

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління  
(повна назва)

Кафедра Автоматизації проектування обчислювальної техніки  
(повна назва)

## КВАЛІФІКАЦІЙНА РОБОТА

### Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Система розпізнавання облич на основі  
нейронної мережі

(тема)

Виконав:

Студент II курсу, групи СКСМ-22-2  
Кахаєв М.С.  
(прізвище, ініціали)

Спеціальність 123 – Комп'ютерна інженерія  
(код і повна назва спеціальності)

Тип програми освітньо-професійна  
(освітньо-професійна або освітньо-наукова)

Освітня програма Спеціалізовані  
комп'ютерні системи  
(повна назва освітньої програми)

Керівник: доц. Ларченко Л.В.  
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри АПОТ

  
(підпис)


Чумаченко С.В.  
(прізвище, ініціали)

2023 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління  
Кафедра Автоматизації проектування обчислювальної техніки  
Рівень вищої освіти другий (магістерський)  
Спеціальність 123 – Комп'ютерна інженерія  
(код і повна назва)  
Тип програми освітньо-професійна  
(освітньо-професійна або освітньо-наукова)  
Освітня програма Спеціалізовані комп'ютерні системи  
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри   
(підпис)

“ ” \_\_\_\_\_ 20\_\_ р.

**ЗАВДАННЯ  
НА АТЕСТАЦІЙНУ РОБОТУ**

студентові Кахаєву Михайлу Сергійовичу  
(прізвище, ім'я, по батькові)

1. Тема роботи Система розпізнавання облич на основі нейронної мережі

затверджена наказом по університету від " 03 " 11 2023 р. № 1282 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 19.01.2024

3. Вихідні дані до роботи

Згорткові нейронні мережі (CNN)

Мова програмування Python

Програмне забезпечення для глибокого навчання TensorFlow

4. Перелік питань, що потрібно опрацювати в роботі

1 Аналіз предметної області та постановка завдання

2 Дослідження архітектур нейронних мереж для розпізнавання облич

3 Аналіз новітніх тенденцій

4 Дослідження математичної бази

5 Розробка моделей нейронних мереж на основі попередніх досліджень

## 6 Дослідження ефективності розроблених моделей

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) 18 слайдів

6. Консультанти розділів роботи (заповнюється 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	Термін виконання етапів роботи	Примітка
1	Отримання завдання	01.09.2023-06.09.2023	
2	Аналіз предметної області	06.09.2023-30.09.2023	
3	Аналіз джерел з проблемної галузі	30.09.2023-08.10.2023	
4	Розробка архітектури нейронної мережі для розпізнавання облич	08.10.2023-12.10.2023	
5	Підбір навчальної вибірки для проведення дослідження	12.10.2023-20.10.2023	
6	Написання програмної реалізації розроблених моделей нейронних мереж	20.10.2023-20.11.2023	
7	Проведення дослідження ефективності розроблених моделей	20.11.2023-30.11.2023	
8	Оформлення пояснювальної записки	30.11.2023-10.12.2023	
9	Оформлення графічного матеріалу	10.12.2023-20.12.2023	
10	Перевірка виконаного проекту керівником	20.12.2023-30.12.2023	

Дата видачі завдання 01.09.2023

Студент Кох  
(підпис)

Керівник роботи ЛВ  
(підпис)

доц. Ларченко Л.В.  
(посада, прізвище, ініціали)

## РЕФЕРАТ

Пояснювальна записка містить 66 сторінок, 23 рисунків, 2 таблиці, 17 джерел за переліком посилань.

### РОЗПІЗНАВАННЯ ОБЛИЧ, НЕЙРОННІ МЕРЕЖІ, DEEP LEARNING, MACHINE LEARNING, РЕГУЛЯРИЗАЦІЯ, PYTHON, 3D-CNN, BACKPROPAGATION

В кваліфікаційній роботі досліджено та розроблено систему розпізнавання облич на основі нейронної мережі з використанням новітніх підходів до розробки архітектури і навчання.

На основі аналізу джерел виявлено перспективну технологію для дослідження її ефективності у порівнянні з стандартними підходами до розпізнавання облич, а саме використання тривимірних даних для навчання замість двовимірних. Проведено дослідження теоретичної бази і виявлено необхідні інструменти для ефективного реалізації моделей з використанням 3D і 2D даних. Досліджено основні математичні аспекти в проектуванні нейронних мереж, проведено розрахунки для основного механізму роботи всіх нейронних мереж, а саме backpropagation. Проведено реалізацію архітектури 3D-CNN і 2D-CNN мовою Python і підібрані відповідні навчальні сети. На основі розроблених моделей проведено дослідження ефективності використання відповідних навчальних даних, а також виявлені необхідні алгоритми регуляризації і їх ефективність у поєднанні з вищезазначеними архітектурами. Проведено аналіз отриманих результатів і дані рекомендації щодо використання досліджуваного методу на практиці.

## ABSTRACT

The explanatory note contains 66 pages, 23 figures, 2 tables, and 17 references.

FACE RECOGNITION, NEURAL NETWORKS, DEEP LEARNING, MACHINE LEARNING, REGULARIZATION, PYTHON, 3D-CNN, BACKPROPAGATION

In the qualification work, neural network models for face recognition are researched and developed using the latest approaches to architecture development and training. The object of research is the effectiveness of the latest approaches in the development of neural networks.

Based on the analysis of sources, a promising technology has been identified to study its effectiveness in comparison with standard approaches, namely the use of three-dimensional data for training instead of two-dimensional data. The theoretical framework is studied and the necessary tools for effective implementation of models using 3D and 2D data are identified. The main mathematical aspects of neural network design are investigated, and calculations are made for the main mechanism of all neural networks, namely backpropagation. The 3D-CNN and 2D-CNN architectures were implemented in Python and the corresponding training networks were selected. On the basis of the developed models, the efficiency of using the relevant training data was studied, and the necessary regularization algorithms and their effectiveness in combination with the above architectures were identified. The results obtained are analyzed and recommendations are given on the use of the investigated method in practice.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ .....	8
ВСТУП.....	9
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ.....	11
1.1 Огляд принципів роботи і проектування систем розпізнавання облич на основі нейронних мереж.....	11
1.2 Традиційні методи розпізнавання облич .....	15
1.3 Огляд новітніх трендів.....	19
1.4 Постановка завдання.....	21
2 ДОСЛІДЖЕННЯ АРХІТЕКТУР НЕЙРОННИХ МЕРЕЖ ДЛЯ РОЗПІЗНАВАННЯ ОБЛИЧ.....	23
2.1 Базові архітектури.....	23
2.2 Модернізація CNN .....	26
2.3 Підготовка даних .....	27
2.4 Гіперпараметри та функції активації .....	30
2.4.1 Гіперпараметри .....	30
2.4.2 Функції активації .....	31
2.5 Методи регуляризації і нормалізації.....	34
3 ДОСЛІДЖЕННЯ МАТЕМАТИЧНОЇ БАЗИ.....	37
3.1 Нейрон .....	37
3.2 Функція втрат .....	38
3.3 Ваги між нейронами і механізм їх корегування .....	40
3.4 Регуляризація .....	44
4 РЕАЛІЗАЦІЯ ДОСЛІДЖУВАНОЇ МОДЕЛІ ТА ПРОВЕДЕННЯ ДОСЛІДЖЕННЯ .....	46
4.1 Архітектура моделі .....	46
4.2 Збір даних .....	49

4.3 Перетворення 3D моделей в данні для навчання .....	50
4.4 Проведення навчання мереж без регуляризації .....	51
4.5 Проведення навчання мереж з регуляризацією .....	55
4.6 Аналіз отриманих даних .....	59
ВИСНОВКИ .....	63
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ .....	65
ДОДАТОК А .....	67
ДОДАТОК Б .....	76

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

CNN - Convolutional Neural Networks - згорткові нейронні мережі

RNN - Recurrent Neural Networks - рекурентні нейронні мережі

CRNN - Convolutional Recurrent Neural Networks - згорткові рекурентні нейронні мережі

PCA - Principal Component Analysis - аналіз головних компонентів

GMM - Gaussian Mixture Models - змішані моделі Гауса

HMM - Hidden Markov Models - скриті Марковські моделі

ReLU - Rectified Linear Unit - випрямлена лінійна функція

GAN - Generative Adversarial Network - генеративно змагальні мережі

LSTM - Long Short-Term Memory - довготривала короткочасна пам'ять

GRU - Gated Recurrent Unit - модифікована рекурентна нейронна мережа, яка використовує механізм воріт

RGB - Red Green Blue - червоний зелений синій

GCNs - Graph Convolutional Networks - графові згорткові мережі

SGD - Stochastic Gradient Descent - стохастичний градієнтний спуск

MSE - Mean Squared Error - середньоквадратична помилка

LFW - Labeled Faces in the Wild - позначені обличчя в природних умовах

CASIA - Chinese Academy of Sciences Institute of Automation - інститут автоматизації китайської академії наук

## ВСТУП

Системи розпізнавання облич на основі нейронної мережі є однією з найцікавіших та актуальних областей досліджень у сфері комп'ютерного зору та штучного інтелекту. Вони займають центральне місце у багатьох сучасних технологіях, таких як автоматичне розпізнавання осіб, системи безпеки, контроль доступу, відео спостереження та багато інших. Завдяки зростанню обчислювальних можливостей та проривам у глибокому навчанні, нейронні мережі стали потужним інструментом для розв'язання завдань розпізнавання облич. Вони дозволяють автоматично виявляти та розпізнавати обличчя на зображеннях і відео з високою точністю і швидкістю. Однак, щоб досягти оптимальної продуктивності таких систем, потрібно проводити дослідження та оптимізацію їх архітектури і параметрів.

У цій роботі робиться акцент на аналізі і дослідженні нових перспективних методах в проектуванні і розробці нейронних мереж для систем розпізнавання облич.

При дослідженні архітектури нейронних мереж було проаналізовано різні архітектури нейронних мереж для завдання розпізнавання облич, визначено найбільш раціональні архітектури для проведення дослідження.

При дослідженні методів регуляризації були проведені дослідження щодо різних методів. Метою є визначення оптимального підходу для регуляризації даних, який допоможе підвищити точність розпізнавання облич для розроблених моделей.

Використання 3D моделей обличчя: однією з ключових інновацій цієї роботи є дослідження використання тривимірних (3D) моделей обличчя замість звичайних двовимірних (2D) зображень. Метою є визначення яким чином використання 3D інформації може покращити розпізнавання облич та дозволити системі краще опрацьовувати данні. Цей аспект дослідження є

особливо актуальним у випадках, коли точність розпізнавання грає критичну роль, наприклад, у системах безпеки або медичних застосунках.

Метою кваліфікаційної роботи є розробка системи розпізнавання облич на основі нейронної мережі. Основний акцент роботи полягає у дослідженні ефективності удосконаленої моделі нейронної мережі для розпізнавання облич. Дана робота спрямована на розвиток нових тенденцій у розробці нейронних мереж.

Результати кваліфікаційної роботи мають практичне застосування в сучасному світі і можуть допомогти поліпшити системи безпеки, створити ефективні інструменти для відстеження та аналізу облич у великих обсягах даних, а також сприяти розвитку багатьох інших сфер. Напрямок роботи визначає важливість і актуальність вивчення та розвитку систем розпізнавання облич на основі нейронної мережі та закликає до подальших досліджень у цій області.

## 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ

В розділі наводиться аналіз процесу проектування систем розпізнавання облич. Проаналізовано основні існуючі і новітні тенденції в розробці актуальних систем розпізнавання облич, які в змозі відповідати сучасним вимогам та поставленим завданням, сформовано мету і постановку завдання дослідження.

### 1.1 Огляд принципів роботи і проектування систем розпізнавання облич на основі нейронних мереж

Розпізнавання облич є однією з найважливіших та актуальних проблем у сфері комп'ютерного зору та штучного інтелекту. Проблема розпізнавання облич відноситься до задачі автоматичного визначення та ідентифікації осіб на зображеннях або відео за допомогою комп'ютерних алгоритмів.

Процес ідентифікації або верифікації особи здійснюється на основі її фізичних ознак, таких як обличчя, його вирази, структура та інші біометричні параметри. Цей процес має низку важливих застосувань в багатьох сферах, таких як безпека, медицина, автоматизована ідентифікація та відновлення даних, робототехніка, розваги та багато інших.

Зокрема, системи розпізнавання облич використовуються у системах безпеки для контролю доступу, розслідування злочинів та виявлення загроз. Банківський сектор використовує системи розпізнавання облич для автоматичного розпізнавання клієнтів та забезпечення безпеки фінансових транзакцій. Крім зазначених систем, вони використовуються в розважальних технологіях, де дозволяють створювати інтерактивні ігри та розважальні додатки.

Проектування систем розпізнавання облич на основі нейронних мереж є складним завданням та вимагає глибокого розуміння як загальних

принципів нейронних мереж, так і специфічних особливостей завдання розпізнавання облич. Для виконання проектування нейронної мережі необхідно провести збір даних, на цьому етапі можна визначити яким способом буде здійснюватись їх обробка, потім вибирається архітектура та гіперпарметри (виконується перше проектування нейронної мережі), надалі проводиться аналіз та вводяться коригування залежно від отриманого результату, потім цикл із тренування та коригування повторюється поки не будуть отримані задовільні показники розробленої моделі нейронної мережі.

Збір та підготовка даних є одним із найважливіших етапів проектування системи розпізнавання облич на основі нейронних мереж. Правильно підготовлені дані грають важливу роль у досягненні високої точності та продуктивності системи. Для цього треба зібрати навчальні дані, які будуть використовуватися для тренування нейронної мережі. Вони мають бути якомога більше різноманітними з точки зору осіб, поз та умов освітлення. Проведення аугментації робиться для створення додаткових даних шляхом застосування трансформацій до існуючих фотографій. Це допомагає збільшити різноманітність та покращити загальну ефективність навчання. Можливі методи аугментації включають вирізання, обертання, зміну масштабу, відображення та додавання шуму.

Зібрані дані потрібно перевести в числовий формат, який може бути використаний нейронною мережею, що означає перетворення кожного пікселя на числове значення, наприклад, в діапазоні від 0 до 255 (якщо використовується 8-бітова градація сірого). Дані розділяють на тренувальну, валідаційну та тестову вибірки у пропорції 70% на 15% на 15% відповідно при відносно невеликій кількості зібраних даних (до 100 000) і пропорції 90:5:5 відповідно при наявності великої кількості даних[1].

За необхідністю або при виявленні під час навчання некоректних результатів можуть бути використані корегуючі дії, такі як балансування класів. Балансування класів виконується шляхом додавання більшої кількості прикладів менш представлених класів або застосування ваг для класів під час

навчання, наприклад, тоді, коли класи облич нерівномірно розподілені в даних і недостатня наявність прикладів жіночих або чоловічих облич в тій чи іншій вибірці тощо. При виконанні балансування класів здійснюється перевірка та очищення даних, шляхом їх видалення, на наявність помилок або артефактів, які можуть вплинути на тренування моделі. Також для зменшення ймовірності виникнення помилок і артефактів використовується нормалізація даних, яка полягає у зменшенні шуму та спрощенні тренування моделі, що включає в себе центрування і масштабування даних.

Вибір архітектури нейронної мережі є критичним етапом при проектуванні системи розпізнавання облич. Різні архітектури можуть мати різні характеристики і призначення. Для більшості завдань розпізнавання облич глибокі згорткові нейронні мережі (CNN) показують добрі результати. Деякі популярні архітектури, які широко використовуються, включають VGG, Inception, ResNet, MobileNet, EfficientNet, вони вже мають підтверджену ефективність у сфері розпізнавання облич.

Після визначення загальної архітектури нейронної мережі також необхідно визначити різні аспекти, які можуть бути налаштовані для досягнення оптимальної продуктивності в завданні розпізнавання облич, які розглянуті нижче:

- глибина мережі (кількість шарів): глибокі нейронні мережі мають багато шарів, що дозволяє їм вивчати складні закономірності в даних. Однак, занадто велика глибина може призвести до проблем з перенавчанням, особливо якщо недостатньо навчальних даних. Необхідно обирати оптимальну кількість шарів, яка відповідає сету даних і завданню;

- функція активації визначає, як нейрони реагують на вхідні сигнали. Для завдань розпізнавання облич використовують функції активації, такі як ReLU або їх модифікації. Функція активації впливає на швидкість навчання і здатність мережі вивчати нелінійні залежності;

- фільтри та їх кількість: у згорткових шарах мережі використовуються фільтри для виявлення особливостей в зображеннях. Розмір і кількість

фільтрів впливають на здатність мережі виявляти різні образи. Великі фільтри можуть виявити більші особливості, але вимагають більше обчислювальних ресурсів;

- швидкість навчання (learning rate) визначає крок, яким змінюються ваги під час навчання. Велика швидкість навчання може призвести до швидкого навчання, але може бути нестійкою та викликати перенавчання. Надто мала швидкість навчання може призвести до повільного навчання або застрягання в локальних мінімумах;

- регуляризація включає методи для попередження перенавчання або недонавчання, такі як dropout, L1- та L2-регуляризація. Вони допомагають покращити загальну продуктивність мережі;

- розмір пакету (batch size) впливає на швидкість навчання та використання обчислювальних ресурсів. Великі пакети можуть пришвидшити навчання, але вимагають більше пам'яті;

- вибір правильної функції втрат та метрики є важливим для визначення того, як мережа оцінюється та тренується;

- адаптація до різних діапазонів значень пікселів: якщо дані мають різний діапазон значень пікселів, то важливо нормалізувати ці значення, щоб забезпечити стабільність навчання.

Процес навчання нейронної мережі зазвичай включає кілька десятків епох тренування. Одна епоха вказує на одну ітерацію навчання з усіма тренувальними даними. Власне процес тренування нейронної мережі включає в себе етапи валідації, налаштування гіперпараметрів та зупинки тренування для досягнення оптимальної продуктивності моделі[2]. Докладний опис цих етапів наведено нижче:

- після кожної епохи тренування, модель оцінюється на окремому валідаційному наборі даних. Це важливий етап для визначення, наскільки добре модель узгоджується з новими даними та для визначення, чи відбувається перенавчання;

- після валідації обчислюються метрики продуктивності моделі на

валідаційних даних. Даний етап включає в себе обчислення точності, втрат, витрат часу та інших метрик, які відображають якість моделі;

- аналіз результатів валідації допомагає визначити, наскільки добре модель працює. Якщо результати покращуються, то тренування може продовжуватися. Якщо якість моделі не задовольняє, це може сигналізувати про нестабільність навчання або перенавчання;

- після аналізу результатів валідації може виникнути потреба в налаштуванні гіперпараметрів моделі. Гіперпараметри включають в себе швидкість навчання (learning rate), кількість шарів, розмір пакету (batch size), функції активації, регуляризацию тощо. Можливі зміни гіперпараметрів для покращення стабільності та уникнення перенавчання;

- після внесення змін у гіперпараметри навчання модель повторно тренується з оновленими налаштуваннями. Процес тренування може повторюватися декілька разів, до того моменту, як буде досягнута оптимальна продуктивність;

- зупинка тренування може відбутися, якщо результати на валідаційних даних не покращуються протягом декількох епох або якщо перенавчання стає очевидним.

## 1.2 Традиційні методи розпізнавання облич

Традиційні методи розпізнавання облич були розвинуті до появи глибинного навчання та нейронних мереж і використовували різні підходи для вирішення задачі класифікації зображення. Основні методи можна розділити на дві категорії: методи на основі особливостей (feature-based) та методи на основі моделей (model-based).

Методи на основі особливостей базуються на виділенні та описуванні характеристик обличчя, які використовуються для розпізнавання. Найпоширенішими особливостями, які використовуються, є габаритні орієнтири (риси), текстури та геометричні форми обличчя. Однак,

використання методів на основі особливостей має обмеження, пов'язане зі складністю опису всіх можливих варіацій облич та проблем пов'язаних з універсальністю. Найкращим прикладом, що демонструє даний підхід є метод Хаара.

Метод Хаара - це метод в області комп'ютерного зору та обробки зображень, який використовує особливий тип фільтрів, відомих як фільтри Хаара, для виявлення областей зображення з різними інтенсивностями[3].

Основна ідея методу Хаара полягає у використанні прямокутних фільтрів різного розміру і форми для визначення змін інтенсивності пікселів у різних частинах зображення. Фільтри Хаара можуть виявляти зміни інтенсивності вздовж вертикальних, горизонтальних і діагональних ліній, а також комбінації цих напрямків.

Нижче розглянуто процес виявлення обличчя за допомогою методу Хаара.

1. Створення векторів ознак (feature vectors): на цьому етапі виконується визначення прямокутних фільтрів Хаара різних розмірів і форм (рис. 1.1). Кожен фільтр генерує вектор ознак, який представляє різницю між сумою пікселів в білій та чорній областях фільтра.

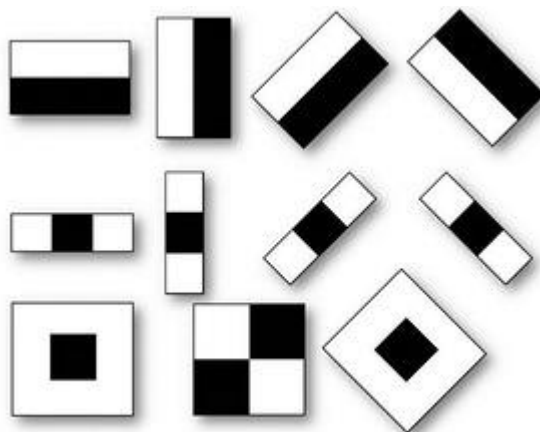


Рисунок 1.1 - Приклад векторів ознак

2. Використання каскадів класифікаторів: каскад класифікаторів - це

послідовність класифікаторів, кожен з яких використовується для відсіювання нецільових областей. Задачею класифікатору у каскаді є розрізнити зображення з обличчям і тими, де його немає.

3. Сканування зображення: представляє собою проведення сканування зображення за допомогою каскаду класифікаторів для виявлення областей, які можуть містити обличчя (рис. 1.2).

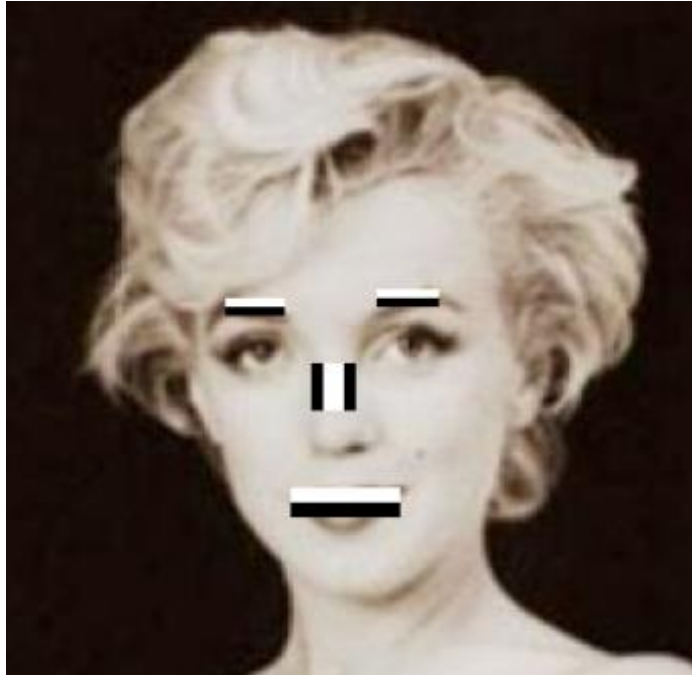


Рисунок 1.2 - Приклад сканування зображення методом Хаара

4. Додаткова обробка: на цьому кроці застосовуються додаткові етапи обробки, такі як уточнення областей, виявлених методом Хаара, і впровадження додаткових евристик для поліпшення точності виявлення обличчя.

Модельно-орієнтовані методи для розпізнавання обличч базуються на створенні або використанні моделей, які описують або представляють обличчя. Ці методи можуть включати як статичні, так і динамічні характеристики обличчя[3]. Популярними моделями у цьому підході є модель головних компонентів (PCA), модель суміші Гауса (GMM), модель

прихованих Марківських ланцюгів (НММ).

РСА є методом лінійного аналізу, який використовується для зменшення розмірності даних. Головна ідея полягає в тому, щоб знайти нові ортогональні вектори, які представляють напрямки максимальної дисперсії у даних. Зменшуючи розмірність, РСА дозволяє втратити мінімальну кількість інформації. В контексті розпізнавання обличчя, РСА може бути використаний для зменшення розмірності простору ознак, що поліпшує обчислювальну ефективність і може покращити роботу алгоритмів розпізнавання[3].

GMM - це статистична модель, що представляє розподіл даних як суміш гаусівських розподілів. Кожен компонент GMM характеризується власним набором параметрів, таких як середнє значення, матриця коваріації та вага компонента[3]. GMM може бути використаний для моделювання розподілу ознак обличчя. Кожен компонент може відповідати певному аспекту обличчя, наприклад, очам, носу чи роту (рис. 1.3).



Рисунок 1.3 – Приклад розпізнавання за методом GMM

НММ - це статистична модель, яка використовується для моделювання систем зі станами, які можуть бути прихованими. Система рухається між різними станами за допомогою випадкових переходів, і кожен стан видає відомості про спостереження відповідної стану випадкової величини. НММ може бути використаний для моделювання часових залежностей в

динамічних зображеннях обличчя[3]. Наприклад, визначення різних фаз руху обличчя, які можуть бути корисними для виявлення жестів, виразів або розпізнавання міміки.

Однак, традиційні методи розпізнавання обличчя мають свої обмеження, бо зазвичай вимагають побудованої вручну моделі обличчя або вибору певних характеристик, що можуть бути чутливі до змін у вигляді обличчя, освітлення та пози. Окрім того, традиційні методи не завжди забезпечують достатню точність та швидкодію для застосування в реальному часі чи великомасштабних системах.

### 1.3 Огляд новітніх трендів

За останній час з'явилося багато нових трендів в розробці систем розпізнавання обличчя, що свідчить про неймовірний розвиток цієї галузі. Таким чином в останні роки замість традиційних методів заснованих на описі особливостей обличчя, зокрема, метод Хаара, або використанні побудованих моделей обличчя - методи PCA, GMM та HMM, прийшли глибокі нейронні мережі, зокрема згорткові нейронні мережі (CNN) і рекурентні нейронні мережі (RNN), стали більш потужними та ефективними в розпізнаванні обличчя і аналізі емоцій. Це пов'язано з тим, що глибокі нейронні мережі можуть автоматично вивчати репрезентації даних на різних рівнях абстракції. Вони спроможні виявити складні закономірності та кореляції в даних без необхідності вручну створювати властивості, як це робить PCA, чи моделювати розподіли даних, як GMM. Також глибокі нейронні мережі мають багато прихованих шарів, що дозволяє їм вирішувати складні завдання, які потребують декількох рівнів абстракції[4]. Це особливо корисно для розпізнавання обличчя, оскільки обличчя мають багато складних особливостей.

Основною причиною зросту популярності глибокого навчання стало збільшення можливостей систем обробляти великі набори даних, а також можливість ефективно збирати необхідні набори для навчання в потрібній

кількості[4]. Це пов'язано з тим що нейронні мережі потребують більше даних для навчання, але, якщо доступно достатньо даних, вони можуть вести до кращих результатів у порівнянні з методами, зазначеними вище[4].

Головною перевагою глибокого навчання є можливість навчатися на вхідних даних та вихідних мітках, які позначають очікуванні результати, без потреби вручну корегувати ознаки або важелі, що значно спрощує процес навчання. Також глибокі нейронні мережі здатні до ефективного аналізу динамічних змін, в той час, як НММ може бути обмеженим у цьому відношенні.

Також цікавою тенденцією є використання генеративних змагальних мереж (GAN) для генерації реалістичних облич і досягнення високої якості генеративних результатів. Це корисно для створення даних для тренування та покращення розпізнавання облич. Також важливим аспектом стала обробка в реальному часі завдяки покращенню обчислювальної потужності та алгоритмів, що розширює їх можливості в застосуваннях безпеки та відео спостереження. Одним з найперспективніших трендів і водночас найменш дослідженим є використання для навчання нейронної мережі 3D моделей облич замість 2D зображень[5]. Основна ідея такого підходу полягає у тому, що такі моделі дають більше інформації про обличчя, що дає можливість мережі знайти більше факторів по яким вона може ідентифікувати особу. До того ж це має великі можливості для редагування, додавання емоцій, зміни освітлення, положення обличчя, додавання аксесуарів тощо[5]. Це зробить навчену мережу стійкою до названих факторів. Однак використання 3D моделей облич у навчанні мережі в порівнянні з 2D моделями має свої переваги і недоліки. Перевагами використання 3D моделей облич є:

- 3D моделі можуть зберігати більше інформації, так як вони містять глибокiсть та просторову інформацію. Це дозволяє нейронним мережам отримувати додаткові дані для розпізнавання облич та аналізу;

- 3D моделі можуть бути корисні для проведення навчання на низькій кількості наявних даних[5];

- 3D моделі можуть покращити точність розпізнавання, особливо в умовах низького освітлення або при наявності перешкод, які можуть впливати на зовнішні риси обличчя;

- 3D моделі можуть взаємодіяти з нейронними мережами, які враховують орієнтацію та рухи обличчя, що корисно у відстеженні обличчя в різних положеннях і під кутами[6].

Недоліки використання 3D моделей облич:

- отримання 3D моделей облич може бути більш складним процесом порівняно з 2D зображеннями. Це вимагає спеціалізованого обладнання, такого як 3D-сканери[6];

- обробка 3D даних вимагає більше обчислювальних ресурсів порівняно з 2D даними[6]. Це може бути проблемою для мобільних пристроїв та інтеграції в реальний час;

- на відміну від 2D даних, використання 3D моделей облич не так поширене, тому ускладнено пошук необхідної літератури, моделей та готових наборів для навчання.

#### 1.4 Постановка завдання

Метою кваліфікаційної роботи є розробка системи розпізнавання облич на основі нейронної мережі.

Об'єктом дослідження є процедури комп'ютерного розпізнавання облич.

Предметом дослідження є архітектури нейронних мереж для розпізнавання облич, математичні методи роботи нейронних мереж і регуляризації даних, показники ефективності розробленої мережі.

Для досягнення сформульованої мети визначені завдання дослідження:

- аналіз архітектури нейронної мережі для розв'язання задачі розпізнавання облич;

- аналіз існуючих методів регуляризації даних;

- математичне обґрунтування методів розпізнавання;
- аналіз можливостей підходу використання 3D моделей облич, для навчання нейронної мережі;
- розробка досліджуваної моделі з потрібною архітектурою;
- проведення навчання на 3D моделях і 2D зображеннях;
- дослідження ефективності навчених моделей;
- дослідження ефективності моделей з різними типами регуляризації;
- аналіз отриманих результатів.

## 2 ДОСЛІДЖЕННЯ АРХІТЕКТУР НЕЙРОННИХ МЕРЕЖ ДЛЯ РОЗПІЗНАВАННЯ ОБЛИЧ

В поточному розділі розглядаються різні архітектури нейронних мереж, призначених для розпізнавання облич, їх структуру, принципи роботи та методи навчання. Розуміння подібних архітектур дозволяє розробникам та дослідникам створювати системи розпізнавання облич з високою точністю та швидкістю, що відкриває безліч можливостей для подальшого розвитку та впровадження в різні сфери життя.

### 2.1 Базові архітектури

Для розпізнавання облич існують різні архітектури нейронних мереж, які можуть бути використані залежно від конкретних вимог та характеристик задачі. Основні архітектури, що використовуються для розпізнавання облич, включають згорткові мережі, рекурентні мережі та комбінації цих архітектур.

Згорткові мережі, CNN, є найпоширенішими архітектурами для обробки зображень та розпізнавання облич. Вони ефективно використовують згорткові шари для виявлення локальних характеристик зображення, таких як границі, текстури та форми, а також пулінгові шари для зменшення розмірності даних. Ці мережі можуть бути глибокими, що дозволяє їм автоматично вивчати складні залежності між різними рисами обличчя. Згорткові мережі показали вражаючі результати в задачах розпізнавання облич і стали стандартом для багатьох систем розпізнавання облич[7].

Принцип роботи згорткових нейромереж базується на використанні згорткових шарів (convolutional layers) та пулінгових шарів (pooling layers), які дозволяють ефективно витягати різні ознаки з вхідних даних (рис. 2.1). Основна ідея полягає в тому, що згорткові шари виконують локальні згортки між вхідними даними і фільтрами, а пулінгові шари зменшують розмір

отриманого представлення.

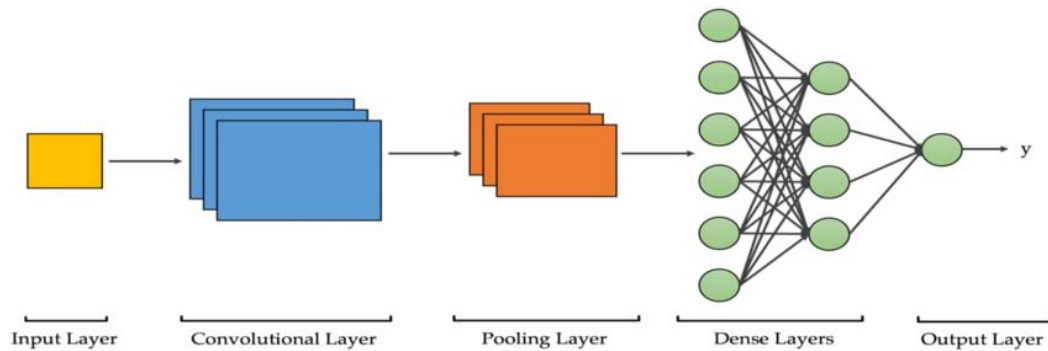


Рисунок 2.1 - Схема архітектури згорткової мережі

Рекурентна мережа (RNN) є іншою архітектурою, яка може бути використана для розпізнавання облич. Мережі RNN добре підходять для моделювання послідовностей даних, таких як послідовності пікселів зображення або послідовності фреймів відео[7]. Рекурентні шари у таких мережах здатні зберігати інформацію про попередні стани, що дозволяє враховувати контекст та динаміку обличчя. Це особливо важливо для задач, які вимагають аналізу динамічних змін, таких як визначення жестів або емоцій на обличчі.

Крім того, існують комбіновані архітектури, які поєднують елементи згорткових та рекурентних мереж[7]. Наприклад, згорткові рекурентні нейронні мережі (CRNN) поєднують переваги обох типів мереж, дозволяючи моделювати як локальні характеристики зображень, так і послідовну динаміку.

Основна відмінність від CNN є наявність рекурентного шару. У CRNN (рис.2.2) цей шар може бути реалізований з використанням рекурентних клітин, таких як Long Short-Term Memory (LSTM) або Gated Recurrent Unit (GRU), які можуть зберігати і передавати контекстну інформацію через певний час.

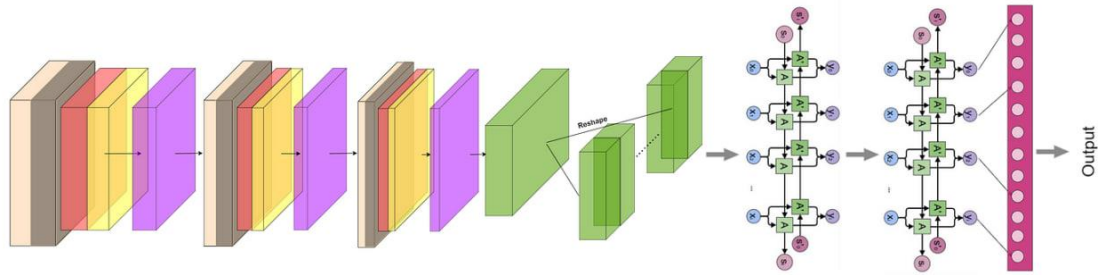


Рисунок 2.2 - Схема архітектури мережі CRNN

Архітектура CNN широко використовується для розпізнавання облич, і це має багато плюсів, які роблять її вигідною для використання в дослідженні через її простоту та зручність налаштування, основні переваги наведені нижче:

- простота архітектури: CNNs мають простий та легко зрозумілий дизайн. Вони складаються з послідовних шарів згортки та пулінгу. Це дозволяє створити надійну модель і досліджувати її на різних завданнях розпізнавання облич;

- легкість налаштування гіперпараметрів: конфігурація CNN має невелику кількість гіперпараметрів, що легше налаштовувати порівняно з більш складними архітектурами. Це нівелює похибку в дослідженні пов'язану зі складністю налаштування самої мережі;

- доступність попередньо навчених моделей: на ринку існують попередньо навчені моделі мереж згортки, такі як VGG, ResNet та MobileNet[7], які можуть служити як орієнтир для дослідження;

- підходять як для великих наборів даних, так і для маленьких: CNNs показують добрі результати при навчанні на обох наборах даних, що дозволяє провести дослідження на незначній вибірці і загальні висновки будуть актуальними для великого набору даних;

- зручність у візуалізації: CNN легко інтерпретуються, дає змогу наочно продемонструвати ефективність моделей.

## 2.2 Модернізація CNN

Модернізація звичайних згорткових нейронних мереж CNN може включати в себе різні підходи та техніки для поліпшення їх продуктивності та ефективності. Основним методом модифікації є зміна архітектури CNN для роботи з більш специфічними даними або завданнями, це такі мережі як VGG, ResNet, Inception, MobileNet і багато інших. Для роботи з тривимірними даними існує модифікована версія CNN, відома як 3D-CNN[8]. 3D-CNN є модернізацією звичайних CNN або як їх іноді називають 2D-CNN, яка розширює можливості згорткових нейронних мереж для обробки тривимірних даних, таких як відео, медичні зображення та великі об'єми даних в тривимірних форматах[8]. Основні елементи такі як методи регуляризації, нормалізації, оптимізації, функції активації, механізм backpropagation залишаються, наявність слоїв згортки та пулінгу залишаються незмінними. Основні відмінності між 3D-CNN і 2D-CNN полягають у використанні тривимірних даних і фільтрів в 3D-CNN, а також в специфічній конфігурації їхніх архітектур[8], а саме:

- у 2D-CNN використовуються двовимірні фільтри з двома вимірами (висота і ширина), тоді як у 3D-CNN використовуються тривимірні фільтри з трьома вимірами (висота, ширина і глибина). Тривимірні фільтри дозволяють моделі аналізувати просторові залежності в тривимірних даних;

- 3D-CNN включають тривимірні згорткові шари, тривимірні пулінгові шари та тривимірні повністю з'єднані шари для обробки тривимірних даних. У 2D-CNN використовуються двовимірні шари відповідно до вимірів даних;

- тривимірні дані мають третій вимір, який представляє глибину об'єктів. У 3D-CNN цей глибокий вимір враховується при використанні тривимірних фільтрів та згорткових операцій;

- у 2D-CNN вхідні дані подаються у вигляді двовимірного зображення, тоді як у 3D-CNN вхідні дані мають тривимірну форму, яка включає висоту, ширину і глибину або час.

Для проведення дослідження ефективності використання 3D даних для навчання нейронної мережі, було обрано модель архітектури мережі 3D-CNN, це обумовлюється однаковими основними принципами роботи з звичайними 2D-CNN і наявністю потрібних механізмів для обробки 3D даних, що дозволило провести об'єктивне порівняння ефективності застосування 3D і 2D даних для навчання.

### 2.3 Підготовка даних

Збір даних для навчання мережі розпізнавання облич - це важливий етап в створенні ефективної системи. Зазвичай збирають фотографії або відео, які включають в себе обличчя, при тому важливо отримати різноманітні дані, щоб навчена модель мала можливість розпізнавати різні типи облич за різних обставин. Але такий підхід має ряд значних недоліків:

- втрата просторової інформації: 2D зображення не передають тривимірну інформацію про об'єкти. Це може призвести до втрати деякої глибини та просторової розмірності;

- повороти та пересування: 2D зображення вразливі до змін масштабу, поворотів та пересувань об'єктів на зображеннях. Моделі, навчені на 2D даних, можуть бути менш стійкими до таких змін;

- недостатня інформація про текстури: для деяких завдань, таких як реалістична апроксимація фізичних матеріалів, наприклад, шкіра, метал, дерево, 2D зображення не містять достатньо інформації про текстури та блики;

- варіативність освітлення: зміни в освітленні на 2D зображеннях можуть значно впливати на вигляд об'єктів і робити навчання менш стійким до варіацій освітлення;

- неоднорідність даних: збір даних у великих масштабах може бути важким завданням через неоднорідність осіб, освітлення, позицій та засобів фотографування;

- обмежені можливості для аналізу просторової глибини: мережі, навчені на 2D зображеннях, мають обмежені можливості для аналізу просторової глибини та об'єктів, що перекриваються;

- підвищена чутливість до шуму: 2D зображення можуть бути схильними до впливу шуму та артефактів, що впливає на якість розпізнавання.

Використання 3D моделей замість 2D зображень може вирішити багато з перерахованих проблем і виявитись ефективнішими. Для проведення дослідження необхідно перетворити 2D і 3D зображення у набір даних для навчання мережі.

Перетворення 2D зображення у вхідні дані для навчання відбувається за рахунок перетворення зображення у вектор або матрицю чисел, які відповідають пікселям зображення у значенні RGB. Зображення у форматі RGB складається з пікселів, кожен з яких має значення для червоного (R), зеленого (G) і синього (B) каналів, що визначають колір даного пікселя. Всі ці значення можуть бути числами від 0 до 255 або 0 до 1 у випадку нормалізації, де 0 означає відсутність кольору (чорний), а 255 або 1 - максимально можливий насичений колір. Для того щоб перетворити кожен піксель у 2D зображенні в числове значення у форматі RGB (червоний, зелений, синій) потрібно пройти по кожному пікселю зображення і отримати значення червоного (R), зеленого (G) і синього (B) каналів. Ці значення зберігаються у трійку векторів однакової розмірності, які відповідають кольоровим каналам. Ці вектори і будуть подані на вхід нейронній мережі для проведення навчання.

Перетворення 3D моделі у формат, зручний для роботи з нейронними мережами відбувається інакше ніж аналогічний процес з 2D зображеннями. Поки не існує універсального підходу перетворення 3D об'єктів у зручний формат даних для роботи з нейронною мережею, натомість використовуються різні процеси, такі як отримання 2D зображень з різних кутів або використання характеристик облич[9]. Відомо про наступні

способи перетворення 3D моделей обличчя у набір даних:

а) перетворення кожної 3D моделі обличчя у вектор ознак. Цей вектор може містити інформацію про форму обличчя, текстури, особливі риси, тощо[9]. Використання глибоких нейронних мереж, які приймають вектори ознак, може бути ефективним підходом;

б) виготовлення 2D проєкцій з 3D моделі обличчя під різними кутами або в різних проєкціях. Даний метод не підходить для дослідження;

в) використання методів глибокого навчання на графах для обробки 3D моделей. Такі підходи, як Graph Convolutional Networks (GCNs), дозволяють працювати з графами, які можуть представляти структуру обличчя і зв'язки між його точками;

г) перетворення 3D моделей у хмари точок (point clouds) і використання точок як вхідних даних для нейронної мережі[9]. Можна використовувати алгоритми, які визначають найважливіші точки на обличчі;

д) виділення локальних особливостей обличчя, наприклад, очі, ніс, рот, та використання їх як окремих характеристик. Такий підхід може допомогти мережі краще розпізнавати важливі частини обличчя.

Для деяких завдань, особливо в областях де важко збирати тривимірні дані, використання 2D зображень є необхідним компромісом. Однак у деяких випадках може бути корисним застосовувати тривимірні дані або комбінувати 2D і 3D інформацію для покращення ефективності й стабільності мережі.

В дослідженні використовується технологія перетворення 3D моделей у хмари точок (point clouds), бо ця технологія має гарну сумісність з 3D-CNN, а також вона має одну з найкращих можливостей зберігати необхідні особливості обличчя без втрат ознак. Крім того, за принципом роботи метод хмар точок найбільш близький до алгоритму, який використовується в обробці 2D даних, що має важливе значення для об'єктивного порівняння ефективності моделей нейронних мереж з використанням 3D і 2D даних для навчання.

## 2.4 Гіперпараметри та функції активації

### 2.4.1 Гіперпараметри

Гіперпараметри в нейронних мережах - це параметри, які не вивчаються самостійно під час навчання моделі, але вони визначають, як відбувається саме навчання. Гіперпараметри керують архітектурою моделі, процесом навчання та регулюють якість та швидкість навчання. Нижче розглянуто деякі з найважливіших гіперпараметрів і їх функції.

1. Розмір пакета (batch size): представляє собою кількість прикладів, які обробляються одночасно під час одного кроку навчання. Великий розмір пакета може прискорити навчання, але вимагає більше пам'яті. Малий розмір пакета може зробити навчання більш стійким[10].

2. Кількість шарів та нейронів у кожному шарі (архітектура моделі): визначає, які функції модель може навчати та її загальну складність[10]. Додавання більше шарів або нейронів може покращити продуктивність, але може також призвести до перенавчання.

3. Швидкість навчання (learning rate): є кроком, з яким ваги моделі оновлюються після кожного пакета даних під час навчання[10]. Правильний вибір швидкості навчання важливий для досягнення швидкого та стабільного навчання.

4. Кількість епох навчання: це кількість разів, коли модель проходить через весь набір даних. Повний прохід називається епохою[10]. Вибір оптимальної кількості епох допомагає уникнути недонавчання або перенавчання.

5. Регуляризація: гіперпараметри, пов'язані з регуляризацією, такі як коефіцієнт регуляризації L1 або L2, допомагають контролювати перенавчання шляхом штрафування великих значень ваг.

6. Функція активації: вибір функцій активації впливає на навчання та рішення моделі.

7. Оптимізатор: вибір алгоритму оптимізації, такого як стохастичний градієнтний спуск (SGD), Adam, RMSprop, впливає на швидкість та якість навчання.

Гіперпараметри налаштовуються під час підготовки моделі до навчання, часто на підставі експериментів або власного досвіду[10]. Оптимальні значення гіперпараметрів можуть залежати від конкретної задачі, ресурсної бази та набору даних, і їх пошук може бути завданням оптимізації гіперпараметрів.

#### 2.4.2 Функції активації

Функції активації є ключовими компонентами нейронних мереж і мають великий вплив на роботу систем розпізнавання облич. Вони відповідають за нелінійність та прийняття рішень в мережі, допомагаючи їй наблизитися до складних залежностей у даних [11]. На відміну від інших гіперпараметрів, які підбираються з досвіду або емпіричним шляхом, функції активації мають більш конкретні випадки коли та чи інша функція працює ефективніше.

Сигмоїдна функція (Sigmoid) (рис. 2.3) приймає значення у діапазоні від 0 до 1. Вона дозволяє моделі виробляти ймовірності та використовується часто в бінарному класифікації облич. Однак, вона має проблему втрати градієнту для значень, що знаходяться близько до 0 або 1. Втрата градієнту у сигмоїдальній функції впливає на процес навчання нейронних мереж, особливо у глибоких архітектурах, і це явище відоме як проблема зниклих градієнтів (vanishing gradients problem). Для подолання цієї проблеми, іноді використовують інші функції активації, такі як ReLU або Leaky ReLU, які не мають такої схильності до зникання градієнтів[11]. Тому сигмоїдна функція використовується переважно в останньому шарі мережі для бінарної класифікації, де потрібно вирішити, чи належить вхідний приклад до одного з двох класів, наприклад ідентифікація чи є на зображенні обличчя чи ні.

Використовується також у шарах, де потрібна обмежена вихідна активація.

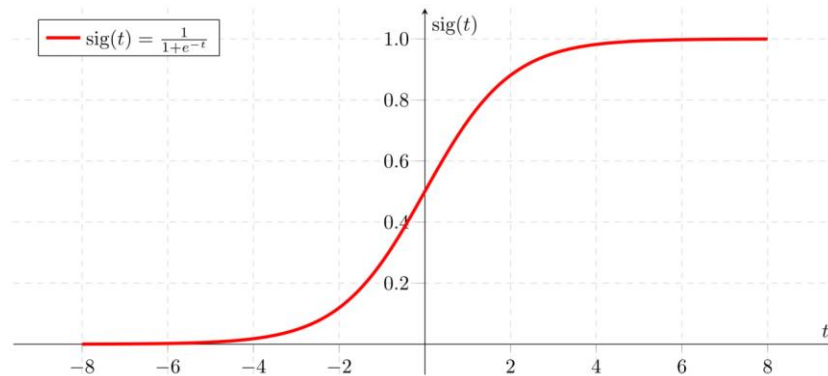


Рисунок 2.3 - Графік сигмоїдної функції

Функція ReLU (Rectified Linear Unit) - нелінійна функція, яка повертає 0 для від'ємних значень і саме значення для невід'ємних значень (рис. 2.4). ReLU допомагає уникнути проблеми втрати градієнту та є швидшою у обчисленнях[11]. Вона широко використовується в згорткових нейронних мережах для розпізнавання облич за рахунок своєї ефективності. Переваги: пришвидшення обчислення, відсутність проблем з градієнтом при позитивних значеннях. Функція ReLU застосовується для швидкого навчання мережі та прискорення збіжності і використовується у прихованих шарах.

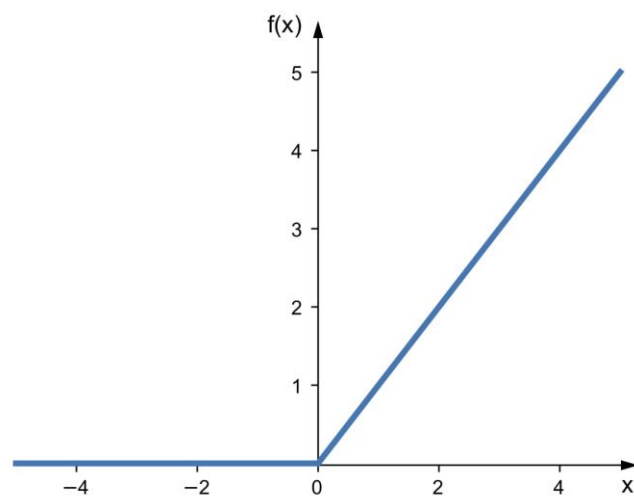


Рисунок 2.4 - Графік функції ReLU

Функція Leaky ReLU (рис.2.5) - модифікація функції ReLU, вирішує її проблему "мертвих нейронів" (dead neurons), коли значення нейрону стає нулем[11]. Краща версія функції Leaky ReLU дозволяє невеликий ненульовий нахил від'ємних значень, зберігаючи простоту та швидкість обчислень ReLU.

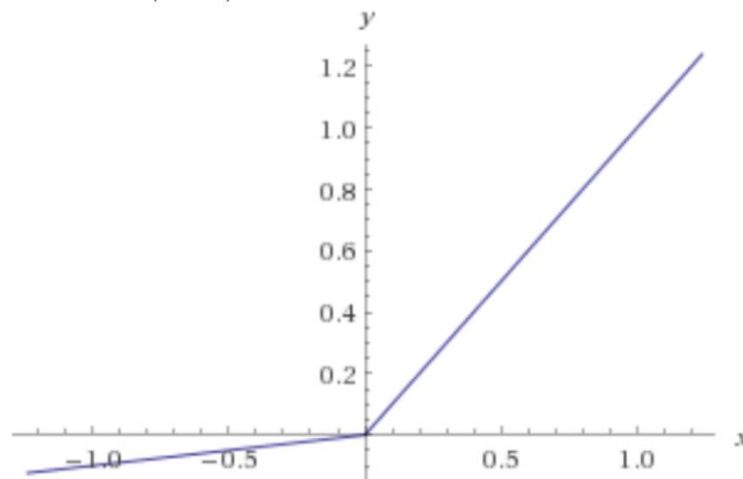


Рисунок 2.5 - Графік функції Leaky ReLU

Функція softmax - використовується у вихідному шарі мережі для отримання ймовірностей класів облич (рис.2.6). Вона перетворює вихідні значення в діапазоні від 0 до 1 таким чином, що сума всіх ймовірностей дорівнює 1. Softmax допомагає вирішити задачу класифікації облич. Функція Softmax використовується в розпізнаванні облич у контексті багатокласової класифікації, коли потрібно визначити, до якого класу належить обличчя серед різних можливих класів або осіб. Важливо відзначити, що сама функція Softmax не визначає обличчя на фотографії, але вона допомагає визначити ймовірність того, що обличчя належить до кожного можливого класу або особи. Функція Softmax гарантує, що ймовірності для всіх класів осіб складаються до 1, що робить її ідеальною для задач багатокласової класифікації, включаючи розпізнавання облич[11].

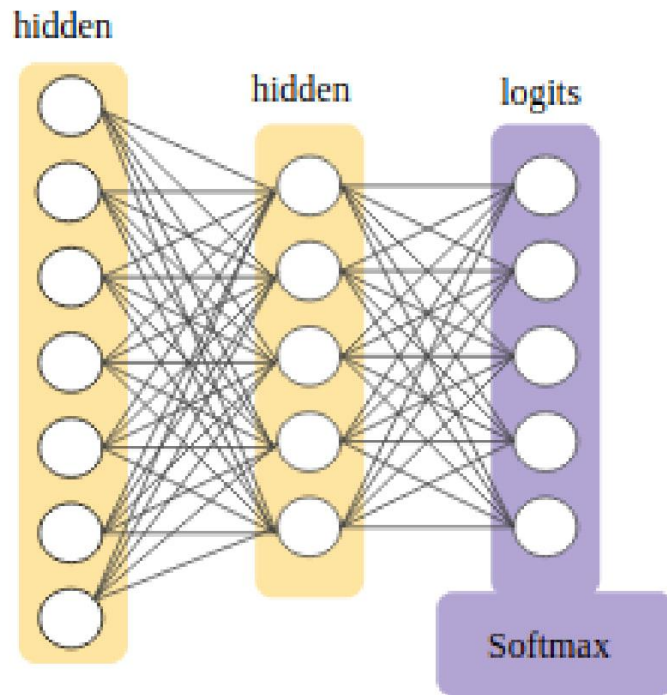


Рисунок 2.6 - Принцип роботи функції softmax

Дослідивши наявну літературу і рекомендації стає однозначним використання функції softmax у останньому шарі мережі і функції ReLU для всіх інших слоїв при розробці моделі нейронної мережі для розпізнавання облич. При виникненні проблем з розробленою моделлю доцільним є заміна в деяких слоях ReLU на Sigmoid функцію для зменшення діапазонів.

## 2.5 Методи регуляризації і нормалізації

Регуляризація і нормалізація - це дві різні техніки, які використовуються в нейронних мережах для покращення навчальної здатності та загальної продуктивності. Регуляризація - це процес додавання додаткових обмежень до моделі мережі під час навчання з метою уникнення перенавчання. Перенавчання - це ситуація, коли модель надто добре адаптується до навчальних даних, але не може узагальнити свої знання на нові дані. Перенавчання є однією з ключових проблем в системах розпізнавання облич при навчанні моделі, це виявляється при аналізі

отриманих результатів коли модель показує високу точність на навчальному наборі даних, але показує низьку точність на нових, раніше не бачених даних (рис. 2.7).

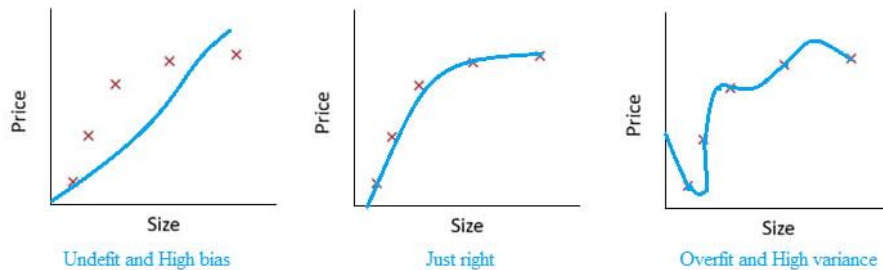


Рисунок 2.7 - Демонстрація ненавченої, навченої і перенавченої мережі

Регуляризація допомагає зменшити ризик перенавчання і недонавчання[12]. Два основних типи регуляризації в нейронних мережах:

а) L1 регуляризація включає додавання до функції втрати моделі штрафу, що базується на абсолютних значеннях ваг моделі. Це означає, що модель намагатиметься зменшити абсолютні значення ваг, що призводить до розріджених ваг, де деякі ваги стають точно нульовими. L1 регуляризація допомагає зменшити кількість ознак у моделі, обираючи лише найважливіші ознаки для розв'язання завдання, і тим самим сприяє видаленню зайвого шуму в даних[12].

б) L2 регуляризація включає додавання до функції втрати моделі штрафу, що базується на квадратичних значеннях ваг. Це призводить до того, що всі ваги у моделі стають невеликими, але не точно нульовими. L2 регуляризація сприяє більш рівномірному розподілу значень ваг[12]. Це допомагає покращити здатність моделі працювати на нових даних.

Нормалізація - це процес стандартизації вхідних даних та ваг моделі для полегшення і прискорення навчання мережі. Дана техніка допомагає моделі швидше збігатися та забезпечує більш стійке та ефективне навчання.

При стандартизації здійснюється процес перетворення даних таким

чином, щоб середнє значення становило нуль, а стандартне відхилення було одиницею. Ваги моделі також можуть бути стандартизовані.

Міні-макс нормалізація: в цьому варіанті дані масштабуються таким чином, щоб значення знаходилися в певному діапазоні, наприклад, від 0 до 1.

Нормалізація батча (Batch Normalization) - це спеціальний тип нормалізації, який використовується для мереж глибокого навчання. Він нормалізує активації шарів моделі для зменшення мінливості та прискорення навчання.

Загалом, нормалізація спрощує оптимізацію моделі розпізання облич і зменшує ймовірність виникнення проблем під час навчання, тоді як регуляризація допомагає контролювати перенавчання та недонавченність з метою поліпшення загальної ефективності моделі. Обидві техніки можуть використовуватися разом для досягнення кращих результатів у завданнях машинного навчання.

В дослідженні було встановлено, які з названих методів регуляризації найбільш ефективні у поєднанні з навчанням на 3D моделях.

## 3 ДОСЛІДЖЕННЯ МАТЕМАТИЧНОЇ БАЗИ

В даному розділі проводиться дослідження математичної бази нейронних мереж, що є важливою частиною для розуміння функціонування та вдосконалення їх дії. Нейронні мережі - це складні системи, які моделюються за допомогою математичних понять та операцій.

### 3.1 Нейрон

Основною одиницею мережі є нейрон. З математичної точки зору, нейрон можна представити як функцію, що обробляє вхідні сигнали та генерує вихід. Зазвичай, використовуються лінійні комбінації вхідних сигналів, за якими слідує нелінійне перетворення, що називається активаційною функцією.

Математичне представлення нейрона в мережі можна виразити за допомогою моделі, відомої як штучна нейронна мережа або перцептрон. Основні компоненти математичного представлення нейрона включають ваги, вхідні дані, функцію активації та вихідну активацію[13].

Вхідні дані, вектор  $x$ : нейрон приймає вхідні дані у вигляді вектора  $x$ , де кожен елемент вектора відповідає певному входу або функції, яку необхідно аналізувати.

Ваги, вектор  $w$ : кожен вхідний сигнал помножується на вагу, яка відповідає важливості цього сигналу для нейрона. Ваги позначаються як вектор  $w$ .

Зважена сума  $a$  обчислюється за формулою, яка є скалярним добутком вектора ваг  $w$  і вектора входів  $x$ [13]:

$$a = \sum_0^n (w_i * x_i), \quad (3.1)$$

де  $n$  - кількість входів,  $w_i$  - вага  $i$ -го входу,  $x_i$  - значення  $i$ -го входу.

Функція активації  $f(a)$  приймає зважену суму  $a$  як вхід і визначає вихід нейрона. Різні функції активації використовуються в залежності від завдання, але однією з найпоширеніших є сигмоїдна функція або функція ReLU, що визначаються за формулами відповідно, які мають вигляд:

$$f(a) = 1 / (1 + e^{-a}), \quad (3.2)$$

$$f(a) = \max(0, a). \quad (3.3)$$

Вихід нейрона - це значення функції активації, який використовується для подальшого передавання інформації до інших нейронів у мережі.

### 3.2 Функція втрат

Функція втрат (loss function) грає критичну роль в навчанні нейромережі, оцінюючи різницю між прогнозами моделі та справжніми значеннями вхідних даних. Основна мета застосування даної функції полягає в тому, щоб надавати моделі об'єктивну оцінку того, наскільки точно вона виконує своє завдання. Чим менше значення функції втрат, тим краще модель адаптується до навчальних даних. Функція втрат може визначатися на різних математичних рівнях, залежно від задачі навчання і типу моделі. Далі розглянуто основні види функцій втрат, які широко використовуються у проектування нейронних мереж.

Середньоквадратична помилка (Mean Squared Error, MSE) використовується для задач регресії та є простою для розрахунку та реалізації, добре працює для задач регресії, де цільові значення знаходяться в неперервному числовому діапазоні. Середньоквадратична помилка вразлива до викидів, оскільки квадратична залежність великої помилки робить її важливою для моделі, що може призвести до неправильного навчання; не

нормалізована за зміною масштабу, тобто величина функції втрат залежить від одиниць виміру цільових значень[13]. Середньоквадратична помилка визначається за формулою:

$$MSE = 1/n * \sum (y_i - \bar{y})^2, \quad (3.4)$$

де  $y_i$  - справжні значення,  $\bar{y}$  - прогнози моделі,  $n$  - кількість прикладів у навчальному наборі.

Перехресна ентропія (Cross-Entropy) використовується для задач класифікації та ефективна для задач класифікації та навчання нейромереж з вчителем; прискорює збільшення точності передбачень нейромережі, оскільки вибуху градієнта (exploding gradients) у багатьох випадках не відбувається. Вибух градієнту це проблема, з якою стикаються під час навчання глибоких нейронних мереж, особливо в контексті методів оптимізації на основі градієнта, таких як зворотне поширення. Ця проблема виникає, коли градієнти втрат мережі щодо параметрів (ваги або зсувів) стають надмірно великими, що призводить до нестабільності та нездатності мережі сходитися до відповідного рішення. Основними недоліками перехресної ентропії є неможливість її використання для задач регресії; вразливість до дисбалансу класів у вхідних даних, що може призвести до незрівноваженої моделі[13]. Математичний вираз для визначення бінарної класифікації має вигляд:

$$\text{CrossEntropy} = -\sum y_i * \log(p_i) + (1 - y_i) * \log(1 - p_i), \quad (3.5)$$

де  $y_i$  - справжні значення (0 або 1),  $p_i$  - прогнози моделі.

Багатокласова класифікація визначається за формулою:

$$\text{CrossEntropy} = -\sum \sum (y_{ij} * \log(p_{ij})), \quad (3.6)$$

де  $Y_{ij}$  - індикаторна функція, яка дорівнює 1, коли об'єкт належить класу  $j$ , і в іншому випадку, а  $P_{ij}$  - ймовірність, що модель призначає об'єкту  $i$  клас  $j$ .

Функція втрати Huber використовується для регресії та менш чутлива до викидів (outliers) і має нормалізовану функцію втрат, вимагає налаштування параметра  $\delta$ , який визначає поріг для розрізнення між квадратичною та лінійною залежністю помилки[13], може потребувати більше часу для обчислень порівняно з квадратичною функцією втрат:

$$L_{\delta}(a) = \begin{cases} \frac{1}{2}a^2 & , |a| \leq \delta \\ \delta \left( |a| - \frac{1}{2} \delta \right) & , \text{інакше} \end{cases} \quad (3.7)$$

де  $a$  - різниця між прогнозами та справжніми значеннями, а  $\delta$  - певний поріг.

Для проведення дослідження та розробки нейронних мереж для розпізнавання облич була обрана функція витрат Cross-Entropy через те, що вона ефективна для задач класифікації і має найменші витрати при обчисленнях, що дозволить провести більш досконале дослідження інших параметрів.

### 3.3 Ваги між нейронами і механізм їх корегування

Зв'язки між нейронами визначаються вагами. Кожен зв'язок має свій ваговий коефіцієнт, який впливає на сигнал, що передається від одного нейрона до іншого. Ваги налаштовуються під час навчання нейромережі для досягнення потрібної функціональності. Один з популярних методів навчання - це зворотний розповсюджений алгоритм помилки (Backpropagation), який використовує градієнт спаду для зміни ваг зв'язків таким чином, щоб мінімізувати помилку мережі[14].

Для математичного аналізу принципу роботи backpropagation

розглянемо нейронну мережу з одним прихованим шаром. Ця мережа має вхідні дані, прихований шар з деякою кількістю нейронів та вихідний шар. Розглядається задача бінарної класифікації, де нейронна мережа намагається визначити, до якого класу належить вхідний приклад.

Feedforward (прямий прохід): вхідні дані подаються на вхідному шарі. Для кожного нейрона в прихованому шарі обчислюється зважена сума входів, додається зсув, і потім застосовується активаційна функція. Дана активаційна функція позначена як  $\sigma$ . Вихід прихованого шару слугує входом для нейронів вихідного шару[14]. Для кожного нейрона в вихідному шарі обчислюється вагова сума і знову застосовується активаційна функція для отримання фінального вихідного сигналу.

Обчислення помилки (функція втрат) використовується функція втрат для оцінки того, наскільки сильно мережа помиляється у визначенні класу для даного вхідного прикладу. Згідно попереднього розділу обрано середньоквадратичну помилку.

Backpropagation (зворотне поширення) обчислюються градієнти функції втрат відносно ваг і зсувів у мережі. Для цього використовується правило ланцюжка (chain rule) диференціювання. Градієнти обчислюються, починаючи з вихідного шару і переміщаючись назад до прихованого шару і вхідного шару[14]. Оновлення ваг і зсувів в мережі виконуються з використанням градієнтного спуску, де змінюють ваги і зсуви в напрямку, протилежному градієнту функції втрат з певною швидкістю навчання (learning rate).

Повторення процесу: кроки 1-3 повторюються для багатьох вхідних прикладів протягом багатьох епох навчання, доки мережа не досягне прийняттого рівня точності.

Цей процес допомагає нейронній мережі адаптуватися до навчальних даних і встановити оптимальні ваги і параметри для вирішення задачі. Backpropagation включає в себе обчислення градієнтів (похідних) функції втрат відносно ваг і зсувів усіх шарів нейронної мережі[14].

Обчислення градієнтів (похідних) функції втрат відбувається за формулами, наведеними нижче.

Градiєнти необхідно обчислити для ваг ( $w$ ) і для зміщення ( $b$ ).

Градiєнт для ваг вихідного шару обчислюється за допомогою ланцюжкового правила:

$$\frac{\partial C}{\partial w_{jk}^l} = \frac{\partial C}{\partial h_j^l} * \frac{\partial h_j^l}{\partial w_{jk}^l}, \quad (3.8)$$

де  $C$  - функція втрат,  $w$  - значення ваг,  $b$  - значення зміщення,  $h$  - значення нейрона,  $a$  - значення функції активації,  $l$  - кількість слоїв,  $j$  - індекс вхідного нейрона,  $k$  - індекс вихідного нейрона.

Значення нейрона розраховується за формулою:

$$h_j^l = \sum_{k=1}^m w_{jk}^l a_k^{l-1} + b_j^l, \quad (3.9)$$

де  $m$  – кількість нейронів у  $l-1$  слою.

Далі за допомогою диференціювання знаходиться значення градієнта нейрона по відношенню до значення ваг:

$$\frac{\partial h_j^l}{\partial w_{jk}^l} = a_k^{l-1}. \quad (3.10)$$

Фінальне значення градієнту функції втрат відносно ваг вихідного шару обчислюється:

$$\frac{\partial C}{\partial w_{jk}^l} = \frac{\partial C}{\partial h_j^l} a_k^{l-1}. \quad (3.11)$$

Гradient для зсувів вихідного шару може бути обчислений наступним чином:

$$\frac{\partial C}{\partial b_j^1} = \frac{\partial C}{\partial h_j^1} * \frac{\partial h_j^1}{\partial b_j^1}. \quad (3.12)$$

Значення нейрона розраховується за формулою 3.9.

Далі за допомогою диференціювання знаходиться значення градієнта нейрона по відношенню до значення зміщення:

$$\frac{\partial h_j^1}{\partial b_j^1} = 1. \quad (3.13)$$

Фінальне значення градієнту функції втрат відносно зміщення вихідного шару обчислюється:

$$\frac{\partial C}{\partial b_j^1} = \frac{\partial C}{\partial h_j^1} 1. \quad (3.14)$$

Гradientи ваг і зміщення далі використовуються для оновлення ваг(формула 3.15) і зсувів(формула 3.16) у мережі за допомогою градієнтного спуску.

Формула для оновлення ваг і зсувів нейронів в під час навчання за допомогою градієнтного спуску має вигляд:

$$W_{jk}^{new} = W_{jk}^{old} - \mu * \frac{\partial C}{\partial w_{jk}^1}, \quad (3.15)$$

$$b_{jk}^{new} = b_{jk}^{old} - \mu * \frac{\partial C}{\partial b_j^1}, \quad (3.16)$$

де  $\mu$  - швидкість навчання.

### 3.4 Регуляризація

L1 та L2 регуляризація - це методи, які використовуються в машинному навчанні для обмеження моделі. Ці методи додають додаткові терміни до функції втрати під час навчання, які впливають на ваги моделі.

L1 регуляризація (регуляризація Лассо) додає абсолютні значення ваг (L1-норма) до функції втрат, що призводить до розріджування ваг, що в свою чергу може призвести до відбору ознак (feature selection)[15].

Функція втрат при L1 регуляризації визначається за формулою:

$$J(w) = J_0(w) + \gamma \sum |w_i|, \quad (3.17)$$

де  $J(w)$  - нова функція втрати з L1 регуляризацією,  $J_0(w)$  - оригінальна функція втрати без регуляризації,  $\gamma$  - гіперпараметр регуляризації, який контролює силу регуляризації,  $w_i$  - вага (коефіцієнт) для  $i$ -тої ознаки.

L2 регуляризація (регуляризація Рідж) додає квадрати ваг (L2-норма) до функції втрат, що призводить до того, що всі ваги зменшуються пропорційно, але жодна з них не стає точно нульовою, тому вона не проводить відбір ознак.

Функція втрат при L2 регуляризації визначається за формулою:

$$J(w) = J_0(w) + \gamma \sum w_i^2, \quad (3.18)$$

де  $J(w)$  - нова функція втрати з L2 регуляризацією,  $J_0(w)$  - оригінальна функція втрати без регуляризації,  $\gamma$  - гіперпараметр регуляризації, який контролює силу регуляризації,  $w_i$  - вага (коефіцієнт) для  $i$ -тої ознаки.

Обидві ці методи регуляризації допомагають зменшити перенавчання або недонавченість моделі та підвищити її загальну ефективність, але вони

впливають на ваги моделі по-різному.

В приведеному дослідженні використані алгоритми L1 і L2 регуляризації і досліджена їх ефективність разом з запропонованим підходом навчання при розробці моделі нейронної мережі розпізнавання облич.

## 4. РЕАЛІЗАЦІЯ ДОСЛІДЖУВАНОЇ МОДЕЛІ ТА ПРОВЕДЕННЯ ДОСЛІДЖЕННЯ

В цьому розділі розглянуто поетапне проектування моделей для проведення дослідження кваліфікаційної роботи, етапи проведення дослідження та проводиться аналіз отриманих результатів. Для реалізації було обрано мову програмування Python з використанням відкритого програмного забезпечення для машинного навчання і глибокого навчання, розробленого компанією Google - TensorFlow, а також бібліотеки Keras. Для того щоб на результати дослідження не мали впливу на індивідуальні характеристики комп'ютера, тренування і тестування моделей відбувається в хмарному сервісі Google Colab, який надає доступ до обчислювальних ресурсів та інструментів для машинного навчання та наукових обчислень.

### 4.1 Архітектура моделі

Для досягнення поставленої мети дослідження було спроектовано 2 нейронні мережі для роботи з 3D і 2D даними. Для того, щоб дослідження було об'єктивним спроектовані моделі мають однакові технічні характеристики за винятком специфіки роботи з різними типами вхідних даних. При розробці моделей нейронних мереж для розпізнавання обличчя, було обрано дві архітектури мереж, а саме 3D-CNN і 2D-CNN, бо вони мають однаковий базовий принцип роботи, і відрізняються лише в адаптуванні до відповідної розмірності даних для навчання.

Спроектовані моделі мають архітектуру стандартної згорткової мережі, з 7-ма шарами згортки, з функцією активації relu, третій та шостий шари мають механізм padding. Також присутні 3 пулінгові шари. Для ідентифікації кластерів в кінці підключається функція активації softmax. Повна інформація про архітектуру і гіперпараметри наведена в таблиці 4.1, також в лістингу 4.1

наведена реалізація архітектури для 2D-CNN без підключених методів регуляризації, аналогічна реалізація для 3D-CNN і підключення L1 і L2 регуляризацій наведені в додатку Б.

Таблиця 4.1 - Структура спроектованих мереж і їх гіперпараметри

Гіперпараметр	Значення	За що відповідає
Кількість слоїв згортки	7	Слої згортки відповідають за виділення різних ознак та шаблонів у вхідних даних
Кількість слоїв пулінгу	3	Слої пулінгу зменшують розмір зображення та кількість параметрів
Повнозв'язні слої	3	Відповідають за комбінування фільтрів та ознак
Розмір пулінгу	2:2	Визначає розмір областей, які об'єднуються під час пулінгу
Основна функція активації	relu	Визначає, як нейрони реагують на вхідні сигнали
Активация останнього слою	softmax	Визначає, як модель генерує вихідні значення
Швидкість навчання	0.002	Визначає швидкість з якою модель оновлює ваги під час процесу навчання
Кількість фільтрів	64-128-256	Вказує, скільки фільтрів використовується в шарі згортки
Функція втрат	Cross-entropy	Визначає, як вимірюється різниця між прогнозованими та фактичними значеннями
Ядро згортки	3(1,2,4,5,7 слої) і 5(3,6 слої)	Визначає, які шаблони або фільтри використовуються для згортки вхідних даних
Значення L1-L2	0.01	Визначають силу регуляризації моделі

регуляризації		
Оптимізатор	Adam	Визначає алгоритм, який використовується для оптимізації оновлення ваг моделі
Макс. кількість нейронів	256	Визначає максимальну кількість нейронів, яку може мати повнозв'язний шар або слой

#### Лістинг 4.1 - Реалізація архітектури 2D-CNN без методів регуляризації

```

model = Sequential(name = model_name)

model.add(Conv2D(64, kernel_size=3, activation='relu',
input_shape=(100, 100, 1)))
model.add(Conv2D(64, kernel_size=3, activation='relu'))
model.add(Conv2D(64, kernel_size=5, padding='same',
activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(128, kernel_size=3, activation='relu'))
model.add(Conv2D(128, kernel_size=3, activation='relu'))
model.add(Conv2D(128, kernel_size=5, padding='same',
activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(256, kernel_size=3, activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dense(256))
model.add(Dense(128))
model.add(Dense(5, activation='softmax'))

adam_optimizer=Adam(learning_rate=0.002)
model.compile(optimizer=adam_optimizer,
loss='categorical_crossentropy', metrics=['accuracy'])
model.summary()

```

## 4.2 Збір даних

Для проведення дослідження ефективності навчання мережі на 2D даних було обрано Labeled Faces in the Wild (LFW)[16] - добре відомий і широко використовуваний набір даних у сфері комп'ютерного зору та машинного навчання. Він був створений для сприяння дослідженням і розробкам у сфері розпізнавання та перевірки облич. LFW, в основному, використовується для порівняльного аналізу та оцінки продуктивності алгоритмів і моделей розпізнавання обличчя. LFW містить понад 13 000 позначених зображень облич, зібраних з Інтернету, що робить його великим і різноманітним набором даних. Кожне зображення обличчя в наборі даних LFW позначено іменем людини. Ці позначені дані використовуються для навчання та тестування систем розпізнавання обличчя.

LFW відома своєю високою варіативністю щодо пози, виразу, освітлення та оклюзії. Ці варіанти створені, щоб зробити набір даних більш репрезентативним для реальних сценаріїв.

CASIA-3D FaceV1 - це набір даних, який використовується для дослідження розпізнавання облич[17]. CASIA-3D FaceV1 фокусується на даних 3D обличчя.

Набір даних призначений для полегшення розробки та оцінки алгоритмів 3D-розпізнавання обличчя. Він містить тривимірні скани обличчя людей, які можуть бути надійнішими та інформативнішими для завдань розпізнавання обличчя порівняно з традиційними 2D-зображеннями. Також набір містить моделі з різними виразами обличчя що буде дуже корисним при навчанні моделі.

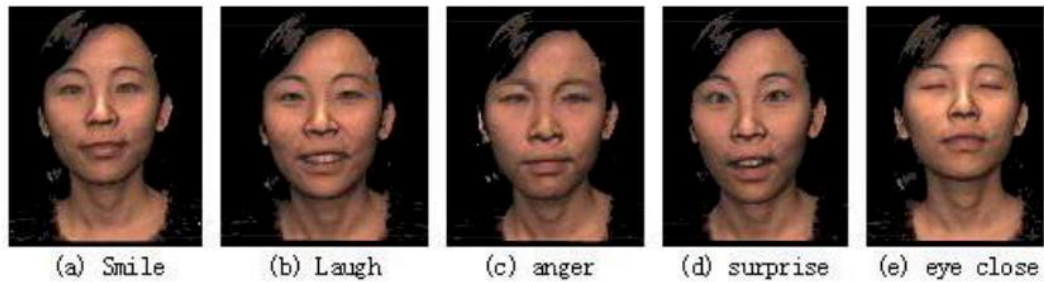


Рисунок 4.1 - Приклад 3D моделей в наборі даних

Тому для проведення досліджень у цій роботі використовуються набір даних CASIA-3D FaceV1 з розставленими метриками для ідентифікації осіб.

Набори даних поділені на набір для навчання та набір для тестування, де набір для навчання містить зображення людей для навчання моделі, а набір для тестування використовується для оцінки ефективності моделі в розпізнаванні облич.

### 4.3 Перетворення 3D моделей в данні для навчання мережі

Для використання 3D моделі обличчя для навчання мережі використана технологія хмар точок, як вхідних даних, що наведена в лістингу 4.2. Хмара точок буде використана як вхідні дані для нейронної мережі. Кожна точка в хмарі представлена як вектор з координатами. Цей вектор подається нейронній мережі для навчання.

#### Лістинг 4.2 - Перетворення 3D моделей на вхідні дані

```
import open3d as o3d
model = o3d.io.read_triangle_mesh("face_model.obj")
vertices = model.vertices
point_cloud = o3d.geometry.PointCloud()
point_cloud.points = o3d.utility.Vector3dVector(vertices)
point_cloud_list = [point_cloud]
```

```

X = point_cloud_list.iloc[:, :100*100].values.reshape(-1,
100, 100, 1)
y = point_cloud_list.iloc[:, -1].values
y = to_categorical(y, num_classes= 1+ df.loc[:,
'class'].unique().shape[0])
X_train,X_test,y_train,y_test=train_test_split(X, y,
random_state=42, test_size=0.15)

```

Приведений фрагмент коду буде використовуватись в дослідженні навчання на 3D даних, аналогічна дія для 2D даних наведена в лістингу 4.3, яка використовується для навчання моделі 2D-CNN.

#### Лістинг 4.3 - Перетворення 2D даних у вхідні данні

```

df = pd.read_csv('Dataset.csv', index_col=0)
X = df.iloc[:, :100*100].values.reshape(-1, 100, 100, 1)
y = df.iloc[:, -1].values
y = to_categorical(y, num_classes= 1+ df.loc[:,
'class'].unique().shape[0])
X_train,X_test,y_train,y_test=train_test_split(X, y,
random_state=42, test_size=0.15)

```

#### 4.4 Проведення навчання мережі без регуляризації

Перед початком тренування моделі необхідно загрузити вибірки на диск, і підключити їх у програму, ця дія наведена в лістингу 4.4.

#### Лістинг 4.4 - Підключення 2D сету даних

```

from google.colab import drive
drive.mount('/content/drive')
%cd 'drive/My Drive/2Ddata'

```

В роботі було проведено дослідження системи розпізнавання обличчя на основі розроблених моделей 2D-CNN і 3D-CNN. Для отримання інформації щодо ефективності розроблених моделей нейронних мереж проведено аналіз всіх отриманих графіків точності моделі (model accuracy) і втрат моделі (model loss) в пункті 4.6.

Для виведення графіків точності і втрат розробленої моделі нейронної мережі використовується фрагмент коду наведений в лістингу 4.5.

Лістинг 4.5 - Виведення графіків для оцінки ефективності нейромережі

```
plt.plot(history.history['acc'])
plt.plot(history.history['val_acc'])
plt.title('Model accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')
plt.show()

plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')
plt.show()
```

Першим проведено навчання моделі 2D-CNN і отримано графіки model accuracy і model loss, які є ключовими інструментами для оцінки та моніторингу мереж в процесі навчання. Після завершення навчання отримано графіки точності і втрат моделі 2D-CNN (рис. 4.2, 4.3), які і будуть використані для проведення дослідження ефективності 2D-CNN для завдання розпізнавання облич.

Графік точності моделі (рис. 4.2) показує, наскільки добре модель виконує передбачення на навчальних та валідаційних даних з плином часу.

Ця метрика важлива для визначення того, наскільки добре модель виконує завдання класифікації чи розпізнавання. Підвищення точності на тренувальних та валідаційних даних свідчить про покращення моделі.

Графік втрат моделі відображає, які втрати виникають під час навчання на тренувальних та валідаційних даних (рис. 4.3). Втрати визначаються як різниця між прогнозами моделі та справжніми значеннями цільової функції. Призначення графіка втрат полягає в тому, щоб дозволити визначити, як модель збільшує або зменшує втрати з кожною ітерацією навчання. Разом представлені два графіка надають повну картину процесу навчання нейронної мережі.

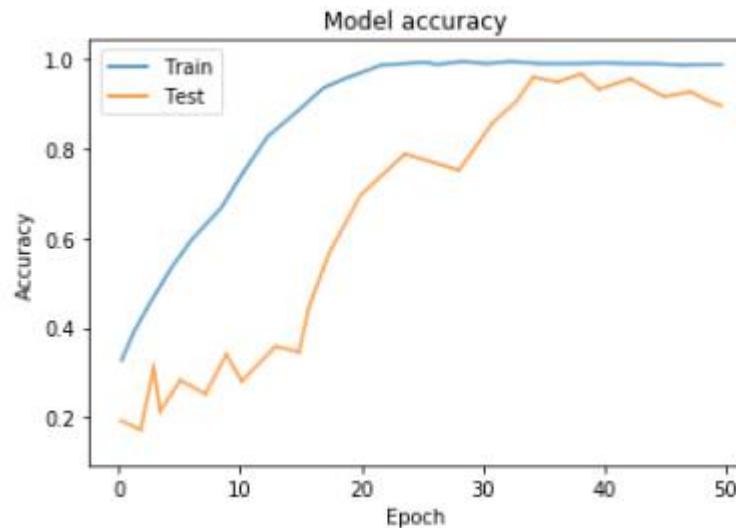


Рисунок 4.2 - Графік точності моделі навченої на 2D даних

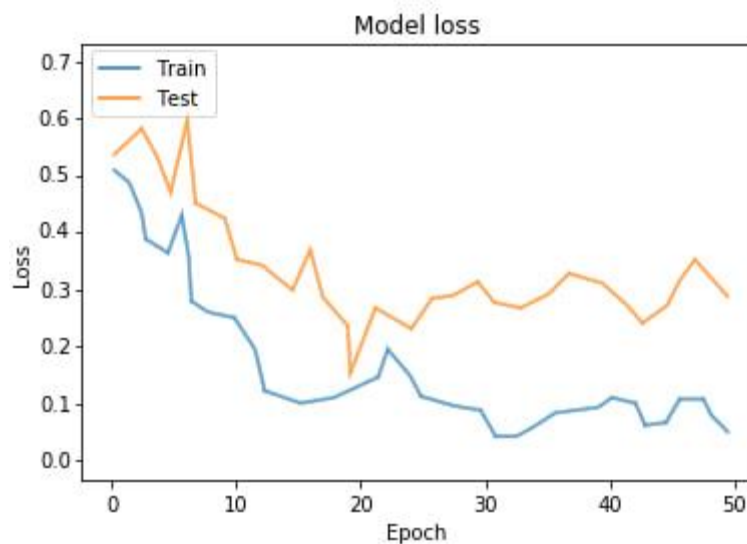


Рисунок 4.3 - Графік втрат моделі навченої на 2D даних

Наступний крок - проведення дослідження на 3D даних. Для цього проводимо навчання моделі 3D-CNN. Моделі навчаються на однакових обчислювальних потужностях, що забезпечує точність дослідження.

Навчання мережі було успішно завершено і отримано графіки точності і втрат моделі 3D-CNN (рис. 4.4, 4.5).

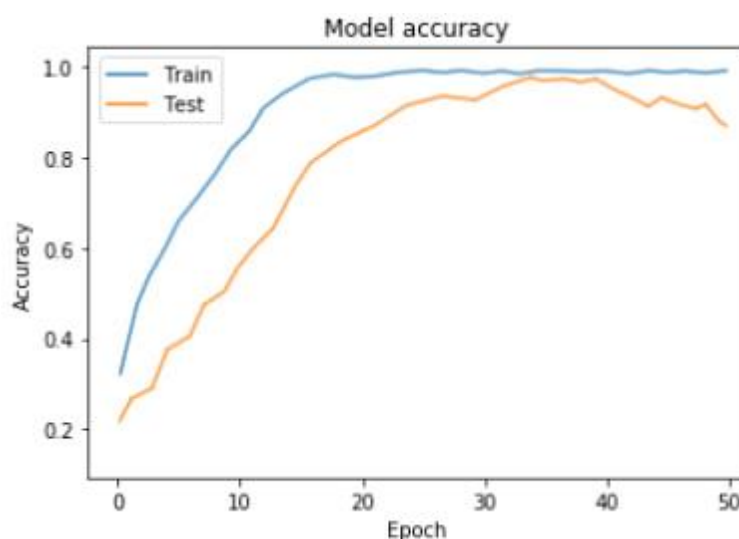


Рисунок 4.4 - Графік точності моделі навченої на 3D даних

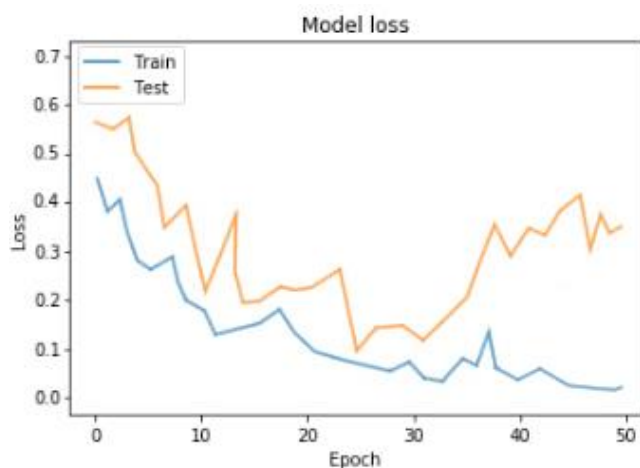


Рисунок 4.5 - Графік втрат моделі навченої на 3D даних

#### 4.5 Проведення навчання мереж з регуляризацією

Більшість проблем нейронних мереж вирішують методи регуляризації тому наступний крок дослідження це проведення навчання моделей з відповідними методами, а саме L1, L2 регуляризації. Першим проведено навчання при увімкненій L1 регуляризації на 2D наборі даних. І відповідно отримано графіки точності і втрат моделі 2D-CNN з використанням L1 регуляризації (рис. 4.6, 4.7).

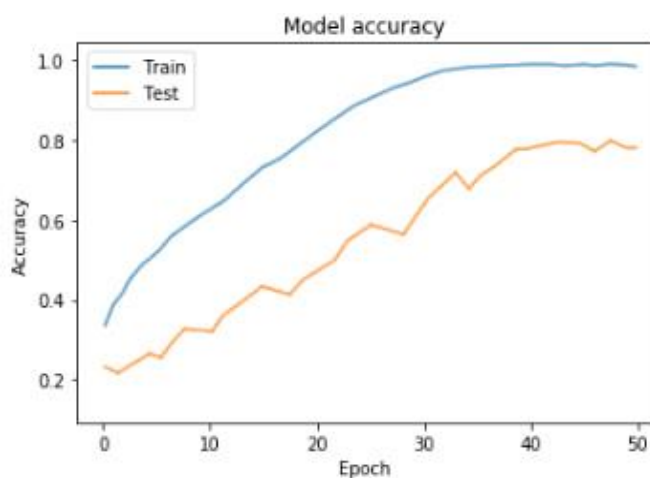


Рисунок 4.6 - Графік точності моделі навченої на 2D даних з увімкненою L1 регуляризацією

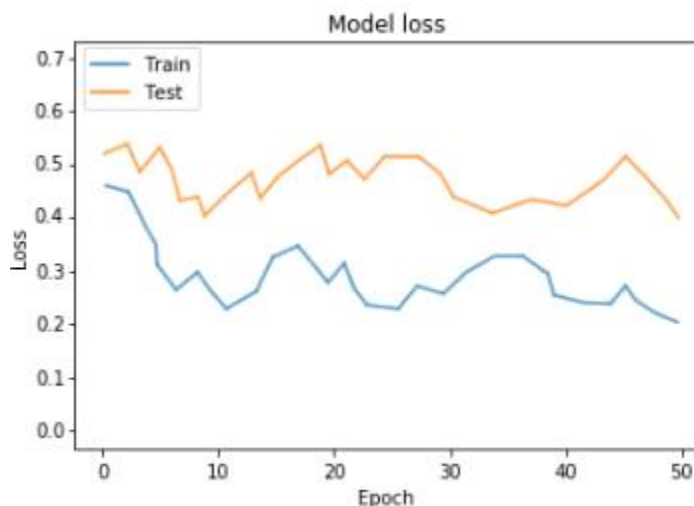


Рисунок 4.7 - Графік втрат моделі навченої на 2D даних з увімкненою L1 регуляризацією

Наступним проведено навчання на 3D наборі даних і отримано відповідні графіки точності і втрат моделі 3D-CNN з використанням L1 регуляризації (рис. 4.8, 4.9).

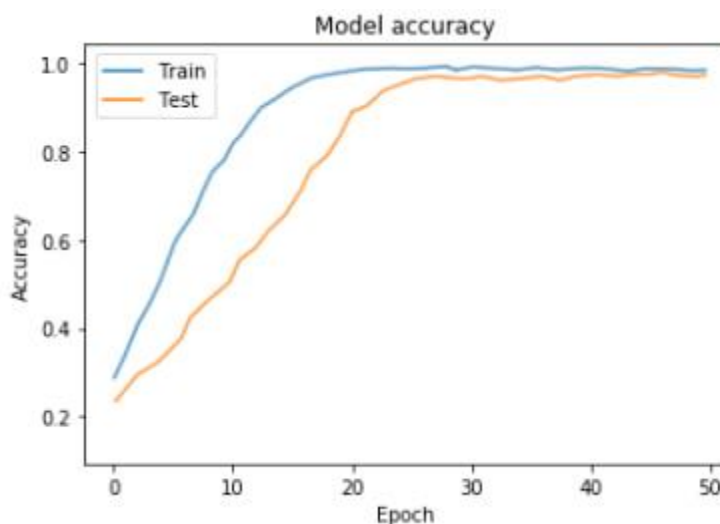


Рисунок 4.8 - Графік точності моделі навченої на 3D даних з увімкненою L1 регуляризацією

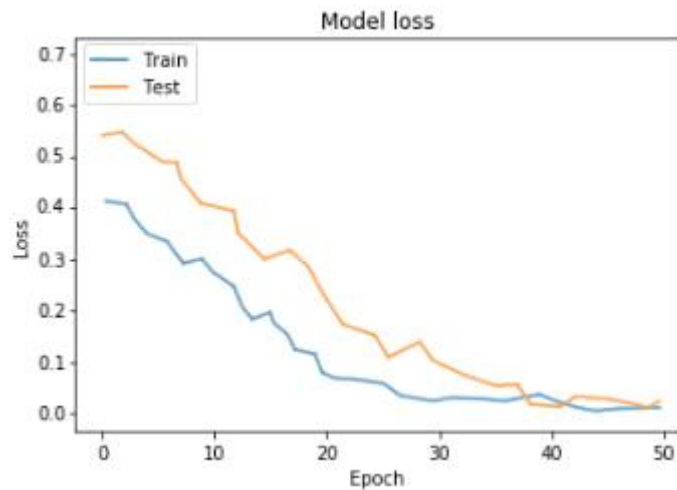


Рисунок 4.9 - Графік втрат моделі навченої на 3D даних з увімкненою L1 регуляризацією

Також для проведення порівняння ефективності використання L1 з L2 регуляризаціями проведено навчання при увімкненій L2 регуляризації на 2D наборі даних і отримано графіки точності і втрат моделі 2D-CNN з використанням L2 регуляризації (рис. 4.10, 4.11).

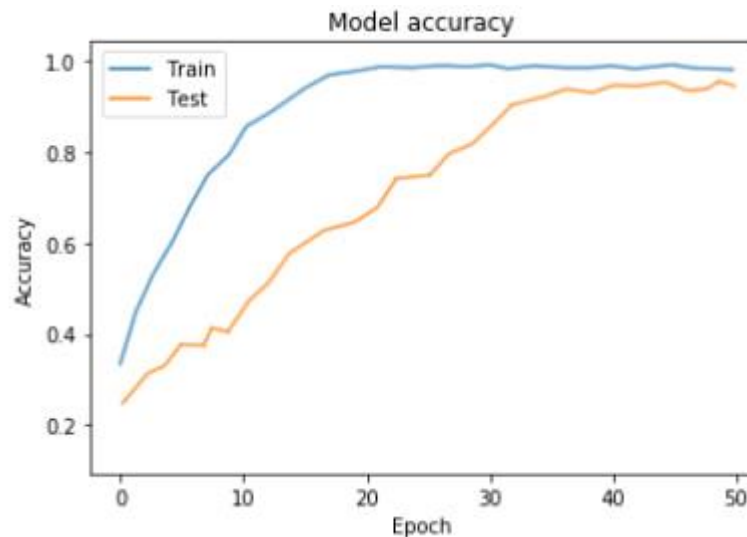


Рисунок 4.10 - Графік точності моделі навченої на 2D даних з увімкненою L2 регуляризацією

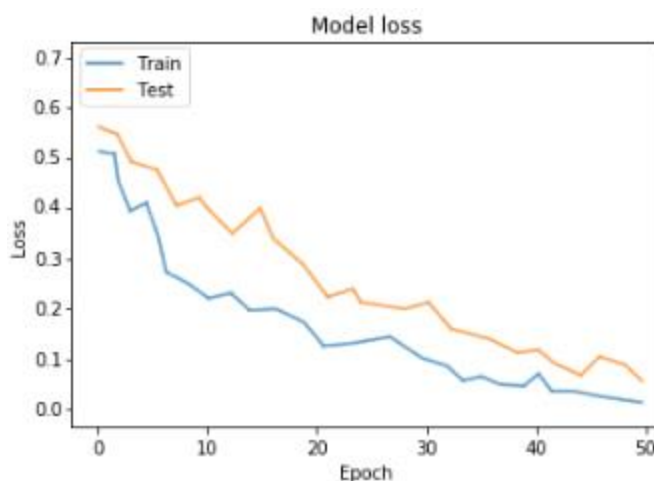


Рисунок 4.11 - Графік втрат моделі навченої на 2D даних з увімкненою L2 регуляризацією

Аналогічно проведено навчання на 3D наборі даних і отримано графіки точності і втрат моделі 3D-CNN з використанням L2 регуляризації (рис. 4.8, 4.9).

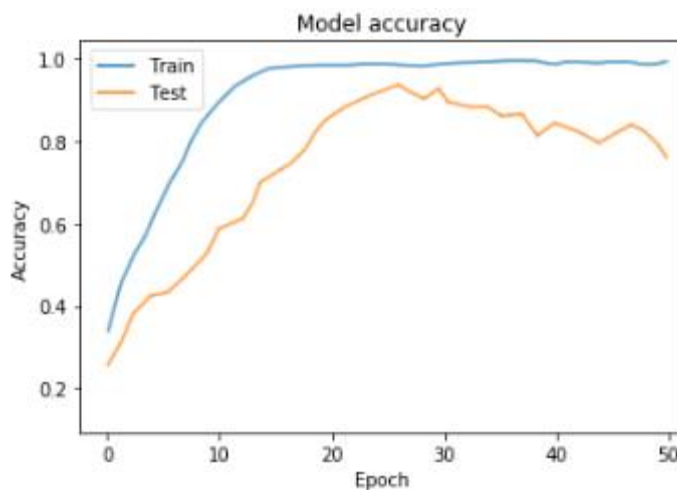


Рисунок 4.12 - Графік точності моделі навченої на 3D даних з увімкненою L2 регуляризацією

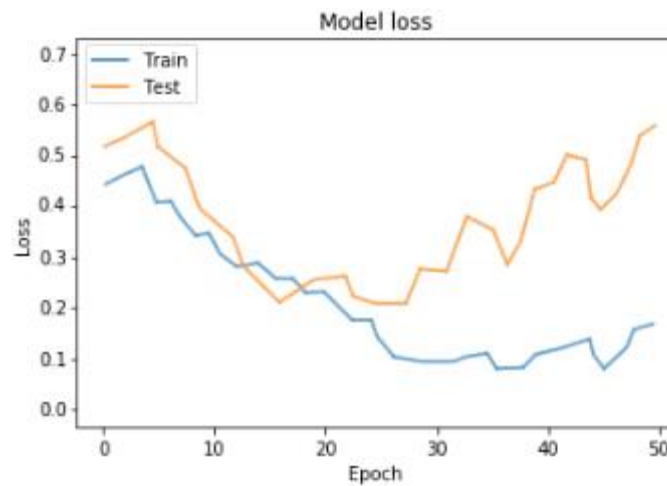


Рисунок 4.13 - Графік втрат моделі навченої на 3D даних з увімкненою L2 регуляризацією

#### 4.6 Аналіз отриманих результатів

Головна мета дослідження нейронних мереж розпізнавання облич - це визначення ефективності використання 3D моделей облич замість 2D зображень, а також дослідження, пов'язане з визначенням того, які методи регуляризації будуть найбільш ефективними. Для цього в попередніх розділах було проведено дослід на спроектованій нейронній мережі згорткового типу. Всі отримані числові значення точності і втрат розроблених моделей записані у таблицю 4.2. Щоб зробити висновки по дослідженню необхідно проаналізувати отриманні графіки.

Таблиця 4.2 - Результати дослідження

	Тренувальний набір: точність	Тестовий набір: точність	Тренувальний набір: функція втрат	Тестовий набір: функція втрат
2D-CNN	0.98461	0.89796	0.06142	0.29096
3D-CNN	0.99784	0.88513	0.02429	0.35483
2D-CNN+L1	0.98532	0.79581	0.20192	0.40310

3D-CNN+L1	0.99701	0.99012	0.00719	0.01039
2D-CNN+L2	0.98912	0.97391	0.01038	0.06307
3D-CNN+L2	0.99931	0.78301	0.18305	0.57502

В першому експерименті було проведено навчання нейронної мережі на 2D і 3D сетах даних. З наведених графіків точності і втрат моделі 2D-CNN навченої на 2D сеті (рис. 4.2, 4.3) видно, що мережа змогла опанувати навчальну вибірку але узагальнення даних відбулось не в достатній мірі, до того ж навчання моделі відбувалось не стабільно. Також графік 4.3 має великі втрати і після досягнення певного значення характеристика не має спадного характеру.

З графіків моделі, яка була навчена на 3D даних, спостерігається протилежна ситуація. Згідно графіку, зображеному на рисунку 4.4, модель 3D-CNN успішно опанувала навчальну вибірку а також змогла досягти вищої позначки на тестовій вибірці в піку, після чого точність прогнозів моделі 3D-CNN без використання методів регуляризації почала зменшуватись. Також на графіку втрат моделі 3D-CNN (рис. 4.5) видно, що залежність значення втрат моделі 3D-CNN на тестовій вибірці від кількості епох зменшується впродовж 30 епох, після досягнення позначки у 30 епох значення втрат почало зростати. Це все свідчить що модель зіткнулась з перенавченістю, коли вона занадто добре адаптувалась до тренувальної вибірки але її здатність щодо узагальнювати данні почала зменшуватись.

З дослідження приведених залежностей можна зробити наступне заключення: основна проблема моделі, навченої на 2D даних, це брак інформації для навчання, через що вона не може повноцінно узагальнити необхідні риси для розпізнавання облич за короткий час. Натомість друга модель має достатньо інформації з 3D облич і змогла досягти високої позначки точності за короткий час навчання, але має великий шанс зіткнутись з проблемою перенавченості, що знизить її ефективність. В цьому експерименті в кінцевому моменті модель навчена на 2D даних виявилась

більш ефективною.

В другому експерименті метою було визначити які методи регуляризації необхідні побудованим моделям для вирішення основних проблем і порівняти їх ефективність з необхідною їм регуляризацією. Було використано L1 і L2 регуляризації. Варто перш за все відзначити, що L1 регуляризація погіршила показники моделі 2D-CNN навченої на 2D даних, це видно з рисунків 4.6 і 4.7, де значно знизилась максимальна отримана точність моделі і крива втрат майже не йшла на спад, що свідчить про дуже погану здатність моделі узагальнювати дані. Натомість L2 регуляризація значно погіршила значення моделі, яка навчалась на 3D наборі згідно рисунків 4.12 і 4.13, де зменшилась точність моделі і крива похибки має більшу крутизну характеристики в порівнянні з першим експериментом.

На рисунках 4.8 і 4.9 наведено графіки які свідчать про покращення точності та стабільності моделі 3D-CNN навченої на 3D даних з ввімкненою L1 регуляризацією. Порівнявши криві залежностей на рисунку 4.8 з відповідними на рисунку 4.4 можна зробити висновок про значне покращення як максимальної точності моделі так і кінцевої точності, яка на відміну від попереднього експерименту не зменшилась. Також на графіках 4.9 і 4.5 зафіксовано значне покращення кривої втрат, яка при ввімкненій L1 регуляризації зменшується впродовж всього часу спостереження. Це все свідчить про вирішення проблеми перенавченості, яка виникла в першому досліді.

В свою чергу L2 регуляризація значно покращила показники моделі, яка навчалась на 2D даних, це видно на рисунках 4.10 і 4.11. Згідно графіку 4.10 модель змогла за відведені 50 епох узагальнити наявні данні й дійти до прийнятної похибки, також порівнявши криві залежностей на графіках точності (рис. 4.10) з відповідним на рисунку 4.2 можна зробити висновок, що навчання проходило значно стабільніше. Також з графіку 4.11 видно що втрати постійно зменшувались що свідчить що модель успішно завершила навчання і проблема браку інформації була частково вирішена.

Результати дослідження: найкращі результати продемонстрували модель 3D-CNN з ввімкненою L1 регуляризацією і 2D-CNN з L2 регуляризацією. Відповідно L1 регуляризація є кращою з протестованих для 3D-CNN а L2 для 2D-CNN. Максимальна ефективність моделі навченої на 3D даних вища за аналогічний показник моделі, яка навчалась на 2D наборі, також перша модель швидше навчається й має менше втрат. Тож в результаті цього дослідження модель, яка навчалась на 3D даних виявилась більш ефективною ніж модель навчена на 2D зображеннях.

Рекомендації щодо використання досліджуваного методу у подальших розробках: 3D дані містять значно більше інформації про риси обличчя у порівнянні з 2D зображеннями, тому доцільним є використання цього методу у випадках, коли збір значної кількості 2D зображень є ускладненим або неможливим, тоді використання 3D даних значно пришвидшить розробку необхідної системи.

## ВИСНОВКИ

Кваліфікаційна робота присвячена дослідженню ефективності використання в розробці системи розпізнавання облич на основі нейронних мереж 3D моделей облич для проведення навчання замість стандартних 2D зображень.

Розробка нейронних мереж для розпізнавання облич є надзвичайно популярною в сучасному світі завдяки зростанню доступності великих обсягів даних та збільшення обчислювальної потужності, що сприяє покращенню точності та швидкості роботи систем комп'ютерного зору.

Основною метою розвитку архітектури нейронних мереж є виявлення більш ефективних підходів для вирішення певних задач. Це дозволяє розробникам застосовувати нейронні мережі для більшого спектру задач і знаходити більш ефективні рішення старих завдань для покращення роботи розроблених систем. Також важливим аспектом є виявлення ефективних методів регуляризації даних, це значно підвищує ефективність систем.

На основі проведеного аналізу наукової літератури, а також власного досвіду було виявлено перспективну і малодосліджену технологію в проектуванні нейронних мереж розпізнавання облич, а саме, використання для навчання мережі 3D моделей замість 2D зображень. Основна виявлена перевага даного підходу полягає у більшій кількості інформації для навчання мережі, яка міститься на 3D моделі. Це допомагає вирішити проблему нестачі даних в тренувальній вибірці, що призводить до недонавченості системи.

В результаті проведено дослідження і аналізу існуючих архітектур і аспектів їх побудови, було виявлено потрібний набір інструментів для проведення подальших досліджень, а отже: була обрана архітектура 3D-CNN, перетворення 3D моделей на вхідні дані за допомогою технології хмар точок, функції активації ReLu і softmax, запропоновані методи регуляризації типу L1 і L2.

В результаті аналізу математичної бази було сформовано основні математичні принципи роботи нейронних мереж, що є необхідною частиною в розробці будь яких систем, що використовують алгоритми глибокого навчання.

Для проведення практичного дослідження було здійснено проектування двох моделей нейронних мереж з архітектурами 2D-CNN і 3D-CNN, для проведення навчання на відповідних наборах даних. Програмування моделей відбувалось мовою python, з використанням бібліотек TensorFlow і Keras.

Для аналізу отриманих результатів дослідження було виведено спеціальні графіки залежностей ефективності і втрат на тренувальній і тестувальній вибірках, що дає повну інформацію про ефективність моделі розпізнавання облич. Провівши аналіз отриманих результатів трьох експериментів, а саме навчання моделей 2D-CNN і 3D-CNN без регуляризації, а також з регуляризацією Лассо і регуляризацією Рідже, отримали наступні результати:

- найкращу ефективність показала 3D-CNN з L1 регуляризацією;
- найкращий тип регуляризації для 2D-CNN є L2 регуляризація;
- відповідно неправильно підібраний метод регуляризації може привести до погіршення результатів.

Згідно проведеного дослідження системи розпізнавання облич з використанням в якості навчальної вибірки 3D моделі з відповідними архітектурами і методами показують кращі результати у випадках, коли збирання великої кількості 2D зображень певних осіб є неможливим або важким, що призводить до недостатньої кількості навчальної інформації на 2D наборі, в той час, коли менша кількість 3D моделей забезпечує потрібний результат.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Ian Goodfellow, Deep Learning (Adaptive Computation and Machine Learning Series) / Ian Goodfellow, Yoshua Bengio, Aaron Courville, Francis Bach. // The MIT Press, 2017 - 800p.
2. Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow - 2nd Edition // O'Reilly Media, Inc., 2019 - 856p. - ISBN: 9781492032649
3. Richard Szeliski, Computer Vision: Algorithms and Applications - 2nd Edition // Springer, 2022 - 947 p.
4. LeCun, Y., Deep learning. / LeCun, Y., Bengio, Y., Hinton, G. // Nature - 2015 - 521(7553). - P. 436-444
5. Кахаєв М.С. Дослідження і оптимізація систем розпізнавання облич на основі нейронної мережі / М.С. Кахаєв // матеріали тез 27-го міжнародного молодіжного форуму «Радіоелектроніка та молодь у XXI столітті» - м. Харків, травень 2023. - С. 47-48.
6. David Novotny, Learning 3D Object Categories by Looking Around Them / David Novotny, Diane Larlus, Andrea Vedaldi. // International Conference on Computer Vision, 2017(ICCV) - 2017 - P. 5228-5237
7. Charu C. Aggarwal, Neural Networks and Deep Learning: A Textbook - 1st Edition // Springer, 2018 - 520p.
8. Shuiwang Ji, 3D Convolutional Neural Networks for Human Action Recognition / Shuiwang Ji, Wei Xu, Ming Yang, Kai Yu // IEEE Transactions on Pattern Analysis and Machine Intelligence ( Volume: 35, Issue: 1, January 2013) - 2012 - P. 221 - 231.
9. Charles R. Qi, PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. / R. Qi Charles, Hao Su, Mo Kaichun, Leonidas J. Guibas // IEEE Conference on Computer Vision and Pattern Recognition (CVPR) - 2017 - P. 77-85.

10. Tanay Agrawal, Hyperparameter Optimization in Machine Learning // Apress Berkeley, CA, 2020 -166p.

11. Activation Functions in Neural Networks [Электронный ресурс]. Режим доступа <https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6>

12. Trevor Hastie, Regularization and Variable Selection Via the Elastic Net / Trevor Hastie, Hui Zou // Journal of the Royal Statistical Society Series B: Statistical Methodology, Volume 67, Issue 2, - 2005 - P. 301–320.

13. David J.C. MacKay, Information Theory, Inference, and Learning Algorithms Version 7.2 (fourth printing) // Cambridge University Press, 2005 - 628p.

14. Grégoire Montavon, Neural Networks: Tricks of the Trade / Grégoire Montavon, Geneviève B. Orr, Klaus-Robert Müller // Springer Berlin, Heidelberg, 2012 - 769p.





15. Trevor Hastie, L1-Regularization Path Algorithm for Generalized Linear Models / Trevor Hastie, Mee Young Park // Journal of the Royal Statistical Society Series B: Statistical Methodology, Volume 69, Issue 4, - 2007 - P. 659–677.

16. Labelled Faces in the Wild (LFW) Dataset [Электронный ресурс]. Режим доступа <https://www.kaggle.com/datasets/jessicali9530/lfw-dataset>

17. CASIA 3D Face Database [Электронный ресурс]. Режим доступа <http://biometrics.idealtest.org/>

## Відомість кваліфікаційної роботи

«Система розпізнавання облич на основі нейронної мережі»

	Прізвище та ініціали відповідальної особи	Підпис	Дата
Роботу виконав студент групи СКСМ-22-2 Структура кваліфікаційної роботи: – пояснювальна записка <u>66</u> с.; – графічний матеріал <u>18</u> арк..	Кахаєв М.С.		01.01.24
Керівник роботи	Ларченко Л.В.		01.01.24
Перевірка на плагіат здійснена. Оригінальність авторського тексту складає <u>99</u> %	Литвинова Є.І.		06.01.24
Нормоконтроль проведено :	Ларченко Л.В.		10.01.24