Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційних радіотехнологій і технічного захисту інформації
(повна назва)
Кафедра Радіотехнологій інформаційно-комунікаційних систем
(повна назва)

# КВАЛІФІКАЦІЙНА РОБОТА
# Пояснювальна записка

рівень вищої освіти **другий (магістерський)**

## ДОСЛІДЖЕННЯ МЕТОДІВ ДЛЯ КОНТЕКСТНО-ЗАЛЕЖНОГО РОЗПІЗНАВАННЯ ДІЯЛЬНОСТІ ЛЮДИНИ З ВИКОРИСТАННЯМ ГІБРИДНИХ МОДЕЛЕЙ ГЛИБОКОГО НАВЧАННЯ

(тема)

Виконав:
студент **II** курсу, групи **АПСм-22-1**
**Д'яченко М. О.**
(прізвище, ініціали)
Спеціальність **126 Інформаційні системи та технології**
(код і повна назва спеціальності)
Тип програми **освітньо-професійна**

Освітня програма **Архітектурне проєктування інформаційних систем**
( повна назва освітньої програми)
Керівник **професор Кузьомін О. Я.**
(посада, прізвище, ініціали)

Допускається до захисту
В.о. зав. кафедри РТІКС _____ **Зарудний О.А..**
(підпис) (прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційних радіотехнологій і технічного захисту інформації

Кафедра Радіотехнологій інформаційно-комунікаційних систем

Рівень вищої освіти другий (магістерський)

Спеціальність 126 Інформаційні системи та технології
(код і повна назва)

Тип програми Освітньо-професійна

Освітня програма Архітектурне проєктування інформаційних систем
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«____»_____ 2023 р.

# ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові **Д'ЯЧЕНКО МИКИТІ ОЛЕКСАНДРОВИЧУ**
(прізвище, ім'я, по батькові)

1.Тема роботи **ДОСЛІДЖЕННЯ МЕТОДІВ ДЛЯ КОНТЕКСТНО-ЗАЛЕЖНОГО РОЗПІЗНАВАННЯ ДІЯЛЬНОСТІ ЛЮДИНИ З ВИКОРИСТАННЯМ ГІБРИДНИХ МОДЕЛЕЙ ГЛИБОКОГО НАВЧАННЯ**

затверджена наказом по університету від **03 листопада 2023 р**. **№ 1295 Ст**

2. Термін подання студентом роботи до екзаменаційної комісії **10 січня 2024 р.**

3. Вихідні дані до роботи _____
3.1 Провести огляд та аналіз аналогічних систем
3.2 Розробити алгоритм гібридної моделі глибокого навчання
3.3 Розробити програмне забезпечення для гібридної моделі

4. Перелік питань, що потрібно опрацювати в роботі _____
Вступ. 1 Огляд та аналіз аналогічних систем. 2 Розробка гібридної моделі глибокого навчанняі. 3 Розробка програмного забезпечення. Висновки. Перелік джерел посилання. Додатки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) (п.5 включається до завдання за рішенням випускової кафедри)_____
Слайди комп'ютерної презентації_____
_____
_____
_____

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1 )

| Найменування розділу | Консультант (посада, прізвище, ім'я, по батькові) | Позначка консультанта про виконання розділу | |
|---|---|---|---|
| | | підпис | дата |
| | | | |
| | | | |

## КАЛЕНДАРНИЙ ПЛАН

| № | Назва етапів роботи | Терміни виконання етапів роботи | Примітка |
|---|---|---|---|
| 1 | Вступ | 03.10-06.10.2023 | виконано |
| 2 | Огляд та аналіз аналогічних систем | 09.10-04.11.2023 | виконано |
| 3 | Розробка алгоритму гібридної моделі | 06.11-10.11.2023 | виконано |
| 4 | Вибір та обґрунтування засобів розробки | 13.11-17.11.2023 | виконано |
| 5 | Розробка програмного забезпечення | 20.11-29.11.2023 | виконано |
| 6 | Висновки | 30.11.2023 | виконано |
| 7 | Оформлення пояснювальної записки | 22.12.2023 | виконано |
| 8 | Оформлення ілюстрацій | 29.12.2023 | виконано |
| 9 | Представлення роботи на кафедру | 10.01.2024 | виконано |

Дата видачі завдання **03 листопада 2023 р.**

Студент _____ М.О. Д'яченко
(підпис)

Керівник роботи _____ проф. О.Я. Кузьомін

# РЕФЕРАТ

Пояснювальна записка до практики: 99 с., 26 рис., 3 табл., 39 джерел, 3 додатка.

КОНТЕКСТ. РОЗПІЗНОВАННЯ. ДІЯЛЬНІСТЬ. ІНТЕЛЕКТ. ТЕХНОЛОГІЯ. МОДЕЛЬ. НАВЧАННЯ. ПАМ'ЯТЬ. МЕРЕЖА.

Об'єктом дослідження є процес розпізнавання людської діяльності на основі різних видів даних.

Предметом дослідження є методи і моделі контекстно-залежного розпізнавання людської діяльності.

Метою роботи є дослідження методів контекстно-залежного розпізнавання людської діяльності за допомогою гібридних моделей глибокого навчання.

Для розв’язання поставлених задач у даній кваліфікаційній роботі використано: реалізацію архітектур CNN, LSTM, техніки пре-обробки зображень, механізм уваги.

На основі літературних джерел розглянуто сучасний стан проблеми, проаналізовано те, які методи та моделі використовуються для контекстно-залежного розпізнавання людської діяльності та сформовано постановку задачі.

Наступним кроком було розглянуто існуючі підходи та фреймворки для вирішення задач контекстно-залежного розпізнавання людської діяльності на основі даних відео.

Для кожного із розглянутих підходів у виді програмного застосунку було розроблено модель. Ці моделі було навчено на датасеті UCF-101 та кожен із результатів навчання було проаналізовано й порівняно між собою.

У результаті роботи було зроблено висновки та зроблено рекомендації для подальшого дослідження теми.

# ABSTRACT

Explanatory note to practice: 99 p., 26 figures, 3 tables, 39 sources, 3 appendixes

CONTEXT. RECOGNITION. ACTIVITY. INTELLIGENCE. TECHNOLOGY. MODEL. TEACHING. MEMORY. NETWORK.

The object of research is the process of recognizing human activity based on various types of data.

The subject of research is methods and models of context-dependent recognition of human activity.

The purpose of the work is to research methods of context-dependent recognition of human activity using hybrid models of deep learning.

To solve the problems in this qualification work, the following are used: implementation of CNN, LSTM architectures, image pre-processing techniques, attention mechanism.

Based on literary sources, the current state of the problem is considered, which methods and models are used for context-dependent recognition of human activity are analyzed, and the formulation of the problem is formed.

The next step was to consider existing approaches and frameworks for solving the problems of context-dependent recognition of human activity based on video data.

For each of the considered approaches, a model was developed in the form of a software application. These models were trained on the UCF-101 dataset and each of the training results was analyzed and compared with each other.

As a result of the work, conclusions were made and recommendations were made for further research on the topic.

ЗМІСТ

# ПЕРЕЛІК ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

IoT (Internet of Things) – інтернет речей;

AR (Augmented reality) – доповнена реальність;

VR (Virtual reality) – віртуальна реальність;

HAR (Human activity recognition) – розпізнавання діяльності людини;

CAHAR (Context-aware human activity recognition) – контекстно-залежне розпізнавання діяльності людини;

CNN (Convolutional Neural Networks) – згорткові нейронні мережі;

RNN (Recurrent Neural Networks) – рекурентні нейронні мережі;

LSTM (Long Short-Term Memory) – довга короткочасна пам'ять;

NLP (Natural language processing) – обробка природної мови;

ПЗ – Програмне забезпечення;

АЗ – Апаратне забезпечення;

AUC (Area under the curve) – площа під кривою;

RAM (Random-access memory) – оперативна пам'ять;

GPU (Graphics processing unit) – графічний процесор;

CPU (Central processing unit) – центральний процесор.

# ВСТУП

Швидкий розвиток сенсорних технологій, Інтернету речей (IoT) і штучного інтелекту проклав шлях для більш розумних і адаптивних систем, здатних розуміти поведінку людини. Однією з найбільш впливових областей у цій галузі є розпізнавання людської діяльності (HAR). Традиційні системи HAR є ефективними у розпізнаванні основних дій, але часто цього не вистачає, коли справа доходить до розуміння контексту, в якому ці дії відбуваються. Це обмеження є суттєвим, оскільки значення та наслідки діяльності можуть різко змінюватися залежно від контексту [1]. Приведемо декілька прикладів:

1. “Біг” може означати спортивні вправи на вулиці або в спортзалі, але може означати надзвичайну ситуацію в офісній будівлі, де виникла пожежа.
2. Детектор руху у спальні вночі (або в ще темній кімнаті) може означати, що людина прокинулась та їй необхідно включити тускле світло біля полу, щоб вона змогла зорієнтуватися у просторі.

Таких прикладів може бути безліч та в однієї галузі розумних будинків таких прикладів можна придумати тисячі, бо кожна людина унікальна та їй можуть бути необхідні різні речі та сценарії поведінки.

Аналізуючи ці приклади, можемо прийти до висновку, що великого значення набуває тема дослідження методів контекстно-залежного розпізнавання людської діяльності за допомогою гібридних моделей глибокого навчання. Гібридні моделі глибокого навчання, які поєднують різні архітектури або методи нейронної мережі, пропонують багатообіцяючий шлях для охоплення більшості нюансів зв’язку між людською діяльністю та її контекстом. Ці моделі можуть інтегрувати дані з кількох датчиків (наприклад, використовуючи звичайний телефон користувача [1]…[2]) і враховувати часові послідовності або деякі зв’язки, забезпечити таким чином більш цілісне розуміння діяльності в різних середовищах. Це має різні застосування, починаючи від сфери моніторингу здоров’я та закінчуючи системами реагування на надзвичайні ситуації.

Актуальність цієї теми дослідження підкреслюється зростаючою інтеграцією розумних пристроїв у наше повсякденне життя та зростаючим попитом на персоналізовані контекстно-залежні послуги. Незалежно від того, чи це розумні будинки, які регулюють налаштування на основі дій мешканців, чи системи охорони здоров'я, що забезпечують моніторинг у реальному часі та сповіщення про стан здоров'я, вирішення представлених задач є важливим й іноді критичним для сучасного світу. Тому розробка надійних і ефективних методів для контекстно-залежного HAR є необхідною для розвитку дуже різних секторів технологій.

Метою цієї роботи є дослідження методів контекстно-залежного розпізнавання людської діяльності за допомогою гібридних моделей глибокого навчання.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

## 1.1 Опис об'єкта проектування

Об'єктом проектування в цьому дослідженні є система контекстно-залежного розпізнавання людської діяльності (HAR), яка використовує гібридні моделі глибокого навчання для аналізу та класифікації різних видів діяльності на основі контекстних даних. Мета HAR — зрозуміти та визначити, якою діяльністю займається людина, обробляючи дані від таких датчиків, як акселерометри, гіроскопи, магнітометри, камери, мікрофони та інші пристрої.

Системи HAR мають широкий спектр застосувань, зокрема:

1. Охорона здоров’я: моніторинг і оцінка фізичної активності пацієнтів, людей похилого віку або людей із певними захворюваннями. Це може допомогти виявляти падіння, відстежувати прогрес фізичної терапії або надавати відгуки про повсякденну діяльність.
2. Відстеження фізичної активності: системи HAR зазвичай використовуються в переносних фітнес-пристроях для відстеження та запису таких дій, як ходьба, біг, їзда на велосипеді або плавання.
3. Безпека: виявлення та сповіщення органів влади про підозрілі або несанкціоновані дії, включаючи вторгнення, вандалізм або інші події, пов’язані з безпекою.
4. Розумні будинки: автоматизація різних процесів і послуг у домашньому середовищі на основі виявлених дій, таких як увімкнення світла, коли хтось входить у кімнату, або налаштування термостата.
5. Доповнена реальність і віртуальна реальність: покращення взаємодії з додатками AR і VR шляхом реагування на дії та рухи користувача в реальному світі.
6. Спортивний аналіз: Аналіз рухів і дій спортсменів під час спортивних тренувань і оцінка продуктивності.
7. Промисловість і виробництво: моніторинг і підвищення безпеки та

ефективності працівників у промислових умовах.

Системи HAR можуть використовувати різні методи та датчики, такі як машинне навчання, глибоке навчання та обробка сигналів, для обробки та аналізу даних, зібраних із датчиків. Дані акселерометрів і гіроскопів у смартфонах і пристроях, що носяться, зазвичай використовуються для розпізнавання активності завдяки їх доступності та здатності фіксувати дані про рух і орієнтацію.

Розробка систем HAR часто передбачає навчання моделей машинного навчання на позначених даних для класифікації дій. Ці моделі можуть варіюватися від традиційних алгоритмів машинного навчання (наприклад, дерева рішень, опорні векторні машини) до більш просунутих методів глибокого навчання, таких як згорткові нейронні мережі (CNN) і рекурентні нейронні мережі (RNN). Вибір моделі та методів вилучення ознак залежить від конкретного застосування та характеристик даних.

## 1.2 Загальні відомості

### 1.2.1 Машинне навчання

Машинне навчання – це галузь інформатики та комп’ютерних наук, що використовує статистичні методи, для того, щоб дати можливість комп’ютерним системам “вчитися” на тестових даних, без явного програмування алгоритмів, для вирішення подальших задач без втручання людини.

Також, машинне навчання використовують здебільшого для вирішення складних проблем, та проблем, які потребують адаптації. Можна сказати, що це такий клас завдань, який неможливо вирішити якимось певним, чітким алгоритмом, та при цьому потрібно зважати на вже отримані дані, результати. Далі, наведемо декілька прикладів таких задач.

1) Завдання, які виконуються людиною або твариною: є безліч задач, які ми виконуємо регулярно, але аналіз того, як ми їх виконуємо недостатньо

продуманий для того, щоб визначити чіткий алгоритм цих дій. Серед таких задач, можна виділити водіння, розпізнавання образів або мови тощо. Усі програми, що використовують машинне навчання для вирішення подібних задач, досягають непоганих результатів, коли навчаються на великій кількості тренувальних даних;

2) Завдання, які виходять за межі людських можливостей: це ще один широкий клас задач, що отримують користь від використання машинного навчання. Вони тісно пов’язані з аналізом та використанням великих та складних масивів даних. Це можуть бути: астрономічні дані, дані для прогнозування курсу валют, двигун для пошуку інформації у веб, перенесення медичних даних, архівів у медичні знання тощо. Серед великої кількості цієї інформації є така, що є важливою для людини, але людина не здатна самостійно віднайти цю інформацію, бо даних надто багато і зазвичай ці дані дуже складні. Однією з перспективних областей є вміння визначати значущі моделі у великих та складних наборах даних, яке в поєднанні з програмами, що навчаються з необмеженою кількістю пам’яті та постійно зростаючою швидкістю обробки даних, сильно полегшує роботу з даним типом задач;

Тож, метою машинного навчання є передбачення результату за вхідними даними. Чим різноманітнішими будуть вхідні дані, тим простіше машині знайти закономірності і тим точнішим буде результат. Для того, щоб навчити машину потрібно 3 речі:

1) Дані. Для того, щоб виявити спам – потрібно мати приклади спам-листів, щоб передбачити курс акцій – потрібно мати історію цін, щоб дізнатися інтереси користувача – потрібні його пости у соціальних мережах та його лайки. Даних потрібно чим більше, бо чим їх буде більше – тим кращим буде фінальний результат. Дані збирають по-різному, дехто збирає їх вручну – це тривалий процес, але хоч даних менше, зате ці дані будуть без помилок. Інші збирають дані в автоматичному режимі – віддають машині всю знайдену інформацію та розраховують на хороший результат. Найхитріші – компанії, типу Google, використовують своїх користувачів для навчання подібних систем. Згадайте ту

ж саму ReCaptch-у, що вимагає від користувача вказати на картинки, на яких зображено дорожні знаки, автомобілі, тощо.

2) Ознаки. Їх ще називають фічами (features). Ознаки, властивості, характеристики – це може бути що завгодно – ціна акцій, стать користувача, пробіг мотоцикла, навіть той же ітератор частоти появи певного слова в тексті. Машина повинна конкретно знати, на що їй дивитися, що шукати. Добре мати потрібні дані, наприклад в табличках в базі даних, де назви колонок і будуть фічами, тобто ознаками чи характеристиками. А що, якщо у нас є п'ятдесят гігабайтів картинок з собаками, які мають велику кількість характеристик? У випадку, коли ознак надто багато, модель працюватиме дуже повільно та неефективно. Відбір правильних ознак, найчастіше займає більшу кількість часу при навчанні подібних систем. Також, бувають зворотні ситуації, коли користувач вирішує самотужки вибрати "правильні" на його думку характеристики, що призводить до того, що система починає неправильно працювати та помилятися.

3) Алгоритм. Зазвичай, одну й ту ж саму задачу можна вирішити різними методами або способами. Від вибору методу буде залежати швидкість, точність роботи та головне – розмір готової моделі. Але завжди потрібно зважати на одну річ: якщо тестові дані – "сміття", то в даному випадку ніякий алгоритм не допоможе. Чим більше буде "правильних" тестових даних, тим більшою буде імовірність отримати задовільний результат [7].

Існують декілька типів машинного навчання.

Машинне навчання із вчителем. Фахівці по роботі з даними надають алгоритмам позначені та певні навчальні дані для оцінки кореляцій. Демонстраційні дані визначають як вхідні дані, і вихідні дані алгоритму. Наприклад, зображення рукописних цифр анотуються, щоб вказати, якій кількості вони відповідають. Система навчання з вчителем може розпізнавати кластери пікселів та фігур, пов'язаних з кожним числом, за наявності достатньої кількості прикладів. Згодом система розпізнає написані руки цифри, стабільно розрізняючи числа 9 і 4 чи 6 і 8.

Сильні сторони машинного навчання з учителем – простота та легкість структури. Така система корисна при прогнозуванні можливого обмеженого набору результатів, поділ даних на категорії або об'єднанні результатів двох інших алгоритмів машинного навчання. Однак маркування мільйонів немаркованих наборів даних є складним завданням. Давайте розглянемо це докладніше.

Маркування даних – це процес категоризації вхідних даних із відповідними ним певними вихідними значеннями. Позначені навчальні дані необхідні навчання з учителем. Наприклад, мільйони зображень яблук і бананів мають бути позначені словами "яблуко" або "банан". Потім програми машинного навчання могли б використовувати ці навчальні дані, щоб вгадувати назву фрукта зображення фрукта. Однак маркування мільйонів нових даних може бути трудомістким та складним завданням. Сервіси колективної роботи, такі як Amazon Mechanical Turk, можуть певною мірою подолати це обмеження алгоритмів навчання з учителем. Ці послуги забезпечують доступ до великої кількості доступних робочих ресурсів у світі, що полегшує збирання даних [8].

Машинне навчання без вчителя. Алгоритми навчання без вчителя навчаються на нерозмічених даних. Такі алгоритми переглядають нові дані, намагаючись встановити значущі зв'язки між вхідними та наперед визначеними вихідними даними. Вони можуть виявляти закономірності та класифікувати дані. Наприклад, алгоритми без вчителя можуть групувати статті новин з різних новинних веб-сайтів у загальні категорії, такі як спорт, кримінал і т. д. Вони можуть використовувати обробку природної мови для розуміння сенсу та емоцій у статті. У роздрібній торгівлі навчання без вчителя допоможе знайти закономірності у покупках клієнтів та надати результати аналізу даних, такі як: покупець, швидше за все, купить хліб, якщо також купить олію.

Навчання без вчителя корисне для розпізнавання образів, виявлення аномалій та автоматичного групування даних за категоріями. Оскільки навчальні дані не вимагають маркування, налаштування просте. Ці алгоритми також можна використовувати для автоматичного очищення та обробки даних для подальшого

моделювання. Обмеження цього у тому, що не може дати точних прогнозів. З іншого боку, він може самостійно виділяти конкретні типи вихідних даних [8].

Машинне навчання із частковим залученням вчителя. Як випливає з назви, цей метод поєднує у собі навчання з учителем і без нього. Цей метод ґрунтується на використанні невеликої кількості розмічених даних та великої кількості нерозмічених даних для навчання систем. Спочатку розмічені дані використовуються для часткового навчання алгоритму машинного навчання. Після цього частково навчений алгоритм сам розмічає нерозмічені дані. Цей процес називається псевдомаркуванням. Потім модель перенавчається на результуючому наборі даних без програмування.

Перевага цього методу в тому, що вам не потрібні великі обсяги даних. Це зручно при роботі з такими даними, як довгі документи, читання та маркування яких забирає надто багато часу в людини [8].

Навчання з підкріпленням. Навчання з підкріпленням – це метод, у якому значення винагороди прив'язані до різних кроків, які мають пройти алгоритм. Таким чином, мета моделі – накопичити якнайбільше призових балів і зрештою досягти кінцевої мети. Більшість практичного застосування навчання з підкріпленням за останнє десятиліття була пов'язана з відеоіграми. Передові алгоритми навчання з підкріпленням досягли вражаючих результатів у класичних та сучасних іграх, часто значно перевершуючи ручні аналоги.

Хоча цей метод найкраще працює у невизначених та складних середовищах даних, він нечасто застосовується у бізнес-контексті. Це неефективно для чітко визначених завдань і упередженість розробників може вплинути на результати. Оскільки спеціаліст із роботи з даними розробляє нагороди, вони можуть впливати на результати [8].

### 1.2.2 Глибоке навчання

Нейронна мережа – це методологія у сфері штучного інтелекту, яка вчить комп'ютери обробляти дані у такий самий спосіб, як і людський мозок. Моделі

глибокого навчання можуть розпізнавати складні закономірності у зображеннях, тексті, звуках та інших даних для отримання точних відомостей та прогнозів. Методи глибокого навчання можна використовувати для автоматизації завдань, які зазвичай вимагають застосування людського інтелекту, таких як опис зображень або перетворення звукового файлу на текст [9].

Традиційне машинне навчання вимагає значної взаємодії людини через конструювання ознак отримання результатів. Наприклад, якщо ви навчаєте модель машинного навчання класифікувати зображення котів та собак, вам необхідно вручну налаштувати її для розпізнавання таких рис, як форма очей, хвоста, вух, контури носа тощо.

Оскільки метою машинного навчання є зниження необхідності втручання людини, методи глибокого навчання позбавляють людей необхідності маркувати дані на кожному етапі.

Хоча глибоке навчання існує вже багато десятиліть, на початку 2000-х років такі вчені, як Ян ЛеКун, Йошуа Бенджіо та Джеффрі Хінтон, вивчили цю область докладніше. Дослідники вдосконалювали глибоке навчання, проте великі та складні набори даних у цей час були обмежені, а обчислювальні потужності, необхідні для навчання моделей – дорогими. За останні 20 років ці умови покращилися, і глибоке навчання стало комерційно вигідним [10].

Глибоке навчання – частина машинного навчання. Алгоритми глибокого навчання можна визначити як непросту і математично складну еволюцію алгоритмів машинного навчання [8].

Алгоритми глибокого навчання – це нейронні мережі, змодельовані на зразок людського мозку. Наприклад, у людському мозку є мільйони взаємозалежних нейронів, які спільно вивчають та обробляють інформацію. Так само нейронні мережі глибокого навчання, або штучні нейронні мережі, складаються з безлічі шарів штучних нейронів, які спільно працюють всередині комп'ютера.

Штучні нейрони – це програмні модулі, які називаються вузлами, які обробляють дані з використанням математичних обчислень. Штучні нейронні

мережі – це алгоритми глибокого навчання, які застосовують ці вузли на вирішення складних завдань [9]. Приклад того, як виглядають вузли нейронної мережі можна побачити на рисунку 1.1.

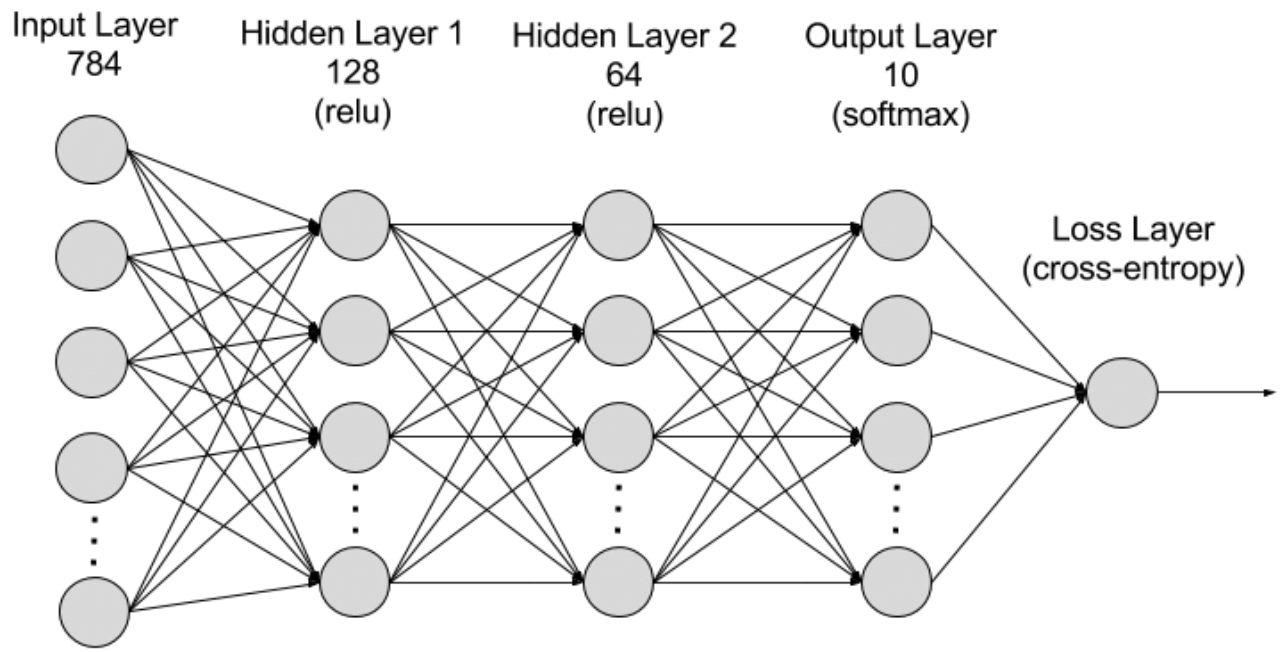


Рисунок 1.1 – Приклад вузлів нейронної мережі

Нижче наведені компоненти глибокої нейронної мережі.

Штучна нейронна мережа містить кілька вузлів, якими до неї надходять дані. Ці вузли становлять вхідний шар системи [9].

Вхідний шар обробляє та передає дані на шари, розташовані далі в нейронній мережі. Ці приховані шари обробляють інформацію на різних рівнях, адаптуючи свою поведінку з отриманням нових даних. Мережі глибокого навчання мають сотні прихованих верств, які можуть використовувати для аналізу проблеми з різних точок зору.

Наприклад, якщо вам дали зображення невідомої тварини, яку потрібно класифікувати, ви порівняли б її з тваринами, яких ви вже знаєте. Ви подивилися б на форму очей та вух, розмір, кількість ніг та малюнок вовни. Ви б спробували визначити закономірності, наприклад такі:

1) у тварини є копита, отже, це може бути корова чи олень;
2) у тварини котячі очі, отже, це може бути якась дика кішка.

Приховані шари у глибоких нейронних мережах працюють аналогічним чином. Якщо алгоритм глибокого навчання намагається класифікувати зображення тварини, кожен із її прихованих шарів обробляє різні характеристики тварини і намагається точно класифікувати його [9].

Вихідний шар складається з вузлів, що виводять дані. Моделі глибокого навчання, які виводять відповіді «так» чи «ні», мають на вихідному шарі лише два вузли. З іншого боку, моделі, які виводять ширший діапазон відповідей, містять більше вузлів [9].

Мережа глибокого навчання має такі переваги, порівняно з традиційним машинним навчанням:

– Ефективна обробка неструктурованих даних. Методи машинного навчання ускладнюють обробку неструктурованих даних, таких як текстові документи, оскільки навчальний набір даних може мати нескінченну кількість варіацій. З іншого боку, моделі глибокого навчання можуть розуміти неструктуровані дані та проводити загальні спостереження без отримання ознак вручну [9]. Наприклад, нейронна мережа може встановити, що дві різні вхідні пропозиції мають одне й те саме значення:

– Чи не підкажете як зробити оплату?

– Як мені переказати гроші?

– Приховані зв'язки та виявлення закономірностей. Додаток глибокого навчання може проводити поглиблений аналіз великих обсягів даних та виявляти нові ідеї, для пошуку яких він, можливо, навіть не був навчений. Наприклад, розглянемо модель глибокого навчання, яка навчена аналізувати споживчі покупки. У моделі є дані лише про товари, які ви вже придбали. Однак штучна нейронна мережа може пропонувати новинки, які ви не купували, порівнюючи ваші моделі покупок із моделями інших аналогічних клієнтів [9].

– Навчання без вчителя. Моделі глибокого навчання з часом можуть навчатися та покращуватися залежно від поведінки користувачів. Вони вимагають великих варіацій маркованих наборів даних. Наприклад, розглянемо нейронну мережу, яка автоматично виправляє або пропонує слова, аналізуючи

вашу поведінку під час набору тексту. Припустимо, що вона навчена англійської мови та може перевіряти орфографію англійських слів. Однак, якщо ви часто вводите слова не англійською мовою, як danke, нейронна мережа автоматично вчить і автоматично виправляє ці слова [9].

– Обробка нестійких даних. Нестійкі набори даних містять значні зміни. Один із прикладів – суми погашення кредиту у банку. Нейронна мережа з глибоким навчанням також може класифікувати та сортувати ці дані, наприклад, аналізуючи фінансові транзакції та позначаючи деякі з них для виявлення шахрайства [9].

### 1.2.3 Труднощі глибокого навчання

Оскільки глибоке навчання – відносно нова технологія, її практичне впровадження пов'язане із певними проблемами.

Великі обсяги високоякісних даних. Алгоритми глибокого навчання дають кращі результати, якщо навчати їх у великих обсягах високоякісних даних. Відхилення або помилки у вхідному наборі даних можуть суттєво вплинути на глибоке навчання. Наприклад, у нашому прикладі із зображенням тварин модель глибокого навчання може класифікувати літак як черепаху, якщо набір даних були випадково введені зображення, не пов'язані з тваринами [9].

Щоб уникнути таких неточностей, необхідно очистити та обробити великі обсяги даних, перш ніж навчати моделі глибокого навчання. Для попередньої обробки вхідних даних потрібно сховище великого обсягу.

Великі обчислювальні потужності. Для правильного функціонування алгоритмів глибокого навчання потрібні великі обчислювальні ресурси та інфраструктура з достатньою обчислювальною потужністю. В іншому випадку обробка результатів займає багато часу [9].

### 1.2.4 Розпізнавання діяльності людини (HAR)

Розпізнавання діяльності людини (HAR) — це галузь обчислювальної науки та техніки, яка намагається створити системи та методи, здатні автоматично розпізнавати та класифікувати дії людини на основі даних датчиків. Це здатність використовувати датчики для інтерпретації жестів або рухів людського тіла та визначення активності чи руху людини.

Системи HAR, як правило, контролюються або не контролюються, і їх можна використовувати в різних сферах діяльності, включаючи оздоровлення, легку атлетику, охорону здоров'я, безпеку, спортивні досягнення тощо.

Під час моделювання мета системи HAR полягає в тому, щоб спрогнозувати мітку дії людини на основі зображення чи відео, що зазвичай виконується за допомогою розпізнавання активності на основі відео та розпізнавання діяльності на основі зображень [21].

### 1.2.5 Контекстно-залежне розпізнавання діяльності людини (CAHAR) та їх приклади

Контекстно-залежне розпізнавання людської діяльності (CAHAR) – це область дослідження машинного навчання та штучного інтелекту, яка зосереджена на ідентифікації та розумінні людської діяльності, враховуючи не лише самі дії, але й контекст, у якому вони відбуваються. Цей підхід має на меті підвищити точність і релевантність систем розпізнавання активності шляхом інтеграції додаткових даних, які забезпечують контекст, таких як фактори навколишнього середовища, інформація про користувача або часові та просторові дані.

Прикладами таких систем можуть бути:

1. Розумні будинки: у розумному будинку системи CAHAR можуть використовувати датчики для виявлення таких дій, як приготування їжі, сон або перегляд телевізора. Система враховує час доби, розташування в будинку та інші фактори навколишнього середовища, такі як температура або рівень освітлення, щоб точно визначити активність.

2. Моніторинг охорони здоров’я. Носимі пристрої, оснащені датчиками, можуть відстежувати фізичну активність пацієнта (наприклад, ходьбу, стояння або лежання) і враховувати додаткову контекстну інформацію, як-от час доби, місцезнаходження чи історію здоров’я пацієнта, для кращого догляду за пацієнтом.

3. Фітнес-відстеження. Фітнес-трекери використовують CAHAR для точнішої інтерпретації фізичної активності, враховуючи контекст, наприклад різницю між бігом на вулиці та на біговій доріжці або ходьбою на дозвіллі та поспішною ходьбою, щоб встигнути на поїзд.

У контексті відеоданих системи CAHAR аналізують як візуальне представлення діяльності, так і навколишній контекст, щоб краще зрозуміти дії людини. Ось кілька конкретних прикладів:

1. Системи спостереження: у системі відеоспостереження системи CAHAR можуть розрізняти звичайні та підозрілі дії, враховуючи контекст. Наприклад, людина, що біжить, може бути звичайним заняттям у парку, але може викликати підозру в зоні обмеженого доступу.

2. Аналіз спорту: у спорті системи CAHAR на основі відео можуть аналізувати рухи гравців і враховувати контекст гри, наприклад позицію на полі, фазу гри або взаємодію з іншими гравцями, щоб забезпечити більш глибоке розуміння продуктивності.

3. Інтерактивні ігри: в інтерактивних іграх система CAHAR може використовувати відеовхід для виявлення рухів і жестів гравця, враховуючи контекст гри, щоб правильно інтерпретувати дії та покращити ігровий досвід.

4. Освітнє та навчальне середовище: CAHAR на основі відео можна використовувати в навчальних закладах для аналізу залученості студентів або в навчальних сценаріях для оцінки набуття навичок, беручи до уваги контекст навчального середовища.

Включаючи контекстну інформацію, системи CAHAR на основі відео пропонують більш складне та детальне розуміння людської діяльності, що веде до більш точних та ефективних застосувань у різних областях. Цей контекстний

підхід особливо важливий у складних середовищах, де дії можуть мати різні інтерпретації залежно від оточуючих обставин.

## 1.3 Основні кроки для розробки системи HAR

Розробка системи розпізнавання людської діяльності (HAR) зазвичай включає кілька етапів, включаючи збір даних, попередню обробку даних, виділення ознак, розробку моделі, навчання та оцінку. Нижче наведемо детальний опис цих етапів.

Збір даних є першим і одним з найважливіших кроків у розробці системи HAR. Дані можуть бути зібрані з різних джерел, включаючи датчики руху, акселерометри, гіроскопи та камери. Важливо забезпечити, щоб зібрані дані були репрезентативними та враховували всі можливі сценарії діяльності людини. Для цього може знадобитися збір даних в різних умовах та середовищах. Також важливо забезпечити правильну анотацію даних, щоб можна було використовувати їх для навчання та оцінки моделі [11].

Після збору даних вони повинні бути попередньо оброблені для видалення шуму, нормалізації, сегментації та інших процесів підготовки. Це допомагає підготувати дані до ефективного витягу функцій та класифікації. Попередня обробка може включати фільтрацію для видалення шуму, нормалізацію для приведення всіх даних до одного масштабу, а також сегментацію для розділення даних на менші частини, які легше аналізувати [12].

Витяг функцій є критичним етапом у розробці системи HAR. Цей процес включає в себе визначення та витяг характеристик з даних, які можуть бути використані для класифікації дій. Це може включати в себе статистичні показники, спектральний аналіз, вейвлет-перетворення та інші методи. Важливо вибрати такі характеристики, які найкраще відображають властивості діяльності, яку потрібно класифікувати [12].

Після витягу функцій можна приступити до розробки моделі машинного

навчання. Важливо вибрати відповідну модель та архітектуру, яка найкраще відповідає характеристикам даних та завданню класифікації. Гібридні моделі, які поєднують в собі різні типи нейронних мереж, такі як CNN та RNN, можуть бути використані для покращення ефективності та точності системи HAR, оскільки вони дозволяють ефективно обробляти як просторові, так і часові залежності в даних [13].

Наступним етапом є навчання моделі на позначених даних. Під час навчання модель вчиться розпізнавати різні види діяльності на основі вхідних даних та відповідних міток. Цей процес вимагає великої кількості даних та обчислювальних ресурсів [14].

Після навчання моделі необхідно оцінити її ефективність та точність. Для цього можна використовувати різні метрики, такі як точність, відгук, F1-оцінка тощо. Оцінка допомагає визначити, наскільки добре модель справляється з завданням класифікації та які аспекти моделі можуть бути покращені [15].

## 1.4 Існуючі архітектури для контекстно-залежного розпізнавання людської діяльності за допомогою моделей глибокого навчання

Оскільки вирішення проблеми вдосконалення контекстно-залежного розпізнавання людської діяльності стосується дуже багатьох галузь, то і готових методів та моделей існує багато. Дуже часто ці моделі вдосконалюються та налаштовуються під конкретну задачу [1][2][3]. Далі, проаналізуємо декілька основних архітектур, що лежать в основі побудування багатьох методів для контекстно-залежних HAR. Зазвичай гібридні моделі HAR можуть імплементувати та поєднувати в собі такі архітектури як CNN (Convolutional Neural Networks), RNN (Recurrent Neural Networks) або LSMT (Long Short-Term Memory).

### 1.4.1 Архітектура CNN (Convolutional Neural Networks)

Згорткові нейронні мережі (CNN) – це підвид глибоких нейронних мереж, переважно застосовуваний до розпізнавання об’єктів та класифікації зображень. Згорткові нейронні мережі сприймають та обробляють дані у вигляді тензорів, що дозволяє працювати з даними зображень у природній формі. Кожне вхідне зображення має як параметри ширину та довжину, а також глибину, яка визначається кодуванням зображення. Найбільш поширеним із них є RGB-кодування, в якому колір кожного пікселя на зображенні кодується за допомогою значень для трьох кольорів – червоного, зеленого та синього. Такі категорії значень, що описують зображення, називаються каналами. Інший вимір тензорів утворюється в ході роботи мережі і містить карти виявлених ознак зображення. Таким чином, згорткові нейронні мережі розглядають кожне зображення як чотиривимірний масив даних. Зазвичай для кожної мережі визначається конкретна необхідна ширина і довжина зображень, хоча також існують мережі, здатні до масштабування.

Для задач розпізнавання в умовах використання великих об’ємів даних необхідно, щоб модель мала високу здатність до навчання та великий відсоток правильних припущень щодо ознак зображення. У порівнянні з традиційними нейронними мережами прямого поширення зі схожою кількістю шарів, згорткові нейронні мережі мають вищу здатність до навчання, оскільки містять набагато менше параметрів та зв’язків. Для традиційних повнозв’язних нейронних мереж для їх правильного навчання в задачах розпізнавання образів необхідна набагато більша кількість даних, оскільки кожне вхідне зображення може мати розмірність у щонайменше кілька сотень тисяч, що вимагатиме відповідну кількість прикладів з різними значеннями для кожного виміру [16].

Ще однією відмінністю згорткових нейронних мереж є те, що вони зазвичай не вимагають попередньої обробки даних, оскільки використовують дані зображень напряму, що дозволяє спрощувати структуру мережі, зважаючи на упорядкованість початкових даних. Водночас, незважаючи на численні переваги, їх застосування все ще вимагає значної витрати ресурсів, особливо для розв’язання задач із зображеннями з високою роздільною здатністю.

Окрім застосування для задач розпізнавання та задач класифікації зображень, згорткові нейронні мережі завдяки модифікаціям широко застосовуються до задач обробки природної мови (NLP), зокрема, вони показали високу ефективність в задачах семантичного парсингу, а також моделюванні та класифікації речень [16].

Структура згорткової нейронної мережі відповідає загальноприйнятій структурі нейронної мережі. Мережа складається з вхідного шару, певної кількості прихованих шарів та вихідного шару. Приховані шари зазвичай складаються зі згорткових шарів, шарів агрегування (субдискретизації), нормалізуючих та повнозв'язних шарів. Ці шари пов'язані між собою шарами з визначеними активаційними функціями. Головним елементом згорткової нейронної мережі є згорткові шари, де до даних з попереднього шару застосовується операція згортки [17]. Загально структура згорткової нейронної мережі показана на рисунку 1.2.

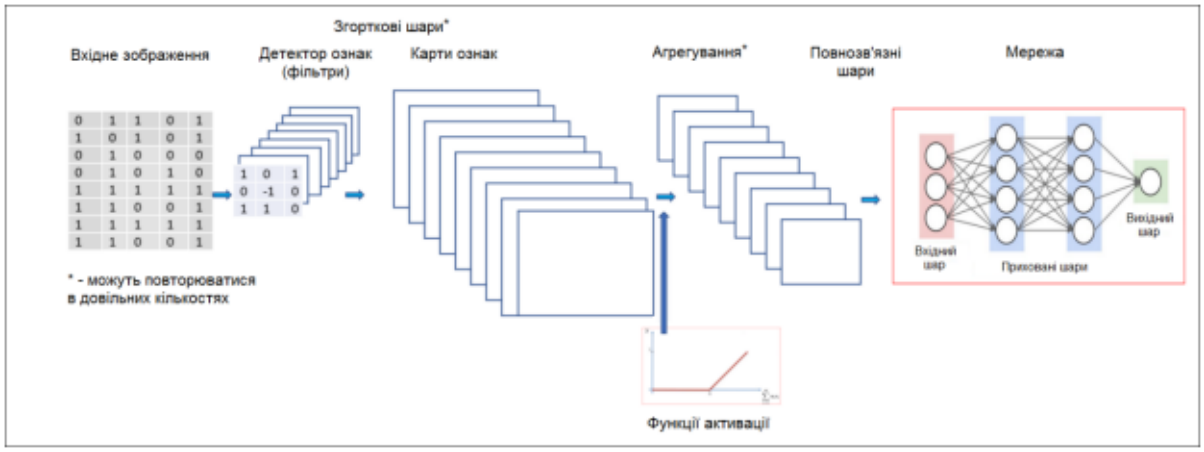


Рисунок 1.2– Загальна структурна схема побудови згорткової нейронної мережі

Згорткові нейронні мережі здобули широку популярність в задачах розпізнавання зображень завдяки тому, що вони уникають або зменшують головні проблеми, пов'язані з обробкою зображень в традиційних повнозв'язних мережах. Велика кількість зв'язків в таких мережах та велика кількість параметрів у кожному зображенні потребує великої кількості вхідних прикладів, а у випадку їх нестачі швидко виникає перенавчання мережі, тобто, модель стає

занадто складною та навчається на нехарактеристичних другорядних ознаках, знайдених у вхідних даних. Ще більшою проблемою є те, такі моделі не є стійкими до збурень або будь-яких змін у зображеннях, наприклад, до розташування об’єкта в іншій частині зображення або зміні кута погляду на нього. Крім того, такі мережі не здатні до врахування топології вхідних даних: зображення є структурованими даними, де пікселі, що розташовані близько один від одного, мають високу кореляцію, що робить необхідним знаходження локальних ознак та взаємозв’язків для ефективного визначення образів.

В згорткових нейронних мережах ці проблеми вирішуються завдяки використанню кількох ідей, найголовнішою з яких є локальна обробка значень на прихованих рівнях. Кожен з рівнів поділений на частини, кожна з яких сприймає лише дані попереднього рівня, розташовані в певній невеликій області. Такі області називаються локальними рецептивними полями. Використовуючи локальні рецептивні поля, мережа може визначати найпростіші елементи та ознаки зображення, як повороти або грані між ділянками зображення. Іншою ідеєю в основі згорткових нейронних мереж є спільні ваги, використання яких зменшує чутливість мережі до змін положення об’єктів на зображенні, незначних поворотів та його спотворення: так, детектор певної простої ознаки зображення, що використовується на певній його ділянці, може визначати аналогічні ознаки і на іншій його ділянці. Завдяки цьому детектори для локальних рецептивних полів на різних ділянках будуть мати однакові ваги, і, як наслідок, в різних частинах зображення для схожих ознак виконуються однакові перетворення.

Крім того, важливим елементом є операція субдискретизації, яка дозволяє зменшувати загальну розмірність даних, при цьому зменшуючи їх чутливість до значних збурень. Зазвичай субдисретизація в архітектурі мережі чергується із операціями згортки, таким чином поступово зменшуючи розмірність даних та збільшуючи кількість карт ознак.

Згорткова нейронна мережа відома своєю потужністю і ефективністю в галузі глибокого навчання. Ці моделі стали невід'ємною частиною багатьох

сучасних застосунків, таких як виявлення об'єктів на зображеннях, розпізнавання мови та аналіз зображень у комп'ютерному баченні. Основна перевага CNN полягає в їх здатності автоматично вивчати і вилучати ключові ознаки з вхідних даних, що робить їх відмінним вибором для завдань, де ручне визначення ознак може бути важким або неможливим [4, 5].

Основне використання: обробка зображень, комп’ютерний зір.

Сильні сторони:

1. Ефективно фіксує просторові ієрархії в даних.
2. Спільне використання параметрів зменшує кількість параметрів, які можна навчити.
3. Добре підходить для сіткових даних, таких як зображення та сітки в текстових даних (наприклад, вбудовування на рівні символів)

Слабкі сторони:

1. Обмежено обробку послідовних даних або даних із тимчасовими залежностями.
2. Не підходить для завдань, які вимагають моделювання довгострокових залежностей.

### 1.4.2 Архітектура RNN (Recurrent Neural Networks)

Recurrent Neural Networks відрізняються від інших моделей тим, що вони мають внутрішню пам'ять. Ця пам'ять дозволяє RNN враховувати часові залежності в даних, що робить їх ідеальними для завдань, де порядок вхідних даних має важливе значення. Завдяки цьому RNN можуть обробляти послідовності даних, такі як текст або часові ряди, з урахуванням контексту попередніх даних [4, 5].

Рекурентна нейронна мережа має спрямовані зв’язки між елементами, причому, вихід елемента нейронної мережі може подаватися на вхід. Це забезпечує послідовну обробку тексту з можливістю запам’ятовувати інформацію і обробляти тексти природної мови. Дуже оптимальна модель для

обробки послідовних даних, часових рядів, людської мови. Розробники бібліотеки Keras запрограмували цю модель і представили її для використання в простому вигляді, та з простою можливістю кастомізації. Влаштовані функції дають змогу обійти налаштування складних конфігураційних налаштувань. Також є можливість створювати власні шари. Це дозволяє з легкістю спробувати перспективні ідеї, протестувати прототипні рішення з мінімальною кількістю коду. Цей алгоритм глибокого навчання зазвичай використовуються для порядкових або тимчасових проблем, таких як мовний переклад, NLP, розпізнавання мовлення та підписання зображень; вони включені до популярних програм, таких як Siri, голосовий пошук та Google Translate. Вони відрізняються своєю «пам'яттю», оскільки вони беруть інформацію з попередніх входів, щоб впливати на поточний вхід і вихід. Хоча традиційні глибокі нейронні мережі припускають, що входи та виходи незалежні один від одного, вихід повторюваних нейронних мереж залежить від попередніх елементів у послідовності. Хоча майбутні події також будуть корисними для визначення результату даної послідовності, однонаправлені рекурентні нейронні мережі не можуть врахувати ці події в своїх прогнозах. Але така модель страждає від градієнтів, що зникають і переповнюються [18]. Архітектура базової моделі RNN наведена на рисунку 1.3.

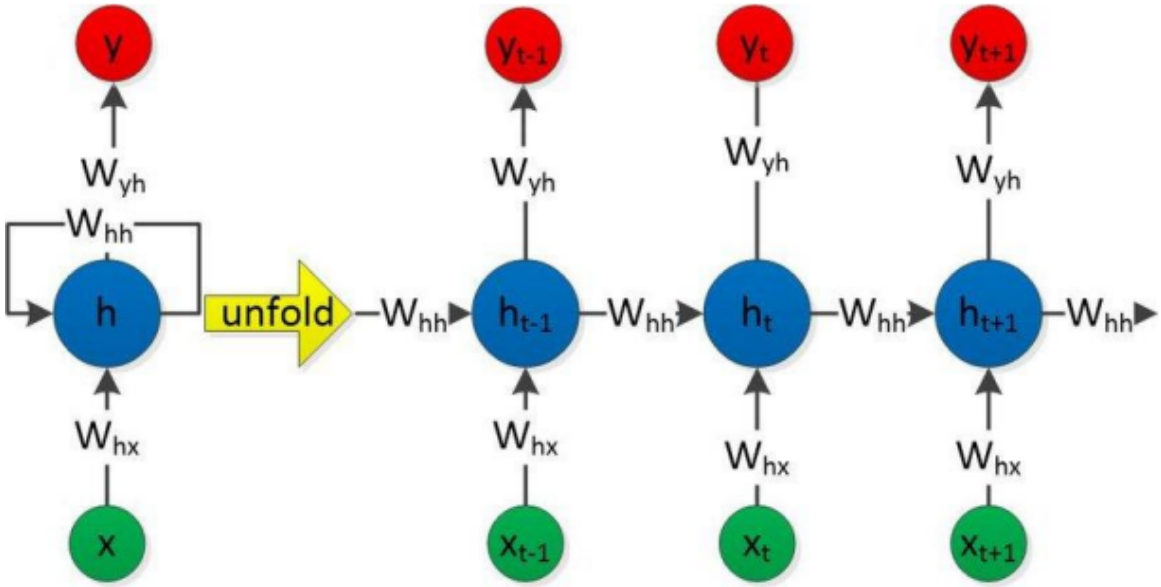


Рисунок 1.3– Архітектура базової моделі RNN

Основне використання: послідовні дані, часові ряди

Сильні сторони:

1. Може моделювати послідовності довільної довжини.
2. Ефективно фіксує тимчасові залежності та контекст.
3. Підходить для завдань, які включають послідовні дані.

Слабкі сторони:

1. Страждає від проблеми зникнення градієнта, що ускладнює фіксацію довгострокових залежностей [6].
2. Навчання може бути повільним через послідовну обробку.

### 1.4.3 Архітектура LSTM (Long Short-Term Memory)

Недоліком RNN є неврахування довгострокової залежності і обмеженість в часі. Одним із рішень цього є модель довготривалої короткочасної пам'яті під назвою (LSTM). LSTM модель – це модель рекурентної нейронної мережі, яка використовує концепцію нейронів із закритим доступом, що означає - кожен нейрон має три різні входи, які контролюють використання та поведінку внутрішньої пам’яті з урахуванням часу. Шлюз забуття та шлюз введення контролюють адаптивність поточної пам’яті, вирішуючи, скільки пам’яті зберігається та скільки вона оновлюється відповідно. Після адаптації пам’яті, вихідний вентиль вирішує, скільки пам’яті буде використано для передбачення наступного слова в послідовності відповідно вхідному параметру. Цей механізм дає змогу моделі запам’ятовувати історію дій, що, у свою чергу, призводить до вищої передбачуваної потужності та робить модель LSTM сучасним рішення у мовному моделюванні та багатьох інших сферах. Однак проблема того ж мовного моделювання залишається в тому, що для навчання моделі доступна лише обмежена кількість даних, тоді як мова є продуктивним і творчим процесом, який в принципі може генерувати нескінченну кількість комбінацій слів і нових слів. Неоднозначність і різноманітність мовних засобів, різні форми

подання тексту призводить до створення гібридних LSTM на рівні аналізу слова і символу [18]. Архітектура базової моделі LSTM наведена на рисунку 1.4.

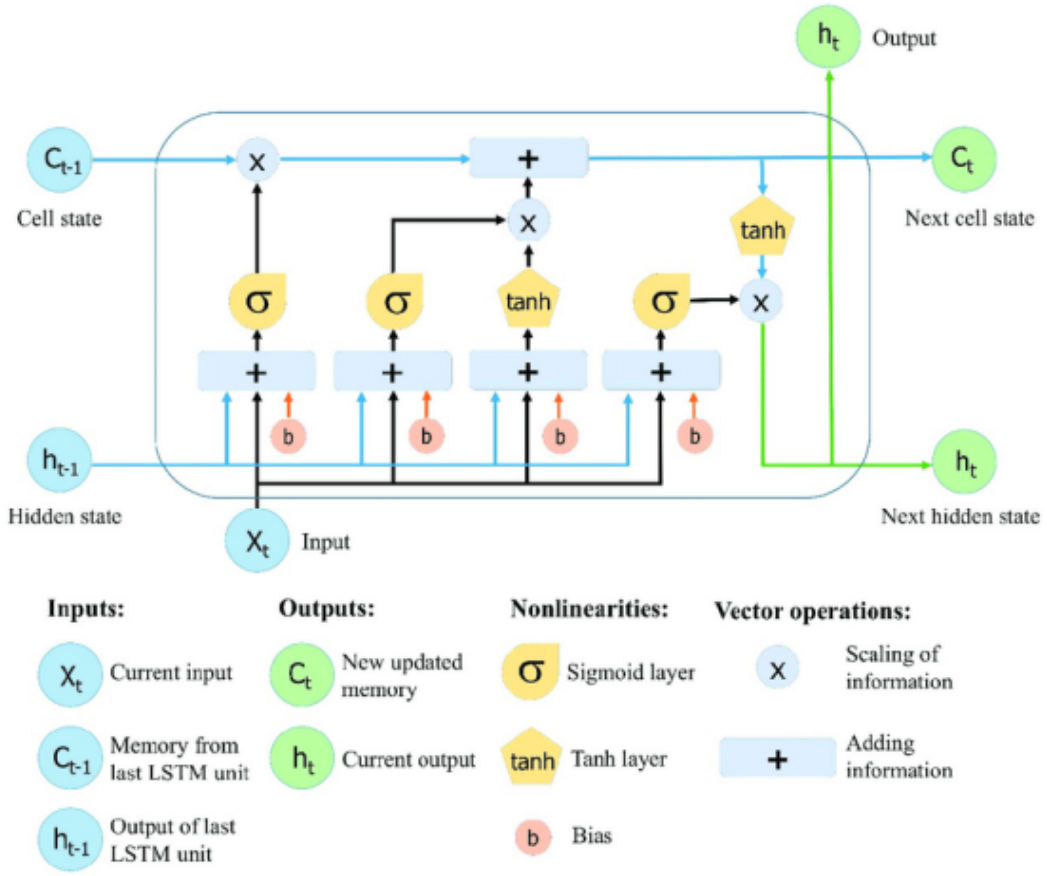


Рисунок 1.4 – Архітектура базової моделі LSTM

Основне використання: послідовні дані з довгостроковими залежностями.

Сильні сторони:

1. Може фіксувати довгострокові залежності в даних.
2. Пом’якшує проблему зникнення градієнта за допомогою механізмів стробування.
3. Ефективно для послідовних даних зі складними шаблонами.

Слабкі сторони:

1. Більш складні, ніж стандартні RNN, вимагають більше обчислень.
2. Все ще складно тренуватися на дуже довгих послідовностях.

## 1.5 Гібридні мережі

Гібридні нейронні мережі – це моделі, які комбінують різні типи нейронних мереж для досягнення більшої точності або ефективності. Наприклад, можна створити гібридну модель, що комбінує згорткові нейронні мережі (CNN) для обробки просторових даних і рекурентні нейронні мережі (RNN) для обробки послідовних даних.

Гібридною системою є система, яка має в собі дві чи більше інтегровані різнорідні підсистеми, які мають загальну ціль або схожі діями (при цьому ці підсистеми можуть бути різної природи) [19].

Одним з прикладів гібридних моделей є CNN-LSTM мережі. Вони використовують CNN для виявлення просторових властивостей в даних, а потім подають ці властивості на вхід до LSTM для моделювання часових залежностей. Це може бути особливо корисно для роботи з даними про акції, оскільки ціни акцій можуть впливати одна на одну, а також можуть мати тренди та циклічність у часі. Приклад поєднання CNN та LSTM у гібридну мережу наведено на рисунку 1.5.

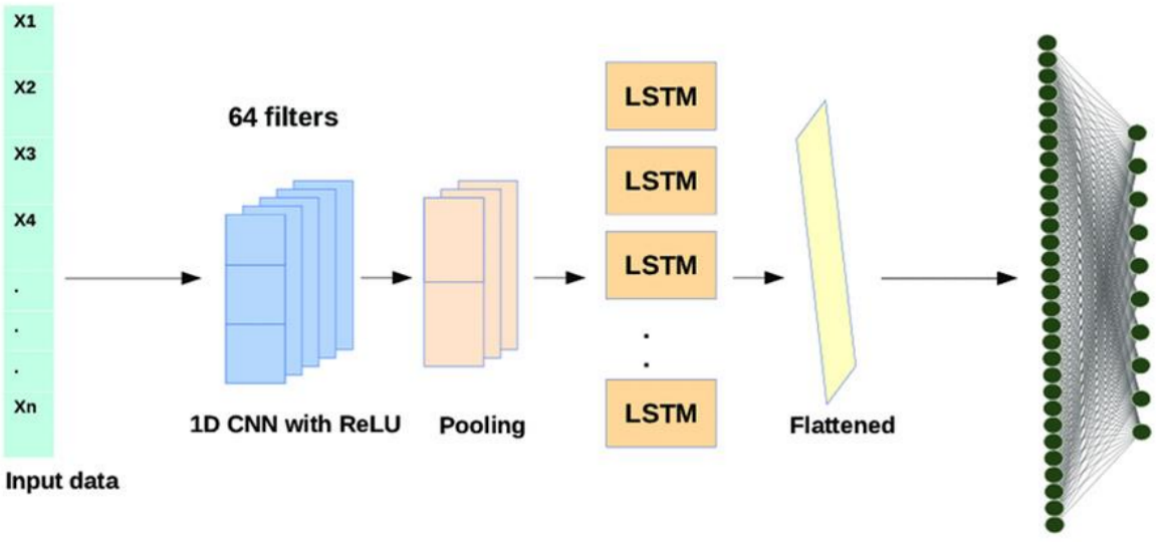


Рисунок 1.5 – Приклад архітектури CNN-LSTM гібридної моделі

Іншим прикладом гібридних моделей є моделі, що комбінують згорткові та Transformer мережі. Transformer мережі можуть бути особливо корисними для моделювання довготривалих залежностей в даних, тоді як CNN може бути

ефективним для виявлення просторових властивостей.

Варто відзначити, що побудова гібридних моделей може бути більш складною, ніж побудова окремих моделей, оскільки потрібно враховувати взаємодію між різними типами мереж та налаштувати їх відповідним чином. Однак, якщо це зроблено правильно, гібридні моделі можуть виявитися дуже потужними засобами прогнозування.

Підсумовуючи, гібридні нейронні мережі представляють собою обіцяючий напрямок в машинному навчанні, комбінуючи переваги різних типів нейронних мереж для створення більш потужних та гнучких моделей. Однак, побудова і налаштування таких моделей вимагає глибокого розуміння як окремих типів мереж, так і способів їх взаємодії [20].

## 1.6 Постановка задачі та загальна схема її розв’язання

Об'єктом дослідження є процес розпізнавання людської діяльності на основі різних видів даних. Предметом дослідження є методи і моделі контекстно-залежного розпізнавання людської діяльності.

Для досягнення мети роботи необхідно виконати такі дослідження:

1) визначити загальну схему вирішення поставленої задачі;

2) провести аналіз існуючих методів контекстно-залежного розпізнавання людської діяльності за допомогою гібридних моделей глибокого навчання;

3) реалізувати методи для системи HAR за допомогою гібридної моделі глибокого навчання;

4) підготувати дані для використання в експерименті;

5) провести тренування та оцінку реалізації на основі заданого експерименту.

Для реалізації і дослідження методів HAR можна виділити такі кроки:

1) збір даних;

2) попередня обробка даних;

3) розробка моделі;

4) модельне навчання;

5) оцінка;

6) Аналіз та порівняння.

# 2 ВИБІР ЗАСОБІВ ВИРІШЕННЯ ЗАДАЧІ ТА РОЗРОБКА АЛГОРИТМІВ

## 2.1 Вибір мови програмування для дослідження методів контекстно-залежного розпізнавання діяльності людини

Розробка програмного забезпечення, що використовує штучний інтелект, може здатися викликом, особливо на початковому етапі. Важливо розуміти, що кожна проблема вимагає унікального рішення та підходу. Для вирішення деяких завдань може знадобитися глибоке розуміння та багаторічні дослідження у цій області, а також використання можливостей, які надає обрана мова програмування. Вибір мови програмування, яка не тільки відповідає технічним вимогам, але й має зручний інструментарій, широку документацію, численні приклади в Інтернеті та легкість у використанні, є ключовим для успішної розробки. Також важливо, щоб мова була адаптована для легкого перенесення проекту на різні платформи, включаючи сервери та портативні пристрої.

### 2.1.1 Мова R

Мова R спеціалізується на аналізі та візуалізації даних, що робить її ідеальною для статистичних обчислень, моделювання та інших видів кількісного аналізу. Вона включає в себе широкий спектр пакетів для різноманітних аналітичних завдань, що дозволяє ефективно використовувати її в сфері штучного інтелекту. Однак, попри її потужність у специфічних доменах, R може бути обмеженою, коли справа доходить до деяких більш спеціалізованих або новітніх бібліотек у галузі штучного інтелекту. Розробка власних рішень в R для таких завдань може бути часомісткою та вимагати значних ресурсів, тому важливо враховувати ці аспекти при виборі мови для конкретного проекту. Незважаючи на це, R залишається однією з найпопулярніших мов у сфері даних завдяки своїй здатності до глибокого статистичного аналізу та ефективної візуалізації, а також завдяки активному співтовариству, яке постійно розширює

її можливості через розробку нових пакетів та розширень.

### 2.1.2 Мова Python

Python є відмінним вибором для проекту завдяки його чистоті синтаксису, логічній структурі та легкості в освоєнні. Ця мова програмування відома своєю універсальністю та високим рівнем портативності, що дозволяє легко переміщати розроблені програми між різними платформами. Величезна кількість доступних ресурсів та документації робить Python ідеальним для імплементації широкого спектру алгоритмів. Його популярність у сфері штучного інтелекту пояснюється наявністю обширної екосистеми, яка включає в себе численні фреймворки, інтегровані розробницькі середовища (IDE) та бібліотеки, такі як TensorFlow, Keras та Scikit-learn. Ці інструменти спрощують процес розробки, надаючи потужні засоби для машинного навчання, візуалізації даних та статистичного аналізу. Вони також забезпечують зручність у відображенні інформації та її аналізі, що робить Python незамінним інструментом для реалізації складних проектів у галузі даних та штучного інтелекту.

### 2.1.3 Мова Java

Мова Java відома своєю універсальністю та потужними можливостями для створення різноманітних програм, включаючи додатки штучного інтелекту. Як мова з багатим набором бібліотек, Java пропонує розробникам інструменти для побудови складних систем, що вимагають високої продуктивності та надійності. Однією з ключових переваг Java є її велика колекція відкритих і добре підтримуваних бібліотек для машинного навчання та обробки даних, які дозволяють розробникам ефективно імплементувати алгоритми штучного інтелекту. Ця мова також славиться своєю крос-платформеністю, що дозволяє запускати написані на ній програми на будь-якій операційній системі без змін у коді, що робить її ідеальною для розробки програмного забезпечення, яке може

бути легко адаптоване під різні середовища. Java постійно оновлюється та вдосконалюється, щоб відповідати сучасним вимогам до програмування, і її стабільність та масштабованість роблять її однією з найпопулярніших мов для великих корпоративних систем.

## 2.2 Вибір середовища розробки

Для створення ефективної гібридної моделі глибокого навчання, важливо, щоб відповідний програмний модуль був адаптований для гнучкого перенесення між різними платформами розробки. Це означає, що модуль повинен бути універсальним, зберігаючи свою функціональність і структуру без потреби в модифікаціях при переході від одного середовища до іншого. Важливими аспектами також є його стабільність і можливість швидкого впровадження в експлуатацію, що вимагає ретельного тестування та оптимізації. Середовище розробки, яке використовується для цих цілей, має надавати інтуїтивно зрозумілі інструменти та ресурси, які сприяють швидкій і якісній розробці. Таке середовище повинно підтримувати інтеграцію з різними бібліотеками та фреймворками, забезпечувати високий рівень сумісності з різними технологіями та мати можливість легкої інтеграції з іншими інструментами розробки.

### 2.2.1 VS Code

Visual Studio Code (VS Code) – це вільний, потужний та легкий редактор коду, розроблений Microsoft. Він підтримує безліч мов програмування та має велику кількість розширень, які можуть бути встановлені для розширення його функціональності. VS Code відомий своєю високою продуктивністю, вбудованим Git контролем, інтеграцією з різними системами збірки та тестування, а також можливістю налаштування робочого середовища під індивідуальні потреби розробника. Це робить його ідеальним вибором для розробки гібридних моделей глибокого навчання, де потрібна гнучкість та

швидкість роботи.

### 2.2.2 IntelliJ IDEA

IntelliJ IDEA – це інтегроване середовище розробки (IDE) від компанії JetBrains, яке спеціалізується на розробці програмного забезпечення для JVM-мов, таких як Java, Kotlin, Scala та інші. IntelliJ IDEA пропонує розробникам розширені можливості для аналізу коду, автоматичного виправлення помилок, рефакторингу та підтримки проектів на різних мовах програмування. Його розумні код-навігаційні функції та інтеграція з сучасними фреймворками роблять його незамінним інструментом для розробки складних додатків.

### 2.2.3 Google Colaboratory

Google Colaboratory, або просто «Colab», – це безкоштовний хмарний сервіс, який дозволяє писати та виконувати Python код через браузер. Colab широко використовується для машинного навчання, аналізу даних та освіти, оскільки він надає доступ до потужних обчислювальних ресурсів, таких як GPU та TPU, без необхідності будь-якої конфігурації з боку користувача. Це робить його ідеальним місцем для експериментування та прототипування моделей глибокого навчання.

### 2.2.4 Anaconda

Anaconda – це популярна дистрибуція Python та R для наукових обчислень, яка спрощує управління пакетами та розгортання. Вона включає в себе бібліотеку conda, яка дозволяє легко встановлювати, запускати та оновлювати наукові пакети та їх залежності. Anaconda також надає доступ до більш ніж 1500 пакетів для обробки даних, машинного навчання, візуалізації та інших завдань. Її інструменти для управління середовищами дозволяють розробникам

створювати ізольовані середовища для різних проектів, що забезпечує чистоту та організацію робочого простору.

## 2.3 Огляд основних фреймворків до використання у роботі

### 2.3.1 PyTourch

PyTorch – це відкритий фреймворк машинного навчання, який набув широкої популярності в наукових дослідженнях завдяки своїй гнучкості, швидкості та зручності в роботі. Він надає потужні інструменти для глибокого навчання та тензорних обчислень з підтримкою автоматичного диференціювання. PyTorch відомий своєю інтуїтивною архітектурою та легкістю в експериментуванні з нейронними мережами, що робить його ідеальним для прототипування та дослідження нових алгоритмів глибокого навчання [22].

### 2.3.2 Numpy

Numpy – це основний пакет для наукових обчислень в Python, який надає підтримку для великих, багатовимірних масивів та матриць, разом з великою колекцією математичних функцій для роботи з цими масивами. Його ефективність та широка функціональність роблять його незамінним інструментом у багатьох областях обробки даних, від фізики та інженерії до фінансів та машинного навчання [23].

### 2.3.3 Scikit-learn

Scikit-learn – це один з найпопулярніших фреймворків для машинного навчання в Python, який включає в себе широкий спектр алгоритмів класифікації, регресії, кластеризації та зниження розмірності. Цей фреймворк відзначається своєю простотою використання, документацією та гнучкістю, дозволяючи легко

інтегрувати статистичне моделювання та машинне навчання в різноманітні додатки та системи [24].

### 2.3.4 Pandas

Pandas – це високорівнева бібліотека Python, яка надає широкі можливості для аналізу та маніпуляції даними. Завдяки зручним структурам даних, таким як DataFrame та Series, Pandas дозволяє виконувати складні операції з очищення, трансформації, агрегування та візуалізації даних. Ця бібліотека є незамінною для обробки та аналізу великих наборів даних, що робить її однією з ключових інструментів у сфері обробки даних та машинного навчання [25].

### 2.3.5 Keras

Keras – це високорівневий API для нейронних мереж, створений для швидкого прототипування та експериментування. Він працює поверх TensorFlow, що дозволяє легко та інтуїтивно створювати глибокі навчальні моделі, забезпечуючи при цьому гнучкість для дослідницьких робіт. Keras підтримує всі основні типи нейронних мереж, включаючи конволюційні, рекурентні та комбіновані [26].

### 2.3.6 TensorFlow

TensorFlow – це комплексний фреймворк відкритого коду для машинного навчання, розроблений Google. Він дозволяє розробникам створювати складні архітектури нейронних мереж з використанням графів обчислень та автоматичного диференціювання. TensorFlow широко використовується для розробки та розгортання моделей машинного навчання в продукції [27].

### 2.3.7 Ray.io

Ray.io – це відкрита система для паралельних та розподілених обчислень, яка спрощує масштабування додатків машинного навчання та штучного інтелекту. Ray.io надає єдиний, уніфікований API для різноманітних завдань, включаючи гіперпараметричний пошук, тренування моделей та виробниче розгортання, та інтегрується з популярними бібліотеками, такими як PyTorch та TensorFlow [28].

## 2.4 Вибір датасету для дослідження контекстно-залежного розпізнавання людської діяльності

Вибір відповідного датасету є критично важливим кроком у процесі розробки та оцінки систем контекстно-залежного розпізнавання діяльності людини (HAR). Якість та релевантність зібраних даних безпосередньо впливають на ефективність навчання та точність моделей, що використовуються для ідентифікації та класифікації різноманітних дій та поведінки людей. У цьому розділі ми розглянемо датасети, які найкраще підходять для розробки та тестування гібридних моделей глибокого навчання, з акцентом на їх здатність відображати контекстні залежності та різноманітність людської поведінки.

Важливість контексту в HAR не може бути недооцінена, оскільки він впливає на інтерпретацію дій людини в різних ситуаціях та умовах. Тому, датасети, які містять багатовимірні вхідні дані з різних джерел та сенсорів, є особливо цінними.

У якості датасету для дослідження було вибрано «UCF101 - Action Recognition Data Set». UCF101 — це набір даних для розпізнавання дій із реалістичними екшн-відео, зібраними з YouTube, які містять 101 категорію дій. Цей набір даних є розширенням набору даних UCF50, який містить 50 категорій дій.

З 13320 відео з 101 категорії дій, UCF101 забезпечує найбільшу різноманітність з точки зору дій і з наявністю великих варіацій у русі камери, зовнішньому вигляді об’єкта та позі, масштабі об’єкта, точці огляду,

захараченому фоні, умовах освітлення тощо, це найбільш складні дані на сьогоднішній день. Оскільки більшість доступних наборів даних розпізнавання дій не є реалістичними та інсценуються акторами, UCF101 має на меті заохотити подальші дослідження розпізнавання дій шляхом вивчення та вивчення нових реалістичних категорій дій.

Відео в 101 категорії дії згруповані в 25 груп, де кожна група може складатися з 4-7 відеороликів дії. Відео з однієї групи можуть мати деякі спільні риси, як-от подібний фон, схожа точка зору тощо. Категорії дій можна розділити на п’ять типів:

1) взаємодія людина-об’єкт;
2) лише рух тіла;
3) взаємодія людина-людина;
4) гра на музичних інструментах;
5) спорт.

Повну інформації та список усіх категорій можна знайти за посиланням [29].

## 2.5 Розробка алгоритму гібридної моделі глибокого навчання

Як вже було розглянуто раніше, згорткові нейронні мережі (CNN) чудово обробляють зображення, тоді як мережі довго-короткочасної пам’яті (LSTM) вправно обробляють послідовні дані. Інтегруючи ці два типи мереж, ми можемо ефективніше вирішувати складні проблеми комп’ютерного зору, наприклад класифікацію відео із датасету, що був обраний.

Процес розпізнавання картинок є дуже популярною та дослідженою темою, тому існує багато застосунків для класифікації того, що користувач може бачити на картинці. З іншого боку, процес розпізнавання відео може бути погано класифікований за допомогою тих самих алгоритмів, бо кожне відео може мати свій власний контекст, в залежності від якого, зміст відео може дуже сильно мінятися. Розглянемо декілька прикладів:

1) процес присідання може бути розглянуто як процес присідання на стілець, але якщо після присідання людина знову встає – це може бути йога або інші заняття спортом, в залежності від іншого контексту;

2) якщо людина на картинці падає, то це процес падіння, але ж якщо іншій картинці ця ж людина стоїть на землі – може бути таке, що в процесі людина перегорнулась та це заняття легкою атлетикою, або паркуром.

Далі, буде розглянуто декілька способів класифікації дії на відео [30].

### 2.5.1 Використання CNN та LSTM архітектур для класифікації відео

#### 2.5.1.1 Згорткові нейронні мережі для класифікації зображення.

Згорткова нейронна мережа (CNN), також відома як ConvNet, — це спеціалізована архітектура глибокого навчання, призначена для ефективної обробки та інтерпретації даних зображень. Він використовує ядра або фільтри для проходження зображення та створення карт функцій, які вказують на наявність або відсутність певних функцій у різних місцях зображення.

Спочатку CNN створює невелику кількість карт функцій, але в міру того, як мережа заглиблюється вглиб, кількість цих карт збільшується, а їхні розміри зменшуються. Це масштабування досягається за допомогою операцій об’єднання, які згущують дані зображення та зберігають важливу інформацію. Приклад того, як CNN класифікує зображення наведено на рисунку 2.1.

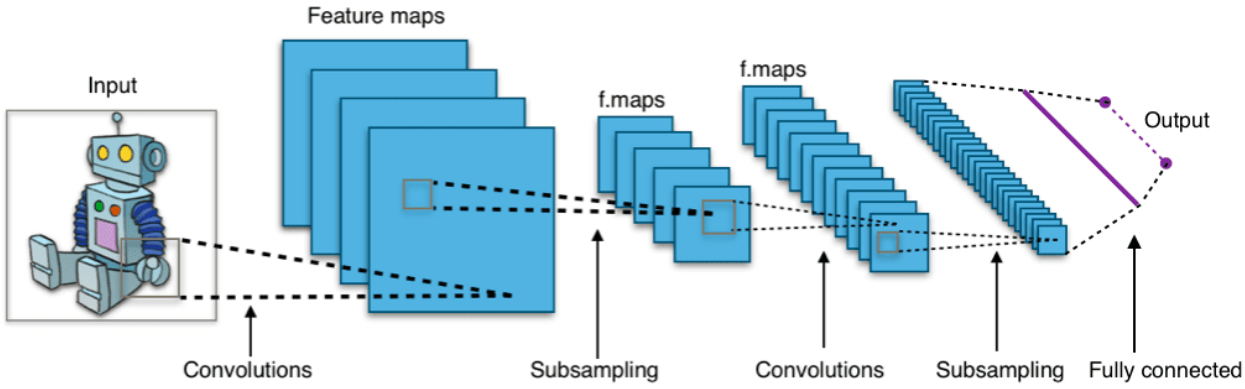


Рисунок 2.1 – Приклад CNN для класифікації зображень

Рівні CNN здатні вивчати дедалі складніші функції. Наприклад, ранні шари можуть ідентифікувати прості грані та кути, тоді як більш глибокі шари здатні розпізнавати складні об’єкти, такі як людські фігури в різних позах.

У сфері згорткових нейронних мереж (CNN) є декілька передових

архітектур, кожна з яких адаптована для певних типів завдань і даних. Серед них C3D (3D Convolutional Neural Networks), R(2+1)D, I3D (Inflated 3D ConvNets) і ResNet-50 (Residual Networks) виділяються своїми унікальними підходами та застосуваннями.

C3D, або 3D Convolutional Neural Networks, особливо вправні в обробці відеоданих. Розширюючи згорткові шари в три виміри, C3D може фіксувати часову динаміку, що робить його придатним для таких завдань, як розпізнавання дій у відео. Цей часовий аспект дозволяє C3D розуміти прогресування рухів у часі, що є вирішальним для точної інтерпретації відеовмісту.

R(2+1)D або (2+1)D Convolutional Network — це ще одна архітектура, яка чудово підходить для аналізу відео. Він розкладає тривимірні згортки на двовимірні просторові згортки, за якими слідує одновимірна тимчасова згортка. Ця декомпозиція дозволяє мережі більш ефективно вивчати просторові та часові характеристики, що призводить до підвищення продуктивності в таких завданнях, як розпізнавання дій і виявлення подій у відео.

I3D, або Inflated 3D ConvNets, — це варіант, який роздуває фільтри та об’єднує ядра попередньо навчених 2D CNN у 3D, дозволяючи їм вивчати просторово-часові характеристики з відеоданих. Цей підхід використовує потужність архітектур 2D CNN, таких як Inception [31], для класифікації відео та завдань розпізнавання дій, забезпечуючи міст між розпізнаванням 2D зображень і аналізом 3D відео.

ResNet-50, частина сімейства Residual Network, особливо відома своєю глибокою архітектурою з 50 рівнів. ResNet-50 використовує пропуск з’єднань або ярлики для переходу через деякі шари, що допомагає вирішити проблему зникнення градієнта в дуже глибоких мережах. Ця архітектура дуже універсальна і широко використовується в задачах класифікації зображень. Так, при виявленні об’єктів ResNet-50 можна використовувати для ідентифікації та визначення місцезнаходження об’єктів на зображенні. Приклад використання для знаходження кримінальних елементів серед відвідувачів супермаркетів на зображеннях або відео можна знайти за посиланням [32].

Таким чином, кожна з цих архітектур CNN – C3D, R(2+1)D, I3D і ResNet-50 має унікальні сильні сторони. Від аналізу часової динаміки у відео до розпізнавання складних візерунків у зображеннях, ці мережі розширюють можливості традиційних CNN, прокладаючи шлях для передових застосувань комп’ютерного зору, таких як автономні транспортні засоби, системи спостереження та інтерактивні медіа.

2.5.1.2 Довготривала короткочасна пам'ять для класифікації відео**.** Мережа LSTM спеціально розроблена для роботи з послідовністю даних, оскільки вона враховує всі попередні вхідні дані під час генерації виходу. LSTM насправді є типом нейронної мережі, яка називається рекурентною нейронною мережею, але RNN невідомі як ефективні для роботи з довгостроковими залежностями у вхідній послідовності через проблему, яка називається проблемою вибухаючого або зникаючого градієнта, коли при великої кількості вхідних даних мережа може «забувати» те, що було раніше. Пояснення такого механізму, котрий називається «зворотне поширення по часу» можна побачити на рисунку 2.2, а приклад на рисунку 2.3.

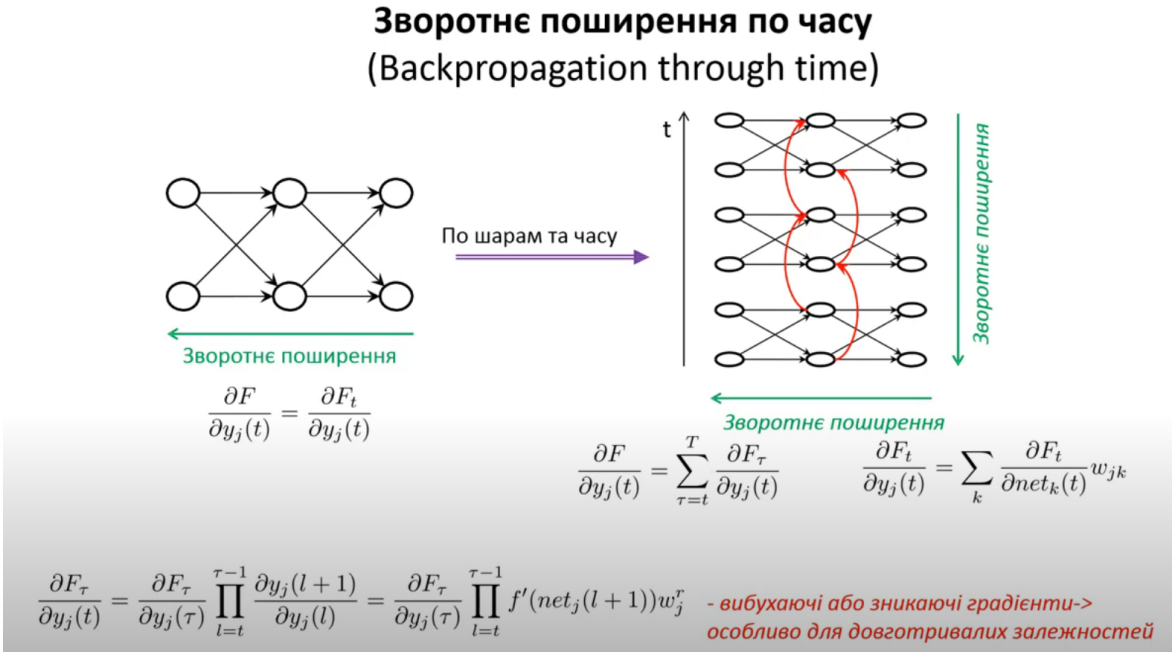


Рисунок 2.2 – зворотне поширення по часу

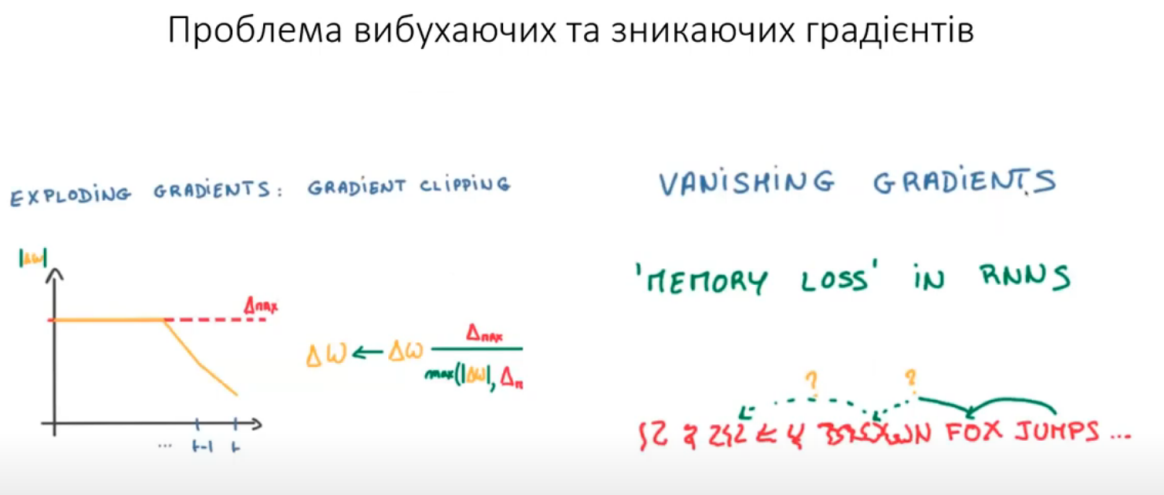


Рисунок 2.3 – Приклади вибухаючих та зникаючих градієнтів

LSTM були розроблені, щоб подолати зникаючий градієнт, і тому комірка LSTM може запам’ятовувати контекст для довгих вхідних послідовностей. Приклад вузлів LSTM наведено на рисунку 2.4.

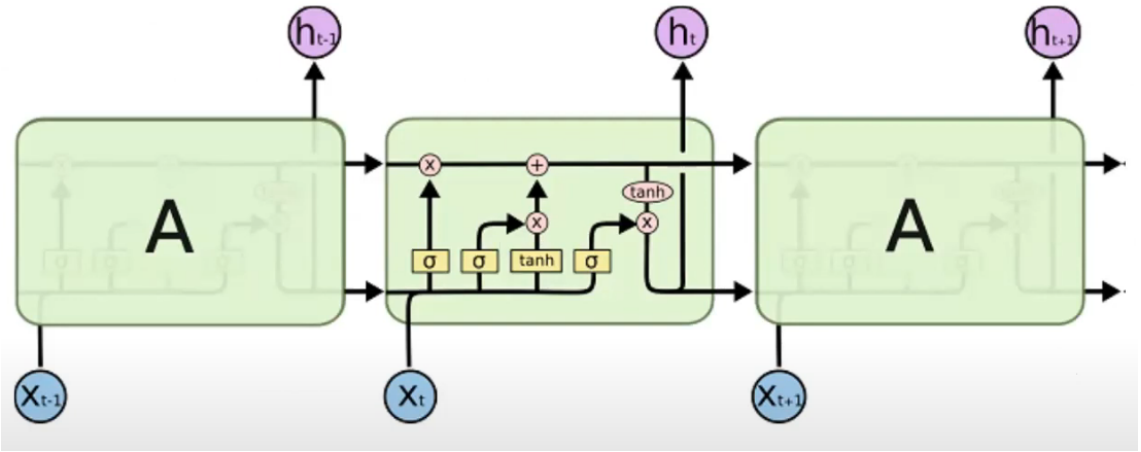


Рисунок 2.4 – Приклад вузлів LSTM мережі

Розглянемо окремий вузол LSTM мережі та проаналізуємо його [33]. Першим кроком у цієї мережі робиться так званий вентиль забування (рисунок 2.5). Він існує для того, щоб мережа «забути» щось із внутрішнього стану пам’яті мережі. Таке рішення про забування приймає на себе сигмоїд, котрий видає на виході число 0 або 1, де 0 – «забути», а 1 – повністю залишити.

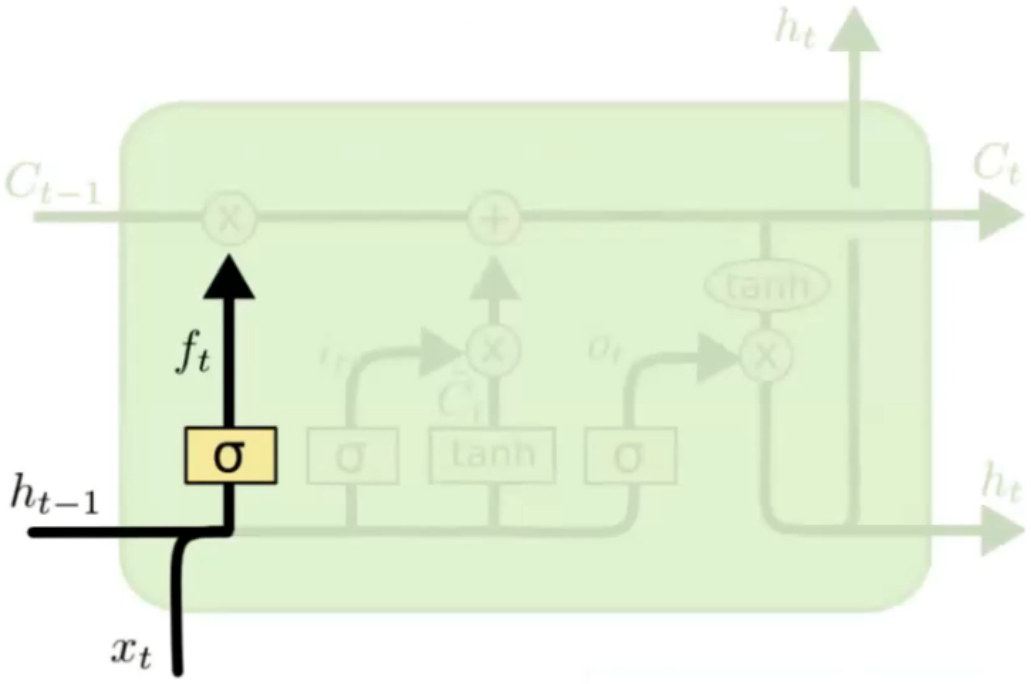


Рисунок 2.5 – Вентиль «забування» LSTM мережі

$$f_t = \sigma\big(W_f \cdot [t_{t-1}, x_t] + b_f\big), \tag{2.1}$$

Далі йде вхідний вентиль (рисунок 2.6), що обчислює, наскільки йому цікава нова інформація, щоб її запам’ятовувати.

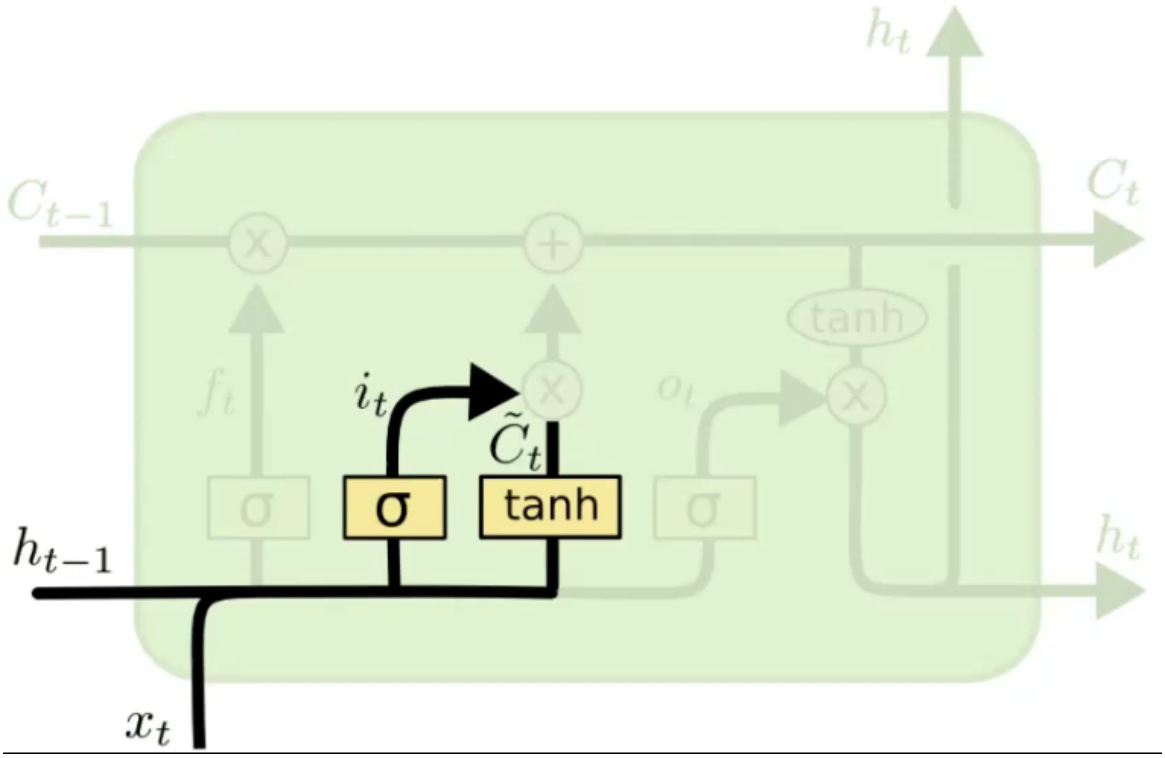


Рисунок 2.6 – Вхідний вентиль LSTM мережі

Цей вентиль складається із двох частин:

1) перша – сигмоїдний шар, що вирішує, які значення будуть модифіковані;
2) друга – шар тангенсу, що створює вектор значень нового кандидата, який може бути доданий до внутрішнього стану.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \quad (2.2)$$

$$\widetilde{C_t} = \tanh(W_C[h_{t-1}, x_t] + b_C), \quad (2.3)$$

На наступному кроці ці дві частини комбінуються для модифікації стану (рисунок 2.7).

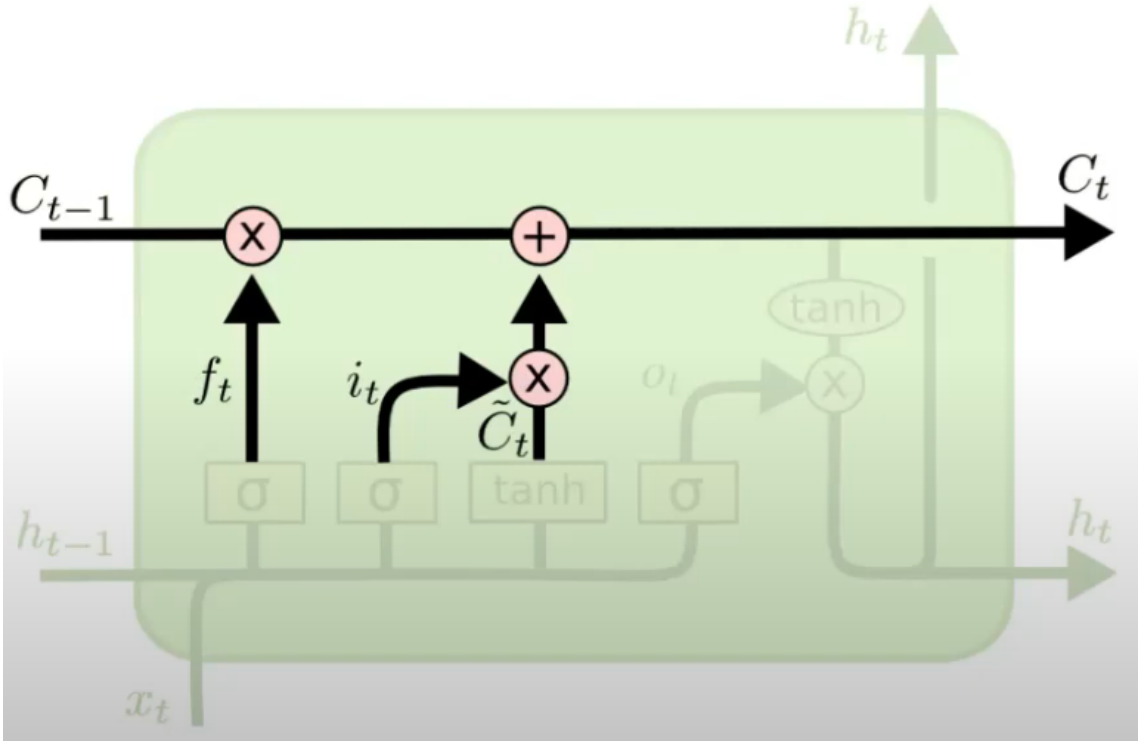


Рисунок 2.7 – Модифікація стану LSTM мережі

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t, \quad (2.4)$$

де $C_t$ – лінійна комбінація пам'яті $C_{t-1}$ і спостереження $C_t$ з тільки обчисленими вагами для кожної з компонент.

Останнім кроком в LSTM мережі виступає формування поточного виходу мережі (рисунок 2.8). Так як частина вхідного сигналу вже в пам'яті, не потрібно обчислювати активацію по всьому сигналу. Спочатку сигнал проходить через сигмоїду, яка вирішує, яка його частина важлива для подальших рішень. Далі,

гіперболічний тангенс «розмазує» вектор пам’яті на відрізок від -1 до 1. Наприкінці, ці два вектори перемножуються, а отримані $h_t$ та $C_t$ передаються далі по ланцюжку.

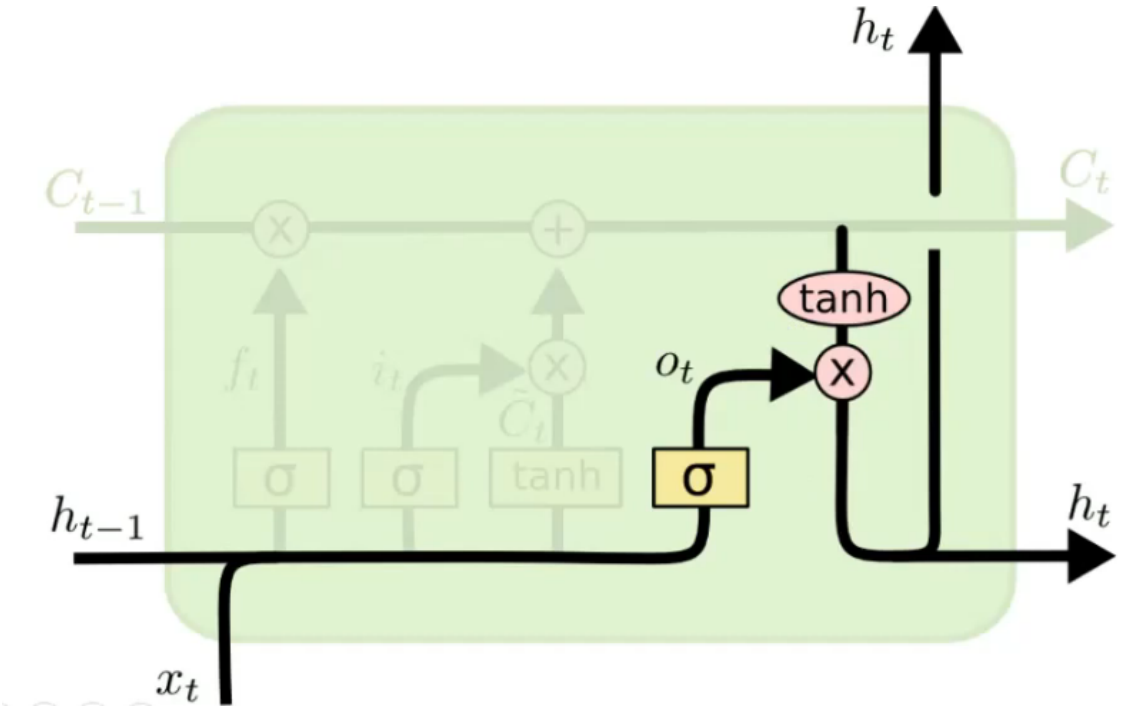


Рисунок 2.8 – Обчислення виходу LSTM мережі

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \tag{2.5}$$

$$h_t = o_t * \tanh(C_t), \tag{2.6}$$

LSTM мережі є оптимальним вибором для аналізу відеоданих, оскільки вони здатні вловлювати часові залежності та послідовності, які є критичними для розуміння контексту в динамічних сценах. В контексті роботи з відео, LSTM можуть використовуватися для визначення та класифікації людських дій, враховуючи не тільки окремі кадри, але й їх послідовність у часі, що дозволяє моделі "запам'ятовувати" та використовувати інформацію з попередніх кадрів для більш точного висновку.

Використання LSTM у поєднанні з згортковими нейронними мережами (CNN) для аналізу відео дозволяє створювати гібридні моделі, які ефективно обробляють як просторові (зображення), так і часові (послідовність зображень) характеристики. Такі моделі можуть використовувати просторові характеристики, видобуті CNN з кожного кадру, а потім інтегрувати цю

інформацію через час за допомогою LSTM, що дозволяє розпізнавати складні дії та взаємодії, які розгортаються протягом декількох кадрів.

Ця здатність до довгострокового запам'ятовування робить LSTM ідеальними для завдань, де необхідно враховувати тривалі контекстуальні зв'язки, таких як розпізнавання послідовності рухів у спорті, взаємодії між людьми або навіть для прогнозування наступних дій на основі поточної активності. Використання LSTM у дослідженні відеоданих відкриває широкі можливості для покращення точності та ефективності систем відеонагляду, робототехніки та інших застосувань, де важливо розуміння динамічних візуальних інформаційних потоків.

### 2.5.2 Основні підходи для розпізнавання дій на відео

2.5.2.1 Single-Frame Classification. Найфундаментальніший метод класифікації дій у відео передбачає застосування класифікатора зображень до кожного відеокадру, розглядаючи їх як окремі сутності для класифікації. Наприклад, якщо застосувати цю техніку до відео, на якому особа виконує сальто назад, результати будуть такими, як показано на рисунку 2.9.

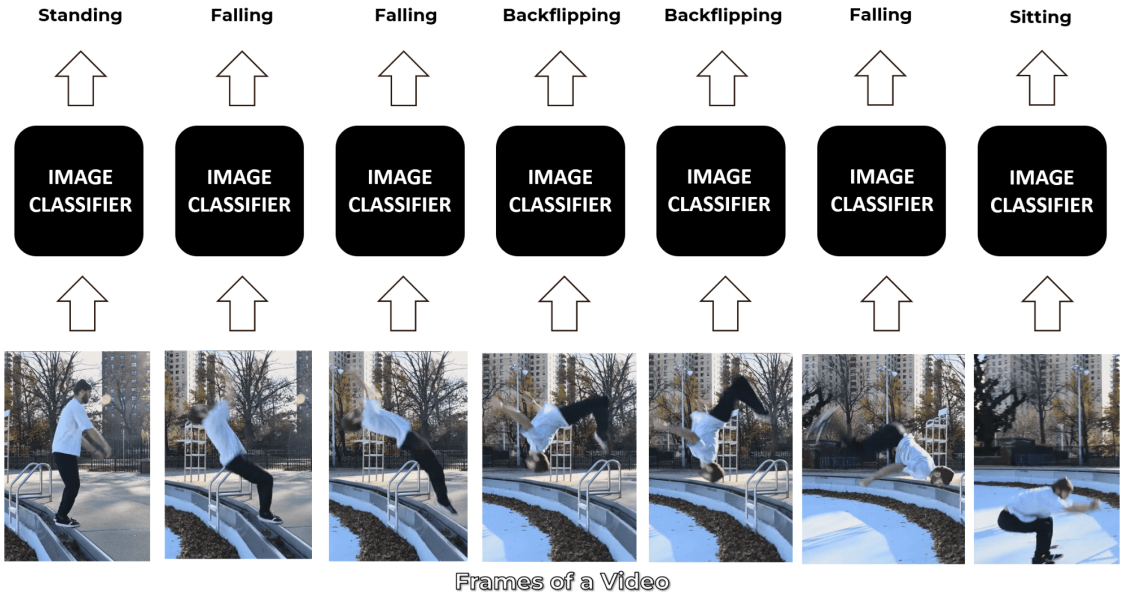


Рисунок 2.9 – Класифікація кожного кадру з відео

Через відсутність урахування послідовного контексту кадрів класифікатор може помилково визначити дії «Падання» в певних кадрах, де суб’єкт фактично виконує «Сальто назад». Ця неправильна класифікація виникає тому, що метод не враховує часову послідовність дій; навіть спостерігач, який розглядає кадри окремо, може неправильно витлумачити дію як «Падіння».

Щоб отримати переконливу класифікацію для всього відео, можна підрахувати найбільш часто передбачені дії в кадрах. Хоча цей спрощений підхід може бути достатнім у простих випадках, у розглянутому прикладі він призводить до неточного висновку про «Падіння». Більш складна стратегія передбачає усереднення прогнозованих ймовірностей за кадрами для отримання більш надійного загального прогнозу, як показано на рисунку 2.10.

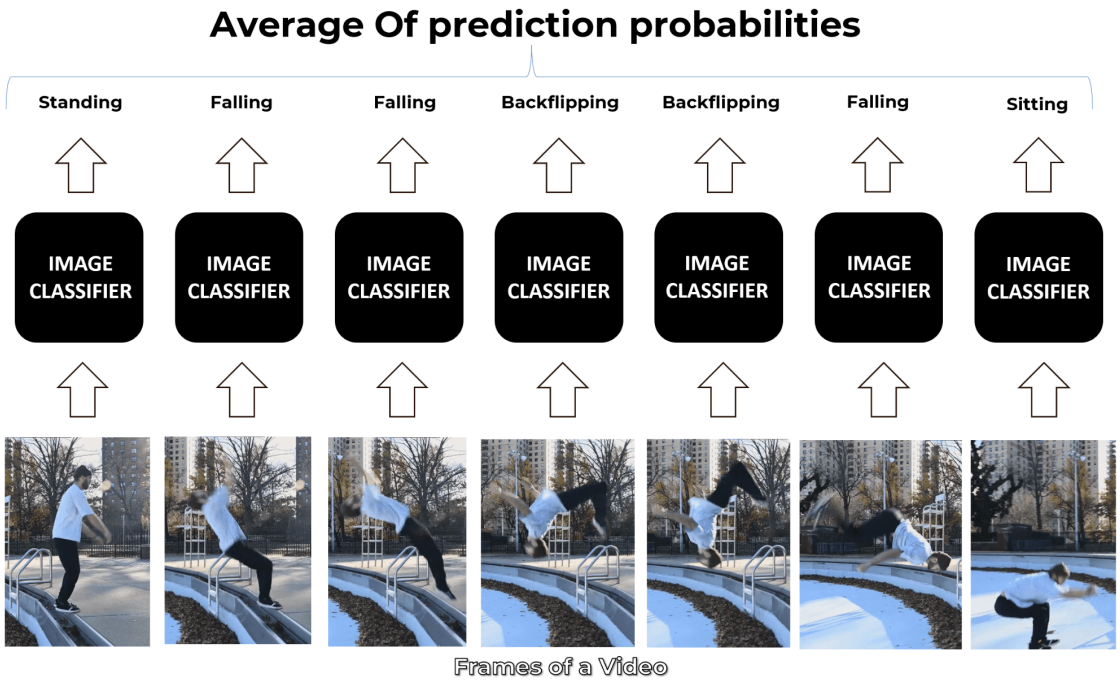


Рисунок 2.10 – Усереднений результат покадрової класифікації

2.5.2.2 Late Fusion. Альтернативний метод відомий як пізнє злиття. У цій стратегії індивідуальні прогнози кадрів спочатку робляться незалежно. Згодом ці результати подаються на злитий рівень, який об’єднує зібрані дані для отримання остаточного прогнозу, використовуючи переваги часової динаміки, присутньої в послідовності (рисунок 2.11).

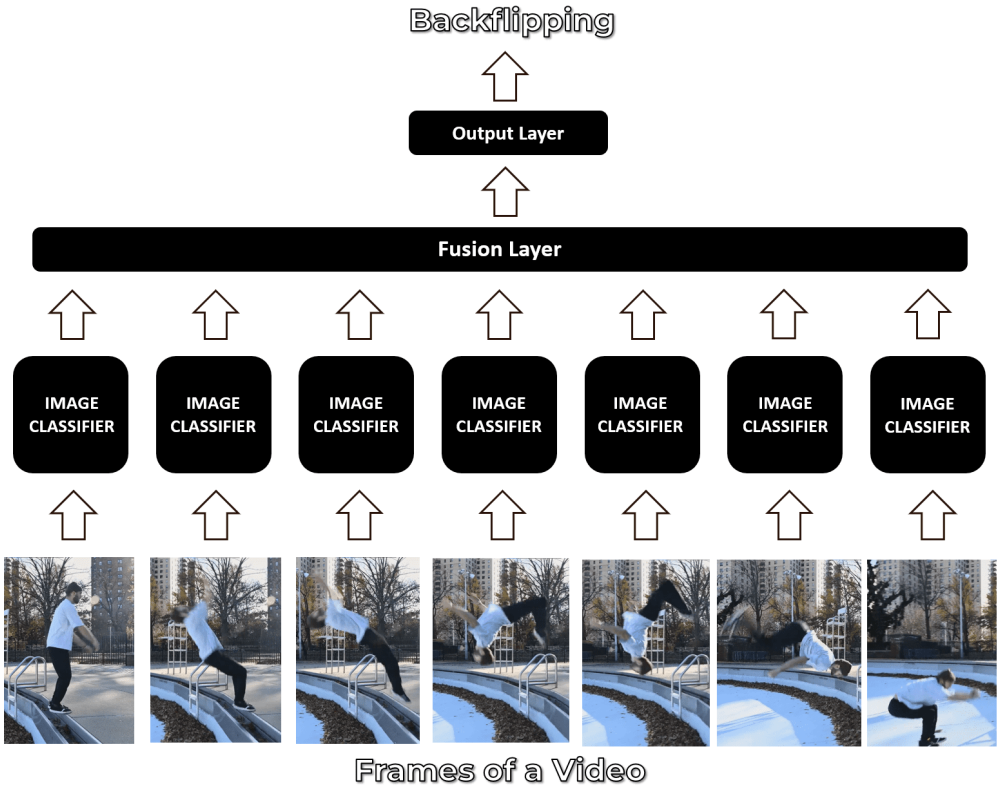


Рисунок 2.11 – Приклад роботи пізнього злиття

Хоча ця методика дає задовільні результати, вона має свої обмеження щодо ефективності.

2.5.2.3 Early Fusion. Альтернативним методом класифікації відео є техніка раннього злиття (рисунок 2.12), коли всі дані об'єднуються на початкових етапах мережі. Це контрастує зі стратегією пізнього злиття, яка об'єднує дані на кінцевій фазі обробки. Хоча ранній синтез є потужною технікою, він також має власний набір обмежень. Зокрема, раннє злиття може призвести до простору даних великої розмірності, обробка якого може потребувати інтенсивних обчислень. Це може призвести до довшого часу навчання та вимагати більше пам'яті, що може бути неможливим для всіх обчислювальних середовищ. Крім того, при попередньому об'єднанні всієї інформації може виникнути ризик втрати тимчасових зв'язків у даних, які є ключовими для розуміння динамічних моделей у відеоконтенті.

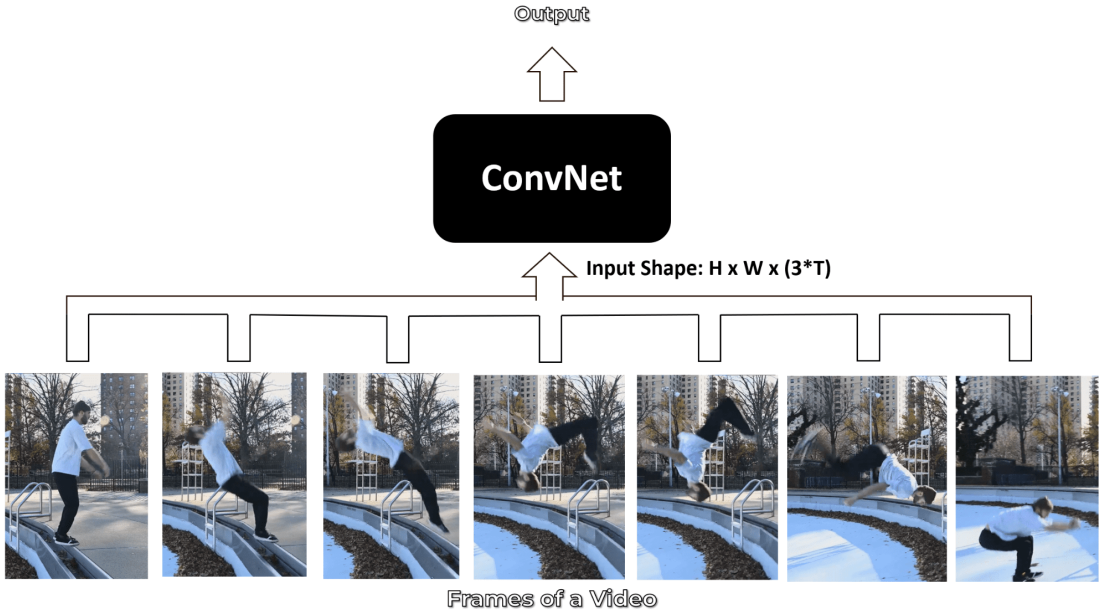


Рисунок 2.12 – Приклад роботи Early Fusion

Slow Fusion (3D CNN)**.** Альтернативний підхід передбачає застосування тривимірної згорткової мережі, яка поступово інтегрує часові та просторові дані на всіх рівнях мережі, процес, який влучно називають «повільним злиттям» (рисунок 2.13). Перевагою цього методу є повне змішування інформації, що має вирішальне значення для завдань, що потребують детального часово-просторового аналізу. Однак одним суттєвим недоліком техніки Slow Fusion є її висока обчислювальна вимога, що призводить до меншої швидкості обробки.

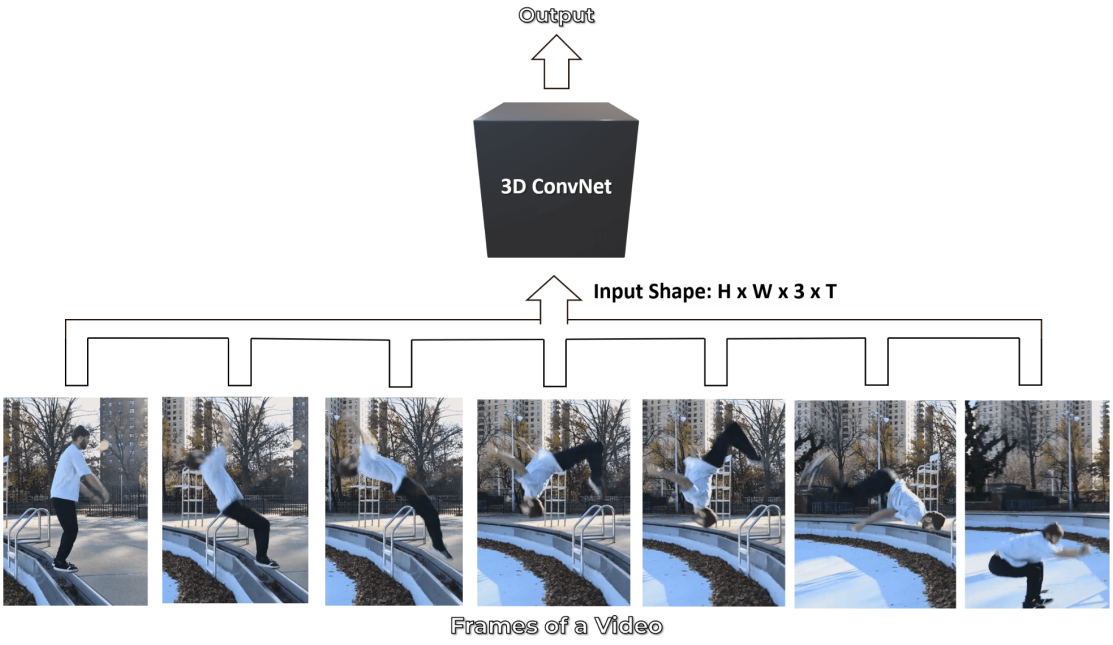


Рисунок 2.13 – Приклад 3D CNN

Це може бути обмежувальним фактором, особливо коли ви маєте справу з великими наборами даних або потребуєте аналізу в реальному часі, де ефективність є настільки ж важливою, як і точність.

2.5.2.4 Pose Detection та LSTM. Альтернативний підхід передбачає розгортання мережі визначення пози для отримання орієнтирних координат людей з кожного кадру відео. Згодом ці орієнтири вводяться в мережу LSTM для визначення активності людини. Приклад показаний на рисунку 2.14.

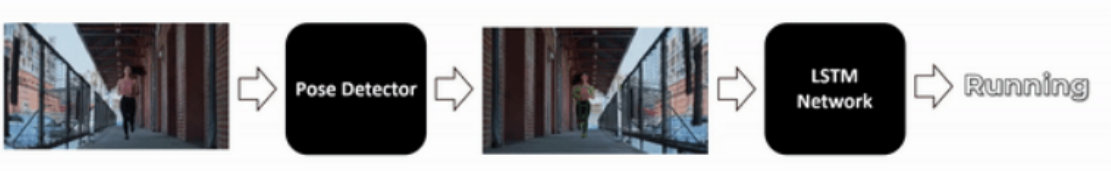


Рисунок 2.14 – Приклад роботи Pose Detection із LSTM

Ринок пропонує безліч досвідчених систем визначення пози, придатних для цього методу. Однак одним із недоліків цього методу є виключення всіх даних, крім орієнтирів. Контекстні деталі, наприклад ознаки навколишнього середовища, можуть значно підвищити точність прогнозування активності. Наприклад, у контексті розпізнавання категорії футбольних дій наявність стадіону та відмінність уніформи команд можуть надати цінну інформацію, яка допоможе моделі зробити більш точні прогнози.

Цей метод, зосереджений на динаміці людського тіла через орієнтири, хоч і надійний, може не помічати багатства навколишньої сцени, яка часто містить важливу інформацію для розуміння складної діяльності. Таким чином, хоча мережі визначення пози спрощують процес, зосереджуючись на людських фігурах, інтеграція додаткового контексту сцени може ще більше покращити продуктивність моделі.

2.5.2.5 CNN + LSTM**.** Наступний підхід використовуватиме мережу згорткової нейронної мережі (CNN) та мережу довгострокової короткочасної пам’яті (LSTM) для виконання розпізнавання дій із використанням просторово-

часового аспекту відео. Згорткова нейронна мережа (CNN) може використовуватися для вилучення просторових характеристик з окремих кадрів у відео послідовності. Згодом мережа довгострокової пам’яті (LSTM) використовується для визначення часової динаміки цих кадрів, забезпечуючи комплексний аналіз як просторових, так і часових характеристик. Приклад цього підходу наведено на рисунку 2.15.

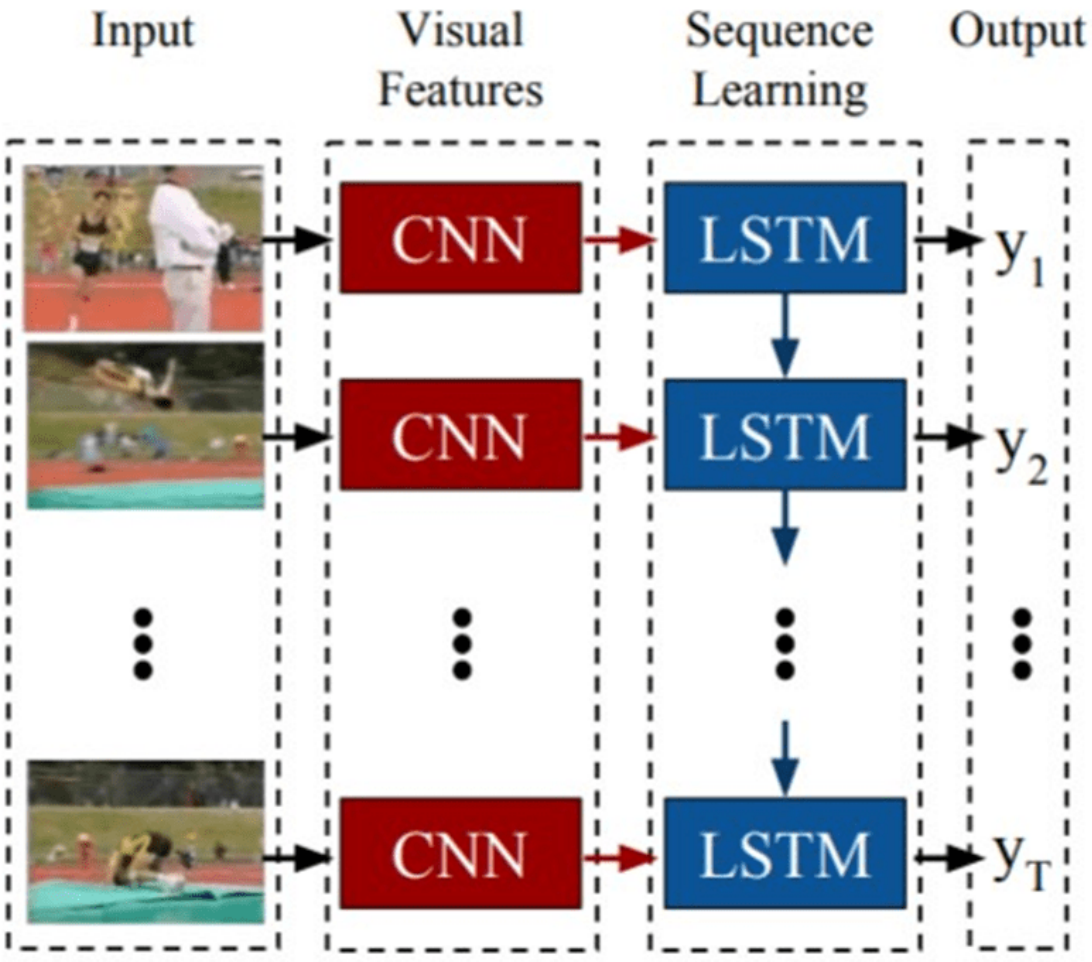


Рисунок 2.15 – Приклад використання CNN+LSTM підходу

У цьому дослідженні робота буде зосереджена на двох основних архітектурах, які інтегрують CNN з мережами LSTM.

Перша така архітектура – ConvLSTM, яка поєднує згорткові шари з шарами LSTM в уніфіковану структуру, що дозволяє моделі обробляти вхідні дані з просторовим і часовим контекстом одночасно.

ConvLSTM — це тип рекурентної нейронної мережі для просторово-

часового прогнозування, яка має згорткові структури як у переходах із входу в стан, так і зі стану в стан. ConvLSTM визначає майбутній стан певної комірки в сітці за вхідними даними та минулими станами її локальних сусідів. Цього можна легко досягти, використовуючи оператор згортки в переходах із стану в стан і введення в стан. Ілюстративний приклад такого підходу наведено на рисунку 2.16.

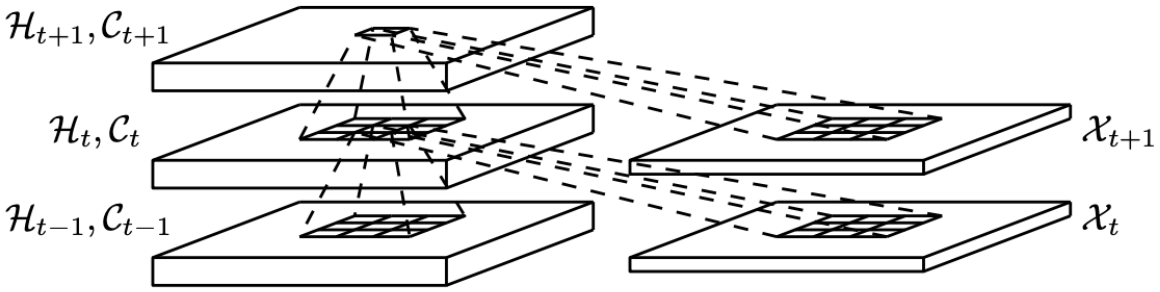


Рисунок 2.16 – Внутрішня структура ConvLSTM моделі

Ключові рівняння ConvLSTM показані нижче

$$i_t = \sigma(W_{xi} * X_t + W_{hi} * H_{t-1} + W_{ci} \odot C_{t-1} + b_i), \quad (2.7)$$

$$f_t = \sigma(W_{xf} * X_t + W_{hf} * H_{t-1} + W_{cf} \odot C_{t-1} + b_f), \quad (2.8)$$

$$C_t = f_t \odot C_{t-1} i_t + \tanh \odot (W_{xc} * X_t + W_{hc} * H_{t-1} + b_c), \quad (2.9)$$

$$o_t = \sigma(W_{xo} * X_t + W_{ho} * H_{t-1} + W_{co} \odot C_t + b_o), \quad (2.10)$$

$$H_t = o_t \tanh \odot (C_t), \quad (2.11)$$

де $\sigma$ – сигмоїдна активаційна функція, яка перетворює вхідний сигнал у діапазоні від 0 до 1, що дозволяє моделювати ймовірність та бінарні рішення;

$W_{xi}, W_{xf}, W_{xc}, W_{xo}$ – вагові матриці для вхідних даних $X_t$ для вхідного, забувального, вхідного модуляційного та вихідного воріт відповідно;

$W_{hi}, W_{hf}, W_{hc}, W_{ho}$ – вагові матриці попереднього стану прихованої комірки $H_{t-1}$ для вхідного, забувального, вхідного модуляційного та вихідного воріт відповідно;

$W_{ci}, W_{cf}, W_{co}$ – вагові матриці для попереднього стану комірки $C_{t-1}$ для вхідного, забувального та вихідного воріт відповідно;

$b_i, b_f, b_c, b_o$ – зміщення для вхідного, забувального, вхідного

модуляційного та вихідного воріт;

$X_t$ – вхідний вектор на часовому кроці $t$;

$H_{t-1}$ – вектор попереднього стану прихованої комірки на часовому кроці $t-1$;

$C_{t-1}$ – вектор попереднього стану комірки на часовому кроці $t-1$;

$C_t$ – вектор поточного стану комірки на часовому кроці $t$;

$H_t$ – вектор поточного стану прихованої комірки на часовому кроці $t$;

$*$ – оператор згортки [34];

$\odot$ – оператор елемент-візного множення (Hadamard product), який виконує множення відповідних елементів двох векторів або матриць [35];

$tanh$ – гіперболічна тангенс активаційна функція, яка перетворює вхідний сигнал у діапазоні від -1 до 1, дозволяючи моделювати позитивні та негативні зміни в даних.

Якщо ми розглядаємо стани як приховані представлення рухомих об’єктів, ConvLSTM з більшим перехідним ядром повинен мати можливість фіксувати швидші рухи, а той, з меншим ядром, може фіксувати повільніші рухи.

Щоб гарантувати, що стани мають таку саму кількість рядків і стовпців, що й вхідні дані, перед застосуванням операції згортання необхідне доповнення. Тут заповнення прихованих станів на граничних точках можна розглядати як використання стану зовнішнього світу для розрахунку. Зазвичай, перш ніж надходить перший вхід, треба ініціалізувати всі стани LSTM до нуля, що відповідає «повному незнанню» майбутнього. [36].

Наступна архітектурний підхід, що розглядається в цієї роботі – повторювана згорткова мережа (LRCN), який застосовує повторювані шари поверх згорткових функцій, дозволяючи послідовну обробку відеоданих з часом.

Цей підхід поєднує глибокий ієрархічний візуальний екстрактор функцій (наприклад, CNN) із моделлю, яка може навчитись розпізнавати та синтезувати часову динаміку для завдання, що включають послідовні дані (входи або виходи), візуальні, лінгвістичні чи інші. На рисунку 2.15 зображено ядро підходу. LRCN працює, передаючи кожен візуальний вхід $x_t$ (ізольоване

зображення або кадр із відео) через перетворення ознаки $\phi_V(.)$ з параметрами $V$, зазвичай CNN, щоб створити векторне представлення $\phi_V(x_t)$ фіксованої довжини. Потім результати $\phi_V$ передаються в навчальний модуль рекурентної послідовності.

У своїй найзагальнішій формі рекурентна модель має параметри $W$ і відображає вхід $x_t$ і попередній прихований стан $h_{t-1}$ у вихідний $z$ та оновленому прихованому стані $h_t$. Тому висновок потрібно запускати послідовно (тобто зверху вниз, у вікні вивчення послідовності на рисунку 2.9), обчислюючи в такому порядку: $h_1 = f_W(x_1, h_0) = f_W(x_1, 0)$, потім $h_2 = f_W(x_2, x_1)$ і так далі до $h_T$. Деякі з моделей накладають кілька LSTM одна на іншу.

Щоб передбачити розподіл $P(y_t)$ за результатами $y_t \in C$ (де $C$ — дискретний скінченний набір результатів) на етапі часу $t$, виходи $z_t \in R^{d_z}$ послідовної моделі проходять через рівень лінійного передбачення $\widehat{y_t} = W_z z_t + b_z$, де $W_z \in R^{|C| \times d_z}$ і $b_z \in R^{|C|}$ є вивченими параметрами. Нарешті, прогнозований розподіл $P(y_t)$ обчислюється шляхом взяття softmax від $\widehat{y_t}$:

$$P(y_t = c) = softmax(\widehat{y_t}) = \frac{\exp(\widehat{y_t}, c)}{\sum_{c' \in C} \exp(\widehat{y_t}, c')}, \quad (2.12)$$

Успіх останніх глибоких моделей для розпізнавання об’єктів свідчить про те, що стратегічне складання багатьох «шарів» нелінійних функцій може призвести до потужних моделей для проблем сприйняття. Для великого $T$ наведена вище повторюваність вказує на те, що кілька останніх прогнозів від поточної мережі з $T$ часовими кроками обчислюються за допомогою дуже «глибокої» ($T$-рівня) нелінійної функції, що свідчить про те, що отримана рекурентна модель може мати таку саму репрезентативну силу, як і глибока $T$-рівнева мережа. Важливо, однак, те, що ваги цієї моделі послідовності $W$ повторно використовуються на кожному часовому кроці, змушуючи модель вивчати загальну покрокову динаміку часу (на відміну від динаміки, обумовленої $t$, індексом цієї послідовності) і запобігаючи зростанню розміру параметра пропорційно максимальній довжині послідовності [37].

## 2.6 Висновки до другого розділу

У цій кваліфікаційній роботі було обрано середовище розробки VsCode, оскільки воно забезпечує гнучкі можливості для кодування та інтеграції з різними бібліотеками. Для розробки моделі буде використано Python у поєднанні з бібліотекою TensorFlow, що є стандартом у сфері глибокого навчання та обробки відео. Для навчання моделі обрано датасет UCF101, який є особливо цінним через велику кількість відео та різноманітність сценаріїв, які містять складні випадки для алгоритмів розпізнавання. Наприклад, різні сцени з музичними інструментами та натовпами можуть становити виклик для точного класифікування відео в залежності від контексту, що спонукає до глибшого дослідження алгоритмів розпізнавання.

У дослідженні будуть реалізовані та порівняні дві архітектури для розпізнавання людської діяльності: ConvLSTM та LCRN. Обидві архітектури будуть досліджені з точки зору їхньої здатності адаптуватися до контексту на відео, який часто може бути динамічним та непередбачуваним.

Крім того, в роботі буде зосереджено увагу на оптимізації цих моделей. Планується використовувати різні методи аугментації даних, такі як перетворення відео в чорно-білий формат або застосування різних фільтрів, щоб покращити здатність моделей узагальнювати навчання та підвищити їхню точність у складних умовах. Це також допоможе у розробці більш стійких моделей до різноманітності умов у реальному світі та їхньому застосуванні у широкому спектрі сценаріїв.

# 3 ПРОГРАМНА РОЗРОБКА ГІБРИДНОЇ МОДЕЛІ ГЛИБОКОГО НАВЧАННЯ ДЛЯ КОНТЕКСТНО-ЗАЛЕЖНОГО АНАЛІЗУ ДІЯЛЬНОСТІ ЛЮДИНИ

У наступних підрозділах буде розглянуто процес побудови програмного забезпечення (ПЗ) для побудови різноманітних гібридних моделей глибокого навчання. Увесь код буде наведено у посиланні [38]. Після побудови моделей буде проведена їх оцінка та порівняння між різними їх типами. Також, збудовані моделі буде протестовано не реальних прикладах застосування. У якості матеріалу для перевірки буде взято різноманітні відео з мережі YouTube, після чого буде зроблено остаточні висновки за побудованими моделями та припущення на наступні поліпшення запропонованих моделей.

## 3.1 Збір на попередня обробка даних

У якості даних буде використовуватися UCF101 датасет, що включає в себе різноманітні відео, поділенні на категорії. Перш ніж використовувати ці відео в навчанні моделей, потрібно підготувати цей датасет.

Попередня обробка даних відіграє вирішальну роль у функціональності системи. Цей етап є ключовим у перетворенні необроблених даних у формат, сумісний і оптимальний для споживання моделями, що розробляються. Під час цього дослідження буде використано різноманітні методи попередньої обробки даних, щоб налаштувати набір даних відповідно до вимог моделей. Ці методи допоможуть не лише забезпечити сумісність із моделями, але й при можливості підвищити їх продуктивність. Застосування цих стратегій попередньої обробки є невід’ємною частиною підготовки даних, сприяючи ефективному аналізу та досягненню більш точних результатів (таблиця 3.1).

Таблиця 3.1 – Використані техніки пре-обробки відео даних

| Назва техніки | Опис | Ефект від застосування / Мета |
|---|---|---|
| Зміна розміру відео | Розмір кожного відео рівномірно змінюється до 64x64 пікселів. | Основною метою цього підходу є досягнення однакових вхідних розмірів для всіх відео в наборі даних. |
| Нормалізація | Кожен колір пікселя представляється у виді від 0 до 255. Таким чином, кожен піксель можна поділити на 255 та отримати RGB нормалізований вигляд. | Нормалізація допомагає зменшити варіативність даних та покращує збіжність під час тренування, оскільки модель легше адаптується до даних, що мають однаковий масштаб. |
| Аугментація даних | Щоб розширити набір даних і зменшити ризик перенавчання, застосовуються кілька методів розширення. Ці методи включають довільне кадрування, гортання та обертання відео. | Цей підхід не тільки збільшує розмір набору даних, але також значно покращує здатність моделі узагальнювати різні сценарії. |
| Вибір фреймів | З кожного відео вибирається фіксована кількість кадрів, незалежно від його довжини. | Однакова обробка відео різної довжини та фіксоване використання пам’яті. |

Кожен метод попередньої обробки, що був впроваджений, має вирішальне значення для підвищення ефективності впроваджених моделей. Зміна розміру відео стандартизує вхідні розміри всіх відео, що є необхідністю для роботи CNN шарів. Завдяки нормалізації буде досягнуто однорідності відео, забезпечено постійну якість, незважаючи на різні умови освітлення та кути камери. Методи збільшення даних, включаючи випадкове кадрування, перевертання та обертання, не лише розширюють наш набір даних, але й допомагають запобігти переобладнанню. Вибір фреймів вирішує проблему відео різної довжини та оптимізує використання пам’яті.

На самому початку роботи з обраним датасетом, буде проведена ретельна перевірка даних на їх адекватність та відповідність очікуванням. Для цього з кожного класу датасету буде відібрано зразки відео, які потім будуть аналізовані

з відповідними мітками для детальної візуальної перевірки. Цей етап є критично важливим, оскільки візуальний огляд даних, що використовуються, вважається однією з практик у сфері обробки та аналізу даних. Це дозволить не лише переконатися в якості та релевантності даних, але й забезпечити їхню надійність для подальших етапів обробки та використання в моделях машинного навчання. Далі на рисунках 3.1-3.5 ілюстративні приклади такого візуального аналізу.

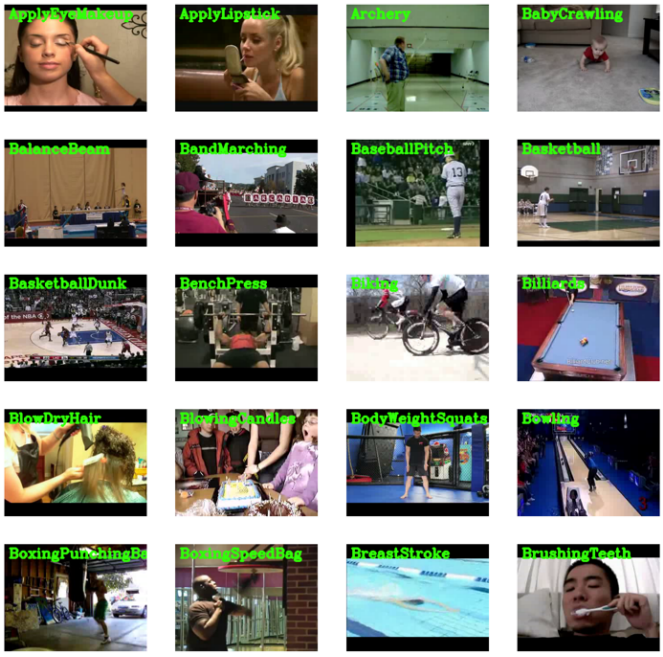


Рисунок 3.1 – Перший семпл датасету


Рисунок 3.2 – Другий семпл датасету

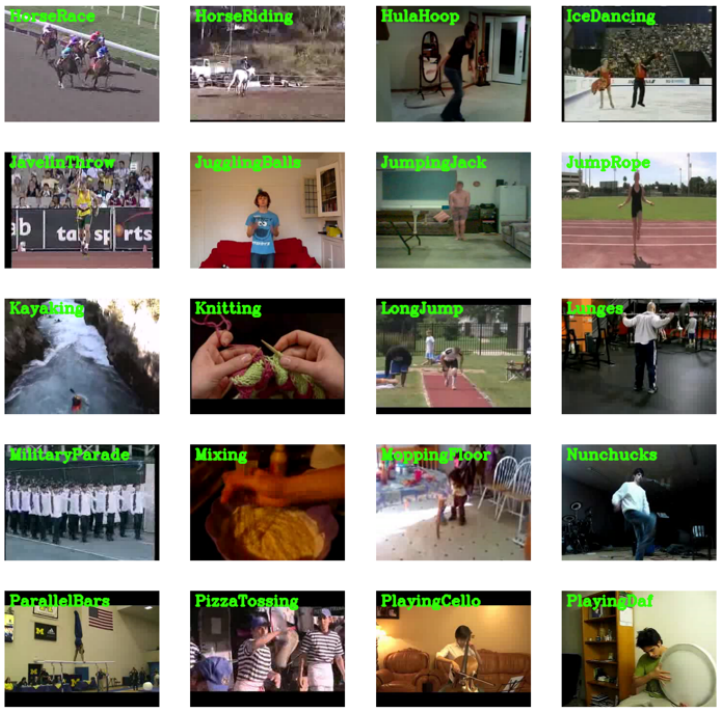


Рисунок 3.3 – Третій семпл датасету

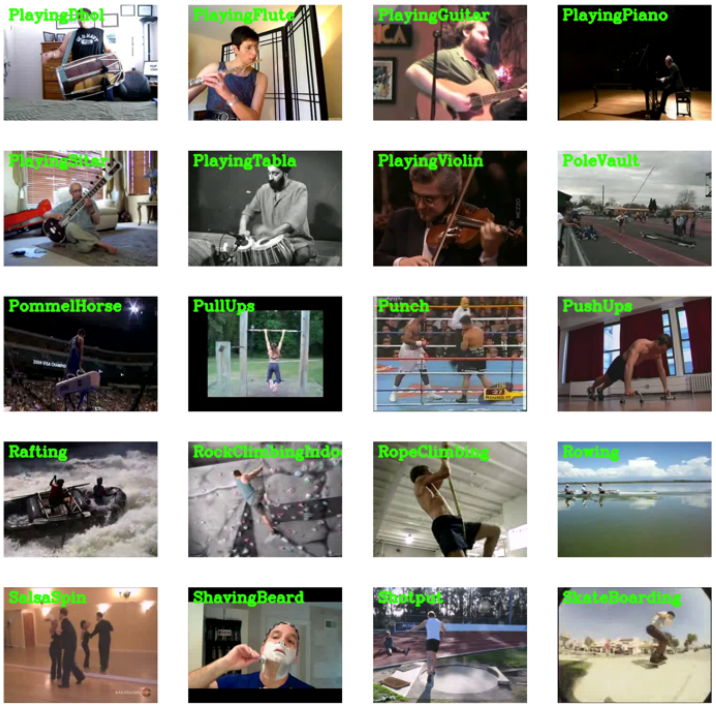


Рисунок 3.4 – Четвертий семпл датасету


Рисунок 3.5 – П’ятий семпл датасету

Наведений датасет характеризується значним об'ємом інформації, що ставить перед роботою виклик обробки цих масивних даних без ризику перевантаження оперативної пам'яті (RAM). Відповідно, у роботі буде використана техніка, відома як "пакетна обробка даних" (Data batch processing). Цей метод передбачає розбиття загального набору даних на менші пакети, які потім обробляються послідовно. Такий підхід не тільки знижує навантаження на пам'ять, але й забезпечує більш ефективне управління ресурсами системи. Крім того, пакетна обробка даних сприяє підвищенню швидкості обробки та зменшує можливість виникнення помилок, пов'язаних із перевантаженням пам'яті, що є ключовим фактором для успішної реалізації наших дослідницьких завдань.

Для ефективної реалізації пакетної обробки даних, буде створено спеціалізований клас процесора даних, код якого представлений на додатку А.1. Цей клас буде відповідати за оптимізоване розділення відеоданих на керовані пакети, що дозволить системі обробляти великі обсяги інформації без зайвого навантаження на оперативну пам'ять.

Важливим аспектом обробки кожного відео є нормалізація, яка включає стандартизацію розмірів фреймів, коригування яскравості та контрасту, що

сприяє однорідності даних та покращує якість їх аналізу. Відповідні процеси та методики будуть реалізовані функціях для розбиття відео на відповідну кількість фреймів та обробки й нормалізації кожного відео. Код цих функцій представлений на додатках А.2 та А.3.

Впровадження цих методів при роботі з великими даними сприяє більш ефективному використанню ресурсів, забезпечуючи стабільну та високопродуктивну роботу системи у процесі розпізнавання та аналізу відеоданих.

У процесі розробки та впровадження описаних вище функцій та класів, відкриваються широкі можливості для їх застосування у навчанні моделей машинного навчання. Ці спеціально розроблені функції охоплюють повний спектр технік, що були детально представлені у таблиці 3.1 Ці функції, будучи оптимізованими та адаптованими під специфічні вимоги дослідження та датасету, стають невід'ємною частиною вирішення комплексних задач, пов'язаних із машинним навчанням та впровадження моделей. Їх використання не лише полегшує роботу з даними, а й робить результати дослідження більш порівняльними між собою. Також, для того, щоб результати пре-обробки були однакові для всіх моделей, у роботі буде використане константне встановлення сіда (Seed number) для усіх методів випадковості. Код такої операції представлений на додатку А.4.

## 3.2 Побудова та налаштування гібридних моделей глибоких нейронних мереж на основі архітектур ConvLSTM, LRCN, CNN-LSTM

Для дослідження у цій роботі буде побудовано декілька моделей гібридної глибокої нейронної мережі.

Перша розроблена модель базується на принципах ConvLSTM, поєднуючи в собі елементи як конволюційних, так і LSTM (Long Short-Term Memory) мереж [30]. Модель починається з першого шару, який використовує невелику кількість фільтрів із середніми розмірами ядра для вилучення ознак із вхідних даних. Цей

шар також містить механізм для запобігання перенавчання. Після першого шару слідує шар максимального згортання, який допомагає зменшити розмірність даних, зберігаючи при цьому важливу інформацію. Цей процес повторюється кілька разів зі збільшенням кількості фільтрів у кожному наступному ConvLSTM шарі, дозволяючи моделі вилучати більш складні та високорівневі ознаки з даних. Кожен шар ConvLSTM супроводжується додатковими шарами для запобігання перенавчання та шарами максимального згортання. Це забезпечує баланс між здатністю моделі вилучати складні ознаки та запобіганням її перенавчанню на конкретних особливостях навчальних даних. В кінцевій частині моделі, після проходження через декілька шарів ConvLSTM та максимального згортання, дані спрощуються до одновимірної форми. Після цього застосовується шар, який класифікує дані до заданої кількості класів, використовуючи функцію активації, яка відповідає за визначення вірогідності належності вхідних даних до кожного класу. Код наведеної моделі представлений у додатку А.5.

Друга модель є розробкою довготривалої рекурентної конволюційної мережі (LRCN) [30]. Вона використовує поєднання конволюційних та рекурентних шарів, щоб ефективно обробляти та аналізувати відеодані. На початковому етапі модель включає кілька конволюційних шарів, кожен з яких використовує невелику кількість фільтрів з метою вилучення основних ознак із відео. Ці шари застосовуються послідовно, збільшуючи складність та глибину вилучення ознак. Кожен з конволюційних шарів супроводжується шарами пулінгу для зменшення розмірності даних, а також шарами, які запобігають перенавчанню, знижуючи ризик втрати загальної здатності моделі до адаптації. Після проходження через ряд конволюційних шарів, дані піддаються обробці рекурентним шаром LSTM. Цей шар має за мету врахувати часові відносини між послідовними кадрами, що дозволяє моделі краще розуміти та аналізувати динаміку людської активності в часі. На завершальному етапі, модель включає шар класифікації, який використовує функцію активації для визначення ймовірності кожного класу, на основі ознак, отриманих з попередніх шарів. Код наведеної моделі представлений у додатку А.6.

Наступна модель реалізує ту ж саму архітектуру, що й попередня, але на відміну від неї вносить деякі ключові зміни, що оптимізують її для специфічних задач розпізнавання людської активності. Першою помітною відмінністю є кількість рівнів у конволюційних нейронних мережах (CNN), де використовуються лише три рівні замість чотирьох. Це може вплинути на здатність моделі до деталізації аналізу, але також робить її більш компактною та ефективною з точки зору обчислень. Другою зміною є розмір пулінгу. У цій моделі для деяких шарів використовується пулінг (2, 2), на відміну від (4, 4), що застосовувалось у попередній моделі. Ця зміна може призвести до більш детального збереження інформації в процесі обробки даних. Наостанок, зазначимо, що кількість фільтрів у LSTM шарі була змінена до 64 замість 16, що може забезпечити більш глибокий аналіз часових послідовностей в даних. Це збільшення кількості фільтрів може дозволити моделі краще виявляти складні залежності та патерни в динаміці людської активності, що відбивається у відео. Код наведеної моделі представлений у додатку А.7.

У подальшому дослідженні наступна модель буде реалізовувати архітектуру, що поєднує конволюційну нейронну мережу (CNN) з довгостроковою короткотерміновою пам'яттю (LSTM), інтегруючи при цьому механізм уваги. Детальніше про цю архітектуру та її дослідження можна дізнатися з статті, опублікованої за посиланням [39].

Механізм уваги в архітектурі нейронних мереж є перспективним напрямком, який дозволяє моделі більш ефективно визначати важливі частини вхідних даних. Цей підхід імітує людську увагу, дозволяючи моделі фокусуватися на ключових аспектах вхідних даних, що забезпечує краще розуміння та обробку інформації. У рамках реалізації механізму уваги, буде інтегрувати підхід, що передбачає складання окремих кадрів відео в єдиний метафрейм. Цей метод дозволяє моделі не тільки розглядати кожен кадр окремо, але й оцінювати відео як єдине ціле, об'єднуючи інформацію з різних моментів часу в один консолідований знімок. Таке поєднання дозволяє моделі більш точно визначати, які моменти в відео є ключовими, та зосереджувати увагу на найбільш значимих частинах даних.

У якості шару CNN буде використано C3D (Convolutional 3D) архітектуру, яка є ефективною для аналізу відеоданих. C3D архітектура розроблена для вилучення просторово-часових ознак, що робить її гарним вибором для аналізу відео, оскільки вона здатна розпізнавати інформацію як у просторових (зображення), так і у часових (послідовність зображень) вимірах. Використання такої архітектури дає змогу глибше аналізувати та інтерпретувати відеодані, забезпечуючи більш точні та вичерпні результати.

Реалізація моделі починається з вхідного шару, після чого відбувається застосування серії 3D конволюційних шарів. Ці шари використовують активаційну функцію для вилучення просторових ознак із відео. Шари максимального згортання, які йдуть після кожного конволюційного шару, допомагають зменшити розмірність даних і зберегти ключову інформацію. Додаткове застосування нормалізації допомагає стабілізувати навчання. Після обробки за допомогою 3D CNN, дані потім передаються до шару LSTM. Цей шар ефективно обробляє часову послідовність даних, вилучаючи важливі часові ознаки. Ключовим елементом моделі є інтеграція механізму уваги, який виконує роль фільтра, виділяючи найбільш значущі елементи у часовій послідовності. Це досягається шляхом призначення ваг кожному елементу в послідовності, що дозволяє моделі зосереджувати увагу на ключових аспектах даних. Після цього, дані підсумовуються для створення єдиного представлення, яке відображає найважливіші ознаки відео. Останнім кроком є застосування шарів для класифікації вихідних даних в задану кількість класів. Код наведеної моделі представлений у додатку А.8

## 3.3 Навчання та тестування моделей

У моделях використовується алгоритм оптимізації під назвою Адам, який користується високою популярністю та визнанням у сфері глибокого навчання за рахунок високої ефективності. Назва "Адам" (Adam) є абревіатурою від англійських слів Adaptive Moment Estimation, яка відображає його ключову здатність – адаптивно оптимізувати навчальний процес.

Для забезпечення більшої ефективності та контролю за моделюванням

також використовуються функції зворотного виклику, такі як EarlyStopping та ReduceLROnPlateau. EarlyStopping дозволяє припинити навчання моделі до завершення запланованого процесу, якщо вона перестане покращувати свої результати, тим самим запобігаючи перенавчанню. ReduceLROnPlateau допомагає регулювати швидкість навчання, зменшуючи її у випадку, коли модель не показує покращення, що сприяє більш точному та ефективнішому підбору параметрів.

Крім того, в моделях застосовано техніку втрати центру, що використовується для вирішення проблеми зміни в середніх класах у глибокому навчанні. Ця методика передбачає використання спеціальної допоміжної функції втрат, яка допоможе підвищити щільність внутрішньокласових зв’язків, одночасно забезпечуючи чітке відокремлення між високими класами. Це дозволяє досягти більш точного представлення ознак, які модель вивчає, забезпечуючи більш ефективне та надійне розпізнавання.

Для повного розуміння технічного аспекту реалізацій моделей та обмежень, що були присутні при розробці цих моделей, далі буде наведено докладний опис програмного та апаратного забезпечення, яке використовувалося у процесі створення та тестування моделей. Ця інформація важлива для забезпечення прозорості та відтворюваності досліджень, а також для надання можливості іншим дослідникам адаптувати або розширити роботу. Деталі програмного та апаратного забезпечення, які використовуються в реалізаціях моделі, представлені на наступній таблиці (таблиця 3.2).

Таблиця 3.2 – Програмне та апаратне забезпечення для навчання моделей

| ПЗ / АЗ | Версія / Деталі |
|---|---|
| Python | 3.11.5 |
| TensorFlow | 2.14.0 |
| Keras | 2.14.0 |
| OpenCV | 4.8.1.78 |
| Розмір пакета даних | 128 |
| Кількість фреймів відео | 20 |
| Кількість епох для навчання | 50 або 100 |
| Розмір фрейма | 64x64 |
| Кількість обробників для нитей процесору | 4 |
| Використання GPU для навчання | Ні (використовується лише CPU) |
| Процесор | Apple M1 Pro |

Продовження таблиці 3.2.

| ПЗ / АЗ | Версія / Деталі |
|---|---|
| RAM | 32 GB |
| Оптимізатор | Adam |
| Метрики | Accuracy (точність), AUC |

Під час навчання кожної з моделей було зафіксовано інформацію про загальні втрати (total loss) та загальні втрати на валідації (total validation loss), а також загальну точність (total accuracy) та загальну точність на валідації (total validation accuracy) для кожної епохи навчання. Ці дані допомагають відстежувати та аналізувати ефективність моделі в процесі всього процесу навчання. Крім того, була збережена інформація про площу під кривою (AUC), яка є показником якості класифікаційних моделей, вказуючи на їх здатність розрізняти між класами.

Кожна з моделей використовує поточний набір даних з рівномірно розподіленими класами, і при цьому для механізмів випадковості використовується один початковий елемент (сід), що забезпечує узгодженість та повторюваність у навчальних моделях, що є критичним для об’єктивного порівняння їх результатів.

Далі, наведено список із реалізованими моделями:

1) ConvLSTM-50B (реалізація від Bleed);
2) LRCN-50B (реалізація від Bleed);
3) LRCN-100B (реалізація від Bleed);
4) LRCN-100D (реалізація від Djamaco);
5) LRCN-100D-BW (реалізація від Djamaco, відео пре-оброблено до чорно-білих фреймів);
6) LRCN-100D-Augm (реалізація від Djamaco, використано механізм аугментації при навчанні);
7) CNN-LSTM-100D-Augm-Atten (реалізація від Djamaco, використано механізми аугментації та уваги при навчанні).

На наступній таблиці 3.3 показано результат порівняння процесу навчання усіх розглянутих моделей.

Таблиця 3.3 – Порівняння результатів навчання для моделей

| Назва моделі | Кількість епох (фактична / максимальна) | Час навчання (мілісекунди) | Отримана точність (%) | Отримані втрати |
|---|---|---|---|---|
| ConvLSTM-50B | 26/50 | 8245822 | 63.0 | 1.672 |
| LRCN-50B | 50/50 | 2659457 | 57.8 | 1.677 |
| LRCN-100B | 100/100 | 5327893 | 71.8 | 1.253 |
| LRCN-100D | 63/100 | 5876008 | 78.5 | 0.952 |
| LRCN-100D-BW | 48/100 | 3786014 | 72.5 | 1.282 |
| LRCN-100D-Augm | 93/100 | 12481469 | 83.6 | 0.769 |
| CNN-LSTM-100D-Augm-Atten | 100/100 | 87082435 | 79.3 | 0.885 |

## 3.4 Аналіз побудованих моделей

### 3.4.1 ConvLSTM архітектура

Модель ConvLSTM-50B проходила тренування протягом 26 епох, що становить трохи більше половини від максимально запланованих 50 епох. Це призвело до скромної продуктивності, про що свідчить його точність 63,0% і значення втрати 1,672. Той факт, що навчання припинилося на 26-й епосі, означає, що модель, можливо, досягла точки достатньої конвергенції на ранній стадії, або це може вказувати на початок перенавчання, коли додаткове навчання не дало б значних покращень. Порівняно з іншими моделями в дослідженні продуктивність ConvLSTM-50B була відносно нижчою, що можна пояснити менш складною архітектурою або коротшою тривалістю навчання. Це вказує на можливість того, що розширення його навчання або підвищення його архітектурної складності може дати кращі результати.

### 3.4.2 LRCN архітектура

Різні версії моделі LRCN продемонстрували різноманітний спектр результатів, з точністю від 57,8% до помітних 83,6%, а показники втрат охоплювали від 0,769 до 1,677. Починаючи з варіанту "LRCN-50B", який тренувався протягом усього 50-епохового циклу, він зафіксував найнижчу

точність серед своїх аналогів – 57,8%. Цей результат свідчить про те, що ця конкретна модель може виграти від більш складного дизайну або інтеграції додаткових функцій для підвищення її продуктивності.

Переходячи до моделей «LRCN-100B» і «LRCN-100D», обидві продемонстрували покращену продуктивність завдяки більшої кількості епох та більш детальним налаштуванням відповідно. «LRCN-100D», зокрема, вирізнявся вражаючою точністю 78,5%. З іншого боку, варіант «LRCN-100D-BW», який представляв модель, що обробляє чорно-білі дані, зазнав незначного зниження продуктивності порівняно з аналогом «100D», але в свою чергу виграв у часі навчання та швидкістю сходження.

Найбільше заслуговує на увагу модель "LRCN-100D-Augm", яка включає в себе методи доповнення даних (аугментацію). Ця модель досягла найвищої точності з усіх – 83,6%, чітко демонструючи значний вплив, який збільшення даних може мати на підвищення ефективності моделей глибокого навчання. Це підкреслює потенціал використання різноманітних і збагачених навчальних наборів даних для підвищення можливостей навчання та загальної точності таких моделей. Успіх «LRCN-100D-Augm» є сильним показником цінності розширення даних у сфері глибокого навчання та розробки моделей, особливо в сценаріях, де складність і глибина моделі є вирішальними факторами. Можливою сферою дослідження у покращенні результатів буде об’єднання логіки та архітектури «LRCN-100D-Augm» із пре-обробкою фреймів у чорно-білий формат.

### 3.4.3 C3D-LSTM архітектура із механізмом уваги

Модель «CNN-LSTM-100D-Augm-Atten», яка має механізм привернення уваги, досягла помітного показника точності 79,3%. Ця модель виділялася своєю тривалою тривалістю навчання, найдовшою серед усіх оцінених моделей, але цікаво, що вона не досягла найвищої точності. Це явище свідчить про те, що додаткова складність, створена механізмом уваги, не обов’язково гарантує

пропорційне підвищення ефективності.

Однак відносно висока точність цієї моделі підкреслює переваги інтеграції механізмів уваги з архітектурами CNN і LSTM. Ця комбінація здається особливо ефективною для вирішення складних завдань, коли модель має більш ефективно зосереджуватися на конкретних функціях даних. Механізм уваги дозволяє моделі «зосереджуватися» на найбільш відповідних частинах вхідних даних, що може бути вирішальним у складних сценаріях, де кожна деталь має значення.

Таким чином, результати моделі CNN-LSTM-100D-Augm-Atten пропонують цінну інформацію про компроміси між складністю моделі, часом навчання та продуктивністю. Вони припускають, що в той час як розширені функції, такі як механізми уваги, можуть підвищити здатність моделі складно обробляти дані та навчатися на них, необхідно знайти баланс, щоб гарантувати, що додаткова складність виправдана підвищенням продуктивності. Ця модель служить прикладом того, як складні архітектури нейронних мереж можуть бути розроблені та налаштовані для вирішення вимогливих і складних завдань машинного навчання. Загалом, у результаті аналізу можна сказати, що подальше дослідження такої архітектури є доцільним та можливим місцем продовження дослідження є експерименти із архітектурою (її спрощення та ускладнення) та зміна CNN шарів на R(2+1)D або I3D шари як зазначено в статті за посиланням [39].

### 3.4.4 Загальні висновки по імплементованих моделях

Результати цих моделей чітко підкреслюють критичну роль, яку відіграють складність дизайну моделі та ступінь її навчання для досягнення високого рівня точності. Моделі, які є складнішими та проходять більш тривалий період навчання, такі як серія LRCN-100, як правило, демонструють чудову продуктивність. Це спостереження вказує на ефективність глибокого та ретельного навчання в покращенні можливостей навчання та прогнозування моделі.

Серед ключових висновків виділяється значний вплив розширення даних, особливо як показано на моделі "LRCN-100D-Augm". Успіх цієї моделі свідчить про те, що використання різноманітного та збагаченого навчального набору даних може значно покращити здатність моделі узагальнювати та ефективно працювати в різних сценаріях. У цьому контексті розширення даних стає потужним інструментом для підвищення глибини та різноманітності досвіду навчання для моделей.

З іншого боку, дослідження також виявляє цікаве застереження щодо надскладних моделей. Модель "CNN-LSTM-100D-Augm-Atten", незважаючи на свій механізм уваги, не досягла рівня підвищення точності, відповідного підвищеній складності. Цей висновок вказує на те, що більш складні функції, такі як механізми уваги, не завжди прирівнюються до кращої продуктивності, і що може бути така точка, коли мова заходить про додавання складності без відповідного результату.

Крім того, різні рівні успіху в різних моделях підкреслюють необхідність адаптації архітектури моделі, тривалості навчання та функцій до конкретних вимог поставленого завдання. Це не одна ситуація, яка підходить усім; різні проблеми можуть вимагати різних підходів для отримання оптимальних результатів.

# ВИСНОВКИ

У ході виконання цієї роботи було розглянуто архітектури моделей глибокого навчання, такі як CNN, RNN та LSTM, та їх застосування в машинному навчанні та штучному інтелекті. Моделі CNN ідеально підходять для аналізу даних зображення, оскільки вони ефективно вилучають просторові особливості. З іншого боку, моделі RNN, зокрема LSTM, виявилися ефективними для аналізу часових рядів завдяки їхній здатності враховувати тимчасові залежності. Експерименти на різних наборах даних [4] підтверджують, що CNN відмінно справляється з завданнями класифікації зображень, тоді як LSTM показує високу ефективність у аналізі часових рядів.

Далі, було розглянуто основні архітектури гібридних моделей глибокого навчання та їх алгоритми, котрі можуть розпізнавати діяльність людини на відео в залежності від контексту, представленого на ньому.

Спираючись на отриманні знання, було розроблено декілька гібридних моделей, які об’єднали сильні сторони архітектур CNN, RNN і LSTM. Ці моделі були ретельно навчені та перевірені на наборах даних, щоб оцінити їх продуктивність у різних сценаріях. Завдяки порівнянню їхніх результатів було отримано цінні перспективи щодо того, як поєднання цих архітектур покращує загальну продуктивність моделі, особливо під час вирішення складних завдань, пов’язаних як з просторовими, так і з часовими даними, що можуть служити прикладом контексту при аналізі людської діяльності.

Підводячи підсумок після аналізу усіх імплементованих моделей можна сказати, що вони ілюструють важливий урок у сфері машинного та глибокого навчання: підвищення складності моделі та диверсифікація навчальних даних справді можуть призвести до покращення результатів, але вкрай важливо знайти тонкий баланс. Оптимізація моделі — це пошук правильного поєднання складності, глибини навчання та включення функцій, що відповідає конкретному характеру завдання, забезпечуючи ефективність без надмірного навантаження на модель непотрібними складнощами.

У якості наступних кроків для можливого дослідження для поліпшення результатів та виправлення можливих помилок розглядаються наступні кроки:

1. Експериментування з різними параметрами. Налаштування та експериментування з різними параметрами в кожній архітектурі може призвести до більш оптимізованих моделей, потенційно розкриваючи більш ефективні способи обробки даних.

2. Чорно-біла попередня обробка. Застосування чорно-білої попередньої обробки відеоданих для більшої кількості моделей може дати зрозуміти, як інформація про колір впливає на продуктивність моделі та чи може спрощення візуальних даних допомогти в певних типах аналізу.

3. Впровадження R(2+1)D та I3D CNN. Дослідження використання R(2+1)D та I3D CNN в архітектурах CNN-LSTM може дати покращення, особливо в завданнях, які вимагають тонкого розуміння просторових і часових моделей.

4. Експерименти з даними, котрі не є відео. Експериментування з різними типами даних, таких як дані акселерометра або вимірювача пульсу, може відкрити нові можливості в розпізнаванні дій людини на основі контенту, урізноманітнивши сфери застосування цих моделей та показати як різні моделі можуть краще вирішувати свій власний спектр задач.

5. Використання різних систем і GPU. Тестування моделей на різних системах і використання графічних процесорів для процесу навчання може виявити, як різні апаратні конфігурації впливають на ефективність навчання та продуктивність моделей.

6. Використання GRU та biLSTM замість LSTM у гібридних моделях.

# ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1 Sunder Ali Khowaja, Bernardo Nugroho Yahya, Seok-Lyong Lee. CAPHAR: context-aware personalized human activity recognition using associative learning in smart environments [Електронний ресурс]. – Режим доступу: https://hcis-journal.springeropen.com/articles/10.1186/s13673-020-00240-y (дата звернення: 01.10.2023).

2 Weiping Ding, Mohamed Abdel-Basset, Reda Mohamed. HAR-DeepConvLG: Hybrid deep learning-based model for human activity recognition in IoT applications [Електронний ресурс]. – Режим доступу: https://www.sciencedirect.com/science/article/abs/pii/S0020025523009799 (дата звернення: 03.10.2023).

3 Imran Ullah Khan, Sitara Afzal, Jong Weon Lee. Human Activity Recognition via Hybrid Deep Learning Based Model [Електронний ресурс]. – Режим доступу: https://www.mdpi.com/1424-8220/22/1/323 (дата звернення: 03.10.2023).

4 Farhad Mortezapour Shiri, Thinagaran Perumal, Norwati Mustapha, Raihani Mohamed. A Comprehensive Overview and Comparative Analysis on Deep Learning Models: CNN, RNN, LSTM, GRU [Електронний ресурс]. – Режим доступу: https://browse.arxiv.org/pdf/2305.17473.pdf (дата звернення: 03.10.2023).

5 Understanding Deep Learning: DNN, RNN, LSTM, CNN and R-CNN [Електронний ресурс]. – Режим доступу: https://medium.com/@sprhlabs/understanding-deep-learning-dnn-rnn-lstm-cnn-and-r-cnn-6602ed94dbff (дата звернення: 03.10.2023).

6 Intuitive Comparison of Four NLP Models - Neural Network, RNN, CNN, and LSTM [Електронний ресурс]. – Режим доступу: https://www.alibabacloud.com/blog/599283 (дата звернення: 04.10.2023).

7 Паламарчук І.О. Система розпізнавання домашніх тварин для розумного дому [Електронний ресурс]. – Режим доступу: https://ela.kpi.ua/jspui/bitstream/123456789/31801/1/Palamarchuk_magistr.pdf, (дата звернення: 28.10.2023).

8 What is Machine Learning? [Електронний ресурс]. – Режим доступу:

https://aws.amazon.com/ru/what-is/machine-learning (дата звернення: 28.10.2023).

9 What is Deep Learning? [Електронний ресурс]. – Режим доступу: https://aws.amazon.com/what-is/deep-learning/ (дата звернення: 28.10.2023).

10 What’s the Difference Between Machine Learning and Deep Learning? [Електронний ресурс]. – Режим доступу: https://aws.amazon.com/ru/compare/the-difference-between-machine-learning-and-deep-learning/ (дата звернення: 28.10.2023).

11 Kei Tanigaki, Tze Chuin Teoh, Naoya Yoshimura, Takuya Maekawa, Takahiro Hara. Predicting Performance Improvement of Human Activity Recognition Model by Additional Data Collection [Електронний ресурс]. – Режим доступу: https://dl.acm.org/doi/pdf/10.1145/3550319 (дата звернення: 28.10.2023).

12 Zheqi Yu, Adnan Zahid, Shuja Ansari, Hasan Abbas, Hadi Heidari, Muhammad A. Imran, Qammer H. Abbasi. IMU Sensing–Based Hopfield Neuromorphic Computing for Human Activity Recognition [Електронний ресурс]. – Режим доступу: https://www.frontiersin.org/articles/10.3389/frcmn.2021.820248/full (дата звернення: 29.10.2023).

13 Wei Zhong Tee, Rushit Dave, Jim Seliya, Mounika Vanamala. A Close Look into Human Activity Recognition Models using Deep Learning [Електронний ресурс]. – Режим доступу: https://arxiv.org/ftp/arxiv/papers/2204/2204.13589.pdf (дата звернення: 29.10.2023).

14 Loknath Sai Ambati, Omar El-Gayar. Human Activity Recognition: A Comparison of Machine Learning Approaches [Електронний ресурс]. – Режим доступу: https://www.metrostate.edu/sites/default/files/2021-02/A4-2021-jan.pdf (дата звернення: 29.10.2023).

15 F-міра [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/F-%D0%BC%D1%96%D1%80%D0%B0 (дата звернення: 29.10.2023).

16 Bengio Y., Lecun, Y. Convolutional Networks for Images, Speech, and Time-Series. 1997. [Електронний ресурс]. – Режим доступу: https://www.researchgate.net/profile/Yann_Lecun/publication/2453996_Convolution

al_Networks_for_Images_Speech_and_TimeSeries/links/0deec519dfa232550200000 0.pdf (дата звернення: 29.10.2023).

17 Khandelwal R. Convolutional Neural Network (CNN) Simplified. 2018. [Електронний ресурс]. – Режим доступу: https://medium.com/datadriveninvestor/convolutional-neuralnetwork-cnn-simplified-ecafd4ee52c5 (дата звернення: 29.10.2023).

18 Лейзьо С.І. «Бібліотека з функцією форуму на основі глибинного навчання [Електронний ресурс]. – Режим доступу: https://ela.kpi.ua/jspui/bitstream/123456789/46203/1/Leizo_magistr.pdf (дата звернення: 29.10.2023).

19 Рязановський. К.Д. Структурно-параметричний синтез гібридних нейронних мереж ансамблевої топології [Електронний ресурс]. – Режим доступу: https://ela.kpi.ua/bitstream/123456789/35615/1/Riazanovskii_bakalavr.pdf (дата звернення: 29.10.2023).

20 Качан Д.С. «Прогнозування цін акцій методами глибоких нейронних мереж [Електронний ресурс]. – Режим доступу: https://ela.kpi.ua/bitstream/123456789/61298/1/Kachan_bakalavr.pdf (дата звернення: 29.10.2023).

21 Human Activity Recognition (HAR): Fundamentals, Models, Datasets [Електронний ресурс]. – Режим доступу: https://www.v7labs.com/blog/human-activity-recognition#h1 (дата звернення: 04.12.2023).

22 Офіційний сайт PyTourch [Електронний ресурс]. – Режим доступу: https://pytorch.org/ (дата звернення: 04.12.2023).

23 Офіційний сайт NumPy [Електронний ресурс]. – Режим доступу: https://numpy.org/ (дата звернення: 04.12.2023).

24 Офіційний сайт Scikit-learn [Електронний ресурс]. – Режим доступу: https://scikit-learn.org/stable/ (дата звернення: 04.12.2023).

25 Офіційний сайт Pandas [Електронний ресурс]. – Режим доступу: https://pandas.pydata.org (дата звернення: 04.12.2023).

26 Офіційний сайт Keras [Електронний ресурс]. – Режим доступу: https://keras.io (дата звернення: 04.12.2023).

27 Офіційний сайт Tensorflow [Електронний ресурс]. – Режим доступу: https://tensorflow.org (дата звернення: 04.12.2023).

28 Офіційний сайт Ray.io [Електронний ресурс]. – Режим доступу: https://ray.io (дата звернення: 04.12.2023).

29 UCF101 - Action Recognition Data Set [Електронний ресурс]. – Режим доступу: https://www.crcv.ucf.edu/data/UCF101.php (дата звернення: 04.12.2023).

30 Human Activity Recognition using TensorFlow (CNN + LSTM) [Електронний ресурс]. – Режим доступу: https://bleedaiacademy.com/human-activity-recognition-using-tensorflow-cnn-lstm (дата звернення: 04.12.2023).

31 Hamik Bhattacharjee, Rasha Alshehhi, Dattaraj Dhuri, Shravan M. Hanasoge, Supervised convolutional neural networks for classification of flaring and non-flaring active regions using line-of-sight magnetograms [Електронний ресурс]. – Режим доступу: https://www.researchgate.net/publication/341699454_Supervised_convolutional_neural_networks_for_classification_of_flaring_and_nonflaring_active_regions_using_line-of-sight_magnetograms (дата звернення: 04.12.2023).

32 Kuzomin O. CREATION OF INTELLIGENT SYSTEMS FOR ANALYZING SUPERMARKET VISITORS TO IDENTIFY CRIMINAL ELEMENTS /Berkovskyi, D., Kuzomin O. // Collection of Scientific Papers «SCIENTIA», (May 5, 2023; Sydney, Australia), pp. 113–118.

33 Л6. Рекурентні нейронні мережі [Електронний ресурс]. – Режим доступу: https://www.youtube.com/watch?v=UF7kqjad1Mg (дата звернення: 04.12.2023).

34 Згортка (обробка зображень) [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Згортка_(обробка_зображень) (дата звернення: 04.12.2023).

35 Добуток Адамара [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Добуток_Адамара (дата звернення: 04.12.2023).

36 ConvLSTM [Електронний ресурс]. – Режим доступу: https://paperswithcode.com/method/convlstm (дата звернення: 04.12.2023).

37 Jeff Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Long-term recurrent convolutional networks for visual recognition and description

[Електронний ресурс]. – Режим доступу: https://www.researchgate.net/publication/308034527_Long-term_recurrent_convolutional_networks_for_visual_recognition_and_description (дата звернення: 04.12.2023).

38 Репозиторій дипломного застосунку «har_project» [Електронний ресурс]. – Режим доступу: https://github.com/djamaco/har_project (дата звернення: 04.12.2023).

39 El Mehdi Saoudi, Jaafar Jaafari, Said Jai Andaloussi, Advancing human action recognition: A hybrid approach using attention-based LSTM and 3D CNN [Електронний ресурс]. – Режим доступу: https://www.sciencedirect.com/science/article/pii/S2468227623002521 (дата звернення: 04.12.2023).