tewari. Virtual Mouse using Hand Gestures. International Conference on Technological Advancements and Innovations. 2021. P. 12-13.

2. Mr.E.Sankar, B.Nitish Bharadwaj, A.V.Vignesh. Virtual Mouse Using Hand Gesture. International Journal of Scientific Research in Engineering and Management. Vol. 7. No. 5. May 2023. P. 2-4.
3. C. Cao, Y. Gao, S. Zhang. Real-Time Hand Gesture Detection and Recognition Using Convolutional Neural Networks. 2017. P. 30-42.
4. S. Shriram, B. Nagaraj, J. Jaya, S. Shankar, P. Ajay. Deep Learning-Based Real-Time AI Virtual Mouse System Using Computer Vision to Avoid COVID-19 Spread. Journal of Healthcare Engineering. 2021. P. 4-7.
5. Ala-Addin Nabulsi. Hand Gesture Recognition via Electromyographic (EMG) Armband for CAD Software control. 2018. P. 49-59.
6. Prutha Atre, Sahil Bhagat, Nevil Pooniwala, Payal Shah. Efficient and Feasible Gesture Controlled Robotic Arm. 2018. P. 34-54.
7. Mahmoud, Nourelhoda, Fouad, Hassan, Soliman, Ahmed. Smart healthcare solutions using the internet of medical things for hand gesture recognition system. Complex & Intelligent Systems. 2020. P. 7-12.
8. Michael Hamilton, Patrick Mead, Megan Kozub, Alexander Felid. Gesture Recognition Model for Robotic Systems of Military Squad Commands. 2016. P. 23-25.
9. Як технології комп'ютерного зору застосовуються в ритейлі. URL: <https://www.imena.ua/blog/computer-vision-technologies-in-retail/>
10. MediaPipe. Framework Concepts. URL: <https://chuoling.github.io/mediapipe/>
11. Documentation Plant UML. URL: <https://plantuml.com/>

References

1. Roshnee Matlani, Roshan Dadlani, Sharv Dumbre, Shruti Mishra, Abha Tewari. Virtual Mouse using Hand Gestures. International Conference on Technological Advancements and Innovations. 2021. P. 12-13.
2. Mr.E.Sankar, B.Nitish Bharadwaj, A.V.Vignesh. Virtual Mouse Using Hand Gesture. International Journal of Scientific Research in Engineering and Management. Vol. 7. No. 5. May 2023. P. 2-4.
3. C. Cao, Y. Gao, S. Zhang. Real-Time Hand Gesture Detection and Recognition Using Convolutional Neural Networks. 2017. P. 30-42.
4. S. Shriram, B. Nagaraj, J. Jaya, S. Shankar, P. Ajay. Deep Learning-Based Real-Time AI Virtual Mouse System Using Computer Vision to Avoid COVID-19 Spread. Journal of Healthcare Engineering. 2021. P. 4-7.
5. Ala-Addin Nabulsi. Hand Gesture Recognition via Electromyographic (EMG) Armband for CAD Software control. 2018. P. 49-59.
6. Prutha Atre, Sahil Bhagat, Nevil Pooniwala, Payal Shah. Efficient and Feasible Gesture Controlled Robotic Arm. 2018. P. 34-54.
7. Mahmoud, Nourelhoda, Fouad, Hassan, Soliman, Ahmed. Smart healthcare solutions using the internet of medical things for hand gesture recognition system. Complex & Intelligent Systems. 2020. P. 7-12.
8. Michael Hamilton, Patrick Mead, Megan Kozub, Alexander Felid. Gesture Recognition Model for Robotic Systems of Military Squad Commands. 2016. P. 23-25.
9. How computer vision technologies are used in retail. URL: <https://www.imena.ua/blog/computer-vision-technologies-in-retail/>
10. MediaPipe. Framework Concepts. URL: <https://chuoling.github.io/mediapipe/>
11. Documentation Plant UML. URL: <https://plantuml.com/>