

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи освітнього ступеня «магістр»
за спеціальністю 121 «Інженерія програмного забезпечення»
(освітня програма «Інженерія програмного забезпечення»)

на тему:

**«Система розпізнавання та аналізу рухів тіла з використанням штучного
інтелекту для спрощення соціальної адаптації у використанні ПК людям
з обмеженими фізичними можливостями»**

Виконав студент групи ІПЗмп-23-1
ДРАК Тарас Сергійович

Керівник роботи:
ЄФРЕМОВ Юрій Миколайович

Рецензент:
ВЛАСЕНКО Олег Васильович

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

«ЗАТВЕРДЖУЮ»

Зав. кафедри інженерії
програмного забезпечення
Тетяна ВАКАЛЮК
«14» лютого 2025 р.

ЗАВДАННЯ
на кваліфікаційну роботу

Здобувач вищої освіти: **ДРАК Тарас Сергійович**

Керівник роботи: **ЄФРЕМОВ Юрій Миколайович**

Тема роботи: **«Система розпізнавання та аналізу рухів тіла з використанням штучного інтелекту для спрощення соціальної адаптації у використанні ПК людям з обмеженими фізичними можливостями»,**

затверджена Наказом закладу вищої освіти від «14» лютого 2025 р., №60/с

Вихідні дані для роботи: аналіз та порівняння математичних моделей для прогнозування епідемічних показників.

Консультанти з магістерської кваліфікаційної роботи із зазначенням розділів, що їх стосуються:

Розділ	Керівник / Консультант	Завдання видав	Завдання прийняв
1	Єфремов Ю. М.	15.02.2025р.	15.02.2025р
2	Єфремов Ю. М.	15.02.2025р	15.02.2025р
3	Єфремов Ю. М.	15.02.2025р	15.02.2025р

РЕФЕРАТ

Кваліфікаційна магістерська робота полягає в розробці та впровадженні алгоритму для аналізу та розпізнавання рухів людського тіла з використанням методів штучного інтелекту, зокрема штучних нейронних мереж.

У дослідженні було виконано аналітичний огляд методів, технічних та програмних засобів, які використовують штучний інтелект для розпізнавання та аналізу рухів. Було проведено аналіз існуючих алгоритмів, розглянуто їх сильні та слабкі сторони, а також запропоновано власний підхід до реалізації алгоритму розпізнавання рухів тіла.

В результаті застосування запропонованого алгоритму вдалося вирішити задачу розпізнавання рухів тіла і досягти прийнятної точності результатів.

Пояснювальна записка до кваліфікаційної магістерської роботи складається із вступу, трьох розділів, списку літератури з 58 сторінок, 29 рисунків, 3 таблиці, 2 додатка.

КЛЮЧОВІ СЛОВА: АНАЛІЗ РУХІВ, РОЗПІЗНАВАННЯ РУХІВ, ШТУЧНИЙ ІНТЕЛЕКТ, ШТУЧНА НЕЙРОННА МЕРЕЖА, ГЛИБОКЕ НАВЧАННЯ, ОБРОБКА ЗОБРАЖЕНЬ.

ABSTRACT

The master's qualification work consists in the development and implementation of an algorithm for the analysis and recognition of human body movements using artificial intelligence methods, in particular artificial neural networks.

The study includes an analytical review of methods, technical and software tools that utilize artificial intelligence for movement recognition and analysis. An analysis of existing algorithms was conducted, highlighting their strengths and weaknesses, and a novel approach to implementing a body movement recognition algorithm was proposed.

As a result of the proposed algorithm's application, the task of body movement recognition was successfully solved, achieving an acceptable level of accuracy.

The explanatory note to the master's thesis consists of an introduction, four chapters, a bibliography of 58 pages, 29 figures, 3 tables, 2 appendices.

KEY WORDS: MOTION ANALYSIS, MOTION RECOGNITION, ARTIFICIAL INTELLIGENCE, ARTIFICIAL NEURAL NETWORK, DEEP LEARNING, IMAGE PROCESSING.

ЗМІСТ

РЕФЕРАТ.....	1
ABSTRACT	4
ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	7
ВСТУП.....	8
РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ СИСТЕМ РОЗПІЗНАВАННЯ АНАЛІЗУ РУХІВ ТІЛА З ВИКОРИСТАННЯМ ШТУЧНОГО ІНТЕЛЕКТУ	10
1.1 Поняття і принципи роботи систем розпізнавання рухів.....	10
1.2 Технології та методи штучного інтелекту для аналізу рухів	14
1.3 Сучасні досягнення в розпізнаванні рухів тіла в медичних та реабілітаційних технологіях	18
1.4 Особливості використання систем розпізнавання рухів тіла для людей з фізичними обмеженнями.....	21
1.5 Аналіз існуючих рішень для адаптації осіб з обмеженими можливостями до роботи з персональними комп'ютерами	22
Висновки до першого розділу	30
РОЗДІЛ 2 РОЗРОБКА СИСТЕМИ РОЗПІЗНАВАННЯ ТА АНАЛІЗУ РУХІВ ТІЛА ДЛЯ ПОЛЕГШЕННЯ СОЦІАЛЬНОЇ АДАПТАЦІЇ ОСІБ З ФІЗИЧНИМИ ОБМЕЖЕННЯМИ ПРИ ВИКОРИСТАННІ ПК	31
2.1 Постановка задачі та вимоги до системи.....	31
2.2 Опис архітектури системи та вибір компонентів та технологій	32
2.3 Алгоритми розпізнавання та аналізу рухів тіла за допомогою штучного інтелекту.....	34
2.4 Проектування інтерфейсу користувача для осіб з фізичними обмеженнями.....	40
Висновки до другого розділу.....	41
3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	42
3.1 Розробка програми та інтерфейсу для розпізнавання та аналізу рухів тіла	42
3.2 Результати роботи програми та її функціональні можливості	46
3.3 Тестування роботи програми	50

3.4 Рекомендації та перспективи щодо подальшого розвитку системи	53
Висновки до третього розділу	54
ВИСНОВКИ	55
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	56
ДОДАТКИ	59

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

МН – Машинне навчання

ШІ – Штучний інтелект

SVM (Support Vector Machine) – Мащини опорних векторів

ШНМ – Штучні нейронні мережі

CNN (Convolutional Neural Networks) – Згорткові нейронні мережі

FPS (frames per second) – Кадри в секунду

ВСТУП

Актуальність теми дослідження полягає у важливості створення інноваційних рішень для підтримки людей із фізичними обмеженнями, особливо в умовах сучасних викликів. Внаслідок повномасштабної війни Росії проти України кількість людей із обмеженими можливостями значно зросла, оскільки багато травм стали результатом бойових дій та обстрілів. Протягом трьох років цей показник збільшився на 300 тисяч, тобто на 10%. Це створює нагальну потребу у швидкому впровадженні інклюзивних технологій, адаптованих до умов воєнного часу. Попри існування різноманітних фондів, які допомагають людям у реабілітації, багато користувачів зіштовхуються зі складнощами у користуванні персональним комп'ютером через відсутність доступних технологічних рішень. Хоча на ринку є спеціальні системи, вони зазвичай є дорогими і недоступними для широкої аудиторії.

Мета роботи. Метою є всебічний аналіз і вдосконалення існуючих методів розпізнавання та аналізу рухів тіла. Це включає в себе розробку та дослідження роботи схожих систем, дослідження роботи машинного навчання, створення програмного забезпечення яке допомагає користуватись персональним комп'ютером.

Для досягнення цієї мети буде вирішено наступні завдання.

1. Провести аналіз існуючих технологій розпізнавання рухів тіла, включаючи методи комп'ютерного зору, алгоритми машинного навчання та їх ефективність у різних прикладних сферах.
2. Вивчення алгоритмів та моделей машинного навчання, що лежать в основі процесу розпізнавання рухів.
3. Оцінити апаратні та програмні засоби, що використовуються для реалізації систем розпізнавання рухів.
4. Створити експериментальний прототип системи розпізнавання рухів із можливістю її тестування у реальних прикладних сценаріях.

5. Провести порівняльний аналіз запропонованих рішень із існуючими системами, щоб оцінити їхню перевагу за точністю, швидкістю та стабільністю.
6. Розробити оптимізовані підходи для покращення точності, швидкості та адаптивності систем розпізнавання рухів тіла.

Предмет дослідження. Предметом дослідження є використання методів штучного інтелекту для відстежування рухів тіла за допомогою камери.

Об'єкт дослідження. Об'єктом дослідження є сучасні системи розпізнавання рухів тіла, які базуються на методах штучного інтелекту.

Наукова новизна полягає у розробці прототипу програми, що дозволяє людям з обмеженими фізичними можливостями, зокрема рухів верхніх кінцівок, керувати комп'ютером за допомогою рухів руки або біонічного протезу верхніх кінцівок. Програма забезпечує функціонал керування курсором, вибір елементів інтерфейсу та набір тексту за допомогою віртуальної клавіатури.

Методологія та методи дослідження. При дослідженні були використані такі методологічні дослідження як емпіричне моделювання, порівняльний аналіз та прототипування та тестування.

РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ РОЗПІЗНАВАННЯ АНАЛІЗУ РУХІВ ТІЛА З ВИКОРИСТАННЯМ ШІ

1.1 Поняття і принципи роботи систем розпізнавання рухів

Системи розпізнавання рухів – це технології, які здатні автоматично визначати, інтерпретувати та аналізувати рухи тіла людини з метою управління пристроями, створення аналітичних звітів або інтерактивної взаємодії.

В своїй основі існують дві системи для розпізнавання рухів а саме через камери та через датчики. Іноді вони можуть об'єднуватись для більш ретельного відстежування але частіше за все використовується один з них (рис. 1.1). Зображення представляє блок-схему системи розпізнавання рухів людини. Ця система використовується для автоматичного визначення, аналізу та інтерпретації рухів людського тіла. Вона може застосовуватися для керування різними пристроями, створення аналітичних звітів або для забезпечення інтерактивної взаємодії людини з обладнанням.

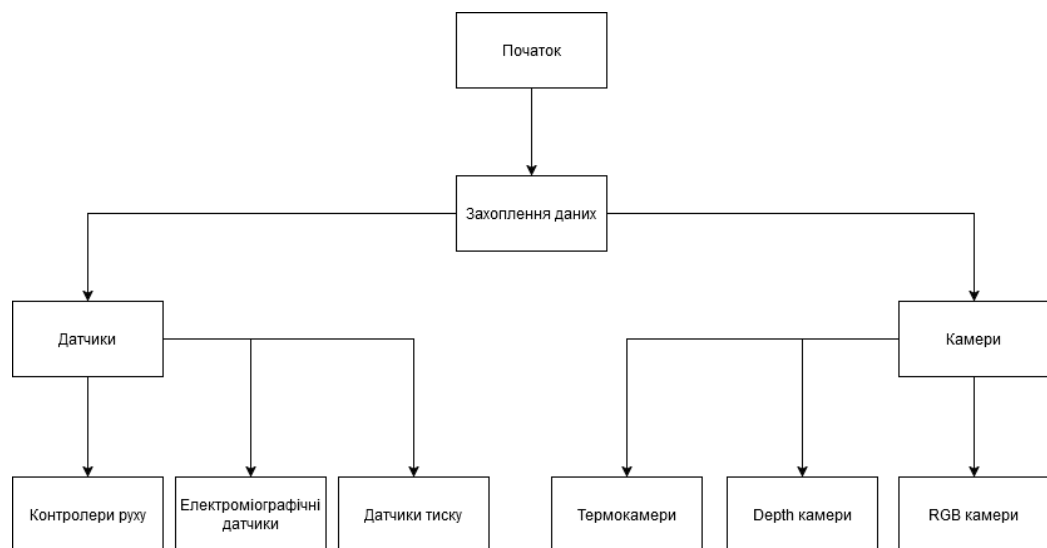


Рис. 1.1. Принципи розпізнавання руху

Системи розпізнавання рухів людини можна поділити на дві основні категорії: ті, що працюють на основі камер, і ті, що використовують датчики. Кожна з цих систем має свої особливості, переваги та обмеження, які визначають їхню придатність для різних застосувань.

Системи на основі камер працюють за принципом аналізу візуальної інформації, отриманої від відеокамер. Камери фіксують рухи тіла людини, а програмне забезпечення аналізує їх, використовуючи алгоритми комп'ютерного зору та машинного навчання. Сучасні системи часто використовують 3D-камери або RGBD-камери, які дозволяють захоплювати як кольорове зображення, так і дані про глибину[1]. Це дає можливість отримувати більш точну інформацію про положення частин тіла в просторі. Наприклад, технології на зразок Microsoft Kinect або OpenPose забезпечують точне розпізнавання скелетної моделі людини, що дозволяє відстежувати рухи навіть в умовах складного фону. Проте ці системи мають обмеження, пов'язані з якістю освітлення, кутом огляду камери та складністю обробки великого обсягу даних.

RGB-камери є найбільш поширеним типом і працюють за принципом захоплення кольорового зображення у видимому спектрі. Вони реєструють тривимірну сцену у вигляді двовимірного зображення з використанням червоного (R), зеленого (G) і синього (B) каналів. Алгоритми комп'ютерного зору, такі як OpenPose або MediaPipe, аналізують зображення для ідентифікації ключових точок людського тіла: ліктів, колін, суглобів тощо.

Depth-камери можуть відстежувати об'єкти навіть у слабкому освітленні, оскільки працюють у інфрачервоному спектрі. Алгоритми розпізнавання рухів аналізують тривимірну позицію кожної частини тіла, що забезпечує високу точність у складних середовищах. Наприклад, такі камери використовуються у Microsoft Kinect для створення "скелетних" моделей користувачів[2].

Термокамери особливо корисні у спеціалізованих застосуваннях, таких як спостереження в умовах низької видимості, безпека та охорона, а також у медичних дослідженнях, наприклад, для моніторингу температурних змін тіла[3].

Системи на основі датчиків ґрунтуються на використанні спеціальних пристроїв, які безпосередньо кріпляться до тіла людини або знаходяться

поруч. До таких датчиків належать інерційні вимірювальні пристрої (IMU), інфрачервоні сенсори, ультразвукові системи або навіть сенсори тиску. Вони реєструють рухи на основі змін у фізичних параметрах, наприклад, прискорення, обертання чи наближення до об'єкта. Подібні системи є надзвичайно точними, оскільки працюють незалежно від зовнішніх умов, таких як освітлення чи перешкоди в полі зору. Наприклад, у медичних застосуваннях інерційні датчики використовуються для моніторингу реабілітаційних вправ, а в спортивній аналітиці вони допомагають відстежувати продуктивність атлетів. Проте використання датчиків може бути менш зручним через потребу носіння додаткового обладнання.

Контролери руху – це пристрої, які зазвичай кріпляться на тіло або утримуються в руці для відстеження переміщень у просторі. Вони базуються на інерційних вимірювальних пристроях (IMU), що включають акселерометри, гіроскопи та магнітометри. Ці сенсори реєструють параметри руху, такі як прискорення, обертання та зміна положення. Сигнали з контролера обробляються для створення точного уявлення про траєкторію рухів. Наприклад, контролери, як-от Nintendo Wii Remote або VR-контролери, дозволяють гравцям взаємодіяти з віртуальними середовищами, відстежуючи кожен рух у реальному часі. Такі пристрої особливо корисні для ігор, тренувань або управління робототехнікою. Основні переваги – висока точність і автономність роботи без прив'язки до камер, але їх використання може бути обмежене, якщо потрібно розпізнати складні рухи всього тіла[4].

ЕМГ-датчики реєструють електричну активність м'язів. Ці пристрої базуються на технології електроміографії, яка вимірює електричні сигнали, що виникають у м'язах під час скорочення. ЕМГ-датчики зазвичай кріпляться на шкіру за допомогою електродів або впроваджуються у вигляді внутрішніх сенсорів. ЕМГ-датчики широко використовуються в медичних і спортивних застосуваннях. Наприклад, у реабілітації такі системи дозволяють моніторити прогрес пацієнтів із неврологічними або ортопедичними травмами. У спортивній аналітиці вони допомагають визначити ефективність тренувань,

аналізуючи роботу окремих м'язів. Іншою перспективною областю є протезування, де ЕМГ-датчики інтегруються з біонічними кінцівками для управління рухами[5].

Датчики тиску використовуються для вимірювання сили, яку людина прикладає до поверхні, або для реєстрації точок контакту. Вони зазвичай складаються з чутливих елементів, які реагують на механічний вплив (наприклад, вагу, натискання або силу удару). Ці пристрої можуть бути вбудовані в підлогу, сидіння, устілки взуття або рукавиці. Датчики тиску особливо корисні для оцінки біомеханіки рухів. Наприклад, у спортивній діагностиці вони допомагають аналізувати техніку бігу або стрибків, відстежуючи розподіл ваги під час руху. У медицині такі датчики використовуються для моніторингу ходьби пацієнтів після травм або операцій. У промислових застосуваннях їх інтегрують у робототехніку для забезпечення точного захоплення предметів.

Отож зробимо таблицю де розглянемо переваги та недоліки кожної з систем (табл. 1.1).

Таблиця 1.1

Порівняння технологій відстеження рухів

Критерій	Системи на основі камер	Системи на основі датчиків
Принцип роботи	Візуальний аналіз рухів за допомогою камер	Безпосереднє вимірювання фізичних параметрів тіла
Переваги	Широкий спектр застосувань, можливість відстеження великої кількості точок	Висока точність, незалежність від зовнішніх умов, можливість використання в складних умовах
Недоліки	Вразливість до освітлення та перешкод, високі вимоги до обчислювальних ресурсів, може бути складно налаштувати	Обмежена зона покриття, необхідність фізичного контакту, вища вартість

Точність	Залежить від якості камери та алгоритмів, може бути знижена при складних фонах	Дуже висока точність, особливо для локальних рухів
Час відгуку	Залежить від обчислювальних ресурсів та алгоритмів, може бути великим при складних обчисленнях	Зазвичай дуже швидкий
Енергоспоживання	Залежить від типу камери та напруги обчислювального блоку	Зазвичай низьке
Вартість	Відносно доступна для простих систем, може бути високою для складних систем	Вища вартість через необхідність придбання датчиків
Приклади застосування	Ігри, віртуальна реальність, відеоспостереження, аналіз рухів	Медицина, спорт, робототехніка, інтерфейси "мозок-комп'ютер"

Системи розпізнавання рухів демонструють значний прогрес і широке застосування в різних сферах, від розваг і медицини до безпеки й робототехніки. Аналіз особливостей систем на основі камер і датчиків показує, що кожен підхід має свої сильні сторони, але водночас стикається з певними проблемами.

1.2 Технології та методи штучного інтелекту для аналізу рухів

Аналіз рухів за допомогою штучного інтелекту має багаторічну історію розвитку, яка пройшла шлях від простих систем до складних моделей, здатних точно аналізувати біомеханіку рухів. Початок цієї галузі можна простежити до середини XX століття, коли з'явилися перші спроби створити системи розпізнавання жестів для керування пристроями. Ті системи базувалися на простих алгоритмах обробки зображень, як-от аналіз оптичного потоку, і мали обмежену точність[6]. Наприклад, система розпізнавання жестів, створена IBM [7] у 1992 році, працювала з датчиками для виявлення рухів рук і використовувала метод опорних векторів (SVM) для класифікації [8]. Це стало першим практичним застосуванням класифікаційних алгоритмів у системах розпізнавання жестів.

У 1990-х роках розвиток технологій призвів до створення перших нейронних мереж (CNN), які стали проривом у машинному навчанні[9]. Ян ЛеКун розробив LeNet-5, модель, яка була ефективною для розпізнавання рукописних символів, наприклад, цифр на банківських чеках[10]. Хоча LeNet-5 не використовувалася для аналізу рухів, вона заклала основу для CNN, здатних обробляти складніші дані. Розвиток апаратного забезпечення в 2000-х роках, зокрема поява потужних графічних процесорів (GPU), відкрив нові можливості для масштабного навчання нейронних мереж. Це дозволило використовувати CNN для аналізу рухів і жестів у реальному часі.

У 2012 році модель AlexNet стала революцією у сфері глибокого навчання. Вона продемонструвала високі результати в задачах розпізнавання зображень і довела ефективність CNN у складних візуальних задачах. Після цього CNN почали активно застосовувати для аналізу відео потоків і виявлення рухів. Одним із ключових рішень у цій сфері стала бібліотека OpenPose, розроблена в 2017 році. OpenPose дозволяє визначати ключові точки людського тіла, рук і обличчя, аналізуючи рухи навіть кількох осіб одночасно[11]. Її використання стало можливим у спорті, медицині, розвагах та безпеці. Розглянемо більш детально SVM і CNN та зробимо їх порівняння.

SVM - це алгоритм машинного навчання, який широко використовується для задач класифікації та регресії. На (рис. 1.2) ми бачимо схематичне представлення того, як SVM працює з зображеннями. Центральне місце займає зображення розміром 28x28 пікселів.

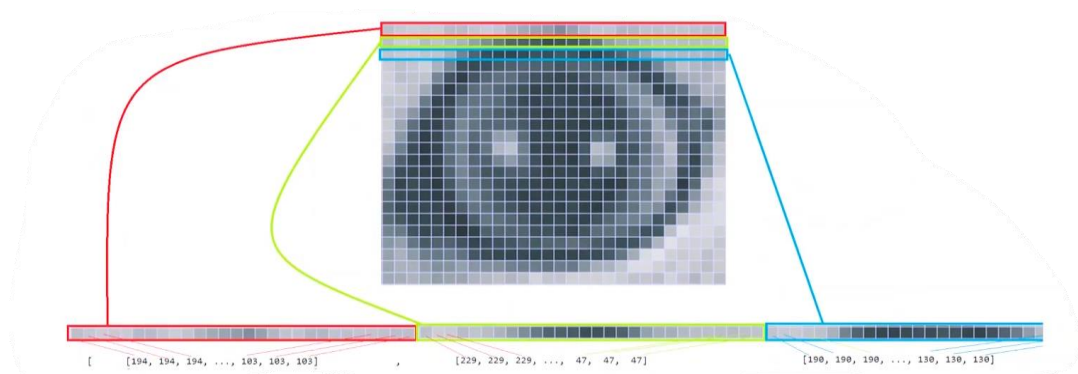


Рис. 1.2. схематичне представлення роботи SVM

Воно поділене на окремі пікселі, кожен з яких має своє значення. Над зображенням розташовані різнокольорові смужки. Кожна смужка відповідає певному набору пікселів, які SVM вважає важливими для класифікації. Ці смужки називаються опорними векторами. Саме вони визначають, де проходить межа між різними класами.

На (рис. 1.3) зображено спрощену схему роботи згорткової нейронної мережі (CNN) при розпізнаванні рукописних цифр. Центральне місце займає зображення цифри "9" розміром 28x28 пікселів. Це типовий розмір зображень, які використовуються для навчання багатьох моделей розпізнавання рукописних цифр. В середині зображення є основна частина нейронної мережі - приховані шари. Вони складаються з багатьох нейронів, з'єднаних між собою. Кожен нейрон в одному шарі отримує сигнали від кількох нейронів попереднього шару. Ці з'єднання мають різні ваги, які під час навчання мережі підлаштовуються таким чином, щоб мережа могла правильно класифікувати зображення.

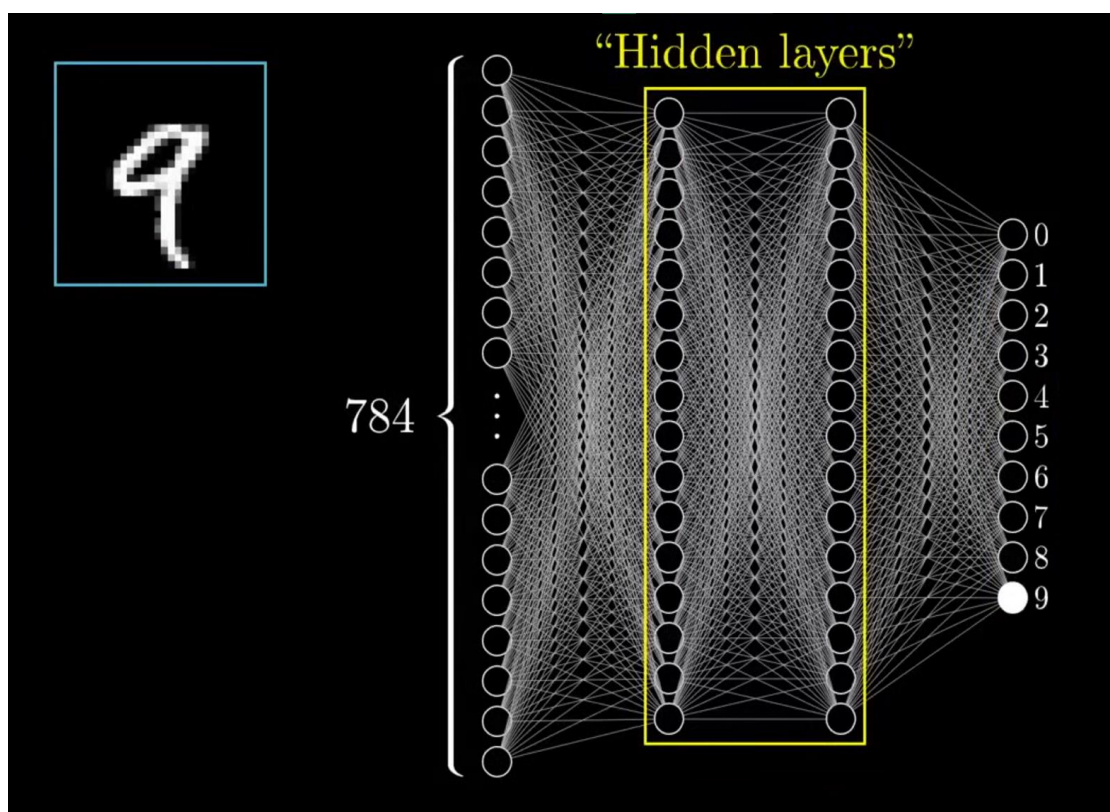


Рис. 1.3. схематичне представлення роботи CNN

Зробимо більш детальне порівняння обох методів в (табл. 1.2).

Таблиця 1.2

Порівняльна характеристика SVM та CNN

Характеристика	Метод опорних векторів (SVM)	Згорткові нейронні мережі (CNN)
Принцип роботи	Шукає оптимальну гіперплощину для розділення даних на класи.	Використовує згортки для виявлення ознак у зображеннях та класифікації.
Архітектура	Проста і зрозуміла.	Може бути складною, з великою кількістю шарів і параметрів.
Навчання	Відносно швидке навчання, особливо на невеликих наборах даних.	Вимагає великих обчислювальних ресурсів та великих наборів даних для навчання.
Вибір ознак	Вимагає ручного вибору ознак, що може бути трудомістким.	Автоматично вивчає ознаки з даних.
Нелінійні розділення	Використовує ядра для розділення даних у нелінійних просторах.	Може легко обробляти нелінійні залежності завдяки згорткам.
Інваріантність до змін	Менш інваріантна до змін масштабу, обертання та інших геометричних трансформацій.	Більш інваріантна до змін масштабу, обертання та інших геометричних трансформацій завдяки механізму згортки.
Узагальнення	Зазвичай добре узагальнюється на нові дані, особливо при невеликих наборах даних.	Може перенавчатися на великих наборах даних.
Застосування	Широко використовується для класифікації, регресії, розпізнавання образів.	Часто використовується для задач комп'ютерного зору, таких як класифікація зображень, сегментація, виявлення об'єктів.

Якщо порівнювати SVM і CNN у контексті аналізу рухів, можна побачити ключові відмінності[12]. SVM є класичним підходом, який ефективний на невеликих наборах даних і залежить від попередньо виділених ознак. CNN, навпаки, мають здатність автоматично виділяти ознаки, що робить їх більш точними й придатними для роботи з великими наборами даних. Проте CNN вимагають значних обчислювальних ресурсів. Незважаючи

на це, вони стали домінуючими в складних завданнях, таких як аналіз відео та виявлення складних рухів.

1.3 Сучасні досягнення в розпізнаванні рухів тіла в медичних та реабілітаційних технологіях

Сучасна медицина все активніше використовує технології для покращення діагностики та лікування різних захворювань. Особливо важливу роль відіграють технології, що дозволяють аналізувати рухи тіла людини. Ці системи знаходять широке застосування в реабілітації, спортивній медицині, а також в дослідженнях рухової активності людини. Аналіз руху дуже важливий через те що він допомагає виявити ранні ознаки різних захворювань, таких як хвороба Паркінсона, інсульт, артрит тощо. За допомогою систем аналізу рухів можна відстежувати прогрес пацієнтів під час реабілітації та коригувати план лікування. Завдяки аналізу рухів можна розробляти більш досконалі протези, які більш точно відтворюють природні рухи людини. Існують основні 3 технології для аналізу рухів на основі камер, датчиків руху, та EMG.

Luna EMG – це багатофункціональний реабілітаційний робот-тренажер, який використовується для відновлення рухових функцій верхніх та нижніх кінцівок, покращення нервово-м'язової координації, діапазону рухів і м'язової сили. Його унікальність – поєднання активного, пасивного та роботизованого тренування з біологічним зворотнім зв'язком на основі електроміографії, навіть якщо візуально рух кінцівкою ще не є помітним», – стверджує Волинська служба новин.

Luna EMG (рис. 1.4) використовує реакційну електроміографію для навчання рухів. Особливо ефективна вона для клінічно слабких пацієнтів, що дозволяє підвищити кількість рухових здібностей пацієнта при активації більшої кількості м'язів, ніж це можливо при ручному лікуванні.



Рис. 1.4. Luna EMG

За допомогою вбудованої 6-канальної електроміографії, Luna EMG дозволить вам надавати спеціалізовані клінічні оцінки:

- інноваційна діагностика та підбір рухових одиниць (через електроміографію поверхневих та внутрішніх тазових м'язів);
- вимірювання динамометричної сили;
- діапазон вимірювань руху (пасивний, допоміжний і активний);
- кількісно вимірює пластичність і жорсткість під час пасивного руху;
- послідовність активації м'язів;
- координація.

За допомогою Luna EMG можна виконувати Ізокінетичні, пропріоцептивні вправи, вправи з ваговим і пружним опором. Щоб заняття на тренажері не були рутинними, його можна запрограмувати на інтерактивні реабілітаційні ігри, які зацікавляють пацієнтів, особливо дітей.

GLOREHA ARIA – це сенсорний пристрій для терапевтичного лікування верхньої кінцівки та для когнітивних тренувань.

Інтерактивні ігри заохочують вільні рухи руки, зап'ястя та кисті в просторі без гравітації. Надзвичайне співвідношення ціни та якості: Gloreha

Aria (рис. 1.5) знаходиться в межах досяжності кожного реабілітаційного центру та фізіотерапевтичного центру, що лікує пацієнтів з дефіцитом рухових функцій верхніх кінцівок.



Рис. 1.5. Gloreha Aria

Пристрої Gloreha оснащені двома динамічними опорами, які дозволяють пацієнту рухати верхню кінцівку без жодних складнощів. Рівень компенсації регулюється відповідно до ваги руки та залишкових можливостей контролю та рухів пацієнта. Ці опори особливо корисні під час функціонального навчання, тому що в іншому випадку це було б неможливим без адекватного зниження ваги верхньої кінцівки.

Для комплексної реабілітації нейро-когнітивного відновлення має бути підтримка моторного відновлення. Конкретні вправи, розроблені командою Gloreha за підтримки нейропсихологи, тренують когнітивні здібності, такі як уміння уваги, вирішення проблем, пам'ять, навички зміни, вибіркову увагу, візуально-просторові дослідницькі навички тощо. Поєднання рухових та пізнавальних завдань також тренує розділену увагу, навик, що має надзвичайно важливе значення для виконання багатьох пацієнтів.

Технології аналізу рухів мають великий потенціал для покращення якості життя людей з різними захворюваннями. Однак, висока вартість обладнання є серйозною перешкодою для широкого його використання. Для

вирішення цієї проблеми необхідні спільні зусилля виробників обладнання, медичних закладів, держави та благодійних організацій

1.4 Особливості використання систем розпізнавання рухів тіла для людей з фізичними обмеженнями

Системи розпізнавання рухів тіла відкривають безліч можливостей для людей з обмеженими фізичними можливостями. Вони дозволяють замінити традиційні способи управління пристроями, використовуючи рухи тіла, жести або навіть погляд. Це стає особливо актуальним для людей, які втратили здатність використовувати стандартні пристрої вводу, такі як миша чи клавіатура.

Одним із головних аспектів є персоналізація цих систем під індивідуальні потреби користувачів. Кожна людина має унікальні фізичні можливості, тому налаштування системи має враховувати як її поточний стан, так і потенційні зміни в майбутньому. Наприклад, люди з частковою рухливістю кінцівок можуть використовувати системи, які розпізнають невеликі жести пальців, тоді як для користувачів із паралічем доцільніше застосовувати айтрекери або управління рухами голови.

Технології на основі камери, як системи комп'ютерного зору (наприклад, OpenPose), дозволяють розпізнавати складні рухи, зокрема жести рук, повороти голови чи положення тіла. Вони є безконтактними та менш інвазійними, що робить їх зручними для щоденного використання. Однак точність таких систем може залежати від освітлення та якості камери, що є викликом у певних умовах.

Адаптивні технології також відіграють важливу роль у реабілітації. Наприклад, пристрої на зразок **GLORENA ARIA** допомагають відновлювати рухові функції верхніх кінцівок, забезпечуючи підтримку під час виконання вправ. Це поєднує терапевтичну функцію з тренуванням моторики. Інтерактивні ігри, інтегровані в такі системи, мотивують пацієнтів до регулярного використання пристроїв, що пришвидшує процес реабілітації.

Додатковим напрямом розвитку є імпланти, які доповнюють або повністю замінюють рухові функції користувачів. Такі пристрої інтегруються з системами розпізнавання жестів, що дозволяє керувати ними за допомогою залишкових рухів тіла або навіть сигналів мозкової активності. Це значно розширює можливості для людей із серйозними порушеннями рухливості.

Програмне забезпечення для управління комп'ютером також адаптується до різних сценаріїв використання. Наприклад, віртуальні клавіатури та програми для голосового набору тексту, інтегровані з технологіями розпізнавання рухів, дозволяють виконувати складні завдання навіть за мінімальних фізичних зусиль. Такі рішення стають ключовими для забезпечення доступності освіти та професійної діяльності.

Однак, незважаючи на всі переваги, існують і певні обмеження. Основними викликами залишаються висока вартість систем, технічні труднощі їх налаштування, а також складність інтеграції в існуючі технології. Крім того, індивідуальні відмінності користувачів можуть ускладнювати створення універсальних рішень.

Майбутнє таких систем передбачає інтеграцію кількох технологій — від комп'ютерного зору та штучного інтелекту до біонічних пристроїв і сенсорних систем. Завдяки цьому можна очікувати появу більш доступних і ефективних рішень, які не лише покращать якість життя людей із фізичними обмеженнями, але й допоможуть у їхній соціальній адаптації.

1.5 Аналіз існуючих рішень для адаптації осіб з обмеженими можливостями до роботи з персональними комп'ютерами

Технології для адаптації осіб із обмеженими можливостями до роботи з персональними комп'ютерами стають дедалі більш різноманітними та доступними, завдяки стрімкому розвитку штучного інтелекту, комп'ютерного зору та інтерфейсів взаємодії. Існуючі рішення охоплюють як апаратні, так і програмні засоби, які дають змогу людям із фізичними обмеженнями успішно працювати з комп'ютером, виконуючи завдання від базової навігації до створення складних документів.

Одним із найбільш популярних рішень є програмне забезпечення для керування комп'ютером за допомогою голосу. Наприклад, такі системи, як **Dragon NaturallySpeaking** або вбудовані функції голосового керування в Windows і macOS, дозволяють користувачам виконувати завдання без використання миші або клавіатури. Ці системи використовують алгоритми розпізнавання мовлення для виконання команд, таких як відкриття файлів, запуск програм чи набір тексту. Проте голосове управління має певні обмеження: воно може бути менш ефективним у шумному середовищі або для людей з порушеннями мовлення.

Трішки детальніше розглянемо розробку компанії Microsoft а саме Kinect який працює за системою Depth камер. «Kinect» — це інноваційний пристрій, який революціонізував спосіб взаємодії з ігровими консолями та комп'ютерами. Замість традиційних контролерів, таких як геймпади, Kinect використовує безконтактний інтерфейс, який дозволяє керувати пристроєм за допомогою жестів, голосу та розпізнавання обличчя (рис. 1.6).



Рис. 1.6. Kinect

Kinect відкриває нові горизонти в ігровому світі, дозволяючи грати, використовуючи все тіло. Від спортивних симуляторів до танцювальних ігор, цей пристрій робить ігри більш інтерактивними та захоплюючими. Крім розваг, Kinect можна використовувати для домашніх тренувань, оскільки він відстежує рухи та надає зворотний зв'язок. Також пристрій застосовується в різних розважальних програмах, таких як віртуальні тури, музичні ігри та інтерактивне спілкування.

Пристрій має широкі можливості. Він дозволяє керувати іграми, програмами та інтерфейсом за допомогою рухів рук і тіла. Також

підтримується функція голосових команд, що дає змогу виконувати різні дії без використання фізичних кнопок. Kinect може розпізнавати обличчя, що використовується для персоналізації досвіду користувача. Відстежуючи рухи всього тіла, пристрій створює ефект повного занурення в гру або програму. Додатково Kinect здатен створювати 3D-моделі об'єктів і людей, відкриваючи нові можливості для творчості та розробки.

Kinect працює завдяки інтеграції декількох ключових технологій, які разом забезпечують точне розпізнавання рухів тіла, об'єктів та голосу. Пристрій оснащений набором датчиків, що дозволяють йому взаємодіяти з навколишнім середовищем. Камера глибини створює 3D-зображення простору перед пристроєм, що дає можливість Kinect точно відстежувати рухи тіла користувача. Кольорова камера забезпечує деталізоване зображення оточення, що допомагає розпізнавати обличчя та предмети. Масив мікрофонів записує голосові команди та дозволяє Kinect розпізнавати мову. Також в пристрої використовується інфрачервоний проектор, який створює невидиму ІЧ-сітку, що допомагає у визначенні глибини (рис. 1.7):

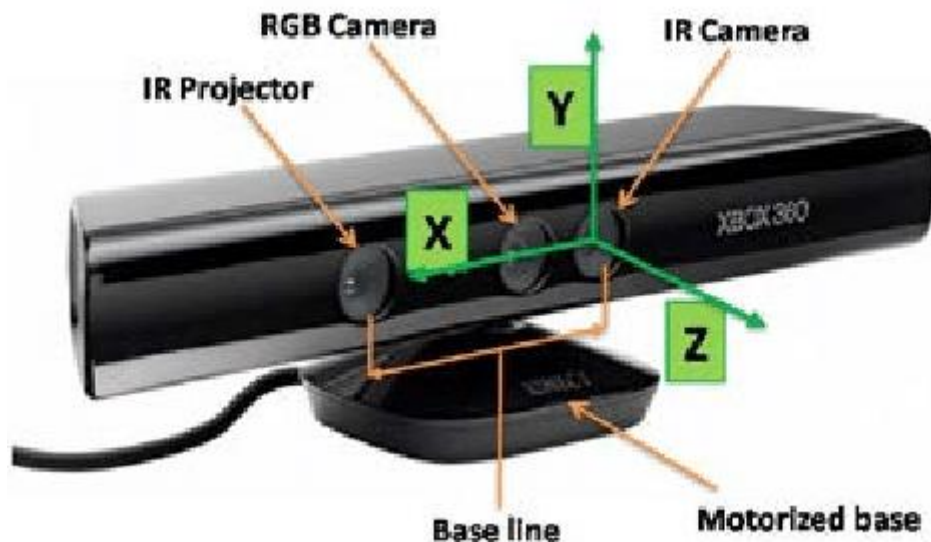


Рис. 1.7. Kinect датчики

Камера глибини формує своєрідну «кімнату» з точками, кожна з яких відповідає певному положенню у просторі. Ця модель постійно оновлюється,

щоб Kinect міг в реальному часі відстежувати рухи користувача. Програмне забезпечення пристрою аналізує зміни у 3D-моделі та визначає, які частини тіла рухаються і в якому напрямку. Масив мікрофонів записує голосові команди, перетворюючи звукові хвилі в текст та порівнюючи їх із базою команд. Кольорова камера формує зображення обличчя користувача, яке потім аналізується програмним забезпеченням для визначення емоційних виразів і особливостей. Завдяки поєднанню цих технологій Kinect може розпізнавати жести, голосові команди та міміку, що робить взаємодію з пристроєм максимально природною

Завдяки цій складній системі датчиків Kinect може точно відстежувати рухи тіла, розпізнавати жести, голосові команди та вирази обличчя, що робить його незамінним інструментом для інтерактивних ігор, фітнесу, розваг та інших застосувань.

Ай-трекінг (eye-tracking) — технологія відстеження положення очей, також звана технологією відстеження ліній погляду або точок погляду ще іноді називають «окулографія», але термін не прижився. Технологія ця відносно нова і в Україні вона ще не отримала належного поширення, хоча в останні роки ситуація змінюється на краще.

Пристрої для відстеження погляду або ай-трекери, складаються як правило з апаратної частини, яка підключається до комп'ютера (ноутбуку, планшету) і програмного забезпечення. Хоча існують і самодостатні пристрої «все в одному».

Трекінг очей чимось схожий на Kinect тим що використовується інфрачервоне підсвічування. Воно відбивається в очах користувача після чого камера айтрекера вловлює ці відображення та шляхом фільтрації та математичних розрахунків пристрою визначає положення погляду (рис. 1.8).



Рис. 1.8. Приклад системи Eye-tracking

Технологія ай-трекінгу застосовується у різних сферах, зокрема в нейромаркетингу, медичних дослідженнях, спорті, освіті та науці. Вона використовується для аналізу поведінки покупців, тестування інтерфейсів, професійної підготовки пілотів та в когнітивній психології. Окремий напрямок розвитку — реабілітаційні ай-трекери, які допомагають людям з особливими потребами управляти комп'ютером лише за допомогою погляду. Вони дозволяють набирати текст, користуватися інтернетом та спілкуватися в соціальних мережах без сторонньої допомоги, фактично замінюючи клавіатуру та мишу.

Можливості цієї технології включають відстеження рухів очей, аналіз емоційних реакцій на різні стимули, оцінку зручності інтерфейсів та пристроїв, розробку ефективних навчальних матеріалів, а також глибше дослідження емоційних реакцій користувачів

Айтрекінг — це інноваційний інструмент для дослідження когнітивних процесів, поведінки користувачів та ефективності різних інтерфейсів. Він дозволяє зазирнути в розум людини і зрозуміти, як вона сприймає інформацію. Застосування айтрекінгу відкриває нові можливості для поліпшення продуктів і послуг, а також для більш глибокого розуміння людської психіки.

Контролери руху — це інноваційні пристрої, які дозволяють користувачам взаємодіяти з цифровим світом більш природним та інтуїтивним

способом, імітуючи реальні рухи рук. Замість традиційних кнопок і джойстиків, ці контролери використовують різноманітні датчики для відстеження положення і орієнтації в просторі, дозволяючи користувачам маніпулювати віртуальними об'єктами, грати в ігри та взаємодіяти з додатками більш інтуїтивно.

Ідея використання рухів тіла як механізму керування спочатку розраховувалась для відеоігор та почалась в 1960-х і 70-х років. Atari і Magnavox досліджували примітивні контролери з датчиками руху і такі ігри, як боксерські симулятори. Однак технології того часу були надто обмеженими для складного виявлення рухів.

У 1980-х роках Mattel випустила аксесуар Power Glove для Nintendo Entertainment System (рис. 1.9). Хоча він випередив свій час, він покладався на просте розпізнавання захоплення і виявився занадто неточним для більшості ігор. Технології ще не були готові до свого часу.



Рис. 1.9. Power Glove

Сучасні контролери руху використовуються у різних сферах. Віртуальні реальність та геймінг активно застосовують такі пристрої, як Oculus Touch, HTC Vive Controllers, Valve Index Controllers, а також ігрові контролери. Вони дають змогу імітувати рухи рук, ніг та інших частин тіла для більш реалістичної взаємодії з віртуальним середовищем. Також контролери

використовуються для створення віртуальних тренерів і симуляторів у спорті, а також для керування роботами та промисловими пристроями.



Рис. 1.10. рукавички для VR

Технологія контролерів руху дозволяє точно відстежувати рухи рук, пальців та навіть всього тіла. Висока швидкість обробки забезпечує мінімальну затримку між рухом користувача та реакцією системи, що робить взаємодію максимально плавною та природною. Пристрої легко інтегруються з іншими технологіями, такими як віртуальна і доповнена реальність, а також штучний інтелект. Крім того, вони можуть бути налаштовані відповідно до конкретних потреб користувача, що розширює їхні можливості в різних галузях.

Розробки на основі комп'ютерного зору, такі як **OpenPose** або інтеграції на базі технологій Microsoft Kinect, стали основою для створення систем, які дозволяють керувати комп'ютером за допомогою жестів. Ці системи аналізують рухи користувача в реальному часі, що забезпечує безконтактне управління. Подібні технології вже застосовуються в реабілітаційних центрах, проте їх ефективність залежить від умов освітлення та просторових обмежень.

Нарешті, спеціалізовані віртуальні клавіатури, такі як **Dasher** або функції екранної клавіатури, інтегровані в операційні системи, дозволяють людям набирати текст за допомогою лише одного пристрою введення, наприклад, кнопки або айтрекера. Ці рішення є особливо корисними для людей із частковими або повними порушеннями рухливості рук.

Попри різноманітність існуючих рішень, кожне з них має свої переваги та обмеження, залежно від конкретних потреб користувача тому зробимо таблицю для кращого угруповання характеристик (табл. 1.3).

Таблиця 1.3

Порівняння методів керування інтерфейсом

Характеристика	Голосове управління	Kinect	Eye-tracking	Контролери руху
Принцип роботи	Розпізнавання мовлення	Аналіз зображення з камери глибини, розпізнавання жестів, голосу	Відстеження руху очей	Відстеження рухів рук і тіла
Області застосування	Набір тексту, керування програмами	Ігри, фітнес, реабілітація, інтерфейси	Дослідження, реабілітація, маркетинг, дизайн інтерфейсів	Ігри, віртуальна реальність, промисловість
Переваги	Не вимагає фізичних дій, інтуїтивно зрозумілий	Інтерактивний, багатофункціональний, безконтактний	Точний аналіз зорової уваги, неінвазивний	Інтуїтивний, природний інтерфейс
Недоліки	Залежить від якості мікрофона, може бути неефективним у шумному середовищі	Вимагає вільного простору, може бути чутливим до освітлення	Висока вартість обладнання, обмежена область застосування	Вимагає калібрування, може бути незручний при тривалому використанні
Точність	Залежить від якості розпізнавання мови	Залежить від якості камери та алгоритмів обробки	Висока точність відстеження погляду	Висока точність відстеження рухів
Швидкість	Залежить від потужності процесора та якості алгоритмів	Залежить від частоти кадрів камери	Дуже висока швидкість	Залежить від частоти оновлення даних з датчиків
Вартість	Відносно дешево	Середня вартість	Висока вартість	Середня вартість
Складність використання	Простий у використанні	Вимагає деякого навчання	Необхідне калібрування	Відносно простий

Як ми можемо помітити висока вартість сучасних адаптивних технологій та складність їхнього налаштування залишаються основними проблемами. Подальший розвиток у цій сфері передбачає інтеграцію різних підходів, таких як комп'ютерний зір, штучний інтелект і біомеханічні пристрої, для створення універсальних рішень, які будуть доступними для широкого кола користувачів із фізичними обмеженнями.

Висновки до першого розділу

У першому розділі розглянуто основи систем розпізнавання рухів тіла, що базуються на застосуванні ШІ. Було проаналізовані основні поняття, та принципи роботи таких систем, а також технології, які забезпечують їхню ефективність. Особливу увагу приділено сучасним досягненням у сфері розпізнавання рухів, зокрема їх використанню в медицині, реабілітаційних технологія.

Розвиток методів штучного інтелекту дозволяє створювати більш точні, бюджетні та адаптивні системи аналізу рухів, що відкриває нові можливості для інтеграції таких технологій у повсякденне життя. Було проведено аналіз існуючих рішень, які допомагають адаптувати осіб з обмеженими можливостями до роботи з комп'ютерами. Виявлено, що, незважаючи на значний прогрес у розробці таких систем, залишається ряд проблем, пов'язаних із доступністю, вартістю та необхідністю індивідуального налаштування.

РОЗДІЛ 2 РОЗРОБКА СИСТЕМИ АНАЛІЗУ РУХІВ ТІЛА ДЛЯ ПОЛЕГШЕННЯ АДАПТАЦІЇ ОСІБ З ФІЗИЧНИМИ ОБМЕЖЕННЯМИ ПРИ ВИКОРИСТАННІ ПК

2.1 Постановка задачі та вимоги до системи

Для створення ефективної системи, яка допомагає людям із фізичними обмеженнями працювати з персональними комп'ютерами, необхідно врахувати ряд технічних, функціональних та користувацьких вимог. Вимоги до системи розпізнавання та аналізу рухів тіла для спрощення соціальної адаптації людей з фізичними обмеженнями повинні враховувати широкий спектр технічних, функціональних і ергономічних аспектів. Насамперед система має бути доступною для людей із різними фізичними можливостями, тобто підтримувати адаптивність до індивідуальних потреб користувачів. Це передбачає можливість налаштування під особливості рухів кожної людини, а також забезпечення стабільної роботи в умовах різних фізичних обмежень.

Система повинна забезпечувати високу точність розпізнавання рухів, щоб уникнути помилкових інтерпретацій або некоректного виконання команд. Для цього необхідно використовувати передові алгоритми машинного навчання, які здатні адаптуватися до складних сценаріїв, включаючи зміни освітлення, фонового шуму чи швидкості рухів. Крім того, система має працювати в реальному часі, забезпечуючи миттєву реакцію на рухи користувача, що є критично важливим для інтерактивної взаємодії.

Інтерфейс системи повинен бути простим, зрозумілим і інтуїтивно доступним. Це означає, що навіть користувачі без спеціальної підготовки зможуть легко освоїти її функціонал. Також важливою вимогою є адаптивність системи до змінних умов використання. Висока стабільність роботи за таких умов підвищить її надійність і корисність у повсякденному житті.

2.2 Опис архітектури системи та вибір компонентів та технологій

На схемі зображено загальну структуру системи розпізнавання рухів. Спочатку система захоплює зображення з камери, тобто робить фотографію в поточний момент часу. Отримане зображення піддається попередній обробці, що включає в себе різні фільтри та алгоритми для покращення якості зображення. Після обробки, система шукає на зображенні так звані "ключові точки" - це можуть бути кінчики пальців, суглоби, очі тощо. Ці точки є своєрідними маркерами, за якими система буде відстежувати рухи.

Далі, система починає відслідковувати зміну положення цих ключових точок з часом. Тобто, порівнює їх положення на попередньому і поточному зображеннях. Це дозволяє визначити, як рухаються різні частини тіла. Отримані дані про рух ключових точок аналізуються системою для визначення того, який саме рух виконав користувач. Це може бути жест, поза або будь-яка інша дія. На основі цієї інформації система виконує відповідну дію. Наприклад, якщо користувач показав жест "вгору", система може збільшити гучність звуку. Робота систем розпізнавання рухів базується на аналізі візуальної або сенсорної інформації, отриманої від користувача на прикладі блок схеми (рис. 2.1).

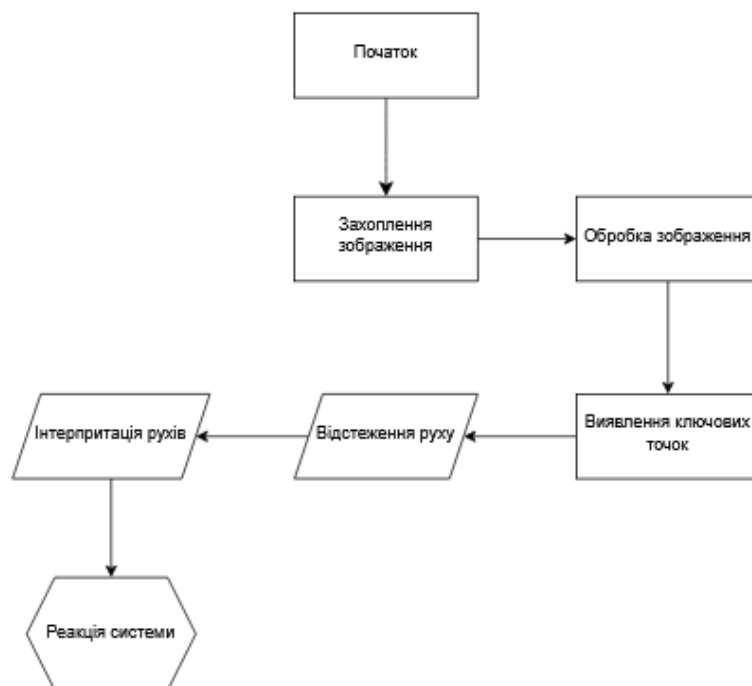


Рис. 2.1. Архітектура системи розпізнавання рухів

Розберемо детальніше кожен з представлених етапів. Перший етап — **захоплення зображення**. На цьому етапі здійснюється фіксація візуальної інформації з реального світу за допомогою різних пристроїв, таких як веб-камери, цифрові фотоапарати або смартфони. Захоплене зображення перетворюється в цифровий формат, який складається з матриці пікселів. Кожен піксель містить інформацію про колір або рівень яскравості. Це перетворення дозволяє комп'ютеру працювати з візуальними даними у вигляді числових масивів, що є основою для подальшого аналізу.

Далі слідує **обробка зображення**, яка включає низку операцій для покращення якості даних. На цьому етапі видаляються шуми, нормалізується освітлення, а також виконується попереднє перетворення зображення у формат, придатний для подальшої роботи. Наприклад, зображення може бути переведено в відтінки сірого або масштабовано до заданих розмірів. Такі дії роблять дані більш однорідними та полегшують їх аналіз.

Наступним важливим кроком є **визначення ключових точок**. Цей етап полягає у виявленні характерних елементів на зображенні, які будуть використані для відстеження рухів. Ключовими точками можуть бути кути очей, кінчики пальців, суглоби чи інші важливі частини тіла. Для їхнього виявлення застосовуються спеціалізовані алгоритми. Наприклад, алгоритм Viola-Jones часто використовується для виявлення обличчя, тоді як більш сучасні підходи, засновані на глибоких нейронних мережах, такі як OpenPose, дозволяють визначати десятки точок на тілі людини.

Етап **відстеження руху** зосереджується на аналізі змін положення ключових точок у часі. Це дає змогу зрозуміти, як рухається тіло людини. Одним із найпопулярніших методів для цього є алгоритм Optical Flow, який аналізує зміни яскравості пікселів між послідовними кадрами. Завдяки цьому алгоритму система може визначити траєкторію руху, навіть якщо ключові точки змінюють своє положення поступово чи нелінійно.

Завершальним етапом є **інтерпретація рухів**. На цьому етапі вступає в дію машинне навчання. Для того щоб система могла розпізнавати нові рухи, її

навчають на великому наборі прикладів, які містять різноманітні дії та пози. Моделі глибокого навчання, такі як згорткові нейронні мережі (CNN), або машинного навчання такі як SVM, здатні аналізувати та класифікувати рухи в реальному часі. Завдяки цьому система може не лише визначити, що виконує людина, але й адаптуватися до її індивідуальних особливостей.

Усі ці етапи разом забезпечують комплексний аналіз рухів, перетворюючи візуальні дані в інструмент для інтерактивної взаємодії між користувачем і комп'ютером. Від початкового захоплення зображень до складної інтерпретації дій, системи розпізнавання рухів є прикладом того, як сучасні технології можуть розширювати можливості взаємодії людини з цифровим світом.

Для реалізації такої системи було обрано фреймворк MediaPipe. MediaPipe - це фреймворк з відкритим кодом, розроблений Google, який надає готові рішення для багатьох завдань комп'ютерного зору, включаючи виявлення обличчя, відстеження рук та позу. Відстеження ключових точок здійснюється за допомогою моделі MediaPipe Hands, яка базується на BlazePalm, спеціально навченою для виявлення рук. Для визначення жестів використовується нейронна мережа, навчена за допомогою TensorFlow/Keras, яка класифікує положення ключових точок. Точність системи оцінюється на спеціально зібраному наборі даних жестів і досягає 95% .

Запропонована архітектура системи забезпечує ефективне розпізнавання рухів рук за допомогою сучасних технологій. Використання MediaPipe Hands дозволяє створити систему, яка працює швидко, точно і легко адаптується до різних завдань.

2.3 Алгоритми розпізнавання та аналізу рухів тіла за допомогою штучного інтелекту

Розпізнавання рухів тіла є однією з ключових задач комп'ютерного зору та штучного інтелекту, оскільки має широкий спектр застосувань: від розуміння жестової мови до управління пристроями за допомогою жестів. Основна складність розпізнавання полягає в тому, що руки та пальці часто

перекривають одна одну, мають низький контраст і можуть бути частково приховані. У цьому контексті використання MediaPipe Hands дає змогу значно покращити точність і швидкість аналізу.

Ключовою складовою MediaPipe є BlazePalm – модель, що виконує точне та швидке виявлення долонь. BlazePalm вирізняється здатністю розпізнавати долоні в складних умовах, таких як різний масштаб, перекриття або відсутність чітких візуальних орієнтирів. Особливістю цієї моделі є використання квадратних рамок, які забезпечують швидкість і точність, а також зниження обчислювальної складності завдяки оптимізації розмірів об'єктів. "BlazePalm досягає середньої точності 95.7%, що є значним показником у порівнянні з іншими підходами" – зазначено google reseacrh.[21] Після виявлення долоні використовується Hand Landmark Model, яка визначає координати 21 ключової точки руки у тривимірному просторі. Ця модель була навчена на поєднанні реальних та синтетичних даних, що забезпечило високу стійкість до складних сценаріїв, таких як часткове перекриття пальців або само перекриття руки.

Графік (рис. 2.2) наочно демонструє процес виявлення та відстеження рук за допомогою MediaPipe. Отриманий з камери відео потік проходить через низку послідовних етапів обробки. Спочатку, для оптимізації процесу, система перевіряє, чи є необхідність у повному аналізі кадру. Якщо рука була виявлена на попередньому кадрі і не зникла з поля зору, то система може пропустити деякі проміжні етапи обробки, що значно прискорює роботу.

Якщо ж рука не була виявлена або зникла, система запускає повний цикл обробки. Спочатку на зображенні шукається область, яка найбільш схожа на долонь. Після виявлення долоні, виділяється область навколо неї, і на цій під області запускається більш точна модель, яка визначає координати 21 ключової точки на руці. Ці точки відповідають кінчикам пальців, суглобам та іншим характерним точкам. Отримані координати перетворюються в зручний формат для подальшої обробки та візуалізації. На фінальному етапі на вихідне зображення наносяться лінії, що з'єднують ключові точки, що дозволяє

візуально оцінити результат роботи системи. Завдяки стрімкому розвитку технологій комп'ютерного зору, такі системи стають все більш доступними, забезпечуючи інтеграцію штучного інтелекту у повсякденне життя. Удосконалення моделей, таких як MediaPipe Hands, дозволяє розширювати можливості розпізнавання жестів, роблячи його ще більш точним і швидким.

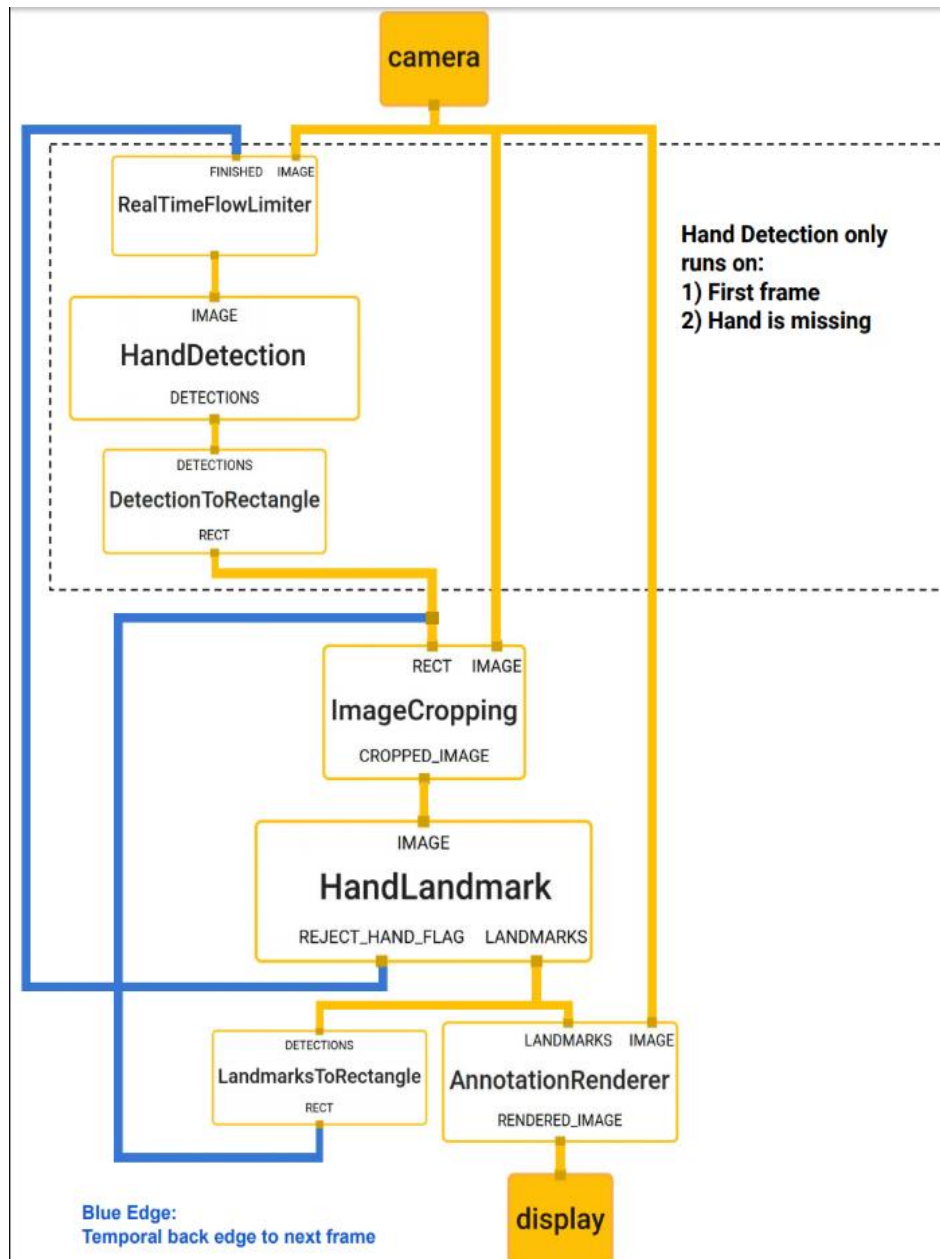


Рис. 2.2. Алгоритм розпізнавання руху руки за допомогою Mediapipe
Ключові етапи:

1. Отримання відео потоку з камери.
2. Перевірка необхідності повної обробки кадру.
3. Виявлення області долоні на зображенні (якщо необхідно).

4. Вирізання області з долонею.
5. Визначення координат ключових точок на руці.
6. Перетворення координат у зручний формат.
7. Візуалізація результатів на зображенні

Підсумовуючи цей графік демонструє алгоритм розпізнавання руки за допомогою MediaPipe Hands. Він показує послідовність обробки зображення з камери, виявлення руки, визначення ключових точок та їх візуалізацію.

Було проведено тестування використання інструменту MediaPipe для різних задач. Для ілюстрації я продемонструю результати роботи трьох моделей: виявлення рук, які визначають положення пальців; виявлення ключових точок обличчя, що дозволяє аналізувати міміку; і визначення пози тіла, що використовується в аналізі рухів. Першим було обрано оглянути роботу саме виявлення руки з ключовими точками на ній (рис. 2.3).

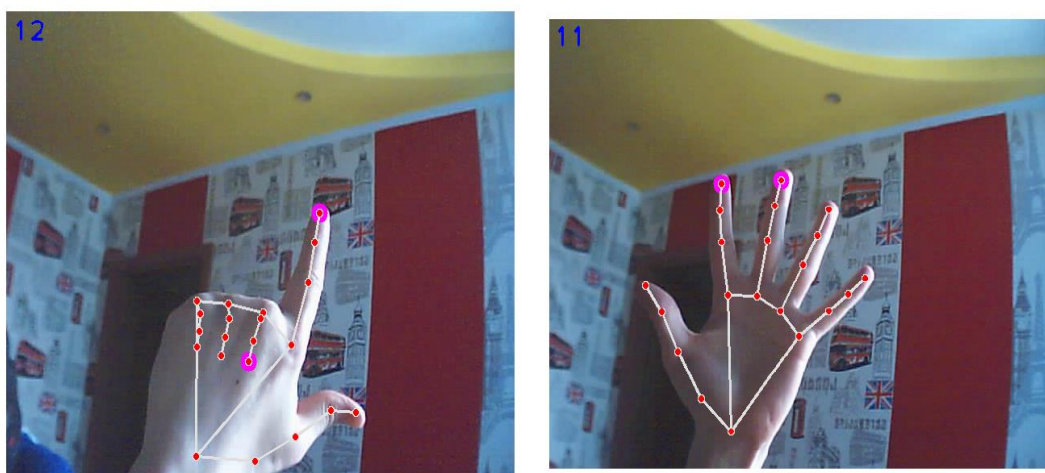


Рис. 2.3. Приклад виявлення рук.

Проводиться пошук 3 значень (x,y,z) у просторі для кожної з 21 точки руки. На зображенні показано дві руки, кожна з яких має 21 ключову точку, виділену кольоровими колами. Модель успішно виявляє точки навіть у випадках, коли пальці частково перекриваються, як видно на лівому зображенні. Точки візуалізуються з високою точністю, що дозволяє точно визначити положення пальців. Варто зазначити, що швидкість обробки становить приблизно 10-30 кадрів в секунду, що робить її придатною для

інтерактивних додатків. На зображеннях можна помітити виділені точки на двох пальцях і, що дуже важливо, точки прихованих пальців визначено надзвичайно точно (рис. 2.4).

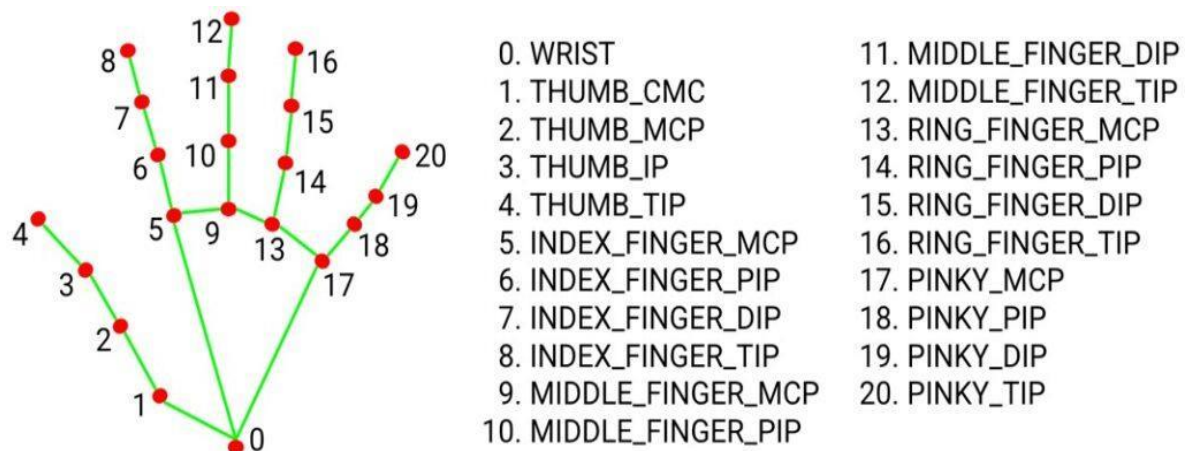


Рис. 2.4. Приклад візуалізації точок руки.

Розпізнавання обличчя у цій програмі працює за допомогою алгоритмів машинного навчання, інтегрованих у бібліотеку MediaPipe. Камера захоплює відео потік у режимі реального часу, і кожен кадр передається на обробку. Спочатку зображення перетворюється у потрібний формат, щоб його могли аналізувати моделі розпізнавання. Далі нейронна мережа шукає ключові точки на обличчі, руках та тілі, використовуючи спеціально навчені моделі для визначення цих структур.

Модель для обличчя знаходить 468 ключових точок, які відповідають різним частинам — очам, носу, губам, щокам тощо. Для рук виявляються точки на пальцях і долонях, а для тіла — основні позиційні маркери, такі як положення плечей, колін і стегон. Після цього знайдені точки передаються на візуалізацію, де на вихідному зображенні малюються лінії та з'єднання, які відображають розташування розпізнаних елементів. Водночас у процесі обчислюється FPS, яка показує, наскільки швидко система може обробляти відео потік. Результат у вигляді зображення з накладеними графічними елементами виводиться у вікно. Це дозволяє в реальному часі спостерігати за тим, як працює алгоритм, і бачити розпізнані елементи на обличчі та тілі (рис. 2.5).

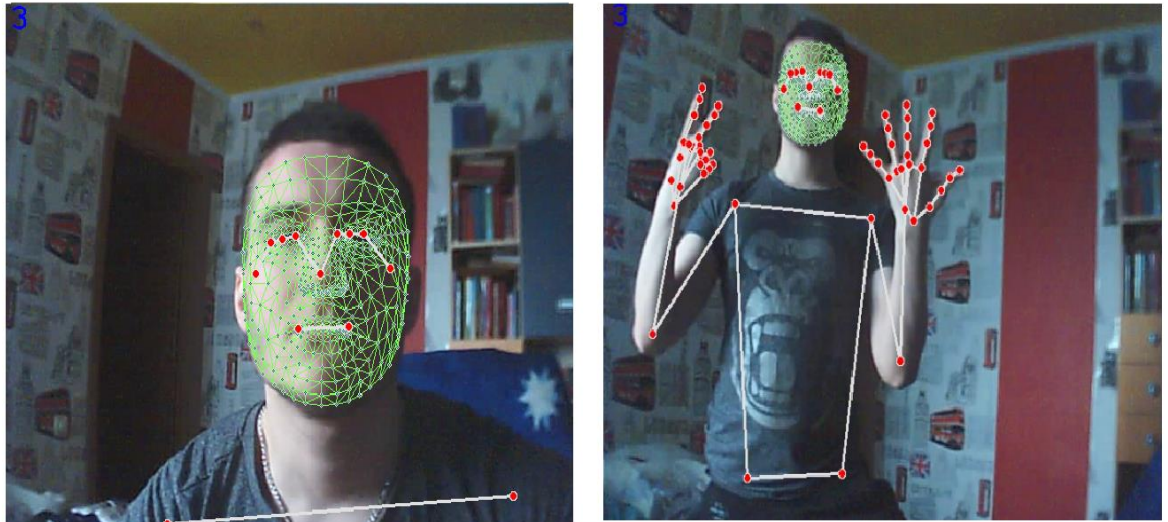


Рис. 2.5. Приклад розпізнавання обличчя та тіла.

Наступне зображення демонструє результати роботи алгоритму MediaPipe, який спеціалізується на виявленні та відстеженні ключових точок на людському тілі в реальному часі. Ці точки, представлені червоними кружечками, відповідають певним анатомічним орієнтирам, таким як очі, ніс, роти, суглоби та кінцівки (рис. 2.6).

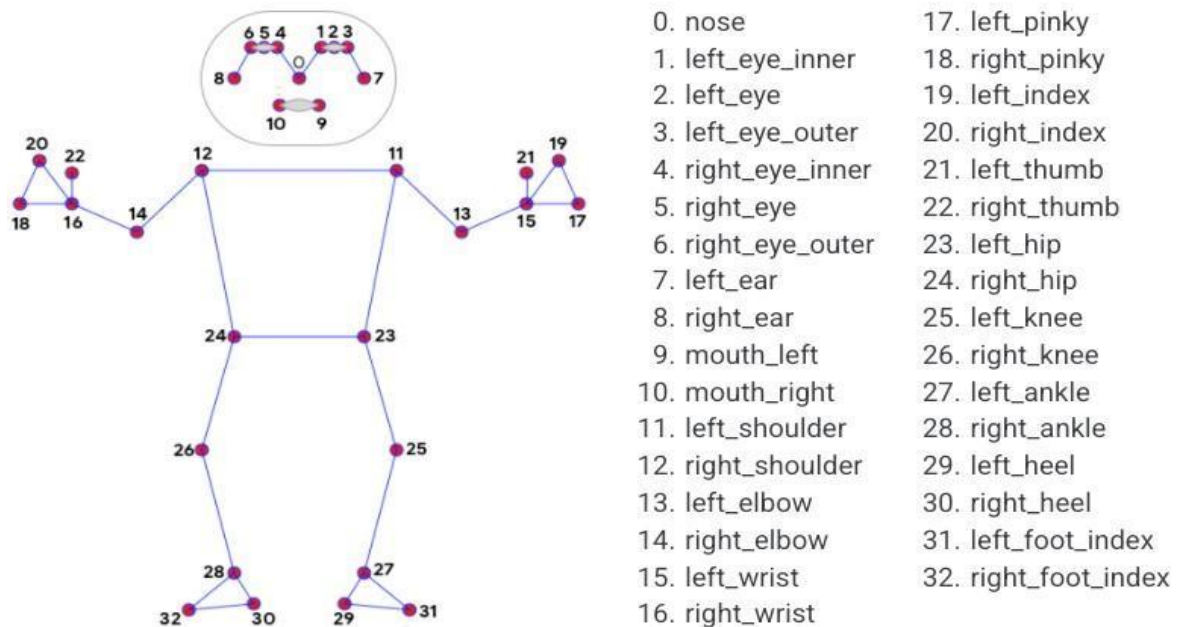


Рис. 2.6. Приклад візуалізації точок тіла.

Обрані алгоритми розпізнавання рухів, зокрема на базі MediaPipe і BlazePalm, демонструють високу точність і швидкість роботи навіть в умовах низької контрастності та часткового перекриття пальців. Вони забезпечують

можливість реального часу відстеження та інтерпретації рухів, що є важливим кроком до створення інтуїтивних інтерфейсів для людей із фізичними обмеженнями. Такий підхід дозволяє адаптувати технологію до індивідуальних потреб користувачів, сприяючи їхній соціальній адаптації та інтеграції в цифрове середовище.

2.4 Проектування інтерфейсу користувача для осіб з фізичними обмеженнями

Основна мета такого інтерфейсу полягає в тому, щоб забезпечити простоту використання без необхідності освоювати складні технічні аспекти. Наприклад, важливо, щоб усі елементи інтерфейсу були зрозумілими та добре структурованими. Для цього необхідно уникати перевантаження екрану зайвими елементами та забезпечити чіткі інструкції, зрозумілі навіть для тих, хто вперше стикається з такою системою.

Важливим елементом є підтримка альтернативних способів взаємодії. Окрім миші та клавіатури, користувачі повинні мати можливість керувати системою за допомогою голосових команд, сенсорних екранів або навіть спеціальних пристроїв, таких як айтрекери чи адаптивні джойстики. Це особливо важливо для тих, хто має обмежену рухливість або повністю не може користуватися стандартними пристроями введення.

Доступність контенту також є ключовим аспектом. Інтерфейс повинен включати можливості для людей із порушенням фізичних обмежень. Це може бути реалізовано через альтернативний текст для зображень, субтитри для відео або можливість збільшення шрифтів. Також слід забезпечити високу контрастність кольорів, що полегшує сприйняття контенту людьми із порушеннями зору.

Структура інтерфейсу повинна бути логічною та передбачуваною, щоб користувачі могли швидко знайти потрібну інформацію або функцію. Використання заголовків, підзаголовків і списків допомагає організувати контент і зробити його доступним навіть для користувачів, які використовують екранні читалки.

При цьому важливо уникати надмірного обмеження часу на виконання завдань. Наприклад, таймери повинні бути налаштовані так, щоб люди з обмеженими можливостями мали достатньо часу для виконання необхідних дій, не відчуючи тиску.

Декілька прикладів інклюзивних інтерфейсів допомагають зрозуміти, як ці принципи застосовуються на практиці. Наприклад, Google Maps пропонує маршрути з урахуванням потреб людей з обмеженими можливостями, включаючи інформацію про пандуси та доступні входи. Платформа Airbnb надає фільтри для пошуку житла, яке відповідає таким потребам, наприклад, квартири з ліфтами або зручними під'їздами. Продукти Apple, такі як функція VoiceOver, дозволяють людям із порушенням зору користуватися пристроями за допомогою голосового управління.

Створення інклюзивного інтерфейсу не лише сприяє соціальній рівності, але й робить продукт доступним для ширшої аудиторії, забезпечуючи позитивний досвід для всіх користувачів. Інклюзивний дизайн стає стандартом у цифровому середовищі, сприяючи створенню справедливого світу, де технології доступні кожному.

Висновки до другого розділу

Другий розділ дослідження був присвячений розробці системи розпізнавання та аналізу рухів тіла, яка допомагає особам із фізичними обмеженнями адаптуватися до використання персональних комп'ютерів. Було визначено основні вимоги до системи, включаючи її адаптивність, точність і зручність для користувача. У розділі детально розглянуто архітектуру системи, вибір компонентів та технологій, що впливають на ефективність роботи алгоритмів розпізнавання рухів.

Запропонована система базується на використанні ІІІ та алгоритмів МН, що дозволяють аналізувати рухи користувача у реальному часі. Значна увага приділена оптимізації програмних засобів для забезпечення високої точності розпізнавання.

3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Розробка програми та інтерфейсу для розпізнавання та аналізу рухів тіла

На початковому етапі було створено датасет за допомогою OpenCV[30]. Було підключено наступні бібліотеки, налаштована камера та параметри.

```
import cv2
from cvzone.HandTrackingModule import HandDetector
import numpy as np
import math
import time

cap = cv2.VideoCapture(0)
detector = HandDetector(maxHands=1)

offset = 20
imgSize = 300
folder = "DataV2/"
counter = 0
```

В подальшому був зроблений основний цикл обробки та перевірка чи знайдена рука за допомогою HandTrackingModule та створення білого фону для обрізання зображення з рукою

```
while True:
    success, img = cap.read()
    hands, img = detector.findHands(img)

    if hands:
        hand = hands[0]
        x, y, w, h = hand['bbox']

        imgWhite = np.ones((imgSize, imgSize, 3), np.uint8) * 255
        imgCrop = img[y - offset:y + h + offset, x - offset:x + w + offset]
```

Далі пройшло масштабування зображення та його відображення у трьох різних вікнах а саме основного, обрізаного та кінцевого. Три вікна було створено для зручності створювання фотографій для датасету (рис. 3.1).

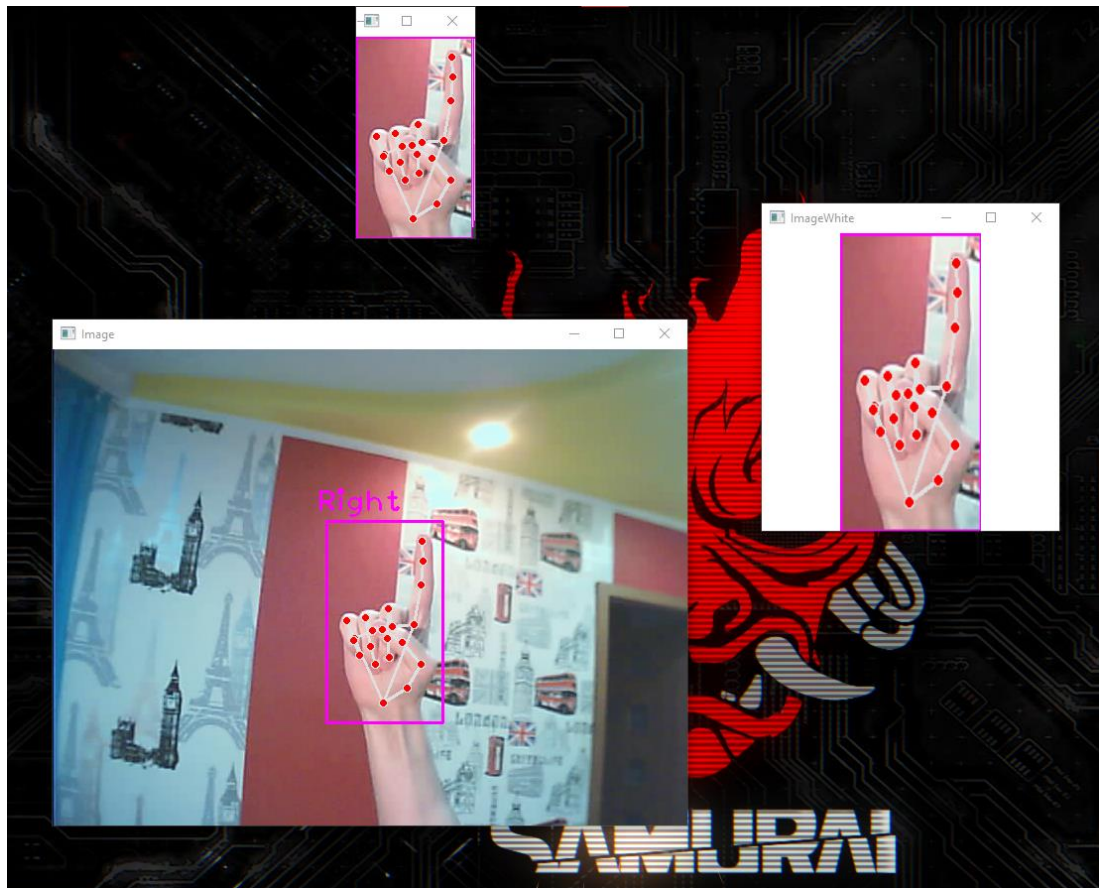


Рис. 3.1. Приклад обробки зображень перед зберіганням

$\text{aspectRatio} = h / w$

```

if aspectRatio > 1:
    k = imgSize / h
    wCal = math.ceil(k * w)
    imgResize = cv2.resize(imgCrop, (wCal, imgSize))
    wGap = math.ceil((imgSize - wCal) / 2)
    imgWhite[:, wGap:wCal + wGap] = imgResize
else:
    k = imgSize / w
    hCal = math.ceil(k * h)
    imgResize = cv2.resize(imgCrop, (imgSize, hCal))
    hGap = math.ceil((imgSize - hCal) / 2)
    imgWhite[hGap:hCal + hGap, :] = imgResize

cv2.imshow("ImageCrop", imgCrop)
cv2.imshow("ImageWhite", imgWhite)

```

Далі при натисканні клавіші "s" створюється фотографія з камери та зберігається у папці "DataV2". Зображення попередньо сортуються по папках, щоб підготувати їх для навчання нейромережі.

```

cv2.imshow("Image", img)
if key == ord("s"):
    counter += 1
    cv2.imwrite(f'{folder}/Image_{time.time()}.jpg', imgWhite)
    print(counter)

if key == 27: # 27 is the ESC key
    break

cap.release()
cv2.destroyAllWindows()

```

Після чого було проведено обробку та візуалізацію зображень для їх підготовки до навчання CNN за допомогою TensorFlow (рис. 3.2).

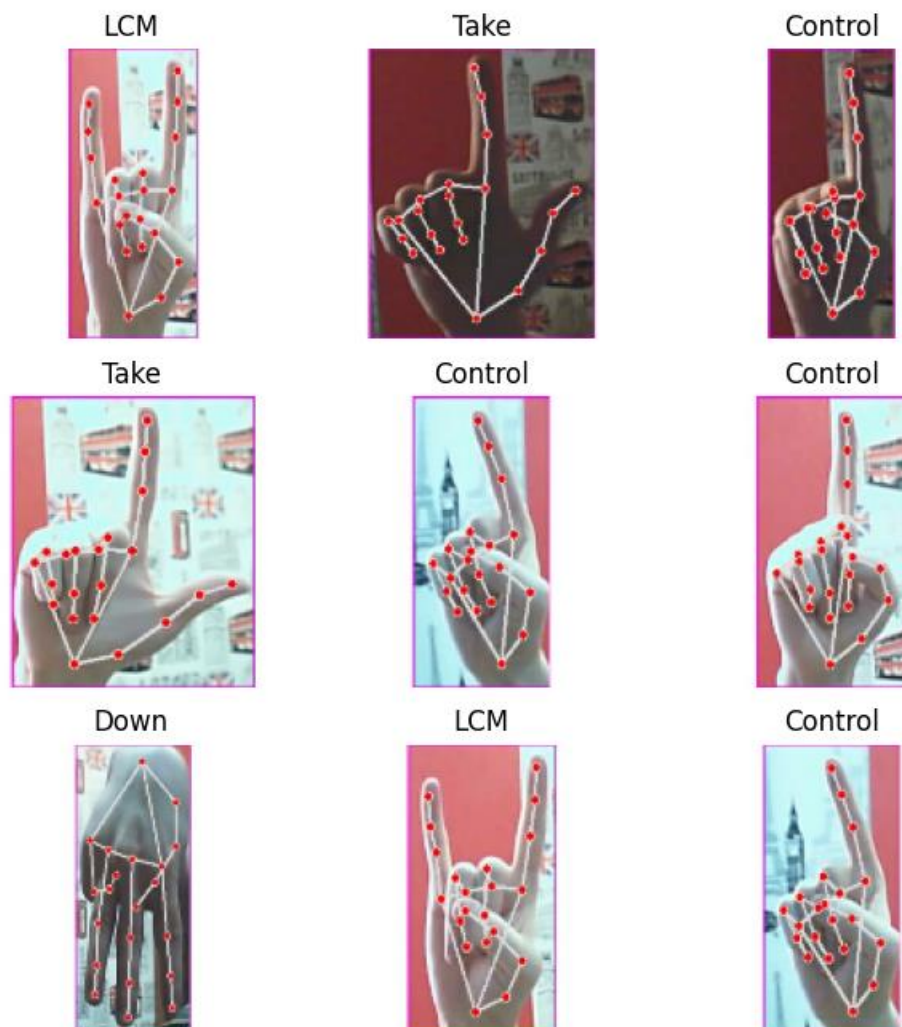


Рис. 3.2. Приклад обробки зображень перед обробкою

Після цього виконано тренування моделі, у результаті чого отримано файл M5_model.h5. Модель навчалась на 50 епохах, зображень 224x224 , розмір пакету становив 16 зображень. На основі цієї моделі розпочалася розробка програми для трекінгу та аналізу положення руки.

На початку імпортуються бібліотеки, необхідні для роботи коду

```
import cv2
import numpy as np
import time
import pyautogui
from cvzone.HandTrackingModule import HandDetector
from cvzone.ClassificationModule import Classifier
import math
```

Потім було ініціалізовано камеру та її налаштування

```
cap = cv2.VideoCapture(0)
cap.set(cv2.CAP_PROP_FRAME_WIDTH, 640)
cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)

cv2.namedWindow("Hand Recognition", cv2.WINDOW_NORMAL)
cv2.resizeWindow("Hand Recognition", 640, 480)
cv2.setWindowProperty("Hand Recognition", cv2.WND_PROP_TOPMOST, 1)
```

Після чого було додане виявлення руки та класифікація жестів

```
detector = HandDetector(maxHands=1)

classifier = Classifier("Model5/keras_model.h5", "Model5/labels.txt")

offset = 20
imgSize = 300
labels = ["Control", "Down", "LCM", "RCM", "Take", "Up"]
if confidence > 80:
    if label == "Control":
        lmList = hand["lmList"]
        if lmList:
            index_finger = lmList[8]
            screen_x = np.interp(index_finger[0], [frameR, wCam - frameR], [0,
pyautogui.size()[0]])
            screen_y = np.interp(index_finger[1], [frameR, hCam - frameR], [0,
pyautogui.size()[1]])
            pyautogui.moveTo(screen_x, screen_y)

    elif label == "Down":
        pyautogui.scroll(-100)
```

```
elif label == "Up":  
    pyautogui.scroll(100)  
  
elif label == "LCM":  
    pyautogui.click(button='left', clicks=2)  
  
elif label == "Take":  
    pyautogui.mouseDown(button='left')
```

3.2 Результати роботи програми та її функціональні можливості

Запускаємо програму в нас вмикається веб камера і система одразу починає сканувати зображення на наявність руки. Поки система проводить пошук руки FPS варіюється між 20-30 (рис. 3.3).

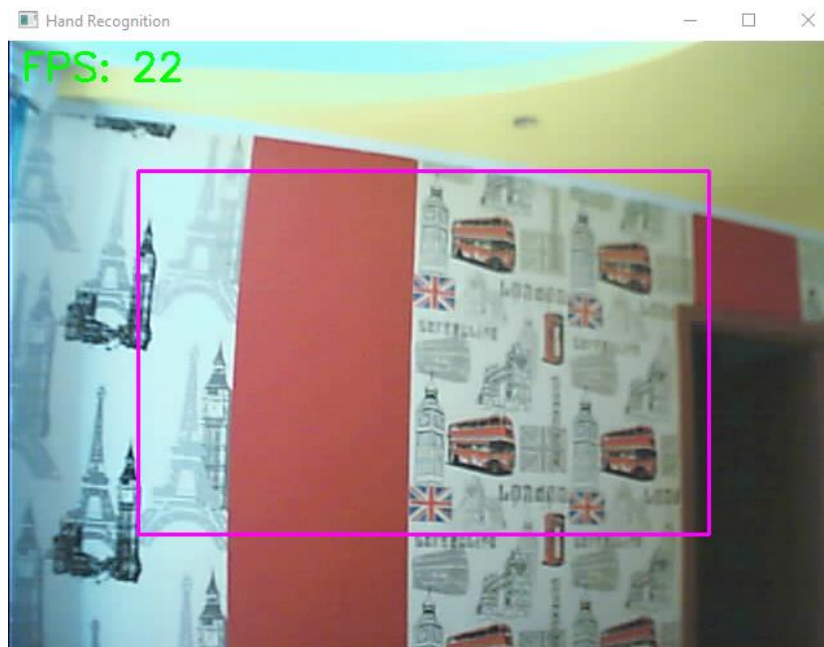


Рис. 3.3. демонстрація програми

Далі підведемо руку до камери щоб вона зробила її модель та визначила положення як можна помітити FPS різко зменшився до 4 через те що почалась обробка по ключовим точкам інша модель аналізує положення руки і система починає виконувати дії в залежності від цього (рис. 3.4).

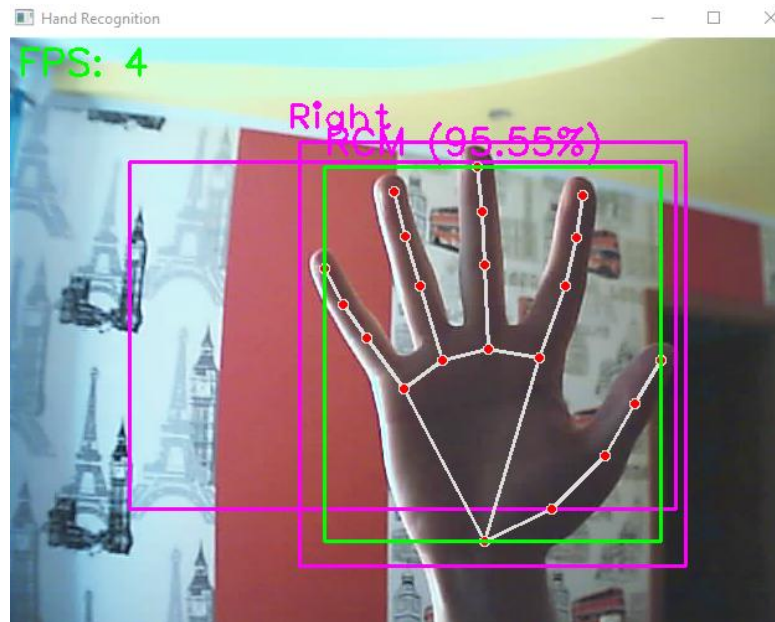


Рис. 3.4. демонстрація програми

Потім перейдемо в режим переміщення мишки а саме підіймемо вказівний палець та перевіримо чи переміщується курсор (рис. 3.5).

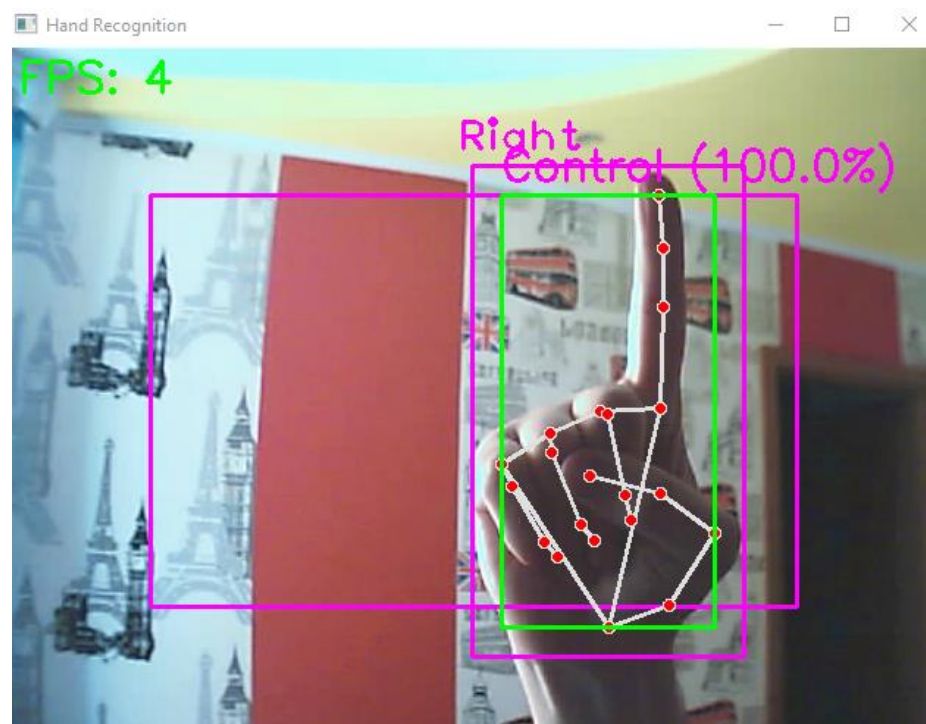


Рис. 3.5. демонстрація програми

Після перевірки того що мишка рухається перевіримо інші функції а саме подвійний клік мишкою за допомогою підняття вказівного пальця та мізинця та спробуємо відкрити файл на робочому столі (рис. 3.6).

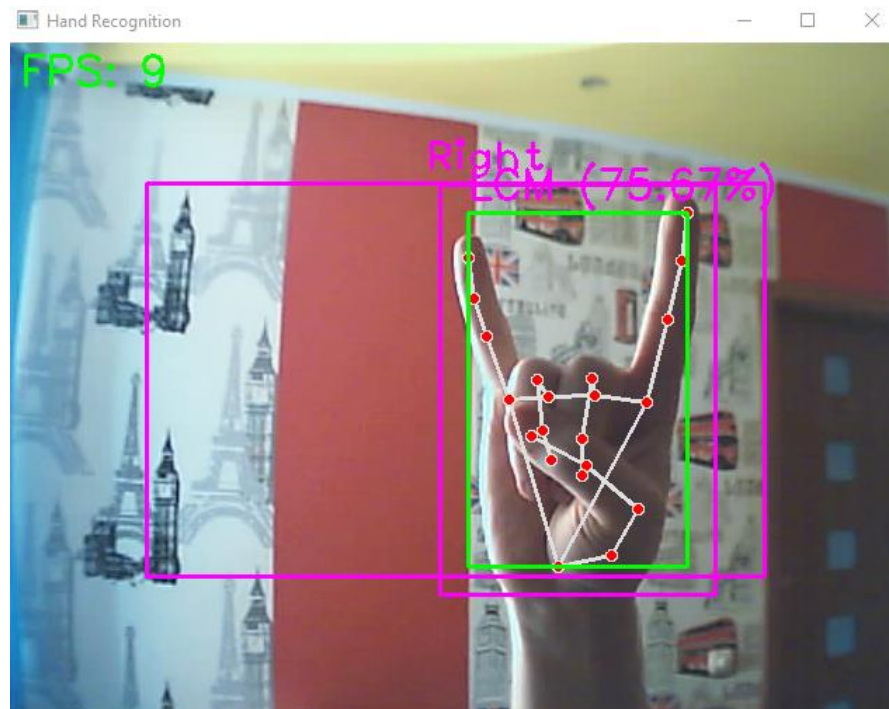


Рис. 3.6. демонстрація програми

Далі перевіряємо чи працює утримання лівої кнопки миші через підняття вказівного та великого пальців було переміщено декілька файлів (рис. 3.7).

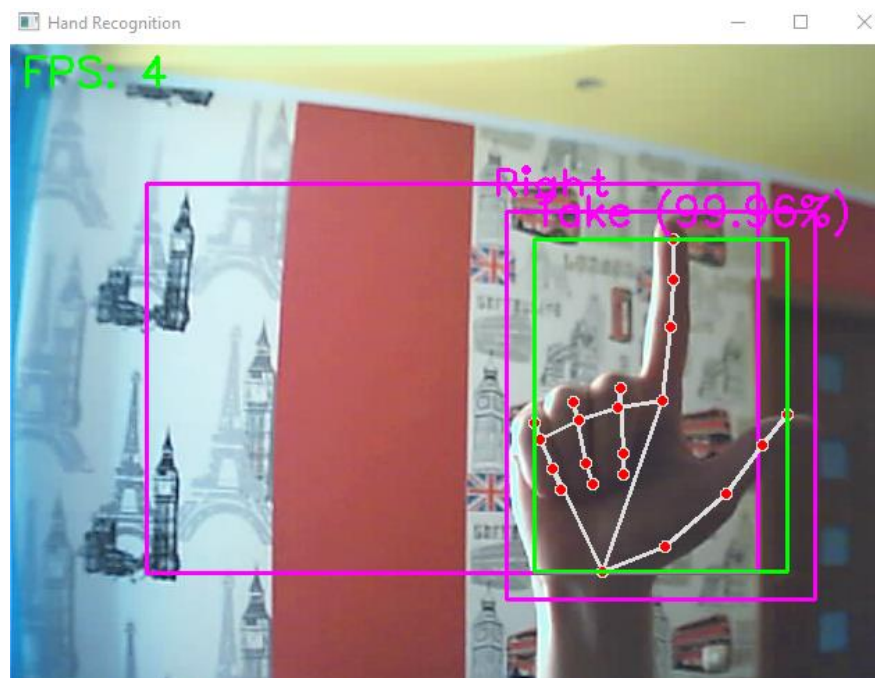


Рис. 3.7. демонстрація програми

Наступним було реалізовано гортання вгору за допомогою підняття трьох пальців в гору (рис. 3.8).

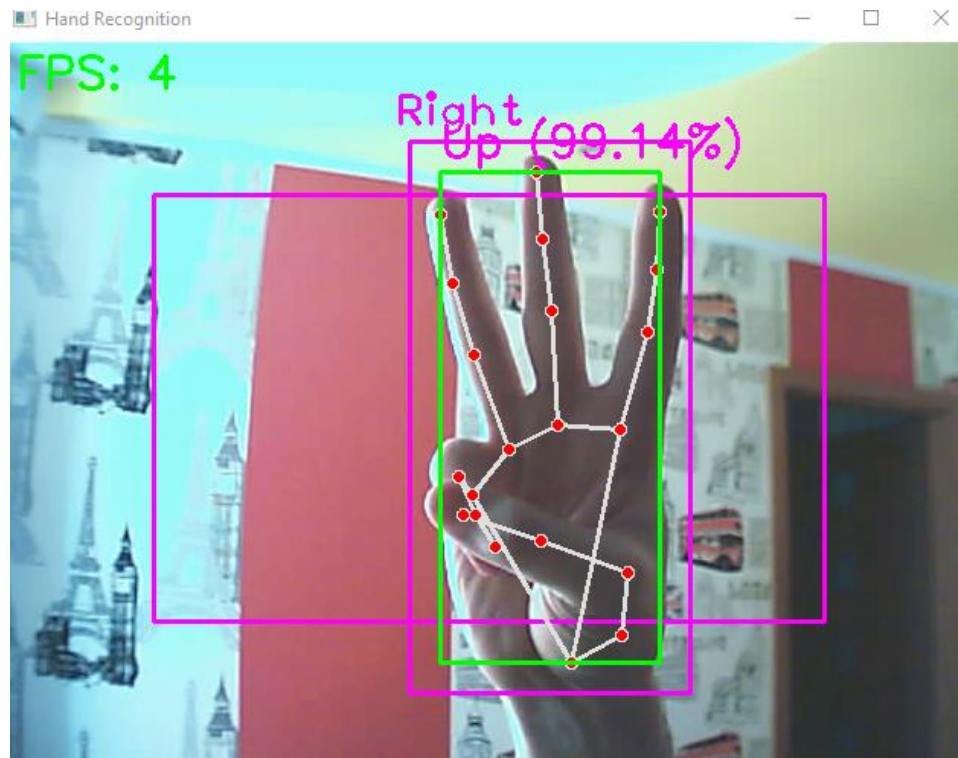


Рис. 3.8. демонстрація програми

Наступним при опусканні трьох пальців вниз було реалізовано гортання вниз (рис. 3.9).

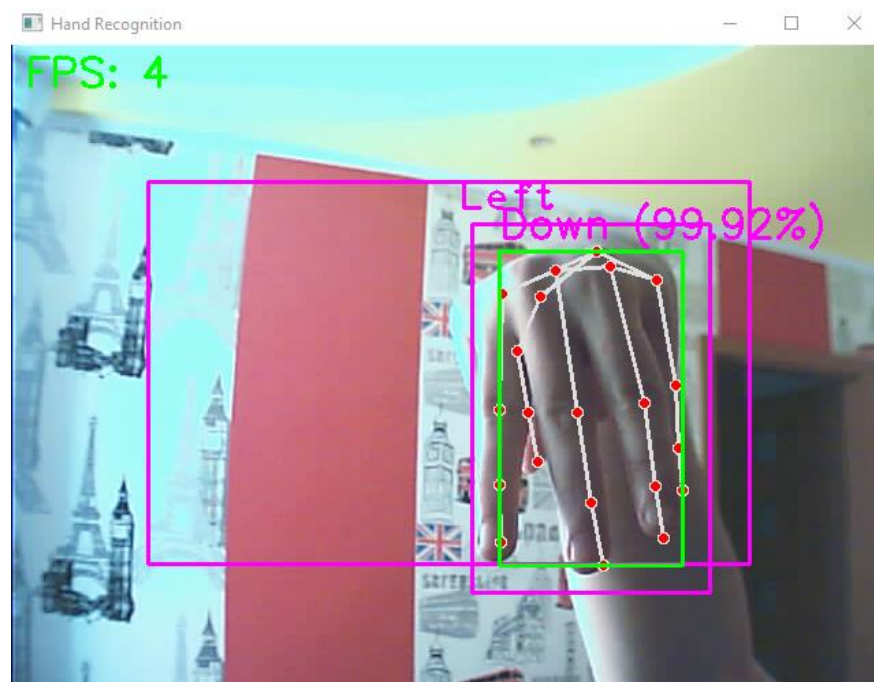


Рис. 3.9. демонстрація програми

Бібліотека Mediapipe використовується для обробки зображень та трекінгу положення рук у режимі реального часу. Вона базується на використанні центрального процесора (CPU) для виконання складних

математичних обчислень, необхідних для аналізу ключових точок на зображенні. Проте через обмежену продуктивність системи, продуктивність програми була суттєво обмежена, швидкість роботи програми становила лише 3-6 кадрів на секунду.

3.3 Тестування роботи програми

Проведемо тестування при різних умовах освітленості.

На (рис. 3.10) представлено результати тестування програми за умов низької освітленості. Як видно, якість обробки зображення значно погіршується, спостерігаються шуми, розмиття контурів та зниження контрастності. Незважаючи на це програма може обробити долонь та виконувати свої функції.

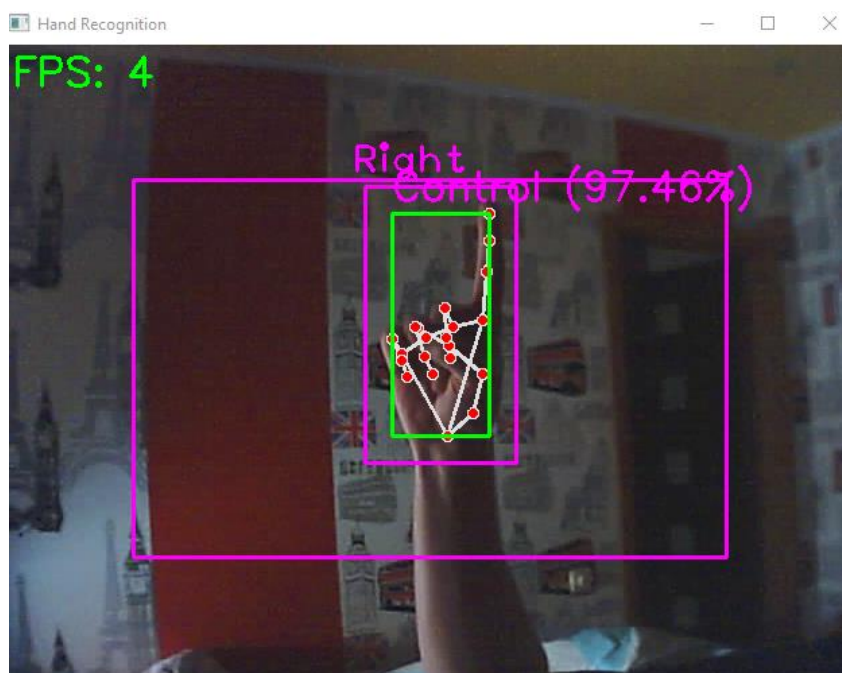


Рис. 3.10. демонстрація програми при низькій освітленості

На (рис. 3.11) демонструється робота програми за умов надмірної освітленості. Незважаючи на високий рівень яскравості, програма продовжує функціонувати, однак спостерігається нестабільність у роботі алгоритмів обробки зображень. Це проявляється у вигляді випадкових збоїв та некоректної обробки ключових точок.

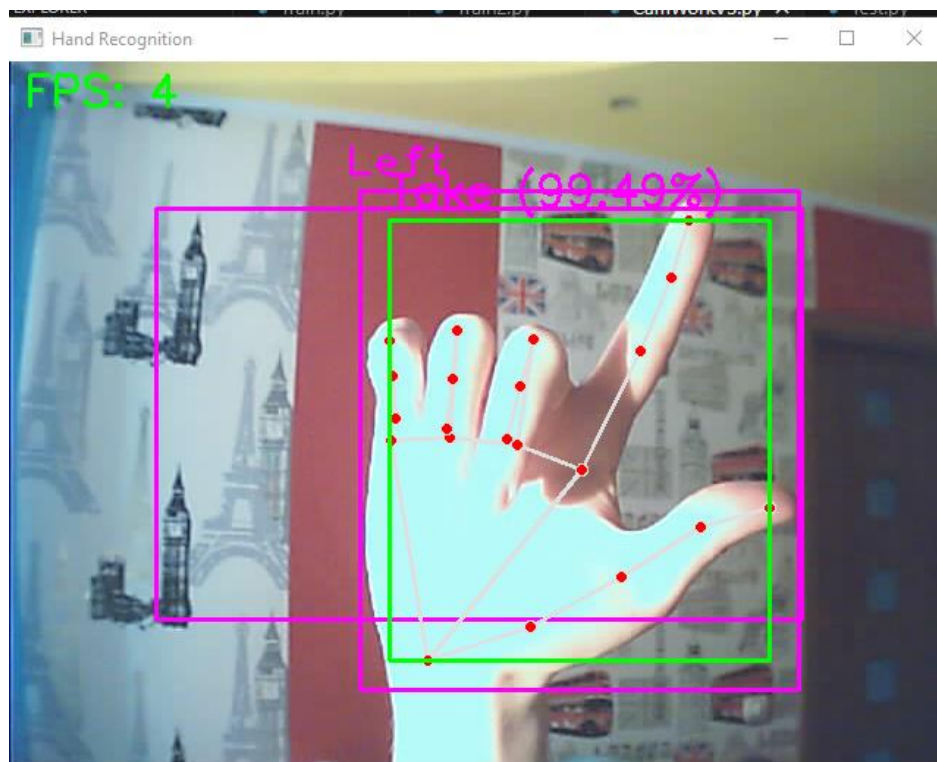


Рис. 3.11. демонстрація програми при надмірній освітленості

На (рис. 3.12) показано, що коли відстань до об'єкта, який відстежує програма, перевищує метр, починаються проблеми з точністю розпізнавання рухів. Програма може неправильно інтерпретувати рухи рук або навіть зовсім їх не помічати, особливо якщо рука тримається близько до тіла.

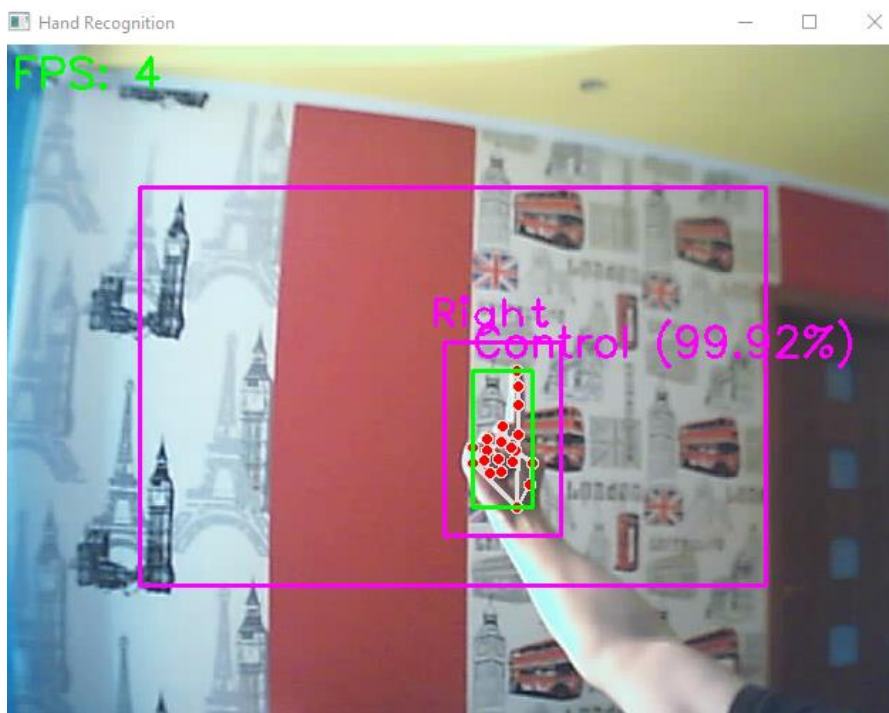


Рис. 3.12. демонстрація програми при віддаленій руці

На (рис. 3.13) показано, що програма працює стабільно на відстані близько 40-50 сантиметрів від об'єкта відстеження. На такій відстані система точно розпізнає рухи і жести, забезпечуючи надійну роботу. Якщо рука знаходиться ближче тоді модель не вірно починає розпізнавати положення долоні.

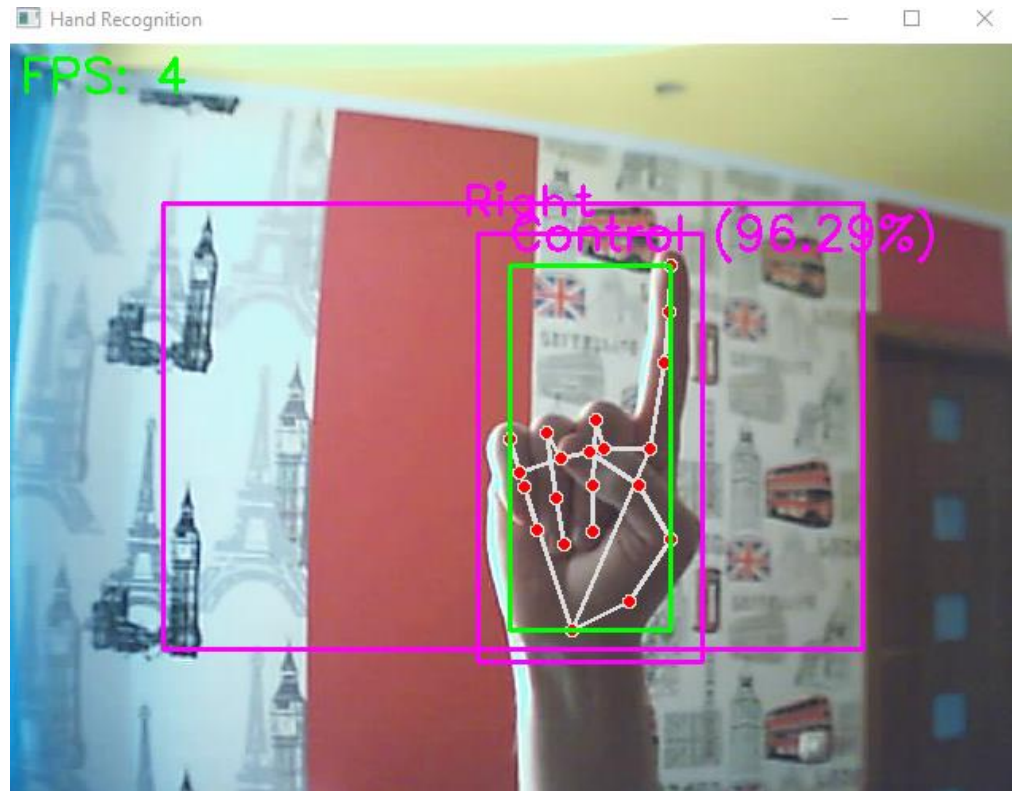


Рис. 3.13. демонстрація програми при близькій дистанції

Проведені тести показали, що розроблена система розпізнавання рухів руки для управління курсором комп'ютера є працездатною та здатною виконувати основні функції. На базовому обладнанні (процесор i5-7400) частота кадрів становить 3–6 FPS. Така швидкість є достатньою для демонстраційних цілей, однак може викликати певний дискомфорт під час тривалої взаємодії через затримки між рухами руки та реакцією курсора.

Система успішно розпізнає жести, включаючи рухи курсора, кліки та гортання. Проте точність розпізнавання залежить від умов освітлення: за нормального освітлення вона складає близько 90%, у той час як за низького чи занадто яскравого освітлення точність знижується через чутливість алгоритму

до шумів та контрасту. Основною причиною обмеження продуктивності є значні обчислювальні ресурси, необхідні для обробки відео та виконання алгоритмів розпізнавання. Це можна покращити шляхом створення більш оптимізованої нейронної моделі наприклад на більшому датасеті та провівши навчання на більшій кількості епох.

3.4 Рекомендації та перспективи щодо подальшого розвитку системи

Робота над системою розпізнавання жестів для людей з обмеженими можливостями має значний потенціал для подальшого розвитку. Перспективним напрямком є збільшення обсягу даних та ефективності навчання для власної нейронної мережі, спеціально адаптованої під потреби користувачів з різними фізичними обмеженнями. Навчання такої мережі на великих, різноманітних датасетах дозволить підвищити точність розпізнавання жестів і адаптувати систему до індивідуальних особливостей користувачів.

Оптимізація коду системи є ще одним важливим аспектом. Це дозволить підвищити швидкість роботи програми, зменшити споживання ресурсів комп'ютера та зробити систему більш стабільною. Додатково можна розглянути можливість вбудовування функції голосового керування, що зробить користування системою ще зручнішим для людей з обмеженими можливостями.

Важливо також розглянути питання відкритого коду. Зробивши систему доступною для всіх, ми зможемо залучити більшу кількість розробників, які зможуть вносити свої пропозиції та покращення. Це дозволить створити більш універсальний і ефективний інструмент, який допоможе багатьом людям з обмеженими можливостями.

Таким чином, подальший розвиток системи розпізнавання жестів може мати значний соціальний ефект, покращуючи якість життя людей з обмеженими можливостями та надаючи їм більше самостійності.

Висновки до третього розділу

У третьому розділі було здійснено практичну реалізацію системи розпізнавання та аналізу рухів тіла, що базується на методах ІІІ. Було розроблено програмне забезпечення, яке інтегрує сучасні алгоритми комп'ютерного зору та МН для точного визначення положення тіла та жестів користувачів.

Запропонована система базується на використанні ІІІ та алгоритмів МН, що дозволяють аналізувати рухи користувача у реальному часі. Значна увага приділена оптимізації програмних засобів для забезпечення високої точності розпізнавання та створені власної моделі. Тестування розробленого рішення підтвердило його ефективність у розпізнаванні рухів та жестів, що відкриває перспективи подальшого вдосконалення системи.

ВИСНОВКИ

У даній кваліфікаційній роботі було виконано повний цикл розробки системи розпізнавання рухів людського тіла із застосуванням методів штучного інтелекту, зокрема нейронних мереж. Розглянуто актуальність проблеми, пов'язаної з аналізом і розпізнаванням рухів, яка має широкий спектр застосувань у спортивній аналітиці, медицині, ігровій індустрії та системах безпеки. Проведено аналітичний огляд існуючих алгоритмів, методів і програмних засобів для розпізнавання рухів, зокрема бібліотек OpenCV та Mediapipe, що дозволило визначити їх сильні та слабкі сторони.

Запропонований алгоритм розробки базувався на сучасних архітектурах нейронних мереж із використанням ключових точок для визначення положення частин тіла та власній моделі спеціально розробленій для визначення положення руки. У ході роботи були реалізовані функціональні можливості системи, що дозволяють розпізнавати базові жести руки для інтерактивного управління, такі як рух курсора, кліки миші та утримання кнопок а також гортання вгору та вниз.

Результати тестування підтвердили працездатність системи, однак виявили обмеження продуктивності при роботі на середньому апаратному забезпеченні (частота кадрів 3–6 FPS). Основними проблемами залишаються висока чутливість до умов освітлення та малий обсяг датасету при навчанні. У цілому, система показала задовільний рівень точності (70–90% залежно від умов) та стабільності розпізнавання жестів.

Робота має подальші перспективи вдосконалення, такі як використання більш потужних алгоритмів глибокого навчання, адаптація до змінних умов освітлення та впровадження індивідуальних налаштувань для кінцевого користувача. Запропоновані рішення можуть бути використані в прикладних розробках інтерактивних систем для різних галузей.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. González, R. C., Woods, R. E. – Digital Image Processing. – Pearson, 2017. – 1184 с.
2. Shotton, J., Fitzgibbon, A., Cook, M., et al. – Real-Time Human Pose Recognition in Parts from a Single Depth Image. – Microsoft Research, 2011. – 15 с.
3. Vollmer, M. – Infrared Thermal Imaging: Fundamentals, Research and Applications. – Wiley-VCH, 2017. – 612 с.
4. Woodman, O. J. – An Introduction to Inertial Navigation. – University of Cambridge, 2007. – 43 с.
5. Merletti, R., Parker, P. – Electromyography: Physiology, Engineering, and Non-Invasive Applications. – Wiley-IEEE Press, 2004. – 520 с.
6. Optical Flow documentation[Електронний ресурс] :– Режим доступу: https://docs.opencv.org/3.4/d4/dee/tutorial_optical_flow.html (дата звернення 18.12.2024).
7. IBM Documentation [Електронний ресурс] - Режим доступу: <https://www.ibm.com/docs/en> (дата звернення 25.12.2024).
8. Support Vector Machines (SVM) [Електронний ресурс] – Режим доступу: <https://scikit-learn.org/stable/modules/svm.html> (дата звернення 20.12.2024).
9. Convolutional Neural Network (CNN) [Електронний ресурс] – Режим доступу: https://d2l.ai/chapter_convolutional-neural-networks/ (дата звернення 20.12.2024).
10. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P. – Gradient-Based Learning Applied to Document Recognition. – Proceedings of the IEEE, 1998. – 2278–2324 с.
11. OpenPose [Електронний ресурс]: – Режим доступу: <https://cmu-perceptual-computing-lab.github.io/openpose/web/html/doc/index.html> (дата звернення 20.12.2024).
12. Cortes, C., Vapnik, V. – Support-Vector Networks. – Machine Learning, 1995. – 273–297 с.

13. Phinyomark, A., Nuidod, A., Phukpattaranont, P., Limsakul, C. – Feature Extraction and Reduction of Wavelet Transform Coefficients for EMG Pattern Classification. – Elektronika ir Elektrotechnika, 2012. – 27–36 с.
14. Каллан Р. Основні поняття нейронних мереж // Пер. з англ. – Видавництво М. Вільямса, 2011.
15. PyAutoGui Documentory [Електронний ресурс] – Режим доступу: <https://pyautogui.readthedocs.io/en/latest/> (дата звернення 20.12.2024).
16. Haykin, S. – Neural Networks and Learning Machines. – Pearson, 2008. – 936 с.
17. Осовський С. Нейронні мережі для обробки інформації: пер. з польської. // С. Осовський. – М.: Фінанси і статистика, 2012. – 344 с.
18. Дж. Лонг, Е. Шелхамер, Т. Даррелл: Повністю згорткові мережі для семантичної сегментації 2014 р.
19. Олаф Роннебергер, Філіп Фішер, Томас Брокс U-Net: згорткові мережі для сегментації біомедичних зображень.
20. Murphy K. P. – Machine Learning: A Probabilistic Perspective. – MIT Press, 2012. – 1104 с..
21. Real-Time Hand Tracking with MediaPipe [Електронний ресурс]– Режим доступу: <https://research.google/blog/on-device-real-time-hand-tracking-with-mediapipe/> (дата звернення 20.12.2024).
22. Brownlee J. – Introduction to Time Series Forecasting with Python. – Machine Learning Mastery, 2018. – 250 с.
23. Хабарлак К.С. – Методи класифікації та сегментації зображень на основі згорткових нейронних мереж. – 2020. – 180 с..
24. Бішоп К. – Моделі машинного навчання: Практичне керівництво. – Київ: Наукова думка, 2019. – 512 с.
25. Зайченко Ю.П. – Математичні методи та моделі в управлінні: Нейронні мережі. – Київ: НАУ, 2010. – 328 с.
26. Паттерсон Дж., Гібсон А. – Глибоке навчання: Поглиблене занурення в світ нейронних мереж. – Київ: Діалектика, 2018. – 400 с..

27. Goodfellow I., Bengio Y., Courville A. – Deep Learning. – MIT Press, 2016. – 800 с.
28. C. Tomasi, R. Manduchi Bilateral Filtering for Gray and Colour Images // Proceedings of the 1998 IEEE International Conference on Computer Vision, Bombay, India
29. Nielsen M. – Neural Networks and Deep Learning. – Determination Press, 2015. – 336 с.
30. Aggarwal C. C. – Neural Networks and Deep Learning: A Textbook. – Springer, 2018. – 497 с.
31. Python documentation [Електронний ресурс] – Режим доступу: <https://docs.python.org/3/> (дата звернення 17.12.2024).
32. TensorFlow documentation [Електронний ресурс] – Режим доступу: https://www.tensorflow.org/api_docs (дата звернення 20.12.2024).
33. Lecun Y., Bottou L., Bengio Y., Haffner P. – Gradient-Based Learning Applied to Document Recognition. – 1998.
34. Charniak E. – Introduction to Deep Learning. – MIT Press, 2019. – 187.
35. Matplotlib [Електронний ресурс] – Режим доступу: <https://matplotlib.org/stable/#matplotlib-release-documentation> вільний. - Загл. з екрану. (дата звернення 20.12.2024).
36. Rawat W., Wang Z. – Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review. – 2017.
37. Aggarwal J. K., Ryoo M. S. Human Activity Analysis: A Review // ACM Computing Surveys (CSUR). – 2011. – Т. 43, №3. – С. 16.
38. Redmon J., Farhadi A. YOLOv4: Optimal Speed and Accuracy of Object Detection // arXiv preprint arXiv:2004.10934. – 2020
39. Edunov, S., Ott, M., Auli, M., Grangier, D. Understanding Back- Translation at Scale. // EMNLP 2018
40. Graves A. Generating Sequences with Recurrent Neural Networks // ArXiv, abs/1308.0850, 2013

ДОДАТКИ

CamWorkV3.py

```
import cv2
import numpy as np
import time
import pyautogui
from cvzone.HandTrackingModule import HandDetector
from cvzone.ClassificationModule import Classifier
import math

cap = cv2.VideoCapture(0)
cap.set(cv2.CAP_PROP_FRAME_WIDTH, 640)
cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)

cv2.namedWindow("Hand Recognition", cv2.WINDOW_NORMAL)
cv2.resizeWindow("Hand Recognition", 640, 480)
cv2.setWindowProperty("Hand Recognition", cv2.WND_PROP_TOPMOST, 1)

detector = HandDetector(maxHands=1)
classifier = Classifier("Model5/M5_model.h5", "Model5/labels.txt")

offset = 20
imgSize = 224
labels = ["Control", "Down", "Up", "LCM", "Take"]

prev_time = time.time()
frameR = 100

mouse_held = False

while True:
    success, img = cap.read()
    if not success:
        continue

    hCam, wCam, _ = img.shape

    cv2.rectangle(img, (frameR, frameR), (wCam - frameR, hCam - frameR), (255, 0, 255), 2)

    hands, img = detector.findHands(img, draw=True)

    if hands:
        hand = hands[0]
        x, y, w, h = hand['bbox']
```

```

cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 2)

y1, y2 = max(0, y - offset), min(img.shape[0], y + h + offset)
x1, x2 = max(0, x - offset), min(img.shape[1], x + w + offset)
imgCrop = img[y1:y2, x1:x2]

if imgCrop.size == 0:
    continue

imgWhite = np.ones((imgSize, imgSize, 3), np.uint8) * 255

aspectRatio = h / w
if aspectRatio > 1:
    k = imgSize / h
    wCal = math.ceil(k * w)
    imgResize = cv2.resize(imgCrop, (wCal, imgSize))
    wGap = math.ceil((imgSize - wCal) / 2)
    imgWhite[:, wGap:wGap + wCal] = imgResize
else:
    k = imgSize / w
    hCal = math.ceil(k * h)
    imgResize = cv2.resize(imgCrop, (imgSize, hCal))
    hGap = math.ceil((imgSize - hCal) / 2)
    imgWhite[hGap:hGap + hCal, :] = imgResize

prediction, index = classifier.getPrediction(imgWhite)
label = labels[index]
confidence = round(max(prediction) * 100, 2)

print(f"Жест: {label}, Індекс: {index}, Впевненість: {confidence}%")

cv2.putText(img, f"{label} ({confidence}%)", (x, y - 10),
            cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 0, 255), 2)

if confidence > 80:
    lmList = hand["lmList"]

    if label == "Control" and lmList:
        index_finger = lmList[8]
        screen_x = np.interp(index_finger[0], [frameR, wCam - frameR], [0,
pyautogui.size()[0]])

```

```

        screen_y = np.interp(index_finger[1], [frameR, hCam - frameR], [0,
pyautogui.size()[1]])
        pyautogui.moveTo(screen_x, screen_y)

    elif label == "Down":
        pyautogui.scroll(-100)

    elif label == "Up":
        pyautogui.scroll(100)

    elif label == "LCM":
        pyautogui.click(button='left', clicks=2)

    elif label == "Take" and lmList:
        index_finger = lmList[8]
        screen_x = np.interp(index_finger[0], [frameR, wCam - frameR], [0,
pyautogui.size()[0]])
        screen_y = np.interp(index_finger[1], [frameR, hCam - frameR], [0,
pyautogui.size()[1]])

        if not mouse_held:
            pyautogui.mouseDown(button='left')
            mouse_held = True

        pyautogui.moveTo(screen_x, screen_y)

    else:
        if mouse_held:
            pyautogui.mouseUp(button='left')
            mouse_held = False

    else:
        if mouse_held:
            pyautogui.mouseUp(button='left')
            mouse_held = False

    curr_time = time.time()
    fps = 1 / (curr_time - prev_time)
    prev_time = curr_time
    cv2.putText(img, f"FPS: {int(fps)}", (10, 30), cv2.FONT_HERSHEY_SIMPLEX, 1, (0,
255, 0), 2)

    cv2.imshow("Hand Recognition", img)

    key = cv2.waitKey(1)
    if key == 27:
        break

```

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

DatasetTake.py

```
import cv2
from cvzone.HandTrackingModule import HandDetector
import numpy as np
import math
import time

cap = cv2.VideoCapture(0)
detector = HandDetector(maxHands=1)

offset = 20
imgSize = 300
folder = "DataV2/"
counter = 0

while True:
    success, img = cap.read()
    hands, img = detector.findHands(img)

    if hands:
        hand = hands[0]
        x, y, w, h = hand['bbox']

        imgWhite = np.ones((imgSize, imgSize, 3), np.uint8) * 255
        imgCrop = img[y - offset:y + h + offset, x - offset:x + w + offset]

        aspectRatio = h / w

        if aspectRatio > 1:
            k = imgSize / h
            wCal = math.ceil(k * w)
            imgResize = cv2.resize(imgCrop, (wCal, imgSize))
            wGap = math.ceil((imgSize - wCal) / 2)
            imgWhite[:, wGap:wCal + wGap] = imgResize
        else:
            k = imgSize / w
            hCal = math.ceil(k * h)
            imgResize = cv2.resize(imgCrop, (imgSize, hCal))
            hGap = math.ceil((imgSize - hCal) / 2)
            imgWhite[hGap:hCal + hGap, :] = imgResize

        cv2.imshow("ImageCrop", imgCrop)
        cv2.imshow("ImageWhite", imgWhite)
```

```
cv2.imshow("Image", img)

key = cv2.waitKey(1)

if key == ord("s"):
    counter += 1
    cv2.imwrite(f'{folder}/Image_{time.time()}.jpg', imgWhite)
    print(counter)

if key == 27: # 27 is the ESC key
    break

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