

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ІНЖЕНЕРНИЙ НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ім. Ю.М. Потебні
ЗАПОРІЗЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ
КАФЕДРА ЕЛЕКТРОНІКИ, ІНФОРМАЦІЙНИХ СИСТЕМ ТА
ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Кваліфікаційна робота

другий (магістерський)

(рівень вищої освіти)

на тему: **НЕЙРОМЕРЕЖНА СИСТЕМА РОЗПІЗНАВАННЯ ЕМОЦІЙ**

Виконала: студентка 2 курсу, групи 8.1211-2іпз
спеціальності 121 Інженерія програмного

забезпечення

(код і назва спеціальності)

освітньої програми Інженерія програмного

забезпечення

(код і назва освітньої програми)

А. О. Неалова

(підпис ініціали та прізвище)

Керівник доцент, А. І. Безверхий

(посада, вчене звання, науковий ступінь, підпис, ініціали та прізвище)

Рецензент директор ТОВ Дискус П.О. Лютий

(посада, вчене звання, науковий ступінь, підпис, ініціали та прізвище)

Запоріжжя
2022

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ІНЖЕНЕРНИЙ НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ім. Ю.М. Потебні
ЗАПОРІЗЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ

Кафедра електроніки, інформаційних систем та програмного забезпечення

Рівень вищої освіти другий (магістерський)
Спеціальність 121 Інженерія програмного забезпечення
(код та назва)
Освітня програма Інженерія програмного забезпечення
(код та назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри Т.В. Критська
“ 12 ” вересня 2022 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ

Неаловій Анастасії Олександрівні
(прізвище, ім'я, по батькові)

1. Тема роботи Нейромережна система розпізнавання емоцій
керівник роботи Безверхий Анатолій Ігоревич, доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом ЗНУ від 02.06.2022 р. №597-с
2. Строк подання студентом кваліфікаційної роботи 1 грудня 2022 р.
3. Вихідні дані магістерської роботи
 - комплект нормативних документів ;
 - технічне завдання до роботи.
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
 - огляд та збір літератури стосовно теми кваліфікаційної роботи;
 - огляд та аналіз існуючих рішень та аналогів;
 - дослідження проблеми розпізнавання мов та розробка методів її вирішення;
 - створення програмного продукту та його опис;
 - перелік вимог для роботи програми;
 - дослідження поставленої проблеми та розробка висновків та пропозицій.
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
слайдів презентації

6. Консультанти розділів магістерської роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата
		Завдання прийняв

7. Дата видачі завдання 12 вересня 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської роботи	Строк виконання етапів магістерської роботи	Примітка
1	Аналіз предметної області	12.09-10.09.22	виконано
2	Формулювання основної задачі дипломної роботи та узгодження її з науковим керівником	13.09-12.09.22	виконано
3	Аналіз існуючих методів рішення	14.09-14.09.22	виконано
4	Дослідження засобів виявлення об'єктів на зображеннях	15.09-20.09.22	виконано
5	Дослідження засобів виявлення контурів об'єктів та трансформації зображень	21.09-26.09.22	виконано
6	Узгодження подальших дій з науковим керівником	27.09-28.09.22	виконано
7	Збір тренувального набору для нейронної мережі з відкритих джерел та його попередня підготовка	29.09-13.10.22	виконано
8	Навчання моделі нейронної мережі за допомогою підготовленого тренувального набору	14.10-16.10.22	виконано
9	Представлення отриманих результатів науковому керівнику та узгодження плану подальшого дослідження	17.10-19.10.22	виконано
10	Реалізація функціоналу для виявлення контурів об'єктів на зображенні та трансформації зображень	20.10-09.11.22	виконано
11	Порівняння вихідних характеристик роботи системи за допомогою вхідних тестових відео	10.11-17.11.22	виконано
12	Реалізація користувацького інтерфейсу для комп'ютерної системи	18.11-22.11.22	виконано
13	Оформлення звіту	23.11-27.11.22	виконано

Студент _____ Неалова А. О.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Безверхий А. І.
(підпис) (прізвище та ініціали)

Нормоконтроль пройдено

Нормоконтролер _____ І.А. Скрипник
(підпис) (ініціали та прізвище)

АНОТАЦІЯ

Сторінок: 92

Рисунків: 20

Джерел: 35

Неалова А. О. Нейромережна система розпізнавання емоцій: кваліфікаційна робота магістра спеціальності 121 «Інженерія програмного забезпечення» / наук. керівник А. І. Безверхий. Запоріжжя : ЗНУ, 2022. 90 с.

Мета і завдання дослідження полягають у вивченні сучасних підходів до визначення емоцій, а також у розробці комп'ютерної системи, що буде ефективно вирішувати цю задачу.

У процесі дослідження була розглянута проблема визначення емоцій людини та підходи глибинного навчання до її вирішення. Як результат, була розроблена та навчена оптимальна модель глибокої нейронної мережі. Дана мережа розпізнає емоції людей.

Ключові слова: *комп'ютерна система, нейронна мережа, визначення емоцій, згорткова нейронна мережа, глибоке навчання, епоха, Open CV.*

SUMMARY

Pages: 92

Figures: 20

Sources: 35

Nealova A. O. Neural Network System of Emotion Recognition: qualification work of the master of specialty 121 "Software Engineering" / Science head A.I. Bezverkhy. Zaporizhzhia : ZNU, 2022. 90 p.

The aim of the research is to study modern approaches to the definition of emotions, as well as to develop a computer system that will effectively solve this problem.

In the process of research, the problem of determining human emotions and deep learning approaches to its solution were considered. As a result, an optimal deep neural network model was developed and trained. This network recognizes people's emotions.

Keywords: *computer system, emotional human recognition, neural network, deep learning, Open CV.*

ЗМІСТ

ВСТУП.....	9
РОЗДІЛ 1 ДОСЛІДЖЕННЯ ПРОБЛЕМИ РОЗПІЗНАВАННЯ	18
1.1 Огляд проблеми розпізнавання емоцій.....	18
1.2 Складності у розпізнаванні емоцій	20
1.3 Загальний огляд розвитку штучного інтелекту.....	21
1.4 Підходи і напрямки до розуміння штучного інтелекту	23
1.5 Загальні поняття нейронних мереж.....	27
1.6 Класифікація зображень	29
1.7 Технології штучного інтелекту розпізнавання емоцій.....	31
1.8 Висновки з розділу 1	33
РОЗДІЛ 2 ДОСЛІДЖЕННЯ ЗАСОБІВ РОЗРОБКИ НЕЙРОННОЇ МЕРЕЖІ ДЛЯ ВИЗНАЧЕННЯ ЕМОЦІЙ.....	35
2.1 Глибокі нейронні мережі	35
2.2 Огляд структури глибоких нейронних мереж	35
2.3 Зворотне поширення помилки.....	36
2.4 Нейронна мережа прямого поширення	37
2.5 Мережа радіальних базисних функцій	38
2.6 Згорткові нейронні мережі.....	39
2.7 Структура згорткової нейронної мережі	41
2.7.1 Згортковий шар	42
2.8 Рекурсивна нейронна мережа	44

2.9	Пулінг або шар субдискретизації.....	45
2.10	Оптимізатори.....	46
2.11	Рекурентна нейронна мережа.....	47
2.12	Проблеми навчання нейронних мереж.....	47
2.12.1	Перенавчання.....	48
2.12.2	Недонавчання.....	48
2.13	Фреймворки та бібліотеки глибокого навчання.....	49
2.13.1	Theano.....	49
2.13.2	Caffe.....	49
2.13.3	Deeplearning4j.....	50
2.13.4	MatConvNet.....	50
2.13.5	TensorFlow.....	51
2.13.6	Бібліотека NumPy.....	51
2.13.7	Бібліотека Pandas.....	52
2.13.8	Бібліотека Seaborn.....	54
2.13.9	Фреймворк Keras.....	54
2.13.10	Фреймворк PyTorch.....	55
2.13.11	Бібліотека OpenCV.....	56
2.14	Розрахунок споживання пам'яті мережею.....	57
2.15	Висновки з розділу 2.....	58
РОЗДІЛ 3 РОЗРОБКА КОМП'ЮТЕРНОЇ СИСТЕМИ ДЛЯ		59
ВИЗНАЧЕННЯ ЕМОЦІЙ ЛЮДИНИ.....		59
3.2	Підключення необхідних бібліотек для роботи.....	61

3.3	Загальний опис датасету	63
3.4	Розподіл даних для навчання та перевірки	67
3.5	Розробка глибокої згорткової нейронної мережі.....	69
3.6	Висновки з розділу 3	77
РОЗДІЛ 4 ДОСЛІДЖЕННЯ РЕЗУЛЬТАТІВ РОБОТИ КОМП'ЮТЕРНОЇ СИСТЕМИ ДЛЯ ВИЗНАЧЕННЯ ПОЛОЖЕННЯ ЛЮДИНИ В ОБМЕЖЕНОМУ ПРОСТОРИ У РЕАЛЬНОМУ ЧАСІ.....		78
4.1	Результати навчання моделей нейронних мереж для розпізнавання емоцій	78
4.1.1	Запуск процесу навчання	78
4.1.2	Обчислювальний експеримент	78
4.1.3	Результати роботи нейронної мережі	81
4.2	Висновки з розділу 4	86
ВИСНОВКИ.....		88
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ		89

ВСТУП

Актуальність теми

Штучний інтелект – атрибути розумних систем, які виконують інноваційні функції, традиційно вважаються прерогативою людини; наука і технологія створення інтелектуальних машин, особливо інтелектуальних комп'ютерних програм. Штучний інтелект пов'язаний із завданням використання комп'ютерів для розуміння людського інтелекту, але він не обов'язково обмежується біологічно здійсненими методами. Поля застосування сучасних інтелектуальних систем дуже вузькі. Штучний інтелект – це здатність автоматизованих систем формалізувати та виявляти атрибути, пов'язані з поведінкою людини. Розвиток штучного інтелекту пов'язаний з науками психології, нейрофізіології, математики та інформаційних технологій. Штучний інтелект – це дуже молода наукова, цю наукову сферу заснували американські вчені нейрофізіологи. Автори займалися розробкою першої, заснованої на теорії людської мозкової діяльності нейронної моделі. Сьогодні штучний інтелект як і раніше є однією з найбільш перспективних і нерозкритих напрямків розвитку систем та технологій управління інформацією. Сьогодні в поняття штучного інтелекту входять нейронні мережі, нечітка логіка, експертні системи, комп'ютери п'ятого покоління та системи моделювання мислення.

В сучасних умовах розвитку новітніх технологій у всіх галузях і сферах діяльності людини з'явилися нові наукові напрями. Швидкого та інтенсивного піднесення за останні роки зазнала інформатика, що виросла з класу теоретичних фундаментальних дисциплін та значно розширила практичні сфери свого застосування.

Розвиток штучних нейронних мереж тісно пов'язаний з біологією. Означення штучного нейрона – це певним чином спрощена модель біологічного нейрону. Математично він представляє собою деяку нелінійну функцію від одного аргументу, що є лінійною комбінацією вхідних сигналів. Синапси – зв'язки між певними нейронами, за аналогією зі зв'язками між природними нейронами.

Нейронні мережі утворюються завдяки об'єднанню штучних нейронів.

Нейромережі мають важливу властивість, що свідчить про їх досить великі потенціали та достатньо широкі можливості – паралельна обробка інформації одночасно великою кількістю нейронів. Завдяки цьому досягається значне пришвидшення обробки інформації. Іншою є здатність до навчання та узагальнення інформації, не менш важливою особливістю нейронних мереж. Таким чином деяка схожість з роботою головного мозку людини досягається.

Такі поняття як штучний нейрон та штучна нейромережа відносно нові. В перший раз роботу штучних нейронів та її представлення моделі нейромережі було описано у статті нейрофізіолога У. Маккалоха та математика Вольтера Піттса у 1943 р.

Останнім часом спостерігається тенденція зростання інтересу до використання нейронних мереж для вирішення різних завдань і застосування їх в різних сферах людського життя.

З використанням нейромереж відкрилися можливості проведення обчислень в сферах, що до цього відносилися лише до сфери людського інтелекту. З'явилися різні можливості для створення систем, що навчилися самі вчитися, запам'ятовувати та аналізувати інформацію, що дуже схоже на розумові здібності людини.

Типовими задачами, що можуть бути вирішеними за допомогою нейронних мереж та нейрокомп'ютерів є: задача класифікації, автоматизація прогнозування, автоматизація процесу ухвалення рішень, управління, кодування і декодування інформації, розпізнавання образів та ін.

Нейронні мережі можуть використовуватися майже в усіх галузях і сферах діяльності людини: економіці, медицині, зв'язку і безпеці охоронних систем, обробці інформації.

У галузі безпеки і охоронних системах нейронні мережі необхідні для ідентифікації особи, розпізнавання голосу, осіб в натовпі, розпізнавання автомобільних номерів, аналіз аерокосмічних знімків, моніторингу інформаційних потоків, виявлення підробок.

Обробка інформації нейронні мережі можуть застосовуватися для обробки різних зображень, такі як чеки, розпізнавання підписів, відбитків пальців і голосу.

Класифікації за допомогою штучних нейромереж активно застосовуються в медицині: можна навчити нейронну мережу класифікації хвороб або симптомів, наприклад, для МРТ-діагностики.

В агробізнесі розробляється і впроваджується методика аналізу та розпізнавання зображень, при якій дані отримують від відкритих супутників, таких як LSAT, і використовують для прогнозування майбутньої врожайності конкретних земель.

Першим завданнями, яке навчилися вирішувати за допомогою нейронних мереж, стала класифікація зображень.

Системи розпізнавання облич широко застосовуються у різних сферах бізнесу: для дистанційної верифікації клієнтів фінансових установ і банків, онлайн-страхування, безконтактної ідентифікації в логістиці тощо.

Отже, в сучасному світі нейронні мережі це досить близьке майбутнє. Нейроінформатикою та дослідженнями нейромереж у різних галузях займаються науковці з усього світу у різних країнах. За допомогою штучних нейронних мереж можна опрацьовувати, аналізувати та узагальнювати інформації, що аналогічно роботі головного мозку людини. Нейронні мережі використовуються у різних сферах діяльності економіці, медицині, зв'язку, безпеці та охоронних системах, введенні та обробці інформації. Безумовно, даний перелік ще не повний, проте він дозволяє нам отримати уявлення про характер застосування нейромережевих технологій.

Таким чином, актуальним є створення нейронної мережі для розпізнавання емоцій людини.

Мета і завдання дослідження

Мета і завдання дослідження полягають у вивченні сучасних підходів до визначення емоцій людини, а також у розробці комп'ютерної системи, що буде ефективно вирішувати цю задачу.

Об'єкт дослідження

Об'єктом дослідження є набір даних з зображеннями людей з різними емоціями.

Предмет дослідження

Предметом дослідження є розпізнавання емоцій людини кадрів із камери.

Методи дослідження

Для вирішення поставленої задачі використовуються наступні методи дослідження:

1. Аналіз особливостей та існуючих рішень для проблеми визначення емоцій людини.
2. Аналіз моделей глибоких нейронних мереж для розпізнавання емоцій.
3. Аналіз бібліотек глибокого навчання та комп'ютерного зору.
4. Аналіз датасетів із зображеннями людей.
5. Аналіз методів визначення емоцій.
6. Синтез отриманих в процесі дослідження проблеми та методів її вирішення знань.
7. Експериментування з використанням різних датасетів для навчання нейронної мережі.
8. Експериментування зі зміною параметрів роботи розробленої системи.

Наукова новизна одержаних результатів

Наукова новизна одержаних результатів дослідження полягає у тому, що для вирішення задачі визначення емоцій був використаний новий та ефективний підхід, а саме: була розроблена та навчена оптимальна модель глибокої нейронної мережі для розпізнавання емоцій людини.

Практичне значення одержаних результатів

Практичне значення одержаних результатів дослідження полягає у тому, що була розроблена комп'ютерна система, яка у реальному часі визначає емоційний стан людей. Дана система є оптимальною з точки зору апаратних витрат, оскільки вона оброблює зображення з датасету на платформі Kaggle у хмарному середовищі та не потребує додаткової інфраструктури для своєї ро-

боти. З цього можна зробити висновок, що представлений у роботі підхід можливо використовувати в якості ефективного та дешевого рішення для створення подібних комп'ютерних систем.

Апробація одержаних результатів

Результати дослідження були представлені на науково-практичній конференції студентів, аспірантів, докторантів і молодих вчених Запорізького національного університету «Молода наука-2022» [1], а також на науково-технічній конференції студентів, аспірантів, магістрантів і викладачів Інженерного навчально-наукового інституту Запорізького національного університету.

Глосарій

Впевненість нейронної мережі (англ. Confidence) – це значення на вихідному шарі нейронної мережі, що визначає ймовірність приналежності вхідних даних до певного класу об'єкта.

Глибинне навчання (англ. Deep learning) – це сукупність методів машинного навчання (з вчителем, з частковим залучення вчителя, без вчителя та з підкріпленням), що ґрунтуються на навчанні ознак з даних, використовуючи багатошарові нейронні мережі.

Глибока нейронна мережа (англ. Deep neural network) – це нейронна мережа з багатьма прихованими шарами.

Згорткова нейронна мережа (англ. Convolutional neural network) – це нейронна мережа прямого поширення спеціального виду, що працює на основі фільтрів, які займаються розпізнаванням характеристик певних класів об'єктів на зображенні: прямих ліній, кривих певного типу тощо.

Метод зворотного поширення помилки (англ. Backpropagation) – це ітеративний алгоритм, що є модифікацією методу градієнтного спуску, та використовується з метою мінімізації помилки роботи багат шарової нейронної мережі.

Набір даних, датасет (англ. Dataset) – це оброблена та структурована певним чином інформація, необхідна для навчання нейронної мережі.

Помилка нейронної мережі (англ. Loss) – це значення, що визначається функцією помилки, відповіддю якої є дійсне число, що характеризує якість вирішення певної задачі нейронною мережею.

Розмір батча навчання нейронної мережі (англ. Batch size) – це гіперпараметр алгоритму навчання нейронної мережі, що визначає кількість прикладів навчальної вибірки, які повинні бути подані на вхід перед оновленням її внутрішніх параметрів.

Розпізнавання об'єктів (англ. Object detection) – це визначення емоцій людини на зображенні, а також вірогідності їхньої приналежності до того чи іншого класу.

Точність розпізнавання об'єктів (англ. Mean average precision) – це метрика для вимірювання середньої точності детекторів об'єктів, яка математично визначається як площа області під кривою графіка залежності точності (precision) від відкликання (recall).

Трансферне навчання (англ. Transfer learning) – це тип навчання нейронної мережі, що дозволяє використовувати отримані при вирішенні однієї задачі знання для розв'язку аналогічної проблеми.

Функція активації штучного нейрону (англ. Activation function) – це залежність вихідного сигналу штучного нейрону від зваженої суми його вхідних сигналів.

Штучна нейронна мережа (англ. *Artificial neural network*) – це обчислювальна система, побудована за принципом організації та функціонування біологічних нейронних мереж, що складається зі штучних нейронів та зв'язків між ними — синапсів. Вона навчається розв'язувати задачу з прикладів: анотованих зображень, оброблених текстів тощо.

Штучний нейрон (англ. *Artificial neuron*) – це спрощена модель природного нейрону, що є складовою частиною штучної нейронної мережі, та математично представлений як деяка нелінійна функція від лінійної комбінації його вхідних сигналів.

Keras – це відкритий фреймворк для глибинного навчання, розроблений з акцентом на швидке експериментування, що є надбудовою над бібліотекою TensorFlow.

OpenCV – це відкрита бібліотека комп'ютерного зору, що включає у себе алгоритми для обробки та перетворення зображень, функціонал для роботи з відео, а також модуль для використання глибоких нейронних мереж класифікації, розпізнавання, виявлення ключових точок та сегментації об'єктів на зображенні.

Python – це інтерпретована об'єктно-орієнтована мова програмування високого рівня із динамічною типізацією.

PyTorch – це відкритий фреймворк для машинного навчання, що базується на бібліотеці Torch та використовується для вирішення багатьох задач штучного інтелекту: розпізнавання об'єктів, обробка природньої мови тощо.

Qt – це кросплатформна бібліотека, що містить класи для розробки прикладного програмного забезпечення: створення графічного інтерфейсу користувача, роботи з базами даних, потоками тощо.

TensorFlow – це відкрита бібліотека для машинного навчання. Вона може бути використана для широкого кола задач, але особлива увага приділяється навчанню та використанню глибоких нейронних мереж.

РОЗДІЛ 1 ДОСЛІДЖЕННЯ ПРОБЛЕМИ РОЗПІЗНАВАННЯ

1.1 Огляд проблеми розпізнавання емоцій

Однією з найважливіших інновацій у дослідницькому світі протягом останнього десятиліття стало використання технологій розпізнавання. З середини минулого століття проблема розпізнавання об'єктів вважалася найскладнішим завданням у сфері комп'ютерного зору.

Розпізнавання образів одна із найважливіших властивостей нашої зорової системи. Це основа інших когнітивних функцій мозку, таких як рухи та соціальні взаємодії. Тому автоматизація розпізнавання образів є одним із центральних завдань у галузі обчислювальної нейрофізіології.

Візуальне сприйняття вважається найважливішим із п'яти почуттів, які ми використовуємо у повсякденному житті. Майже дві третини інформації, яку ми маємо від наших почуттів, візуального характеру. Під час візуального сприйняття ми обробляємо найширший спектр завдань: знаходимо об'єкти у зображенні, які привертають нашу увагу, розуміємо, що це за об'єкти незалежно від зміни їхнього положення, розміру, зміни освітлення або зміни кольоровості зображення; ми розпізнаємо нечіткі об'єкти та багато іншого. Людина виконує ці завдання без жодних зусиль. Проте їх надзвичайно важко змоделювати на рівні штучної системи.

Процеси сприйняття, що лежать в основі нейронних механізмів, вивчалися фахівцями обчислювальної нейрофізіології протягом багатьох десятиліть. Одним із найвідоміших першопрохідників у галузі вивчення роботи нервових клітин можна вважати Сантьяго Рамона-і-Кахала, який уперше досконально (наскільки це було можливо у 19 столітті) вивчив, зобразив та описав основні типи нервових клітин людського мозку.

Приблизно з 50-х років минулого століття вчені спробували змоделювати роботу мозку за допомогою спочатку аналогових, а потім і цифрових нейронних мереж. Поняття штучної нейронної мережі запроваджено Уорреном Маккалоком і Уолтером Піттсом в 1943 році [2]. Принципи навчання нейронних мереж були взяті спочатку з основ роботи мозку та правил побудови синапсів – зв'язків між фізичними нейронами [3]. В результаті було запропоновано ієрархічне уявлення зорової системи людини, засноване на шарах складних та простих клітин, розташованих у зорових зонах мозку. У другій половині минулого століття ця концепція ієрархій у потоці візуальної обробки вперше була використана вченими у галузі комп'ютерного зору для побудови першої теоретичної ієрархічної моделі розпізнавання об'єктів [4]. Як біологічно правдоподібні моделі, ці нейронні мережі були використані в області комп'ютерного зору протягом наступних десятиліть. Однак донедавна обчислювальна складність такого типу архітектури не дозволяла дослідникам створювати та навчати складну багаторівневу модель, засновану на даній теоретичній структурі. Перші моделі такого роду стали з'являтися лише недавно. Через складну багатопланову структуру ці архітектури були названі «глибокими», або «глибинними».

Точність розпізнавання у сучасних систем комп'ютерного зору у багатьох завданнях вища, ніж у людини [5]. Вони здатні виконувати кілька завдань одночасно: візуальні завдання розпізнавання низького рівня (наприклад, виявлення меж об'єкта), завдання середнього рівня (виділення помітних частин зображення), середнього рівня (ідентифікація об'єкта) та задачі вищого рівня (семантична сегментація).

1.2 Складності у розпізнаванні емоцій

Порівняно із завданнями розпізнавання звичайних предметів, розпізнавання людських осіб є складнішим завданням, оскільки потребує більш точних інструментів виявлення особи та її ідентифікації. Із цим пов'язано багато проблем. Крім звичайних труднощів, як то: наявність предметів, що загороджують обличчя (волосся, окуляри, медична маска, борода); повороти голови, вікові зміни тощо, існує ще ряд специфічних для розпізнавання осіб проблем, наприклад: необхідність уловлювати часом найтонші різниці між особами схожих людей уміння описувати ті класи об'єктів, які присутні у навчальній вибірці.

Всі ці причини ускладнюють завдання розпізнавання особи отримання біометричних даних.

Існує багато систем комп'ютерного зору, що служать для аналізу та розпізнавання образів. Багато методів розпізнавання були використані раніше, такі як SIFT (scale-invariant feature transform) [6], алгоритм для виявлення та опису локальних ознак у зображеннях, або гістограма спрямованих градієнтів (HOG) [7], дескриптори особливих точок, які використовуються в комп'ютерний зір для обробки зображень з метою виявлення та розпізнавання об'єкта. Традиційні підходи до розпізнавання зображень зазвичай поєднували один із цих алгоритмів з класифікатором, таким як метод опорних векторів (SVM) [8], щоб виконати завдання розпізнавання об'єкта. Тим не менш, нейромережевий підхід значно перевищує традиційні підходи.

З 2010 року ефективність різних методів розпізнавання образів почала вимірюватися в ході щорічного змагання з широкомасштабного розпізнавання образів ImageNet, масивної бази даних зображень (ILSVCR), найбільшому конкурсі, в якому різні дослідницькі групи змагалися у класифікації та розпізнаванні об'єктів та сцен. У 2012 році глибока згортова нейронна мережа досягла

максимальної в історії точності, перевершивши конкурентів на 16%. З того часу згорткові нейронні мережі за наявності достатньої кількості даних значно перевершують інші методи у всіх подібних змаганнях.

До початку 2015 року, спираючись на приклади виконання деяких завдань ILSVCR, фахівці повідомили, що комп'ютер перевищив людські можливості розпізнавання.

Аналогічний конкурс, але для розпізнавання осіб був запущений у 2015 році (MegaFace challenge). Набір даних для цього завдання включав мільйон фотографій більш ніж 690 тисяч різних людей. Алгоритми лицьової біометрії, створені на основі нейронних мереж, перевершили всі традиційні методи та виграли конкурс 2015 та 2016 років [9].

1.3 Загальний огляд розвитку штучного інтелекту

Розвиток сучасних систем штучного інтелекту розпочався з середини двадцятого століття. Програма, що започаткувала це була призначена для доведення теорем в численні. Автори, цього проєкту називають систему експертною. Робота поклала початок дослідженням в галузі штучного інтелекту, який тісно пов'язаний з розробкою програм, які розв'язують задачі на основі використання різноманітних евристичних методів. Цей етап розробки обумовив появу і розповсюдження скрізь терміну штучний інтелект. Спеціалісти в галузі штучного інтелекту завжди прагнули розробити такі програми, які розв'язувати задачі таким чином, який би вважався розумним при вирішенні цієї проблеми людиною [9]. Проблема вважається інтелектуальною, якщо алгоритм її розв'язування апіорі не відомий. На початку розвитку штучного інтелекту були спроби моделювати процес мислення людини, але ці спроби зазнали

краху. Розробити універсальні програми, як стало зрозуміло, є безперспективною справою [12]. В зв'язку з тим, що важко забезпечити універсальність програми, зосередження розробок перейшло на загальні методи і прийоми спеціальних програм. З 70-х років зусилля вчених концентрувалися на таких напрямках: розробка методів представлення, тобто способів формулювання проблем таким чином, щоб їх можна було легко вирішити; розробка методів пошуку, тобто доцільних способів управління ходом рішення завдання, щоб воно вирішувалося протягом реального часу за допомогою реальних засобів [10]. На початку восьмидесятих років було зроблено висновок, "ефективність програми вирішення задач можна сказати залежить від знань, якими вона володіє, а не від формул та схем висновків, які вона використовує". Сьогодні розвиток фундаментальних досліджень в галузі штучного інтелекту передбачає вирішення зокрема таких проблем: автоматизоване створення програмного продукту; автоматизований переклад, інформаційний пошук, генерація документів, організація природного діалогу між користувачем і комп'ютером; обробка та сприйняття природної мови та тексту; системи технічного зору та розпізнавання образів; створення баз знань; створення експертних систем [11].

Штучний інтелект (ШІ) – це така наука і технологія творення розумних машин (машин з програмним забезпеченням), здатні керувати окремими функціями інтелектуальної діяльності людини (наприклад, вибирати та приймати оптимальні рішення, спираючись на минуле досвід та раціональний аналіз зовнішніх впливів). Під іншим визначенням розуміється штучний інтелект науковий напрямок, у якому встановлюються та вирішуються завдання апаратного чи програмного моделювання [3]. Штучний інтелект дозволяє будь-яким комп'ютерам навчатися на власному досвіді, а саме обробляти великий обсяг даних, віднаходити певні закономірності; адаптуватися до заданих параметрів і виконувати багато тих завдань, які раніше були під силу лише людині.

1.4 Підходи і напрямки до розуміння штучного інтелекту

Штучний інтелект – технологія створення інтелектуальних машин, особливо інтелектуальних комп'ютерних програм. Штучний інтелект пов'язаний з подібним завданням використання комп'ютерів розуміння людського інтелекту, але з обов'язково обмежується біологічно правдоподібними методами. Точного визначення цієї науки немає, оскільки у філософії не розв'язане питання про природу і статус людського інтелекту [25]. Немає і точного критерію досягнення комп'ютером «розумності», хоча перед штучним інтелектом було запропоновано низку гіпотез, наприклад, Тест Тюринга або гіпотеза Ньюела-Саймона. Нині є безліч підходів як до розуміння задач штучний інтелект, так і до створення різних інтелектуальних систем.

Інтелект чи розум - якість психіки, що складається з здатності пристосуватися до нових ситуацій, здатності до навчання та запам'ятовування на основі досвіду, розуміння та застосування абстрактних концепцій та використання своїх знань для управління навколишнім середовищем. Інтелект - це загальна здатність до пізнання та вирішення труднощів, яка поєднує всі пізнавальні здібності людини: відчуття, сприйняття, пам'ять, уявлення, мислення, уяву.

На початку 1980-х років. вчені в галузі теорії обчислень Барр та Файгенбаум запропонували наступне визначення штучного інтелекту: «Штучний інтелект - це область інформатики, яка займається розробкою інтелектуальних комп'ютерних систем, тобто систем, які мають можливості, які ми традиційно пов'язуємо з людським розумом, - розуміння мови, навчання, здатність міркувати, вирішувати проблеми тощо.

Одна така класифікація може виділити два підходи до розробки штучного інтелекту:

1) нисхідний, семіотичний – створення символічних систем, моделюючих високорівневі психічні процеси: мислення, судження, мова, емоції, творчість і т. д.;

2) висхідний, біологічний – вивчення нейронних мереж і еволюційні обчислення, моделюючі інтелектуальну поведінку на основі більш менших «не інтелектуальних» елементів.

Ця наука пов'язана з психологією, нейрофізіологією, трансгуманізмом та іншими. Вона використовує математичний апарат, як і всі подібні комп'ютерні науки. Особливе значення для неї мають філософія і робототехніка. Штучний інтелект – дуже молода область досліджень, започаткована ще у 1956 року минулого сторіччя[13]. Історичний шлях її нам нагадує синусоїду, кожен максимум функції якої ініціювався якою-небудь новою ідеєю. На сьогодні її розвиток перебуває на мінімумі, тому що поступається застосуванню уже досягнутих результатів в інших областях науки, промисловості, бізнесі і навіть повсякденному житті [31].

Існують різні підходи до створення систем штучного інтелекту. На даний момент можна виділити чотири підходи – логічний, структурний, еволюційний та імітаційний.

Основою для вивчення логічного підходу слугує Булева алгебра. Кожна людина, що займається програмуванням знайома з нею. Свого розвитку отримала така наука як Булева алгебра у вигляді обчислення предикатів – в якому вона розширена за рахунок введення таких предметних символів та відношень між ними. Крім цього, кожна така машина має блок генерації цілі, і система виводу намагається довести дану ціль як теорему [14]. Якщо ціль досягнута, то

послідовність використаних правил дозволить отримати ланцюжок дій, необхідних для реалізації поставленої цілі (таку систему ще називають експертною системою). Потужність такої системи визначається можливостями генератора цілей і машинного доведення теорем. Нечітка логіка – новий підхід для того щоб досягти кращої виразності. Головною відмінністю цього напрямку є те, що істинність вислову може приймати окрім значень так/ні ще й проміжне значення – не знаю. Такий підхід більш подібний на мислення людини, оскільки він рідко відповідає так або ні.

Під структурним підходом ми розуміємо спроби побудови штучного інтелекту шляхом моделювання структури людського мозку. Однією з перших таких спроб був перцептрон Френка Розенблатта [19]. Головною моделюючою структурною одиницею в перцептронах (як і в більшості інших варіантах моделювання мозку) є нейрон. Пізніше виникли і інші моделі, більш відомі під назвою нейронні мережі і їхні реалізації – нейрокомп'ютери. Ці моделі відрізняються за будовою окремих нейронів, за топологією зв'язків між ними і алгоритмами навчання. Серед найбільш відомих на даний час варіантів НМ можна назвати нейронні мережі зі зворотнім розповсюдженням помилки, сітки Кохонена, сітки Хопфілда, стохастичні нейрони сітки. Відмінності між логічним та структурним підходом не стільки принципові, як це здається на перший погляд. Алгоритми спрощення і вербалізації нейронних мереж перетворюють моделі структурного підходу в явні логічні моделі [3]. З іншої сторони, ще в 1943 році Маккалок і Піттс показали, що нейронна сітка може реалізувати будь-яку функцію алгебри логіки [4].

Під час побудови системи ШІ за даним методом основну увагу зосереджують на побудові початкової моделі, і правилам, за якими вона може змінюватися (еволюціонувати). Причому модель може бути створена за найрізноманітнішими методами, це може бути і НМ, і набір логічних правил, і будь-яка

інша модель. Після цього ми вмикаємо комп'ютер і він, на основі перевірки моделей відбирає найкращі з них, і по цих моделях за найрізноманітнішими правилами генеруються нові моделі. Серед еволюційних алгоритмів класичним вважається генетичний алгоритм [18].

Імітаційний підхід є класичним для кібернетики з одним із її базових понять чорний ящик. Об'єкт являє собою так званий чорний ящик – це об'єкт, поведінка якого імітується. Для нас не важливо, які моделі в нього всередині і як він функціонує, головне, щоби наша модель в аналогічних ситуаціях поведила себе без змін. Таким чином тут моделюється інша властивість людини – здатність копіювати те, що роблять інші, без поділу на елементарні операції і формального опису дій. Часто ця властивість економить багато часу об'єкту, особливо на початку його життя [17].

В рамках гібридних інтелектуальних систем намагаються об'єднати ці напрямки. Експертні правила висновків можуть генеруватися нейронними мережами, а породжуючі правила отримують з допомогою статистичного вивчення. Новий підхід, який ще називають підсилення інтелекту, розглядають досягнення ШІ в процесі еволюційної розробки як поточний ефект підсилення людського інтелекту технологіями [16].

Основною метою досліджень з питань штучного інтелекту є дослідження таємниць та секретів мислення та створення моделі людського мозку. Принципова можливість моделювання інтелектуальних процесів впливає з основного гносеологічного результату кібернетики, який полягає у тому, що будь-яку функцію мозку, будь-яку розумову діяльність, описану мовою з суворо однозначною семантикою за допомогою скінченного числа слів, в принципі можна передати електронній цифровій обчислювальній машині. Сучасні ж наукові уяв-

лення про природу мозку дають підстави вважати, що принаймні в суто інформаційному аспекті найістотніші закономірності мозку визначаються скінченною (хоч, може, й надзвичайно великою) системою правил [21].

Проблема штучного інтелекту досить тісно пов'язана з потребами практики, не ыснує єдиної загальної практичної задачі, яка б чітко визначала розвиток теорії, проте є багато задач, які є частковими, вузькими. Тому проблема штучного інтелекту – це можна сказати цілий комплекс задач, які характеризуються різним ступенем загальності, абстрактності, складності й розробленості і кожній з яких властиві свої принципи й практичні труднощі. Це такі задачі, як розпізнавання образів, навчання й самонавчання, евристичне програмування, створення загальної теорії самоорганізовуваних систем, побудова фізичної моделі нейрона та ін., багато з яких мають велике самостійне значення. По всіх цих напрямках одержано важливі результати, як практичного так і теоретичного характеру, продовжуються інтенсивні дослідження [24].

1.5 Загальні поняття нейронних мереж

Штучні нейронні мережі (ШНМ) – це програмна імплементація нейронних структур нашого мозку. Ми не будемо обговорювати складну біологію нашої голови, досить знати, що мозок містить нейрони, які є свого роду органічними перемикачами. Вони можуть змінювати тип переданих сигналів в залежності від електричних або хімічних сигналів, які в них передаються. Нейронна мережа у людському мозку – величезна взаємопов'язана система нейронів, де сигнал, який передається одним нейроном, може передаватися у тисячі інших нейронів. Навчання відбувається через повторну активацію деяких нейронних з'єднань. Через це збільшується імовірність виведення потрібного результату

при відповідній вхідній інформації (сигналах). Такий вид навчання використовує зворотний зв'язок – при правильному результаті нейронні зв'язки, які виводять його, стають більш щільними [26].

Штучні нейронні мережі імітують поведінку мозку у простішому вигляді. Вони можуть бути навчені контрольованим та неконтрольованим шляхами. У контрольованій ШНМ, мережа навчається шляхом передавання відповідної вхідної інформації та прикладів вихідної інформації. Наприклад, спам-фільтр у електронній поштовій скриньці: вхідною інформацією може бути список слів, які зазвичай містяться у спам-повідомленнях, а вихідною інформацією - класифікація для відповідного повідомлення (спам, чи не спам). Такий вид навчання додає ваги зв'язкам ШНМ, але це буде обговорено пізніше.

Неконтрольоване навчання у ШНМ намагається "змусити" ШНМ "зрозуміти" структуру переданої вхідної інформації "самостійно".

Розвиток інтернету та процеси глобалізації сприяли тому, що з'явилося дуже багато інформації, опрацювати яку самотужки людина фізично не в змозі. Нейронні мережі знайшли застосування у:

- 1) аналізі та класифікуванні даних за заданими параметрами;
- 2) формуванні аналітичних прогнозів, керуючись вхідною інформацією;
- 3) порівнянні та розпізнаванні ідентичних даних.

Останній пункт, наприклад, використовується в системах безпеки аеропортів. Виконується це шляхом фіксації обличчя людей, та порівняння їх із базою злочинців. Ще один приклад – функція Google по пошуку схожого зображення. Достатньо завантажити фото і система знайде усі схожі картинки [28].

Існує декілька видів нейронного зв'язку. Найчастіше використовуються синапсоїдальний та ReLU. В першому випадку нейронна мережа використовує дані в діапазоні від -1 до 1 (що фактично відповідає -100% до 100%). В другому

вхідні дані передаються через значення 0 та \inf (інформація любого характеру) [29].

Для того, щоб пояснити, як проходить системний аналіз, краще підходить синапсоїдальна функція, оскільки обмежений діапазон вхідної інформації дає більше наочності. Алгоритм обчислення:

- 1) дані поступають на нейрон;
- 2) обчислюється їх вага;
- 3) результати обчислень передаються на наступний нейрон;
- 4) процес повторюється.

Кількість обчислень задається шляхом встановлення кількості шарів. Сучасні нейронні мережі мають десятки, а іноді навіть сотні шарів обчислення. Книги з програмування містять приклади коду на Java, що свідчить про розвиток технології не тільки в сфері десктопних програм, а також і для мобільних платформ. Це свідчить про ефективність нейронних мереж.

1.6 Класифікація зображень

Класифікація зображень передбачає наявність певного вхідного набору зображень, який вже був попередньо класифікований. В межах комп'ютерного бачення необхідно врахувати ці категорії для нового набору тестових зображень, зробити певні прогнози щодо того, які зображення увійдуть у нові класифікації, та виміряти точність прогнозів, що були зроблені. Існують різні проблеми, пов'язані з цим завданням, включаючи зміну масштабу, освітлення та кута огляду, внутрішньокласові варіації зображення, деформацію зображення, оклюзії тощо [30].

Для вирішення завдань класифікації у сфері комп'ютерного бачення використовується наступний підхід. Замість того, щоб визначати кожен окрему

категорію, комп'ютер забезпечується великою кількістю прикладів кожного класу зображення, а потім виконується алгоритм навчання, який при обробці цих прикладів дізнається про візуальний зовнішній вигляд кожного класу. Іншими словами, спочатку накопичується навчальна множина завантажених зображень, а потім комп'ютер обробляє ці дані. З огляду на цей факт, повна класифікація зображень може бути формалізована наступним чином:

- 1) вхідними даними є готова тренувальна множина, яка складається з N зображень, які відповідають певному значенню K ;
- 2) навчальний набір використовується для підготовки класифікатора, щоб дізнатися, як кожен з класів виглядає;
- 3) проводиться оцінка якості класифікатора. Використовується тестувальна множина, і визначається, як часто класифікатор помилявся при визначенні класу об'єкту на зображенні.

Найбільш популярною архітектурою, що використовується для класифікації зображень, є згорткова нейронна мережа. Згорткова мережа, як правило, починається з вхідного «сканеру», який не призначений для розбору всіх даних тренувальної множини за один раз. Наприклад, для введення зображення 100×100 пікселів, не потрібно, щоб вхідний вектор мав розмірність 10000. Замість цього, створюється сканування вхідного шару розмірністю 10×10 , який репрезентує перші 10×10 пікселів зображення. Після проходження першої частини зображення, використовуються наступні 10×10 пікселів зображення, така техніка називається «ковзне вікно». Ці вхідні дані, замість звичайних шарів, подаються за допомогою згорткових шарів. Кожен вузол стосується лише сусідніх вузлів. Такі згорткові шари, як правило, зменшуються, коли вони стають більш глибокими, оскільки вхідні дані легко ділити.

Окрім цього, в згорткових нейронних мережах часто присутній агрегувальний шар. Агрегування є способом відфільтрувати деталі: загальноприйнята

методика агрегації – взяти 2 x 2 пікселі та передати піксель з найбільшою величиною певного атрибута. В даний час, більшість методів класифікації зображень проходять навчання в ImageNet – набір даних з приблизно 1,2 мільйона навчальних зображень з високою роздільною здатністю. Тестові зображення представляються без первинної анотації (сегментації або міток) і алгоритм повинен самостійно виробляти мітки, вказуючи, які об'єкти присутні у зображеннях. Деякі з найкращих існуючих методів комп'ютерного бачення були випробувані на цьому наборі даних за допомогою провідних груп з вивчення комп'ютерного бачення в Оксфорді, INRIA та XRCE [32]. Як правило, системи комп'ютерного зору використовують складні багатоступеневі алгоритми, а їх ранні етапи, зазвичай, налаштовуються шляхом оптимізації кількох параметрів.

1.7 Технології штучного інтелекту розпізнавання емоцій

Багато хто з нас вже стикалися з технологією розпізнавання облич під час розблокування смартфонів або коли відмічали друзів на фото у фейсбуці. Розвинені країни домоглися величезного прогресу в застосуванні цієї технології в різних сферах. Китай – один з лідерів в галузі, ставить для себе амбітну мету стати лідером з кількості встановлених камер з функцією розпізнавання країни та вже встановив біля 170 мільйонів камер [33].

Японія в свою чергу впроваджує розпізнавання облич у виробництво автомобілів, де ця функція використовується для визначення ознак втоми водія та запобігання аварій, у системи контролю доступу, що були створені перед Олімпіадою в Токіо у 20 році, а також були створені на робочих місцях співробітників для перевірки їх стану та підвищення продуктивності.

В Україні розвиток та впровадження технології розпізнавання облич тільки починається, але вже зараз можна впевнено заявити, що технологія зробила крок в корпоративний сектор. Багато підприємств починають впроваджувати розпізнавання облич у свої системи безпеки. Особливо великі перспективи для впровадження технології мають системи контролю доступу та обліку робочого часу персоналу, де вона дозволяє позбавитися від ID карт та підвищити рівень безпеки компанії [33].

Крім того, розглядаються можливості розпізнавання облич банками та платіжними системами. Наприклад, ПриватБанк спільно з віза тестує систему FacePay24, яка дозволить користувачам відмовитися не тільки від носіння гаманця, але й від потреби використовувати смартфон. Усі покупки будуть оплачуватися в автоматичному режимі з використанням системи розпізнавання облич [34].

Серед крупних компаній світового рівня, Amazon вже активно використовує дане рішення в своїх автоматизованих магазинах, використовується технології розпізнавання облич й в сфері Smart retail.

Біометрична технологія розпізнавання облич все частіше використовується для ідентифікації людей, що з'являються в поле зору камер відеоспостереження. Якщо коротко, то розпізнавання облич людей можна віднести до здатності програмного забезпечення розпізнавати особистість людини по її обличчю. За канону, дана технологія зображає людське обличчя, визначає відстань між різними точками (наприклад, між очима або за формою носа) – а потім розпізнає інші обличчя, які відповідають унікальній біометричній характеристиці оригінальної особи.

Сьогодні технологія розпізнавання облич заснована на методах глибокого навчання (підмножина штучного інтелекту), відповідно до яких мережа

навчається через вплив помічених даних. Якщо наслідувати те, як вчать людину, мережа навчається, та може виявляти, ідентифікувати та класифікувати дані – оскільки з часом вона піддається більшій кількості інформації. Розпізнавання облич на основі програмного забезпечення глибокого навчання дозволяє витягувати унікальні функції ідентифікації з вхідних даних зображення обличчя та зіставляти їх з банком еталонних функцій, для визначення особистості людини в запиті. Хоча точна внутрішня робота алгоритмів, розроблених машиною, не є повністю прозорою, масові дослідження в сегменті глибокого навчання та розробка апаратного забезпечення дозволили алгоритмам Deep learning досягти надлюдської продуктивності, виявляючи збіги облич в сценаріях, де люди не здатні цього зробити [34].

Застосування технологій розпізнавання облич продовжує розвиватися та стає більш точною й надійною у порівнянні з попередніми розробками. З появою на ринку комплектів відеоспостереження з більш високою роздільною здатністю та інновацій для штучного інтелекту, користувачі можуть очікувати появи нових рішень розпізнавання облич в правоохоронних органах, банках, аеропортах та роздрібній торгівлі в найближчі роки.

1.8 Висновки з розділу 1

Показано, що на даний момент в рамках глибинного навчання нейронних мереж є багато перспективних напрямків, одним із основних серед яких є комп'ютерне бачення, що представляє собою розпізнавання візуальних середовищ та їх контекстів. Проведений детальний огляд методів вирішення задач комп'ютерного бачення показав, що для створення системи розпізнавання деталей одягу доцільно обрати нейромережевий метод. Проаналізовано прин-

ципи організації згорткових нейронних мереж та основні методи навчання таких мереж. Розглянуті принципи організації навчання нейронних мереж показали, що на сьогодні залишаються невирішеними питання ефективного навчання нейронних структур для підвищення точності класифікації об'єктів.

Зрозуміло, що для вирішення задач, поставлених в межах дослідження даної роботи доцільно використовувати саме згорткові нейронні мережі, адже вони чудово себе показують в задачах розпізнавання образів на зображеннях.

РОЗДІЛ 2 ДОСЛІДЖЕННЯ ЗАСОБІВ РОЗРОБКИ НЕЙРОННОЇ МЕРЕЖІ ДЛЯ ВИЗНАЧЕННЯ ЕМОЦІЙ

2.1 Глибокі нейронні мережі

Глибинні нейронні мережі набули великої популярності через свою здатність вирішувати складні завдання. Існує сім основних видів глибинних нейронних мереж: нейронна мережа прямого поширення, мережа радіальних базисних функцій, самоорганізаційна карта Кохонена, рекурсивна нейронна мережа, рекурентна нейронна мережа, згорткова нейронна мережа, модулярна нейронна мережа.

Як і в разі штучних нейронних мереж, можуть виникнути труднощі з навчанням глибоких нейронних мереж, якщо вони навчаються примітивними наївними способами. Дві основні проблеми – перенавчання і великий час обчислень.

2.2 Огляд структури глибоких нейронних мереж

Глибока нейронна мережа є штучною нейронною мережею з декількома прихованими шарами між вхідним і вихідним шарами. Подібно звичайній штучній нейронній мережі, глибока мережа може моделювати складні нелінійні відносини. Глибоку нейронну мережу архітектури, наприклад, для виявлення об'єкта і синтаксичного аналізу, генерують композиційні моделі, де об'єкт виражається у вигляді шаруватої композиції примітивів зображення. Глибокі нейронні мережі, як правило, виконані у вигляді мереж прямого поширення, але тут також дуже успішно застосовуються рекурентні нейронні мережі. Зго-

рткові нейронні мережі використовуються в області комп'ютерного зору. Згорткові нейронні мережі також були застосовані до акустичного моделювання для автоматичного розпізнавання мови, де вони були успішні в порівнянні з попередніми моделями.

2.3 Зворотне поширення помилки

Глибока нейронна мережа може бути навчена стандартним алгоритмом backpropagation. Згідно з різними джерелами, основи безперервного зворотного поширення були отримані в контексті теорії управління Генрі Дж. Келлі в 1960 р і Артуром Е. Брайсоном в 1961 р, який застосував принципи динамічного програмування. У 1962 р Стюарт Дрейфус опублікував понад простий висновок, заснований тільки на ланцюговому правилі. У 1970 р Сеппо Ліннайрмаа опублікував загальний метод автоматичного диференціювання дискретних пов'язаних мереж за допомогою вкладених диференціювань. Це відповідає сучасній версії методу backpropagation, який є ефективним, навіть коли мережі слабо пов'язані. В 1973 р Стюарт Дрейфус використовується алгоритм backpropagation для адаптації параметрів контролерів пропорційно градієнтам помилок. У 1974 р Пол Уербос згадав про можливість застосування цього принципу в штучних нейронних мереж, і в 1982 р, він застосував метод вкладених диференціювань Linnairmaa до нейронних мереж, який широко використовується в даний час. В 1986 р Девід Е. Румельхарт, Джеффри Е. Хінтон і Рональд Дж. Вільямс показали, за допомогою комп'ютерних експериментів, що цей метод може генерувати корисні внутрішні уявлення вхідних даних в прихованих шарах нейронних мереж. У 1993 р Ерік А. Ван був першим, хто виграв міжнародний конкурс розпізнавання образів за допомогою backpropagation.

Як і в разі штучних нейронних мереж, можуть виникнути труднощі з навчанням глибоких нейронних мереж, якщо вони навчаються примітивними наївними способами. Дві основні проблеми – перенавчання і великий час обчислень. Домінуючий метод для навчання структур глибоких нейронних мереж – це корекція помилок навчання (наприклад, *backpropagation* з градієнтним спуском) завдяки простоті реалізації і тенденції сходитися на краще локальних оптимумів, ніж інші методи навчання. Проте, ці методи можуть бути обчислювально дорогими, особливо для глибоких нейронних мереж (тобто вимагати більшого часу навчання або пам'яті). Є багато параметрів навчання, які необхідно враховувати при навчанні глибоких нейронних мереж, такі як розмір (кількість шарів і кількість одиниць на шар), швидкість навчання та початкових ваг. Охоплення всього простору параметрів для оптимальних параметрів може виявитися неможливим через витрати часу і обчислювальних ресурсів. Було показано, що різні підходи, такі як використання міні-пакування, корисні для прискорення обчислень. Велика пропускна здатність обробки графічних процесорів виправила значні прискорення в навчанні в силу векторноматричного характеру обчислень, які тут необхідні і добре виконуються графічними процесорами.

2.4 Нейронна мережа прямого поширення

Нейронна мережа прямого поширення, нейромережа прямого розповсюдження (*Feedforward neural network*) – вид нейронної мережі, в якій сигнали поширюються в одному напрямку, починаючи від вхідного шару нейронів, через приховані шари до вихідного шару і на вихідних нейронах отримується результат опрацювання сигналу. В мережах такого виду немає зворотних зв'яз-

ків. Протилежним видом нейронних мереж із зворотними зв'язками є рекурентні нейронні мережі. Прикладом нейронної мережі прямого поширення є перцептрон Розенблатта, від якого і беруть свій початок нейромережі прямого розповсюдження. Основна математична задача, з якою він здатний впоратися - це лінійне розділення довільних нелінійних множин, так зване забезпечення лінійної сепарабельності. В літературі часто термін перцептрон, багат шаровий перцептрон та нейромережа прямого поширення застосовуються синонімічно. Власне, між різними видами перцептронів спільне одне – вони усі є нейромережами з прямим поширенням сигналу, різняться в основному кількістю шарів, функцією активації та методом навчання.

2.5 Мережа радіальних базисних функцій

Мережа радіально базисних функцій використовується у математичному моделюванні – це штучна нейронна мережа, яка має радіальні базисні функції у якості функції активації. Виходом мережі є лінійна комбінація радіальних базисних функцій входу та параметрів нейрона. Мережі радіальних базисних функцій мають багато застосувань, зокрема, такі як апроксимацію функції, прогнозування часових рядів, задачі класифікації та керування системою.

Мережі радіальних базисних функцій мають ряд переваг перед розглянутими багат шаровими мережами прямого поширення. По-перше, вони моделюють довільну нелінійну функцію за допомогою всього одного проміжного шару, тим самим, позбавляючи розробника від необхідності вирішувати питання про кількість шарів. По-друге, параметри лінійної комбінації в вихідному шарі можна повністю оптимізувати за допомогою добре відомих методів лінійної оптимізації, які працюють швидко і не відчувають труднощів з локаль-

ними мінімумами, так заважають при навчанні з використанням алгоритму зворотного поширення помилки. Тому мережа радіальних базисних функцій навчається дуже швидко – на порядок швидше, ніж з використанням алгоритму зворотного поширення.

Недоліки мереж радіальних базисних функцій: дані мережі володіють поганими екстраполюючих властивостями і виходять досить громіздкими при великій розмірності вектора входів.

Нейронна мережа радіальних базисних функцій містить у найбільш простій формі три шари: звичайний вхідний шар, що виконує розподіл даних зразка для першого шару ваг; шар прихованих нейронів з радіально симетричної активаційний функцією, кожен j -й з яких призначений для зберігання окремого еталонного вектора у вигляді вектора ваг $w_j(h)$; вихідний шар.

2.6 Згорткові нейронні мережі

Згорткова нейронна мережа, СНС, CNN – основний інструмент для класифікації та розпізнавання об'єктів, облич на фотографіях, розпізнавання мови. Існує велика кількість варіантів застосування штучних згорткових нейронних мереж, такі як глибокі мережі, Region-згорткові мережі, Fully Convolutional Neural Networks та Mask R-згорткові нейронні мережі та інші.

Згорткові нейромережі у порівнянні з алгоритмами класифікацій зображень, використовують досить мало попередньої обробки. Створена мережа навчається використовуючи фільтрів, що в зазвичай у традиційних алгоритмах потрібно розробляти власноруч. Згорткові нейронні мережі мають шари входу та виходу, а також певну кількість прихованих шарів. Приховані шари Згорткова ж нейронна мережа за рахунок застосування спеціальної операції – власне згортки – дозволяє водночас зменшити кількість інформації, що зберігається в

пам'яті інформації, за рахунок чого краще працює з картинками більш високої роздільної здатності, та виділити необхідні опорні ознаки зображення, а саме контури або грані. На наступному рівні обробки можна розпізнати фрагменти текстур, що повторюються, які далі можуть об'єднатися в певні фрагменти зображення.

Згорткова нейронна мережа – спеціальна архітектура штучних нейронних мереж, націлена на ефективне розпізнавання образів, входить до складу технологій глибокого навчання. Згорткові нейронні мережі будуються на трьох базових ідеях: локальне рецептивне сприйняття (Local receptive fields), колективні ваги (ядро згортки), субдискретизація (Pooling).

Сьогодні глибинне навчання лежить в основі послуг багатьох компаній: Facebook використовує нейронні мережі для алгоритмів автоматичного створення тегів, Google – для пошуку серед фотографій користувача, Amazon – для генерації рекомендацій товарів, Pinterest – використовує для персоналізації домашніх сторінок своїх користувачів, а Instagram – для пошукової інфраструктури. Але класичний, і, можливо, найпопулярніший варіант використання мереж це обробка зображень.

Особливість згорткової нейронної мережі полягає в тому, що в ній нейрони перших рівнів впорядковані в особливу структуру, а саме: на перших шарах нейрони розбиті на зображення певного розміру (їх ще іноді називають картами), і різні карти всередині одного шару відповідають нейронам різного типу, які реагують на різні особливості зображень. І обчислення активації наступного шару в згорткових нейронних мережах бувають двох основних типів. У першому типі обчислень активація нейронів наступного рівня обчислюється як лінійна комбінація активацій нейронів попереднього рівня, причому ваги цих лінійних активацій залежать тільки від взаємного положення нейронів, типів нейронів, але не залежать від положення даного нейрона всередині карти.

У другому типі обчислень активація нейронів на наступному рівні просто повторює активацію нейронів на попередньому рівні, але зображення стає меншого розміру за рахунок того, що активація поруч розташованих нейронів замінюється на їх максимум або їх середнє – так звана процедура пулінг. Подібна додаткова структура робить згорткові нейронної мережі дуже придатними для роботи з зображеннями.

Завдання, які нейронна мережа може виконувати – автоматичне водіння автомобілів на дорогах, створення приголомшливо реалістичних осіб комп'ютерної графіки, машинного перекладу, виявлення шахрайства, читання наших думок, розпізнавання; нейронні мережі стоять за багатьма з найбільших досягнень у штучному інтелекті.

2.7 Структура згорткової нейронної мережі

Згорткові нейронні мережі застосовуються досить широко і в різних областях. У згортковій нейронній мережі виходи проміжних шарів утворюють матрицю (зображення) або набір матриць (кілька шарів зображення). Так, наприклад, на вхід згорткової нейронної мережі можна подавати три шари зображення (R-, G-, B-канали зображення). Основними видами шарів в ЗНМ є згорткові шари (Convolutional layer), пулінгові шари (Pooling layer) і повнозв'язні шари (Fully-connected layer).

Згорткова нейронна мережа – це нейронна мережа прямого поширення спеціального виду, яка займається розпізнаванням ознак певних класів об'єктів на зображенні. У порівнянні з нейронною мережею прямого поширення згорткова нейронна мережа використовує значно меншу кількість параметрів. На рисунку 1 зображена її архітектура.

Класифікації за допомогою CNN активно застосовуються в медицині: можна навчити нейронну мережу класифікації хвороб або симптомів, наприклад, для МРТ-діагностики.

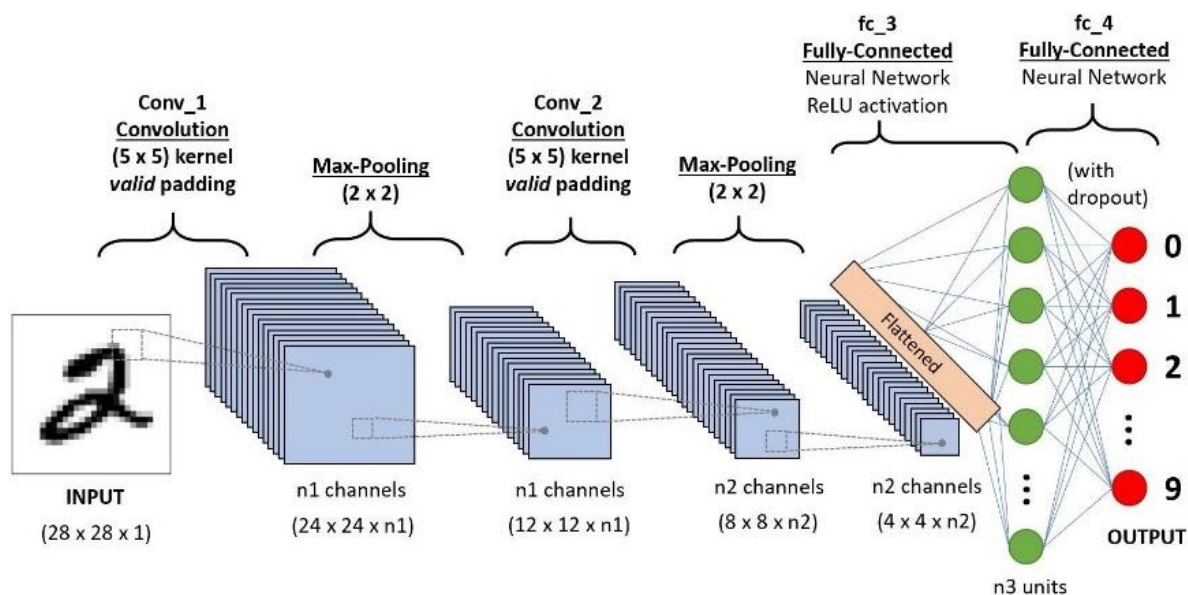


Рис. 1 Архітектура згорткової нейронної мережі

2.7.1 Згортковий шар

Згортковий шар є основним будівельним блоком ОНР. Параметри шару складаються з набору фільтрів для навчання (або ядер), які мають невеликі рецептивні поля, але простягаються на всю глибину вхідного об'єму. Протягом прямого проходу кожен фільтр здійснює згортку по ширині і висоті вхідного об'єму, обчислюючи скалярний твір даних фільтра і входу, і формуючи двовимірну карту активації цього фільтра. У результаті мережа вчиться, які фільтри активуються, коли вони бачать певний конкретний тип ознаки в певному просторовому положенні при вході.

Складання активаційних карт всіх фільтрів уздовж вимірювання глибини формує повний вихідний об'єм згорткового шару. Таким чином, кожен запис у

вихідному об'ємі може також трактуватися як вихід нейрона, який дивиться на невелику область на вході, і ділить параметри з нейронами тієї ж активіаційної карти.

Згортковий шар реалізує ідею – локальних рецептивних полів, тобто кожен вихідний нейрон з'єднаний тільки з певною (невеликою) областю вхідної матриці і таким чином моделює деякі особливості людського зору. На рис. 2 схематично зображено шар згортки.

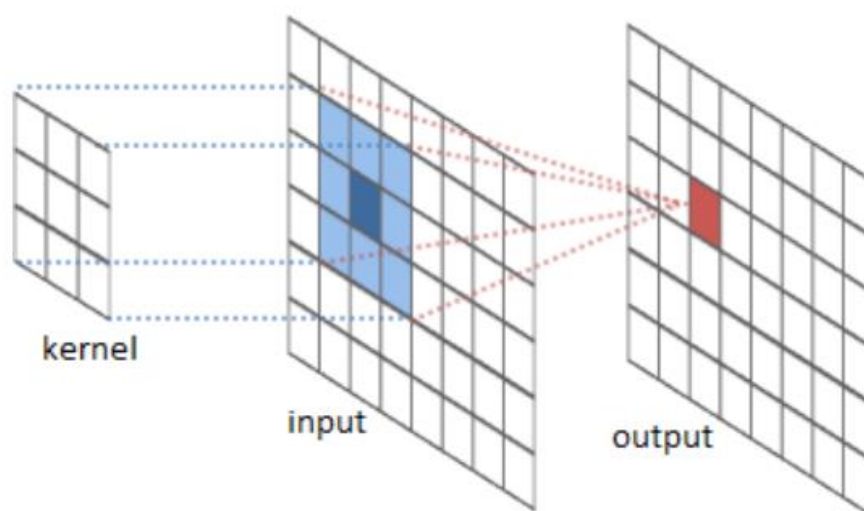


Рис. 2 Шар згортки

Перший шар згорткової нейронної мережі – згортковий (convolution layer). Згорткова нейронна мережа працює на основі фільтрів, які займаються розпізнаванням ознак певних класів об'єктів на зображенні: прямих ліній, кривих певного типу тощо. Фільтр – це матриця чисел (ваг), які навчаються виявляти ознаки певних об'єктів на зображенні. Фільтр переміщається уздовж зображення та виявляє, чи присутня деяка ознака у його частині. Для отримання

такої відповіді здійснюється операція згортки (convolution operation), яка математично визначається як сума добутків елементів фільтру та частини вхідного зображення. Назва згорткової нейронної мережі походить саме від назви цієї операції. На рисунку 3 зображена операція згортки.

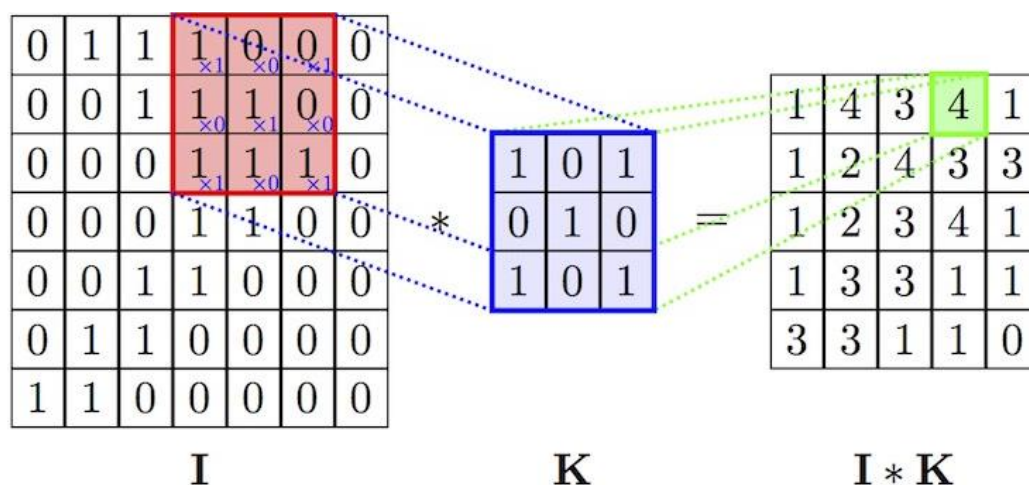


Рис. 3 Операція згортки

2.8 Рекурсивна нейронна мережа

Рекурсивна нейронна мережа – тип глибокої нейронної мережі, сформована при застосуванні одних і тих же наборів ваг рекурсивно надструктурою, щоб зробити скалярне або структуроване пророкування надвхідною структурою змінного розміру через активацію структури в топологічному порядку (рисунки 4). У простій архітектурі нелінійність, така як тангенціальна функція активації і матриця ваг використовуються для об'єднання вузлів в батьківські об'єкти.

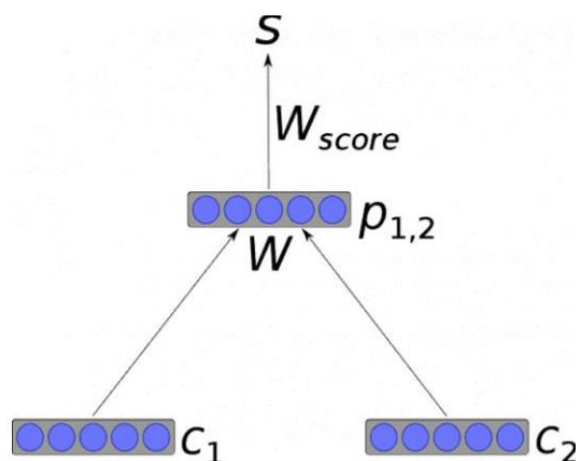


Рис. 4 Загальний вигляд рекурсивної нейронної мережі

2.9 Пулінг або шар субдискретизації

Шар пулінг (інакше підвибірки, субдискретизації) являє собою нелінійне ущільнення карти ознак, при цьому група пікселів (зазвичай розміру 2×2) ущільнюється до одного пікселя, проходячи нелінійне перетворення. Найбільш споживані при цьому функція максимуму (рисунок 5). Перетворення зачіпають непересічні прямокутники або квадрати, кожен з яких скорочується в один піксель, при цьому вибирається піксель, що має максимальне значення. Операція пулінг дозволяє істотно зменшити просторовий обсяг зображення. Пулінг інтерпретується так. При виявленні деяких ознак на попередній операції згортки для подальшої обробки вже не потрібне настільки докладне зображення, і воно ущільнюється до менш докладного. До того ж фільтрація вже непотрібних деталей допомагає не перенавчатися. Шар пулінг, як правило, вставляється після шару згортки перед шаром наступної згортки. Крім пулінгу з функцією максимуму можна використовувати і інші функції - наприклад, середнього значення

або L2-нормування. Однак практика показала переваги саме пулінгу з функцією максимуму, який включається в типові системи.

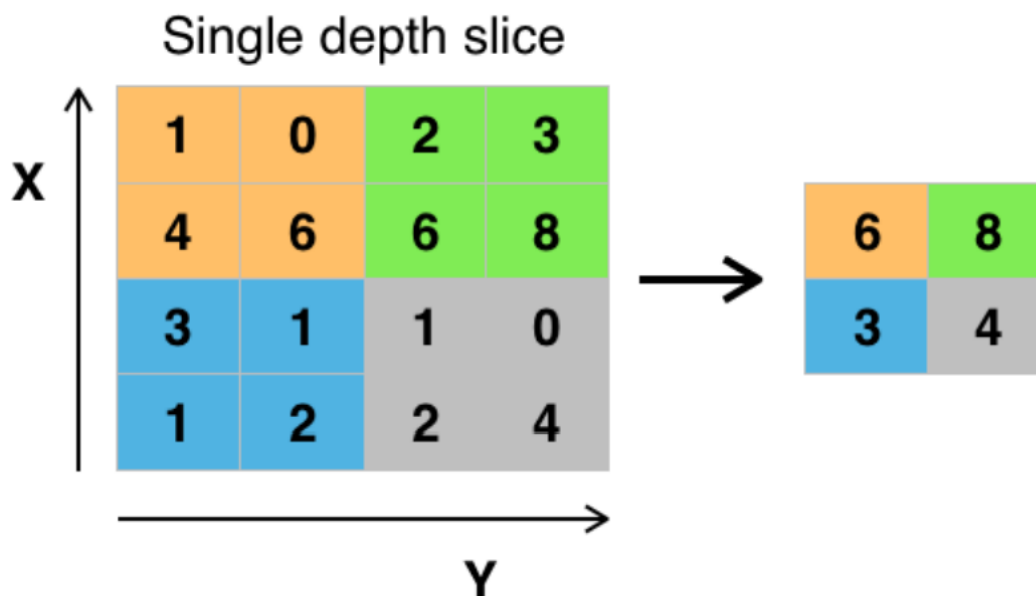


Рис. 5 Макспулінг

Крім завдань ущільнення зображення (що корисно для запобігання перенавчання), знаходять поширення також ідеї використання малих фільтрів або повна відмова від шарів пулінг в архітектурі мережі.

2.10 Оптимізатори

Оптимізатор організовує оптимізацію моделі, координуючи прямий і зворотний градієнти мережі для формування оновлень параметрів, які намагаються поліпшити функцію втрат. Обов'язки навчання розподіляються наступним чином: оптимізатор використовується для контролю оптимізації та гене-

рації оновлень параметрів, а мережа – для отримання збитків і градієнтів. Стохастичний градієнтний спуск (часто скорочений в SGD), також відомий як інкрементний градієнтний спуск, є стохастичною апроксимацією методу оптимізації спуску градієнта для мінімізації цільової функції, яка записується як сума диференціуються. Іншими словами, SGD намагається знайти мінімуми або максимуми шляхом ітерації.

2.11 Рекурентна нейронна мережа

Рекурентні нейронні мережі – це клас штучних нейронних мереж, у якому з'єднання між вузлами утворюють граф орієнтований у часі. Це створює внутрішній стан мережі, що дозволяє їй проявляти динамічну поведінку в часі. На відміну від нейронних мереж прямого поширення, РНМ можуть використовувати свою внутрішню пам'ять для обробки довільних послідовностей входів. Це робить їх застосовними до таких задач, як розпізнавання несегментованого неперервного рукописного тексту та розпізнавання мовлення.

Рекурентна нейронна мережа, на відміну від прямої нейронної мережі, є варіантом рекурсивної ШНМ, в якій зв'язки між нейронами – спрямовані цикли. Останнє означає, що вихідна інформація залежить не тільки від поточного входу, але також від станів нейрона на попередньому кроці. Така пам'ять дозволяє користувачам вирішувати завдання нейролінгвістичного програмування: розпізнавання рукописного тексту або мови.

2.12 Проблеми навчання нейронних мереж

При навчанні нейронних мереж у будь-якому з підвидів навчання, можна стикнутись з багатьма проблемами, які не дадуть досягти хороших результатів

у вирішенні поставленої задачі. Тому є потреба в конфігурації самих нейронних мереж та процесу навчання. Розглянемо нижче основні проблеми при навчанні нейронних мереж [22].

2.12.1 Перенавчання

Перенавчання нейронної мережі – це стан мережі, коли вона так привчає себе до певного набору даних, на якому вона навчається, що прогнозування на інших наборах даних стають повністю неправильними. Таким чином, коли ми навчаємо занадто довго нейронну мережу на одному і тому ж наборі даних, наша модель починає, замість вивчення узагальненої структури даних або їх закономірностей, вивчати шум даних або просто нерелевантну інформацію. Таку нейронну мережу називають перенавченою. Тоді нейронна мережа не в стані застосовувати вивчені закономірності до нових наборів даних, відповідно не в стані робити коректні прогнозування [22]., [23].

2.12.2 Недонавчання

Недонавчання – це стан нейронної мережі, коли вона недостатньо вловила закономірності в даних, тому точність прогнозувань є дуже маленькою. Це протилежний стан перенавчанню нейронної мережі [23].

В задачах регресії та класифікації, нескладно помітити, що недонавчена нейронна мережа не в стані ловити закономірності в даних на противагу перенавченій, яка починає вивчати шум в даних. Причини недонавченості нейронної мережі може бути недостатній час навчання, неправильна ініціалізація вагів, при якій модель не вчиться та мала складність моделі, при якій модель не встані взагалі вивчити хоч якісь закономірності в даних.

2.13 Фреймворки та бібліотеки глибокого навчання

2.13.1 Theano

Theano - бібліотека чисельного обчислення в Python. Обчислення в Theano виражаються подібним до NumPy синтаксисом і компілюються для ефективних паралельних обчислень як на звичайних CPU, так і на GPU. Theano є проектом з відкритим вихідним кодом, основним розробником якого є група машинного навчання в Монреальському університеті. Дозволяє користувачеві писати символічні формальні математичні вирази, з яких автоматично генерується похідний код. Таким чином користувачеві не потрібно програмувати градієнти або зворотне поширення помилки. Такі вираження автоматичний компілюються в шейдерний код для CUDA для оптимізації обчислень на GPU.

2.13.2 Caffe

Caffe підтримує багато типів машинного навчання, націлених в першу чергу на вирішення завдань класифікації та сегментації зображень. Підтримує згорткові нейронні мережі, RCNN, довгу короткострокову пам'ять і повнозв'язні нейронні мережі. При цьому для прискорення навчання застосовується система графічних процесорів (GPU), що підтримує архітектуру CUDA та застосовує бібліотеку CuDNN від фірми Nvidia. Caffe дозволяє використовувати готові промислові конфігурації нейронних мереж, що пройшли апробацію. У комплект входить, зокрема AlexNet, яка перемогла в 2012 році в змаганні з розпізнавання зображень ImageNet, і GoogLeNet, яка перемогла в змаганнях ImageNet 2014 року. Caffe маніпулює блобами - багатовимірними масивами даних, які використовуються в паралельних обчисленнях, які поміщаються в CPU або GPU. Навчання CNN реалізується як паралельні багатопроцесорні обчислення блобів від шару до шару (прямим і зворотним ходом). Solver (вирішувач)

координує весь процес навчання - прямий хід від вхідних до вихідних даних, отримання функції помилок, зворотний хід (Метод backpropagation) назад від вихідного шару з використанням градієнтів помилок. При цьому Caffe реалізує різні стратегії навчання для Solvera.

2.13.3 Deeplearning4j

Бібліотека на Java і Scala використовує фреймворк з відкритим вихідним кодом для реалізації розподіленої обробки неструктурованих і слабоструктурованих даних Apache Spark. Є бібліотекою для глибинного навчання загального призначення, призначена для запуску на JVM оточенні. Ядром бібліотеки є блок наукових обчислень написаний на C ++. Дозволяє створювати шари з заданими параметрами. Інтегрований в пакети Hadoop і Kafka. Ліцензії Apache 2.0. Deeplearning4j – бібліотека програм на мові Java, яка використовується як фреймворк для глибокого навчання. Включає реалізацію обмеженою машини Больцмана, глибокої мережі довіри, глибокого автокодировщика, стекового автокодировщика з фільтрацією шуму, рекурсивної тензорною нейронної мережі, word2vec, doc2vec, and GloVe. Ці алгоритми включені також у версії бібліотеки, що підтримують розподілені обчислення, інтегровані з архітектурою Apache Hadoop і Spark. Додаткова бібліотека ND4J відкритого доступу забезпечує обчислення на графічних процесорах з підтримкою CUDA. Крім того, є засоби для роботи з бібліотекою мовою Python через фреймворк Keras.

2.13.4 MatConvNet

MatConvNet – це реалізація з відкритим кодом конволюційних нейронних мереж (CNN) з глибокою інтеграцією в середовище MATLAB. Панель інструментів розроблена з акцентом на простоту та гнучкість. Це розкриває будівельні блоки CNN як прості у використанні функції MATLAB, забезпечуючи

підпрограми для 32 обчислення згортків з банками фільтрів, об'єднання функцій, нормалізацію та набагато більше. MatConvNet можна легко розширити, часто використовуючи лише код MATLAB, що дозволяє швидко прототипувати нові CNN архітектури. У той же час він підтримує ефективність обчислення на процесорі та графічному процесорі, що дозволяють тренувати складні моделі на великих датасетах, таких як ImageNet ILSVRC, що містять мільйони навчальних прикладів.

2.13.5 TensorFlow

TensorFlow – відкрита програмна бібліотека для машинного навчання, розроблена компанією Google для вирішення завдань побудови і тренування нейронної мережі з метою автоматичного знаходження та класифікації образів, досягаючи якості людського сприйняття. Застосовується як для досліджень, так і для розробки власних продуктів Google. Основний API для роботи з бібліотекою реалізований для Python, також існують реалізації для C Sharp, C ++, Haskell, Java, Go і Swift. Підтримує обчислення на CPU, GPU і спеціально розробленими компанією Google TPU (Tensor processing units).

2.13.6 Бібліотека NumPy

NumPy – це одна з основних бібліотек Python для обробки масивів даних. Ця бібліотека надає високопродуктивні об'єкти багатовимірних масивів та інструменти для роботи з цими масивами.

NumPy – розширення мови Python, що додає підтримку великих багатовимірних масивів і матриць, разом з великою бібліотекою високорівневих математичних функцій для операцій з цими масивами. Попередник NumPy, Numeric, був спочатку створений Jim Hugunin. NumPy – відкрите програмне забезпечення і має багато розробників.

Оскільки Python – інтерпретована мова, математичні алгоритми, часто працюють в ньому набагато повільніше ніж у компільованих мовах, таких як С або навіть Java. NumPy намагається вирішити цю проблему для великої кількості обчислювальних алгоритмів забезпечуючи підтримку багатовимірних масивів і безліч функцій і операторів для роботи з ними. Таким чином будь-який алгоритм, який може бути виражений в основному як послідовність операцій над масивами і матрицями, працює так само швидко, як еквівалентний код, написаний на С.

2.13.7 Бібліотека Pandas

Pandas – це бібліотека, яка надає дуже зручні з точки зору використання інструменти для зберігання даних і роботи з ними. Особливість pandas полягає в тому, що ця бібліотека дуже швидка, гнучка і виразна. Це важливо, тому що вона використовується з мовою Python, яка не відрізняється високою продуктивністю. Pandas прекрасно підходить для роботи з одновимірними і двовимірними таблицями даних, добре інтегрована з іншими зовнішніми системами – є можливість працювати з файлами CSV, таблицями Excel, а також існує можливість використання разом мовою R.

Pandas добре підходять для різних типів даних:

- 1) табличні дані (схожі на таблиці SQL або Excel);
- 2) дані про дату або час;
- 3) довільні матричні дані (введені однорідно або неоднорідно) з мітками рядків і стовпців;
- 4) будь-яка інша форма спостережних або статистичних наборів даних.

Існує дві основні структури даних Pandas – це Series (одновимірна) та DataFrame (двовимірна), обробляють переважну більшість типових випадків

використання у фінансах, статистиці, суспільствознавстві та багатьох галузях інженерії. Бібліотека Pandas побудована на основі бібліотеки NumPy і може добре інтегруватися в наукове обчислювальне середовище з багатьма іншими бібліотеками сторонніх організацій.

Основні переваги Pandas:

- 1) проста робота з відсутніми даними (представленими як NaN), з цілими числами та з плаваючою точкою;
- 2) легко модифікувати: стовпці можна вставляти та видаляти з DataFrame;
- 3) автоматичне та явне вирівнювання даних: об'єкти можуть бути явно вирівняні використовуючи набір міток;
- 4) потужна та гнучка функція groupBy дозволяє виконувати операції комбінування і поділу на наборах даних для агрегування або перетворення даних;
- 5) легко конвертувати необроблені та по-різному індексовані дані в інших структурах даних Python та NumPy в об'єкти DataFrame;
- 6) інтелектуальний поділ даних на основі міток;
- 7) інтуїтивне об'єднання та приєднання наборів даних;
- 8) гнучка перестановка та поворот наборів даних;
- 9) надійні інструменти вводу-виводу для завантаження даних із файлів CSV, файлів Excel, баз даних та збереження і завантаження даних у надшвидкому форматі HDF5;
- 10) функціонал, орієнтований на роботу з часом: генерація діапазону дат і перетворення частоти, зміщення дати тощо.

2.13.8 Бібліотека Seaborn

Seaborn – це бібліотека для створення статистичної графіки в Python. Вона побудована поверх бібліотеки matplotlib і тісно інтегрована з структурами даних Pandas. Ось деякі з функцій, які пропонує бібліотека Seaborn:

- 1) API, орієнтований на набір даних, для вивчення зв'язків між декількома змінними;
- 2) спеціалізована підтримка використання категоричних змінних;
- 3) варіанти візуалізації уніваріантних або біваріантних розподілів та порівняння їх між підмножинами даних;
- 4) автоматичне оцінювання та побудова лінійних моделей регресії для різних змінних;
- 5) зручні графіки, що дають уявлення про загальну структуру складних наборів даних;
- 6) абстракції високого рівня для структурування сіток з декількома ділянками, які дозволяють легко будувати складні візуалізації;
- 7) інструменти для вибору кольорової палітри.

Seaborn дозволяє зробити візуалізацію центральною частиною вивчення та розуміння даних. Функції, орієнтовані на набір графіків, можуть працювати з фреймами даних та масивами, що містять цілі набори даних, внутрішньо виконують необхідне семантичне відображення та статистичну агрегацію для отримання інформативних графіків.

2.13.9 Фреймворк Keras

Keras – це відкритий фреймворк для глибинного навчання, написаний на мові програмування Python, що є надбудовою над бібліотекою TensorFlow. TensorFlow – це відкрита бібліотека для машинного навчання від компанії Google, написана на мові програмування C++. Головним розробником Keras є

Франсуа Шолле, інженер із Google. При розробці фреймворка головний акцент був зроблений на надання дослідникам можливостей для швидкого експериментування.

Фреймворк Keras можна інстальовати на Linux, Windows чи macOS. Як і Darknet, він дозволяє навчати нейронну мережу на декількох графічних процесорах одночасно. Основні структури даних, які використовує Keras – це шари (layers) та моделі (models). Найпростішою моделлю є Sequential, яка представляє собою послідовність із шарів нейронної мережі. Фреймворк надає безліч готових шарів нейронної мережі: згортковий шар, шар підвибірки, повністю пов'язаний шар тощо. Для побудови більш складних архітектур Keras надає Functional API, який дозволяє будувати графи із шарів та створювати власні моделі (models). Keras включає у себе навчені моделі нейронних мереж (MobileNet [Помилка! Джерело посилання не знайдено.], VGG [31] тощо), а також деякі датасети для різних задач (датасет рукописних цифр MNIST, датасет з відгуками про фільми з сайту IMDB тощо).

2.13.10 Фреймворк PyTorch

PyTorch – це відкритий фреймворк для машинного навчання, написаний на мовах програмування C, C++ та Python, що розроблений на базі бібліотеки Torch. Torch – це відкрита MATLAB-подібна бібліотека для мови програмування Lua, що реалізує велику кількість алгоритмів для глибинного навчання. PyTorch розроблюється лабораторією штучного інтелекту компанії Facebook.

Фреймворк PyTorch можна інстальовати на Linux, Windows чи macOS. PyTorch забезпечує дві основні функціональності: матричні обчислення з прискоренням на графічних процесорах та використання глибоких нейронних мереж, побудованих на базі системи автоматичного диференціювання. Фрейм-

ворк складається з трьох модулів: `autograd`, `optim` та `nn`. Модуль `autograd` включає у себе систему автоматичного диференціювання, модуль `optim` реалізує алгоритми оптимізації, які використовуються при побудові нейронних мереж, а модуль `nn` надає високорівневі абстракції над модулем `autograd`. Як і `Darknet` та `Keras`, `PyTorch` включає у себе навчені моделі нейронних мереж (`AlexNet` [19], `SqueezeNet` [29] тощо). Як і `Keras`, фреймворк включає у себе датасети для різних задач (`COCO` [Помилка! Джерело посилання не знайдено.], датасет рукописних цифр `MNIST` тощо).

2.13.11 Бібліотека `OpenCV`

`OpenCV` – це відкрита бібліотека комп’ютерного зору, написана на мові програмування `C++`. Також існують прив’язки (`bindings`) на мовах програмування `Python`, `Java`, `MATLAB` та інших. Розробником `OpenCV` є компанія `Intel`. `OpenCV` включає у себе алгоритми для обробки та перетворення зображень, функціонал для роботи з відео, а також модуль для використання глибоких нейронних мереж класифікації, розпізнавання, виявлення ключових точок та сегментації об’єктів на зображенні.

`OpenCV` можна інстальовати на платформи `Linux`, `Windows` чи `macOS`. Готова збірка `OpenCV` може не включати підтримки необхідного функціоналу, тому нормальною практикою є створення файлів збірки за допомогою `CMake` та подальша компіляція бібліотеки. Наприклад, готова збірка `OpenCV` не підтримує обчислення на графічному процесорі, що є важливим для застосунків, які повинні працювати у режимі реального часу.

`OpenCV` дозволяє використовувати навчені моделі нейронних мереж за допомогою модуля `dnn`. Бібліотека підтримує навчені на різних фреймворках моделі (моделі різного формату): `Darknet`, `Keras`, `PyTorch` та інших. Модуль `dnn`

підтримує основні шари нейронної мережі. Також він містить клас моделі нейронної мережі (Model), який є базовим класом для моделей класифікації (ClassificationModel), розпізнавання (DetectionModel), виявлення ключових точок (KeypointsModel) та сегментації (SegmentationModel). Кожна із моделей вирішує конкретну задачу комп'ютерного зору.

2.14 Розрахунок споживання пам'яті мережею

Сучасні графічні процесори мають ліміт пам'яті в 3, 4 або 6 Гб, в той час як кращі GPU розташовують 12 Гб. Є три основних джерела об'єктів, які займають пам'ять. Це необроблена кількість активацій на кожному шарі штучної нейронної мережі, а також їх градієнти. Зазвичай більшість активацій розташовується на найперших шарах згорткової нейромережі. Це слід мати на увазі, так як дані активації потрібні для backpropagation. Однак реалізація, яка запускає штучні нейронні мережі тільки під час випробувань, може зменшити кількість активацій шляхом збереження лише поточних з них на будь-якому шарі і відкидання всіх попередніх на шарах нижче. Це числа, які містять параметри мережі, їх градієнти під час зворотного поширення і, як правило, крок кеша. Таким чином, пам'ять для зберігання вектору параметрів повинна бути помножена на коефіцієнт, приблизно рівний 3; будь-яка реалізація згорткової нейронної мережі повинна підтримувати різнорідну пам'ять начебто пакета даних зображень, можливо, їх вдосконалені реалізації. Як тільки отримано приблизний розмір загальної кількості значень (для активацій, градієнтів і іншого), результат необхідно перетворити в Гб. Потрібно взяти кількість значень, помножити його на 4, щоб отримати число байтів, а потім розділити на 1024^3 , щоб отримати пам'ять в гігабайтах.

2.15 Висновки з розділу 2

Розглянуто основні види глибинних нейронних мереж. На основі проведеного порівняльного аналізу архітектур нейронних мереж показано, що саме згорткові нейронні мережі знайшли широке застосування при обробці зображень. Описані основні елементи згорткової мережі та особливості вибору параметрів для навчання нейронних мереж.

Розглянуті бібліотеки необхідна для створення нейронної мережі для обробки зображень для визначення емоцій. Проаналізовані переваги та недоліки кожної бібліотеки. Також розглянуті та проаналізовані фреймворки, написані на різних мовах програмування, що необхідні для роботи нейронної мережі.

РОЗДІЛ 3 РОЗРОБКА КОМП'ЮТЕРНОЇ СИСТЕМИ ДЛЯ ВИЗНАЧЕННЯ ЕМОЦІЙ ЛЮДИНИ

3.1 Налаштування середовища для навчання

Першим етапом, необхідним для навчання моделі розпізнавати емоції, є налаштування робочого середовища. Навчання було проведено на комп'ютері з наступними характеристиками:

1. Чотириядерний Intel Pentium Silver N5000 (1.1 - 2.7 ГГц).
2. Тип оперативної пам'яті DDR4-2133 МГц.
3. 4 ГБ оперативної пам'яті.
4. Графічний адаптер, об'єм пам'яті: Intel UHD Graphics 605
5. Операційна система Windows 10.

В даний час існує величезна кількість мов програмування, а також програмних бібліотек для роботи з штучними нейронними мережами та генетичними алгоритмами, реалізованих на різних мовах програмування, таких як C++, C#, Python і ін. В якості інструменту для вирішення задач, і реалізації алгоритмів вибрана мова програмування Python. Python – високорівнева мова програмування загального призначення, орієнтований на підвищення продуктивності розробника і читання коду. Синтаксис ядра Python мінімалістичний. У той же час стандартна бібліотека включає великий обсяг корисних функцій. Python підтримує структурний, об'єктно-орієнтоване, функціональне, імперативне і аспектно-орієнтоване програмування. Основні архітектурні риси – динамічна типізація, автоматичне керування пам'яттю, повна інтроспекція, механізм обробки виключень, підтримка багатопоточних обчислень, високорівневі структури даних. Підтримується розбиття програм на модулі, які, в свою чергу, можуть об'єднуватися в пакети. Виходячи з цього, в якості мови програмування

була обрана мова програмування Python, як середовище програмування було обрано програмне забезпечення Microsoft Visual Studio Code. Для вирішення завдань машинного навчання існує безліч бібліотек і систем, що підтримують інтерфейс мови Python. Деякі з них: Apache Singa, Caffe, Keras, Microsoft Cognitive Toolkit, MXNet, TensorFlow, Theano, PyTorch. Для реалізації розпізнавання обрана бібліотека Keras – відкрита нейромережева бібліотека, написана на мові Python – вона націлена на оперативну роботу з мережами глибокого навчання, при цьому спроектована так, щоб бути компактною, модульною та розширюється та OpenCV – бібліотека функцій та алгоритмів комп'ютерного зору, обробки зображень і чисельних алгоритмів загального призначення з відкритим кодом.

Для розробки нейромережі була використана платформа Kaggle. Середовище організоване як публічна веб-платформа, на якій користувачі та організації можуть публікувати набори даних, досліджувати та створювати моделі, взаємодіяти з іншими фахівцями з даних та інженерами з машинного навчання, організовувати конкурси з дослідження даних та брати участь у них. У системі розміщено набори відкритих даних, надаються хмарні інструменти для обробки даних та машинного навчання. Також реалізовані навчальні ресурси, є розділ для розміщення вакансій роботодавцями, де також можлива організація конкурсів для відбору найкращих кандидатів.

Також середовище Kaggle – платформа для змагань з аналітики та передбачуваного моделювання, в рамках якого статистики та добувачі даних конкурують у створенні найкращих моделі для прогнозування та опису даних, запропонованих компаніями або користувачами. Цей клаудсорсинговий підхід ґрунтується на тому, що є безліч стратегій, які можуть бути застосовані до будь-якого завдання з передбачуваного моделювання, і наперед не відомо, яка методика або аналітичний підхід буде найбільш ефективним.

Для роботи з даною платформою необхідно зареєструватися та налаштувати свій профіль (рисунок 6). Після цього можна знаходити набори даних, що є у відкритому доступі та використовувати за своїми потребами.

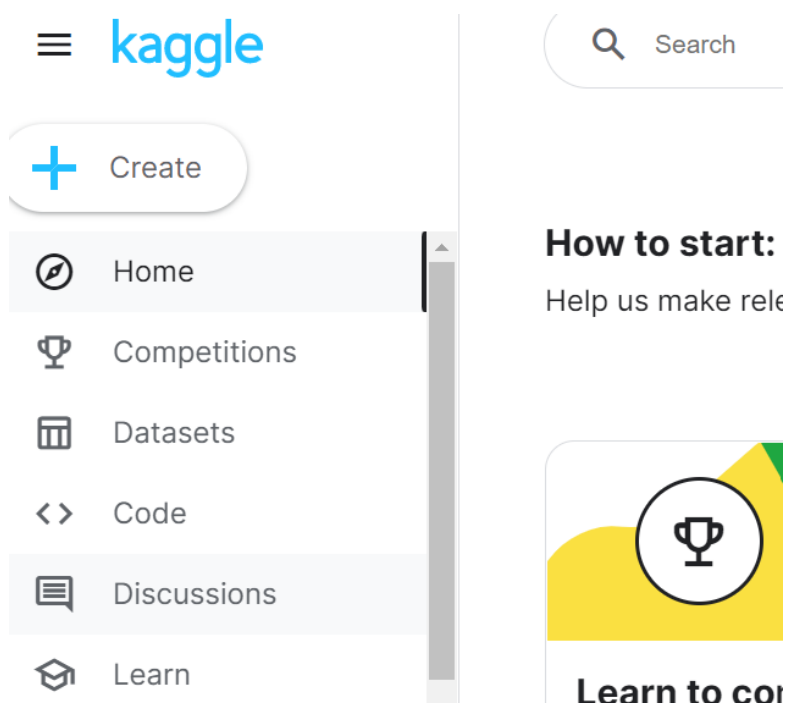


Рис. 6 Стартова сторінка платформи Kaggle

3.2 Підключення необхідних бібліотек для роботи

Для створення нейронної мережі в першу чергу були імпортовані всі необхідні бібліотеки.

Лістинг 1 демонструє імпорт всіх необхідних для роботи бібліотек.

Лістинг 1 Налаштування середовища

```
import numpy as np
import pandas as pd
```

```
import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

import math
import numpy as np
import pandas as pd

import scikitplot
import seaborn as sns
from matplotlib import pyplot

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import classification_report

import tensorflow as tf
from tensorflow.keras import optimizers
from tensorflow.keras.datasets import mnist
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Flatten, Dense, Conv2D,
MaxPooling2D
from tensorflow.keras.layers import Dropout,
BatchNormalization, LeakyReLU, Activation
from tensorflow.keras.callbacks import Callback,
EarlyStopping, ReduceLROnPlateau
from tensorflow.keras.preprocessing.image import
ImageDataGenerator
```

```
from keras.utils import np_utils
```

Після успішного підключення всіх необхідних бібліотек можна створювати нейронну мережу.

3.3 Загальний опис датасету

Дані складаються із зображень облич у градаціях сірого розміром 48x48 пікселів. Обличчя розташоване більш-менш по центру та займає приблизно однакову кількість місця на кожному зображенні. Задача для нейромережі була поставлена так, щоб класифікувати кожне обличчя на основі емоцій, показаних у виразі обличчя, в одну із семи категорій (0=Злість, 1=Огида, 2=Страх, 3=Щастя, 4=Сум, 5=Подив, 6=Нейтральний).

Файл `train.csv` містить два стовпці: "емоції" та "пікселі". Стовпець «емоція» містить числовий код від 0 до 6 включно для емоції, яка присутня на зображенні. Стовпець «пікселі» містить рядок у лапках для кожного зображення. Вміст цього рядка – значення пікселів, розділених пробілами, у порядку старшого рядка. Файл `test.csv` містить лише стовпець «пікселі», завдання нейронної мережі – передбачити стовпець емоцій.

Наступним кроком є підключення датасету, в якому зібрані зображення з різними людьми, різних вікових категорій, з різними типами шкіри, очей, зачісками, які виражають певні емоції.

```
df = pd.read_csv('../input/facial-expression-
recognitionferchallenge/fer2013/fer2013/fer2013.csv')
print(df.shape)
df.head()
```

В результаті отримали таблицю з розподілом емоцій на картинках (рисунок 7).

	emotion	pixels	Usage
0	0	70 80 82 72 58 58 60 63 54 58 60 48 89 115 121...	Training
1	0	151 150 147 155 148 133 111 140 170 174 182 15...	Training
2	2	231 212 156 164 174 138 161 173 182 200 106 38...	Training
3	4	24 32 36 30 32 23 19 20 30 41 21 22 32 34 21 1...	Training
4	6	4 0 0 0 0 0 0 0 0 0 0 3 15 23 28 48 50 58 84...	Training

Рис. 7 Таблиця з розподілом емоцій і пікселів

Далі отримаємо масив з кодами емоцій.

Лістинг 2 Реалізація функціоналу кодування емоцій людини

```
df.emotion.unique()

array([0, 2, 4, 6, 3, 5, 1])

emotion_label_to_text = {0:'anger', 1:'disgust', 2:'fear',
3:'happiness', 4: 'sadness', 5: 'surprise', 6: 'neutral'}
```

```
df.emotion.value_counts()
Out[6]:
3      8989
6      6198
4      6077
2      5121
0      4953
5      4002
1       547
Name: emotion, dtype: int64
```

В результаті роботи програми отримали таблицю з підрахованою кількістю емоцій. Найбільше всього фотографій людей, що відображають щастя – 8989 зображень. На другому місці нейтральні обличчя – 6198, потім обличчя,

що виражають сум – 6077 штук, страх – 5121, злість – 4953, подив – 4002 і найменше всього зображень з огидою – 547.

Графічно кількість емоцій можна зобразити у вигляді гистограми (рисунок 8)

Далі будемо працювати з тими емоціями, яких більше всього, а саме: щастя, нейтральні, сум.

Лістинг 3 Графічне зображення кількості емоцій різних типів

```
fig = pyplot.figure(1, (14, 14))

k = 0
for label in sorted(df.emotion.unique()):
    for j in range(7):
        px = df[df.emotion==label].pixels.iloc[k]
        px = np.array(px.split(' ')).reshape(48,
48).astype('float32')

        k += 1
        ax = pyplot.subplot(7, 7, k)
        ax.imshow(px, cmap='gray')
        ax.set_xticks([])
        ax.set_yticks([])
        ax.set_title(emotion_label_to_text[label])
pyplot.tight_layout()
```

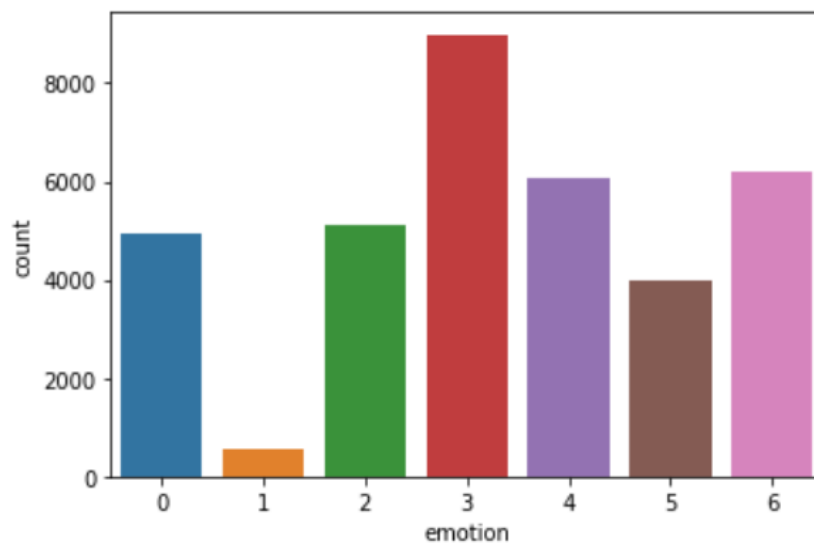


Рис. 8 Кількість емоцій у датасеті

Отже, зображення у датасеті, розподілені за виразом обличчя мають наступний вигляд (рис. 9, 10):



Рис. 9 Зображення емоцій: злість, огида та страх



Рис. 10 Зображення емоцій: щастя, сум, подив та нейтральні

3.4 Розподіл даних для навчання та перевірки

У процесі розробки нейронної мережі відбувається поділ даних на набір для навчання та перевірки.

Лістинг 4 Розподіл тренувальних масивів

```
X_train, X_valid, y_train, y_valid =
train_test_split(img_array, img_labels,
shuffle=True, stratify=img_labels,
test_size=0.1, random_state=42)
```

```
X_train.shape, X_valid.shape, y_train.shape,  
y_valid.shape  
print(X_train)
```

Отримаємо ось такі масиви:

```
[[[215.]  
 [112.]  
 [ 45.]  
 ...  
 [ 54.]  
 [ 51.]  
 [ 63.]]
```

```
[[183.]  
 [ 72.]  
 [ 42.]  
 ...  
 [ 47.]  
 [ 54.]  
 [ 52.]]
```

```
[[151.]  
 [ 46.]  
 [ 56.]  
 ...  
 [ 50.]  
 [ 49.]  
 [ 49.]]
```

```
....
```

```
del df
del img_array
del img_labels
In [18]:
img_width = X_train.shape[1]
img_height = X_train.shape[2]
img_depth = X_train.shape[3]
num_classes = y_train.shape[1]
```

Також необхідно зробити нормалізацію результатів, оскільки нейронні мережі дуже чутливі до ненормалізованих даних.

Лістинг 5 Нормалізація результатів

```
X_train = X_train / 255.
X_valid = X_valid / 255.
```

3.5 Розробка глибокої згорткової нейронної мережі

Згорткові нейронні мережі це чудовий спосіб структурування в мережі, побудови так званої інваріантності (виділення незмінних ознак).

Згорткові нейронні мережі досить щільно використовуються при роботі з аудіо та відео, включаючи медичні зображення. Також вони використовуються в мовних технологіях, де фахівці використовують глибоке навчання для розуміння та відтворення мовних конструкцій.

Двовимірна згортка – це згортка, де ядро має таку саму глибину, як і вхідні дані, тому, теоретично, вам не потрібно вказувати глибину ядра, якщо ви знаєте глибину вхідних даних.

Використовуючи TensorFlow, потрібно лише вказати ширину та висоту ядра в класі Conv2D, оскільки глибина автоматично обчислюється на основі глибини введення. Таким чином, 2 походить від того, що ви ковзаєте між двома вимірами (тобто шириною та висотою).

Лістинг 6 Розробка згорткової мережі

```
net = Sequential(name='DCNN')

net.add(
    Conv2D(
        filters=64,
        kernel_size=(5,5),
        input_shape=(img_width, img_height, img_depth),
        activation='elu',
        padding='same',
        kernel_initializer='he_normal',
        name='conv2d_1'
    )
)

net.add(BatchNormalization(name='batchnorm_1'))

net.add(
    Conv2D(
        filters=64,
        kernel_size=(5,5),
        activation='elu',
        padding='same',
        kernel_initializer='he_normal',
        name='conv2d_2'
    )
)
```

```
net.add(BatchNormalization(name='batchnorm_2'))

net.add(MaxPooling2D(pool_size=(2,2),
name='maxpool2d_1'))
net.add(Dropout(0.4, name='dropout_1'))

net.add(
    Conv2D(
        filters=128,
        kernel_size=(3,3),
        activation='elu',
        padding='same',
        kernel_initializer='he_normal',
        name='conv2d_3'
    )
)
net.add(BatchNormalization(name='batchnorm_3'))
net.add(
    Conv2D(
        filters=128,
        kernel_size=(3,3),
        activation='elu',
        padding='same',
        kernel_initializer='he_normal',
        name='conv2d_4'
    )
)
net.add(BatchNormalization(name='batchnorm_4'))
```

```
net.add(MaxPooling2D(pool_size=(2,2),
name='maxpool2d_2'))
net.add(Dropout(0.4, name='dropout_2'))

net.add(
    Conv2D(
        filters=256,
        kernel_size=(3,3),
        activation='elu',
        padding='same',
        kernel_initializer='he_normal',
        name='conv2d_5'
    )
)
net.add(BatchNormalization(name='batchnorm_5'))
net.add(
    Conv2D(
        filters=256,
        kernel_size=(3,3),
        activation='elu',
        padding='same',
        kernel_initializer='he_normal',
        name='conv2d_6'
    )
)
net.add(BatchNormalization(name='batchnorm_6'))

net.add(MaxPooling2D(pool_size=(2,2),
name='maxpool2d_3'))
```



```
net.add(Dropout(0.5, name='dropout_3'))

net.add(Flatten(name='flatten'))

net.add(
    Dense(
        128,
        activation='elu',
        kernel_initializer='he_normal',
        name='dense_1'
    )
)

net.add(BatchNormalization(name='batchnorm_7'))

net.add(Dropout(0.6, name='dropout_4'))

net.add(
    Dense(
        num_classes,
        activation='softmax',
        name='out_layer'
    )
)

net.compile(
    loss='categorical_crossentropy',
    optimizer=optim,
    metrics=['accuracy']
)
```

```
net.summary()

return net
```

Модель розробленої нейронної мережі для кодувальника та декодувальника має наступний вигляд – вхідний шар отримує зображення розміром $N \times M$.

Далі відбувається чергування шарів: згортковий шар (Conv2D) з фільтрами F_1 з ядром $x_1 \times y_1$, шар субдискретизації (MaxPooling2D) з ядром $k_1 \times l_1$, на якому відбувається зменшення розмірів вхідного зображення $\frac{N}{k_1} \times \frac{M}{l_1}$, наступний шар – згортковий з фільтрами F_2 з ядром $x_2 \times y_2$, шар субдискретизації з ядром $k_2 \times l_2$, після цього зображення має розміри $\frac{N}{k_1 k_2} \times \frac{M}{l_1 l_2}$, і т. д. Кількість необхідних шарів можна визначити експериментальним способом. Наступним етапом є кодування декодувальника, тобто розробка шарів транспонованої згортки (Conv2DTranspose). Кількість шарів декодувальника повинна відповідати кількості шарів кодувальника. У шарах транспонованої згортки відбувається зворотній процес – накладаються фільтри і розміри зображення збільшуються, тобто на виході отримуємо зображення розміром $N \times M$.

Під час обчислювального експерименту була розроблена модель штучної нейронної мережі, яка на вході отримує зображення розміром 48×48 . Далі чергуються внутрішні шари згортки Conv2D з шарами субдискретизації MaxPooling2D, у яких на зображення накладаються фільтри з ядром певного розміру. Кількість шарів Conv2D дорівнює 4. В останньому шарі кодувальника зображення має розмір 6×6 .

Функцією активації у кодувальнику вибрано ELU. Далі йдуть 4 шари декодувальника Conv2DTranspose, в яких розмір зображення збільшується поетапно з 6×6 до 48×48 . Тобто на виході отримаємо зображення, розмір якого співпадає з розміром зображення на вході. Приклад роботи зображено на рисунках 11 – 13.

```

Model: "DCNN"
-----
Layer (type)                Output Shape                Param #
-----
conv2d_1 (Conv2D)           (None, 48, 48, 64)         1664
-----
batchnorm_1 (BatchNormalizat (None, 48, 48, 64)         256
-----
conv2d_2 (Conv2D)           (None, 48, 48, 64)         102464
-----
batchnorm_2 (BatchNormalizat (None, 48, 48, 64)         256
-----
maxpool2d_1 (MaxPooling2D)  (None, 24, 24, 64)         0
-----
dropout_1 (Dropout)         (None, 24, 24, 64)         0
-----
conv2d_3 (Conv2D)           (None, 24, 24, 128)        73856
-----
batchnorm_3 (BatchNormalizat (None, 24, 24, 128)        512
-----

```

Рис. 11 Приклад конфігурації згорткової нейронної мережі

Завдяки використанню загальних синаптичних вагових коефіцієнтів в площинах згорткових нейронних мереж досягається зменшення кількості параметрів, що настраюються.

При використанні класичних нейронних мереж кожен ділянку зображення розраховується окремо від усіх інших ділянок, це призводить до великих обсягів обчислень і зниження швидкості роботи системи. У згорткових нейронних мережах даний недолік усувається використанням загальних синаптичних коефіцієнтів і так званого методу загальних обчислень. Локальні рецептивні поля згорткових верств частково нашаровуються один на одного і мають загальні значення нейронів.

Тому немає необхідності обчислювати значення всіх нейронів локального рецептивного поля, а використовувати значення, обчислені на попередніх ітераціях. Використання цього методу дозволяє значно знизити час і обсяг обчислень.

conv2d_4 (Conv2D)	(None, 24, 24, 128)	147584
batchnorm_4 (BatchNormalizat	(None, 24, 24, 128)	512
maxpool2d_2 (MaxPooling2D)	(None, 12, 12, 128)	0
dropout_2 (Dropout)	(None, 12, 12, 128)	0
conv2d_5 (Conv2D)	(None, 12, 12, 256)	295168
batchnorm_5 (BatchNormalizat	(None, 12, 12, 256)	1024
conv2d_6 (Conv2D)	(None, 12, 12, 256)	590080
batchnorm_6 (BatchNormalizat	(None, 12, 12, 256)	1024

Рис. 12 Приклад конфігурації згорткової нейронної мережі (продовження)

maxpool2d_3 (MaxPooling2D)	(None, 6, 6, 256)	0
dropout_3 (Dropout)	(None, 6, 6, 256)	0
flatten (Flatten)	(None, 9216)	0
dense_1 (Dense)	(None, 128)	1179776
batchnorm_7 (BatchNormalizat	(None, 128)	512
dropout_4 (Dropout)	(None, 128)	0
out_layer (Dense)	(None, 3)	387
Total params: 2,395,075		
Trainable params: 2,393,027		
Non-trainable params: 2,048		

Рис. 13 Приклад конфігурації згорткової нейронної мережі (продовження)

3.6 Висновки з розділу 3

У даному розділі запропоновано неромережеві моделі для емоцій людини по заданому зображенню.

Проведено обчислювальні експерименти з базовими архітектурами нейромереж. Проаналізовано вплив функцій активації та алгоритмів активації на штучні нейронні мережі. Проведені дослідження на вплив кількості нейронів на нейромережеве визначення емоцій.

Розроблено нейронну мережу для визначення емоцій людини.

Розроблено модель штучної згорткової нейронної мережі автокодувальника для прогнозування емоцій.

РОЗДІЛ 4 ДОСЛІДЖЕННЯ РЕЗУЛЬТАТІВ РОБОТИ КОМП'ЮТЕРНОЇ СИСТЕМИ ДЛЯ ВИЗНАЧЕННЯ ПОЛОЖЕННЯ ЛЮДИНИ В ОБМЕЖЕНОМУ ПРОСТОРИ У РЕАЛЬНОМУ ЧАСІ

4.1 Результати навчання моделей нейронних мереж для розпізнавання емоцій

4.1.1 Запуск процесу навчання

У процесі розробки нейронної мережі для прогнозування емоцій було проведення навчання її на датасеті, що містить 28709 зображень людей. Люди на фотографіях мають різний вік, різну національність, у всіх людей різна довжина волосся й різного кольору. Також на зображеннях всі люди мають різний нахил та розворот голови. У деяких людей є макіяж на обличчі, окуляри або головні убори.

Процес навчання нейромережі починається з тренування на датасеті `train.csv`. Даний датасет містить два стовпці: "емоції" та "пікселі". Стовпець «емоція» містить числовий код від 0 до 6 включно для емоції, яка присутня на зображенні. Стовпець «пікселі» містить рядок у лапках для кожного зображення. Вміст цього рядка – значення пікселів, розділених пробілами, у порядку старшого рядка.

Після тренування розпочинається процес прогнозування. Нейронній мережі пропонується датасет `test.csv`, що містить лише стовпець «пікселі», а її завдання – передбачити стовпець емоцій.

4.1.2 Обчислювальний експеримент

Зазвичай дослідження проводяться з великими обсягами даних, які вимагають витрат ресурсів і часу. Щоб ефективно працювати з великими обсягами

даних, потрібно використовувати параметри (епоха, розмір партії, ітерації), так як часто немає можливості завантажити відразу всі дані в обробку.

Для подолання цієї проблеми дані поділяються на частину меншого розміру, завантажують їх по черзі і оновлюють ваги нейромережі в кінці кожного кроку, підстроюючи їх під дані.

У термінології нейронних мереж епоха (epoch) – одна ітерація у процесі тренування (отримання вихідних значень та оновлення ваг всіх тренувальних екземплярів). Одна епоха (epoch) – весь датасет пройшов через нейронну мережу у прямому та зворотному напрямку лише один раз.

Для отримання певних результатів недостатньо одного повного проходу даних через нейронну мережу. Необхідно пустити повний набір даних по мережі кілька разів.

Також потрібно врахувати, що використовується обмежений набір даних, щоб оптимізувати навчання та створити криву інформацію. Це робиться за допомогою градієнтного спуску – ітеративного процесу. Тому оновлення ваг після одного проходження недостатньо.

Одна епоха призводить до недонавчання, а велика кількість епох – до перенавчання.

Зі збільшенням числа епох, ваги нейронної мережі змінюються дедалі більше разів. Крива щораз краще підлаштовується під дані, переходячи послідовно з погано навченого стану в оптимальне. Якщо вчасно не зупинитися, то може відбутися перенавчання – коли крива дуже точно підлаштувалася, а узагальнююча здатність зникла.

В ході виконання роботи по навчанню нейронної мережі для розпізнавання емоцій було проведено декілька обчислювальних експериментів з різною кількістю епох. В перший раз було проведено експеримент з 10, потім з 15 епохами, що зайняло 3 години роботи комп'ютера. В третій раз був проведений

обчислювальний експеримент з 20 епохами. Даний експеримент зайняв 4 години роботи комп'ютера.

Отримали наступні результати, як зображено на графіках (рис. 12), де синім кольором зображено тренування нейронної мережі, червоний відображає прогнозування нейронної мережі:

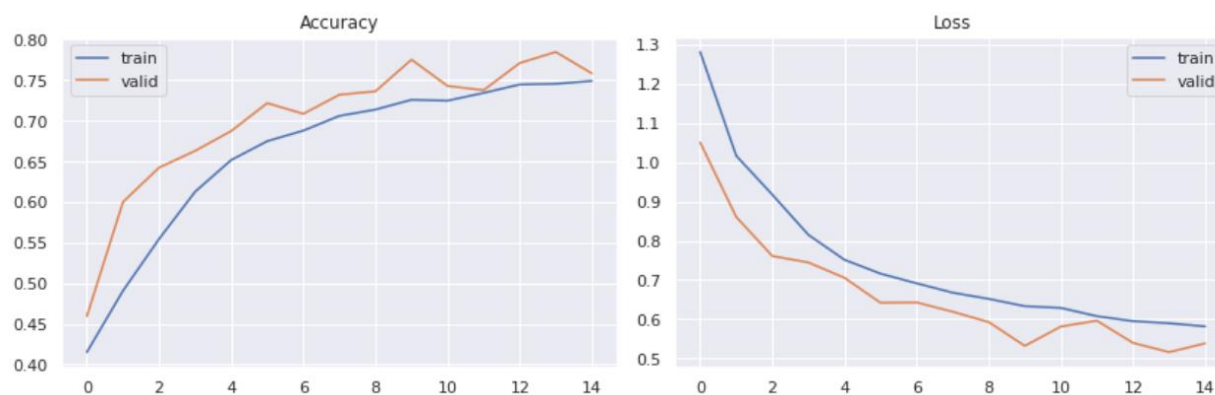


Рис. 14 Порівняння результатів роботи згорткової нейронної мережі на 15 епохах

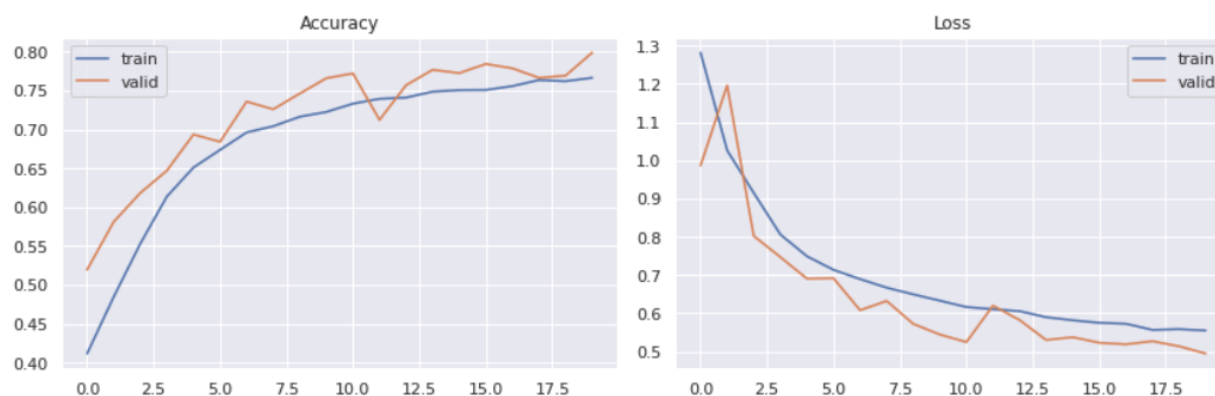


Рис. 15 Порівняння результатів роботи згорткової нейронної мережі на 20 епохах

Зміна показників на графіках свідчить про те, що точність прогнозування досягає 80%. Але після 15 епох відбувається перенавчання нейронної мережі.

На 10 епохах графіки показали, що нейронна мережа недонавчена і потрібно збільшити кількість епох.

4.1.3 Результати роботи нейронної мережі

В результаті роботи нейронної мережі отримали графік кореляції (рис. 16), а також з отриманих результатів можна побачити, що розроблена нейронна мережа зробила 428 зовсім неправильних прогнозувань.

Лістинг 7 Графік залежності параметрів

```
yhat_valid = savedModel.predict_classes(X_valid)
print(yhat_valid)
scikitplot.metrics.plot_confusion_matrix(np.argmax(y_valid,
axis=1), yhat_valid, figsize=(7,7))
pyplot.savefig("confusion_matrix_dcnn.png")

print(f'total wrong validation predictions:
{np.sum(np.argmax(y_valid, axis=1) != yhat_valid)}\n\n')
print(classification_report(np.argmax(y_valid, axis=1),
yhat_valid))
[1 2 2 ... 0 2 0]
total wrong validation predictions: 428
```

	precision	recall	f1-score	support
0	0.93	0.89	0.91	899
1	0.75	0.68	0.71	608

2	0.67	0.78	0.72	620
accuracy			0.80	2127
macro avg	0.79	0.78	0.78	2127
weighted avg	0.81	0.80	0.80	2127

Отримана матриця чітко показує, що розроблена модель добре справляється з визначенням емоцій "Щасливий", але її продуктивність низька для інших двох емоцій. Однією з причин цього може бути те, що ці два класи мають менше даних. Але після власного перегляду зображень, виявилось, що на деяких зображеннях із цих двох класів дуже важко визначити, сумна ця людина чи нейтральна. Вираз обличчя також залежить від людини. Чиєсь нейтральне обличчя виглядає сумним. Тому нейронна мережа показала дуже хороші результати.

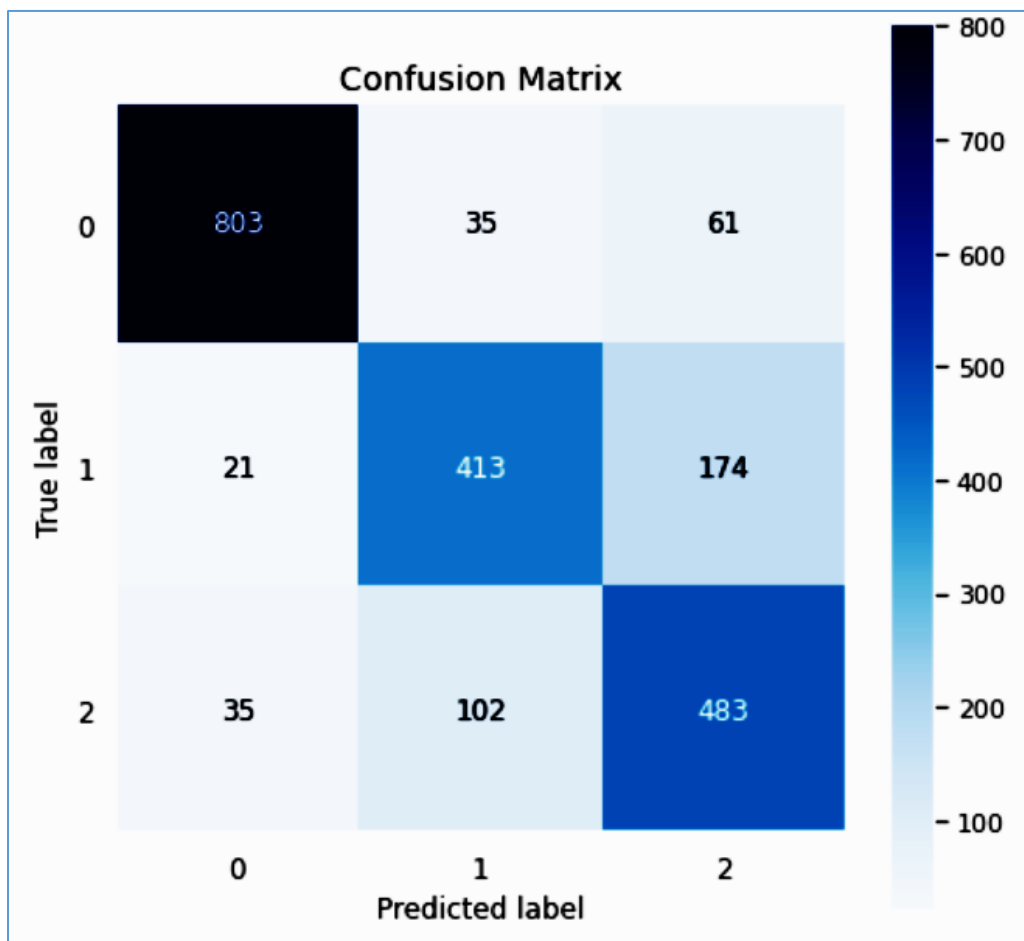


Рис. 16 Графік залежності параметрів нейронної мережі

Лістинг 7 Результати роботи нейронної мережі

```

np.random.seed(2)
random_sad_imgs = np.random.choice(np.where(y_valid[:,
1]==1)[0], size=9)
random_neutral_imgs = np.random.choice(np.where(y_valid[:,
2]==1)[0], size=9)

fig = pyplot.figure(1, (18, 4))

for i, (sadidx, neuidx) in enumerate(zip(random_sad_imgs,
random_neutral_imgs)):

```

```
ax = pyplot.subplot(2, 9, i+1)
sample_img = X_valid[sadidx, :, :, 0]
ax.imshow(sample_img, cmap='gray')
ax.set_xticks([])
ax.set_yticks([])
ax.set_title(f"true:sad,
pred:{mapper[model.predict_classes(sample_img.reshape(1, 48, 48, 1))][0]}",
```

```
ax = pyplot.subplot(2, 9, i+10)
sample_img = X_valid[neuidx, :, :, 0]
ax.imshow(sample_img, cmap='gray')
ax.set_xticks([])
ax.set_yticks([])
ax.set_title(f"t:neut,
p:{mapper[model.predict_classes(sample_img.reshape(1, 48, 48, 1))][0]}",
```

```
pyplot.tight_layout()
```

Отримали наступні результати (рис. 17):



Рис. 17 Результат роботи нейронної мережі



Рис. 18 Результат роботи нейронної мережі (продовження)



Рис. 19 Результат роботи нейронної мережі (продовження)

Можна зробити висновки, що на рисунку 18, фото №1 зображення виглядає скоріше нейтральним, ніж сумним, і розроблена модель навіть передбачила, що воно нейтральне. Тоді як 2 зображення на рисунку 18 дуже сумне. Також нейронна мережа зробила помилку у зображеннях №1, 2, 3 на рисунку 19, передбачивши, що там зображені нейтральні емоції замість сумних.

4.2 Висновки з розділу 4

Розпізнавання зображень знаходить широке застосування в різних областях: це може бути класифікація об'єктів, розпізнавання осіб, символів, маркування об'єктів, автомобільних номерних знаків, відбитків пальців і т. д. На сьогоднішній день дані технології застосовуються в мобільних телефонах, планшетних комп'ютерах, системах контролю доступу до секретних об'єктах, оперативному пошуку в картотеці і т. д. Однак до сих пір при вирішенні задач в цій області виникає ряд складних наукових і технічних проблем.

Досліджені результати навчання та точність навчених моделей нейронних мереж: модель, якій було задано 15 епох показала більшу точність при навчанні ніж нейромережі з 10 та 20 епохами навчання.

Отже, розроблена штучна нейронна мережа дозволяє прогнозувати емоції людини. Абсолютна відсоткова похибка становить не більше 20%. Точність розробленої системи залежить від якості навчання моделі нейронної мережі, яка у свою чергу залежить від якості датасету.

ВИСНОВКИ

1. Досліджена проблема визначення емоцій людини: розглянуті переваги та недоліки існуючих технологій та систем для її вирішення, сфери застосування таких систем та сучасні підходи комп'ютерного зору до її розв'язання.

2. Досліджені засоби для вирішення поставленої проблеми – розглянуті моделі нейронних мереж, придатні для розпізнавання емоцій людини по зображенню обличчя.

3. Проведено навчання моделі нейронної мережі для розпізнавання емоцій людини: налаштоване середовище для навчання, підготовлені зображення людей з відповідного датасету, створені необхідні для тренування файли та успішно виконано навчання моделі.

4. Розроблена готова до впровадження комп'ютерна система, яка визначає емоції людини по зображенню обличчя.

5. Досліджені результати навчання та точність навчених моделей нейронних мереж: модель, якій було задано 15 епох навчання показала більшу точність при навчанні ніж нейронні мережі з 10 та 20 епохами.

6. Розроблена штучна нейронна мережа дозволяє прогнозувати емоції людини. Абсолютна відсоткова похибка становить не більше 20%. При цьому виявлено, що точність розробленої системи залежить від якості навчання моделі нейронної мережі, яка у свою чергу залежить від якості датасету.

7. Поставлена проблема визначення емоцій людини повністю вирішена з допустимою точністю.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Неалова А. О., магістрант, Безверхий А. І., доцент - науковий керівник. Особливості алгоритмів опрацювання зображень для нейромережної системи розпізнавання емоцій. *Молода наука-2022* : зб. наук. праць студентів, аспірантів і молодих вчених. Запоріжжя : ЗНУ, 2022. Т. 5. С. 178.

2. Неалова А. О., магістрант, Безверхий А. І., доцент - науковий керівник. Нейромережна система розпізнавання емоцій. 18-20 жовтня 2022 року ІІ Всеукраїнська науково-практична конференція за участю молодих науковців «АКТУАЛЬНІ ПИТАННЯ СТАЛОГО НАУКОВО-ТЕХНІЧНОГО ТА СОЦІАЛЬНО-ЕКОНОМІЧНОГО РОЗВИТКУ РЕГІОНІВ УКРАЇНИ». Запоріжжя : ЗНУ, 2022. С. 156.

3. Неалова А. О., Безверхий А. І., Нейромережна система розпізнавання емоцій. Міжнародна науково-практична конференція Інженерного навчально-наукового інституту ім. Ю.М. Потебні Запорізького національного університету «Перспективи сталого розвитку в умовах глобалізації в економічному, управлінському та інженерному аспектах» 3 – 4 листопада 2022 р. С. 249–250.

4. Акулов П.В. Решение задач прогнозирования с помощью нейронных сетей. URL: www.dgtu.donetsk.ua (дата звернення 21.11.2022).

5. Оссовский С. Нейронные сети для обработки информации. Пер. с польского И.Д. Рудинского. Москва : Финансы и статистика, 2002. 344 с.

6. Кальченко Д. Нейронные сети: на пороге будущего. ж. КомпьютерПресс 2005. N1. URL : [http //www.compr.ru](http://www.compr.ru) (дата звернення: 21.11.2022).

7. Rosenblatt, Frank (1957), The Perceptron - a perceiving and recognizing automaton. Report 85 – 460 – 1, Cornell Aeronautical Laboratory.

8. A logical calculus of the ideas immanent in nervous activity // Bulletin of Mathematical Biology. New York : Springer New York, 1943. T. 5, № 4. C. 115.

9. Donald Olding Hebb. The Organization of Behavior: A Neuropsychological Theory. Wiley, 1949. 335 p.

10. Fukushima K., Miyake S., Takayuki I. Neocognitron: A neural network model for a mechanism of visual pattern recognition. IEEE Transaction on Systems, Man and Cybernetics SMC 13(5):826 - 34. 1983.

11. Fukushima K., Miyake S., Takayuki I. Neocognitron: A neural network model for a mechanism of visual pattern recognition. IEEE Transaction on Systems, Man and Cybernetics SMC 13(5):826 - 34. 1983.

12. Lowe D. (1999). «Object recognition from local scale-invariant features» (PDF). Proceedings of the International Conference on Computer Vision. 2. P. 1150.

13. Dalal N., Triggs B.(2005) Histograms of Oriented Gradients for Human Detection. In proc. Of CVPR conference.

14. Vapnik, Vladimir N. The Nature of Statistical Learning Theory, New York : Springer-Verlag, 1995. P. 314.

15. «The MegaFace Benchmark: 1 Million Faces for Recognition at Scale», Ira Kemelmacher-Shlizerman, Steve Seitz, Daniel Miller, Evan Brossard, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016

16. Mahony N., Campbell S., Carvalho A. Deep Learning vs. Traditional Computer Vision. *Computer Vision Conference* : conference materials, Las Vegas, 25 - 26 Apr. 2019. Las Vegas, 2019. Vol. 1. P. 128.

17. Dalal N., Triggs B. Histograms of oriented gradients for human detection. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* : conference materials, San Diego, 20 - 25 June 2005. San Diego, 2005. P. 886 .

18. LeCun Y., Bengio Y., Haffner P. Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*. 1998. Vol. 86, No 11. P. 2278.

19. Krizhevsky A., Sutskever I., Hinton G. ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems*. 2012. Vol. 25, No 2. P. 1097.

20. Girshick R., Donahue J., Darrell T., Malik J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. *IEEE Conference on Computer Vision and Pattern Recognition* : conference materials, Columbus, 23-28 June 2014. Columbus, 2014. P. 580.

21. Girshick R. Fast R-CNN. *IEEE International Conference on Computer Vision* : conference materials, Santiago, 7-13 Dec. 2015. Santiago, 2015. P. 1440.

22. Shaoqing R., He K., Girshick R., Sun J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2016. Vol. 39, No 6. P. 1137.

23. Redmon J., Divvala S., Girshick R., Farhadi A. You Only Look Once: Unified, Real-Time Object Detection. *IEEE Conference on Computer Vision and Pattern Recognition* : conference materials, Las Vegas, 27-30 June 2016. Las Vegas, 2016. P. 779.

24. Bochkovski A., Wang C., Liao H. YOLOv4: Optimal Speed and Accuracy of Object Detection. 2020. URL: <https://arxiv.org/pdf/2004.10934.pdf> (дата звернення: 25.09.2022).

25. Convolutional Neural Networks from the ground up. URL: <https://towardsdatascience.com/convolutional-neural-networks-from-the-ground-up-c67bb41454e1> (дата звернення: 18.11.2022).

26. Liu W., Anguelov D., Erhan D. SSD: Single Shot MultiBox Detector. *European Conference on Computer Vision* : conference materials, Amsterdam, 11-14 Oct. 2016. Amsterdam, 2016. P. 2.

27. Howard A., Sandler M., Chu G. Searching for MobileNetV3. *IEEE/CVF International Conference on Computer Vision* : conference materials, Seoul, 27 Oct.-2 Nov. 2019. Seoul, 2019. P. 1314.

28. Depthwise Separable Convolution. URL:
<https://www.youtube.com/watch?v=T7o3xvJLuHk> (дата звернення: 20.11.2022).

29. Iandola F., Han S., Moskewicz M. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size. 2016. URL:
<https://arxiv.org/pdf/1602.07360.pdf> (дата звернення: 21.11.2022).

30. Нейросети: куда это все движется. URL:
<https://habr.com/ru/post/482794/> (дата звернення: 21.11.2022).

31. Simonyan K., Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *International Conference on Learning Representations* : conference materials, San Diego, 7-9 May 2015. San Diego, 2015.

32. Fang W., Wang L., Ren P. Tinier-YOLO: A Real-time Object Detection Method for Constrained Environments. *IEEE Access*. 2019. Vol. 8. P. 1935.

33. Kuznetsova A., Rom H., Alldrin N. The Open Images Dataset V4: Unified Image Classification, Object Detection, and Visual Relationship Detection at Scale. *International Journal of Computer Vision*. 2020. Vol. 128, No 4. P. 1956.

34. Braun M., Krebs S., Flohr F., Gavrila D. EuroCity Persons: A Novel Benchmark for Person Detection in Traffic Scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2019. Vol. 41, No 8. P. 1844.

35. Yu F., Chen H., Wang X. BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning. *IEEE/CVF Conference on Computer Vision and Pattern Recognition* : conference materials, Seattle, 13-19 June 2020. Seattle, 2020. P. 263

Декларація
академічної доброчесності
здобувача ступеня вищої освіти ЗНУ

Я, Неалова Анастасія Олександрівна, студентка 2 курсу магістратри, денної форми навчання, Інженерного навчально-наукового інституту ім. Ю.М. Потебні, спеціальність 121 Інженерія програмного забезпечення, адреса електронної пошти ipz17bd-19@stu.zsea.edu.ua, - підтверджую, що написана мною кваліфікаційна робота на тему «**Нейромережна система для визначення емоцій**» відповідає вимогам академічної доброчесності та не містить порушень, що визначені у ст.42 Закону України «Про освіту», зі змістом яких ознайомена.

- заявляю, що надана мною для перевірки електронна версія роботи є ідентичною її друкованій версії;
- згодна на перевірку моєї роботи на відповідність критеріям академічної доброчесності у будь-який спосіб, у тому числі за допомогою інтернет-системи, а також на архівування моєї роботи в базі даних цієї системи.

Дата 30.11.2022 _____ Неалова Анастасія Олександрівна
(студентка)

Дата 30.11.2022 _____ Безверхий Анатолій Ігорович
(науковий керівник)