

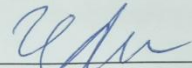
Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

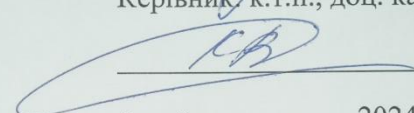
на тему:

**СИСТЕМА РОЗПІЗНАВАННЯ ЕМОЦІЙ ЛЮДИНИ ЗА ВИРАЗОМ
ОБЛИЧЧЯ**

Виконав: студент 2 курсу, групи 2КІ-23м
спеціальності 123 – «Комп'ютерна інженерія»

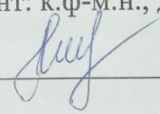
 Чмих Р. В.

Керівник: к.т.н., доц. каф. ОТ

 Кадук О. В.

«__» _____ 2024 р.

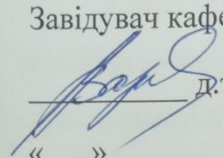
Опонент: к.ф-м.н., доц. каф. МБІС

 Салієва О. В.

«__» _____ 2024 р.

Допущено до захисту

Завідувач кафедри ОТ

 д.т.н., проф. Азаров О. Д.

«__» _____ 2024 р.

ВНТУ 2024

ВІННИЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій та комп'ютерної інженерії

Кафедра обчислювальної техніки

Галузь знань — Інформаційні технології

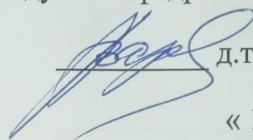
Освітній рівень — магістр

Спеціальність — 123 Комп'ютерна інженерія

Освітньо-професійна програма — Комп'ютерна інженерія

ЗАТВЕРДЖУЮ

Завідувач кафедри обчислювальної техніки



д.т.н., проф. Азаров О. Д

« 18 » вересня 2024 р.

ЗАВДАННЯ

НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ

студенту Чмиху Роману Віталійовичу

1 Тема роботи «Система розпізнавання емоцій людини за виразом обличчя» керівник роботи Азаров О. Д., д.т.н., проф. каф. ОТ, затверджено наказом вищого навчального закладу № 310 від 17.09.2024 року.

2 Строк подання студентом роботи 19.12.2024 р.

3 Вихідні дані до роботи: призначення системи — автоматизація розпізнавання емоцій на основі зображень облич, що забезпечує підвищення ефективності аналізу стану користувачів у різних сценаріях, таких як маркетингові дослідження, освітні програми та психологічна підтримка. Тип взаємодії — обробка вхідних даних у вигляді зображень.

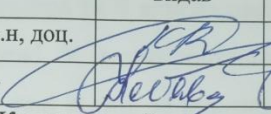
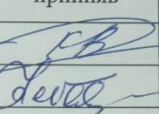
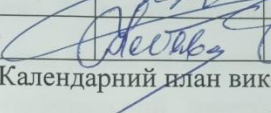
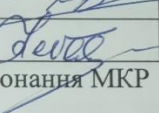
4 Зміст текстової частини (перелік питань, які потрібно розробити): вступ, аналіз впливу психологічних особливостей людини на вимоги до функціоналу

системи розпізнавання емоцій, (постановка задачі, опис інструментів розробки та визначення принципів роботи системи), реалізація програми, тестування та оцінка точності моделі, економічна частина.

5 Перелік графічного матеріалу: використано 7 таблиць та 23 рисунки.

6 Консультанти розділів роботи приведені в таблиці 1.

Таблиця 1 — Консультанти розділів роботи

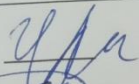
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		видав	прийняв
1-4	Кадук Олександр Володимирович, к.т.н, доц.		
5	Небава Микола Іванович, к.е.н., проф.		

7 Дата видачі завдання 18.09.2024 р. Календарний план виконання МКР приведений в таблиці 2.

Таблиця 2 — Календарний план

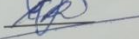
№ з/п	Назва етапів МКР	Строк виконання	Примітка
1	Постановка задачі	20.09.2024	Виконано
2	Огляд існуючих рішень	24.09.2024	Виконано
3	Теоретичні дослідження	29.09.2024	Виконано
4	Розробка структурної схеми	03.10.2024	Виконано
5	Розрахунок аналогової частини	07.10.2024	Виконано
6	Розробка принципової схеми	13.10.2024	Виконано
7	Моделювання роботи системи	20.10.2024	Виконано
8	Розрахунок економічної частини	27.10.2024	Виконано
9	Оформлення ПЗ та ілюстративного матеріалу	03.11.2024	Виконано
10	Виконання магістерської кваліфікаційної роботи	10.11.2024	Виконано
11	Перевірка якості виконання МКР та усунення недоліків	14.12.2024	Виконано
12	Підписи у керівника, опонента, нормоконтролера	15.12.2024	Виконано
13	Перевірка «антиплагіат»	17.11.2024	Виконано
14	Попередній захист	18.11.2024	Виконано

Студент



Чмих Р. В.

Керівник



к.т.н., доц. каф. ОТ Кадук О. В.

АНОТАЦІЯ

УДК 004

Чмих Р. В. "Система розпізнавання емоції людини за виразом обличчя".

Магістерська кваліфікаційна робота зі спеціальності 123 —

Комп'ютерна Інженерія, Вінниця: ВНТУ, 2024, 102 с.

На укр. мові. Бібліогр.: 30 назв; рис.: 24; табл. 5.

У дипломній роботі розроблено та підвищено точності визначення емоцій шляхом створення моделі класифікатора та її впровадження у вигляді системи на базі згорткової нейронної мережі. У межах дипломного проєкту реалізовано систему, яка здатна визначати такі емоції за виразом обличчя, як злість, відраза, страх, щастя, спокій (нейтральний вираз), смуток та здивування.

Ключові терміни: глибоке навчання, згорткова нейронна мережа, комп'ютерний зір, розпізнавання емоцій, штучний інтелект.

ANNOTATION

УДК 004

Chmykh R.V. "Emotion Recognition System Based on Facial Expressions."
Master's Qualification Thesis in the specialty 123 — Computer Engineering, Vinnytsia:
VNTU, 2024, 102 pages.

In Ukrainian language. Bibliography: 30 entries; figures: 24; tables: 5.

The aim of the development is to increase the accuracy of emotion detection by creating a classifier model and implementing it in the form of a system based on a convolutional neural network. Within the framework of the diploma project, a system has been implemented that is capable of recognizing emotions such as anger, disgust, fear, happiness, calmness (neutral expression), sadness, and surprise based on facial expressions.

Key terms: deep learning, convolutional neural network, computer vision, emotion recognition, artificial intelligence.

Зміст

ВСТУП	8
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ	11
1.1 Постановка мети для задачі розпізнавання емоцій людини за виразом обличчя	11
1.2 Опис і особливості задачі розпізнавання емоцій	12
1.3 Існуючі методи вирішення задачі	13
1.4 Існуючі рішення розпізнавання емоцій	19
1.5 Постановка вимог до розробки системи розпізнавання емоцій людини	21
2 ПРОЄКТУВАННЯ СИСТЕМИ ДЛЯ РОЗПІЗНАВАННЯ ЕМОЦІЙ ЛЮДИНИ ЗА ВИРАЗОМ ОБЛИЧЧЯ	23
2.1 Вибір моделі нейронної мережі для вирішення задачі класифікації емоцій	23
2.1.1 Штучні нейронні мережі	23
2.1.2 Згорткові нейронні мережі	25
2.1.3 Рекурентні нейронні мережі	30
2.1.4 Багатошарові нейронні мережі	34
2.2 Вибір набору даних	37
2.3 Вибір бібліотеки для виявлення обличчя на відео	38
2.4 Вибір алгоритму оптимізації навчання ЗНМ	40
2.5 Вибір функції втрат для задачі класифікації	45
2.6 Вибір метрики оцінки якості класифікації	47
2.7 Опис вхідних даних	49
2.8 Побудова моделі	51
2.9 Тренування моделі	56
3 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ	61
3.1 Вибір засобів розробки	61

					08-54.МКР.039.00.000 ПЗ			
		№	Підпис					
Розробив	Чмих Р. В.				Система розпізнавання емоцій людини за виразом обличчя Пояснювальна записка	Літ.	Аркуш	Акрушів
Керівник	Кадук О. В.						6	102
Опонент	Шиян А. А.					ВНТУ, гр. 2КІ-23м		
Н. Контр.	Швець С. І.							
Затвердив	Азаров О. Д.							

4 ТЕСТУВАННЯ ПРОГРАМИ	63
4.1 Побудова графіків та аналіз отриманих результатів	63
4.2 Результати тестування навченої моделі.....	68
5 ЕКОНОМІЧНА ЧАСТИНА.....	71
5.1 Комерційний та технологічний аудит науково-технічної розробки.....	71
5.2 Прогнозування витрат на виконання науково-дослідної (дослідно-конструкторської) роботи.....	74
5.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором	79
ВИСНОВКИ.....	85
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	86
ДОДАТОК А Технічне завдання	88
ДОДАТОК Б Схема нейронної мережі.....	93
ДОДАТОК В Порівняння оптимізаторів	94
ДОДАТОК Г Лістинг файлу Main.py	95
ДОДАТОК Д Лістинг файлу Model.py	97
ДОДАТОК Е Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень	100

ВСТУП

У сучасному світі інформаційних технологій люди постійно взаємодіють з різноманітними пристроями та програмами, що вимагає активної участі. Це створює необхідність розуміти людські емоції та поведінку, що є ключовим аспектом для розвитку інтелектуальних систем. Одним із рішень цієї проблеми є розробка систем, здатних розпізнавати емоції за виразом обличчя. Такі системи можуть автоматично аналізувати міміку та визначати настрій людини. Розпізнавання емоцій є складним завданням, оскільки люди виражають свої почуття по-різному залежно від культурних, соціальних і індивідуальних факторів. Важливо враховувати такі елементи, як рухи очей, положення губ і інші зміни в міміці, щоб досягти точніших результатів. Ці системи мають значення в багатьох галузях, таких як психологія, медицина, реклама, розваги та безпека. Наприклад, у психології вони допомагають визначати емоційний стан людини, а в медицині – виявляти патології, як-от депресію чи тривожність. У рекламній сфері та індустрії розваг аналіз емоцій може покращувати взаємодію користувачів із контентом. У сфері безпеки такі системи можуть допомагати запобігати насильству, аналізуючи емоції людей у публічних місцях. Сучасні технології глибокого навчання, зокрема згорткові нейронні мережі, відкривають нові можливості для покращення точності та швидкості таких систем. Головна мета полягає в створенні надійної та точної системи, яка здатна в режимі реального часу розпізнавати емоції за виразом обличчя. Хоча сьогодні досягнуто значних успіхів у цій галузі, існують виклики, пов'язані з підвищенням точності та швидкості аналізу. Попит на такі технології зростає через розвиток соціальних мереж і віртуальної реальності, де розпізнавання емоцій може підвищити ефективність користувацького досвіду та рекламних кампаній. Тому ця робота є важливим внеском у розвиток технологій, пов'язаних з аналізом емоцій, і має великий потенціал для покращення якості життя людей у різних сферах.

Отже, виходячи із розглянутого, задачі подальшого розроблення та вдосконалення методів пошуку, виявлення та виділення рухомих об'єктів у

цифрових зображеннях, що є кращими за критеріями достовірність та швидкість, на теперішній час є **актуальними**.

Метою дослідження є підвищення ефективності розпізнавання емоцій за рахунок удосконалення метода класифікатора та системи, що його реалізує на базі згорткової нейронної мережі.

Задачі дослідження:

- виконати аналіз існуючих систем розпізнавання обличчя;
- запропонувати поліпшений метод розпізнавання емоцій за виразом обличчя;
- створити алгоритм та розробити програму розпізнавання емоцій людини за виразом обличчя.

Об'єкт дослідження — процес розпізнавання емоцій людини за допомогою аналізу виразу обличчя.

Предмет дослідження — алгоритми та методи глибокого навчання для автоматичного розпізнавання емоцій людини на основі аналізу мімічних характеристик обличчя.

Методи дослідження:

- аналіз літературних джерел щодо існуючих підходів до розпізнавання емоцій;
- використання методів комп'ютерного зору для збору та обробки зображень обличчя;
- моделювання нейронних мереж, зокрема згорткових нейронних мереж, для автоматизації процесу розпізнавання емоцій. Експериментальне дослідження ефективності розробленої моделі на тестових наборах даних. Порівняльний аналіз точності та швидкості роботи розробленої системи з існуючими рішеннями.

Наукова новизна одержаних результатів полягає у тому, що розроблено модель розпізнавання емоцій людини на основі згорткових нейронних мереж, яка забезпечує підвищену точність у порівнянні з традиційними методами. Запропоновано метод обробки мімічних характеристик, який враховує індивідуальні та культурні особливості виразу емоцій. Забезпечено можливість

роботи системи в режимі реального часу для застосування у сферах психології, медицини, безпеки та індустрії розваг.

Практичне значення одержаних результатів:

- психологія та медицина;;
- безпека;
- індустрія розваг;
- соціальні мережі та віртуальна реальність.

Система може допомагати в діагностиці психічних станів, таких як депресія або тривожність, на основі аналізу емоцій пацієнтів. Використання системи у публічних місцях може допомогти виявляти підозрілу поведінку або попереджати потенційні акти насильства, аналізуючи емоції людей у натовпі.

Вона також може використовуватися для моніторингу емоційного стану пацієнтів під час терапії або реабілітації.

Вона може поліпшити взаємодію в онлайн-комунікаціях, забезпечуючи більш точне відображення емоцій в аватарах або інших віртуальних репрезентаціях користувачів.

Система може використовуватися в іграх або віртуальних середовищах для адаптації контенту до емоційного стану користувача, підвищуючи рівень взаємодії та занурення у гру.

Апробація матеріалів роботи виконана під час LIV конференції.

Публікація за темою роботи: Р. В. Чмих, О. В. Кадук. СИСТЕМА РОЗПІЗНАВАННЯ ЕМОЦІЇ ЛЮДИНИ ЗА ВИРАЗОМ ОБЛИЧЧЯ. ВІТКП ВІНТУ. Факультет інформаційних технологій та комп'ютерної інженерії. Матеріали LIV Всеукраїнської науково-технічної конференції факультету інформаційних технологій та комп'ютерної інженерії (2025) .

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Постановка мети для задачі розпізнавання емоцій людини за виразом обличчя

Метою роботи є підвищення ефективності розпізнавання емоцій шляхом розробки моделі класифікатора та системи на базі згорткової нейронної мережі. Основними показниками ефективності системи є точність та швидкість розпізнавання. Точність визначається відсотком правильно класифікованих емоцій серед усіх тестових зображень. Оскільки система працює в режимі реального часу, її швидкість є ключовим параметром. Система повинна розпізнавати емоції в режимі реального часу, відображаючи відеопотік з веб-камери та негайно показуючи результат класифікації емоцій. Алгоритм розпізнавання емоцій можна поділити на дві основні задачі:

- виявлення обличчя людини у відеопотоці;
- класифікація емоцій на основі виразу обличчя (емоції, як-от злість, відраза, страх, щастя, спокій, смуток, здивування).

Для вирішення цих завдань використано методи глибокого навчання, які дозволяють автоматично виділяти ключові ознаки зображень для подальшої класифікації. Завдання дипломної роботи включають:

- вивчення особливостей розпізнавання емоцій;
- огляд існуючих методів та рішень для цієї задачі;
- визначення вимог до системи розпізнавання емоцій;
- аналіз і вибір найбільш відповідної нейронної мережі для реалізації завдання; вибір відповідного набору даних;
- вибір бібліотек для виявлення обличчя у відеопотоці;
- вибір алгоритму оптимізації навчання нейронної мережі, функції втрат і метрик оцінки якості класифікації;

Дипломний проєкт спрямований на практичне застосування, оскільки система розпізнавання емоцій може бути корисною у таких сферах, як медицина, психологія, маркетинг, розваги тощо.

1.2 Опис і особливості задачі розпізнавання емоцій

Емоційний штучний інтелект — це напрямок штучного інтелекту, який дозволяє комп'ютерним системам розпізнавати та інтерпретувати людські емоції через аналіз виразу обличчя, мови тіла та голосу. Такий ШІ виступає інструментом для покращення природної взаємодії між машинами і людьми, оскільки здатен аналізувати тонкі зміни в міміці (мікровирази), голосі й жестах, реагуючи на них подібно до людини [1].

Розвиток технологій комп'ютерного зору і машинного навчання робить розпізнавання емоцій більш точним і доступним для широкого використання. Технологія розпізнавання емоцій на основі обличчя є частиною обробки зображень. Вона може допомагати в маркетингу, медицині та інших галузях, де необхідно розуміти емоційні реакції людини на події або стимули. Зважаючи на зростання інтересу до аналізу емоцій, тема дипломної роботи "Система розпізнавання емоцій людини за виразом обличчя" є надзвичайно актуальною. Особливо це важливо в сучасних умовах, коли соціальні мережі, відеоконференції та віддалене спілкування стали невід'ємною частиною життя, посилюючи потребу у точному розпізнаванні емоцій для покращення комунікації. Метою цієї роботи є створення системи, здатної розпізнавати емоції людини в режимі реального часу через веб-камеру, застосовуючи методи комп'ютерного зору, обробки зображень та машинного навчання.

Ця система має практичне значення для багатьох галузей, таких як медицина, психологія, маркетинг і розваги. Розпізнавання емоцій за виразом обличчя є технічно складним завданням, оскільки емоції людей виражаються по-різному, залежно від індивідуальних особливостей, культурних і соціальних факторів. Додаткові виклики створюють змінні умови освітлення та якість зображення, що впливає на точність роботи системи. Крім того, люди часто виражають одночасно кілька емоцій, що ускладнює точне розпізнавання.

Окрім традиційних застосувань, емоційний штучний інтелект знаходить нові перспективи в контексті розвитку технологій, таких як віртуальна та доповнена

реальність. У цих сферах розпізнавання емоцій дозволяє створювати більш інтерактивні та персоналізовані досвіди для користувачів. Наприклад, у відеоіграх емоційний ШІ може адаптувати поведінку персонажів в залежності від реакцій гравця, що значно підвищує рівень занурення у гру. Крім того, розпізнавання емоцій може бути корисним у сфері освіти. Системи, що аналізують емоційний стан учнів, можуть допомогти викладачам зрозуміти, коли учні зацікавлені, а коли втрачають мотивацію. Це відкриває нові можливості для адаптивного навчання, що може покращити результати учнів та забезпечити індивідуальний підхід до кожного студента.

Важливим аспектом розробки системи є також етика та конфіденційність. Використання емоційного штучного інтелекту викликає питання щодо збору та зберігання особистих даних, що може вплинути на приватність користувачів. Тому важливо враховувати правові та етичні аспекти при впровадженні таких технологій у різні сфери. Зрештою, розпізнавання емоцій за виразом обличчя може стати важливим інструментом для створення більш чутливих до людських потреб систем. Воно має потенціал не лише для покращення технологічних рішень, але й для розвитку більш гуманістичного підходу в міжособистісній взаємодії між людьми та машинами.

Отже, розпізнавання емоцій на основі виразу обличчя є комплексним завданням, що потребує використання сучасних методів машинного навчання та обробки зображень для забезпечення високої точності та ефективності роботи системи.

1.3 Існуючі методи вирішення задачі

Для кращого розуміння варто ознайомитися з компонентами штучного інтелекту, зображеними на рисунку 1.1. Штучний інтелект охоплює всю область досліджень у цій сфері. До його складових належать машинне навчання і комп'ютерний зір, які є окремими напрямками.

Нейромережі виступають одним із видів машинного навчання, а глибоке навчання представляє архітектуру нейромереж, що визначає підходи до їх проєктування та навчання.



Рисунок 1.1 – Складові штучного інтелекту

Штучний інтелект є частиною комп'ютерної лінгвістики, яка спрямована на створення інтелектуальних машин, здатних виконувати завдання, що зазвичай потребують людського інтелекту. Це можуть бути завдання від розпізнавання мови або зображень до складніших, як-от ігри чи керування автомобілем.

Комп'ютерний зір, як окрема галузь штучного інтелекту, займається обробкою зображень та відео для таких завдань, як розпізнавання облич, ідентифікація об'єктів і аналіз зображень.

Основою таких систем є алгоритми машинного навчання, які дозволяють визначати закономірності, відрізнити об'єкти та знаходити подібності. Подібно до того, як людина вчиться розпізнавати об'єкти з раннього дитинства, системи комп'ютерного зору навчаються розпізнаванню через аналіз даних.

Для вирішення завдань, таких як ідентифікація об'єктів чи класифікація зображень, широко застосовуються згорткові нейронні мережі. Хоча машинне навчання може працювати з різними типами даних, як-от числові, текстові чи аудіо, комп'ютерний зір орієнтується лише на обробку візуальної інформації, наприклад, фотографій чи відео. Машинне навчання, у свою чергу, є підгалуззю штучного інтелекту, яка досліджує методи створення алгоритмів, здатних до самонавчання.

Його основною метою є розв'язання завдань на основі вхідних даних, і чим різноманітніші ці дані, тим точніше модель може визначати закономірності. Для початку машинного навчання необхідні три складові: дані, ознаки й алгоритми.

Дані є набором інформації, що використовується для навчання алгоритмів, і можуть бути представлені у форматах тексту, зображень, відео, аудіо чи таблиць. Вони бувають розміченими, коли містять додаткові позначки або теги, та нерозміченими, що потребує використання спеціальних алгоритмів для їх обробки. Ознаки — це характеристики об'єктів дослідження, представлені у числовій чи категоріальній формі, як-от пробіг автомобіля чи вік користувача.

Алгоритм — це формальний опис процесу навчання моделі, який включає послідовність дій для обробки даних і створення моделі.

Різні алгоритми можуть застосовуватися для виконання однієї задачі, а вибір методу впливає на точність, швидкість роботи та розмір моделі.



Рисунок 1.2 — Складові машинного навчання

Взагалі розрізняють два типи машинного навчання:

- навчання по прецедентах (індуктивне навчання);
- дедуктивне навчання.

Оскільки, дедуктивне навчання відносять до області експертних систем, то терміни «машинне навчання» і «навчання по прецедентах» можна вважати синонімами. Навчання по прецедентах поділяється на три основні типи:

- контрольоване навчання або навчання з учителем;
- неконтрольоване навчання або навчання без учителя;
- навчання з підкріпленням.

Крім перелічених методів, існують також інші способи навчання, такі як активне, багатозадачне, різноманітне, трансферне тощо. Особливо динамічно розвивається глибоке навчання, яке дозволяє об'єднувати алгоритми навчання з учителем та без учителя [5].

Розглянемо детальніше основні типи навчання на основі прецедентів: Контрольоване навчання – метод, що використовується у випадках, коли доступні великі обсяги маркованих даних. Наприклад, якщо є тисячі зображень фруктів із мітками ("яблуко", "банан"), потрібно створити алгоритм, який здатен визначити, що зображено на новому, раніше невідомому зображенні – яблуко чи банан. Роль "вчителя" у цьому процесі виконує людина, яка заздалегідь розмітила дані. Машина самостійно обирає ознаки, за якими розрізняє об'єкти. Завдяки цьому створений алгоритм можна швидко адаптувати до подібних задач.

Неконтрольоване навчання – метод, що застосовується, коли дані не мають міток. Нерозмічені дані трапляються значно частіше, ніж марковані. Завдання машини у цьому випадку полягає у виявленні зв'язків між даними, пошуку закономірностей, шаблонів або структури, класифікації чи впорядкування інформації. Наприклад, неконтрольоване навчання використовується в рекомендаційних системах, коли після перегляду відео на платформі YouTube користувачу пропонують схожі відеоролики. Навчання з підкріпленням – метод, у якому "вчителем" виступає середовище. Агента (машина) не володіє початковими знаннями про середовище, але здатний виконувати в ньому дії. У відповідь

середовище надає зворотний зв'язок, який дозволяє агенту вчитися на своїх діях.

По суті, агент і середовище утворюють систему зі зворотним зв'язком. Навчання з підкріпленням схоже на процес навчання людей: правильні дії заохочуються, а помилки караються. Цей метод використовують для розв'язання складніших задач, ніж контрольоване та неконтрольоване навчання. Навчання з підкріпленням часто застосовується в системах навігації роботів, коли ті навчаються уникати перешкод шляхом отримання досвіду, заснованого на зворотному зв'язку під час зіткнень. Інші приклади використання включають логістику, складання графіків, планування завдань та навчання гри в логічні ігри [7].

Глибоке навчання є напрямком машинного навчання, що використовує багатошарові нейронні мережі для вирішення завдань, таких як класифікація, розпізнавання об'єктів, синтез мови тощо. Його основна ідея полягає у створенні мережі з багатьма прихованими шарами, здатної автоматично знаходити відповідності між вхідними даними та очікуваними результатами.

Під час навчання параметри моделі коригуються так, щоб мінімізувати різницю між прогнозами та реальними значеннями. Глибоке навчання ефективно використовується для задач, що вимагають розуміння складних даних, таких як зображення, аудіо або текст.

Зокрема, глибокі нейронні мережі знаходять застосування в розпізнаванні облич, мови, автоматичному перекладі та інших сферах обробки природних мов. Застосування глибокого навчання можливе, якщо нейронна мережа має кілька прихованих шарів. Ці шари можуть чергуватися з логічними перетвореннями, що дозволяє кожному наступному рівню виявляти взаємозв'язки попереднього.

Такі мережі здатні аналізувати не лише базові залежності, а й складніші, що виникають між цими залежностями. За типом алгоритмів можна виділити два підходи: класичне навчання, що базується на алгоритмах, розроблених понад 50 років тому, і сучасні нейронні мережі, які вирішують задачі розпізнавання або генерації зображень, відео, моделювання нелінійних об'єктів та інших складних систем.

Глибоке навчання спирається на багатошарові нейронні мережі, які автоматично виявляють складні залежності у даних. У моєму проєкті система розпізнавання емоцій за виразом обличчя поєднує елементи машинного та глибокого навчання, а також комп'ютерного зору.

Комп'ютерний зір використовується для виявлення обличчя на відео з вебкамери, тоді як розпізнавання емоцій (злість, відраза, страх, щастя, нейтральний вираз, смуток і здивування) виконується згортковими нейронними мережами. Оскільки навчання моделі проводиться на маркованих даних, це є прикладом навчання з учителем, де методи машинного навчання поєднуються з підходами глибокого навчання.

Для аналізу емоцій існують два способи: контактний, що використовує фізіологічні показники (наприклад, пульс), і безконтактний, який аналізує міміку, жести та рух очей. У проєкті обрано безконтактний спосіб, оскільки він не потребує дорогого обладнання, спеціалістів або фізичної присутності людини.

Існують два підходи до вирішення завдання: перший передбачає використання набору розмічених зображень облич, що дає можливість нейронній мережі самостійно навчатися розпізнавати емоції.

Другий полягає у позначенні маркерів емоцій, таких як посмішка або нахмурені брови, однак цей підхід вимагає багато часу та не дозволяє визначати емоції без явних маркерів. Через ці обмеження обрано перший підхід, який хоч і потребує великого репрезентативного набору даних, але забезпечує ширші можливості для розпізнавання емоцій.

Система використовує бібліотеку OpenCV для аналізу та обробки зображень у режимі реального часу. Для виявлення обличчя застосовується Haar Cascade Frontal Face Model, що дозволяє точно ідентифікувати обличчя на екрані. Таким чином, проєкт базується на глибокому навчанні, що реалізується через згорткові нейронні мережі, у поєднанні з технологіями комп'ютерного зору. Додатково до використання згорткових нейронних мереж, у системі можна інтегрувати техніки зменшення шуму та попередньої обробки даних для покращення точності аналізу. Наприклад, для підвищення якості вхідних зображень застосовується нормалізація

яскравості, корекція контрасту та усунення зайвих артефактів.

Ще одним можливим напрямком вдосконалення є додавання моделі для аналізу контексту. Це може бути корисно для врахування додаткової інформації, яка впливає на емоційний стан людини, наприклад, тону голосу, пози чи жестів. Інтеграція мультимодальних даних розширить можливості системи, роблячи її здатною більш точно оцінювати складніші емоційні стани. Для цього можна використати рекурентні нейронні мережі (RNN) або трансформери, які добре підходять для обробки послідовних даних.

З точки зору практичного застосування, така система розпізнавання емоцій має великий потенціал у різних галузях. Наприклад, вона може використовуватися у психологічному консультуванні, щоб допомагати фахівцям аналізувати емоційний стан клієнтів.

У сфері освіти система здатна адаптувати навчальний процес відповідно до емоцій учнів. Крім того, у комерції ця технологія може застосовуватися для аналізу реакцій клієнтів на продукти або рекламу, що сприятиме створенню більш персоналізованих пропозицій.

1.4 Існуючі рішення розпізнавання емоцій

Існують різні системи для розпізнавання емоцій на обличчях, які використовують машинне навчання та комп'ютерний зір. Однією з найвідоміших програм є Affectiva, яка дозволяє аналізувати емоції в реальному часі за допомогою відео з веб-камери.

Affectiva використовує сучасні алгоритми машинного навчання та обробки зображень, а також надає готовий API, що забезпечує інтеграцію в інші програми та можливість налаштування власних моделей.

Серед переваг Affectiva слід виділити високу точність розпізнавання, завдяки якій система коректно визначає широкий спектр емоцій, а також гнучкість і зручність використання. Affectiva успішно застосовується у багатьох галузях, таких як реклама, медицина, аналіз ринку, ігрова індустрія та навіть у психологічних дослідженнях. Інтуїтивно зрозумілий API значно спрощує

інтеграцію цієї системи в існуючі рішення, роблячи її популярною серед розробників.

Однак система має і певні недоліки. Одним із головних є висока вартість використання API, що може бути недоступним для індивідуальних розробників або невеликих компаній. Крім того, хоча точність розпізнавання висока, у деяких випадках система може поступатися іншим спеціалізованим рішенням. Обмежені можливості кастомізації алгоритмів розпізнавання також можуть стати перешкодою для реалізації унікальних або вузькоспеціалізованих завдань. Таким чином, Affectiva є потужним інструментом, але його використання доцільне насамперед для великих проєктів із достатнім фінансуванням.

Microsoft Emotion API є хмарним сервісом, який пропонує API для розпізнавання емоцій на зображеннях за допомогою машинного навчання. Цей інструмент здатний аналізувати зображення та визначати емоції на обличчі людини.

Однією з головних переваг є зручний API, який підтримує різні мови програмування, що дозволяє легко інтегрувати його в різні системи. Крім того, він може працювати з як фотографіями, так і відео, що значно розширює можливості використання. Цей сервіс також можна інтегрувати в уже існуючі системи розпізнавання облич та інші API, що робить його універсальним інструментом для розпізнавання емоцій.

Проте, є й деякі недоліки. Microsoft Emotion API не завжди точно визначає емоції на зображеннях, що може призвести до помилок у розпізнаванні. Крім того, використання цього API може бути дорогим, особливо при великій кількості оброблених фотографій або відео, що обмежує його застосування для масштабних проєктів з великими обсягами даних. Оскільки цей сервіс працює через Інтернет, необхідність постійного підключення може стати проблемою в умовах обмеженої мережевої пропускної здатності або відсутності стабільного Інтернет-з'єднання.

Kairos Emotion Recognition є ще одним популярним інструментом для розпізнавання емоцій на основі зображень облич. Система використовує глибинне навчання та штучні нейронні мережі, що дозволяє досягти високої точності

розпізнавання емоцій. Вона підтримує різні формати зображень, включаючи JPEG, PNG, BMP, а також відеофайли у форматі MP4, що дає користувачам гнучкість у виборі джерела даних.

Однак, система має і деякі обмеження, такі як висока вартість використання через підписку, а також залежність від якості зображень. Якщо зображення нечітке або низької якості, це може знизити точність розпізнавання емоцій. У порівнянні з готовими комерційними рішеннями, система розпізнавання емоцій, розроблена власноруч, має кілька вагомих переваг. Вона безкоштовна, має відкритий програмний код, що дозволяє вносити необхідні корективи та налаштування під конкретні потреби проєкту.

Крім того, така система здатна здійснювати розпізнавання емоцій у реальному часі, що робить її придатною для інтеграції в різні сфери, від медичних застосунків до ігрових технологій. Проте важливо врахувати, що власноруч розроблена система може бути менш точною в порівнянні з комерційними рішеннями, якщо якість відео або позиція обличчя на екрані не є оптимальними.

1.5 Постановка вимог до розробки системи розпізнавання емоцій людини

Метою дипломного проєкту є розробка системи розпізнавання емоцій з використанням згорткової нейронної мережі, що дозволить підвищити ефективність класифікації емоцій за виразом обличчя.

Система має розпізнавати сім базових емоцій: злість, відразу, страх, щастя, спокій (нейтральний вираз), смуток і здивування, аналізуючи зображення.

Для досягнення поставленої мети передбачено вирішення таких задач, як розробка, навчання та впровадження моделі класифікатора у вигляді програмного забезпечення. Для виконання проєкту необхідно вивчити принципи роботи штучних нейронних мереж та провести аналіз сучасних архітектур, визначивши найбільш підходящу для розпізнавання емоцій. На основі цього буде створено та навчено модель нейронної мережі, здатну в реальному часі аналізувати вирази обличчя та класифікувати емоції.

Результат роботи системи буде представлено у вигляді віконного додатку, який забезпечить відображення відеопотоку з веб-камери разом із позначенням розпізнаної емоції. Для оцінки роботи моделі будуть побудовані графіки точності та втрат, що дозволить зробити висновки про ефективність системи.

Очікувані результати дозволять створити універсальну систему розпізнавання емоцій, яка знайде застосування у багатьох галузях, включаючи безпеку, рекламу, розваги, психологію та медицину.

Такі рішення сприятимуть впровадженню інновацій у ці сфери, надаючи можливість аналізувати емоційний стан людей у реальному часі, підвищувати рівень персоналізації сервісів та забезпечувати більш глибоке розуміння потреб користувачів.

2 ПРОЄКТУВАННЯ СИСТЕМИ ДЛЯ РОЗПІЗНАВАННЯ ЕМОЦІЙ ЛЮДИНИ ЗА ВИРАЗОМ ОБЛИЧЧЯ

2.1 Вибір моделі нейронної мережі для вирішення задачі класифікації емоцій

2.1.1 Штучні нейронні мережі

Штучні нейронні мережі (ШНМ) є потужними інструментами сучасної науки і техніки, здатними ефективно зберігати, обробляти та аналізувати великі обсяги інформації. Їхня основна ідея полягає у створенні математичної моделі, яка імітує принципи функціонування людського мозку, особливо його здатність до навчання, адаптації, класифікації та узагальнення даних. Завдяки цій здатності ШНМ знаходять застосування в багатьох галузях, де необхідна обробка величезних обсягів інформації, зокрема, в комп'ютерному зорі, обробці природної мови, робототехніці, медичній діагностиці та багатьох інших сферах.

ШНМ складаються з безлічі пов'язаних між собою елементів — нейронів, які організовані у кілька шарів (Рис. 2.1.). Ці шари мають чітку структуру, що включає:

— вхідний шар, який приймає вхідні дані; Кількість нейронів у цьому шарі визначається кількістю параметрів або ознак, які використовуються для навчання. Наприклад, для зображення розміром 28x28 пікселів вхідний шар може мати 784 нейрони.

1. приховані шари — забезпечують обробку інформації; ШНМ можуть містити від одного до десятків, а іноді й сотень прихованих шарів залежно від складності завдання. Нейрони в цих шарах виконують нелінійні перетворення вхідних даних, що дозволяє мережі моделювати складні залежності.

2. вихідний шар — відповідає за формування результату. Кількість нейронів у цьому шарі залежить від задачі. Наприклад, у задачі класифікації з трьома класами вихідний шар матиме три нейрони, кожен з яких представлятиме ймовірність належності об'єкта до конкретного класу.

Зв'язки між нейронами мають ваги, які визначають, як сильно один нейрон впливає на інший. Саме оптимізація цих ваг у процесі навчання дозволяє ШНМ досягати високої точності у вирішенні завдань.

Однією з ключових переваг ШНМ є їх здатність працювати з величезними обсягами даних. Це забезпечується завдяки багатошаровій структурі мережі та використанню спеціальних алгоритмів навчання, які дозволяють виділяти приховані закономірності в даних.

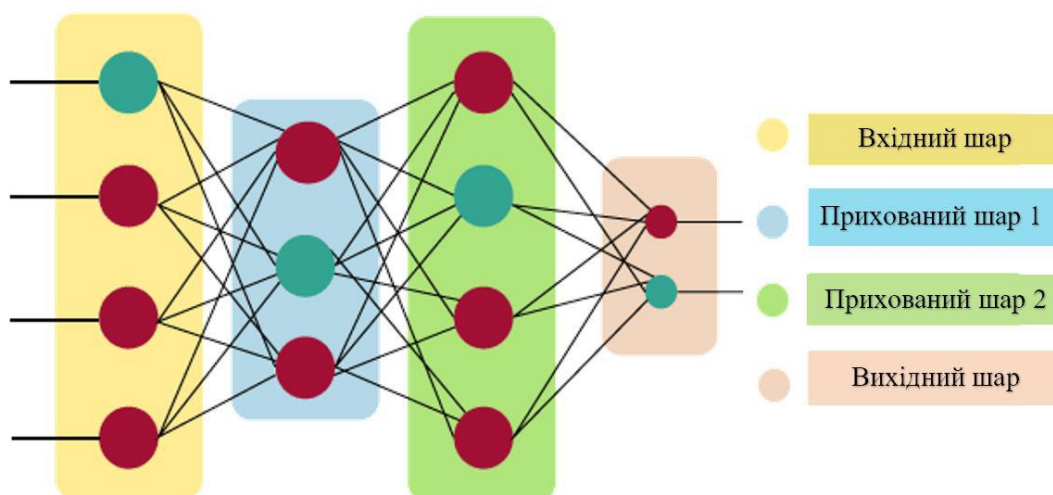


Рисунок 2.1.1.1 – Архітектура ШНМ

Обробка великих обсягів інформації — ШНМ здатні працювати з високорозмірними даними, зокрема зображеннями, аудіо, відео, текстами та числовими масивами. Їхня структура дозволяє зберігати взаємозв'язки між різними елементами даних навіть у випадку, якщо їх кількість перевищує можливості традиційних методів аналізу.

Стиснення даних — ШНМ можуть використовуватися для зменшення розмірності даних без втрати основної інформації. Це досягається за рахунок виділення найважливіших ознак у вхідному масиві даних. Наприклад, у завданнях обробки зображень згорткові нейронні мережі (CNN) стискають інформацію, залишаючи лише ті ознаки, які критично важливі для класифікації або розпізнавання.

Асоціативна пам'ять — це ще одна важлива властивість ШНМ. Вона дозволяє мережі відновлювати втрачену або пошкоджену інформацію на основі частини даних. Наприклад, якщо зображення має пошкодження, нейронна мережа може відновити відсутні фрагменти, використовуючи знання про подібні зображення, які

вона отримала під час навчання.

Процес навчання нейронної мережі включає кілька основних кроків:

— пряме поширення — вхідні дані проходять через всі шари мережі до вихідного, і формується результат;

— Обчислення помилки — результат порівнюється з очікуваним значенням, і на основі цього обчислюється помилка.

— Зворотне поширення помилки — помилка використовується для корекції ваг у зворотному напрямку, починаючи з вихідного шару до вхідного.

— Оновлення ваг — ваги зв'язків між нейронами коригуються таким чином, щоб зменшити помилку в наступних ітераціях навчання.

Процес навчання повторюється до тих пір, поки модель не досягне необхідного рівня точності.

Глибинні нейронні мережі (ГНМ) — це розширення традиційних ШНМ, які містять значно більшу кількість прихованих шарів. Це дозволяє їм вирішувати завдання, що вимагають аналізу складних структур даних. ГНМ застосовуються у завданнях:

1. Комп'ютерний зір — розпізнавання облич, ідентифікація об'єктів на зображеннях, сегментація сцен.
2. Обробка мови — переклад текстів, розпізнавання голосу, аналіз настрою.
3. Рекомендаційні системи — формування індивідуальних рекомендацій для користувачів на основі їхніх уподобань.

Таким чином, штучні нейронні мережі відкривають безмежні можливості для аналізу, зберігання та ефективного використання великих обсягів інформації. Їх універсальність та здатність до адаптації роблять їх ключовими технологіями у світі великих даних.

2.1.2 Згорткові нейронні мережі

Згорткові нейронні мережі (ЗНМ) є однією з найпотужніших моделей машинного навчання, які забезпечують високу ефективність у задачах

розпізнавання зображень, обробки сигналів, а також багатьох інших напрямках аналізу даних. Ця модель отримала широку популярність завдяки своїй здатності автоматично виявляти локальні ознаки у великих наборах даних, таких як зображення, відео чи часові ряди. Схему ЗНМ показано на рисунку. 2.1.2.1.

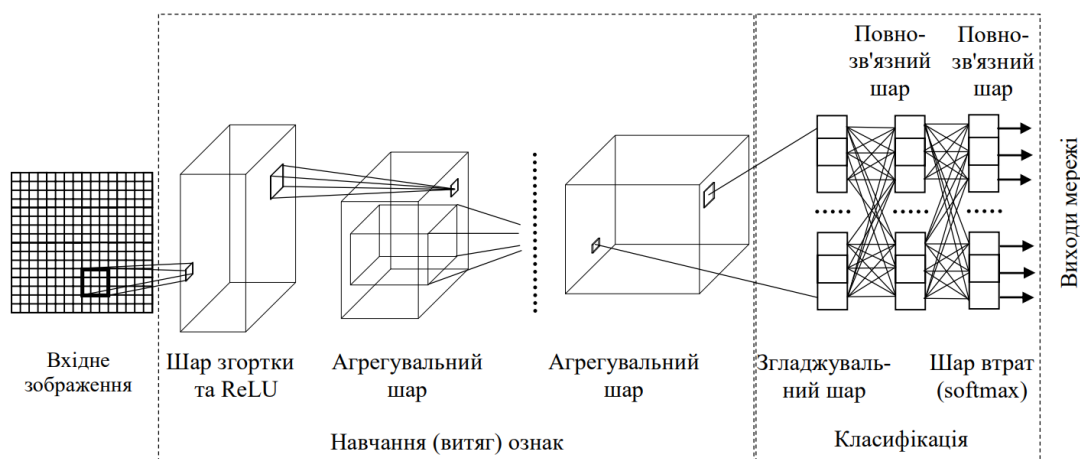


Рисунок 2.1.2.1 – Схема ЗНМ

ЗНМ зазвичай складаються з трьох основних компонентів: вхідного шару, вихідного шару і декількох прихованих шарів. У прихованих шарах реалізовано спеціалізовані операції, які дозволяють моделі навчатися та узагальнювати особливості вхідних даних. Основні типи прихованих шарів включають:

— згорткові шари: ці шари виконують ключову функцію мережі — виділення локальних ознак, таких як краї, текстури, контури об'єктів та інші патерни. Для цього використовуються фільтри (або ядра), які проходять через вхідні дані, виконуючи математичну операцію згортки. Вихід цього шару є тензором ознак, що містить інформацію про локальні взаємозв'язки між пікселями. Завдяки використанню декількох фільтрів, мережа здатна навчатися різноманітних шаблонів, що є важливими для задач розпізнавання.

— агрегувальні шари: після кожного згорткового шару зазвичай додається шар агрегування (наприклад, максимального агрегування або середнього агрегування), який зменшує просторові розміри вихідного тензора. Ця операція сприяє зниженню обчислювальних витрат і кількості параметрів моделі, що

допомагає уникати перенавчання. Наприклад, агрегування за максимумом дозволяє зберегти найбільш релевантні ознаки, ігноруючи менш значущі. Функції активації

Щоб модель могла обробляти складні та нелінійні взаємозв'язки у даних, після згорткових шарів застосовуються функції активації, такі як ReLU (Rectified Linear Unit), сигмоїда або тангенс Гіперболічний. ReLU є найпоширенішою, оскільки вона дозволяє швидко навчати модель, видаляючи негативні значення у виході шару та зберігаючи позитивні.

— згладжувальні шари: перед поданням даних у повнозв'язний шар, згорткові шари виконують згладжування — перетворення багатовимірних тензорів ознак у одновимірний вектор. Це спрощує подальшу обробку та інтеграцію результатів з наступними компонентами мережі.

— повнозв'язні шари: ці шари виконують функцію класифікації або регресії, зводячи обчислення попередніх шарів до фінального результату. Вони беруть вхідний вектор ознак і трансформують його в кінцевий вихідний вектор, де кожен елемент відповідає певному класу або прогнозованому значенню.

— шари нормалізації: щоб забезпечити стабільність і прискорити процес навчання, у ЗНМ використовуються шари нормалізації, які масштабують дані таким чином, щоб їх розподіл залишався стабільним протягом усіх етапів навчання. Це сприяє зменшенню ризику перенавчання та покращенню збіжності алгоритму оптимізації.

— шар втрат: цей шар розраховує відхилення між передбаченими результатами моделі та реальними мітками даних. У ньому використовуються різні функції втрат залежно від задачі: наприклад, для класифікації — крос-ентропія, а для регресії — середньоквадратична похибка.

Згорткові нейронні мережі працюють за принципом багаторівневого аналізу даних.

На нижніх рівнях мережа вивчає базові ознаки, такі як краї та текстури, тоді як на вищих рівнях — складніші патерни, наприклад форми об'єктів або їхню структуру.

ЗНМ забезпечують численні переваги, серед яких:

1. Висока точність у задачах класифікації та розпізнавання зображень.
2. Автоматичне виділення ознак без необхідності ручного інжинірингу.
3. Можливість обробки великих обсягів даних завдяки згортці та агрегуванню.

Згорткові нейронні мережі (ЗНМ) є особливим різновидом багатoshарових перцептронів, розроблених з метою мінімізації потреби у попередній обробці даних. Їх також називають інваріантними відносно зсуву (англ. shift invariant) або просторово інваріантними штучними нейронними мережами (англ. space invariant artificial neural networks, SIANN), що пояснюється їхньою архітектурою зі спільними вагами та здатністю зберігати інваріантність до паралельного перенесення.

Основою для створення ЗНМ став біологічний процес, зокрема структура та функції нейронів зорової кори тварин. Нейрони зорової кори активуються лише у відповідь на стимули в обмеженій області зорового поля, відомій як рецептивне поле. Ці поля частково перекриваються, забезпечуючи покриття всього зорового поля.

На відміну від багатьох інших алгоритмів класифікації зображень, ЗНМ потребують значно менше попередньої обробки даних. Мережа самостійно навчається фільтрів, які в традиційних алгоритмах створювали вручну. Така автоматизація, що зменшує залежність від людських зусиль і апіорних знань, є однією з найбільших переваг ЗНМ. Їхня гнучкість і ефективність роблять їх незамінними у завданнях розпізнавання зображень і відео, у рекомендаційних системах, а також в обробці природної мови.

Завдяки своїй архітектурі та функціональності ЗНМ продовжують знаходити широке застосування у різних галузях, демонструючи високу ефективність у вирішенні складних завдань аналізу даних.

Згорткові нейронні мережі активно використовуються у таких сферах, як:

- комп'ютерний зір — ідентифікація об'єктів, розпізнавання облич, сегментація сцен;
- Медична діагностика — виявлення аномалій на медичних зображеннях,

таких як рентгенівські знімки.

— Автономний транспорт — розпізнавання дорожніх знаків, виявлення пішоходів.

— Обробка сигналів — аналіз аудіо, відео, часових рядів. Таким чином, ЗНМ є ключовим компонентом сучасного машинного навчання та відкривають нові горизонти для автоматизації та аналізу великих обсягів інформації.

Агрегування, або pooling, є важливим елементом згорткових нейронних мереж (ЗНМ) і являє собою метод нелінійного зниження дискретизації. Одним із найбільш поширених видів агрегування є максимізаційне агрегування (max pooling), яке розділяє вхідне зображення на набір прямокутників без перекриття, після чого для кожної підобласті обирає її максимальне значення. Основна ідея цього методу полягає в тому, що точне положення ознаки менш важливе, ніж її приблизне положення відносно інших ознак.

Агрегувальний шар використовується для поступового зменшення просторового розміру представлення, що дозволяє знизити кількість параметрів і обчислень у мережі, а також допомагає контролювати перенавчання. У типових архітектурах ЗНМ агрегувальний шар зазвичай вставляється між послідовними згортковими шарами, що додає інваріантності відносно паралельного перенесення. Агрегувальний шар обробляє кожен зріз глибини входу окремо, зменшуючи його просторовий розмір.

Найчастіше використовуються фільтри розміру 2×2 із кроком 2, які зменшують розмірність кожного зрізу глибини входу вдвічі за шириною і висотою, що призводить до відкидання 75% збуджень. У цьому випадку кожна операція взяття максимуму охоплює чотири значення. Глибина при цьому залишається незмінною. Окрім максимізаційного агрегування, можливе використання й інших функцій, таких як усереднювальне агрегування (average pooling) та агрегування на основі L2-норми (L2-norm pooling). Історично усереднювальне агрегування було досить популярним, але з часом його витіснило максимізаційне агрегування, яке виявилось більш ефективним у багатьох практичних завданнях. Проте через агресивне скорочення розміру представлення останнім часом спостерігається

тенденція до використання менших фільтрів або навіть відмови від агрегувальних шарів у деяких архітектурах.

Варіацією цього підходу є агрегування областей інтересу (Region of Interest pooling, або RoI pooling), де вихід має фіксований розмір, а розмір прямокутника входу задається як параметр. Цей метод широко використовується в задачах виявлення об'єктів, зокрема в архітектурах швидких згорткових нейронних мереж на основі областей (Fast R-CNN). Агрегування є критично важливим компонентом для зменшення обчислювальної складності і підвищення ефективності згорткових нейронних мереж, особливо в задачах, що потребують обробки складних зображень або виділення ознак із високим рівнем деталізації.

Оскільки глибина карти ознак зменшується, шари, розташовані ближче до вхідного шару, зазвичай мають меншу кількість фільтрів, тоді як більш високі шари можуть містити значно більшу їх кількість. Це обумовлено тим, що початкові шари виконують виділення базових ознак, таких як краї або текстури, тоді як більш глибокі шари відповідають за розпізнавання складних структур і абстрактних патернів. Для забезпечення збалансованих обчислень на кожному етапі глибини мережі зазвичай підтримується сталий добуток між кількістю ознак і кількістю піксельних позицій. Це дозволяє уникати втрати важливої інформації про вхідні дані на кожному наступному рівні обробки. Кількість карт ознак напряму визначає ємність моделі, тобто її здатність навчатися і представляти складні залежності. Вибір кількості карт ознак залежить від обсягу доступних навчальних даних і складності поставленої задачі. Якщо прикладів недостатньо, надмірна кількість карт ознак може призвести до перенавчання, тоді як занадто мала їх кількість може обмежити здатність моделі до вивчення необхідних закономірностей. Таким чином, баланс між ємністю і загальною ефективністю моделі є критично важливим для успішного застосування згорткових нейронних мереж.

2.1.3 Рекурентні нейронні мережі

Рекурентні нейронні мережі (РНМ) є спеціалізованим типом нейронних мереж, який створено для обробки послідовних даних, де ключову роль відіграють

часова залежність та контекст інформації. Головною особливістю РНМ є їхня здатність зберігати внутрішню пам'ять, що дозволяє враховувати інформацію з попередніх кроків при прийнятті рішень на поточному кроці. Це досягається за рахунок зворотного зв'язку, що дає змогу передавати інформацію від одного кроку обчислення до наступного, як це продемонстровано на рисунку 2.1.3.1. Тут показано, як нейрон «А» приймає вхідне значення x_t та повертає вихідне значення h_t , яке разом із прихованим станом використовується на наступних етапах.

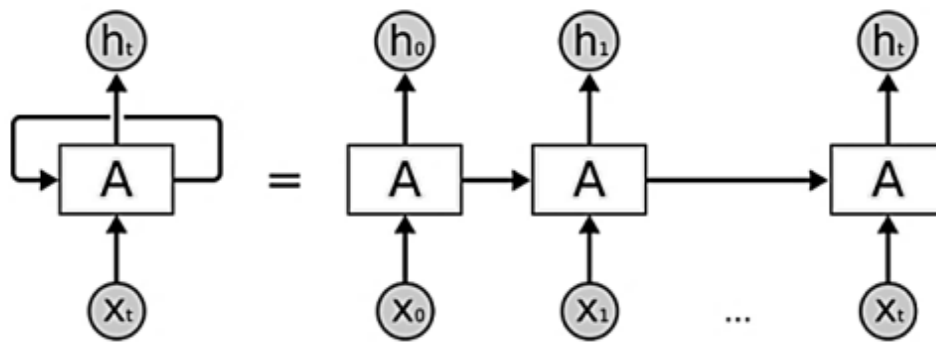


Рисунок 2.1.3.1 – Рекурентна мережа в розгортці

Основна концепція роботи РНМ полягає в обробці послідовностей даних поетапно. На вхід моделі подаються дані, наприклад, послідовність слів у реченні чи часові ряди, які обробляються один елемент за раз. На кожному кроці РНМ використовує вхідні дані поточного елемента та прихований стан, що передається з попереднього кроку. Цей прихований стан зберігає контекстну інформацію, дозволяючи моделі аналізувати взаємозв'язки між елементами послідовності. Завдяки цьому механізму РНМ здатні навчатися контекстуальних залежностей, які можуть тривати навіть у великих послідовностях.

Рекурентні мережі активно застосовуються у численних областях, які потребують аналізу послідовностей. В обробці природної мови вони використовуються для виконання таких завдань, як машинний переклад, текстогенерація, відповіді на запитання, аналіз настрою та інше. Окрім цього, РНМ ефективно працюють у задачах прогнозування часових рядів, обробки аудіо- та

відеосигналів, робототехніці, а також у багатьох інших сферах, де важливо враховувати часовий контекст даних.

Резервуарне обчислення є специфічним підходом у рекурентних нейронних мережах, що використовує фіксовану випадкову структуру, відому як «резервуар», для ефективної обробки тимчасових даних. Резервуар складається з вузлів, з'єднаних випадковим чином, утворюючи динамічну систему, здатну вловлювати складні часові закономірності. На відміну від традиційних рекурентних нейронних мереж, де навчання стосується усіх шарів, в резервуарному обчисленні навчання відбувається лише на вихідному рівні, що значно спрощує процес навчання. Це забезпечує високу обчислювальну ефективність завдяки фіксованій структурі резервуару, що робить його особливо вигідним для завдань, які вимагають інтенсивної пам'яті.

Резервуарні обчислення широко застосовуються в таких сферах, як прогнозування часових рядів і розпізнавання мовлення, продемонструвавши свою ефективність у обробці послідовних даних. Серед типів таких мереж можна виокремити мережі з відлунням стану (Echo State Networks) та рідкі скінченні автомати (Liquid State Machines), які розширюють застосування резервуарного обчислення в різних контекстах. Попри численні переваги, оптимізація продуктивності резервуарного обчислення для конкретних завдань залишається складною проблемою. Однак цей підхід є перспективною архітектурою в рекурентних нейронних мережах, пропонуючи значну обчислювальну ефективність і високу майстерність у обробці часової інформації. Тривають дослідження, спрямовані на вдосконалення його можливостей і розширення сфери застосування в різних галузях. Резервуарне обчислення є важливим кроком уперед у розвитку нейронних мереж, здатним вивести обробку тимчасових даних на новий рівень.

Численні дослідники сьогодні активно використовують мережу довгої короткочасної пам'яті (ДКЧП, англ. Long Short-Term Memory, LSTM), яку було запропоновано Хохрайтером та Шмідгубером у 1997 році. Це система глибокого навчання, яка, на відміну від традиційних рекурентних нейронних мереж (RNN),

не має проблеми зникання градієнту. ДКЧП зазвичай оснащена рекурентними вентилями, що звані забувальними (англ. forget gates), які запобігають зникненню та вибуханню зворотно поширюваних похибок. Це дозволяє похибкам рухатися в зворотному напрямку через необмежену кількість віртуальних шарів РНМ ДКЧП, що дає можливість виконувати завдання «дуже глибокого навчання» (англ. Very Deep Learning), для яких необхідне збереження інформації про події, що сталися тисячі або навіть мільйони тактів тому. Це дозволяє моделі працювати навіть за тривалих затримок та обробляти сигнали з сумішшю низько- та високочастотних складових.

На сьогодні багато застосунків використовують стопки РНМ ДКЧП, тренуючи їх за допомогою нейромережевої часової класифікації (НЧК, англ. Connectionist Temporal Classification, CTC), яка максимізує ймовірність послідовностей міток у тренувальному наборі для заданих відповідних вхідних послідовностей. НЧК дозволяє як вирівнювати, так і розпізнавати послідовності. Близько 2007 року ДКЧП розпочала революцію в розпізнаванні мовлення, перевершуючи традиційні моделі в деяких мовленнєвих застосуваннях. У 2009 році ДКЧП, тренувана НЧК, стала першою РНМ, яка перемогла на змаганнях із розпізнавання образів, вигравши кілька змагань із неперервного рукописного розпізнавання.

У 2014 році китайський пошуковий гігант Baidu застосував РНМ, тренувані НЧК, щоб перевершити еталон розпізнавання мовлення Switchboard Hub5'00, без застосування традиційних методів обробки мовлення. ДКЧП також покращила велико-словникове розпізнавання мовлення, синтез мовлення з тексту, у тому числі для Google Android, а також реалізацію фото-реалістичних голів, що розмовляють. У 2015 році в розпізнаванні мовлення Google за допомогою НЧК-тренуваної ДКЧП повідомлялося про різкий стрибок продуктивності на 49%, що було доступно через Google Voice Search для всіх користувачів смартфонів. ДКЧП також стала надзвичайно популярною в обробці природної мови. На відміну від попередніх моделей, таких як ПММ, ДКЧП може вчитися розпізнавати контекстно-чутливі мови. Вона значно покращила машинний переклад, моделювання мов та

багатомовну обробку текстів. Крім того, у поєднанні з згортковими нейронними мережами (ЗНМ), ДКЧП поліпшила автоматичний опис зображень та знайшла застосування в багатьох інших галузях.

Модель рекурентної нейронної мережі кількох масштабів часу (MTRNN, англ. Multiple Timescales Recurrent Neural Network) є обчислювальною моделлю, яка має потенціал імітувати функціонування головного мозку на нейронному рівні. Ця модель здатна відтворювати функційну ієрархію мозку через самоорганізацію, що залежить не тільки від просторових зв'язків між нейронами, але й від типів нейронної активності, кожен з яких має свої часові властивості.

Основна ідея полягає в тому, що за допомогою різних часових шкал нейронних активностей, неперервні послідовності будь-якої множини поведінки сегментуються на придатні для повторного використання примітиви. Ця концепція має біологічне підтвердження, яке було обговорено у теорії пам'яті-передбачення мозку Джеффом Гокінсом у його книзі «Про інтелект». Гокінс вважає, що мозок використовує подібну ієрархічну структуру для розуміння та передбачення світу навколо нас, де просторові і часові зв'язки між елементами інформації дозволяють ефективно обробляти та зберігати спогади. MTRNN розширює традиційні моделі РНМ, додаючи можливість працювати з різними часовими масштабами, що є важливим для складних задач, таких як прогнозування та обробка багаточасових даних, де кожен етап може мати різні часові характеристики.

Унікальна архітектура РНМ дозволяє їм вирішувати задачі, які є складними для традиційних нейронних мереж, завдяки здатності обробляти дані з динамічною структурою та тривалими залежностями. Це робить їх незамінним інструментом у багатьох сучасних технологіях і наукових дослідженнях.

2.1.4 Багатошарові нейронні мережі

Згідно з описаними типами нейронних мереж, можна зробити висновок, що для задачі розпізнавання емоцій за виразом обличчя найбільш підходить згорткова нейронна мережа (ЗНМ). Це пов'язано з тим, що ЗНМ ефективно обробляють дані, які мають просторову структуру, такі як зображення, завдяки своїй здатності

виділяти локальні ознаки, що зберігаються на різних масштабах через згорткові шари.

У багатошарових нейронних мережах нейрони об'єднані в шари, де кожен шар складається з нейронів, що мають однакові вхідні сигнали. Мережі можуть мати різну кількість шарів, і кожен шар може мати різну кількість нейронів, що не обов'язково має бути однаковим для всіх шарів. Зазвичай мережа складається з кількох типів шарів: вхідного, прихованих та вихідного. Вхідний шар приймає зовнішні сигнали та передає їх на нейрони наступного шару. Приховані шари виконують основну обробку даних. Чим більше прихованих шарів, тим складніші функції може виводити мережа. Вихідний шар дає кінцевий результат, який може бути використаний для інтерпретації або виведення.

Існують два основних типи нейронних мереж за структурою зв'язків між шарами: мережі без зворотних зв'язків та мережі зі зворотними зв'язками.

У мережах без зворотних зв'язків дані передаються від вхідного шару до вихідного шару через кілька проміжних шарів, і немає зворотних зв'язків між шарами. Це мережі прямого розповсюдження, де сигнал проходить лише в одному напрямку — від входу до виходу без циклів. Такі мережі використовуються для завдань, де немає потреби у зворотному зв'язку або збереженні інформації про попередні стани.

У мережах зі зворотними зв'язками існують зв'язки між нейронами, які не тільки передають інформацію вперед, а й дозволяють інформації повертатися від одного шару до іншого. Це створює рекурентні структури, де вихідні сигнали можуть впливати на подальшу обробку даних. Прикладом таких мереж є рекурентні нейронні мережі, які можуть бути частково-рекурентними, як, наприклад, мережі Елмана і Жордана. Мережі Елмана мають невеликі рекурентні петлі, де виходи прихованого шару передаються назад в сам же прихований шар, дозволяючи мережі зберігати інформацію про попередні стани. Мережі Жордана мають рекурентні зв'язки між вихідними та прихованими шарами, що також дає змогу зберігати історію та дозволяє моделювати тимчасову залежність. Нейронні мережі можна класифікувати також за кількістю шарів. Теоретично кількість шарів

і кількість нейронів у кожному шарі можуть бути довільними, але на практиці вони обмежені ресурсами комп'ютера або спеціалізованого апаратного забезпечення. При проектуванні нейронної мережі потрібно враховувати кілька важливих аспектів.

По-перше, чим більше нейронів і шарів, тим більші можливості мережі, але й тим складніші обчислення та більші вимоги до пам'яті. По-друге, зворотні зв'язки можуть значно збільшити можливості мережі, але водночас можуть викликати проблеми з динамічною стійкістю, що ускладнює навчання. По-третє, складність алгоритмів та використання різних типів синапсів (збуджуючі, гальмівні) можуть значно підвищити потужність мережі, але також ускладнюють її функціонування. Для кожного конкретного завдання оптимальна конфігурація мережі може бути різною.

У випадках, коли завдання можна звести до вже відомих типів задач, для яких існують оптимальні конфігурації мереж, вибір структури є менш складним. Однак, коли задача не підходить під жодну з відомих категорій, розробник змушений шукати нові конфігурації мережі, експериментуючи з кількістю шарів, нейронів та зв'язків.

Проте навіть зі значним розвитком теорії та практики штучних нейронних мереж, вибір оптимальної структури мережі для складних завдань не завжди є очевидним. Це особливо актуально для таких складних задач, як прогнозування на основі часових рядів або розпізнавання образів в умовах шуму. Саме для таких випадків рекурентні нейронні мережі (РНМ), як мережі Елмана чи Жордана, можуть бути дуже корисними, оскільки їх здатність враховувати минулий досвід через зворотні зв'язки дозволяє ефективно моделювати тимчасові залежності.

Основні переваги ЗНМ у задачах комп'ютерного зору:

- просторова організація: ЗНМ розглядають зображення в трьох вимірах — ширину, висоту та глибину. Нейрони в згорткових шарах з'єднані лише з обмеженою областю попереднього шару, що дозволяє фокусуватися на локальних ознаках.
- локальна з'єднаність: вони використовують невеликі області

зображення (рецептивне поле), що дає змогу навчати фільтри, які найкраще реагують на локальні деталі зображення, і поступово комбінують ці локальні ознаки для формування більш глобальних, складних патернів.

- спільні параметри (ваги): використання одного і того ж фільтра по всьому зображенню значно знижує кількість параметрів, що потребує менше пам'яті та покращує ефективність мережі.

- інваріантність до зсуву: завдяки повторенню фільтрів по всьому зображенню, ЗНМ можуть виявляти ознаки незалежно від їхнього положення в зоровому полі, що важливо для задач розпізнавання на зображеннях.

Натомість рекурентні нейронні мережі (РНМ), які призначені для обробки послідовних даних, таких як природна мова або часові ряди, не є найкращими для розпізнавання емоцій за допомогою зображень, оскільки вони не ефективно обробляють просторові ознаки, як це робить ЗНМ.

Багатошарові нейронні мережі (БНМ), хоч і можуть бути використані для класифікаційних задач, не мають тих специфічних властивостей для комп'ютерного зору, як ЗНМ, оскільки не мають спеціалізованих механізмів для ефективної обробки просторових даних, таких як зображення. Отже, використання ЗНМ є оптимальним для задач комп'ютерного зору, таких як розпізнавання емоцій за виразами обличчя.

2.2 Вибір набору даних

Вибір набору даних для навчання нейронної мережі для розпізнавання емоцій за виразом обличчя є надзвичайно важливим кроком, оскільки від цього залежить якість і точність навченої моделі. Як зазначено в тексті, набір даних повинен мати високу якість зображень, бути репрезентативним для різних емоцій і включати достатньо розмічених даних, що дозволяє моделі ефективно вивчати різні варіанти виразів обличчя.

Розглянуті набір даних, такі як CK+ та MMI Facial Expression Database, є важливими для дослідження емоційних виразів, але набір FER2013 і його модифікація, Face Expression Recognition Dataset, виглядають найбільш

придатними для використання в проекті. Причиною цього є великий обсяг даних (понад 35 000 зображень у FER2013) та різноманітність емоцій, що дозволяє мережі навчатись на широкому спектрі виразів обличчя. Це особливо важливо для глибоких нейронних мереж, які потребують великої кількості різноманітних даних для досягнення високої точності. Особливістю Face Expression Recognition Dataset є також точна розмітка емоцій та розділення даних на навчальні та валідаційні набори, що є важливим для подальшої перевірки ефективності моделі на невідомих даних.

Розмір зображень 48x48 пікселів є оптимальним компромісом між деталізацією та обчислювальною ефективністю, що дозволяє швидко тренувати моделі без значного навантаження на ресурси. Крім того, використання зображень облич в сірих відтінках дозволяє уникнути впливу кольорових варіацій, зосереджуючи увагу моделі на формах, контурах і текстурах обличчя, що важливо для розпізнавання емоцій за виразами. Загалом, обраний набір даних є найбільш підходящим для задачі розпізнавання емоцій за виразом обличчя завдяки своїй великій кількості зображень, різноманітності емоцій, точній розмітці та високій якості даних.

2.3 Вибір бібліотеки для виявлення обличчя на відео

У цьому розділі знову підкреслюється важливість вибору правильної бібліотеки для реалізації задачі розпізнавання емоцій за виразом обличчя, оскільки саме ця частина системи визначає точність всього процесу. Алгоритм розпізнавання емоцій можна поділити на два основні етапи: виявлення обличчя на відео та класифікація емоцій за виразом обличчя.

Перший етап — виявлення обличчя — є базовим і створює фундамент для подальшого аналізу емоцій. Вибір бібліотеки для виявлення обличчя має велике значення. Це впливає на точність та швидкість визначення обличчя на відео, а також на здатність моделі працювати в реальному часі. Вивчені варіанти — OpenCV, Dlib, TensorFlow/Keras, MTCNN — мають різні переваги та обмеження в контексті завдання розпізнавання емоцій.

Вибір OpenCV з використанням методу ознак Хаара є логічним і аргументованим, оскільки ця бібліотека проста в застосуванні і забезпечує хорошу швидкість обробки. OpenCV є однією з найбільш доступних і популярних бібліотек для виявлення обличчя, і метод ознак Хаара продемонстрував свою ефективність в багатьох додатках. Головною перевагою цього методу є його швидкість, що важливо для роботи з відео в реальному часі. Модель `haarcascade_frontalface_default`, яка входить до складу OpenCV, має достатню точність для більшості стандартних ситуацій і не потребує додаткового навчання, що спрощує її інтеграцію в систему. У таких випадках можливо виникнуть труднощі з точним виявленням обличчя. Для більш складних умов, де ці фактори можуть бути вирішальними, може бути доцільним використання інших бібліотек, таких як Dlib або MTCNN.

Ці бібліотеки мають додаткові можливості, такі як виявлення обличчя в різних орієнтаціях або під час часткового перекриття, що дозволяє вирішити проблеми, з якими можуть стикатися більш прості методи. Проте, як було зазначено в тексті, цей метод має певні обмеження. Його точність може знижуватися при поганому освітленні, значних поворотах голови або частковому перекритті обличчя. Для більш складних умов, де ці фактори можуть бути вирішальними, може бути доцільним використання інших бібліотек, таких як Dlib або MTCNN. Проте для основного завдання — виявлення обличчя в реальному часі за стандартних умов — OpenCV з методом Хаара є оптимальним варіантом (Рис. 2.3.1).

Ця різниця і буде значенням певної ознаки, визначеного розміру, певним чином позиціонованого на зображенні.

Отже, вибір OpenCV з методом ознак Хаара для виявлення обличчя є обґрунтованим, оскільки він поєднує швидкість, простоту використання та достатню точність для основних сценаріїв. Для складніших випадків можна буде розглянути інші методи або бібліотеки, щоб покращити точність виявлення обличчя в більш складних умовах.

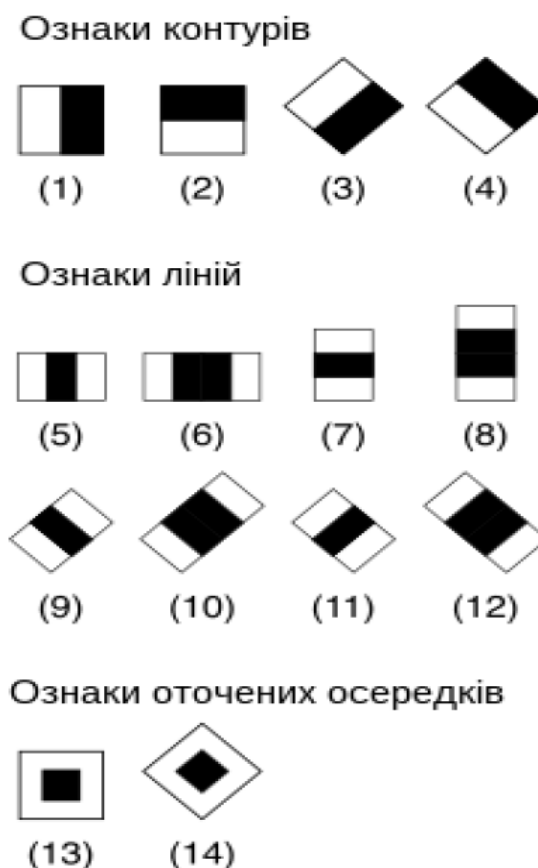


Рисунок 2.3.1 – Ознаки Хаара

2.4 Вибір алгоритму оптимізації навчання ЗНМ

Оптимізація навчання нейронної мережі — це процес налаштування ваг та параметрів моделі з метою мінімізації функції втрат та досягнення найкращої точності прогнозування.

Існує кілька популярних алгоритмів оптимізації, що використовуються для навчання нейронних мереж. Серед них можна виділити такі: градієнтний спуск, Adam, RMSprop, Adagrad.

Градієнтний спуск — це базовий алгоритм оптимізації, що використовується для навчання нейронних мереж. Його головна ідея полягає в тому, щоб знайти мінімум (або максимум) функції, рухаючись у напрямку найшвидшого спаду (або зростання) функції. Процес включає кілька основних етапів:

1. Обчислення градієнта: розрахунок часткових похідних функції втрати по відношенню до кожного параметра моделі. Градієнт визначає напрямок

найшвидшого зростання функції.

2. Оновлення ваг: коригування ваг моделі на основі градієнта та швидкості навчання. Ваги змінюються в напрямку, протилежному градієнту, для зменшення значення функції втрат.

3. Повторення: повторення попередніх кроків до досягнення критерію зупинки, наприклад, вичерпання заданої кількості епох або досягнення бажаної точності моделі (рис. 2.4).

Переваги градієнтного спуску полягають у його простоті та універсальності. Він добре працює для невеликих та середніх розмірів даних і може ефективно використовуватися для навчання нейронних мереж. Однак його недоліки включають можливість потрапляння в локальні мінімуми функції втрат, а також проблеми з розходженням при некоректному налаштуванні швидкості навчання.

Градієнтний спуск може бути вдосконалений різними варіантами:

Пакетний градієнтний спуск — весь навчальний набір переглядається на кожній ітерації для обчислення нового значення вектора ваг.

Стохастичний градієнтний спуск — на кожній ітерації випадковим чином вибирається лише один об'єкт з навчального набору даних.

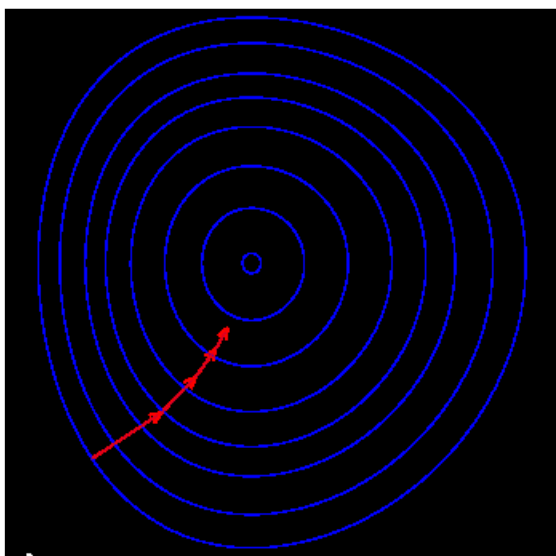


Рисунок 2.4 – Ілюстрація методу найшвидшого спуску на ряді множин рівнів

Adagrad — це адаптивний алгоритм оптимізації, який налаштовує швидкість

навчання для кожного параметра моделі на основі історії градієнтів. Його основна ідея полягає в зменшенні швидкості навчання для параметрів з великими градієнтами та збільшенні швидкості для параметрів з малими градієнтами. Цей підхід дозволяє покращити навчання моделей, особливо в задачах, де параметри мають різні масштаби або змінюються з різною швидкістю. Adagrad відомий своєю здатністю ефективно працювати із розрідженими даними, що робить його корисним для задач обробки текстів та природної мови.

Основна ідея Adagrad полягає в тому, щоб коригувати швидкість навчання окремо для кожного параметра залежно від накопиченої суми квадратів градієнтів цього параметра. Це дозволяє алгоритму автоматично зменшувати швидкість навчання для параметрів, які часто змінюються (тобто мають великі градієнти), і збільшувати її для тих, які змінюються рідко (мають малі градієнти). Таким чином, Adagrad усуває проблему фіксованої швидкості навчання, характерну для базового градієнтного спуску. Adagrad найкраще підходить для задач, де параметри моделі мають різні частоти змін, або коли дані розріджені. Він досягає цього шляхом збереження квадратичного середнього градієнтів для кожного параметра, що дозволяє оптимізувати процес навчання.

RMSprop — алгоритм оптимізації, що також адаптивно налаштовує швидкість навчання для кожного параметра моделі, подібно до Adagrad, який вдосконалює базовий метод Adagrad, зберігаючи його здатність адаптивно змінювати швидкість навчання для кожного параметра моделі, але вирішуючи проблему надмірного зменшення швидкості навчання, характерну для Adagrad.

Основна концепція RMSprop полягає у використанні експоненційно зваженого середнього квадратів градієнтів для нормалізації швидкості навчання. Замість того щоб накопичувати квадрати градієнтів усіх попередніх ітерацій, як це робить Adagrad, RMSprop враховує лише останні градієнти, зменшуючи вплив застарілих значень. Це дозволяє підтримувати збалансовану швидкість навчання навіть у довготривалих процесах оптимізації.

Однак він використовує експоненційно зважене середнє квадратів градієнтів, що дозволяє зменшити негативний вплив накопичення старих градієнтів. RMSprop

нормалізує градієнти за допомогою квадратного кореня середнього значення квадратів градієнтів, зберігаючи лише останні значення, що допомагає ефективно налаштовувати швидкість навчання.

Adam — це алгоритм оптимізації, який комбінує переваги двох популярних алгоритмів: Adagrad і RMSprop. Його основна ідея полягає в адаптації швидкості навчання для кожного параметра моделі окремо, використовуючи експоненційно зважене середнє для градієнтів та їх квадратів.

Це дозволяє ефективно оновлювати параметри, враховуючи не тільки самі значення градієнтів, а й їх змінність, що допомагає прискорити процес навчання та уникнути проблем із занадто високими або низькими темпами оновлення для різних параметрів.

Процес роботи Adam починається з обчислення градієнтів функції втрат для кожного параметра моделі. Ці градієнти використовуються для оновлення середніх значень, які є основою алгоритму.

На першому етапі обчислюється момент першого порядку, який представляє собою зважене середнє градієнтів. Це допомагає моделі визначити основний напрямок зміни параметрів, ігноруючи випадкові коливання. Другий етап полягає в обчисленні моменту другого порядку, який враховує середнє значення квадратів градієнтів. Цей крок дозволяє нормалізувати градієнти, що особливо важливо для параметрів із різними масштабами. Він адаптивно налаштовує швидкість навчання для кожного параметра моделі, враховуючи не лише масштаби градієнтів, а й їхній напрямок.

Adam ефективно працює на великих наборах даних, задачах із великою кількістю параметрів та в нейронних мережах із нерівномірними градієнтами. Це дозволяє адаптивно налаштовувати швидкість навчання в залежності від історії градієнтів. Для порівняння ефективності різних оптимізаторів розглянута робота багатошарової нейронної мережі на наборі даних MNIST. Як видно з рисунка 2.4, оптимізатор Adam має найкращі результати.

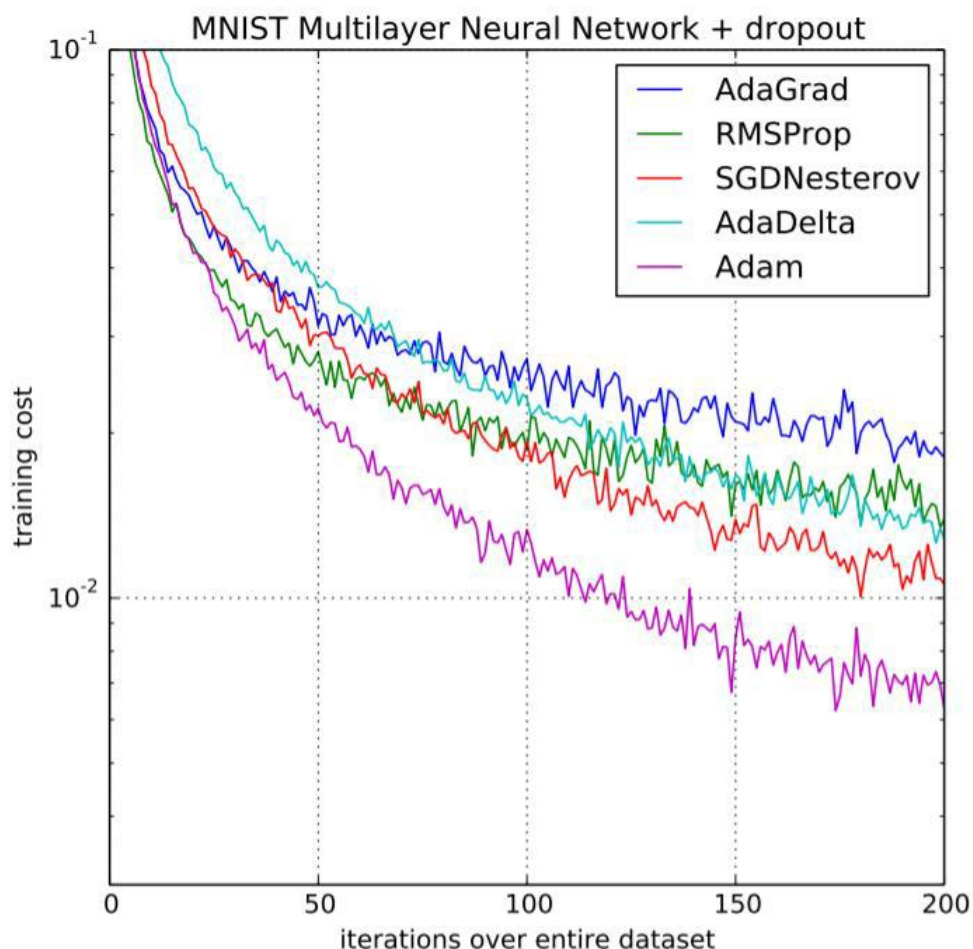


Рисунок 2.4 – Порівняння роботи оптимізаторів

Однією з найбільших переваг Adam є його здатність адаптивно налаштовувати швидкість навчання. Завдяки цьому модель може швидко збігатися до оптимального розв'язку, навіть якщо дані містять значний шум або градієнти мають нерівномірні масштаби.

Крім того, Adam добре працює на задачах, де функція втрат має складний ландшафт із великою кількістю локальних мінімумів. Алгоритм розраховує експоненційно зважене середнє градієнтів (момент першого порядку) і квадратів градієнтів (момент другого порядку). Це дає змогу моделі враховувати як величину, так і напрямок змін параметрів, а також зменшувати негативний вплив шуму. У більшості випадків стандартні значення $\beta_1 = 0.9$, $\beta_2 = 0.999$ та $\epsilon = 10^{-8}$ забезпечують хорошу продуктивність, що робить алгоритм універсальним для різних типів задач. Незважаючи на численні переваги, Adam також має деякі недоліки. Наприклад, у певних ситуаціях алгоритм може не забезпечити найкращої узагальнюваності

моделі, тобто модель може гірше працювати на тестових даних. Крім того, в задачах із дуже нерівномірними градієнтами Adam може не досягати глобального мінімуму функції втрат, зупиняючись у локальних мінімумах. Для вирішення цих проблем у практиці часто використовують модифікації Adam, такі як AdamW, який додає регуляризацію для вагових коефіцієнтів.

Проаналізувавши вище зазначені алгоритми оптимізації, для вирішення задачі розпізнавання емоцій за виразом обличчя обрано оптимізатор Adam через такі переваги:

1. Адаптивна швидкість навчання: Adam адаптує швидкість для кожного параметра окремо, що дозволяє моделі швидше досягати оптимального рішення, особливо при наявності різних масштабів градієнтів в різних частинах моделі. Менше налаштувань гіперпараметрів: на відміну від Adagrad і RMSprop, Adam має менше гіперпараметрів, що полегшує його використання і налаштування.

2. Ефективність на практиці: Adam показує стабільні результати на багатьох типах задач і допомагає уникнути проблеми розбіжності градієнтів, що може виникати з іншими оптимізаторами.

2.5 Вибір функції втрат для задачі класифікації

Функція втрат у машинному навчанні є фундаментальним елементом процесу навчання моделей, оскільки вона дозволяє оцінити, наскільки точними є прогнози моделі порівняно зі справжніми значеннями. Її основна мета полягає у вимірюванні величини помилки, яка в подальшому використовується для налаштування параметрів моделі з метою мінімізації цієї помилки. В процесі навчання модель намагається знайти такі параметри, які зводять функцію втрат до мінімуму. Це досягається за допомогою методів оптимізації, таких як градієнтний спуск, який коригує параметри на основі обчисленого значення.

Функція втрат отримує на вхід два набори даних: прогнозовані значення моделі та справжні мітки, які є еталонними. Вона обчислює числове значення, яке відображає рівень помилки між цими двома наборами. Наприклад, якщо модель передбачила ймовірність приналежності зразка до певного класу, а справжній клас

вказує на інший результат, функція втрат визначить, наскільки ці два значення різняться. Це значення використовують для оновлення параметрів моделі, щоб покращити її продуктивність на подальших етапах.

Правильний вибір функції втрат залежить від типу задачі. Наприклад, у задачах класифікації функції втрат зазвичай спрямовані на мінімізацію відстані між передбаченими ймовірностями класів і справжніми мітками. У задачах регресії функції втрат вимірюють різницю між прогнозованими і фактичними числовими значеннями. У сегментації та інших спеціалізованих задачах функції втрат адаптовані для врахування специфіки даних та вимог задачі. Для задач класифікації найбільш поширеними є такі функції втрат, як бінарна перехресна ентропія, категоріальна перехресна ентропія, зведена категоріальна перехресна ентропія та часткова категоріальна перехресна ентропія.

Бінарна перехресна ентропія використовується для задач бінарної класифікації, де кожен зразок може належати лише до одного з двох класів. Ця функція втрат вимірює відхилення між прогнозованою ймовірністю правильного класу та його справжнім значенням. Наприклад, якщо модель прогнозує ймовірність класу як 0.8 для зразка, справжній клас якого дорівнює 1, функція втрат визначить ступінь помилки, враховуючи логарифм передбаченої ймовірності. Завдяки своїй диференційованій природі, бінарна перехресна ентропія добре підходить для оптимізації за допомогою градієнтного спуску.

Категоріальна перехресна ентропія застосовується в задачах багатокласової класифікації, коли кожен зразок належить до одного з декількох можливих класів. Вона порівнює розподіл ймовірностей, який прогнозує модель, зі справжніми мітками класів, вираженими як "one-hot encoding". Метою є мінімізація розходження між цими двома розподілами, щоб модель більш точно визначала ймовірності для правильного класу. Чим точніше модель передбачає ймовірності, тим меншим буде значення функції втрат.

Зведена категоріальна перехресна ентропія є модифікованою версією категоріальної перехресної ентропії, яка враховує обчислювальну стійкість. Ця функція втрат корисна в задачах, де є ризик виникнення нульових значень або дуже

малих ймовірностей, які можуть викликати проблеми під час обчислень. Щоб уникнути цього, передбачені ймовірності зсуваються на невелику константу, наприклад, $1e^{-7}$, що дозволяє забезпечити стабільність обчислень навіть у складних сценаріях. Часткова категоріальна перехресна ентропія використовується для багатокласової класифікації, коли кожен зразок може належати одразу до кількох класів. Вона враховує ймовірності, передбачені моделлю для кожного класу, і обчислює втрату з урахуванням всіх класів, до яких може належати зразок. Це особливо корисно в задачах мультиміткової класифікації, наприклад, при аналізі текстів або розпізнаванні об'єктів на зображеннях. На основі аналізу цих функцій втрат можна зробити висновок, що для задачі розпізнавання емоцій за виразом обличчя найкраще підходить категоріальна перехресна ентропія. Ця функція втрат ідеально відповідає вимогам багатокласової класифікації, де кожен зразок належить лише до одного класу (емоції). Завдяки своїй здатності враховувати збалансованість даних та точність у багатокласових задачах, вона сприяє досягненню високої ефективності моделей.

2.6 Вибір метрики оцінки якості класифікації

Метрики оцінки якості класифікації є важливим інструментом для аналізу ефективності моделі, дозволяючи зрозуміти, наскільки точно модель виконує свої завдання. Однією з базових метрик є точність, яка вимірює відсоток правильно класифікованих екземплярів від загальної кількості. Її формула проста: точність дорівнює відношенню кількості правильно класифікованих екземплярів до загальної кількості екземплярів у наборі даних. Ця метрика добре підходить для збалансованих даних, але може бути недостатньо інформативною для наборів із сильним перекосом у кількості зразків різних класів. У таких випадках модель може демонструвати високу точність, просто ігноруючи менш представлені класи, тому доцільно використовувати додаткові метрики.

Матриця невідповідностей є ще одним популярним інструментом оцінки, який візуалізує, наскільки ефективно модель класифікує дані. Вона представляє собою квадратну таблицю, де рядки відповідають реальним класам, а стовпці –

передбаченим (Рис.2.6). Кожна комірка матриці показує кількість екземплярів, класифікованих як правильно, так і з помилками.

TN	FP	TN
FN	TP	FN
TN	FP	TN

Рисунок 2.6 – Матриця невідповідностей

Основними елементами матриці є справжньо позитивні (TP) екземпляри – це ті, які були правильно класифіковані як позитивні; справжньо негативні (TN) – екземпляри, правильно класифіковані як негативні; хибно позитивні (FP) – екземпляри, які модель помилково класифікувала як позитивні, та хибно негативні (FN) – екземпляри, які помилково класифіковані як негативні. Для більш глибокого аналізу, особливо у випадках із незбалансованими даними, застосовуються метрики влучності (precision) і повноти (recall).

Влучність визначає, який відсоток екземплярів, передбачених як позитивні, насправді є такими. Її формула виглядає так: влучність дорівнює відношенню кількості справжньо позитивних до суми справжньо позитивних і хибно позитивних. Ця метрика є важливою в тих випадках, коли хибно позитивні результати мають суттєві наслідки. Повнота, у свою чергу, визначає, який відсоток справжніх позитивних екземплярів модель змогла правильно класифікувати. Вона обчислюється як відношення справжньо позитивних до суми справжньо позитивних і хибно негативних.

Повнота є важливою, коли критично врахувати всі позитивні екземпляри, навіть якщо це призводить до збільшення кількості хибно позитивних. Щоб

врахувати баланс між влучністю і повнотою, застосовується F1-показник. Це комбінована метрика, яка обчислюється як гармонічне середнє між влучністю і повнотою.

Формула виглядає так: $F1 = 2 \times (\text{влучність} \times \text{повнота}) / (\text{влучність} + \text{повнота})$. F1-показник є особливо корисним у випадках, коли класи мають незбалансований розподіл і важливо досягти компромісу між двома основними метриками. Високе значення F1-показника свідчить про добре збалансовану модель, яка демонструє високу ефективність як у виявленні позитивних екземплярів, так і в уникненні хибних класифікацій.

Аналізуючи наведені метрики, можна зробити висновок, що вибір метрики залежить від завдання, яке вирішується, і особливостей набору даних. Для загального аналізу зазвичай використовуються точність і матриця невідповідностей, тоді як влучність, повнота і F1-показник забезпечують детальніше розуміння роботи моделі в умовах складніших завдань.

2.7 Опис вхідних даних

Вхідні дані для завдання класифікації складаються із зображень обличч людей, які розподілені за сімома класами емоцій. Кожен клас відповідає певному емоційному стану: 0 – злість, 1 – відраза, 2 – страх, 3 – щастя, 4 – спокій (нейтральний стан), 5 – смуток, 6 – здивування. Загальний набір даних уже поділено на дві частини: навчальну, що включає 28 821 зображення, і валідаційну, яка містить 7066 зображень. Це забезпечує можливість навчання моделі та перевірки її ефективності на відокремленому наборі даних. Усі зображення представлені у відтинках сірого з роздільною здатністю 48x48 пікселів. Це компактне представлення дозволяє ефективно використовувати обчислювальні ресурси при обробці даних.

На кожному зображенні обличчя займає більшу частину простору, розташоване по центру і має схожі масштаби, що зменшує потребу в додатковій попередній обробці. Така уніфікованість даних сприяє підвищенню точності навчання моделі, оскільки мінімізуються варіації, не пов'язані з виявленням

емоцій. На рис. 2.7.1 наведено приклади зображень із кожного класу емоцій. Це допомагає краще зрозуміти особливості вхідних даних.



Рисунок 2.7 – Приклад вхідних даних:

Використання цього набору даних створює хорошу основу для розробки моделей машинного навчання, орієнтованих на розпізнавання емоцій. Навчальні дані забезпечують можливість моделі виявити ключові особливості, що відображають різницю між емоціями.

Валідаційні дані, у свою чергу, дозволяють оцінити здатність моделі до узагальнення, тобто перевірити, наскільки ефективно модель працює на даних, які не використовувалися під час навчання. Перевагою цього набору є його уніфікованість. Однаковий розмір зображень, чітке розташування облич по центру та стандартизовані масштаби спрощують попередню обробку даних. Це зменшує

потребу в складних алгоритмах підготовки даних, таких як вирівнювання облич або нормалізація кольору.

Крім того, компактний розмір зображень (48x48 пікселів) дозволяє створювати моделі, які працюють швидко навіть на обмежених обчислювальних ресурсах. З точки зору практичного застосування, такі моделі можуть бути корисними у створенні систем моніторингу емоційного стану в реальному часі. Наприклад, вони можуть використовуватися для аналізу емоцій користувачів у маркетингових дослідженнях, для виявлення стресових ситуацій в освітніх чи робочих середовищах або для персоналізації інтерфейсів у розумних пристроях.

Розпізнавання емоцій також може мати значення для медичних і психологічних застосувань, таких як аналіз мімічної активності пацієнтів із розладами емоційного спектра або створення інструментів підтримки для людей із труднощами в розпізнаванні емоцій інших. У поєднанні з іншими сенсорними даними, такими як тон голосу або текстова інформація, розпізнавання емоцій із зображень може стати частиною комплексних систем аналізу настрою та поведінки.

Таким чином, цей набір даних відкриває широкі перспективи для розробки інноваційних рішень, які покращать взаємодію між людьми та технологіями. Ретельне опрацювання цих даних і створення високоточної моделі можуть значно вплинути на розвиток технологій аналізу емоцій, зробивши їх більш доступними й універсальними у використанні.

2.8 Побудова моделі

Для початку необхідно визначити тип моделі, яка буде використовуватися для розпізнавання емоцій на основі зображень облич. Для цієї задачі буде обрано Sequential модель із бібліотеки Keras. Ця модель дозволяє створювати архітектуру нейронної мережі шляхом послідовного додавання шарів. Головною перевагою такого підходу є простота: кожен шар має свій вхід та вихід, причому вихід попереднього шару слугує входом для наступного. Це забезпечує логічну й інтуїтивно зрозумілу структуру, яка підходить для вирішення задач класифікації.

Архітектура згорткової нейронної мережі (ЗНМ) моделі буде складатися з кількох компонентів:

- 4 згорткових шари (Conv2D), що відповідають за обробку зображень і виділення ознак;
- 2 повнозв'язних шари (Dense), які виконують класифікацію на основі виділених ознак.

Перший згортковий шар є шаром Conv2D із кількома параметрами: кількість фільтрів становить 64, розмір ядра (фільтра) — 3x3, параметр padding встановлений на "same", що забезпечує збереження розміру вхідних даних на виході шару, а розмір вхідних даних — 48x48x1 (це зображення в сірих відтінках). Conv2D є базовим компонентом згорткових нейронних мереж, що виконує функцію виділення ознак на зображенні. Цей шар застосовує фільтри (ядра), які проходять по вхідному зображенню і обчислюють зважену суму значень пікселів у кожній області.

Завдяки цьому модель здатна виявляти низькорівневі ознаки, такі як лінії, кути, текстури. Ці базові особливості є основою для побудови більш складних ознак, які мають значення для класифікації емоцій. Наприклад, згортка може виявляти специфічні вигини брів, форми рота або положення очей, які є індикаторами певного емоційного стану. Перший Conv2D шар зазвичай має меншу кількість фільтрів, що дозволяє йому виявляти загальні низькорівневі особливості. По мірі того, як сигнал проходить через мережу, наступні Conv2D шари можуть мати більшу кількість фільтрів, що дозволяє мережі виявляти більш складні та абстрактні ознаки, пов'язані з емоційними виразами обличчя.

Цей підхід дозволяє моделі ретельно аналізувати кожну частину зображення, включаючи текстуру, форму та орієнтацію пікселів, будувати абстрактну репрезентацію зображення, яка є придатною для подальшої класифікації, і забезпечує адаптивність до різних зображень, навіть якщо вони мають незначні відмінності в освітленні або позиціюванні. Така структура моделі дає змогу досягти високої точності у розпізнаванні емоцій і забезпечує надійність результатів.

Застосування параметра *padding="same"* в моделі має на меті збереження

однакового розміру вхідних та вихідних зображень після згортки. Це досягається завдяки додаванню додаткових пікселів, переважно нульових, навколо вхідного зображення перед виконанням операції згортки. Такий підхід дозволяє зберегти важливу інформацію, яка може бути розташована на краях зображення, і забезпечує, щоб розмір зображення не зменшувався після кожної операції згортки, що є важливим для подальшої обробки зображень в нейронній мережі.

Також в моделі застосовується Batch Normalization (нормалізація пакету), яка є технікою для стабілізації та прискорення навчання глибоких нейронних мереж. Вона допомагає зменшити внутрішню кореляцію між шарами, нормалізуючи активаційні значення на кожному шарі. Це дає змогу покращити стабільність та швидкість навчання, що в свою чергу підвищує загальну продуктивність моделі. Зазвичай нормалізація пакету застосовується перед функцією активації, що дозволяє моделі ефективно конвертувати та масштабувати вхідні дані для подальшої обробки.

Що стосується функції активації, то в даній моделі використовуватиметься ReLU (Rectified Linear Unit). ReLU є однією з найбільш популярних функцій активації в згорткових нейронних мережах. Вона активує нейрон, якщо вхідний сигнал більший за нуль, інакше значення активації буде нульовим. Ця функція дозволяє ефективно моделювати нелінійні залежності в даних і значно пришвидшує процес навчання, оскільки в ній немає складних математичних операцій, що робить її дуже швидкою для обчислень. Також у моделі використовуватиметься MaxPooling2D з агрегувальним вікном 2x2 та кроком 1 піксель.

Шар MaxPooling2D дозволяє зменшити розміри зображення, зменшуючи кількість параметрів, необхідних для навчання, але при цьому зберігаючи найважливіші ознаки зображення. Під час роботи цього шару зображення розбивається на маленькі області, і для кожної області вибирається максимальне значення пікселя, що дозволяє зберегти основні ознаки зображення при зменшенні його розміру. Для боротьби з перенавчанням використовується метод Dropout, при якому випадковим чином вимикаються певні нейрони під час навчання. Це

дозволяє моделі стати більш стійкою та менш схильною до перенавчання, оскільки зменшується залежність від конкретних нейронів або їх ваг. Крім того, метод Dropout допомагає зменшити взаємозалежність між нейронами, що може прискорити навчання та дозволяє моделі адаптуватися до більшої кількості варіацій у даних. У даній моделі встановлено Dropout на рівні 0,25, що означає, що 25% нейронів буде "вимкнено" під час кожної ітерації навчання, що сприятиме більш ефективному узагальненню результатів.

Другий згортковий шар моделі є ще одним шаром Conv2D з 128 фільтрами та розміром фільтра 5x5. Для цього шару також використовується параметр *padding="same"*, що забезпечує збереження розміру зображення після операції згортки. Після цього шару також застосовується Batch Normalization для стабілізації та прискорення навчання, функція активації ReLU, а також шар MaxPooling2D з вікном 2x2 і кроком 1 піксель для зменшення розмірів зображення. Крім того, застосовується Dropout на рівні 0,25, що допомагає уникнути перенавчання моделі. Третій та четвертий згорткові шари є однаковими. Вони також мають Conv2D з 512 фільтрами, розмір фільтра 3x3 та *padding="same"*. Ці шари використовують техніку Batch Normalization, функцію активації ReLU, та шар MaxPooling2D для зменшення розмірів зображення.

Також використовується Dropout на рівні 0,25 для боротьби з перенавчанням. Після згорткових шарів йде Flatten шар, який перетворює тривимірне зображення з попередніх шарів у одновимірний вектор, що підходить для обробки у повнозв'язних шарах. Перший повнозв'язний шар Dense складається з 256 нейронів. Кожен нейрон у цьому шарі пов'язаний з усіма нейронами попереднього шару, і кожен з них виконує лінійну комбінацію вхідних значень, яку потім пропускають через функцію активації ReLU.

На цьому етапі також застосовується Batch Normalization і Dropout на рівні 0,25 для запобігання перенавчанню. Другий повнозв'язний шар Dense складається з 512 нейронів, і після нього застосовується аналогічна процедура: Batch Normalization, функція активації ReLU, і Dropout. Останній шар є ще одним Dense шаром з 7 нейронами, де кожен нейрон відповідає за один з класів емоцій. Для

цього шару використовується функція активації Softmax, яка перетворює вихідні значення нейронів у ймовірності для кожного класу, що дозволяє моделі виконувати багатокласову класифікацію.

Таким чином, модель здійснює остаточну класифікацію емоції на основі накопиченої інформації на попередніх шарах. Архітектуру побудованої згорткової нейронної мережі можна представити у вигляді таблиці 2.8.1, де буде зазначена послідовність шарів, їхні параметри та функції активації.

Таблиця 2.8.1 – Архітектура моделі згорткової мережі

Шар (тип)	Вихідний розмір	Параметри
con2d (Conv2D)	(None, 48, 48, 64)	640
bath_normalization (BatchNormalization)	(None, 48, 48, 64)	256
activation (Activation)	(None, 48, 48, 64)	0
max_pooling2d (MaxPooling2D)	(None, 24, 24, 64)	0
dropot (Dropout)	(None, 24, 24, 64)	0
conv2d_1 (Conv2D)	(None, 24, 24, 128)	204928
bath_normalization_1 (BatchNormalization)	(None, 24, 24, 128)	512
activation_1 (Activation)	(None, 24, 24, 128)	0
max_pooling2d_1 (MaxPooling2D)	(None, 24, 24, 128)	0
dropot_1 (Dropout)	(None, 24, 24, 128)	0
conv2d_2 (Conv2D)	(None, 12, 12, 512)	590336
batch_normalization_2 (BatchNormalization)	(None, 12, 12, 512)	2048
activation_2 (Activation)	(None, 12, 12, 512)	0
max_poolng2d_2 (MaxPooling2D)	(None, 6, 6, 512)	0
dropout_2 (Dropout)	(None, 6, 6, 512)	0
conv2d_3 (Conv2D)	(None, 6, 6, 512)	2359808

Продовження таблиця 2.8.1

Шар (тип)	Вихідний розмір	Параметри
max_pooling2d_3 (MaxPooling2D)	(None, 3, 3, 512)	0
dropot_3 (Dropout)	(None, 3, 3, 512)	0
flatten (Flatten)	(None, 4608)	0
dense (Dense)	(None, 256)	1179904
batch_normalization_4 (BatchNormalization)	(None, 256)	1024
activation_4 (Activation)	(None, 256)	0
dropout_4 (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 512)	131584
batch_normalization_5 (BatchNormalization)	(None, 512)	2048
activaion_5 (Activation)	(None, 512)	0
dropout_5 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 7)	3591
Загальні параметри: 4,478,727		
Параметри, які підлягають навчанню: 4,474,759		
Параметри, які не підлягають навчанню: 3,968		

2.9 Тренування моделі

Застосовано ImageDataGenerator, який є класом з бібліотеки Keras, що спрощує завантаження, обробку та підготовку даних для тренування та валідації моделі машинного навчання (Рис. 2.9.1). Цей клас дозволяє ефективно управляти даними, забезпечуючи їх перетворення та масштабування перед подачею в модель. Завдяки ImageDataGenerator можна швидко налаштувати генерацію зображень та автоматичне їх завантаження без потреби вручну обробляти кожен файл. Метод `flow_from_directory` дозволяє зручно завантажувати зображення з відповідних директорій для навчання та валідації. Параметри, які налаштовуються в цьому методі:

- `target_size` — визначає розмір вхідного зображення, яке буде подано на вхід моделі;
- `color_mode` — встановлює режим кольору зображення (у цьому випадку "grayscale" для зображень у сірих тонах);
- `batch_size` — вказує кількість зображень, які будуть оброблятися за одну ітерацію (batch);
- `class_mode` — вказує тип класифікації. У нашому випадку встановлено значення "categorical" для багатокласової класифікації, що означає, що кожне зображення буде віднесено до одного з кількох класів;
- `shuffle` — параметр, який визначає, чи потрібно перемішувати дані перед кожною епохою тренування, щоб уникнути пристосування моделі до порядку даних.

Після налаштування генераторів даних, вони автоматично починають завантажувати і обробляти зображення, що дозволяє ефективно і безперервно тренувати модель. Використання методу `fit_generator` дозволяє навчати модель на цих згенерованих даних, що особливо корисно при обробці великих наборів даних, коли немає можливості завантажити їх усі одночасно в пам'ять. Це значно знижує навантаження на систему та забезпечує ефективне тренування моделі.

```
train_datagen = ImageDataGenerator()
validation_datagen = ImageDataGenerator()

train_generator = train_datagen.flow_from_directory(path + "train",
                                                    target_size=(size_of_picture, size_of_picture),
                                                    color_mode="grayscale",
                                                    batch_size=batch_size,
                                                    class_mode='categorical',
                                                    shuffle=True)

validation_generator = validation_datagen.flow_from_directory(path + "validation",
                                                             target_size=(size_of_picture, size_of_picture),
                                                             color_mode="grayscale",
                                                             batch_size=batch_size,
                                                             class_mode='categorical',
                                                             shuffle=False)]
```

Рисунок 2.9.1 – Метод генерації даних

Задля покращення роботи моделі, застосовано такі методи: `ModelCheckpoint`, `EarlyStopping`, `ReduceLROnPlateau` (Рис. 2.9.2).

`ModelCheckpoint` дозволяє зберігати модель з найкращою точністю на валідаційному наборі даних. Він перевіряє значення точності на валідаційному наборі (`monitor='val_acc'`) і зберігає модель в файлі "model1.h5" при кожній покращеній точності. Параметр `verbose=1` означає, що будуть виведено повідомлення про зупинку навчання, включаючи інформацію про те, яка епоха спричинила зупинку та значення точності на цій епосі. Параметр `mode='max'` гарантує, що модель буде збережена лише в разі досягнення максимального значення точності. `EarlyStopping` припиняє процес тренування, якщо значення функції втрат на валідаційному наборі (`monitor='val_loss'`) не покращується протягом заданого періоду (`patience=3`).

Крім того, він відновлює найкращі ваги моделі (`restore_best_weights=True`), щоб зберегти модель з найкращою точністю. Параметр `min_delta` визначає нижню межу, на яку метрика має змінитися, щоб вважатися покращенням. У випадку `min_delta=0`, навіть найменша негативна зміна в метриці буде вважатися погіршенням і може призвести до зупинки навчання. Це дозволяє контролювати точність покращень і уникнути передчасної зупинки навчання, якщо метрика незначно коливається в межах похибки. `ReduceLROnPlateau` автоматично знижує швидкість навчання, коли метрика (зазвичай функція втрат) не покращується протягом певної кількості епох.

Це дозволяє моделі здійснити плавну корекцію процесу навчання та може призвести до кращих результатів, особливо коли модель "застрягає" на певному рівні.

```
checkpoint = ModelCheckpoint("model1.h5", monitor='val_acc', verbose=1, save_best_only=True, mode='max')

early_stopping = EarlyStopping(monitor='val_loss',
                               min_delta=0,
                               patience=3,
                               verbose=1,
                               restore_best_weights=True)

reduce_learningrate = ReduceLROnPlateau(monitor='val_loss',
                                         factor=0.2,
                                         patience=3,
                                         verbose=1,
                                         min_delta=0.0001)

callbacks_list = [early_stopping, checkpoint, reduce_learningrate]
```

Рисунок 2.9.2 – Методи `ModelCheckpoint`, `EarlyStopping`, `ReduceLROnPlateau`

Далі відбувається компіляція моделі. У розділах 2.4-2.6 визначено, що функція втрат — категоріальна перехресна ентропія, оптимізатор — Adam (швидкість навчання — 0,0001), метрика — точність (Рис. 2.9.3).

Категоріальна перехресна ентропія використовується для багатокласової класифікації, коли кожне вхідне зображення належить до одного з кількох класів. Ця функція втрат дозволяє моделі мінімізувати розбіжність між прогнозованими й істинними мітками класів. Оптимізатор Adam є одним із найпопулярніших завдяки своїй ефективності в обчислювальних задачах та здатності адаптувати швидкість навчання для кожного параметра окремо. Швидкість навчання 0,0001 є достатньо низькою, що забезпечує стабільне навчання і дозволяє моделі поступово наближатися до оптимальних ваг. Точність як метрика дає чітке уявлення про ефективність моделі в контексті класифікації, оскільки вона визначає, скільки з прогнозованих класів співпадають з істинними мітками.

```
model.compile(loss='categorical_crossentropy', optimizer = Adam(learning_rate=0.0001), metrics=['accuracy'])
```

Рисунок 2.9.3 – Компіляція моделі

Наступним кроком відбувається навчання моделі, кількість епох — 30, кількість ітерацій — 128 (Рис. 2.9.4). В процесі навчання модель проходить через 30 епох, що дозволяє їй поступово коригувати свої ваги для мінімізації функції втрат. Кожна епоха складається з 128 ітерацій, що відповідає кількості батчів у навчальній вибірці. Кожен батч містить певну кількість зображень, для яких модель обчислює втрати, оновлює свої ваги і оптимізує параметри.

```
history = model.fit_generator(generator=train_generator,
                             steps_per_epoch=train_generator.n//train_generator.batch_size,
                             epochs=epochs,
                             validation_data = validation_generator,
                             validation_steps = validation_generator.n//validation_generator.batch_size,
                             callbacks=callbacks_list)
```

Рисунок 2.9.4 – Навчання моделі

Після закінчення навчання необхідно завантажити на комп'ютер навчену модель для подальшого застосування. Це дозволяє зберегти навчену модель у файлі та згодом використовувати її для передбачень або інтеграції в інші програми без потреби повторного навчання. Для цього використовується метод `model.save('model1.h5')`, який зберігає модель в форматі HDF5 (файл з розширенням `.h5`). Для подальшого застосування цієї моделі можна використати функцію `keras.models.load_model()`, яка дозволяє завантажити модель з файлу для використання в реальному часі. Це дає можливість оперативно використовувати модель для передбачень, не витрачаючи час на її повторне навчання. Повний програмний код, який містить усі етапи навчання, збереження і завантаження моделі, наведено в Додатку А.

Отже, для розв'язання задачі розпізнавання емоцій за виразом обличчя була обрана згортова нейронна мережа. Використаний набір даних – Face expression recognition dataset, оскільки він є найбільшим і має більш точну розмітку емоцій. У наборі представлені емоції: злість, відраза, страх, щастя, спокій (нейтральний стан), смуток та здивування.

Після аналізу існуючих бібліотек для виявлення обличчя було прийнято рішення застосувати бібліотеку OpenCV, зокрема метод Хаара (Haar Cascade), оскільки він є одним з найбільш доступних і простих у використанні. Для вирішення задачі розпізнавання емоцій обрано оптимізатор Adam, оскільки він забезпечує стабільну і швидку оптимізацію. Як функцію втрат було визначено категоріальну перехресну ентропію, яка підходить для багатокласової класифікації. Після аналізу метрик оцінки якості класифікації було обрано точність та матрицю невідповідностей.

Далі були описані вхідні дані та побудована згортова нейронна мережа (ЗНМ), яка складається з 4 згорткових шарів і 2 повнозв'язних шарів. Після цього було проведено навчання моделі, яка продемонструвала хороші результати в розпізнаванні емоцій за виразами обличчя.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ

3.1 Вибір засобів розробки

Для роботи з нейронними мережами було обрано бібліотеку TensorFlow, яка є потужною відкритою платформою для машинного навчання, створеною компанією Google. Ця бібліотека дозволяє реалізовувати моделі, що імітують принципи роботи людського мозку, такі як розпізнавання образів, аналіз даних та прогнозування. TensorFlow підтримує розподілені обчислення, що дає змогу використовувати одночасно декілька CPU або GPU, забезпечуючи високу продуктивність навіть для великих наборів даних.

В основі її роботи лежить концепція графів потоків даних, у яких вузли виконують математичні операції, а ребра передають багатовимірні масиви даних, так звані тензори. Це дає змогу паралельно обробляти інформацію та ефективно працювати з нейронними мережами різного рівня складності.

У складі TensorFlow використовується бібліотека Keras, яка полегшує побудову моделей завдяки інтуїтивному інтерфейсу та готовим модулям. Keras надає широкий вибір інструментів, таких як шари, функції активації, оптимізатори й інші компоненти, що значно пришвидшують процес розробки. Її особливістю є підтримка як стандартних нейронних мереж, так і згорткових та рекурентних моделей, які часто використовуються для роботи з зображеннями й текстом. Бібліотека Keras включає функціонал для автоматичної обробки даних, таких як їх попередня підготовка, нормалізація та аугментація. Це зменшує потребу у використанні зовнішніх інструментів для цієї мети. Keras також підтримує візуалізацію навчання моделей за допомогою TensorBoard, що дозволяє зручно відслідковувати метрики, аналізувати тренди та швидко коригувати гіперпараметри. У 2017 році Keras стала частиною TensorFlow, що зробило її інтеграцію з основною платформою ще простішою.

Основною мовою програмування було обрано Python через його зручний синтаксис та багатий набір бібліотек для машинного навчання. Python спрощує роботу з алгоритмами, дозволяючи швидко реалізовувати складні моделі з

мінімальними витратами часу. Python має величезну кількість додаткових бібліотек, таких як Scikit-learn для традиційного машинного навчання, Matplotlib і Seaborn для візуалізації, а також Jupyter Notebook, яке полегшує інтерактивну розробку, написання та тестування кодів.

Крім того, Python активно підтримується спільнотою, що забезпечує доступ до великої кількості прикладів, документації та рішень поширених проблем. Він підтримує інтеграцію з такими бібліотеками, як TensorFlow, PyTorch, NumPy та Pandas, що робить його універсальним інструментом для аналізу та обробки даних. Для тестування та експериментів використовувалась платформа Kaggle, яка надає доступ до великих наборів даних, конкурсів та обчислювальних ресурсів, включаючи GPU.

Це дозволяє ефективно навчати моделі, тестувати їхню точність та вдосконалювати алгоритми, працюючи в хмарному середовищі.

На Kaggle доступні не тільки набори даних, але й ноутбуки, створені іншими користувачами, які можна використовувати як приклади або шаблони. Крім цього, платформа регулярно проводить хакатони та конкурси з цінними призами, що стимулює участь у глобальній спільноті та обмін знаннями.

Розробка моделі проходила у середовищі Visual Studio Code, яке забезпечує зручний інтерфейс для написання коду, налагодження й управління проєктами. Завдяки великій кількості доступних розширень, таких як підтримка TensorFlow і Keras, VS Code дозволяє ефективно працювати з бібліотеками для глибокого навчання та швидко інтегрувати нові функціональні можливості у проєкт.

Visual Studio Code підтримує розробку через віддалені сервери за допомогою SSH, що дозволяє працювати з потужними обчислювальними ресурсами, такими як хмарні сервери.

Його інтеграція з Git робить управління версіями простим і зручним. Крім того, розширення Live Share дозволяє працювати над одним проєктом кільком розробникам у реальному часі, що значно полегшує командну роботу.

4 ТЕСТУВАННЯ ПРОГРАМИ

4.1 Побудова графіків та аналіз отриманих результатів

Для створення системи розпізнавання емоцій необхідно почати з підключення всіх необхідних бібліотек для роботи з комп'ютерним зором і нейронними мережами. Першим кроком завантажуються два ключових файли: `haarcascade_frontalface_default.xml`, який є каскадним класифікатором від бібліотеки OpenCV для виявлення облич (далі називатимемо його `face_classifier`), та `model1.h5`, що є навченою моделлю для класифікації емоцій (далі — `emotion_classifier`). Список емоцій, які буде розпізнавати система, задається заздалегідь у вигляді списку `emotions`. Зчитування відеопотоку з вебкамери виконується за допомогою функції `cap.read()`. Кожен кадр із відеопотоку перетворюється в зображення у відтінках сірого за допомогою функції `cv2.cvtColor()`. Це спрощує обробку даних, оскільки нейронна мережа працює зменшеним обсягом інформації. Далі, використовуючи `face_classifier`, на зображенні виявляються обличчя за допомогою методу `detectMultiScale()`. Цей метод аналізує зображення та повертає координати прямокутних областей, у яких розташовані обличчя.

Для кожного виявленого обличчя на вихідному кадрі створюється прямокутник, який обрамлює область обличчя. Виявлена область з обличчям обрізається й масштабується до розміру 48x48 пікселів, що є стандартом для роботи з даними в попередньо навченої моделі. Після цього отримане зображення нормалізується, конвертується в багатовимірний масив і передається в нейронну мережу `emotion_classifier` для передбачення емоції.

Моделі повертає ймовірності для кожної емоції зі списку `emotions`, і результат із найвищою ймовірністю використовується для визначення емоції. Ця інформація відображається над рамкою, яка обрамлює обличчя, що дозволяє користувачу одразу бачити розпізнану емоцію.

Відеопотік з результатами аналізу виводиться в окремому вікні за допомогою функції `cv2.imshow()` із назвою вікна "A system for recognizing human emotions based

on facial expressions". Система працює в реальному часі, дозволяючи динамічно аналізувати емоції, поки триває відеопотік. Повний програмний код цієї системи наведено у Додатку Б.

Навчання моделі завершилося на 28 епосі завдяки використанню методу ReduceLROnPlateau, який знижує швидкість навчання при відсутності покращення значення функції втрат на валідаційному наборі протягом визначеної кількості епох (у цьому випадку — 3). Цей механізм запобігав надмірному навчанні та витраті ресурсів, коли покращення моделі вже не відбувається. На тренування моделі за допомогою CPU було витрачено 6 годин 30 хвилин, що є типовим часом для складних нейромереж при обмежених обчислювальних ресурсах (Рис. 4.1.1).

Для пришвидшення процесу навчання було використано графічний прискорювач GPU T4, наданий платформою Kaggle. Використання GPU значно збільшило обчислювальну потужність завдяки паралельній обробці великих обсягів даних. Це дозволило скоротити час навчання до 15 хвилин (Рис. 4.1.2), що суттєво оптимізувало робочий процес. Така різниця у часі демонструє важливість застосування апаратних прискорювачів при роботі з глибоким навчанням.

```
Epoch 28: ReduceLROnPlateau reducing learning rate to 1.9999999494757503e-05.
225/225 [=====] - 806s 4s/step - loss: 0.9915 - accuracy: 0.6269 - val_loss: 1.0938 - val_accuracy: 0.5929 - lr: 1.00
00e-04
Epoch 28: early stopping
CPU times: user 21h 23min 6s, sys: 1h 13min 23s, total: 22h 36min 29s
Wall time: 6h 30min 26s
```

Рисунок 4.1.1 – Результат навчання моделі (CPU)

```
Epoch 25: ReduceLROnPlateau reducing learning rate to 1.9999999494757503e-05.
225/225 [=====] - 29s 129ms/step - loss: 1.0397 - accuracy: 0.6070 - val_loss: 1.0873 - val_accuracy:
0.5973 - lr: 1.0000e-04
Epoch 25: early stopping
CPU times: user 8min 42s, sys: 1min 3s, total: 9min 46s
Wall time: 15min 24s
```

Рисунок 4.1.2 – Результат навчання моделі (GPU)

Результати точності і втрат на навчальній і валідаційній вибірці представлено у таблиці 4.1.1.

Таблиця 4.1.1 – Результат роботи моделі

	Навчальна вибірка	Валідаційна вибірка
Точність	0.6369	0.5829
Функція втрат	0.9015	1.0838

На рисунках 4.1.3 і 4.1.4 представлені графіки втрат та точності відповідно, які ілюструють процес навчання моделі. Графік точності демонструє тенденцію до зростання, що свідчить про покращення здатності моделі коректно класифікувати дані. Навпаки, графік втрат показує спад, що вказує на зменшення похибки у передбаченнях моделі. Така взаємозалежність є типовою для навчання моделей машинного навчання: зі збільшенням точності моделі функція втрат поступово зменшується.

Ця динаміка пояснюється тим, що під час тренувального процесу модель коригує свої параметри, зменшуючи похибку передбачення на кожній ітерації. Таким чином, ці два графіки мають протилежний «характер», що наочно демонструє ефективність процесу навчання.

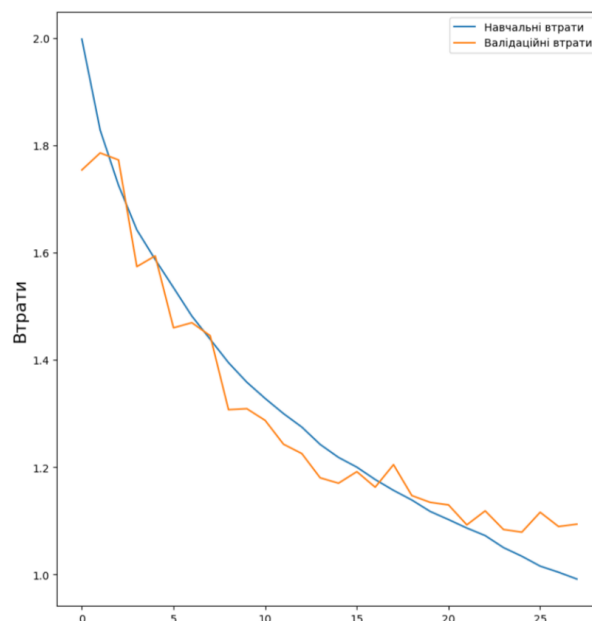


Рисунок 4.1.3 – Графік втрат

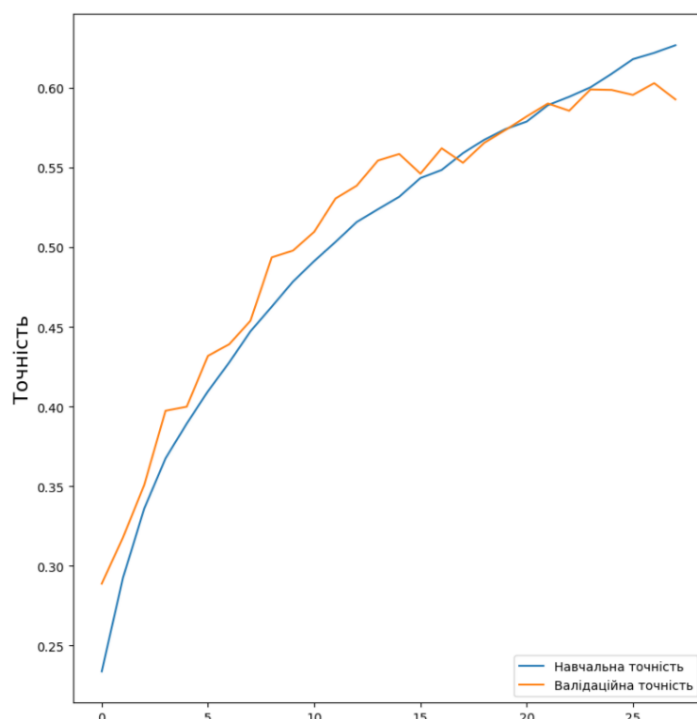


Рисунок 4.1.4 – Графік точності

Аналізуючи графік втрат, можна стверджувати, що функція втрат поступово зменшується, що вказує на успішне навчання моделі. Вчасна зупинка навчання запобігла перенавчанню, що є важливим аспектом якісного тренування нейромережі. Графік значень втрат на навчальній вибірці демонструє плавне зменшення, тоді як графік для валідаційної вибірки має характер ламаної. Це свідчить про те, що модель краще працює на навчальних даних, ніж на валідаційних, що є звичною ознакою процесу навчання.

Для глибшого аналізу роботи моделі було створено матрицю невідповідностей. Ця матриця дозволяє оцінити, які емоції модель розпізнає найкраще, а які викликають найбільші труднощі. Аналіз діагональних значень матриці показав, що модель успішно розпізнає емоції «щастя» (точність 0.8), «здивування» (0.78) і «спокій» (0.67). Водночас емоції «відраза» (0.53), «смуток» (0.51) і «злість» (0.49) розпізнаються гірше. Найменша точність спостерігається для емоції «страх» (0.22), що вказує на складність визначення цієї категорії. Ці результати демонструють, що модель добре працює з позитивними або нейтральними емоціями, але має труднощі з негативними. Це може бути зумовлено

недостатньою кількістю тренувальних даних для деяких емоцій або їх складністю для класифікації.

Отримана інформація може бути корисною для подальшої оптимізації моделі, наприклад, шляхом розширення набору даних або коригування архітектури нейромережі.

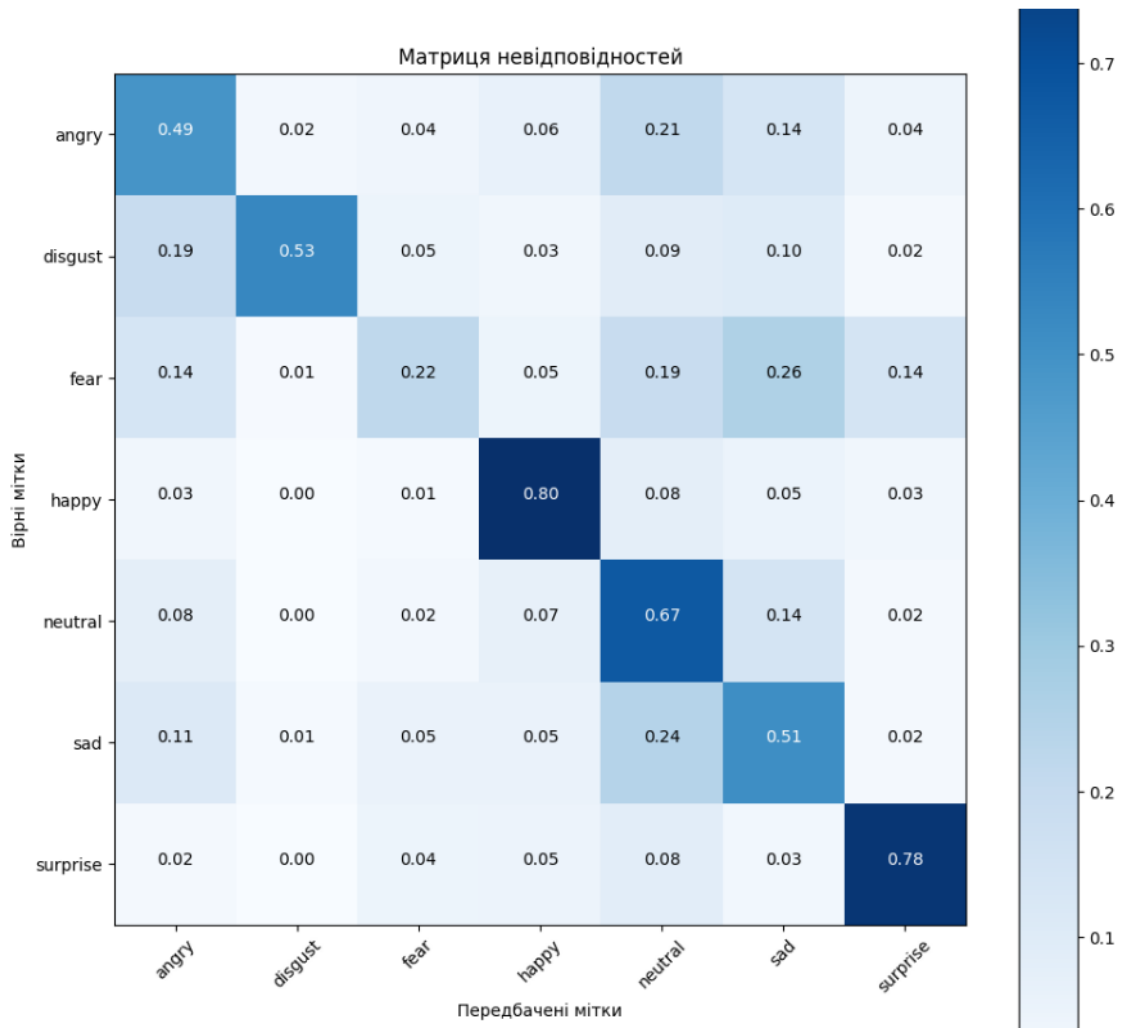


Рисунок 4.2.5 – Матриця невідповідностей

Аналіз отриманих результатів показав, що використаний набір даних FER2013 є недостатньо збалансованим. Цей набір даних, модифікована версія FER2013, відомий своєю складністю, оскільки точність, яку можуть досягти люди, складає $65 \pm 5\%$. У науковій спільноті максимальна точність, досягнута на цьому наборі, становить 75,2%. У ході проведеного дослідження вдалося досягти точності

60%, що є задовільним результатом для задачі розпізнавання емоцій, враховуючи її складність.

Для підвищення точності моделі рекомендовано збільшити кількість вхідних зображень для кожної категорії емоцій, особливо для недостатньо представлених класів, таких як «відраза». Використання методів аугментації даних, таких як зміщення, обертання, масштабування зображень, допоможе створити більш різноманітний і збалансований набір даних.

Крім того, точність можна покращити, застосувавши попередньо навчені моделі, які демонструють вищу ефективність завдяки використанню великої кількості високоякісних даних під час початкового навчання. Наприклад, моделі FER Model, EmotionNet або VGGFace можуть бути інтегровані у поточний проєкт для досягнення кращих результатів. Це дозволить покращити здатність моделі розпізнавати навіть складні для класифікації емоції.

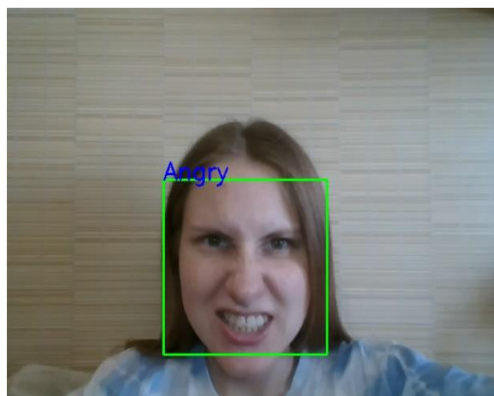
4.2 Результати тестування навченої моделі

Створена система розпізнавання емоцій здатна ідентифікувати такі емоції, як злість, відраза, страх, щастя, спокій (нейтральний вираз), смуток та здивування на основі виразу обличчя. Вона використовує зображення для аналізу емоційних змін на обличчі.

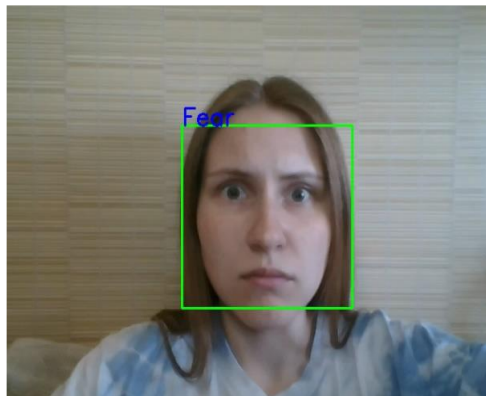
Це забезпечує високий рівень точності, дозволяючи системі визначати емоції майже миттєво, що є важливим для застосувань в області психології, людських взаємодій або розвитку персоналізованих інтерфейсів. Результати роботи цієї системи були зафіксовані на рисунку 3.3.1, який ілюструє процес розпізнавання емоцій на основі виразу обличчя, що демонструє ефективність роботи системи в практичних умовах.

Система, яка була розроблена, може бути використана в ряді сфер, таких як маркетинг, де розпізнавання емоцій допомагає зрозуміти реакції користувачів на продукцію чи послуги, або в сфері охорони здоров'я для моніторингу емоційного стану пацієнтів. Також технологія може бути застосована в навчальних і розвиваючих платформах для покращення взаємодії з користувачами, зокрема для

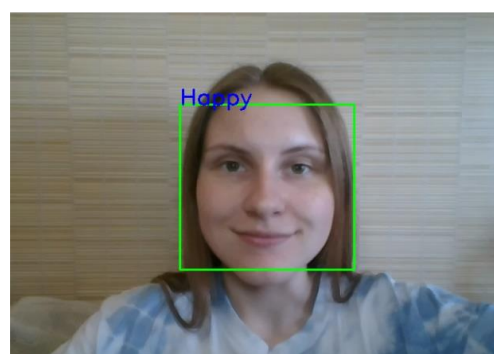
створення адаптивних систем, що враховують емоційний контекст. Завдяки цьому, подібні системи мають великий потенціал для інтеграції в різноманітні галузі, підвищуючи ефективність комунікації та взаємодії з людьми.



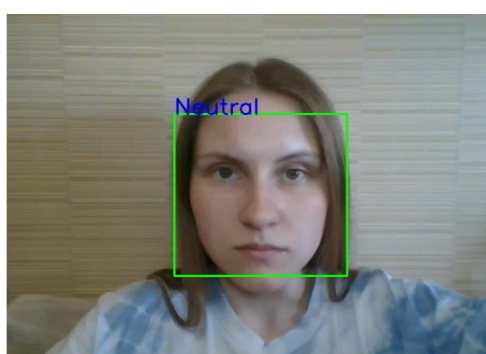
а)



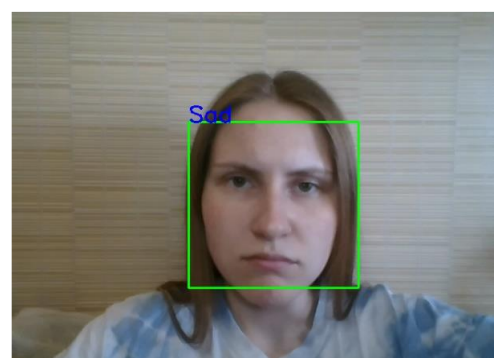
б)



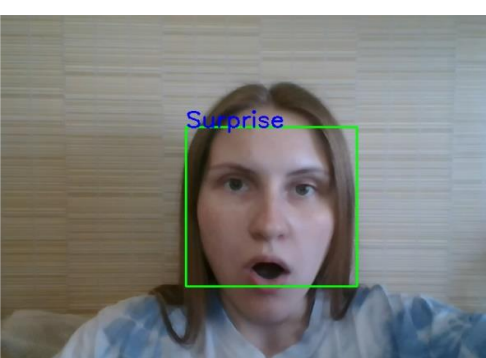
в)



г)



д)



е)

Рисунок 4.2.1 – Результати розпізнавання емоцій: а) злість б) страх в) щастя
г) спокій (нейтральний вираз) д) сум е) здивування

У результаті тестування системи було виявлено, що емоція відрази не розпізнається належним чином. Це, ймовірно, зумовлено тим, що у наборі даних для тренування моделі було обмежена кількість зображень цієї емоції (436 зображень), що є недостатнім для коректного навчання. Водночас, емоції, такі як спокій, щастя та здивування, розпізнаються точніше, оскільки вони мають більшу кількість зображень у наборі даних, і їх вирази обличчя чітко відрізняються. З іншого боку, система іноді має труднощі в розрізненні емоцій страху, злісності та смутку, оскільки їх вирази на обличчі мають схожі ознаки (положення брів, губ).

Крім того, для більш точної роботи системи необхідно забезпечити правильне освітлення в приміщенні та максимально близьке положення обличчя до камери для кращого розпізнавання.

5 ЕКОНОМІЧНА ЧАСТИНА

5.1 Комерційний та технологічний аудит науково-технічної розробки

Метою даного розділу є проведення технологічного аудиту, в даному випадку системи розпізнавання емоції людини за виразом обличчя, які в подальшому можна використати в різних сферах людського життя. Особливістю розробки є оптимізований метод згорткової нейронної мережі, яка забезпечує високу точність розпізнавання емоцій за виразом обличчя.

Вперше інтегровано поєднання бібліотеки OpenCV для виявлення облич із сучасними техніками регуляризації, такими як Dropout та Batch Normalization, для покращення узагальнюючої здатності моделі. Аналогом розробки може бути Affdex SDK від Affectiva — 40000-60000 тис. грн.

Для проведення комерційного та технологічного аудиту залучають не менше 3-х незалежних експертів.

Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням п'ятибальної системи оцінювання за 12-ма критеріями, у відповідності із табл. 5.1.

Таблиця 5.1 — Рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можлива бальна оцінка

Бали (за 5-ти бальною шкалою)					
Кри- те- рій	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах

2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
Ринкові переваги					
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практик на здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витрачати значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовую-

11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Усі дані по кожному параметру занесено в таблиці 5.2

Таблиця 4.2 — Результати оцінювання комерційного потенціалу розробки

Критерії оцінювання	ПІБ експертів		
	Експерт 1	Експерт 2	Експерт 3
	Бали		
Технічна здійсненність концепції	3	4	4
Наявність аналогів на ринку	3	3	4
Цінова політика	4	4	4
Технічні та споживчі властивості виробу	4	3	4
Експлуатаційні витрати	4	4	3
Ринок збуту	4	3	4
Конкурентоспроможність	3	4	3
Фахівці з технічної і комерційної реалізації	4	3	4
Фінансування	4	4	3
Матеріально-технічна база	3	3	3
Термін реалізації ідеї	4	4	4
Супровідна документація	4	3	4
Сума	44	42	44
Середньоарифметична сума балів	$(44+42+44) / 3 = 43,33$		

За даними таблиці 5.2 можна зробити висновок щодо рівня комерційного потенціалу даної розробки. Для цього доцільно скористатись рекомендаціями, наведеними в таблиці 5.3.

Таблиця 5.3 - Рівні комерційного потенціалу розробки

Середньоарифметична сума балів, розрахована на основі висновків	Рівень комерційного потенціалу розробки
0 - 10	Низький
11-20	Нижче середнього
21-30	Середній
31-40	Вище середнього
41-48	Високий

Як видно з таблиці, рівень комерційного потенціалу розроблюваного нового програмного продукту є високим, що досягається завдяки оптимізованому методу згорткової нейронної мережі, яка забезпечує високу точність розпізнавання емоцій за виразом обличчя. Вперше інтегровано поєднання бібліотеки OpenCV для виявлення облич із сучасними техніками регуляризації, такими як Dropout та Batch Normalization, для покращення узагальнюючої здатності моделі.

5.2 Прогнозування витрат на виконання науково-дослідної (дослідно-конструкторської) роботи

5.2.1 Основна заробітна плата розробників, яка розраховується за формулою:

$$Z_o = \frac{M}{T_p} \cdot t, \quad (5.1)$$

де M — місячний посадовий оклад конкретного розробника (дослідника), грн.;

T_p — число робочих днів за місяць, 21 днів;

t — число днів роботи розробника (дослідника).

Результати розрахунків зведемо до таблиці 4.4.

Таблиця 5.4 – Основна заробітна плата розробників

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник проекту	40000	1904,76	35	66666,667
Програміст	36000	1714,29	35	60000,000
Всього				126666,67

Так як в даному випадку розробляється програмний продукт, то розробник виступає одночасно і основним робітником, і тестувальником розроблюваного програмного продукту.

5.2.2 Додаткову заробітну плату прийнято розраховувати як 12 % від основної заробітної плати розробників та робітників:

$$З_д = З_о \cdot 12 \% / 100 \% \quad (5.2)$$

$$З_д = (126666,67 \cdot 12 \% / 100 \%) = 15200,00 \text{ (грн.)}$$

4.2.3 Згідно діючого законодавства нарахування на заробітну плату складають 22 % від суми основної та додаткової заробітної плати.

$$Н_з = (З_о + З_д) \cdot 22 \% / 100\% \quad (5.3)$$

$$Н_з = (126666,67 + 15200,00) \cdot 22 \% / 100 \% = 31210,67 \text{ (грн.)}$$

5.2.4. Оскільки для розроблювального пристрою не потрібно витратити матеріали та комплектуючі, то витрати на матеріали і комплектуючі дорівнюють нулю.

5.2.5 Амортизація обладнання, що використовувалось для розробки в спрощеному вигляді розраховується за формулою:

$$A = \frac{Ц}{T} \cdot \frac{t_{вик}}{12} \text{ [грн.]}. \quad (5.4)$$

де Ц – балансова вартість обладнання, грн.;

T – термін корисного використання обладнання згідно податкового законодавства, років

$t_{вик}$ – термін використання під час розробки, місяців

Розрахуємо, для прикладу, амортизаційні витрати на комп'ютер балансова вартість якого становить 120000 грн., термін його корисного використання згідно податкового законодавства – 2 роки, а термін його фактичного використання – 1,67 міс.

$$A_{обл} = \frac{20000}{2} \times \frac{1,67}{12} = 8333,33 \text{ грн.}$$

Аналогічно визначаємо амортизаційні витрати на інше обладнання та приміщення. Розрахунки заносимо до таблиці 4.5. Так як вартість ліцензійної ОС менше 20000 грн, то вона включається в розробку повністю, $B_{нем.ак.} = 8700$ грн.

Таблиця 5.5 – Амортизаційні відрахування на матеріальні та нематеріальні ресурси для розробників

Найменування обладнання	Балансова вартість, грн.	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн.
Комп'ютер та комп'ютерна периферія	120000	2	1,67	8333,333
Офісне обладнання (меблі)	35000	4	1,67	1215,278
Приміщення	1300000	20	1,67	9027,778
Всього				18576,39

5.2.6 Тарифи на електроенергію для непобутових споживачів (промислових підприємств) відрізняються від тарифів на електроенергію для населення. При цьому тарифи на розподіл електроенергії у різних постачальників (енергорозподільних компаній), будуть різними. Крім того, розмір тарифу залежить від класу напруги (1-й або 2-й клас). Тарифи на розподіл електроенергії для всіх енергорозподільних компаній встановлює Національна комісія з регулювання енергетики і комунальних послуг (НКРЕКП). Витрати на силову електроенергію розраховуються за формулою:

$$B_e = B \cdot \Pi \cdot \Phi \cdot K_{\Pi}, \quad (5.5)$$

де B – вартість 1 кВт-години електроенергії для 1 класу підприємства з ПДВ в 2024 році для Вінницької області за даними Енера-Вінниця, $B = 10,42$ грн./кВт;

Π – встановлена потужність обладнання, кВт. $\Pi = 0,3$ кВт;

Φ – фактична кількість годин роботи обладнання, годин.

K_{Π} – коефіцієнт використання потужності, $K_{\Pi} = 0,9$.

$$B_e = 0,9 \cdot 0,3 \cdot 8 \cdot 35 \cdot 10,42 = 787,752 \text{ (грн.)}$$

5.2.7 Інші витрати та загальновиробничі витрати.

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками. Витрати за статтею «Інші витрати» розраховуються як 50...100% від суми основної заробітної плати дослідників:

$$I_e = (Z_o + Z_p) \cdot \frac{H_{ib}}{100\%}, \quad (5.6)$$

де H_{ib} – норма нарахування за статтею «Інші витрати».

$$I_6 = 126666,67 * 80\% / 100\% = 101333,3 \text{ (грн.)}$$

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін. Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуються як 100...150% від суми основної заробітної плати дослідників:

$$H_{нзв} = (3_o + 3_p) \cdot \frac{H_{нзв}}{100\%}, \quad (5.7)$$

де $H_{нзв}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати».

$$H_{нзв} = 126666,67 * 130 \% / 100 \% = 164667 \text{ (грн.)}$$

5.2.9 Витрати на проведення науково-дослідної роботи.

Сума всіх попередніх статей витрат дає загальні витрати на проведення науково-дослідної роботи:

$$B_{заг} = 126666,67 + 15200,00 + 31210,67 + 18576,39 + 8700 + 787,75 + 101333,3 + 164667 = 467141,47 \text{ грн.}$$

5.2.11 Розрахунок загальних витрат на науково-дослідну (науково-технічну) роботу та оформлення її результатів.

Загальні витрати на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховуються за формулою:

$$3B = \frac{B_{\text{заг}}}{\eta} \text{ (грн)}, \quad (5.8)$$

де η – коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи.

Так, якщо науково-технічна розробка знаходиться на стадії: науково-дослідних робіт, то $\eta=0,1$; технічного проектування, то $\eta=0,2$; розробки конструкторської документації, то $\eta=0,3$; розробки технологій, то $\eta=0,4$; розробки дослідного зразка, то $\eta=0,5$; розробки промислового зразка, то $\eta=0,7$; впровадження, то $\eta=0,9$. Оберемо $\eta = 0,5$, так як розробка, на даний момент, знаходиться на стадії дослідного зразка:

$$3B = 467141,47 / 0,5 = 934283 \text{ грн.}$$

5.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором

В ринкових умовах узагальнювальним позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів цієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку. Саме зростання чистого прибутку забезпечить потенційному інвестору надходження додаткових коштів, дозволить покращити фінансові результати його діяльності, підвищить конкурентоспроможність та може позитивно вплинути на ухвалення рішення щодо комерціалізації цієї розробки.

Для того, щоб розрахувати можливе зростання чистого прибутку у потенційного інвестора від можливого впровадження науково-технічної розробки необхідно:

а) вказати, з якого часу можуть бути впроваджені результати науково-технічної розробки;

б) зазначити, протягом скількох років після впровадження цієї науково-технічної розробки очікуються основні позитивні результати для потенційного інвестора (наприклад, протягом 3-х років після її впровадження);

в) кількісно оцінити величину існуючого та майбутнього попиту на цю або аналогічні чи подібні науково-технічні розробки та назвати основних суб'єктів (зацікавлених осіб) цього попиту;

г) визначити ціну реалізації на ринку науково-технічних розробок з аналогічними чи подібними функціями.

При розрахунку економічної ефективності потрібно обов'язково враховувати зміну вартості грошей у часі, оскільки від вкладення інвестицій до отримання прибутку минає чимало часу. При оцінюванні ефективності інноваційних проєктів передбачається розрахунок таких важливих показників:

- абсолютного економічного ефекту (чистого дисконтованого доходу);
- внутрішньої економічної дохідності (внутрішньої норми дохідності);
- терміну окупності (дисконтованого терміну окупності).

Аналізуючи напрямки проведення науково-технічних розробок, розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором можна об'єднати, враховуючи визначені ситуації з відповідними умовами.

5.3.1 Розробка чи суттєве вдосконалення програмного засобу (програмного забезпечення, програмного продукту) для використання масовим споживачем.

В цьому випадку майбутній економічний ефект буде формуватися на основі таких даних:

$$\Delta\Pi_i = (\pm\Delta\Pi_o \cdot N + \Pi_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\rho}{100}\right), \quad (5.9)$$

де $\pm\Delta\Pi_o$ – зміна вартості програмного продукту (зростання чи зниження) від впровадження результатів науково-технічної розробки в аналізовані періоди часу;

N – кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки;

C_o – основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки, $C_o = C_{\bar{o}} \pm \Delta C_o$;

$C_{\bar{o}}$ – вартість програмного продукту у році до впровадження результатів розробки;

ΔN – збільшення кількості споживачів продукту, в аналізовані періоди часу, від покращення його певних характеристик;

λ – коефіцієнт, який враховує сплату податку на додану вартість. Ставка податку на додану вартість дорівнює 20%, а коефіцієнт $\lambda = 0,8333$.

p – коефіцієнт, який враховує рентабельність продукту;

ϑ – ставка податку на прибуток, у 2024 році $\vartheta = 18\%$.

Припустимо, що при прогнозованій ціні 85000 грн. за одиницю, термін збільшення прибутку складе 3 роки. Після завершення розробки і її вдосконалення, можна буде підняти її ціну на 1000 грн. Кількість одиниць реалізованої продукції також збільшиться: протягом першого року – на 1000 шт., протягом другого року – на 900 шт., протягом третього року на 800 шт. До моменту впровадження результатів наукової розробки реалізації продукту не було:

$$\Delta\P_1 = (0 \cdot 1000 + (85000 + 1000) \cdot 1000) \cdot 0,8333 \cdot 0,18 \cdot (1 - 0,18) = 10454999,582 \text{ грн.}$$

$$\Delta\P_2 = (0 \cdot 1000 + (85000 + 1000) \cdot (1000 + 900)) \cdot 0,8333 \cdot 0,18 \cdot (1 - 0,18) = 20098199,196 \text{ грн.}$$

$$\Delta\P_3 = (0 \cdot 1000 + (85000 + 1000) \cdot (1000 + 900 + 800)) \cdot 0,8333 \cdot 0,18 \cdot (1 - 0,18) = 28560598,858 \text{ грн.}$$

Отже, комерційний ефект від реалізації результатів розробки за три роки складе 59113797,64 грн.

4.3.2 Розрахунок ефективності вкладених інвестицій та періоду їх окупності.

Розраховуємо приведену вартість збільшення всіх чистих прибутків $ПП$, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_1^T \frac{\Delta\Pi_i}{(1+\tau)^t}, \quad (5.10)$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої науково-дослідної (науково-технічної) роботи, грн;

T – період часу, протягом якого виявляються результати впровадженої науково-дослідної (науково-технічної) роботи, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau = 0,05 \dots 0,15$;

t – період часу (в роках).

Збільшення прибутку ми отримаємо, починаючи з першого року:

$$\begin{aligned} ПП &= (10454999,582/(1+0,1)^1) + (20098199,196/(1+0,1)^2) + (28560598,858/ \\ &/ (1+0,1)^3) = 9504545,07 + 16610081,98 + 21458000,64 = 47572627,7 \text{ грн.} \end{aligned}$$

Далі розраховують величину початкових інвестицій PV , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки. Для цього можна використати формулу:

$$PV = k_{инв} * 3B, \quad (5.11)$$

де $k_{инв}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію. Це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай $k_{инв} = 2 \dots 5$, але може бути і більшим;

$3B$ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

$$PV = 2 * 934283 = 1868565,90 \text{ грн.}$$

Тоді абсолютний економічний ефект $E_{абс}$ або чистий приведений дохід (NPV , *Net Present Value*) для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{абс} = ПП - PV, \quad (5.12)$$

$$E_{абс} = 47572627,7 - 1868565,90 = 45704061,80 \text{ грн.}$$

Оскільки $E_{абс} > 0$ то вкладання коштів на виконання та впровадження результатів даної науково-дослідної (науково-технічної) роботи може бути доцільним.

Для остаточного прийняття рішення з цього питання необхідно розрахувати внутрішню економічну дохідність або показник внутрішньої норми дохідності (IRR , *Internal Rate of Return*) вкладених інвестицій та порівняти її з так званою бар'єрною ставкою дисконтування, яка визначає ту мінімальну внутрішню економічну дохідність, нижче якої інвестиції в будь-яку науково-технічну розробку вкладати буде економічно недоцільно.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій E_{ϵ} . Для цього використаємо формулу:

$$E_{\epsilon} = \sqrt[T_{ж}]{1 + \frac{E_{абс}}{PV}} - 1, \quad (5.13)$$

$T_{ж}$ – життєвий цикл наукової розробки, роки.

$$E_{\epsilon} = \sqrt[3]{(1 + 45704061,80/1868565,90) - 1} = 1,942$$

Визначимо мінімальну ставку дисконтування, яка у загальному вигляді визначається за формулою:

$$\tau = d + f, \quad (5.14)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2024 році в Україні $d = (0,09...0,14)$;

f – показник, що характеризує ризикованість вкладень; зазвичай, величина $f = (0,05...0,5)$.

$$\tau_{\min} = 0,14 + 0,05 = 0,19.$$

Так як $E_b > \tau_{\min}$, то інвестор може бути зацікавлений у фінансуванні даної наукової розробки.

Розрахуємо термін окупності вкладених у реалізацію наукового проекту інвестицій за формулою:

$$T_{ок} = \frac{1}{E_g}, \quad (5.15)$$

$$T_{ок} = 1 / 1,942 = 0,51 \text{ р.}$$

Оскільки $T_{ок} < 3$ -х років, а саме термін окупності рівний 0,51 роки, то фінансування даної наукової розробки є доцільним.

Висновки до розділу: економічна частина даної роботи містить розрахунок витрат на розробку нового програмного продукту, сума яких складає 934283 гривень. Також розраховано чистий прибуток, який може отримати виробник від реалізації нового технічного рішення, розраховано період окупності витрат для інвестора та економічний ефект при використанні даної розробки. В результаті аналізу розрахунків можна зробити висновок, що розроблений програмний продукт за ціною дешевший за аналог і є висококонкурентоспроможним. Період окупності складе близько 0,51 роки.

ВИСНОВКИ

Метою проєкту було підвищення точності розпізнавання емоцій шляхом розробки та реалізації системи, побудованої на основі згорткової нейронної мережі. Система дозволяє ідентифікувати емоції, такі як злість, відраза, страх, щастя, нейтральний вираз, смуток та здивування, працюючи в реальному часі з відео, отриманим із веб-камери.

У процесі роботи було вивчено методи розпізнавання емоцій та аналіз існуючих систем. Як основний підхід обрано глибоке навчання, що дає змогу виділяти ключові ознаки для класифікації зображень. Після аналізу різних архітектур було створено згорткову нейронну мережу з 4 згортковими та 2 повнозв'язними шарами. Для виявлення обличчя на відео застосовано модель Нааг Cascade Frontal Face, доступну в бібліотеці OpenCV.

Розробка системи виконувалася мовою Python із використанням фреймворків TensorFlow і Keras. Для створення віконного застосунку було використано середовище Visual Studio Code. Оцінка роботи системи проводилася на основі аналізу графіків точності, втрат і матриці невідповідностей.

Досягнута точність становить 60%, що є задовільним результатом для такої складної задачі. Підвищити точність можливо за рахунок збільшення обсягу навчальних даних, особливо для емоцій із меншою представленістю, таких як відраза, а також завдяки методам аугментації та використанню попередньо навчених моделей (FER Model, EmotionNet, VGGFace). Проєкт включає розроблені схеми алгоритму навчання та роботи нейронної мережі, архітектури моделі та загальної роботи системи. Створену систему можна ефективно застосовувати в різних сферах: безпека, реклама, розваги, психологія, медицина.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Емоційний AI: як технологія набуває людських рис. URL: <https://evergreens.com.ua/ua/articles/emotion-ai.html>.
2. Штучний інтелект. URL: https://uk.wikipedia.org/wiki/Штучний_інтелект.
3. Як розібратися з Computer Vision. URL: <https://dev.ua/news/yakrozibratisya-z-computer-vision>.
4. Комп'ютерне бачення проти машинного навчання. URL: <https://hashdork.com/uk/computer-vision-vs-machine-learning/>.
5. Machine Learning, ML. URL: <https://www.it.ua/knowledge-base/technologyinnovation/machine-learning>.
6. Машинне навчання простими словами. Частина 1. URL: <http://www.mmf.lnu.edu.ua/ar/1739>.
7. Баган Т. Г. Методи машинного навчання при проектуванні автоматизованих систем керування. Київ: КПІ ім. Ігоря Сікорського, 2021. 28 с.
8. Штучні нейронні мережі: що це таке? URL: <https://futurum.today/shtuchni-neironni-merezhi-shcho-tse-take/>.
9. Artificial Neural Network Tutorial. URL: <https://www.javatpoint.com/artificial-neural-network>.
10. Застосування рекурентних нейронних мереж для виконання машинного перайту. URL: [http://www.dgma.donetsk.ua/science_public/science_vesnik/№1\(25E\)_2018/article/11.pdf](http://www.dgma.donetsk.ua/science_public/science_vesnik/№1(25E)_2018/article/11.pdf).
11. Субботін С. О. Нейронні мережі: теорія та практика: навч. посіб. Житомир: О. О. Євенок, 2020. 184 с.
12. Набір даних: Face expression recognition dataset. URL: <https://www.kaggle.com/datasets/jonathanoheix/face-expression-recognition-dataset>.
13. Метод градієнтного спуску для навчання моделі лінійної регресії. URL: https://stud.com.ua/139980/informatika/metod_gradiyentnogo_spusku_navchannya_modeli_liniynoyi_regresiyi.

14. Gentle Introduction to the Adam Optimization Algorithm for Deep Learning. URL: <https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/>. 15. TensorFlow. URL: <https://uk.wikipedia.org/wiki/TensorFlow>. Keras. URL: <https://uk.wikipedia.org/wiki/Keras>.

16. Функція активації нейрона ReLU. URL: <https://conf.ztu.edu.ua/wp-content/uploads/2018/05/53-1.pdf>.

17. Facial Expression Recognition with Deep Learning. URL: http://cs230.stanford.edu/projects_winter_2020/reports/32610274.pdf.

18. Р. В. Чмих. СИСТЕМА РОЗПІЗНАВАННЯ ЕМОЦІЙ ЛЮДИНИ ЗА ВИРАЗОМ ОБЛИЧЧЯ// [Електронний ресурс] / Р. В. Чмих // Матеріали LIV Всеукраїнської науково-технічної конференції факультету інформаційних технологій та комп'ютерної інженерії, Вінниця, 24-27 березня 2025 р. – Електрон. текст. дані. – 2024.

ДОДАТОК А
Технічне завдання

Міністерство освіти та науки України

Вінницький національний технічний університет

Факультет інформаційних технологій та комп'ютерної інженерії

ЗАТВЕРДЖУЮ

Завідувач кафедри ОТ
проф., д.т.н.. Азаров О.Д.

_____ 2024 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання магістерської кваліфікаційної роботи

“Система розпізнавання емоцій людини за виразом обличчя”

08-54.МКР.039.00.000 ПЗ

Керівник роботи к.т.н. доц. каф. ОТ

_____ Кадук О. В.

Студент групи 2КІ–22м

_____ Чмих Р. В.

ВНТУ 2024

1 Підстава для виконання магістерської кваліфікаційної роботи (МКР)

1.1 Розробка системи розпізнавання емоцій за виразом обличчя є актуальною через необхідність вдосконалення технологій комп'ютерного зору для аналізу людських емоцій. Використання таких систем сприяє підвищенню ефективності у сферах безпеки, медицини, реклами та освіти.

1.2 Наказ про затвердження теми МКР.

2.1 Мета роботи — створення високоточного класифікатора емоцій на основі згорткової нейронної мережі, для обробки зображень та розпізнавання емоцій.

2.2 Призначення системи полягає у забезпеченні автоматизованого розпізнавання емоцій (злість, відраза, страх, щастя, спокій, смуток, здивування) за виразом обличчя, що може бути використано для моніторингу психологічного стану, налаштування індивідуального контенту в маркетингу або для автоматизації в роботі з клієнтами.

3.1 Призначення системи — забезпечення автоматизації процесу розпізнавання емоцій для підвищення точності аналізу емоційного стану людини.

3.2 Використовувані інструменти — бібліотека OpenCV для аналізу зображень, TensorFlow і Keras для побудови нейронної мережі, Python як мова програмування.

3.3 Організація — робота системи в режимі реального часу з можливістю інтеграції з різними джерелами відео, такими як веб-камери, або іншими потоковими платформами.

3.4 Архітектура — централізована система зі згортковою нейронною мережею, що забезпечує можливість масштабування та інтеграції з існуючими інформаційними системами.

4.1 Провести аналіз існуючих методів розпізнавання емоцій за виразом обличчя та обрати оптимальну архітектуру згорткової нейронної мережі.

4.2 Розробити та налаштувати класифікатор для розпізнавання семи базових емоцій.

4.3 Здійснити обробку та аналіз зображень з використанням моделі згорткової нейронної мережі.

4.4 Провести тестування системи на відповідність точності класифікації емоцій.

4.5 Оцінити ефективність та можливість інтеграції розробленої системи в різні галузі, такі як безпека чи медицина.

5.1 Перший етап — аналіз наявних рішень, розробка концептуальної моделі.

5.2 Другий етап — проектування архітектури системи та вибір інструментів.

5.3 Третій етап — створення та навчання згорткової нейронної мережі для класифікації емоцій.

5.4 Четвертий етап — інтеграція системи.

5.5 П'ятий етап — тестування точності та аналіз результатів роботи системи.

Таблиця А.1 — Етапи МКР

№ етапу	Назва етапу	Термін виконання		Очікувані результати
		початок	кінець	
1	Аналіз існуючих технологій, огляд аналогів системи.	19.09.2024	15.10.2024	Вступ, Розділ 1
2	Визначення архітектури веб-платформи для локальної мережі меблевого підприємства.	16.10.2024	29.10.2024	Розділ 2

Продовження таблиці А.1

3	Розробка механізмів автоматизації розподілу завдань та функціоналу для внутрішніх комунікацій (схеми маршрутизації та діаграми станів проекту)	30.10.2024	07.11.2024	Розділ 2
4	Проектування бази даних	08.11.2024	19.11.2024	Розділ 3
5	Тестування ефективності та надійності платформи	20.11.2023	26.11.2023	Розділ 3
6	Підготовка економічної частини	27.11.2024	03.12.2024	Розділ 4
7	Оформлення пояснювальної записки, графічного матеріалу і презентації	04.12.2024	10.12.2024	ПЗ, графічний матеріал і презентація
8	Підготовка і підпис супроводжуючих документів, нормоконтроль та тест на плагіат	11.12.2024	15.12.2024	Оформленні документи

6 Матеріали, що подаються до захисту МКР

До захисту подаються: пояснювальна записка МКР, графічні і ілюстративні матеріали, протокол попереднього захисту МКР на кафедрі, відгук наукового керівника, відгук опонента, протоколи складання державних екзаменів, анотації до МКР українською та іноземною мовами.

7 Порядок контролю виконання та захисту МКР

Виконання етапів графічної та розрахункової документації МКР контролюється науковим керівником згідно зі встановленими термінами. Захист МКР відбувається на засіданні Екзаменаційної комісії, затвердженої наказом ректора.

8 Вимоги до оформлювання та порядок виконання МКР

а. При оформлюванні МКР використовуються:

— ДСТУ 3008: 2015 «Звіти в сфері науки і техніки. Структура та правила оформлювання»;

— ДСТУ 8302: 2015 «Бібліографічні посилання. Загальні положення та правила складання»;

— ГОСТ 2.104–2006 «Єдина система конструкторської документації. Основні написи»;

— методичні вказівки до виконання магістерських кваліфікаційних робіт зі спеціальності 123 — «Комп'ютерна інженерія»;

— документи на які посилаються у вище вказаних.

б. Порядок виконання МКР викладено в «Положення про кваліфікаційні роботи на другому (магістерському) рівні вищої освіти СУЯ ВНТУ–03.02.02 П.001.01:21

ДОДАТОК Б

Схема згорткової нейронної мережі

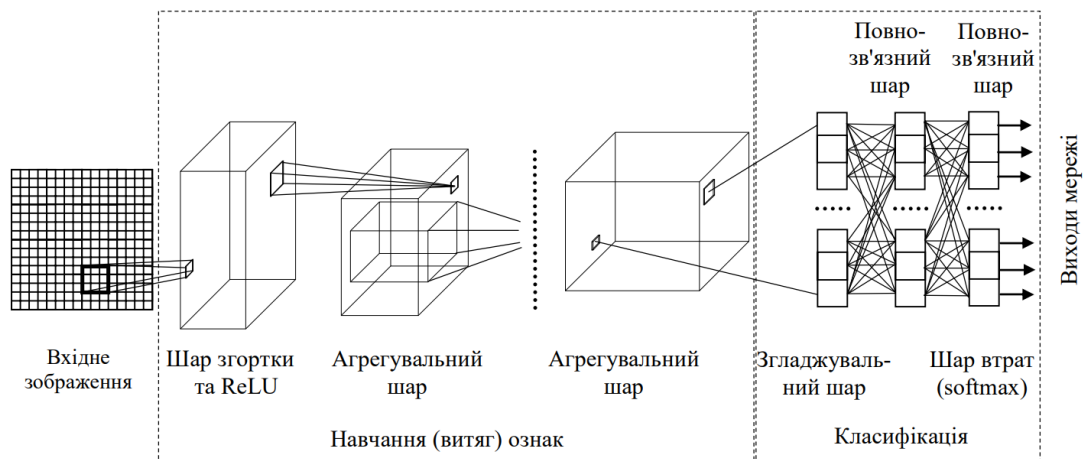


Рисунок Б.1 — Схема згорткової нейронної мережі

ДОДАТОК В

Порівняння роботи оптимізаторів

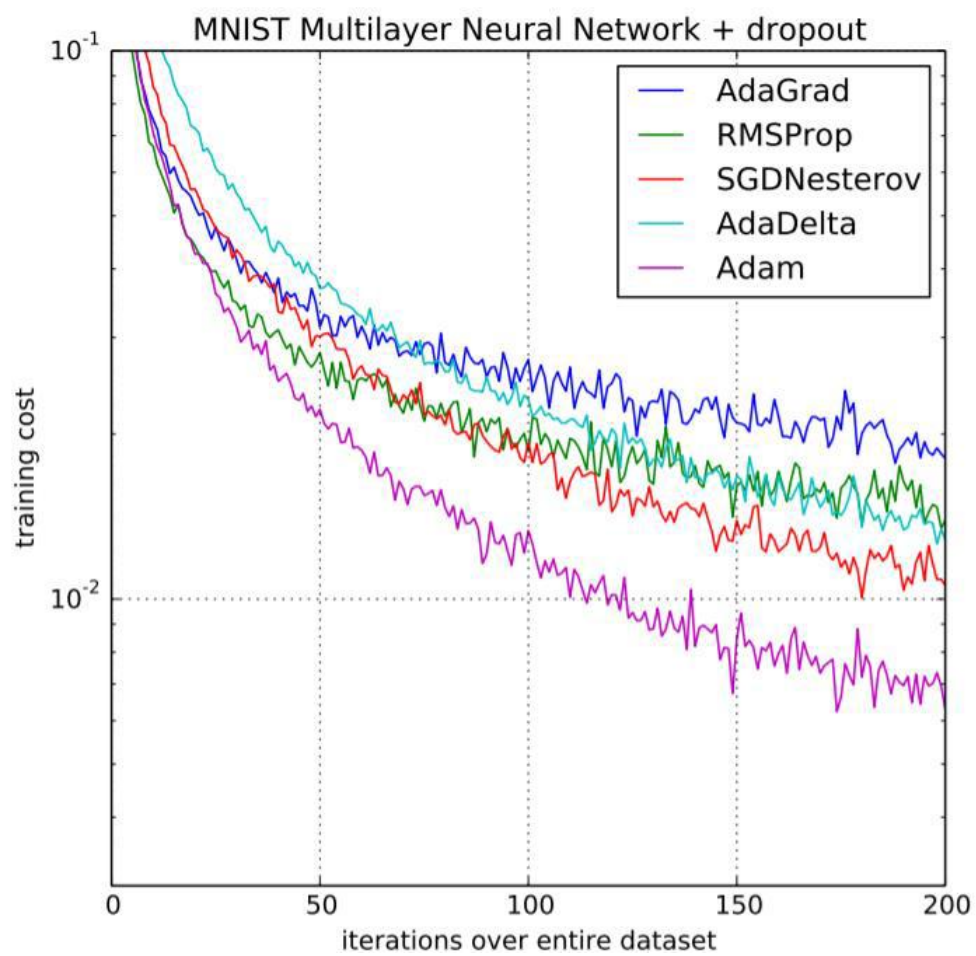


Рисунок В.1 — Порівняння роботи оптимізаторів

ДОДАТОК Г

Лістинг Main.py

```
import tensorflow
from tensorflow import keras
from keras.models import load_model
from time import sleep
from keras.preprocessing import image
from keras.preprocessing.image import img_to_array
import cv2
import numpy as np

face_classifier =
cv2.CascadeClassifier(r'D:\dyplom\haarcascade_frontalface_default.xml')
emotion_classifier = load_model(r'D:\dyplom\model1.h5')

emotions = ['Angry', 'Disgust', 'Fear', 'Happy', 'Neutral', 'Sad', 'Surprise']

cap = cv2.VideoCapture(0)

while True:
    _, frame = cap.read()
    labels = []
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    face = face_classifier.detectMultiScale(gray)
    for (x, y, w, h) in face:
        cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)
```



```

roi_gray = gray[y:y+h,x:x+w]
roi_gray = cv2.resize(roi_gray,(48,48),interpolation=cv2.INTER_AREA)
if np.sum([roi_gray])!=0:
    roi = roi_gray.astype('float')/255.0
    roi = img_to_array(roi)
    roi = np.expand_dims(roi,axis=0)
    prediction = emotion_classifier.predict(roi)[0]
    label=emotions[prediction.argmax()]
    label_position = (x,y)

cv2.putText(frame,label,label_position,cv2.FONT_HERSHEY_SIMPLEX,1,(255,0,0),
2)
else:
    cv2.putText(frame,'No
Face',(30,80),cv2.FONT_HERSHEY_SIMPLEX,1,(255,0,0),2)
cv2.imshow('A system for recognizing human emotions',frame)
if cv2.waitKey(1) & 0xFF == ord('e'):
    break
cap.release()
cv2.destroyAllWindows()

```

ДОДАТОК Д

Лістинг Model.py

```
import matplotlib.pyplot as plt, numpy as np, pandas as pd, seaborn as sns, os, itertools
from tensorflow.keras.preprocessing.image import load_img, img_to_array
from keras.preprocessing.image import ImageDataGenerator
from keras.layers import Dense, Input, Dropout, Flatten, Conv2D, BatchNormalization,
Activation, MaxPooling2D
from keras.models import Model, Sequential
from keras.optimizers import Adam
from keras.callbacks import ModelCheckpoint, EarlyStopping, ReduceLROnPlateau
from sklearn.metrics import confusion_matrix

size_of_picture = 48
path = "../input/face-expression-recognition-dataset/images/"
expression = 'fear'
plt.figure(figsize=(12, 12))
for i in range(1, 10, 1):
    plt.subplot(3, 3, i)
    img = load_img(path + "train/" + expression + "/" + os.listdir(path + "train/" +
expression)[i], target_size=(size_of_picture, size_of_picture))
    plt.imshow(img)
    plt.show()

for expression in os.listdir(path + "train"):
    print(f"{len(os.listdir(path + 'train/' + expression))} {expression} зображень")

batch_size = 128
train_datagen, validation_datagen = ImageDataGenerator(), ImageDataGenerator()
train_generator = train_datagen.flow_from_directory(path + "train",
target_size=(size_of_picture, size_of_picture), color_mode="grayscale",
batch_size=batch_size, class_mode='categorical', shuffle=True)
validation_generator = validation_datagen.flow_from_directory(path + "validation",
target_size=(size_of_picture, size_of_picture), color_mode="grayscale",
batch_size=batch_size, class_mode='categorical', shuffle=False)
classes = 7
```

```

model = Sequential([Conv2D(64, (3, 3), padding='same', input_shape=(48, 48, 1)),
BatchNormalization(), Activation('relu'), MaxPooling2D(pool_size=(2, 2)),
Dropout(0.25),
Conv2D(128, (5, 5), padding='same'), BatchNormalization(), Activation('relu'),
MaxPooling2D(pool_size=(2, 2)), Dropout(0.25),
Conv2D(512, (3, 3), padding='same'), BatchNormalization(), Activation('relu'),
MaxPooling2D(pool_size=(2, 2)), Dropout(0.25),
Conv2D(512, (3, 3), padding='same'), BatchNormalization(), Activation('relu'),
Flatten(), Dense(256), BatchNormalization(), Activation('relu'), Dropout(0.25),
Dense(512), BatchNormalization(), Activation('relu'), Dropout(0.25),
Dense(classes, activation='softmax'))])
print(model.summary())

```

```

epochs = 30
callbacks_list = [EarlyStopping(monitor='val_loss', patience=3, verbose=1,
restore_best_weights=True), ModelCheckpoint("model1.h5", monitor='val_acc',
verbose=1, save_best_only=True, mode='max'),
ReduceLROnPlateau(monitor='val_loss', factor=0.2, patience=3, verbose=1,
min_delta=0.0001)]
model.compile(loss='categorical_crossentropy',
optimizer=Adam(learning_rate=0.0001), metrics=['accuracy'])

```

```

history = model.fit_generator(generator=train_generator,
steps_per_epoch=train_generator.n // train_generator.batch_size, epochs=epochs,
validation_data=validation_generator, validation_steps=validation_generator.n //
validation_generator.batch_size, callbacks=callbacks_list)
model_json = model.to_json()
with open("model1.json", "w") as json_file:
    json_file.write(model_json)
model.save_weights("model1.h5")

```

```

plt.figure(figsize=(20, 10))
plt.subplot(1, 2, 1)
plt.ylabel('Втрати', fontsize=16)
plt.plot(history.history['loss'], label='Навчальні втрати')
plt.plot(history.history['val_loss'], label='Валідаційні втрати')
plt.subplot(1, 2, 2)
plt.ylabel('Точність', fontsize=16)
plt.plot(history.history['accuracy'], label='Навчальна точність')
plt.plot(history.history['val_accuracy'], label='Валідаційна точність')
plt.legend(loc='lower right')
plt.show()

```

```

predictions = model.predict_generator(generator=validation_generator)
y_pred = [np.argmax(probas) for probas in predictions]

```

```

y_test, class_names = validation_generator.classes,
validation_generator.class_indices.keys()

def plot_confusion_matrix(cm, classes, title='Матриця невідповідностей',
cmap=plt.cm.Blues):
cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
plt.figure(figsize=(10, 10))
plt.imshow(cm, interpolation='nearest', cmap=cmap)
plt.title(title)
plt.colorbar()
tick_marks = np.arange(len(classes))
plt.xticks(tick_marks, classes, rotation=45)
plt.yticks(tick_marks, classes)
fmt, thresh = '.2f', cm.max() / 2.
for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
plt.text(j, i, format(cm[i, j], fmt), horizontalalignment="center", color="white" if cm[i,
j] > thresh else "black")
plt.ylabel('Вірні мітки')
plt.xlabel('Передбачені мітки')
plt.tight_layout()

cnf_matrix = confusion_matrix(y_test, y_pred)
np.set_printoptions(precision=2)
plot_confusion_matrix(cnf_matrix, classes=class_names, title='Матриця
невідповідностей')
plt.show()

```

ДАДАТОК Е

ПРОТОКОЛ ПЕРЕВІРКИ НАВЧАЛЬНОЇ (КВАЛІФІКАЦІЙНОЇ) РОБОТИ

Назва роботи: _____ Система розпізнавання емоцій людини за виразом обличчя

Тип роботи: _____ Магістерська кваліфікаційна робота

(кваліфікаційна робота, курсовий проект (робота), реферат, аналітичний огляд, інше (вказати))

Підрозділ _____ кафедра обчислювальної техніки

(кафедра, факультет (інститут), навчальна група)

Керівник _____ Кадук О. В., доц. каф. ОТ _____ (прізвище, ініціали, посада)

Показники звіту подібності

Turnitin	
Оригінальність	78%
Загальна схожість	22%

Аналіз звіту подібності (відмітити потрібне)

- ☒ Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- ☐ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її автора. Роботу направити на доопрацювання.
- ☐ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Заявляю, що ознайомлений (-на) з повним звітом подібності, який був згенерований Системою щодо роботи (додається)

Автор _____ Чмих Р. В.
(підпис) (прізвище, ініціали)

Опис прийнятого рішення

Особа, відповідальна за перевірку _____ Захарченко С.М
(підпис) (прізвище, ініціали)

Експерт _____
(за потреби) (підпис) (прізвище, ініціали, посада)