

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет комп'ютерних наук та кібернетики
Кафедра математичної інформатики

Кваліфікаційна робота
на здобуття ступеня магістра
за спеціальністю 122 Комп'ютерні науки

на тему:

**РОЗРОБКА МОДЕЛЕЙ МАШИННОГО НАВЧАННЯ ДЛЯ ВЗАЄМОДІЇ В
РЕАЛЬНОМУ ЧАСІ З ПРОСТОРОВИМИ ОБЧИСЛЮВАЛЬНИМИ
СИСТЕМАМИ**

Виконав студент 2-го курсу магістратури
Денис КОЧЕТКОВ



(підпис)

Науковий керівник:
професор, доктор фіз.-мат. наук
Василь ТЕРЕЩЕНКО

(підпис)

Засвідчую, що в цій курсовій роботі немає
запозичень з праць інших авторів без
відповідних посилань.

Студент



(підпис)

Роботу розглянуто й допущено до захисту
на засіданні кафедри математичної
інформатики

« 15 » _____ травня _____ 2023 р.,
протокол № 10

Завідувач кафедри
В. М. Терещенко

(підпис)

РЕФЕРАТ

Робота складається зі вступу, 2 розділів, висновків, списку використаних джерел (25 найменувань). Робота містить 18 рисунків. Загальний обсяг становить 50 сторінок, основний текст роботи викладено на 33 сторінках.

Ключові слова: МАШИННЕ НАВЧАННЯ, ПРОСТОРОВІ ОБЧИСЛЕННЯ, НЕЙРОННІ МЕРЕЖІ, ГЛИБОКЕ НАВЧАННЯ, ДОВГА КОРОТКОЧАСНА ПАМ'ЯТЬ, ВИЯВЛЕННЯ РУХУ, РОЗПІЗНАВАННЯ АКТИВНОСТІ.

Об'єкт роботи: розробка моделей машинного навчання для двовимірного розпізнавання людської діяльності за допомогою просторової обчислювальної системи.

Мета роботи: аналіз існуючих технологій та підходів до таких задач комп'ютерного зору як виявлення руху та розпізнавання людської діяльності. Вибір та підготовка наборів даних для навчання, побудова моделі для розпізнавання людської діяльності, аналіз отриманих результатів, підбір параметрів для оптимізації отриманих показників.

Інструменти розробки: вільно поширюване інтегроване середовище розробки Google Colab, мова програмування Python, просторова обчислювальна система OpenPose, бібліотеки tensorflow та keras для роботи з датасетами та нейронними мережами.

Результати роботи: був проведений аналіз існуючих підходів та технологій для розробки моделей машинного навчання для задач просторових обчислень. Було запропоновано модель нейронної мережі, що розпізнає вид людської діяльності за ключовими точками та проводить класифікацію

отриманої інформації. Отримана модель показала хороші результати на заданому наборі даних та досягла точності 96% на тестовому наборі даних.

ЗМІСТ

РЕФЕРАТ	2
СКРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ	6
ВСТУП	7
РОЗДІЛ 1. ОГЛЯД ІСНУЮЧИХ ПІДХОДІВ	10
1.1 Просторові обчислювальні системи	10
1.2 Калібрування камери	11
1.3 Епіполярна геометрія	13
1.4 Виявлення та оцінка руху	14
1.4.1 Оптичний потік	15
1.4.2 Видалення фону з відео	18
1.5 Розпізнавання людської діяльності	21
1.5.1 Розпізнавання діяльності на основі комп'ютерного зору	22
1.5.2 Розпізнавання діяльності на основі датчиків руху	23
1.6 Взаємодія в реальному часі з просторовими обчислювальними системами	25
РОЗДІЛ 2. ПРАКТИЧНА РЕАЛІЗАЦІЯ	27
2.1 Огляд використаних технологій	27
2.1.1 Google Colab	28
2.1.2 Python	29
2.1.3 Keras	29
2.1.4 OpenPose	30
2.2 Попередня обробка даних	31
2.3 Рекурентні нейронні мережі	32

	5
2.3.1 Довга короткочасна пам'ять	34
2.4 Архітектура нейронної мережі	36
2.5 Тренування нейронної мережі	38
2.6 Аналіз та оцінка результатів	38
2.7 Подальші дослідження	41
ВИСНОВКИ	43
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	45
ДОДАТОК А	49

СКОРОВЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ

VR — Virtual Reality

AR — Augmented Reality

MR — Mixed Reality

HAR — Human Activity Recognition

MHAD — Multimodal Human Action Database

RNN — Recurrent Neural Network

CNN — Convolutional Neural Network

LSTM — Long Short-Term Memory

SVM — Support Vector Machine

ВСТУП

Останніми роками галузь просторових обчислень стрімко розвивається з появою нових технологій, таких як віртуальна реальність (VR), доповнена реальність (AR) і змішана реальність (MR). Просторові обчислення дозволили нам взаємодіяти з цифровим вмістом способами, які колись вважалися неможливими, створюючи нові можливості для інновацій і творчості в багатьох галузях, від ігор до охорони здоров'я та освіти.

Машинне навчання — ще одна галузь, що швидко розвивається, і має потенціал кардинально змінити спосіб взаємодії з просторовими обчислювальними системами. Використовуючи методи машинного навчання, ми можемо створити більш персоналізовану та інтуїтивно зрозумілу взаємодію з цими системами, відкриваючи нові можливості для взаємодії в реальному часі. Машинне навчання використовує автоматизовані алгоритми для виявлення закономірностей у даних і забезпечення високоякісних прогнозів. Це суперечить класичній статистиці, яка розроблена для використання формальних рівнянь, висновків і перевірки гіпотез, щоб зробити висновок щодо певного семплу, що має вибірку, тоді як прогнози мають другорядне значення. Машинне навчання часто працює подібно до підходу чорної скриньки, на відміну від чітко визначеної, широко використовуваної статистичної моделі.

Розпізнавання діяльності людини (Human activity recognition, HAR) — це галузь досліджень, яка швидко розвивається, і останніми роками привертає все більше уваги завдяки потенційному застосуванню в різноманітних сферах, таких як охорона здоров'я, безпека та розваги. HAR передбачає виявлення та класифікацію людської діяльності на основі даних датчиків, таких як зображення чи відео. У цьому контексті машинне навчання стало потужним

інструментом для автоматичного розпізнавання людської діяльності за даними датчиків, зокрема на 2D-зображеннях.

Сучасний стан галузі характеризується розробкою нових методів і алгоритмів для HAR, включаючи моделі глибокого навчання, такі як згорткові нейронні мережі (CNN), рекурентні нейронні мережі (RNN) та їх комбінації. Ці моделі показали вражаючу продуктивність у різних додатках HAR, але все ще потрібно подолати багато проблем, таких як робота з високою варіативністю та складністю людської діяльності, обробка шумів і неповних даних, а також покращення інтерпретації та пояснюваності моделей.

Мета цієї роботи полягає в двох аспектах. По-перше, ми надаємо аналіз існуючих методів просторових обчислень, зосередивши увагу на виявленні руху та розпізнаванні людської діяльності. Ми досліджуватимемо принципи та концепції цих методів, їх застосування та обмеження в контексті просторових обчислень.

По-друге, ми розробимо модель машинного навчання для двовимірного розпізнавання людської діяльності у взаємодії з просторовими обчислювальними системами. Зокрема, ми зосередимося на розробці моделей і алгоритмів, які дозволяють точно розпізнавати діяльність людини по двовимірним позиціям ключових точок тіла та класифікувати її за шістьма різними класами. У роботі будуть використані такі методи як нейронні мережі, глибоке навчання, обробка даних, аналіз результатів та їх інтерпретація.

У першій частині цієї дипломної роботи ми надаємо огляд існуючих технологій і методів, які зараз використовуються в області просторових обчислень. Ми також досліджуватимемо принципи та концепції машинного навчання та його застосування у взаємодії в реальному часі з просторовими обчислювальними системами. Ми заглибимося в алгоритми калібровки камери, епіпольярну геометрію та методи виявлення та розпізнавання руху, а також розпізнавання діяльності людини, обговорюючи ключові проблеми та

обмеження, які необхідно вирішити, щоб зробити ці технології справді ефективними в контексті просторових обчислень.

Потім ми представимо власний підхід до розробки моделі машинного навчання для двовимірного розпізнавання людської діяльності під час взаємодії в реальному часі з просторовими обчислювальними системами. Ми опишемо нашу методологію та надамо деталі нашої експериментальної установки, включаючи використані набори даних і показники продуктивності, використані для оцінки нашої моделі. Ми обговоримо наші результати та їхні наслідки для майбутньої взаємодії в реальному часі з просторовими обчислювальними системами.

РОЗДІЛ 1. ОГЛЯД ІСНУЮЧИХ ПІДХОДІВ

1.1 Просторові обчислювальні системи

Просторові обчислювальні системи – це тип технології, яка використовує комп’ютерний зір, машинне навчання та інші методи штучного інтелекту для забезпечення взаємодії з фізичними просторами в реальному часі. Ці системи покладаються на технології просторового відображення та відстеження, такі як камери, датчики глибини та GPS-приймачі, щоб зрозуміти положення користувача та рухи у фізичному просторі.

Термін просторових обчислень вперше з’явився в 2003 році в книжці Саймона Грінволда “Spatial Computing”. Тоді він визначив це як «взаємодія людини з машиною, під час якої машина зберігає та маніпулює посиланнями на реальні об’єкти та простори» [1].

Критично важливими компонентами просторових обчислювальних систем є технології віртуальної (VR), доповненої (AR) та змішаної реальності (MR). Віртуальна реальність створює віртуальне середовище з повним зануренням, тоді як доповнена реальність накладає цифрову інформацію на фізичний світ [2].

Прикладами технологій, які можуть бути впроваджені в просторові обчислювальні системи є, наприклад, виявлення руху та розпізнавання людської діяльності. Вони є важливими функціями просторових обчислювальних систем, які дозволяють їм виявляти та інтерпретувати рухи та дії користувача. Виявлення руху дозволяє системі виявляти рух у фізичному просторі, тоді як розпізнавання людської діяльності використовує методи машинного навчання для класифікації та інтерпретації рухів і дій користувача. Ці технології знаходять застосування в таких сферах, як ігри, відстеження фізичної

активності та моніторинг безпеки, а також у створенні інтерактивних художніх інсталяцій.

В останні роки методи машинного навчання були інтегровані в просторові обчислювальні системи, щоб покращити їхню продуктивність і забезпечити більш розширені функції. Обробка природної мови, розпізнавання об'єктів і розпізнавання жестів є одними з можливостей, які стали можливими завдяки цим методам, які дозволяють більш інтуїтивно зрозумілу та природню взаємодію між користувачами та віртуальним середовищем.

1.2 Калібрування камери

Використання камер для просторових обчислень презентує деякі проблеми. Зокрема, деякі камери-обскури вносять значне спотворення зображення. Два основних види викривлення - це радіальне викривлення та тангенціальне викривлення. Калібрування камери таким чином є важливим етапом у системах просторового обчислення, що включає оцінку внутрішніх та зовнішніх параметрів камери для видалення спотворень з зображень. Неспроможність виправити спотворення може призвести до неточного визнання та відстеження об'єктів, що може впливати на ефективність систем просторового обчислення для розпізнавання діяльності людей.

Радіальне викривлення призводить до того, що прямі лінії виглядають вигнутими або випуклими (Рисунок 1). Радіальне викривлення стає більшим, чим далі точки від центру зображення.

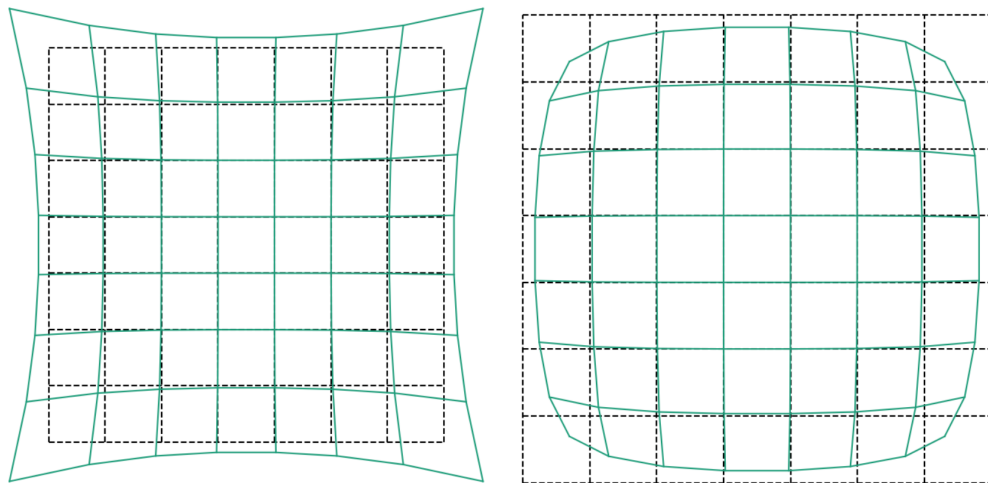


Рисунок 1 – Радіальне викривлення камери.

Тангенціальне викривлення часто виникає через децентрованість лінзи відносно площини зображення, тобто що лінза, що знімає зображення, не вирівняна ідеально паралельно площині зображення.

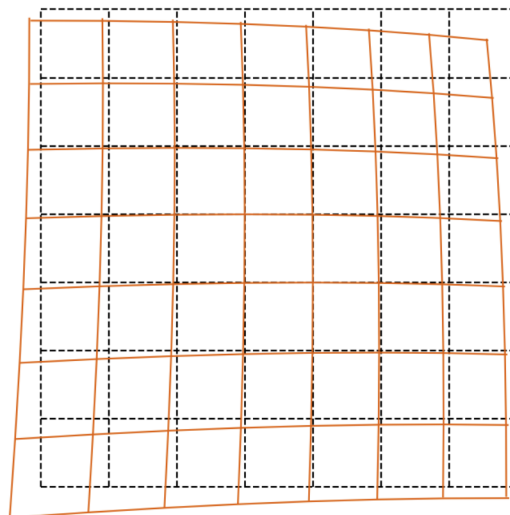


Рисунок 2 – Тангенціальне викривлення камери.

Для виправлення таких видів викривлень існують спеціальні методи, що характеризують викривлення як серію поліномів вищого порядку. Ця модель

бере свій початок в 1919 році в роботі Олександра Конраді [3] та потім була вдосконалена Діном Брауном в 1966 році [4]. Метод Брауна – Конраді все ще складає основу багатьох підходів щодо характеристики і моделювання викривлень камери.

В 2006 році даний метод був адаптований та оптимізований для ширококутних, надширококутних об'єктивів та об'єктивів «риб'яче око» [5]. Моделювання Брауна і Конраді в основному базувалося на фізиці аберацій Зейделя, які вперше були сформульовані приблизно в 1867 році для стандартної фізики лінз того часу, яка не включала ультраширокі лінзи та лінзи типу «риб'яче око».

1.3 Епіполярна геометрія

Епіполярна геометрія – це геометрія стерео зору. Епіполярна геометрія використовується для об'ємної реконструкції реальних об'єктів у просторі. Епіполярна геометрія є фундаментальним поняттям у комп'ютерному зорі та просторовому обчисленні, що описує взаємозв'язок між двома видами сцени, зазвичай знятими двома різними камерами. У системі стереобачення, епіполярна геометрія використовується для обчислення відповідності між точками на двох камерах, що дозволяє створити карти глибини та відновити 3D-структуру сцени.

Якщо ми маємо зображення з двох камер, що розглядають тривимірну сцену з двох різних позицій, ми можемо обчислити ряд геометричних співвідношень між тривимірними точками та їх проекціями на двовимірні зображення, що призводить до обмежень можливих положень точок зображення.

На рисунку нижче показано дві камери, що дивляться на одну й ту саму точку X (Рисунок 3).

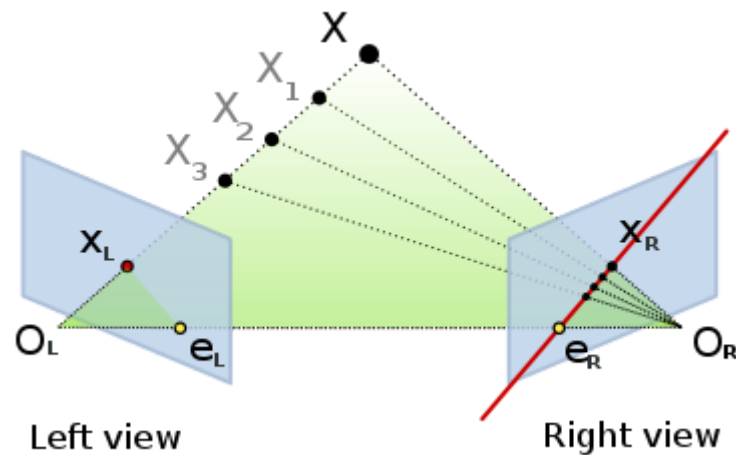


Рисунок 3 – Епіполярна геометрія.

Оскільки оптичні центри двох камер різні, кожен центр можна спроектувати на окрему точку в площині зображення іншої камери. Ці дві точки – e_L та e_R називаються епіполями, або епіполярними точками. Лінії $O_R X$ та $O_L X$, що розглядаються як точки в правій та лівій камерах відповідно, називаються епіполярними лініями.

1.4 Виявлення та оцінка руху

Виявлення та оцінка руху є важливими компонентами просторових обчислювальних систем, оскільки вони дозволяють системі відстежувати та інтерпретувати рухи та дії користувача в реальному часі. У цьому підрозділі ми дослідимо ключові концепції та техніки, що застосовуються в процесі виявлення та оцінки руху, а також їх застосування в контексті просторових обчислювальних систем.

Існує кілька методів, які використовуються для виявлення та оцінки руху, включаючи оптичний потік, видалення фону та методи на основі глибини.

1.4.1 Оптичний потік

Оптичний потік (optical flow) – це шаблон видимого руху об'єктів, поверхонь та країв у візуальній сцені, що спричинений відносним рухом між спостерігачем (камерою) і сценою. Оптичний потік – двовимірне векторне поле, де кожен вектор є вектором переміщення, який показує рух точок з першого кадру до другого.

Концепцію оптичного потоку вперше ввів американський психолог Джеймс Дж. Гібсон у 1940-х роках для опису візуального стимула, який отримують тварини, рухаючись через світ. Гібсон наголошував на важливості оптичного потоку для сприйняття можливостей дії в середовищі.

Також термін оптичний потік використовується робототехніками, що охоплює пов'язані техніки обробки зображень та керування навігацією, включаючи виявлення руху, сегментацію об'єктів, інформацію про час до контакту, обчислення точки розширення, яскравість, кодування з компенсацією руху та вимірювання стереодиспарності.

Оптичний потік спирається на два припущення: інтенсивність пікселів не змінюється між двома послідовними кадрами та сусідні пікселі мають схожий рух.

Розглянемо піксель $I(x, y, t)$ на першому кадрі. Він рухається зі швидкістю (dx, dy) в наступному кадрі, який зроблено після dt часу. Оскільки пікселі однакові, тобто інтенсивність не змінилася, то

$$I(x, y, t) = I(x + dx, y + dy, t + dt)$$

Якщо взяти розклад в ряд Тейлора правої частини та спростити, то отримаємо:

$$f_x u + f_y v + f_t = 0$$

де

$$f_x = \frac{\partial f}{\partial x}; f_y = \frac{\partial f}{\partial y}$$

$$u = \frac{dx}{dt}; v = \frac{dy}{dt}$$

Це рівняння називається рівнянням оптичного потоку. Часткові похідні є градієнтами зображення, (u, v) – невідомі.

Є доволі багато алгоритмів та методів визначення оптичного руху за послідовністю зображень. Зі всіх існуючих алгоритмів варто виділити два – метод Лукаса – Канаде (розріджений), та метод Фарнебека (щільний).

Метод Лукаса – Канаде є широко вживаним диференціальним методом оцінювання оптичного потоку, розроблений Брюсом Д. Лукасом та Такео Канаде в 1981 році [6]. Як зазначено вище, він спирається на припущення, що зміщення вмісту зображення між двома сусідніми моментами (кадрами) є малим і приблизно сталим в межах околу точки p , яку розглядають. Метод розв'язує базові рівняння оптичного потоку для всіх пікселів у цьому околі методом найменших квадратів.

Розв'язання методом найменших квадратів надає однакової важливості всім n пікселям околу точки p . На практиці зазвичай краще надавати більшої ваги пікселям, ближчим до центрального пікселя. Для цього використовують зважену версію рівняння найменших квадратів:

$$A^T W A v = A^T W b$$

Тут W – це діагональна матриця $n \times n$, що містить ваги $W_{ii} = w_i$.

Даний метод спирається на невірність матриці $A^T A$. Це зазвичай означає, що точка перебуває на ребрі, тому доцільно застосовувати метод Лукаса – Канаде на кутах зображення, наприклад визначених методом Сі – Томазі.



Рисунок 4 – Знаходження оптичного потоку на кутових точках зображення.

Метод Фарнебека є щільним алгоритмом знаходження оптичного поток, що рахує оптичний потік для всіх точок зображення. Він заснований на роботі Гуннера Фарнебека «Two-Frame Motion Estimation Based on Polynomial Expansion» в 2003 році [7].



Рисунок 5 – Щільний оптичний потік методом Фарнебека.

1.4.2 Видалення фону з відео

Видалення фону є важливим етапом попередньої обробки в багатьох застосуваннях на основі комп'ютерного зору. Наприклад, можна розглянути випадок підрахування відвідувачів у музеї, де статична камера відстежує кількість людей, що входять або виходять з кімнати, або дорожньої камери, яка отримує інформацію про транспортні засоби і т.д. В усіх цих випадках перш за

все потрібно виділити окремо людей або транспортні засоби. Технічно, потрібно відділити рухомий передній план від статичного фону.

За наявності окремого зображення фону, наприклад, кімнати без відвідувачів, задача стає набагато легшою. Але в більшості випадків такого зображення може не бути, тому ми повинні видалити фон з будь-яких зображень, які маємо. Ускладнює задачу наявність тіней, які можна теж позначити як передній план, що не є коректним.

Для розв'язку цієї задачі було запропоновано декілька алгоритмів. Розглянемо деякі з них для виявлення статичного фону на наступному зображенні, взятому з тестового відео OpenCV:



Рисунок 6 – Кадр з тестового відео.

1. Алгоритм сегментації переднього/фонового плану на основі суміші гаусівських розподілів. Він був запропонований у статті авторів P. KadewTraKuPong та R. Bowden у 2001 році [8]. Алгоритм використовує метод моделювання кожного пікселя фону за допомогою суміші k гаусівських розподілів ($k = 3-5$). Ваги суміші пропорційні часу, який ті кольори перебувають на сцені. Ймовірними кольорами фону є ті, які залишаються на екрані довше та більш статичні.

В бібліотеці OpenCV даний алгоритм позначений як BackgroundSubtractorMOG.



Рисунок 7 – Результат алгоритму BackgroundSubtractorMOG.

2. Алгоритм заснований на Гаусівських змішаних моделях був вдосконалений в двох статтях Z. Zivkovic 2004 та 2006 року [9] [10]. Головною відмінністю є те, що алгоритм обирає підходящу кількість гаусівських розподілів для кожного пікселя замість того, щоб постійно брати певну кількість. Це краще адаптує алгоритм до змін у середовищі, таких як зміна освітлення.



Рисунок 8 – Результат алгоритму BackgroundSubtractorMOG2. Сіра область вказує на наявність тіні.

3. Наступний алгоритм BackgroundSubtractorGMG поєднує в собі статистичну оцінку зображення фону та байєсівську сегментацію пікселів. Він був запропонований Andrew B. Godbehere, Akihiro Matsukawa та Ken Goldberg у статті «Visual Tracking of Human Visitors under Variable-Lighting Conditions for a Responsive Audio Art Installation» у 2012 році [11]. Згідно зі статтею, система успішно працювала у взаємодії зі звуковою арт-інсталяцією «Are We There Yet?» з 31 березня по 31 липня 2011 року в Contemporary Jewish Museum у Сан-Франциско.

Для моделювання фону використовуються перші кілька (за замовчуванням 120) кадрів. Алгоритм використовує ймовірнісну сегментацію переднього плану, яка ідентифікує можливі об'єкти переднього плану за допомогою байєсівського виведення. Оцінки є адаптивними; нові спостереження мають більшу вагу, ніж старі, щоб врахувати зміни в освітленні.

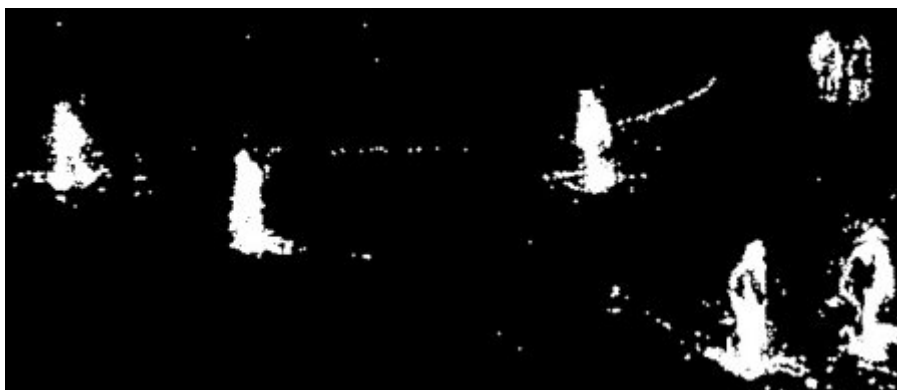


Рисунок 9 – Результат алгоритму BackgroundSubtractorGMG.

1.5 Розпізнавання людської діяльності

Розпізнавання людської діяльності (Human Activity Recognition, HAR) охоплює техніки, які використовуються для автоматичного визначення та класифікації людської діяльності з даних датчиків, таких як відео, звук та

датчики руху. Метою HAR є створення моделі машинного навчання, яка може визнавати та класифікувати людські дії, такі як ходьба, біг, стрибки, сидіння, стояння та інші дії.

Техніки та алгоритми, що використовуються для розпізнавання людської діяльності можуть включати, наприклад, методи навчання з учителем, навчання без учителя та глибоке навчання. Традиційні підходи HAR використовують створені вручну функції та алгоритми класифікації, такі як опорні векторні машини (SVM) і дерева рішень. Однак останні досягнення в глибокому навчанні призвели до значного покращення продуктивності систем HAR. Підходи на основі глибокого навчання можуть автоматично вивчати складні функції з необроблених даних датчика, що може призвести до кращої точності розпізнавання.

Через доволі широке застосування, розпізнавання діяльності може мати вплив на розпізнавання цілей, розпізнавання намірів людини, розпізнавання поведінки, оцінку місця розташування, тощо.

1.5.1 Розпізнавання діяльності на основі комп'ютерного зору

Відстежування активності та діяльності людей за допомогою знятого відео (або наживо) є дуже важливою і складною проблемою. Основною технікою, що використовується, є комп'ютерний зір. Розпізнавання активності на основі зору знайшло багато застосувань, зокрема взаємодія людини з комп'ютером, розробка інтерфейсу користувача, навчання роботів і спостереження тощо. Наукові конференції, на яких часто з'являється робота з розпізнавання діяльності на основі бачення, це ICCV і CVPR.

Розпізнавання діяльності на основі комп'ютерного зору гарно досліджене, в цій області було виконано багато роботи. Дослідники випробували низку методів, таких як оптичний потік, фільтрація Калмана, приховані марковські

моделі тощо, у різних модальностях, таких як одна камера, стерео та інфрачервоне випромінювання. Крім того, дослідники розглянули кілька аспектів цієї теми, включаючи відстеження одного пішохода, групове відстеження та виявлення об'єктів, що впали.

Зазвичай для задач розпізнавання людської діяльності застосовуються спеціальні камери глибини RGBD, такі як Microsoft Kinect. Камери глибини додають додатковий вимір, тобто глибину, яку не може забезпечити звичайна двовимірна камера. Сенсорна інформація з цих камер глибини використовується для створення в реальному часі моделі скелета людей з різними положеннями тіла. Ця скелетна інформація надає значущу інформацію, яку дослідники можуть використовувати для моделювання людської діяльності, яку навчають і пізніше використовують для розпізнавання невідомих дій [12] [13].

В даній роботі ми перевіряємо ефективність використання двовимірних даних про положення тіла та ключових точок для визначення діяльності людини.

1.5.2 Розпізнавання діяльності на основі датчиків руху

Як було сказано вище, застосування RGBD камер в поєднанні з контрольованими методами машинного навчання для задач розпізнавання людської активності доволі широко розповсюджене. Ці підходи на основі комп'ютерного зору зазвичай вимагають кількох полів зору, але не можуть безпосередньо вимірювати рухи тіла. Однак підходи HAR на основі датчиків можуть подолати обмеження систем зору за допомогою легких компактних сенсорів, що кріпляться на тілі, і безпосередньо вимірюють рухи тіла. HAR на основі датчиків зазвичай працює за допомогою датчиків, встановлених на тілі, розумних годинниках і браслетах [14] [15], збираючи дані про рух безпосередньо з певних областей або позицій.

З цими типами сенсорів, традиційні методи машинного навчання застосовуються в системах HAR. Наступним кроком стали глибинні мережі навчання, що стали провідним рішенням, щоб перебороти обмеження традиційних методів машинного навчання [16]. Дослідження з HAR на основі глибинного навчання в основному проводяться з використанням декількох датчиків руху, розташованих на грудях, талії та зап'ясті, які фіксують рухи людського тіла та рухи рук, та досягають загальної точності HAR понад 95%.

Використання екзоскелетного робота з інтегрованими датчиками руху показало дуже гарний результат в роботі 2022 року [17]. В роботі було перевірено можливість використання HAR у реальному часі за допомогою носимого робота, що містить інтегровані датчики руху та вбудовані моделі глибинного навчання в кращому пристрої. Успішні результати цієї роботи демонструють автономну систему HAR, яка може використовуватися для допомоги людським рухам та завданням з використанням носимих роботів. Представлений підхід до HAR знижує загальну затримку відповіді попередніх спроб, зберігаючи точність розпізнавання вище 97% в реальному часі.

Впроваджена в [17] система має архітектуру, представлену на рисунку нижче. Носимий робот має вбудовані сенсори, дані з яких після попередньої обробки поступають на вхід моделям машинного навчання. Класифікація результатів показана на останньому рисунку.

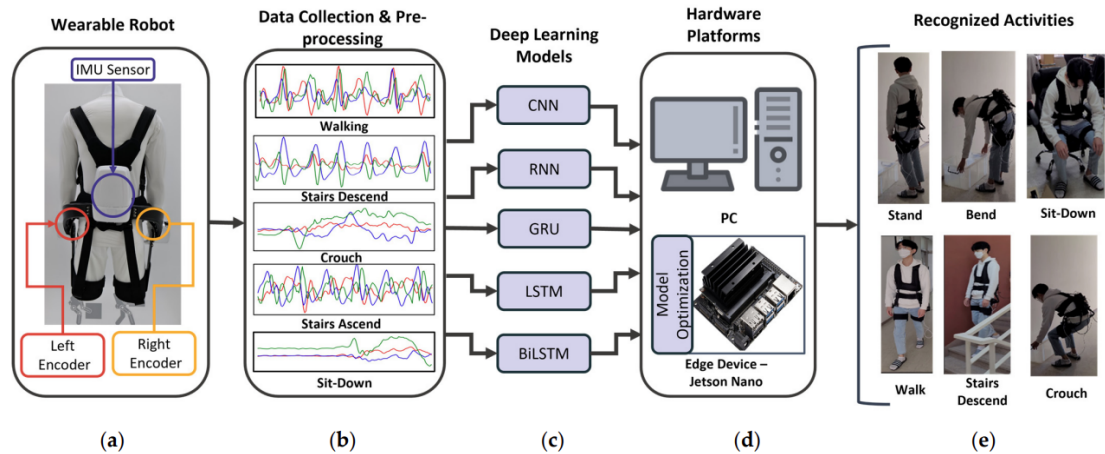


Рисунок 10 – Розпізнавання людської активності за допомогою датчиків руху.

1.6 Взаємодія в реальному часі з просторовими обчислювальними системами

Взаємодія в реальному часі є критично важливим аспектом просторових обчислювальних систем, оскільки вона дозволяє користувачам взаємодіяти з віртуальним середовищем у реальному часі та отримувати негайний зворотний зв'язок. Взаємодія в режимі реального часу передбачає використання методів машинного навчання для виявлення й інтерпретації рухів і дій користувача, а потім реагування на ці дії в режимі реального часу.

Взаємодія в реальному часі може бути досягнута за допомогою різних методів машинного навчання, таких як глибоке навчання, навчання з підкріпленням і байєсовські мережі. Глибоке навчання є особливо ефективним у взаємодії в режимі реального часу, оскільки воно може аналізувати великі обсяги даних і робити прогнози в режимі реального часу. Натренована нейронна мережа зазвичай демонструє дуже гарну ефективність роботи в реальному часі.

Взаємодія в режимі реального часу використовується в широкому діапазоні додатків, включаючи ігри, віртуальну та доповнену реальність і

робототехніку. В іграх взаємодія в режимі реального часу дозволяє гравцям взаємодіяти з ігровим середовищем та іншими гравцями в режимі реального часу, створюючи більш захоплюючий і захоплюючий досвід. У віртуальній і доповненій реальності взаємодія в реальному часі дозволяє користувачам взаємодіяти з віртуальними об'єктами та середовищами та отримувати негайний зворотний зв'язок щодо своїх дій. У робототехніці взаємодія в режимі реального часу дозволяє роботам взаємодіяти з людьми та навколишнім середовищем у режимі реального часу, забезпечуючи більш природну та інтуїтивно зрозумілу взаємодію.

РОЗДІЛ 2. ПРАКТИЧНА РЕАЛІЗАЦІЯ

У даній роботі ми розглядаємо метод класифікації людської активності, що базується на використанні набору даних, який представлений часовим рядом двовимірних поз та рекурентної нейронної мережі LSTM. Ми пропонуємо ідею, що використання двовимірних зображень замість тривимірних поз може бути ефективним методом для точної оцінки діяльності людини та передбачення її наступного стану.

Основною метою нашого експерименту є порівняння точності розпізнавання в двовимірному та тривимірному випадках. Якщо результати будуть позитивними, то це дозволить використовувати звичайні RGB відеокамери для захоплення рухів замість дорогих та спеціалізованих тривимірних камер.

Архітектура нейронної мережі, яку ми використовуємо в цьому дослідженні, має схожість з мережами, які використовуються для розв'язування подібних задач визначення людської активності [18]. Наша мережа є рекурентною та здатна давати точні результати на основі даних, що подаються у вигляді часового ряду двовимірних поз. Ми розглядаємо можливість застосування нашого методу для вирішення різноманітних завдань, пов'язаних з розпізнаванням діяльності людей у різних контекстах.

2.1 Огляд використаних технологій

Розробка методів машинного навчання для взаємодії в реальному часі з просторовими обчислювальними системами вимагає поєднання програмних і апаратних технологій. В своїй роботі я використовую бібліотеку Keras як ключовий механізм побудови та тренування моделі нейронної мережі. Для

розробки використовується середа розробки Google Colab та мова програмування Python.

2.1.1 Google Colab

Google Colab – це безкоштовна хмарна платформа, яка надає можливість запускати код Python на віддалених серверах Google. Це забезпечує широкий доступ до обчислювальних ресурсів, що дозволяє легко створювати та навчати складні моделі машинного навчання без необхідності встановлювати спеціальне обладнання або програмне забезпечення на власному комп'ютері. Google Colab є ідеальною платформою для розробки та тестування моделей машинного навчання, оскільки вона дає змогу користувачам тренувати моделі на великих наборах даних за частку часу, який знадобився б за використання стандартного комп'ютера.

З використанням Google Colab можна створювати та навчати моделі машинного навчання з використанням бібліотек Python, таких як Keras та OpenCV. Google Colab надає доступ до високопродуктивних обчислювальних ресурсів, що дозволяє ефективно навчати складні моделі машинного навчання для різноманітних задач.

Крім того, Google Colab може бути використано для спільної роботи та обміну даними між дослідниками та розробниками. За допомогою Google Colab можна легко навчати моделі на великих обсягах даних та ділитися результатами дослідження з іншими користувачами, що дозволяє прискорити процес розробки та підвищити ефективність моделей машинного навчання.

2.1.2 Python

Python – це мова програмування високого рівня, яка широко використовується в спільноті машинного навчання завдяки простоті використання та багатій екосистемі бібліотек і інструментів [19]. Python особливо добре підходить для розробки моделей машинного навчання завдяки своїй простоті та читабельності, що дозволяє легко писати та налагоджувати складний код.

Популярні бібліотеки машинного навчання Python, такі як TensorFlow, Keras та PyTorch, надають потужні інструменти для будівництва та тренування моделей глибокого навчання. Ці моделі можуть використовуватися для завдань, таких як розпізнавання об'єктів, аналіз зображень та відео, розпізнавання людської активності.

Python також надає бібліотеки для аналізу та маніпулювання даними, такі як pandas та NumPy, які є необхідними для підготовки та обробки даних, використовуваних для тренування моделей машинного навчання. Крім того, у Python є велика та активна спільнота розробників, які створюють та діляться відкритими бібліотеками, інструментами та підручниками, що полегшують розробку моделей машинного навчання.

Загалом, Python є універсальною та потужною мовою програмування для розробки моделей машинного навчання для розпізнавання людської активності та систем просторового обчислення, завдяки її багатому екосистемі бібліотек, фреймворків та інструментів.

2.1.3 Keras

Keras – це одна з популярних бібліотек глибокого навчання, яка надає простий та інтуїтивно зрозумілий інтерфейс для створення та навчання глибоких нейронних мереж [20]. Keras створено на основі TensorFlow,

потужного фреймворку машинного навчання, розробленого Google, і надає високорівневий API для створення складних моделей лише за допомогою кількох рядків коду. Keras також містить широкий спектр попередньо створених шарів і моделей, які можна використовувати як будівельні блоки для користувацьких моделей, а також інструменти для попередньої обробки даних і візуалізації.

Крім того, Keras підтримує кілька серверних модулів, включаючи TensorFlow, Microsoft Cognitive Toolkit і Theano, що дозволяє користувачам вибрати структуру, яка найкраще відповідає їхнім потребам. Загалом Keras є потужним інструментом для створення моделей машинного навчання для різноманітних додатків, включаючи системи розпізнавання людської діяльності та просторові обчислювальні системи.

2.1.4 OpenPose

OpenPose - це система комп'ютерного зору, яка використовує глибоке навчання для аналізу людської постави та рухів зображення людини або відео. Вона є першою системою, яка в режимі реального часу визначає ключові точки людського тіла, рук, обличчя та ніг на окремих зображеннях для декількох людей одночасно [21]. Вона була запропонована в 2019 році.

Ми використовуємо OpenPose в цій роботі для виявлення та відстеження ключових точок людини на відео. Ця інформація далі використовується як вхідні дані для навчання моделі машинного навчання для задачі розпізнавання дій та видів активності.

2.2 Попередня обробка даних

Набір даних, використаний в даній роботі, складається з підмножини набору даних Berkeley Multimodal Human Action Database (MHAD). Цей набір даних містить відео 12 суб'єктів, які виконують 6 різних видів активності по 5 повторів. Зйомка відбувається з чотирьох кутів зору. Види активності наступні:

- Стрибок
- Стрибок з піднятими руками (jumping jacks)
- Бокс
- Махання двома руками
- Махання однією рукою
- Плескання долонями

Всього налічується 1438 відео, що в сумі складають 211200 кадрів.

Для того щоб отримати дані про координати суглобів, ми використовуємо програмний застосунок для OpenPose [21]. Це дуже потужний інструмент, який дозволяє в режимі реального часу визначати ключові точки людського тіла, рук, обличчя та ніг на окремих зображеннях. Ми використовуємо цей інструмент для розпізнавання 18 ключових точок на кожному кадрі заданого відео. Далі двовимірні координати цих ключових точок передаються на вхід нейронній мережі.

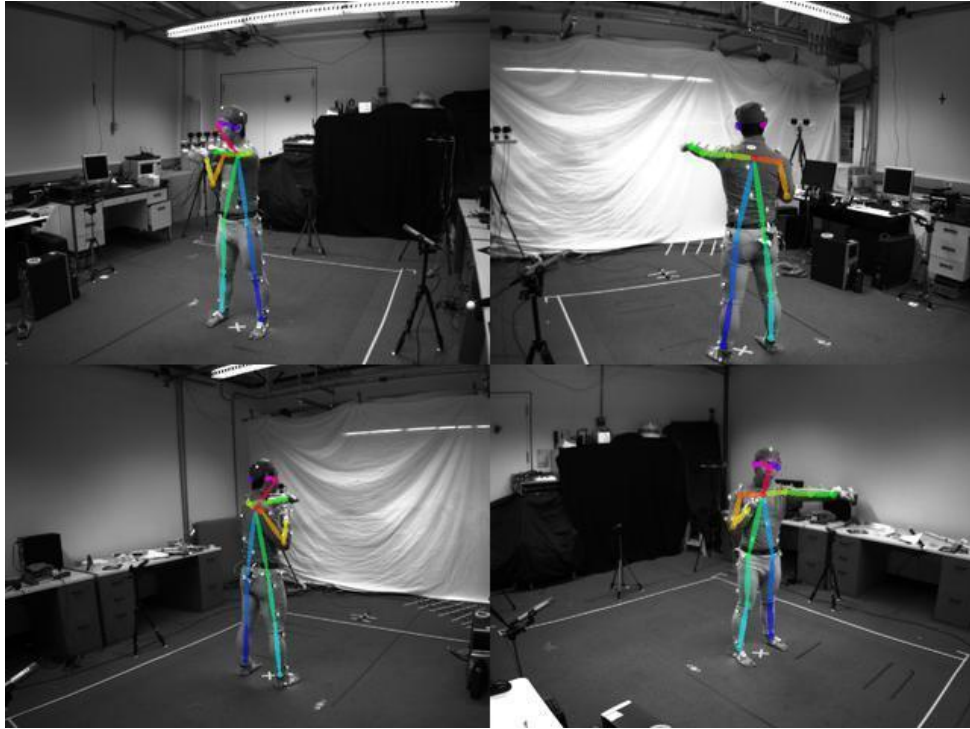


Рисунок 11 – Приклад кадру з набору даних з оцінкою ключових точок.

На вхід до нашої нейронної мережі подається часовий ряд з двовимірними позиціями 18 ключових точок («суглобів») та класифікатор послідовності, тобто вид активності.

Вся попередня обробка даних складається з того, щоб запустити OpenPose на окремих кадрах для кожного суб'єкта та виду активності і привести дані до однієї таблиці. Нормалізація даних відносно позиції на кадрі та випадкових рухів не проводилася.

2.3 Рекурентні нейронні мережі

Рекурентні нейронні мережі (англ. Recurrent Neural Networks, RNN) — це сімейство нейронних мереж, які використовуються для обробки послідовних даних. Останнім часом вони продемонстрували великі успіхи в задачах, пов'язаних із машинним перекладом, аналізом тексту, розпізнаванням

мовлення, аналізом часових рядів та іншими проблемами, що базуються на послідовностях даних.

RNN мають перевагу в тому, що вони узагальнюють послідовності, а не вивчають окремі шаблони. Вони роблять це, фіксуючи динаміку послідовності через з'єднання циклів і спільні параметри. RNN також не обмежені фіксованим розміром послідовності і, теоретично, можуть враховувати всі попередні кроки послідовності. Це робить їх дуже придатними для аналізу послідовних даних.

Технічно RNN можна розглядати як ланцюг із кількох копій однієї статичної мережі A , причому кожна копія працює на одному часовому кроці вхідної послідовності. Копії мережі підключаються через їх приховані стани. Це означає, що кожна з мережевих копій має кілька входів: поточне значення $x(t)$ та приховані стани $h(t - 1)$ як результат попередньої копії.

RNN часто зображують як циклічний зв'язок (Рисунок 12). Мережа A – статична мережа з одним або багатьма шарами. Цикл означає, що вихід мережі A на кроці t є входом для мережі A на кроці $t + 1$.

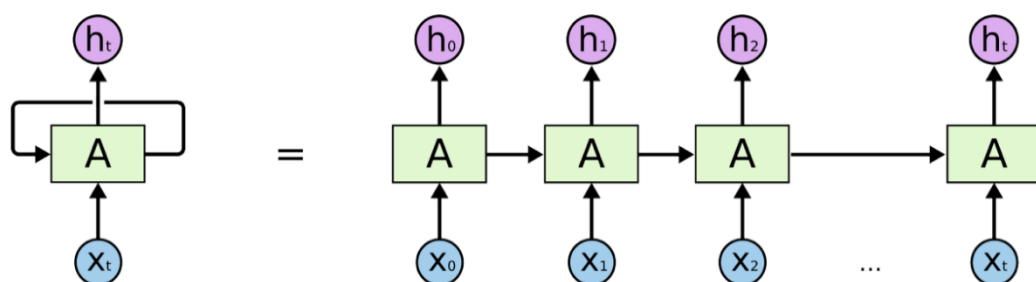


Рисунок 12 – Зображення рекурентної нейронної мережі.

Рекурентні нейронні мережі довгий час здавалися багатообіцяючими, вони незабаром показали свої обмеження в можливостях довгої пам'яті. Саме

тоді архітектура LSTM (Long short-term memory) стали привертати увагу як доповнення до RNN.

2.3.1 Довга короткочасна пам'ять

Довга короткочасна пам'ять (англ. Long short-term memory, LSTM) – це архітектура рекурентних нейронних мереж, запропонована 1997 року Зеппом Хохрайтером та Юргеном Шмідгубером [22].

Мережі LSTM мають ту ж саму ланцюгову структуру, що і RNN. Різниця тільки в структурі мережі А. Вузол LSTM — це вузол рекурентної нейронної мережі, який виділяється запам'ятовуванням значень для довгих, або коротких проміжків часу. Ключем до цієї здатності є те, що він не використовує функції активації в межах своїх рекурентних складових. Таким чином, значення, що зберігається, не розплющується ітеративно з плином часу. Тим самим ми уникаємо проблему зникання градієнту (Рисунок 13) [23].

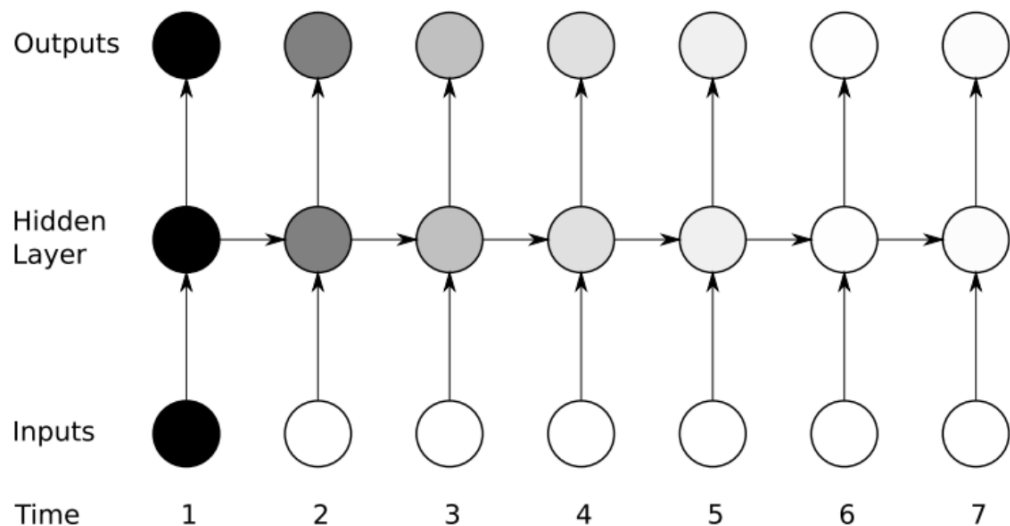


Рисунок 13 – Проблема зникання градієнту. Чутливість до вхідних даних зменшується з часом.

Блоки ДКЧП (Рисунок 14) містять три або чотири «вентилі» (англ. gates), які вони використовують для керування плином інформації до або з їхньої пам'яті. Ці вентилі реалізують із застосуванням логістичної функції для обчислення значень між 0 та 1. Наприклад, «вхідний вентиль» (англ. input gate) керує мірою, до якої нове значення входить до пам'яті. «Забувальний вентиль» (англ. forget gate) керує мірою, до якої значення залишається в пам'яті. А «вихідний вентиль» (англ. output gate) керує мірою, до якої значення в пам'яті використовується для обчислення активування виходу блоку.

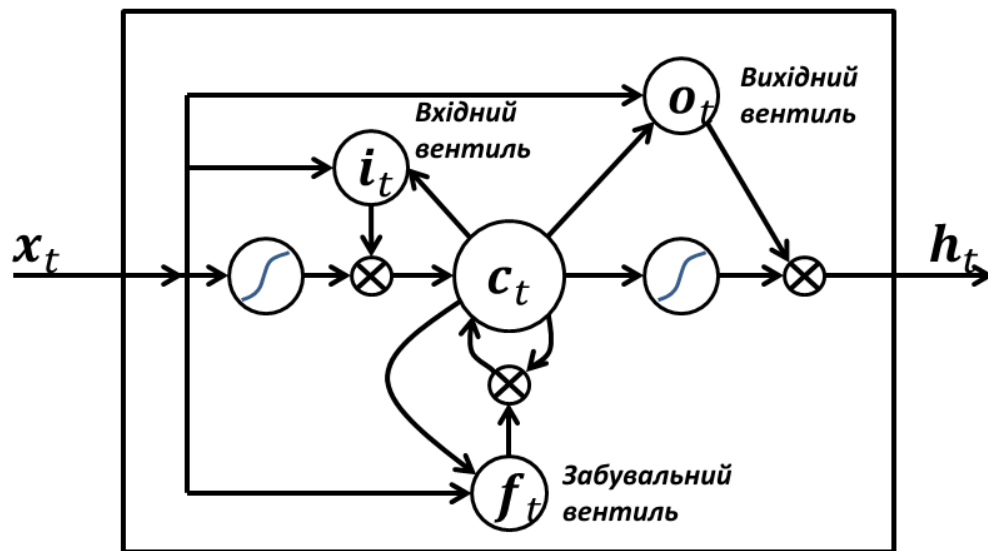


Рисунок 14 – Блок LSTM з вентиллями.

Дії кожного вентиля можна записати наступними математичними формулами. Для вхідного вентиля:

$$i_t = f_i(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i)$$

Для забувального вентиля:

$$f_t = f_f(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f)$$

Для стану блоку:

$$c_t = f_t c_{t-1} + \tanh(W_{hc} x_t + W_{hc} h_{t-1} + b_c)$$

Для вихідного вентиля:

$$o_t = f_o (W_{xo} x_t + W_{ho} h_{t-1} + W_{co} c_t + b_o)$$

Вихідне значення блоку:

$$h_t = o_t \tanh(c_t)$$

Забувальні вентиля мають дуже важливе значення для навчання. Оригінальна форма LSTM має недолік в тому, що блок скидається до нейтрального стану, коли починається нова послідовність навчання. Через це Фелікс Герс запропонував в 2000 році впровадження «скидаючого вентиля» [24].

2.4 Архітектура нейронної мережі

Архітектура нейронної мережі описує загальну структуру мережі, яка складається з різних шарів різних типів, розташованих у певному порядку. Правильно розроблена архітектура може значно підвищити продуктивність нейронної мережі при виконанні певного завдання, тоді як погано спроектована може призвести до неоптимальної продуктивності або навіть до невдачі в навчанні.

У даному підрозділі ми дослідимо архітектуру простої нейронної мережі, яка використовує шари рекурентної нейронної мережі (RNN). Ми представимо концепцію RNN і пояснимо, як вони використовуються для послідовних даних. Потім ми обговоримо структуру мережі та те, як кожен рівень сприяє її загальній функції.

Для навчання ми використовуємо рекурентну нейронну мережу, яка складається з комбінації щільних шарів та шарів довгої короткочасної пам'яті.

Вона показує гарні результати для задачі класифікації послідовностей даних двовимірних позицій.

Наша модель складається з чотирьох шарів – два щільних шари та два LSTM шари. Перший щільний шар слугує початковим екстрактором ознак для послідовностей даних. Далі йдуть два LSTM шари, які виводять один результуючий вектор. Фінальний щільний шар використовує функцію softmax для перетворення цього вектору на вектор з розмірністю рівною кількості класів в нашій задачі класифікації.

Візуальне представлення архітектури нейронної мережі представлено на рисунку нижче.

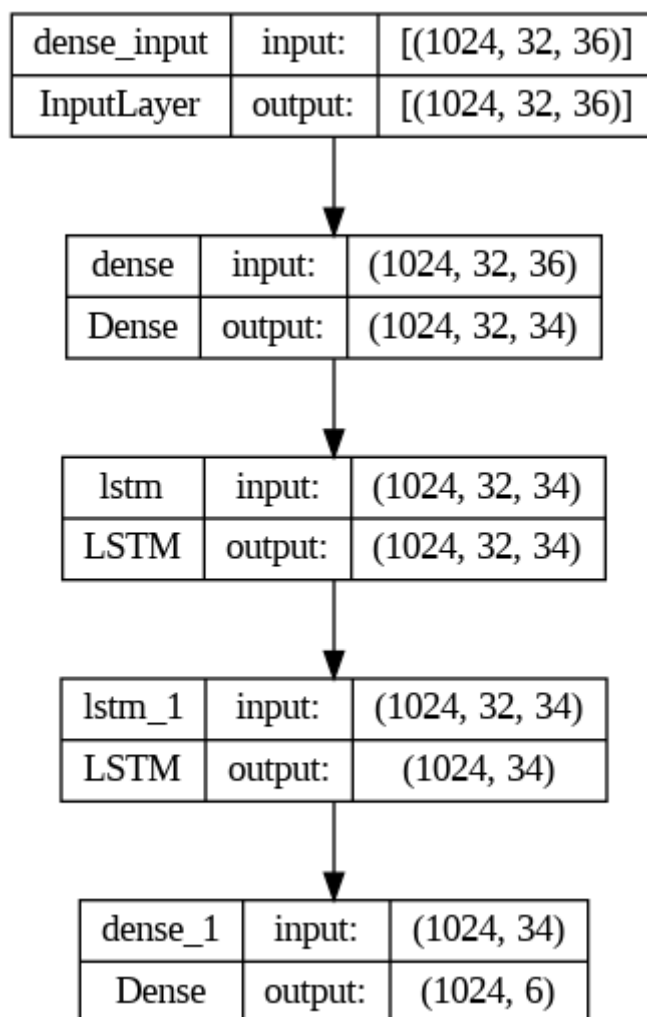


Рисунок 15 – Схема архітектури нейронної мережі.

2.5 Тренування нейронної мережі

Процес тренування нейронної мережі є ключовим етапом в розробці будь-якої моделі машинного навчання. Основна мета цього процесу полягає в налаштуванні ваг і зсувів нейронної мережі з метою досягнення мінімальної помилки на тренувальних даних.

Процес тренування нейронної мережі починається з випадкової ініціалізації ваг та зсувів нейронів. Після цього відбувається пряме поширення даних через мережу, під час якого вихідні значення обчислюються для кожного шару нейронів. Потім обчислюється значення функції витрат, яка оцінює різницю між прогнозованими та фактичними значеннями вихідних даних.

Далі відбувається зворотнє поширення помилки (backpropagation), під час якого похідні функції витрат обчислюються по вагам та зсувам мережі. Значення похідних використовуються для коригування ваг та зсувів таким чином, щоб мінімізувати значення функції витрат. Цей процес повторюється доти, доки значення функції витрат не буде мінімізовано.

Тренування нашої нейронної мережі триває близько 11 хвилин на локальному комп'ютері та 9 хвилин в середовищі розробки Google Colab. Тренування проводилося протягом 100 епох з розміром пакету (batch size) рівним 1024.

Після успішного тренування, визначення результату на одній точці даних займає приблизно 0.3 мілісекунди.

2.6 Аналіз та оцінка результатів

Після тренування фінальна точність нашої моделі становить близько 96%. Це доволі хороший результат, враховуючи, що тренування займає 10 хвилин.

Представлені графіки демонструють тренування нейронної мережі за 100 епох. Показано графік точності (accuracy) та втрат (loss). Графік точності (Рисунок 16) показує, наскільки добре модель працює на тренувальному та тестовому наборах зі збільшенням кількості епох. Ми бачимо тенденцію до зростання в обох лініях з мінімальною різницею між ними. Це вказує на відсутність перенавчання. Графік втрат (Рисунок 17) показує, як функція втрат змінюється протягом тренування для обох наборів. Ми бачимо, що обидві лінії зменшуються, що означає, що модель стає точнішою у своїх прогнозах.

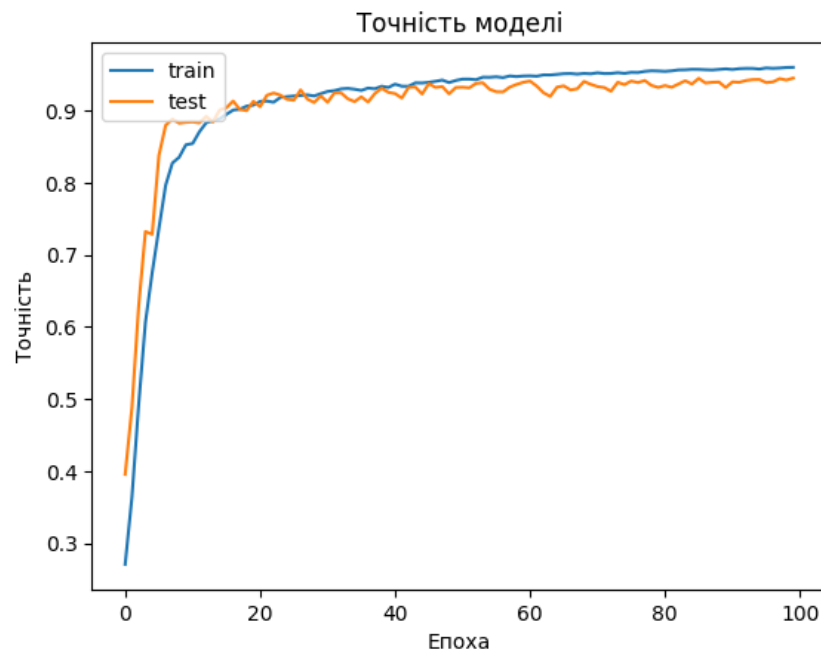


Рисунок 16 – Точність натренованої моделі.

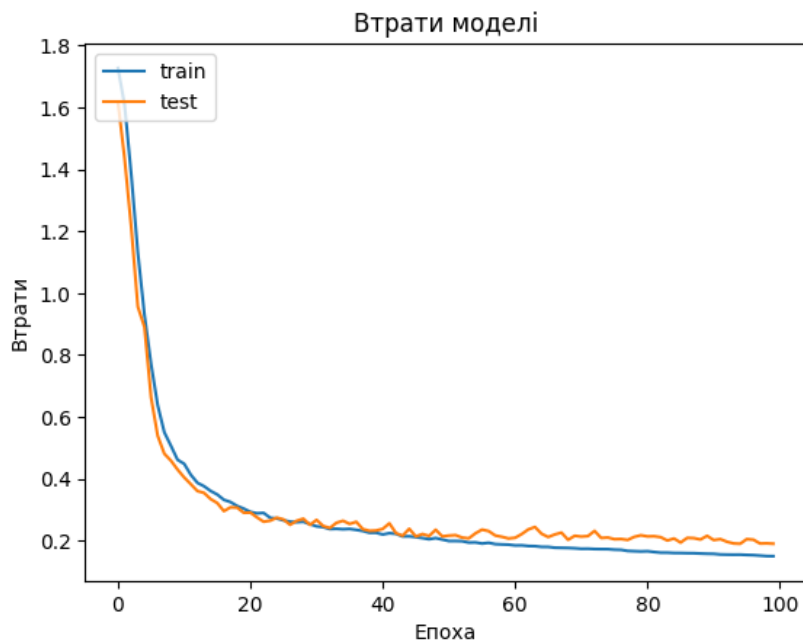


Рисунок 17 – Втрати натренованої моделі.

Далі ми показуємо матрицю невідповідностей, або матрицю помилок [25]. Вона надає деталізований аналіз точності моделі. Матриця невідповідностей показує кількість хибно позитивних, хибно негативних, істинно позитивних та істинно негативних результатів для сукупності класів класифікації. В нашому випадку це являє собою матрицю 6×6 .

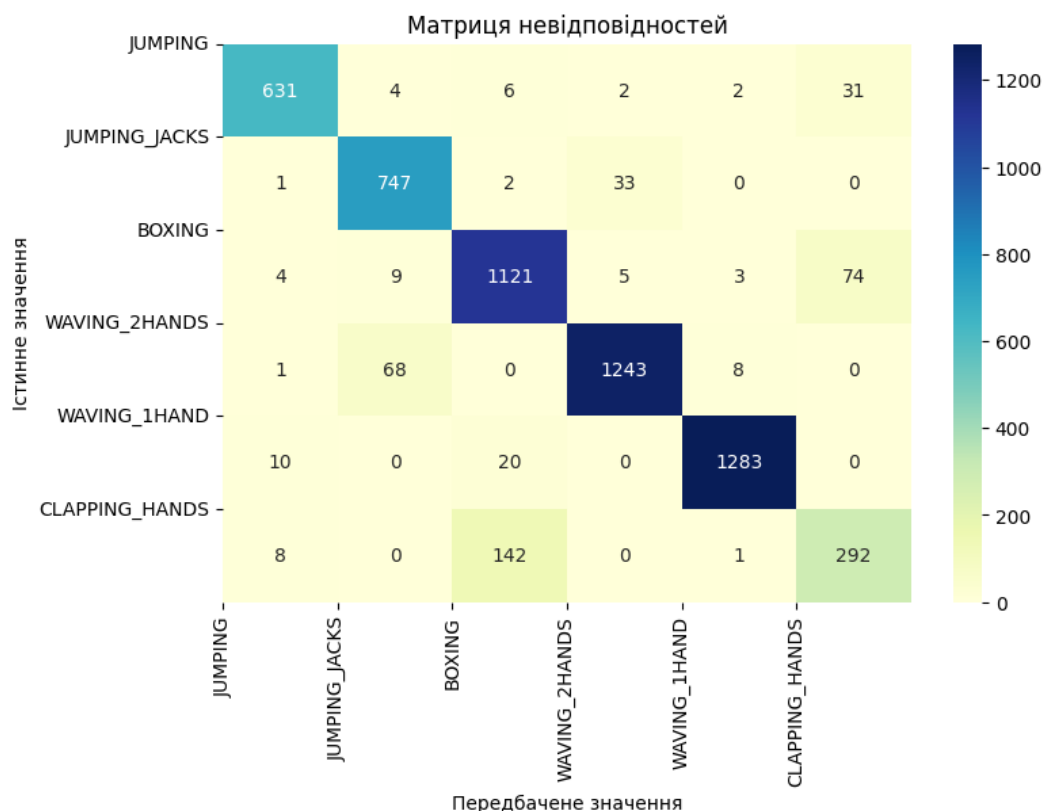


Рисунок 18 – Матриця невідповідностей.

Як бачимо зі значень матриці невідповідностей, є помітні невідповідності між парами класів «Плескання долонями» та «Бокс», та між «Махання двома долонями» та «Стрибками з піднятими руками» (Jumping Jacks). Дані результати є цілком зрозумілими через явну схожість цих пар рухів.

Результати вказують на те, що модель добре застосовна до більших наборів даних з більшою кількістю різноманітних рухів, при умові що є доступ до декількох різних кутів зору.

2.7 Подальші дослідження

Один з можливих видів застосування тренованої моделі оцінки діяльності це інтеграція моделі з живою відеокамерою. Це дозволить аналізувати рухи тіла

в реальному часі використовуючи звичайні двовимірні RGB камери замість спеціалізованих тривимірних камер захоплення руху.

Також подальші дослідження можуть розглядати більш схожі класи активності – наприклад, біг та ходьба, спокійний рух та збуджений рух.

ВИСНОВКИ

В цій роботі було досліджено потенціал методів машинного навчання для взаємодії в реальному часі з просторовими обчислювальними системами. Головною метою був аналіз існуючих технологій просторових обчислень і розробка моделі машинного навчання для двовимірного розпізнавання діяльності людини. Ми почали з огляду літератури про просторові обчислювальні системи, включаючи їх компоненти, застосування та проблеми. Потім ми обговорили виявлення та оцінку руху, які є ключовими технологіями для просторових обчислювальних систем, і представили деякі з найпопулярніших методів, що використовуються в цій галузі.

Далі ми зосередилися на розпізнаванні людської діяльності, яка є критично важливим компонентом багатьох просторових обчислювальних програм. Ми описали різні типи людської діяльності та труднощі їх точного розпізнавання в режимі реального часу. Ми також переглянули найсучасніші методи розпізнавання людської діяльності, включаючи як традиційні підходи, так і підходи на основі машинного навчання.

Нарешті, ми представили нашу модель машинного навчання для двовимірного розпізнавання людської діяльності, яка базується на фреймворку Keras. Ми описали архітектуру моделі, набір даних, який використовується для навчання та тестування, а також показники оцінки, які використовуються для вимірювання її продуктивності. Ми також обговорили результати наших експериментів, які показали, що наша модель досягла високої точності в розпізнаванні різних видів людської діяльності в режимі реального часу.

Загалом ця робота демонструє потенціал методів машинного навчання для взаємодії в реальному часі з просторовими обчислювальними системами. Аналізуючи існуючі технології просторових обчислень і розробляючи модель

машинного навчання для двовимірного розпізнавання людської діяльності, ми зробили внесок у розвиток цієї галузі. Наша робота може мати багато практичних застосувань, включаючи розробку розумних будинків, систем охорони здоров'я та систем безпеки, які можуть точно й ефективно розпізнавати людську діяльність.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

- [1] S. Greenwold, *Spatial Computing*, Massachusetts Institute of Technology, 2003.
- [2] S. Balakrishnan, M. S. Hameed, K. Venkatesan та G. Aswin, «Interaction of Spatial Computing In Augmented Reality,» в *2021 7th International Conference on Advanced Computing and Communication Systems*, 2021.
- [3] A. E. Conrady, «Decentred Lens-Systems,» *Monthly Notices of the Royal Astronomical Society*, № 79, pp. 384-390, 03 1919.
- [4] D. Brown, «Decentering distortion of lenses,» 1966.
- [5] J. Kannala та S. Brandt, «A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses,» *IEEE Transactions on Pattern Analysis and Machine Intelligence*, № 28, pp. 1335-1340, 2006.
- [6] B. Lucas та T. Kanade, «An Iterative Image Registration Technique with an Application to Stereo Vision,» № 81, pp. 121-130, 04 1981.
- [7] G. Farnebäck, «Two-Frame Motion Estimation Based on Polynomial Expansion,» в *Image Analysis*, Berlin, Heidelberg, Springer Berlin Heidelberg, 2003, pp. 363-370.
- [8] P. KaewTraKulPong та R. Bowden, «An Improved Adaptive Background Mixture Model for Real-time Tracking with Shadow

- Detection,» в *Video-Based Surveillance Systems: Computer Vision and Distributed Processing*, Boston, MA, Springer US, 2001, pp. 135-144.
- [9] Z. Zivkovic, «Improved adaptive Gaussian mixture model for background subtraction,» в *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004*, 2004.
- [10] Z. Z. Heijden та F. v. der, «Efficient adaptive density estimation per image pixel for the task of background subtraction,» *Pattern Recognition Letters*, № 27, pp. 773-780, 2006.
- [11] A. B. Godbehere, A. Matsukawa та K. Goldberg, «Visual Tracking of Human Visitors under Variable-Lighting Conditions for a Responsive Audio Art Installation,» в *2012 American Control Conference (ACC)*, 2012.
- [12] L. Piyathilaka та S. Kodagoda, «Gaussian mixture based HMM for human daily activity recognition using 3D skeleton features,» в *2013 IEEE 8th Conference on Industrial Electronics and Applications (ICIEA)*, 2013.
- [13] L. Piyathilaka та S. Kodagoda, «Human Activity Recognition for Domestic Robots,» 12 2013.
- [14] J. W. Hu, Y. Chen, S. Hao, X. Peng та Lisha, «Deep learning for sensor-based activity recognition: A survey,» *Pattern Recognition Letters*, № 119, pp. 3-11, 03 2019.
- [15] M.-W. Lee, A. Khan та T.-S. Kim, «A single tri-axial accelerometer-based real-time personal life log system capable of human activity recognition and exercise information generation,» *Personal and Ubiquitous Computing*, № 15, pp. 887-898, 12 2011.

- [16] L. M. Dang, K. Min, H. Wang, M. Piran, H. Lee та H. Moon, «Sensor-based and vision-based human activity recognition: A comprehensive survey,» *Pattern Recognition*, № 108, 07 2020.
- [17] I. E. Jaramillo, J. G. Jeong, P. R. Lopez, C.-H. Lee, D.-Y. Kang, T.-J. Ha, J.-H. Oh, H. Jung, J. H. Lee, W. H. Lee та T.-S. Kim, «Real-Time Human Activity Recognition with IMU and Encoder Sensors in Wearable Exoskeleton Robot via Deep Learning Networks,» *Sensors*, № 22, 2022.
- [18] Y. Z. Gong, R. Yang, G. Chevalier та Maoguo, «Deep Residual Bidir-LSTM for Human Activity Recognition Using Wearable Sensors,» *CoRR*, 2017.
- [19] V. R. Guido та F. L. Drake Jr, Python reference manual, Centrum voor Wiskunde en Informatica Amsterdam, 1995.
- [20] F. a. o. Chollet, «Keras,» GitHub, 27 03 2015. [Онлайновый]. Available: <https://github.com/keras-team/keras>.
- [21] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei та Y. Sheikh, «OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields,» *IEEE Transactions on Pattern Analysis and Machine Intelligence*, № 43, pp. 172-186, 2019.
- [22] S. Hochreiter та J. Schmidhuber, «Long short-term memory,» *Neural Computation*, т. 9, № 8, p. 1735–1780, 1997.
- [23] S. Hochreiter, «Untersuchungen zu dynamischen neuronalen Netzen,» 04 1991.

- [24] F. Gers, J. Schmidhuber та F. Cummins, «Learning to Forget: Continual Prediction with LSTM,» *Neural computation*, № 12, pp. 2451-71, 10 2000.
- [25] S. V. Stehman, «Selecting and interpreting measures of thematic classification accuracy,» *Remote Sensing of Environment*, № 62, pp. 77-89, 1997.

ДОДАТОК А

Фрагмент коду, написаний мовою програмування Python, що включає архітектуру нейронної мережі та етап її тренування.

```
# Input Data
training_data_count = len(X_train)
test_data_count = len(X_test)
n_input = len(X_train[0][0])
n_hidden = 34
n_classes = 6
learning_rate = 0.0025
decay_rate = 0.02
lambda_loss_amount = 0.0015
training_epochs = 100
batch_size = 1024

model = Sequential(
    [
        # relu activation
        layers.Dense(
            n_hidden,
            activation="relu",
            kernel_initializer="random_normal",
            bias_initializer="random_normal",
            batch_input_shape=(batch_size, n_steps, n_input),
        ),
        layers.LSTM(n_hidden, return_sequences=True, unit_forget_bias=1.0),
        layers.LSTM(n_hidden, unit_forget_bias=1.0),
        layers.Dense(
            n_classes,
            kernel_initializer="random_normal",
            bias_initializer="random_normal",
            kernel_regularizer=regularizers.l2(lambda_loss_amount),
            bias_regularizer=regularizers.l2(lambda_loss_amount),
            activation="softmax",
        ),
    ],
)
```

```
]
)

model.compile(
    optimizer=optimizers.legacy.Adam(lr=learning_rate, decay=decay_rate),
    metrics=["accuracy"],
    loss="categorical_crossentropy",
)
y_train_one_hot = keras.utils.to_categorical(y_train, 6)
y_test_one_hot = keras.utils.to_categorical(y_test, 6)

train_size = X_train.shape[0] - X_train.shape[0] % batch_size
test_size = X_test.shape[0] - X_test.shape[0] % batch_size

start = time.time()
history = model.fit(
    X_train[:train_size, :, :],
    y_train_one_hot[:train_size, :],
    epochs=training_epochs,
    batch_size=batch_size,
    validation_data=(X_test[:test_size, :, :], y_test_one_hot[:test_size,
:]),
)
print(f"finished in {time.time()-start} s.")
```