

МАГІСТЕРСЬКА РОБОТА

МР. ШМ - 57.00.00.000 ПЗ

Група ШМ-23-3

Родик Арсен

2024

Івано-Франківський національний технічний університет нафти і газу

Інститут інформаційних технологій

Кафедра інженерії програмного забезпечення

Родик Арсен Васильович

(прізвище, ім'я, по батькові)

УДК 004.942
(індекс)

МАГІСТЕРСЬКА РОБОТА

Моделі, методи та засоби машинного навчання для ідентифікації

елементів зображень

(назва роботи)

Інженерія програмного забезпечення

(назва освітньої програми)

121 - Інженерія програмного забезпечення

(шифр і назва спеціальності)

Родик А.В.

(підпис, ініціали та прізвище здобувача освітнього ступеня)

Науковий керівник Храбатин Роман Ігорович, к.ф.-м.н., доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Допущено до захисту

Завідувач кафедри

доц. **Бандура В.В.**

(посада) (підпис) (дата) (ініціали та прізвище)

Нормоконтроль

доц. **Вовк Р.Б.**

(посада) (підпис) (дата) (ініціали та прізвище)

Робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

Івано-Франківськ – 2024

Івано-Франківський національний технічний університет нафти і газу

Інститут інформаційних технологій

Кафедра інженерії програмного забезпечення

Освітній рівень магістр

Спеціальність 121 – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:

Зав. кафедрою

ІІЗ

доц.

В.В. Бандура

“ 04 ” вересня 2024 р.

ЗАВДАННЯ

НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

Родику Арсену Васильовичу

(прізвище, ім'я, по-батькові)

1. Тема магістерської роботи “ Моделі, методи та засоби машинного навчання для ідентифікації елементів зображень ”

керівник проекту (роботи) Храбатин Роман Ігорович, к.ф.-м.н., доцент

затверджені наказом закладу вищої освіти від “ 22 ” листопада 2024 р. № 781/7

2. Строк подання студентом проекту (роботи) 15 грудня 2024 р.

3. Вихідні дані до проекту (роботи) Теоретичні концепції та формальні моделі побудови та функціонування інформаційних технологій машинного навчання

4. Зміст розрахунково - пояснювальної записки(перелік питань, які потрібно розробити)

1. Аналіз предметної області застосування машинного навчання для маркування зображень

2. Моделі та методи машинного навчання для ідентифікації елементів на зображеннях

3. Дослідження алгоритмів глибокого навчання

4. Застосування моделей машинного навчання для ідентифікації елементів зображень

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

1. Еволюція штучного інтелекту (рис. 1.1)

2. Ілюстрація архітектури CNN, показуючи розподіл відповідальності між двома GPU (рис. 1.2)

3. Класифікація алгоритмів глибокого навчання (рис. 1.3)

4. Високорівнева абстракція постановки проблеми (рис. 1.4)

5. Джерело отриманих наборів даних для проведення дослідження та експериментів (рис. 1.5)

6. Консультанти розділів проекту (роботи)

Розділ	Консультант	Підпис, дата
Перевірка на плагіат	доц., к.т.н. Вовк Р.Б.	

7. Дата видачі завдання 04 вересня 2024 р.

Керівник _____
(підпис)

Завдання прийняв до виконання _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назви етапів магістерської роботи	Строк виконання етапів роботи	Примітка
1	Підбір і вивчення літератури по темі магістерської роботи	15.09.2024	виконано
2	Аналіз концепцій та алгоритмів предметної області	29.09.2024	виконано
3	Дослідження предметної області застосування машинного навчання для маркування зображень	15.10.2024	виконано
4	Моделі та методи машинного навчання для ідентифікації елементів на зображеннях	08.11.2024	виконано
5	Дослідження алгоритмів глибокого навчання	20.11.2024	виконано
6	Застосування моделей машинного навчання для ідентифікації елементів зображень	01.12.2024	виконано
7	Затвердження пояснювальної записки роботи завідувачем кафедри	15.12.2024	виконано

Студент – магістр _____
(підпис)

Керівник роботи _____
(підпис)

АНОТАЦІЯ

Магістерська робота: 82 с., 29 рис., 2 табл., 49 джерел.

Тема: Моделі, методи та засоби машинного навчання для ідентифікації елементів зображень

Об'єкт дослідження: процеси ідентифікації елементів на зображеннях із використанням методів машинного навчання.

Мета роботи: використання моделей і методів машинного навчання для підвищення ефективності ідентифікації елементів на зображеннях.

Предмет дослідження: моделі, методи і алгоритми машинного навчання, що використовуються для автоматизованої обробки зображень.

Результати дослідження

В роботі Визначено критерії відбору зображень, які підвищують точність класифікації і адаптовано архітектури рекурентних нейронних мереж для вирішення задач ідентифікації статичних зображень.

Висновок

Виконано комплексне дослідження застосування методів машинного навчання для ідентифікації та маркування елементів на зображеннях. Проведена робота охоплює широкий спектр аспектів від аналізу предметної області до впровадження та оцінювання практичних результатів.

МАШИННЕ НАВЧАННЯ, ОБРОБКА ЗОБРАЖЕНЬ, ЗГОРТКОВІ НЕЙРОННІ МЕРЕЖІ, ІДЕНТИФІКАЦІЯ ЕЛЕМЕНТІВ, ГЛИБОКЕ НАВЧАННЯ, РЕКУРЕНТНІ МЕРЕЖІ, АВТОМАТИЗОВАНЕ МАРКУВАННЯ.

ABSTRACT

Master Thesis: 82 pp., 29 fig., 2 tab., 49 sources.

Thesis Subject: Machine learning models, methods and tools for identifying image elements

Object of the study: processes of identifying elements in images using machine learning methods.

Purpose of the work: use of machine learning models and methods to improve the efficiency of identifying elements in images.

Subject of the study: machine learning models, methods and algorithms used for automated image processing.

Research results

The work defines image selection criteria that increase classification accuracy and adapts recurrent neural network architectures to solve static image identification problems.

Conclusion

A comprehensive study of the application of machine learning methods for identifying and labeling elements in images has been carried out. The work covers a wide range of aspects from analysis of the subject area to implementation and evaluation of practical results.

MACHINE LEARNING, IMAGE PROCESSING, CONVULSIVE NEURAL NETWORKS, FEATURE IDENTIFICATION, DEEP LEARNING, RECURRENT NETWORKS, AUTOMATED LABELING.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	9	
ВСТУП.....	10	
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ЗАСТОСУВАННЯ МАШИННОГО НАВЧАННЯ ДЛЯ МАРКУВАННЯ ЗОБРАЖЕНЬ.....		13
1.1. Основні відомості про штучний інтелект та машинне навчання	13	
1.2. Тенденції розвитку машинного навчання в контексті обробки зображень	15	
1.3. Опис та постановка проблеми дослідження розпізнавання та ідентифікації елементів на зображеннях	20	
Висновки до розділу	26	
РОЗДІЛ 2. МОДЕЛІ ТА МЕТОДИ МАШИННОГО НАВЧАННЯ ДЛЯ ІДЕНТИФІКАЦІЇ ЕЛЕМЕНТІВ НА ЗОБРАЖЕННЯХ.....		27
2.1. Опис особливостей елементів Віденської класифікації зображень	27	
2.2. Контрольоване машинне навчання	32	
2.2.1. Алгоритм машини опорних векторів.....	34	
2.2.2. Древа рішень і випадковий ліс	36	
2.2.3 Наївний Байєс	38	
2.3. Дослідження алгоритмів глибокого навчання.....	38	
2.3.1. Згорткова нейронна мережа	42	
2.3.2. Рекурентна нейронна мережа.....	45	
2.4. Архітектури глибокого навчання	47	
2.4.1. Мережа VGG Net	48	
2.4.2. Мережа GoogLeNet.....	49	
2.4.3. Модель ResNet	49	
2.4.4. Стан практичного застосування архітектур глибокого навчання ..	50	
2.5. Фреймворки глибокого навчання.....	51	

Висновки до розділу	53
РОЗДІЛ 3. ЗАСТОСУВАННЯ МЕТОДОЛОГІЇ ТА МОДЕЛЕЙ	
МАШИННОГО НАВЧАННЯ ДЛЯ ІДЕНТИФІКАЦІЇ ЕЛЕМЕНТІВ	
ЗОБРАЖЕНЬ	
3.1. Представлення методології	55
3.1.1. Контрольні імітаційні експерименти.....	56
3.1.2. Результати та методи оцінювання	57
3.1.3. Надійність і валідність.....	58
3.2. Попередня обробка даних.....	60
3.3. Критерії відбору зображення за концептуальними кодами.....	64
3.4. Метрики оцінювання результатів машинного навчання	65
3.5. Представлення результатів та їх аналіз	67
3.5.1. Результати глибокого навчання	67
3.5.2. Згорткові нейронні мережі	67
3.5.3. Модель на основі архітектури LSTM	72
3.5.4. Рекурентна мережа	73
Висновки до розділу	76
ВИСНОВКИ	77
ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	79

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ANN - Approximate Nearest Neighbor
CNN - Convolutional Neural Network
CSV - Comma-Separated Values
DCNN - Deep Convolutional Neural Network
DL - Deep Learning
DT - Decision Trees
EUCTM - European Union Community Trademark
EUIPO - European Union Intellectual Property Office
Fast R-CNN - Fast Region-based Convolutional Network
GRU - Gated Recurrent Unit
k-NN - K-Nearest Neighbor
LBP - Local Binary Patterns
LSTM - Long Short-Term Memory
MLP - Multi-Layer Perceptron
NB - Naïve Bayes
NLP - Natural Language Processing
NNS - Nearest Neighbor Search algorithm
RCN - Recurrent Convolutional Network
ReLU - Rectified Linear Unit
RF - Random Forest
RNN - Recurrent Neural Network
SGD - Stochastic Gradient Descent
SIFT - Scale-invariant feature transformation
SNN - Siamese Neural Network
STN - Spatial Transformer Network
SVM - Support Vector Machines
TM - Trademark

ВСТУП

Актуальність теми.

Завдання автоматизованої ідентифікації елементів на зображеннях є однією з ключових проблем сучасного штучного інтелекту, яка активно досліджується завдяки стрімкому зростанню обсягів візуальної інформації. Обробка зображень знаходить застосування у багатьох сферах, таких як медицина (діагностика захворювань за медичними знімками), транспорт (системи автоматичного розпізнавання номерних знаків і пішоходів), промисловість (контроль якості виробничих процесів), безпека (системи відеоспостереження та розпізнавання облич), а також у розважальній індустрії (доповнена реальність та персоналізовані сервіси).

Використання традиційних методів обробки зображень часто є недостатнім через складність об'єктів, шум у даних і різноманітність реальних умов зйомки. Сучасні методи машинного навчання, особливо глибоке навчання, відкривають нові можливості для ефективної роботи з великими наборами даних і дозволяють досягати високої точності розпізнавання. Такі архітектури, як згорткові нейронні мережі (CNN), VGG Net, GoogLeNet, ResNet та моделі рекурентних мереж (LSTM), значно підвищили рівень автоматизації та точності у задачах класифікації та сегментації зображень.

Актуальність теми також підкріплюється практичною необхідністю розробки адаптивних і масштабованих систем, які можуть інтегруватися у різні прикладні області. Дослідження методів попередньої обробки даних, створення нових критеріїв відбору навчальних зображень та впровадження метрик для оцінки моделей є важливими для підвищення надійності ідентифікації. Це, у свою чергу, сприяє прискоренню технологічного прогресу і розширенню можливостей автоматизації в умовах сучасного цифрового світу.

Мета дослідження – використання моделей і методів машинного навчання для підвищення ефективності ідентифікації елементів на зображеннях.

Об’єкт дослідження - процеси ідентифікації елементів на зображеннях із використанням методів машинного навчання.

Предмет дослідження - моделі, методи і алгоритми машинного навчання, що використовуються для автоматизованої обробки зображень.

Завдання дослідження:

- Провести аналіз предметної області застосування машинного навчання для маркування зображень.
- Дослідити існуючі моделі та архітектури машинного навчання для ідентифікації елементів на зображеннях.
- Розробити методологію для впровадження машинного навчання у задачі обробки зображень.
- Провести експериментальне дослідження ефективності моделей і оцінити їх продуктивність за допомогою метрик точності.

Методи дослідження

- Аналіз наукових джерел та порівняльне дослідження існуючих підходів до обробки зображень.
- Використання алгоритмів контрольованого та глибокого навчання, зокрема згорткових нейронних мереж, рекурентних моделей і архітектур типу ResNet.
- Експериментальне моделювання та оцінювання ефективності моделей за допомогою метрик точності, повноти, F-міри та специфічності.

Наукова новизна отриманих результатів

Розроблено підхід до інтеграції класичних методів машинного навчання з сучасними глибокими архітектурами для покращення ідентифікації елементів на зображеннях. Удосконалено методи попередньої обробки даних для забезпечення якості навчальних наборів.

Практичне значення результатів

Застосовані моделі та методології можуть бути використані для створення автоматизованих систем у сфері комп'ютерного зору, таких як системи медичної діагностики, автоматизовані системи контролю якості, технології розпізнавання образів у транспорті та безпеці.

Структура магістерської роботи. Робота складається зі вступу, трьох розділів та висновків. Загальний обсяг роботи становить 82 сторінки, і містить 29 рисунків, 2 таблиці, список використаних джерел із 49 найменувань.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ЗАСТОСУВАННЯ МАШИННОГО НАВЧАННЯ ДЛЯ МАРКУВАННЯ ЗОБРАЖЕНЬ

1.1. Основні відомості про штучний інтелект та машинне навчання

Штучний інтелект (ШІ) — це наукова та технологічна галузь, яка за останнє десятиліття революціонізувала багато сфер нашого життя. Ідея створення інтелектуальних комп'ютерів, здатних мислити та міркувати, як люди, не нова, і вона почалася в кінці 1940-х років з дослідження Джона Маккарті. Як прихильник штучного інтелекту, він був зосереджений на пошуку зв'язку між математичною логікою та формалізацією здорового глузду людського знання та міркувань, а також на його застосуванні в комп'ютерних програмах [1]. Сфера досліджень штучного інтелекту багата видатними іменами, такими як Алан Тьюрінг, Марвін Мінскі, Джеффри Хінтон та багато інших, які присвятили свою академічну кар'єру розсуванню обмежень і меж ШІ [2].

Поле досліджень штучного інтелекту дуже багате, оскільки ми намагалися визначити галузі та застосування в різних областях. Деякі з добре відомих галузей – це логічний штучний інтелект, пошук, розпізнавання образів, знання здорового глузду та міркування, евристика, генетичне програмування, і все ж існує багато інших галузей, які слід ідентифікувати [1]. Добре відомі програми штучного інтелекту можна знайти в сферах розпізнавання мови, розуміння природної мови, комп'ютерного зору, експертних систем та ігор [3, 4]. Рисунок 1.1 нижче ілюструє еволюцію штучного інтелекту та відповідні піддомени, які виникли в цій галузі.

Машинне навчання (ML) як субдомен штучного інтелекту з'явилося в 1980-х роках, коли акцент було зміщено з математичної науки на природничу, дозволяючи комп'ютерам проводити спостереження за невизначеним світом шляхом проведення експериментів, і замість логіки результати аналізуються за допомогою статистичного підходу [2].

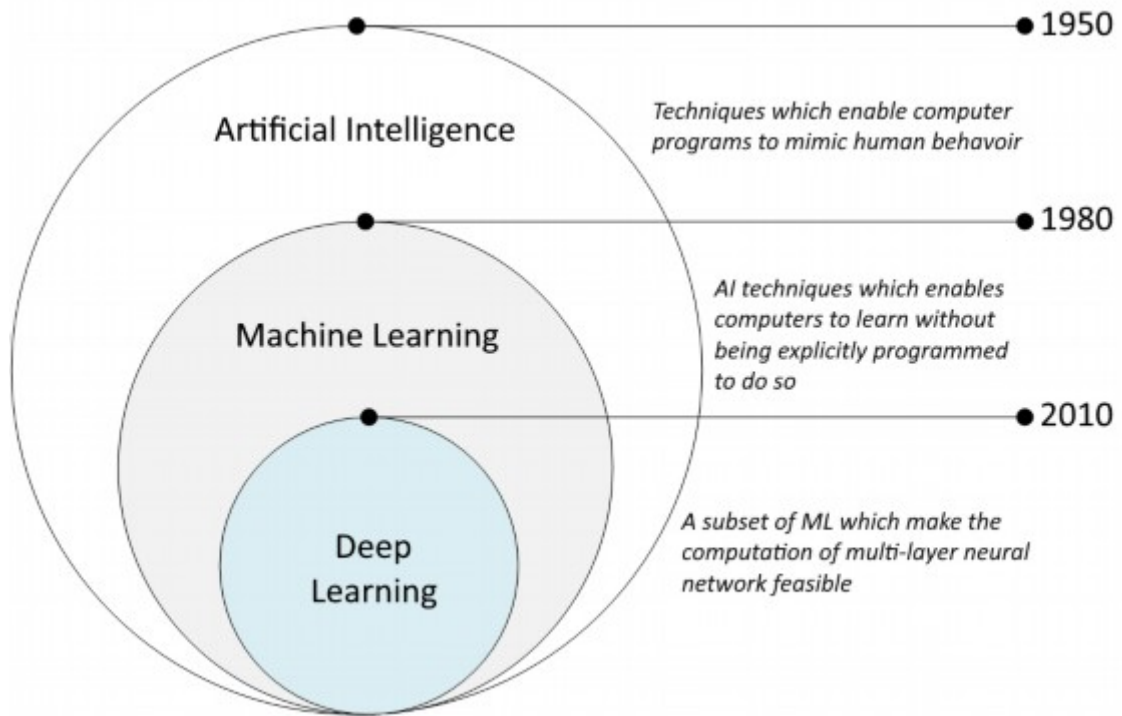


Рис. 1.1. Еволюція штучного інтелекту

У 2010 році в результаті тривалих і тривалих досліджень машинного навчання та штучного інтелекту на горизонті з'явився новий субдомен, відомий як глибоке навчання (DL). Глибоке навчання є однією з найбільш інноваційних парадигм сучасної інформатики, і її широко застосовують як дослідницьке співтовариство, так і промисловість. Концепції глибокого навчання сильно покладаються на біологічні структури людського мозку, і тому він намагається імітувати його функціональність для вирішення складних завдань. Окрім інтересу дослідницької спільноти, його широке зростання та впровадження стали можливими завдяки двом ключовим факторам, що виникають, — технологічному прогресу комп'ютерного обладнання, такого як центральні та графічні процесори, і доступності даних.

Успішне впровадження моделей машинного та глибокого навчання було використано в багатьох галузях, таких як комп'ютерне зір, обробка природної мови (NLP), обробка мови та аудіо, аналіз соціальних мереж, фінанси, автомобільна промисловість, медицина та багато інших [4]. Потенційно успішне застосування моделей ML і DL для аналізу та пошуку

зображень може бути застосоване в домені торгових марок, точніше, у зображеннях торгових марок.

Зображення торгових марок містять фігуративні елементи, такі як форми, кольори та текст, які разом створюють унікальний та відмітний знак, який ідентифікує певні товари та послуги. Відмінні елементи містять семантичне значення, яке описується кодами Віденської класифікації. Дослідження та застосування методів машинного та глибокого навчання для ідентифікації зображувальних елементів у торговельній марці за допомогою зображень торгової марки та кодів Vienna небагато. Основною причиною цього є складність і специфіка галузі, тому дослідницьке співтовариство вважає складним завдання знайти зображення торгових марок, які візуально та концептуально схожі. У нашому дипломному дослідженні ми досліджуємо, порівнюємо та оцінюємо продуктивність алгоритмів навчання під наглядом на реальному наборі даних зображень торгових марок для ідентифікації фігуративних елементів на основі кодів Віденської класифікації.

1.2. Тенденції розвитку машинного навчання в контексті обробки зображень

Останні проривні досягнення в машинному навчанні, моделях глибокого навчання та алгоритмах комп'ютерного зору дозволяють швидше та з вищою точністю виконувати розпізнавання об'єктів на зображеннях [5]. Сучасні підходи, що використовують контрольоване навчання, досягли вищої точності, ніж люди, з понад 97% у розпізнаванні та визначенні об'єктів на реальних зображеннях високої роздільної здатності [5, 6].

Переломний момент у цій галузі досліджень почався з AlexNet у 2012 році, де автори змогли вперше досягти історичних результатів за допомогою глибоких згорткових нейронних мереж на складному та великому наборі даних. У 2014 році були представлені вдосконалені моделі, такі як VGG Net і

GoogLeNet, які в основному були зосереджені на збільшенні глибини нейронних мереж.

Розглянемо загальну архітектуру CNN для розпізнавання та визначення об'єктів на реальних зображеннях високої роздільної здатності. Як зображено на рис. 1.2, мережа містить вісім шарів з вагами; перші п'ять є згортковими, а решта три повністю з'єднані. Вихід останнього повністю з'єданого шару подається на 1000-шляховий softmax, який генерує розподіл по 1000 мітках класів. Наша мережа максимізує цільову функцію багатоміальної логістичної регресії, що еквівалентно максимізації середнього по тренувальних випадках логарифмічної ймовірності правильної мітки під прогнозованим розподілом.

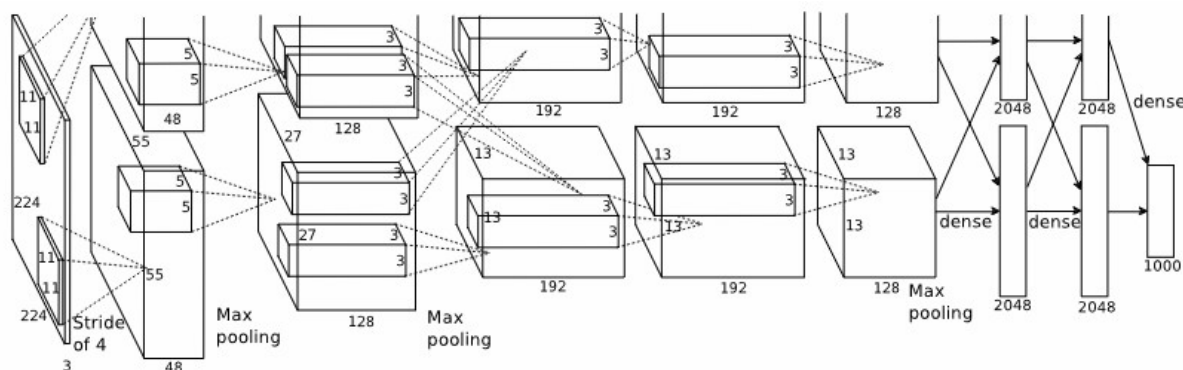


Рис. 1.2. Ілюстрація архітектури CNN, явно показуючи розподіл відповідальності між двома GPU

На рисунку 1.2 один GPU запускає частини шарів у верхній частині малюнка, а інший запускає частини шарів у нижній частині. GPU спілкуються лише на певних шарах. Вхід мережі має розмірність 150 528, а кількість нейронів у решті шарів мережі задається 253 440–186 624–64 896–64 896–43 264–4096–4096–1000.

Ядра другого, четвертого та п'ятого згорткових шарів з'єднані лише з тими картами ядер у попередньому шарі, які знаходяться на тому самому GPU (див. рис. 1.2). Ядра третього згорткового шару з'єднані з усіма картами

ядер у другому шарі. Нейрони у повністю з'єднаних шарах з'єднані з усіма нейронами у попередньому шарі. Шари нормалізації відповіді слідуєть за першим і другим згортковими шарами. Шари максимального пулінгу, слідуєть як за шарами нормалізації відповіді, так і за п'ятим згортковим шаром. Нелінійність ReLU застосовуєтья до виходу кожного згорткового та повністю з'єданого шару.

Перший згортковий шар фільтрує вхідне зображення $224 \times 224 \times 3$ з 96 ядрами розміром $11 \times 11 \times 3$ з кроком 4 пікселя (це відстань між центрами рецептивних полів сусідніх нейронів у карті ядер). Другий згортковий шар приймає на вхід (нормалізований за відповіддю та об'єднаний) вихід першого згорткового шару і фільтрує його 256 ядрами розміром $5 \times 5 \times 48$.

Третій, четвертий і п'ятий згорткові шари з'єднані один з одним без будь-яких проміжних шарів пулінгу або нормалізації. Третій згортковий шар має 384 ядра розміром $3 \times 3 \times 256$, підключені до (нормалізованих, об'єднаних) виходів другого згорткового шару. Четвертий згортковий шар має 384 ядра розміром $3 \times 3 \times 192$, а п'ятий згортковий шар має 256 ядер розміром $3 \times 3 \times 192$. Повністю з'єднані шари мають по 4096 нейронів кожен.

VGG Net збільшив кількість шарів глибокої згорткової нейронної мережі до 19, що покращило продуктивність і точність класифікації зображень на більших наборах даних. GoogLeNet [8] пішов іншим шляхом, запровадивши новий підхід до організації мережі під назвою «Початковий модуль» і пішов глибше, збільшивши кількість глибоких мережевих рівнів до 22. У 2015 році ResNet [9] представив глибокі залишкові мережі від 50 до 152 шари та каркас для полегшення навчання розпізнаванню зображень. Нарешті, у 2017 році ResNeXt [10] було запропоновано як просту та багатогалузеву архітектуру, яка будує мережі у вигляді будівельних блоків, що дозволяє розширення до нових вимірів.

Тенденція до збільшення кількості шарів і вдосконалення існуючих моделей є безперервною і досягла понад 1000 шарів [11]. Успіх моделей глибокого навчання змінив світ і зробив революцію в багатьох сферах, де

застосовуються комп'ютерне бачення, розпізнавання образів і обробка природної мови. Важливий внесок в успішний прогрес у цьому відношенні зіграв технологічний прогрес архітектур GPU, доступність відповідних наборів даних, інструментів і фреймворків. Глибоке навчання не можна описати як єдиний метод, а скоріше як набір алгоритмів, які можна застосовувати до різних областей або конкретних доменів для вирішення конкретних проблем. Набір визначених алгоритмів було класифіковано на контрольоване та неконтрольоване навчання [12]. Автори в [12] у групі згорткової нейронної мережі (CNN), рекуррентної нейронної мережі (RNN), довготривалої короткочасної пам'яті (LSTM) і рекуррентної одиниці з блоками (GRU) під наглядом, а в рамках неконтрольованого навчання об'єднують самоорганізуючі карти, автоматичні кодери та обмежені машини Больцмана.

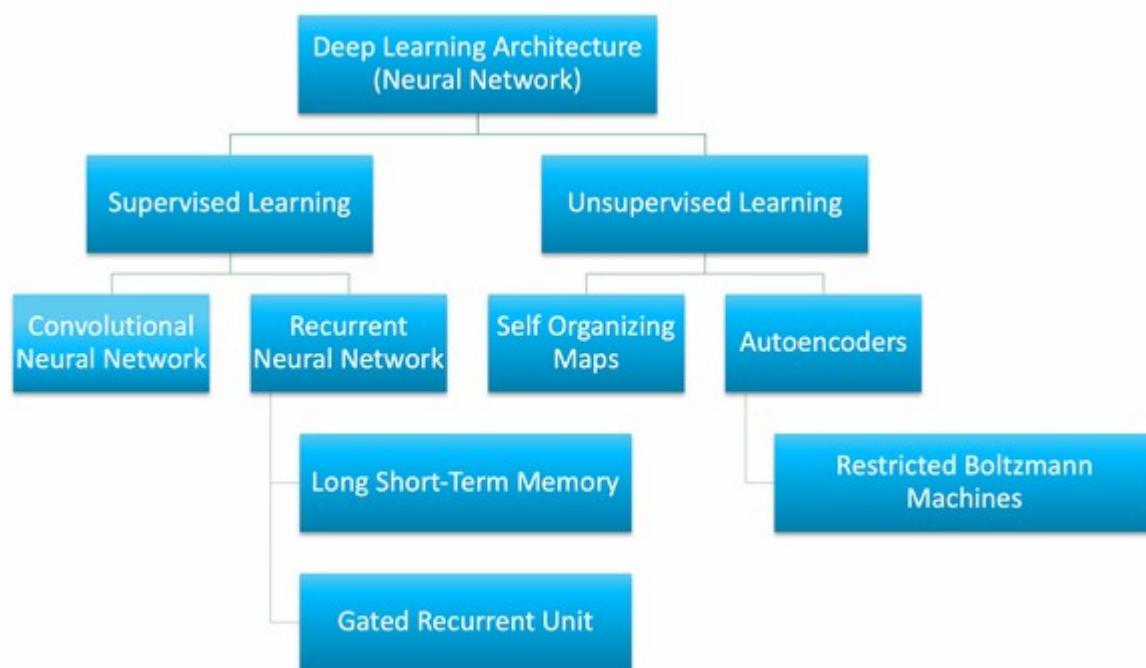


Рис. 1.3. Класифікація алгоритмів глибокого навчання

Згорткові нейронні мережі — це особливий тип нейронних мереж прямого зв'язку, які є композицією таких шарів, як згорткові шари, шари

об'єднання та повністю зв'язані шари [12]. Їх структура натхненна нейронами людського мозку, які утворюють багатошарові перцептрони, точніше їх структура складається з вхідного шару, кількох прихованих шарів і одного вихідного шару. Таким чином, згорткові нейронні мережі здатні розпізнавати значущі особливості в зображеннях і можуть ідентифікувати об'єкти. Ця здатність робить CNN придатною архітектурою в області комп'ютерного зору для завдання класифікації зображень, локалізації об'єктів і виявлення об'єктів у зображеннях реального світу.

Повторювані нейронні мережі — це тип нейронних мереж, які використовують послідовну інформацію в мережі, яка дозволяє попереднім виходам вбудовувати інформацію та використовуватися як вхідні дані, маючи приховані стани [12]. Додатки або RNN здебільшого використовуються для таких проблем, як обробка природної мови, розпізнавання мовлення, генерація музики, класифікація настроїв, розпізнавання імен і машинний переклад. Через мультиплікативні градієнти, які можуть збільшуватися або зменшуватися залежно від кількості шарів, RNN, як відомо, страждають від феномена зникаючих градієнтів. Модулі довгострокової короткочасної пам'яті (LSTM) і Gated Recurrent Units (GRU) обробляють явища зникаючих градієнтів, забезпечуючи блоки пам'яті в своїх рекурентних з'єднаннях [13].

Із звичайних методів машинного навчання для розпізнавання об'єктів і класифікації тексту історично застосовували машину опорних векторів (SVM), оскільки вони здатні вивчати концепції високого рівня з ознак низького рівня в зображеннях [14]. Деревя рішень (DT), з іншого боку, дуже прості у реалізації та здатні відображати функції низького рівня на концепції високого рівня за допомогою правил прийняття рішень [14].

Моделі машинного та глибокого навчання знайшли широке застосування в різних сферах і широке застосування дослідницькою спільнотою, яка постійно розширює межі. Вони також знайшли широке застосування в області торгових марок, з значними дослідженнями методів

пошуку схожих торгових марок за орфографією, фонетичними особливостями та схожістю на основі зображень [9 - 11].

1.3. Опис та постановка проблеми дослідження розпізнавання та ідентифікації елементів на зображеннях

Ідентифікація подібних зображень товарних знаків на основі візуальної подібності та семантичної подібності є дуже складним процесом і є предметом даної роботи. Застосування моделей глибокого навчання для ідентифікації особливостей зображень, створених людьми, наприклад логотипів торгових марок, які складаються з кольорів, тексту, графічного зображення, об'єктів, форм, текстур, суттєво відрізняються від природних (реальних) зображень. Зображення товарних знаків можуть представляти реальний об'єкт у стилізованій формі з використанням геометричних фігур або абстрактних понять. Але найголовніше те, що в поєднанні вони мають семантичне значення, що лише навчені люди-експерти можуть оцінити релевантність кожного елемента. Цей процес є суб'єктивним для кожної людини, оскільки за своєю природою люди сприймають речі по-різному через освітні та культурні особливості.

Ідентифікація, пріоритезація та ранжування кожного елемента окремо дуже складні. Оцінка зображення на семантичну або концептуальну схожість є основою юридичного процесу експертизи та протиставлення подібних або ідентичних торгових марок. Концептуальну подібність у цьому відношенні можна визначити як визначення схожих торгових марок, які є оманливими та заплутаними для кінцевого користувача. Існують різні рівні для визначення концептуальної подібності, такі як візуальний, фонетичний і семантичний, а також є юридичні визначення та рекомендації, які визначають, як цей предмет розглядається в області торгових марок [15].

Тому, намагаючись вирішити проблеми подібності зображення торгової марки та подібності, була винайдена Віденська класифікація.

Віденська класифікація — це міжнародна система класифікації, яку адмініструє Всесвітнє відомство інтелектуальної власності (ВОІВ), і її метою є класифікація зображувальних елементів товарних знаків на основі їх форми.

Для кожного зображення торгової марки застосовується ручне маркування ідентифікованих ознак торгової марки кодами з Віденської класифікації. Потім ці мітки служать для пошуку зображень, позначених тими самими кодами вже зареєстрованих торгових марок, які підлягають дослідженню зображення та концептуальної подібності. Процес класифікації зображень за допомогою кодів Відня є складним, трудомістким і виконується вручну навченими експертами, які дотримуються точних інструкцій з класифікації, і немає доказів наукових спроб автоматизувати цей процес.

Незважаючи на те, що ландшафт інструментів і технологій різко змінився за останнє десятиліття, розглядався лише аспект схожості зображень. З удосконаленням методів глибокого навчання комбінація різних методів глибокого навчання була використана для пошуку подібних зображень товарних знаків, таких як згорткова нейронна мережа (CNN) і сіамська нейронна мережа (SNN) [16], комбінація глибоких згорткових нейронних мереж (DCNN) [17], згорткова нейронна мережа (CNN) і алгоритм пошуку найближчого сусіда (NNS) [18].

Огляд літератури показує, що більше дослідницьких методів і архітектур [16 - 26] було запропоновано та застосовано для пошуку зображень на основі вмісту на основі таких характеристик, як форма, колір і текстура, порівняно з дослідженнями на ідентифікація ознак торгових марок, позначених кодами Віденської класифікації, яких небагато [21].

Прогалина в дослідженнях у цій галузі очевидна. Основною причиною цього є відсутність загальнодоступних баз даних для досліджень, які містять зображення, позначені кодами Vienna, як стверджують автори в [21]. Але на сьогоднішній день є кілька загальнодоступних великих наборів даних [27], [28], які можна використовувати для дослідницьких цілей. Ці набори даних

містять зображення торгових марок, позначені кодами Vienna, і дані вбудовані в бібліографічну інформацію про торгову марку. Формат вихідних даних файлів – XML. В [21] першими використали такий набір даних. Їхні дослідження показують, що хороші результати можна отримати із зображень торгових марок, які містять колір і форму, але гірші результати для інших функцій, незважаючи на використання невеликого набору даних для навчання. Використання міток із Віденської класифікації не дуже зрозуміло щодо того, як воно впливає на їх моделі або як воно покращує їхні результати.

Основною задачею цієї роботи є порівняння продуктивності алгоритмів глибокого та машинного навчання для завдання автоматизації процесу маркування, який передбачатиме коди Vienna для нових немаркованих зображень торгових марок. Оскільки в цьому відношенні не було виконано жодної попередньої роботи, яку ми могли б ідентифікувати з оглядом літератури, охопити як машинне навчання, так і глибоке навчання для багатокласової класифікації зображень торгових марок є складним і мотивуючим. У нашому дослідженні ми використовуємо реальний набір даних із зображеннями торговельних марок із кодами Vienna, які є загальнодоступними на порталі відкритих даних Офісу інтелектуальної власності Європейського Союзу (EUIPO) [27]. Цей набір даних містить понад 780 000 позначених зображень торгових марок із концептуальними кодами Vienna.

За допомогою огляду літератури [16 - 21], [22 - 26] ми виявили прогалину в дослідженнях щодо ідентифікації зображувальних елементів товарних знаків за допомогою існуючих класифікованих наборів даних із класами чи кодами Vienna. Наразі не реалізовано прогнозних моделей, які класифікують нові зображення торгових марок за допомогою передбачуваних кодів на основі історичних даних, класифікованих експертами. Для уточнення, використовуючи «клас», «код» або «ярлики», ми посилаємось на коди Віденської класифікації. На рисунку 1.4 нижче ми

надаємо високорівневу абстракцію нашої постановки проблеми. Наш набір даних зображень складається із зображень із різними характеристиками, такими як форма, колір, текстура, текст тощо, і кожне зображення пов'язане з одним або кількома кодами, що описують ці функції. Навчання моделей глибокого навчання з минулими історичними даними може автоматизувати процес прогнозування нових немаркованих зображень торгових марок за допомогою кодів Vienna. У нашому випадку через обмеження тези він передбачатиме лише категорію Vienna.

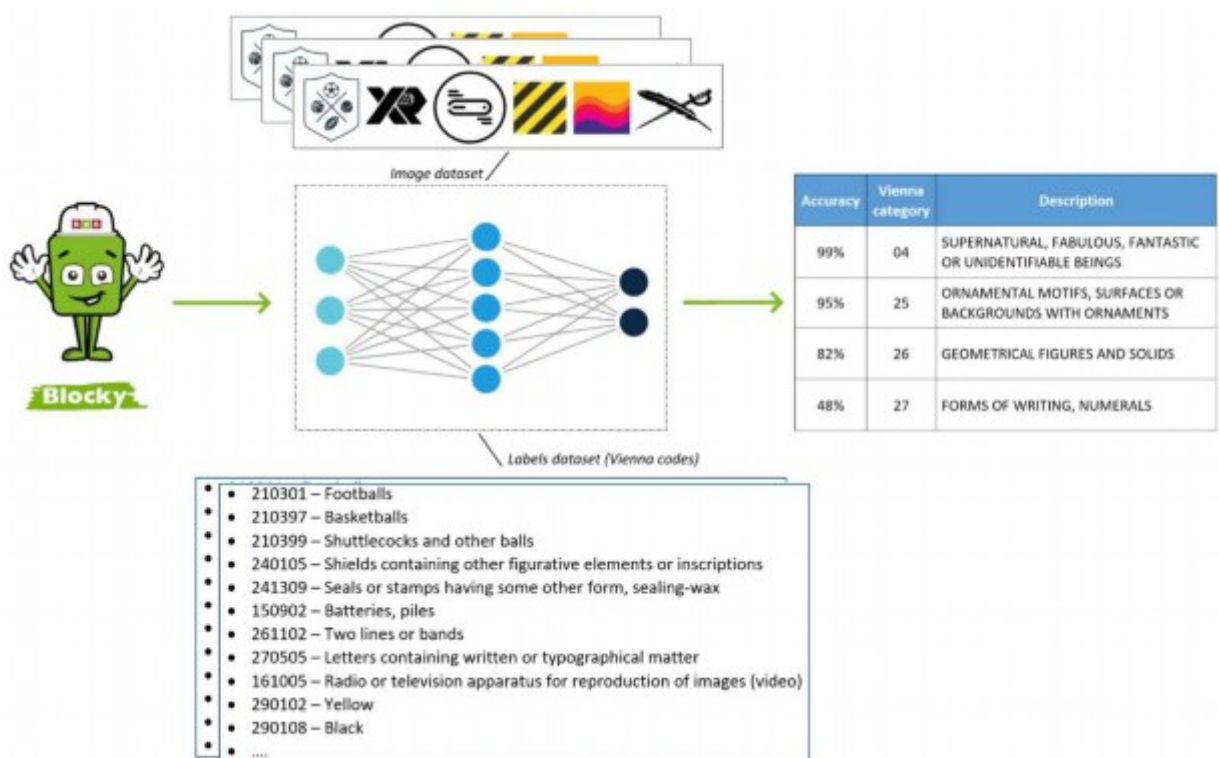


Рис. 1.4. Високорівнева абстракція постановки проблеми

Оскільки моделі глибокого навчання дають результати з вищою точністю порівняно з людьми у виявленні та локалізації об'єктів на зображеннях реального світу [29 - 31], ми хотіли б дослідити, як різні моделі глибокого навчання та звичайні моделі машинного навчання можуть допомогти зменшити розрив між візуальною інформацією та значенням інформації при ідентифікації образотворчих елементів на зображеннях. Крім того, ми хотіли б вивчити їх можливості та порівняти їх продуктивність і

точність із позначеним набором даних зображень торгових марок. Для проведення тестів, експериментів і аналізу ефективності ми завантажили набір даних EUTM [27]. Щоб зрозуміти, як вибрати найкращу підмножину зображень і концептуальних кодів Vienna, виконується аналіз необроблених даних для нормалізації набору даних навчання, тестування та перевірки. Щоб досягти цього, виконується попередня обробка та курування даних, і це один із найважливіших кроків процесу глибокого навчання, оскільки генерування високоякісної інформації на вхідних даних є ключовим для точного аналізу проблеми.

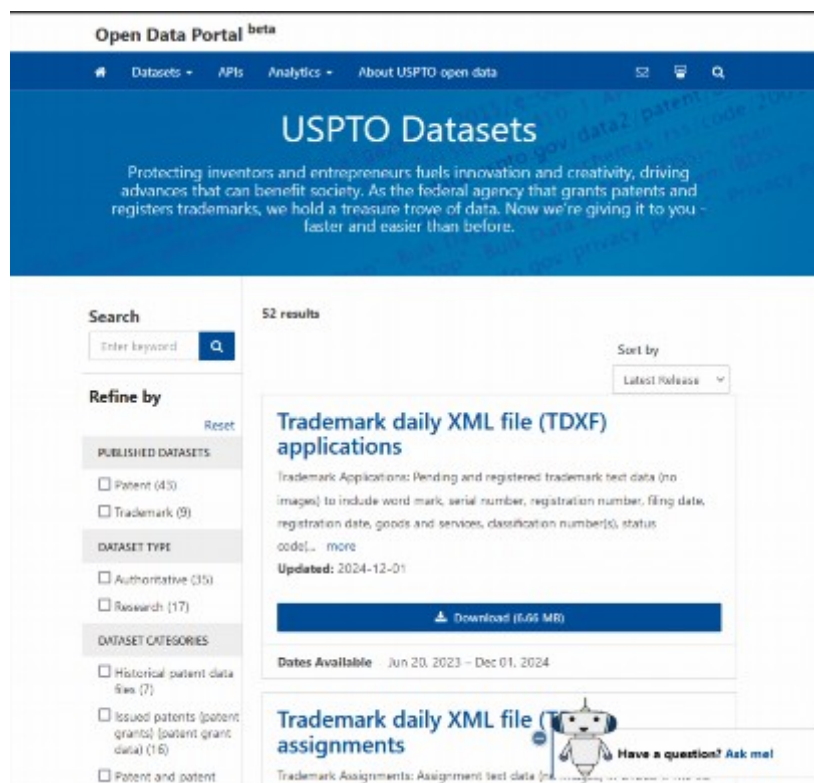


Рис. 1.5. Джерело отриманих наборів даних для проведення дослідження та експериментів

Пов'язані роботи [16 - 21], вказують на те, що концептуальна подібність є відкритою проблемою, яка не була правильно розглянута через семантичний розрив між візуальною інформацією та значенням цієї інформації. Семантична або концептуальна подібність визначається як ступінь відмінності спільних елементів між візуально подібними

зображеннями товарних знаків, які є оманливими та заплутаними для людського ока [32]. Семантична інформація кожного фігуративного елемента закладена в зображення товарних знаків, які позначені та описані концептуальними кодами Vienna. Наша мета полягає в тому, щоб поглиблено дослідити, які можливості мають моделі глибокого навчання, коли навчаються за допомогою зображень із мітками, щоб зменшити цей розрив, яка їхня продуктивність і які архітектурні проблеми вони мають. Моделі глибокого навчання, здатні ідентифікувати візерунки, форми та інші фігуративні елементи з семантичним виводом і з підвищеною точністю ідентифікації, також є кроком вперед у правильному напрямку для інших областей. Щоб отримати більш повний огляд, ми провели контрольовані експерименти, щоб порівняти ефективність різних моделей глибокого та машинного навчання, які стосуються досліджуваної галузі. Показники продуктивності, такі як точність, точність, запам'ятовування, оцінка F1 і час навчання, використовуються для оцінки та вимірювання продуктивності вибраних моделей. Результати та висновки аналізуються, візуалізуються та інтерпретуються з використанням тих самих показників для всіх моделей, які детально пояснюються в третьому розділі.

Для впровадження та перевірки проекту ми не консультувалися з експертами з класифікації, які щодня мають справу з ручним маркуванням зображень торговельних марок через обмеження тези та обсяг впровадження. Для обговорення проблеми та збору відгуків від експертів у галузі включення трьох ієрархічних кодів із Віденської класифікації є обов'язковим, а для цілей оцінки та тестування потрібна розробка системи з інтуїтивно зрозумілим і простим у використанні інтерфейсом користувача.

Дослідження зосереджено на вивченні та порівнянні ефективності існуючих моделей глибокого навчання, алгоритмів, а також звичайних моделей машинного навчання в побудові інтелектуального інструменту для пропонування кодів Vienna на основі розпізнавання фігуративних елементів у торговельній марці з використанням історичних даних.

Висновки до розділу

У першому розділі проведено комплексний аналіз предметної області застосування машинного навчання для маркування зображень. Розгляд основних аспектів штучного інтелекту та машинного навчання дозволив систематизувати знання про ключові принципи, що лежать в основі цих технологій. Було окреслено, що машинне навчання є підходом до створення систем, здатних навчатися на даних без необхідності явного програмування, що робить його ідеальним інструментом для задач обробки візуальної інформації.

Аналіз сучасних тенденцій розвитку в області машинного навчання підкреслив динамічний прогрес у створенні моделей для обробки зображень. Зокрема, акцент зроблено на таких методах, як згорткові нейронні мережі (Convolutional Neural Networks, CNNs), які забезпечують високу ефективність у розпізнаванні образів та класифікації зображень. Іншою важливою тенденцією є інтеграція глибокого навчання з іншими методами аналізу даних, що сприяє розширенню можливостей автоматизації обробки візуальної інформації.

Розглядаючи проблематику розпізнавання та ідентифікації елементів на зображеннях, було визначено основні виклики. Серед них — необхідність якісного маркування даних, що є критичною умовою для навчання моделей; складність обробки великих обсягів зображень у реальному часі; а також боротьба зі спотвореннями, шумами і неоднозначністю візуальних даних. Особливу увагу приділено формулюванню завдання дослідження, що включає оптимізацію процесу маркування та забезпечення високої точності ідентифікації об'єктів.

Таким чином, розділ заклав теоретичну основу для подальшого розроблення методів і моделей, спрямованих на вирішення актуальних задач маркування зображень із використанням машинного навчання.

РОЗДІЛ 2. МОДЕЛІ ТА МЕТОДИ МАШИННОГО НАВЧАННЯ ДЛЯ ІДЕНТИФІКАЦІЇ ЕЛЕМЕНТІВ НА ЗОБРАЖЕННЯХ

2.1. Опис особливостей елементів Віденської класифікації зображень

У цьому розділі ми представляємо Віденську класифікацію, яка є ключовим елементом для розуміння того, що таке Віденські концептуальні коди, як вони структуровані, яка їх роль і зв'язок і чому вони використовуються. Потім ми надаємо теоретичну підготовку щодо алгоритмів навчання під наглядом, архітектур, фреймворків та сучасних тенденцій у цій галузі. Він описує моделі машинного та глибокого навчання, що застосовуються в контексті комп'ютерного зору, зокрема для розпізнавання зображень, виявлення об'єктів і багатокласової класифікації зображень.

Віденська класифікація — це міжнародна система класифікації, яка використовується контрактними відомствами інтелектуальної власності для класифікації зображувальних елементів у торговельній марці. Вона заснована Віденською угодою 1973 року і адмініструється Всесвітньою організацією інтелектуальної власності (ВОІВ) [12].

Віденська класифікація — це ієрархічна система, яка проходить від загального до окремого, класифікуючи зображувальні елементи торговельних марок на основі їхньої форми, використовуючи три групові рівні, такі як категорії, підрозділи та розділи. Три рівні ієрархії містять двозначні коди та відомі як концептуальні коди Відня або класи Відня. EUIPO використовує модифіковану версію, адаптовану для своїх конкретних потреб шляхом створення нових кодів [33]. Набір даних про товарні знаки, який ми використовували в нашому дослідженні, класифікується за версією EUIPO [33] Віденської класифікації, яка складається з:

- 29 категорій;

- 147 підрозділів;
- 1503 розділів.

Категорії — позначаються в діапазоні від 1 до 29, і EUIPO, як договірне відомство інтелектуальної власності, прийняло та модифікувало версію Віденської класифікації та включило ще два класи, позначені кодами 98 і 99. Віденська категорія 98 охоплює інші типи торгових марок, такі як нюхові, звукові, мультимедійні тощо, а Віденська категорія 99 охоплює погані логотипи або погані образні зображення, колір або візерунок товарних знаків. Обидві категорії Vienna, 98 і 99, і відповідні файли зображень не включені в наш набір даних через низьку кількість зображень на категорію. Таблиця 2.1 нижче надає категорію та відповідний опис, як зазначено у версії Віденської класифікації, прийнятій EUIPO [33].

Таблиця 2.1.

Віденська класифікація за категоріями та опис

Category	Description
1	Celestial Bodies, Natural Phenomena, Geographical Maps
2	Human Beings
3	Animals
4	Supernatural, Fabulous, Fantastic or Unidentifiable Beings
5	Plants
6	Landscapes
7	Constructions, Structures for Advertisements, Gates or Barriers
8	Foodstuffs
9	Textiles, Clothing, Sewing Accessories, Headwear, Footwear
10	Tobacco, Smokers' Requisites, Matches, Travel Goods, Fans, Toilet Articles
11	Household Utensils
12	Furniture, Sanitary Installations
13	Lighting, Wireless Valves, Heating, Cooking or Refrigerating Equipment, Washing Machines, Drying Equipment
14	Ironmongery, Tools, Ladders
15	Machinery, Motors, Engines
16	Telecommunications, Sound Recording or Reproduction, Computers, Photography, Cinematography, Optics
17	Horological Instruments, Jewelry, Weights and Measures

18	Transport, Equipment for Animals
19	Containers and Packing, Representations of Miscellaneous Products
20	Writing, Drawing or Painting Materials, Office Requisites, Stationery and Booksellers' Goods
21	Games, Toys, Sporting Articles, Roundabouts
22	Musical Instruments and their Accessories, Music Accessories, Bells, Pictures, Sculptures
23	Arms, Ammunition, Armour
24	Heraldry, Coins, Emblems, Symbols
25	Ornamental motifs, Surfaces or backgrounds with ornaments
26	Geometrical figures and Solids
27	Forms of writing, Numerals
28	Inscriptions in various characters
29	Color
98	Other Marks
99	Too bad logo

Розділи — позначаються в діапазоні від 1 до 19 і в своїй структурі містять як категорію, так і підрозділ, позначений відповідними цифрами. У наведеному нижче прикладі ми надаємо підмножину з таблиці категорій і підрозділів із поясненнями, взятими з [33] для Віденської категорії 6 (Ландшафти). Вибрана категорія містить лише п'ять розділів, а саме:

- 6.1 — Гори, скелі, гроти
- 6.3 — Пейзажі з водою, річкою чи струмком
- 6.6 — Ландшафти пустелі або тропічного типу
- 6.7 — Міські пейзажі або пейзажі села
- 6.19 — Інші пейзажі

Розділи - позначаються діапазоном від 1 до 30 і у своїй структурі містять три ієрархічні рівні, такі як категорія, розділ і розділ. У наведеному нижче прикладі ми надаємо підмножину розділів для категорії Відень 6 (Пейзажі) та для категорії 6.1 (Гори, скелі, гроти). Вибрано п'ять підрозділів, які:

- 6.1.1 — Скелі, скелі, стіни скель
- 6.1.2 — Гори, гірські ландшафти

- 6.1.3 — Вулкани
- 6.1.4 — Гроти
- 6.1.7 — Стилізовані гори або вулкани

Основною метою Віденської класифікації є позначення зображень за допомогою категорій, підрозділів і розділів як єдиної мітки, відомої як Віденський код. Ці мітки використовуються для пошуку схожих торгових марок і створення звітів про пошук під час процесу експертизи торгових марок на предмет можливого порушення торгових марок з попередньо зареєстрованими правами.

Маркування нещодавно зареєстрованих торговельних марок — це завдання, яке виконується вручну спеціалізованими групами, навченими ідентифікувати локальні та глобальні ознаки відповідно до вказівок Віденської класифікації. Для кожної ідентифікованої ознаки застосовуються концептуальні коди з Віденської класифікації. Ці мітки зберігаються в базах даних, які потім служать ключовими словами для пошуку попередньо зареєстрованих торгових марок, які підлягають порівняльному дослідженню іміджу та концептуальної подібності. Пошук зображень не надає ранжованого списку результатів, але отримує список зображень, які відповідають критеріям пошуку.

Зображення торговельних марок позначаються як мінімум одним кодом до багатьох кодів. Інструкції з класифікації можуть відрізнятися, і кількість експертів з класифікації, призначених для виконання цього робочого завдання, залежить від кількості заявок на торговельні марки. Процес класифікації простий. Завдання класифікації виконується експертами з класифікації, а застосовані коди не перевіряються іншими експертами в галузі, оскільки коди Vienna не є обов'язковим елементом заявки на торговельну марку, а скоріше вони служать ключовими словами для пошуку. Наскільки нам відомо, на сьогодні не існує інструментів, які використовуються для автоматизації цього завдання. З нашого аналізу набору даних ми можемо зробити висновок, що сам процес класифікації є

складним процесом і дуже суб'єктивним, а з прикладів, ми можемо чітко побачити, що не всі можливі коди були застосовані. Кількість Віденських кодів, що застосовуються до торговельної марки, також залежить від найкращих практик кожного відомства інтелектуальної власності, яке впроваджує Віденську класифікацію. У багатьох випадках відомства інтелектуальної власності вважають за краще анотувати зображення меншою кількістю кодів Vienna та зосереджуватися на анотуванні лише основних характеристик, визначених експертом з класифікації, з єдиної причини, щоб зменшити шум результату під час пошуку та фільтрації за кодами Vienna.

На рисунку 2.1 нижче ми надаємо приклади зображень із нашого набору даних. Приклади охоплюють образні, кольорові та шаблонні зображення товарних знаків, позначені трьома ієрархічними рівнями Віденських кодів. Вибрані зображення ілюструють, що зображення набору даних є кольоровими та чорно-білими, які складаються з форм, тексту, текстури, візерунка та кольору або будь-якої комбінації того самого.



a) TM type: Figurative, EUTM 000000745²
Vienna codes: 02.09.10, 05.07.13,
05.07.23



b) TM type: Figurative, EUTM 011262656³
Vienna codes: 02.09.10, 05.07.13,
27.03.11



c) TM type: Figurative, EUTM 005271598⁴
Vienna codes: 05.01.03, 05.01.05,
05.01.15, 05.01.16



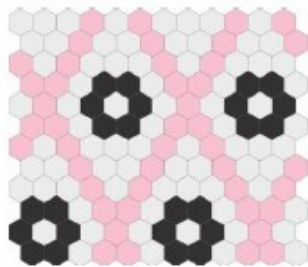
d) TM type: Figurative, EUTM 006588875⁵
Vienna codes: 02.01.16, 02.01.20



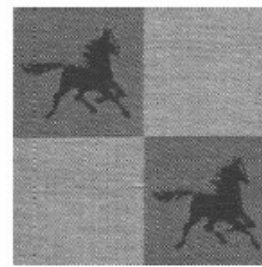
e) TM type: Color, EUTM 001526441⁶
Vienna codes: 25.07.21



f) TM type: Color, EUTM 006725956⁷
Vienna codes: 01.03.02, 01.15.23,
05.05.20, 05.05.21, 15.07.09



g) TM type: Pattern, EUTM 017910192⁸
Vienna codes: 25.07.05, 26.05.04



h) TM type: Pattern, EUTM 018180227⁹
Vienna codes: 03.03.01, 03.03.17,
03.03.24, 09.01.11

Рис. 2.1. Ілюстрація образних, кольорових і візерункових зображень торгових марок, пов'язаних із концептуальними кодами Віденської класифікації

2.2. Контрольоване машинне навчання

Контрольоване навчання можна визначити як статистичні моделі машинного навчання, які використовують позначені набори даних для навчання моделей, здатних передбачати вихідні змінні на основі шаблонів, визначених із вхідних даних [35]. Позначений навчальний набір даних містить як вхідні змінні, так і правильні вихідні змінні, які модель може вивчати з часом за допомогою повторних циклів навчання. Крім того, моделі навчання під наглядом можуть ідентифікувати приховані шаблони в невідомих і неспостережуваних структурах даних, але якість результату значною мірою залежить від якості вхідних даних. Точність моделі

вимірюється через мінімізацію функції втрат. Основна мета навчання під наглядом — побудувати функцію відображення, яка зможе навчитися передбачати мітки на основі вхідних точок даних і витягнутих функцій.

На основі типів проблем контрольоване навчання зазвичай поділяють на алгоритми класифікації та регресії. Алгоритми класифікації використовуються для вивчення шаблонів із вхідних даних і для визначення моделей для прогнозування результату шляхом генерації оцінки ймовірності [34]. Найвідомішими алгоритмами класифікації є опорні векторні машини, дерева рішень, випадковий ліс, наївний байєсівський алгоритм, k-найближчий сусід, згорточні нейронні мережі, довготривала короткочасна пам'ять, стробована рекурентна одиниця та багато інших. Алгоритми регресії використовують минулі дані для прогнозування майбутніх подій і числові дані для вивчення та розуміння залежних і незалежних змінних. Найбільш відомими алгоритмами регресії є логістична регресія та поліноміальна регресія.

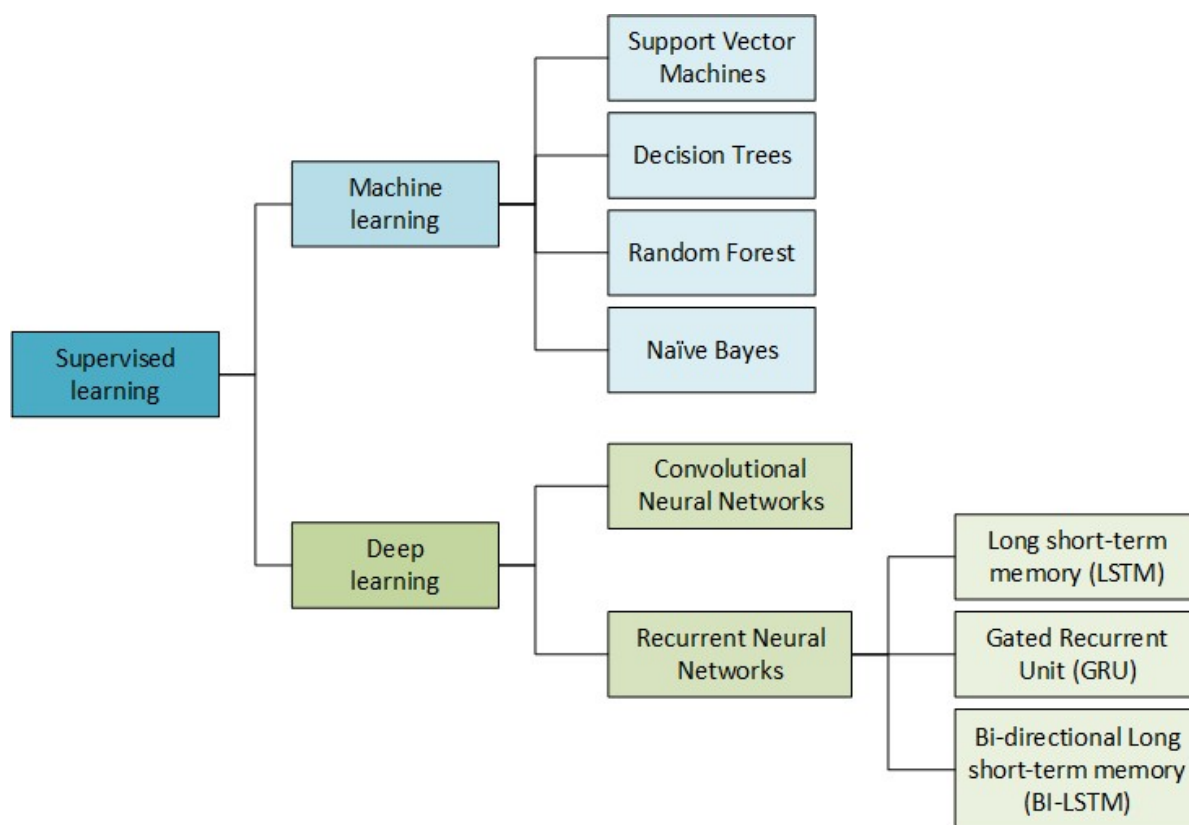


Рис. 2.2. Класифікація контрольованого навчання

На рисунку 2.2 показано класифікацію алгоритму керованого навчання на машинне та глибоке навчання, які далі класифікуються за типами архітектури, реалізовані в області комп'ютерного зору.

Машинне навчання (ML) — це піддомен штучного інтелекту, який змістив фокус з математичної науки на природничу, дозволивши комп'ютерам проводити спостереження за невизначеним світом шляхом проведення експериментів і замість використання логіки результати аналізувати за допомогою статистичного підходу [2].

Згідно з [35], машинне навчання визначається як комп'ютерна програма, налаштована на навчання на основі досвіду стосовно деяких завдань і певного показника продуктивності, якщо виконання цих завдань покращується з досвідом. Іншими словами, це наука про навчання комп'ютерного алгоритму збору зворотного зв'язку з попередньої продуктивності для покращення його продуктивності з часом щодо заздалегідь визначеної цілі. Дані або вхідні дані вводяться в алгоритми, які потім розпізнають зв'язки, шаблони, залежності та приховані структури за допомогою математичних і статистичних методів. У результаті створюється вихід у вигляді прогнозів і приписів.

У наступних підрозділах ми представляємо моделі машинного навчання, які ми досліджували та протестували для класифікації зображень торгових марок за допомогою кодів Vienna.

2.2.1. Алгоритм машини опорних векторів

Добре відомою моделлю машинного навчання, яка використовується як метод класифікації в області розпізнавання зображень, є Support Vector Machines (SVM). Перший алгоритм, запропонований для розпізнавання образів, був запропонований в [36]. Далі алгоритм був вдосконалений шляхом реалізації нелінійних векторів для відображення в просторі дуже високої розмірності [37]. Машини опорних векторів відображають дані у просторі ознак великої розмірності, щоб класифікувати точки даних, які не

можна розділити лінійно. Розділення даних здійснюється шляхом знаходження гіперплощини та перетворення даних у просторі шляхом максимізації поля між точками даних (рис. 2.3).

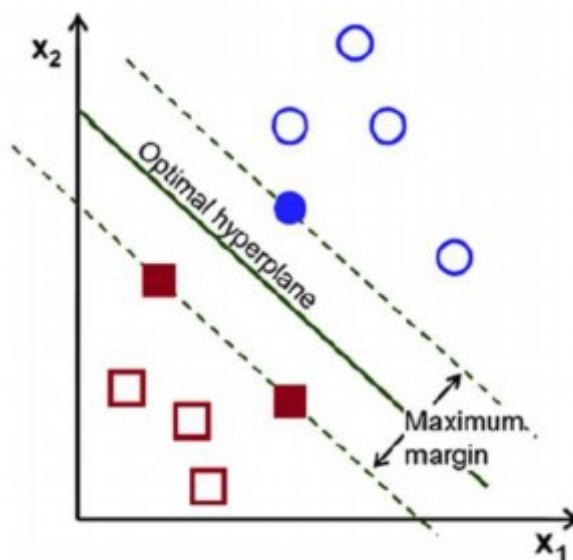


Рис. 2.3. Алгоритм машини опорних векторів

Точки даних з кожного боку лінії представляють відповідні класи. Часто буває важко намалювати гіперплощину, коли класи не є лінійно роздільними. Щоб вирішити цю проблему, функції ядра використовуються для перетворення простору ознак у вищий вимір, доки точки даних не стануть лінійно роздільними. У SVM є кілька функцій, які симулюють методи ядра, наприклад лінійну, поліноміальну, сигмоїдну та радіальну базисну функцію, яка вважається найпопулярнішою серед інших. Радіальну базисну функцію можна представити такою формулою:

$$K(x, x^1) = e^{(-\gamma \|x - x^1\|^2)}$$

Машина опорних векторів — це високоточна техніка класифікації, яка використовується для навчання наборів даних без переобладнання навчального набору. Застосування SVM в основному можна знайти в сферах

розпізнавання зображень і пошуку контенту [22, 23, 38], розпізнавання голосу та мовлення, інтелектуального аналізу даних, виявлення вторгнень та багатьох інших. Основним недоліком є погана масштабованість класифікаторів SVM, на яку впливає збільшення розміру набору даних або роздільної здатності зображення.

2.2.2. Древа рішень і випадковий ліс

Древа рішень – це алгоритми машинного навчання, які використовуються для прийняття рішень на основі параметрів, тобто значень, наданих у листових вузлах дерева. Структури дерев рішень мають кореневий вузол, розділені вузли, листові вузли та гілки. Розділені вузли відомі як внутрішні вузли та позначають перевірку функції чи атрибута. Листові вузли також відомі як кінцеві вузли і представляють мітку класу, тобто рішення, прийняте після обчислення всіх функцій, а гілки представляють результат тесту [41]. Древа рішень зазвичай використовуються для класифікації, тобто рішення підпадають під той самий клас, що й дані. Класифікація — це коли існує кінцевий набір атрибутів, які можна призначити вузлу. Дерево рішень представляє вивчені шаблони, які впорядковані в ряд рішень, які можна використовувати для прийняття майбутніх рішень. Як правило, попередні рішення є найважливішими характеристиками, оскільки вони представляють навчальну поведінку, яку можна використовувати для прийняття майбутніх рішень.

Під час навчання дерева рішень, яке використовує дерева рішень для прийняття рішень на основі параметрів, тобто значень, наданих у листових вузлах дерева, існує кілька проблем. По-перше, слід використовувати найменше можливе дерево, оскільки воно стає експоненціально більш ресурсомістким із появою рівнів гілок у дереві. З тієї ж причини може збільшитися час пошуку кількох дерев. По-друге, існує жадібний алгоритм навчання дерева рішень, який шукає лише певні частини дерева, отже, жадібна назва. У жадібному навчанні дерева вибирається атрибут, який

класифікує більшість прикладів. Якщо залишилися приклади, вибирається найкращий залишився атрибут для розділення даних. Процес продовжується рекурсивно, доки всі дані не будуть класифіковані або поки не залишиться атрибутів для розділення.

Оскільки важко чітко розрізнити кількість хороших і поганих прикладів, ентропія використовується для виявлення різниці. Ентропія відноситься до безладу невидимих випадкових елементів, які вказують на повільний розпад, що призводить до безладу. Як така, ентропія не є бажаною, і в теорії інформації вона використовується для вимірювання невизначеності випадкових величин [42]. Іншими словами, чим більше ентропія, тим більша невизначеність, і зменшення ентропії також відоме як отримання інформації. Ентропію можна описати такою формулою [42]:

$$H(V) = - \sum_k P(V_k) \log_2 (P(V_k))$$

Дерева рішень — це високоточна та ефективна техніка класифікації, яка використовується для завдань класифікації надійних зображень.

Випадковий ліс — ще один алгоритм класифікації машинного навчання, який точно агрегує вихідні дані кількох дерев рішень для обчислення та голосування за кінцевий результат. Алгоритм випадкового лісу є розвитком методу ансамблю, який використовує пакетування ознак або випадковість для створення лісу дерев рішень, які використовуються для класифікації класів з більшістю голосів [45].

Випадкові ліси знайшли широке застосування в задачах класифікації через те, що кількість гіперпараметрів, які потрібно налаштувати, мала. Незважаючи на те, що вони прості у використанні, вони дуже точні та ефективні з великими наборами даних, але мають менший час обчислення порівняно з деревами рішень.

2.2.3 Наївний Байєс

Naive Bayes — це керований алгоритм класифікації машинного навчання, заснований на теоремі Байєса, яка припускає, що ознаки є незалежними від даного класу [46]. Цей імовірнісний підхід припускає, що ідентифіковані ознаки є взаємно незалежними або непов'язаними одна з одною в даному контексті класу. Тому, оскільки це припущення не є дійсним для більшості випадків, ця теорія вважається наївною, хоча на практиці вона виявляється дуже точною. Іншими словами, теорема Байєса дозволяє обчислити подію на основі попередніх доказів або знання умови, яка пов'язана з цією подією, і може бути представлена такою формулою:

$$P(y|x) = \frac{P(x|y) P(y)}{P(x)}$$

Де $P(y|x)$ — це ймовірність y , коли відбувається подія x , або апостеріорна ймовірність класу, а $P(x|y)$ — це ймовірність x , коли відбувається подія y , або ймовірність ймовірності предиктора даного класу, і $P(x)$, $P(y)$ є ймовірністю появи x і y або попереднього класу та попередньої ймовірності предиктора.

Створення простого класифікатора Байєса за допомогою цього методу є простим і вимагає менше налаштування параметрів. Ці класифікатори добре працюють з невеликими наборами даних і здатні масштабувати великі. Наївні класифікатори Байєса знайшли широке застосування в реальних програмах, таких як класифікація тексту, фільтрація спаму, класифікація зображень та багатьох інших областях.

2.3. Дослідження алгоритмів глибокого навчання

Глибоке навчання — це субдомен машинного навчання, який використовує методи вилучення ознак із даних і пошуку значущих

представлень і шаблонів у наборі даних. Порівняно з традиційними методами, де розробник відповідає за вибір і кодування функцій, методи глибокого навчання можуть автоматично вивчати функції та подавати дані з кількома рівнями абстракцій.

Обчислювальні моделі глибокого навчання базуються на моделях штучних нейронних мереж, які технічним способом моделюють функції людського мозку. Основною одиницею нейронної мережі, яка виконує будь-які типи обчислень, є нейрон, також він відомий як сигмоподібний нейрон або перцептрон. Нейрони отримують вхідні дані від інших попередніх штучних нейронів і виконують обчислення перед тим, як передати вихідні дані наступному приймальному нейрону в мережі. Вихідна інформація базується на відносному значенні щодо інших даних і має пов'язану з нею вагу, а також застосовує функції активації до зваженої суми наданих вхідних даних.

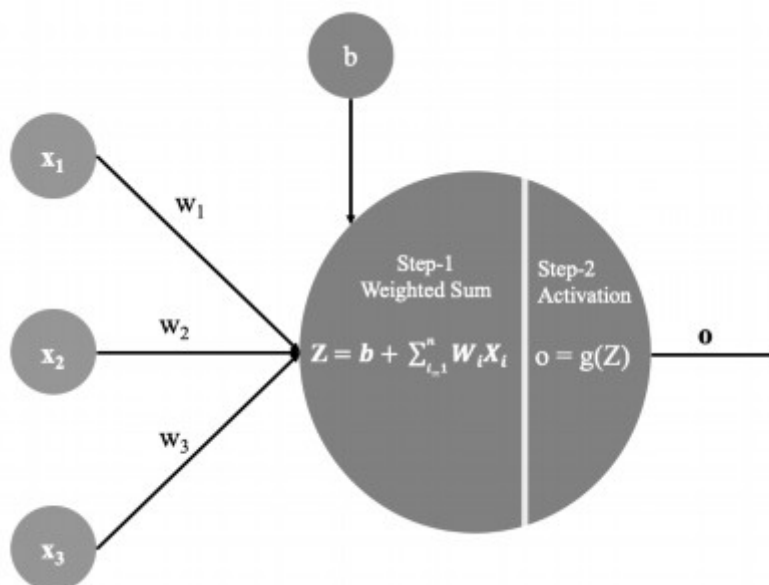


Рис. 2.4. Штучний нейрон

Рисунок 2.4 описує кожен частину нейрона, де всі вхідні дані позначаються від x_1 до x_3 , і кожен вхід нейрона має унікальний набір ваги, позначений від w_1 до w_3 , а зміщення позначається як b і є константою.

Відповідно до правила Розенблата для обчислення нейронів вихідна зважена сума обчислюється як:

$$z = b + \sum_i^n w_i x_i$$

Зважена сума передається до функції активації $g(Z)$, яка перетворює вихідні дані в нелінійний формат. Функції, що застосовуються в нейроні, називаються функціями активації, основна роль яких полягає в тому, щоб ввести нелінійність у вихід нейрона перед передачею виходів наступним нейронам.

Є кілька добре відомих функцій, що використовуються в нейронах, таких як ReLU, leaky ReLU, softmax, sigmoid, binary step, linear, tanh та інші. Випрямлена лінійна одиниця (ReLU) завдяки простоті та швидкому часу навчання стала стандартною функцією активації та широко використовується в нейронних мережах. Порівняно з функціями tanh, вона більш ефективна [5], оскільки не всі нейрони активуються одночасно, і її можна визначити як:

$$g(Z) = \max(0, Z)$$

Якщо обчислене значення позитивне, ReLU перешле його до наступного вузла рівня, а якщо обчислене значення негативне, ReLU проігнорує його. У випадку, якщо вхід негативний значення, ReLU запобіжить запуску нейрона, зробивши його 0, і якщо вхід має позитивне значення, ReLU просто перешле його до наступного вузла рівня. На рисунку 2.5 нижче зображено відповідний графік для функції активації ReLU.

Сигмоїдна функція активації дає змогу моделювати складні нелінійні межі рішень і прогнозувати ймовірності. Вона перетворює значення в діапазоні 0 і 1 і може бути визначена як:

$$g(Z) = \frac{1}{1 + e^{-z}}$$

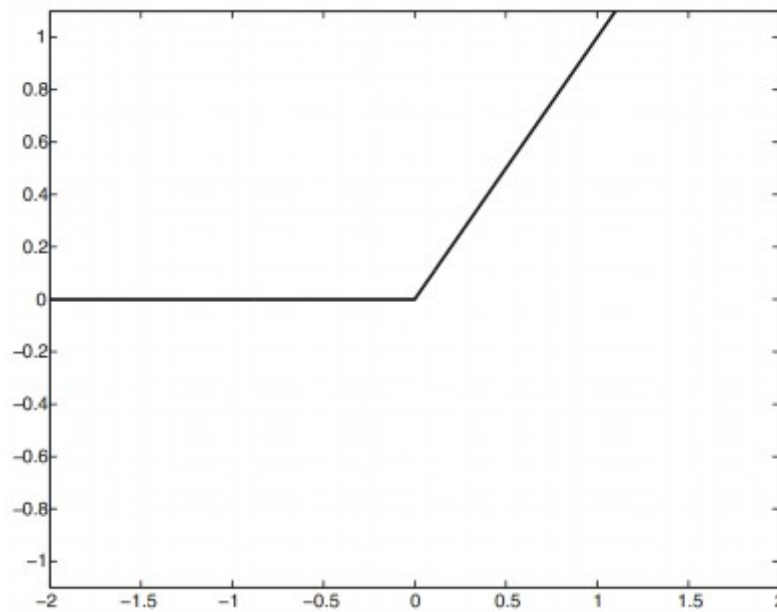


Рис. 2.5. Функція активації ReLU

Функцію активації сигмовидної кишки називають нелінійною функцією, оскільки результати, вироблені сигмовидними нейронами, не можна описати лінійними термінами. На рис. 2.6 нижче зображено відповідний графік для сигмоїдної функції активації.

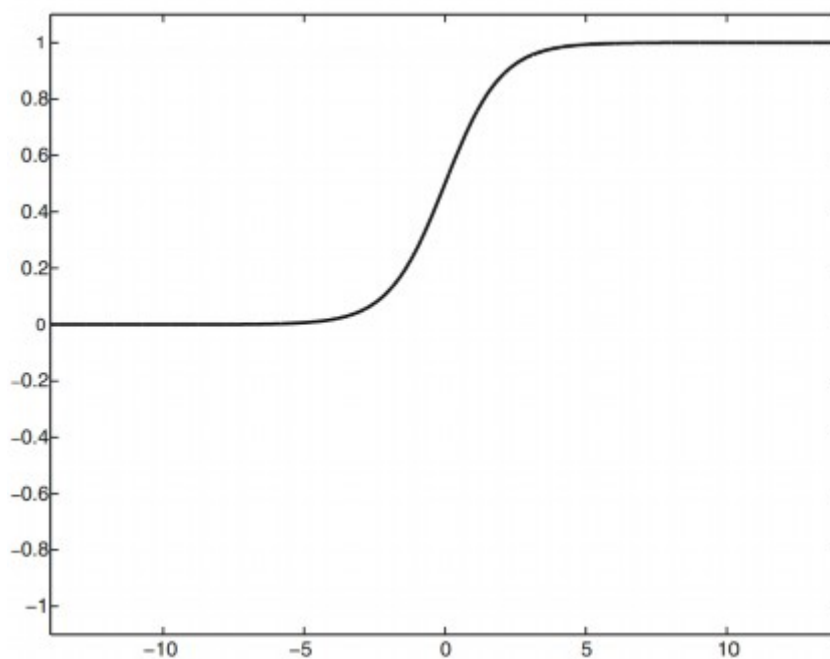


Рис. 2.6. Сигмоїдна функція активації

Глибоке навчання можна описати як застосування багатонейронних, багатошарових нейронних мереж для виконання завдань навчання, таких як класифікація, кластеризація, автоматичне кодування, регресія та інші завдання. Кілька паралельних нейронів запускаються з різними вагами, щоб одночасно навчатися на тих самих вхідних даних. Кожен нейрон навчається на всіх виходах попереднього рівня, і він відповідає за інтерпретацію певного шаблону в даних. На рисунку 2.7 ми зображуємо чотирирівневу нейронну мережу прямого зв'язку, яка має вхідний рівень, два прихованих шари та один вихідний рівень.

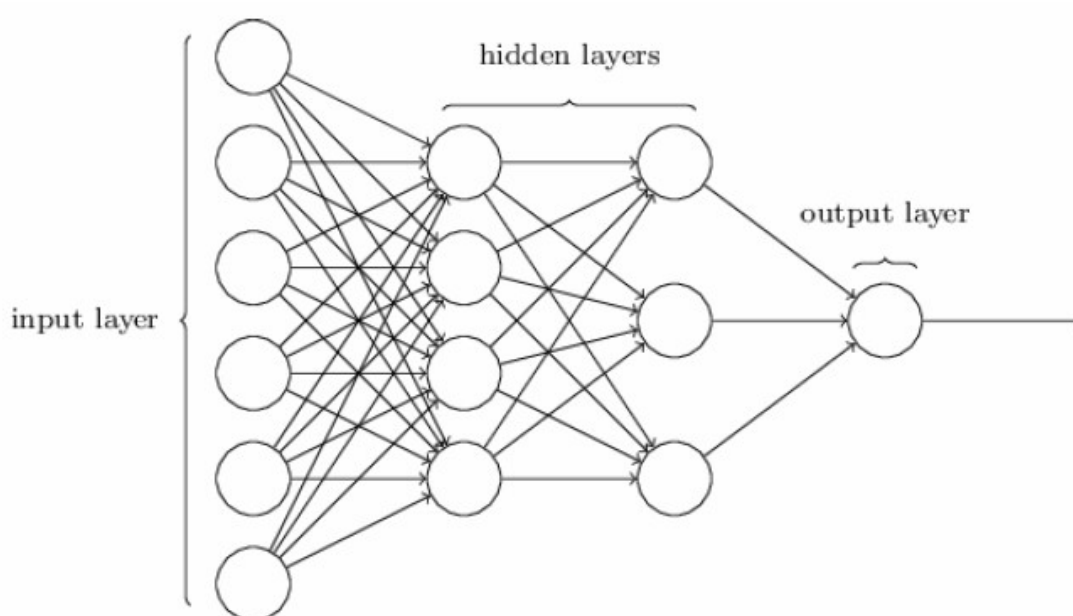


Рис. 2.7. Архітектура нейронної мережі

2.3.1 Згорткова нейронна мережа

Згорткові нейронні мережі (CNN) — це багатошарові нейронні мережі, які містять багато прихованих шарів нейронів, які здатні розпізнавати візерунки, форми та лінії. LeCun запропонував два згорткових шари та три повністю зв'язані шари та операцію об'єднання після кожного згорткового шару. З цією базовою архітектурою було створено перший CNN для розпізнавання рукописних символів (рис. 2.8).

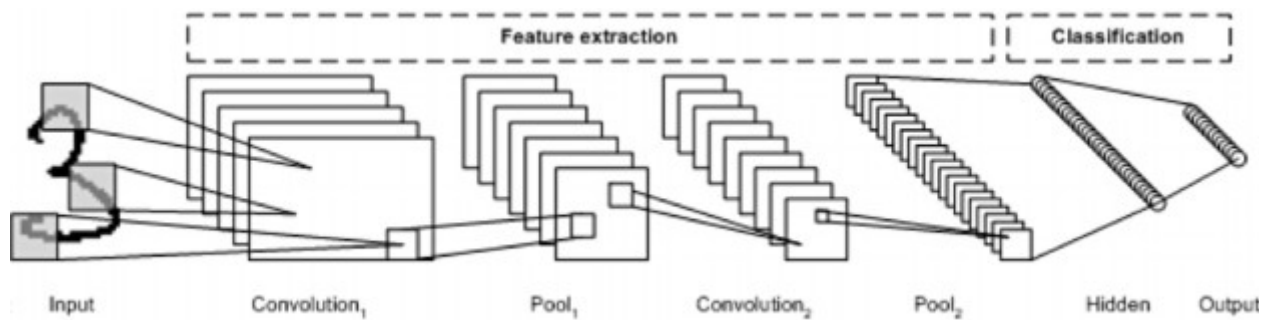


Рис. 2.8. Архітектура LeNet CNN

Архітектура згорткової нейронної мережі — це багаторівнева архітектура прямої нейронної мережі, яка складається з багатьох повторюваних згорткових шарів, за якими слідує шар об'єднання та повністю зв'язані рівні на кінцевій стадії мережі [12]. Згорткові шари є основним блоком мережі. Архітектура використовує фільтри, які виконують згорткові операції над даними вхідного зображення для ідентифікації шаблонів. Фільтри в згортковому шарі також відомі як ядра, основною функцією яких є ковзання по вхідному зображенню в пошуках шаблонів, які генерують результати, які називаються картою функцій або картою активації. Усі карти об'єктів об'єднуються разом і надсилаються на наступний шар для подальшої обробки.

Шари об'єднання застосовуються після згорткових шарів і зазвичай виконують операції зменшення вибірки, такі як максимальне та середнє об'єднання, шляхом зменшення кількості параметрів на вході. Операція максимального об'єднання використовується найчастіше, оскільки вона зберігає виявлені об'єкти та карти об'єктів. Він сканує вхідні дані та вибирає пікселі з максимальним значенням для вихідного масиву. На відміну від цього, об'єднання середніх зразків обчислює карту функцій шляхом обчислення середнього значення вибраних пікселів для вихідного масиву. Зменшення кількості параметрів призводить до зменшення складності мережі, розміру та часу навчання. Повністю пов'язані шари застосовуються до кінця згорткових нейронних мереж, щоб вирівняти вхідні дані, де кожен

вхідний сигнал пов'язаний з усіма нейронами, і оптимізувати такі цілі, як оцінки класу.

Згорткові нейронні мережі розроблені для роботи зі структурованими вхідними даними, представленими у вигляді кількох масивів або матриць, таких як двовимірне зображення. Цей тип даних також демонструє просторову залежність, оскільки суміжні просторові розташування на зображенні часто мають подібні значення кольорів окремих пікселів. Додатковий розмір фіксує різні кольори, що створює тривимірний вхідний об'єм. Таким чином, функції згорткової нейронної мережі залежать одна від одної на основі просторових відстаней. Інші форми послідовних даних, як-от текстові часові ряди та послідовності, також можна вважати окремими випадками сітчасто-структурованих даних із різними типами зв'язків між суміжними елементами. Переважна більшість застосувань згорткових нейронних мереж зосереджується на даних зображень, хоча можна також використовувати ці мережі для всіх типів часових і просторових даних [20].

Важливою властивістю даних зображень є те, що вони виявляють певний рівень інваріантності трансляції, чого немає у багатьох інших типах даних зі структурованою сіткою. Згорткові нейронні мережі мають тенденцію створювати подібні значення ознак із локальних регіонів із подібними шаблонами. Однією з переваг даних зображень є те, що вплив певних вхідних даних на представлення ознак часто можна описати інтуїтивно зрозумілим способом. Незважаючи на те, що CNN можуть легко вивчати та витягувати функції з навчальних даних, вони також можуть легко вписуватися в навчальні дані, що викликає проблему переобладнання. Переобладнання відбувається, коли модель точно відповідає навчальним даним, але не в змозі точно передбачити на невидимих даних і призводить до того, що модель досягає низького узагальнення. Для вирішення проблеми надмірного оснащення моделі можна використовувати різні методи, такі як збільшення даних, рання зупинка, навчання з більшою кількістю даних і методи регуляризації.

2.3.2. Рекурентна нейронна мережа

Рекурентні нейронні мережі (RNN) — це тип нейронних мереж, які використовують послідовну інформацію в мережі, яка дозволяє попереднім виходам вбудовувати інформацію та використовуватися як вхідні дані, маючи приховані стани [12]. У порівнянні з традиційною багатошаровою мережею, яка містить лише прямі з'єднання, рекурентні нейронні мережі можуть мати зв'язки між нейроном і попереднім шаром. Подібно до згорткових нейронних мереж, рекурентні нейронні мережі навчаються з навчальних даних, де на поточний вхід і вихід впливають приховані стани, відомі як комірки пам'яті, які містять інформацію з попереднього рівня. Вхідні та вихідні дані сильно залежать від попередньої вхідної інформації в послідовності. Повторювані нейронні мережі мають кілька переваг перед іншими типами мереж, наприклад здатність обробляти вхідні дані будь-якої довжини без збільшення розміру моделі, тоді як тренувальні ваги розподіляються в часі, а також враховується історична інформація.

Для навчання мереж RNN використовується алгоритм зворотного поширення через час (BPTT) для мінімізації функції втрат шляхом обчислення градієнта через усі параметри щодо ваг кожного окремого параметра, починаючи з останнього кроку за часом до початку. Однак RNN важче навчити, час обчислення повільний і їм важко отримати доступ до інформації давно минулого, оскільки вони стикаються з феноменом зникнення або вибуху градієнтів. Основною причиною цього є експоненціальне збільшення або зменшення мультиплікативних градієнтів щодо кількості шарів мережі. Модулі довготривалої короткочасної пам'яті (LSTM) і Gated Recurrent Units (GRU) справляються з феноменом зникнення градієнтів, забезпечуючи блоки пам'яті в своїх рекурентних з'єднаннях [10].

Застосування RNN здебільшого знаходять у розгортанні послідовного моделювання, такого як обробка природної мови, розпізнавання мовлення, створення музики, класифікація настроїв, розпізнавання імен, субтитри до

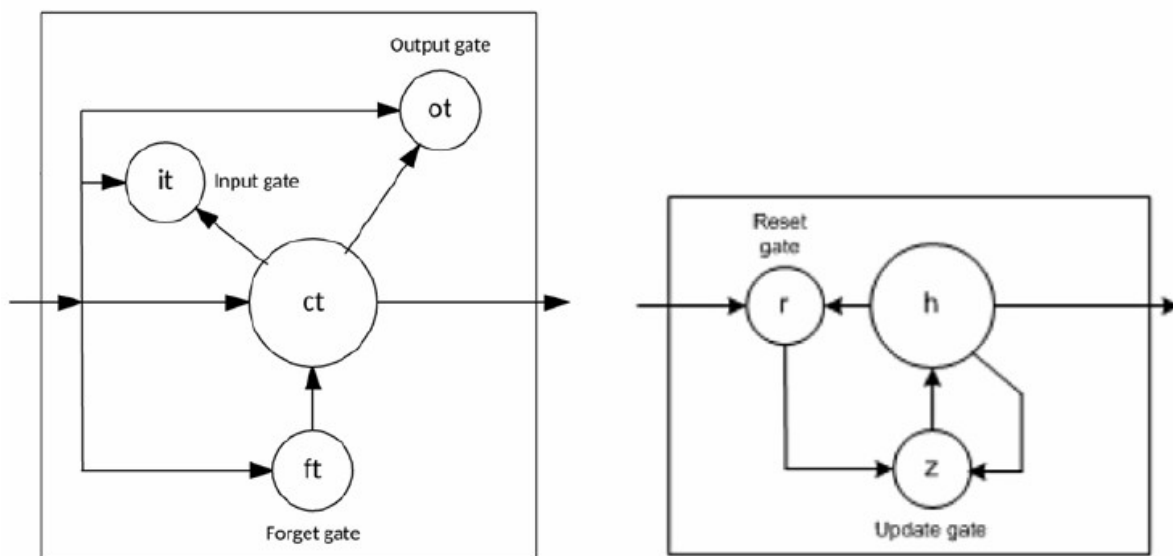
зображень, машинний переклад, прогнозування траєкторії в безпілотних автомобілях та багато іншого.

Ключовою особливістю мереж Long Short-Ter Memory є комірка пам'яті, яка дозволяє відстежувати інформацію протягом багатьох часових кроків. Для вирішення проблеми зникнення градієнта Hochreiter і Schmidhuber запропонували першу архітектуру LSTM. Основна ідея LSTM полягає в додаванні або видаленні інформації про внутрішній стан комірки через структури, які називаються воротами. Шлюзові структури містять стандартний рівень нейронної мережі, такий як сигмоїдна функція активації та поточкове множення.

Роль сигмоїдної функції активації полягає в тому, щоб змусити будь-який вхідний сигнал, що проходить через вентиль, бути між 0 і 1. LSTM використовують цей тип вентилів і функцій, тобто пропускати, зберігати, оновлювати та виводити, для обробки вхідної інформації та керування вихідною інформацією. Функція забутого шлюзу полягає в тому, щоб забути нерелевантну інформацію попереднього кроку. Функція збереження додає відповідну нову інформацію до стану комірки. Шлюз оновлення вибірково оновлює стан комірки, і, нарешті, вихідний шлюз повертає остаточний вихід, який служить вхідним для наступної позначки часу. Робота механізму мультиплікативного вентиля LSTM забезпечує безперервний потік обчислень градієнта з часом, щоб запобігти явищу зникнення градієнта в RNN.

Мережі Gated Recurrent Unit є ще однією спробою вирішити проблему зникаючого градієнта. Мережі GRU зберігають і спрощують конструкцію механізму воріт LSTM. Він містить лише два шлюзи, тобто шлюз оновлення та шлюз скидання. Вихідний затвор видалений з конструкції. Роль шлюзу оновлення полягає в тому, щоб визначити обсяг інформації, яку потрібно зберегти з попереднього кроку, а роль шлюзу скидання полягає в тому, щоб вирішити, як включити нову інформацію з попереднього кроку. Мережі GRU користуються перевагою простоти під час навчання, але мережі LSTM з

більшою кількістю даних можуть надавати кращі прогнози. На рисунку 2.9 нижче показано комірку пам'яті LSTM і GRU.



а) Комірка довгострокової пам'яті

б) Рекурентна елементарна комірка

Рис. 2.9. Ілюстрація комірки пам'яті різних архітектур RNN [12]

2.4. Архітектури глибокого навчання

Архітектури глибокого навчання можна визначити за кількістю нейронів, шарів і типів з'єднань між шарами. Багатошарові нейронні мережі прямого зв'язку мають найпростішу архітектуру і складаються з вхідного рівня, одного або кількох прихованих рівнів і мають один вихідний рівень. Кожен шар має кількість нейронів, які повністю пов'язані з нейронами наступного шару. Архітектурна складність архітектурних типів мала еволюційний прогрес, незважаючи на брак обчислювальних ресурсів, що уповільнило програми та їх широке впровадження. З технологічним прогресом GPU та їх обчислювальними можливостями для множення великомасштабних матриць і з розробкою фреймворків, які допомагають абстрагуватися від реалізації архітектур глибокого навчання [4],

проектування, тестування, розгортання та повторне використання моделей стало легким.

2.4.1. Мережа VGG Net

Моделі глибокого навчання продовжували покращувати точність і продуктивність. У 2014 році представили нову модель, яка в основному зосереджена на глибині нейронної мережі за допомогою дуже маленьких згорткових файлів 3x3 на всіх рівнях, відому як VGG Net [7].

Ця модель має глибину 16-19 вагових шарів, і вона перевершила попередню реалізацію в класифікаційному завданні ImageNet, вигравши виклик локалізації ILSVRC-2014 з 25,3% помилкою тесту. Глибока згорткова нейронна мережа складається з шістнадцяти шарів, де тринадцять шарів згорткові з розміром фільтра 3x3, а три шари пов'язані між собою. Крок згортки та відступ згортки встановлено на 1 піксель для згорткових шарів 3x3. Просторове об'єднання здійснюється 5 максимальними шарами об'єднання. Усі рівні в конфігурації повністю з'єднані та забезпечені нелінійністю ReLU. Набір даних був навчений шляхом поділу кожного набору зображень на кілька партій GPU та обробки за допомогою реалізації з кількома GPU. На рисунку 2.10 нижче зображено високорівневе представлення архітектури VGG-Net16.

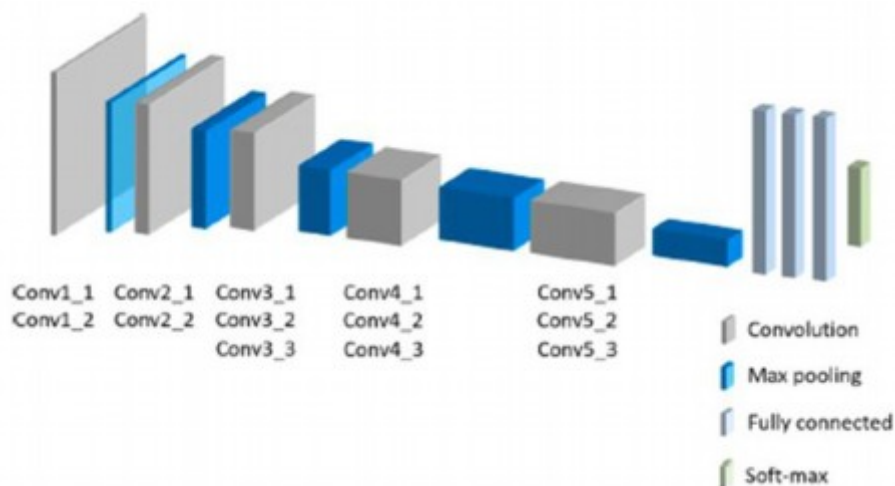


Рис. 2.10. Архітектура VGG-Net16

2.4.2. Мережа GoogLeNet

В роботі [8] представлено GoogLeNet, новий підхід до організації мережі під назвою «Початковий модуль» і пішов глибше, збільшивши глибину мережі до 22 рівнів. Їхня модель використовує дуже малі згортки, що позитивно впливає на зменшення кількості параметрів до 4 мільйонів порівняно з AlexNet [5] і в той же час покращує використання ресурсів. Він дозволяє виконувати об'єднання та згорткові операції паралельно, і він перевершує всі попередні моделі в задачі класифікації та виявлення на ImageNet ILSVRC-2014 з похибкою 6,7%.

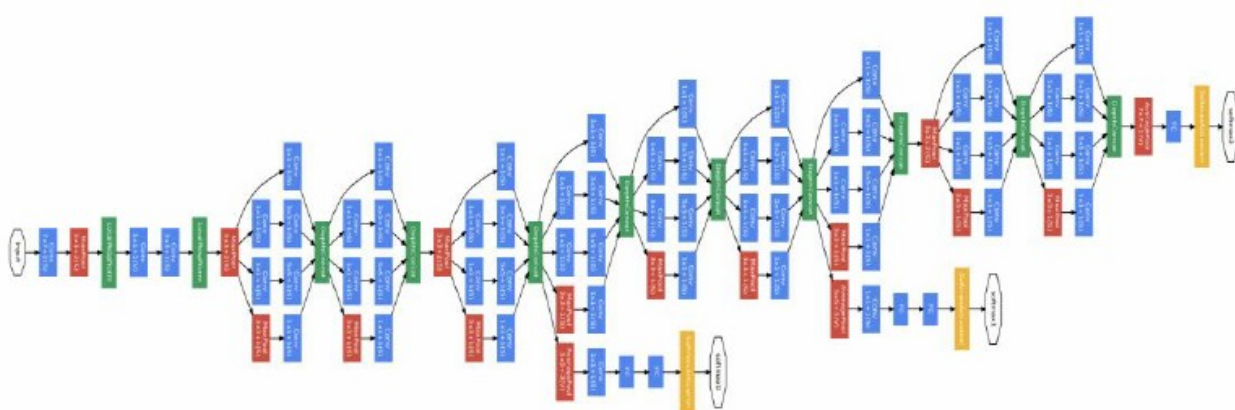


Рис. 2.11. Нейронна мережа GoogLeNet

2.4.3. Модель ResNet

Він та ін. [9] представив Residual Networks (ResNet), модель, яка охоплює 50-152 шари, але все ще має меншу складність порівняно з VGG Net [7]. На наборі даних ImageNet вони досягли 3,57% помилки, що призвело до перемоги в завданні класифікації зображень ILSVRC-2015. Продуктивність цієї моделі є чітким свідченням того, що вона може перевершити людей у класифікації зображень. З глибшими мережами була виявлена проблема деградації, яка була вирішена за допомогою введення глибинної основи остаточного навчання. Рисунок 2.12 нижче представляє основний блок залишкового навчання.

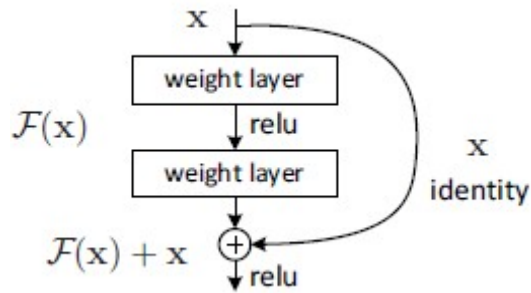


Рис. 2.12. Залишкове навчання: основний блок

В [10] запропоновано ResNeXt як прийнятну версію ResNet і VGG. У проекті запропонованої мережі використовується багатогілкова архітектура з невеликою кількістю гіперпараметрів, які відкривають новий вимір під назвою «кардинальність». Проведені експерименти [10] підтверджують, що точність вища з вищою потужністю порівняно з більш глибокими або широкими мережами. ResNeXt використовує кілька гілок у залишковому блоці. Кожна гілка виконує перетворення, які агрегуються за допомогою операції підсумовування. ResNeXt перевершує свою оригінальну версію (ResNet), використовуючи половину шарів.

2.4.4. Стан практичного застосування архітектур глибокого навчання

Розробка AlexNet [5], архітектури нейронної мережі, здатної досягти рекордної продуктивності на традиційно складному наборі даних, уперше продемонструвала переваги згорткових нейронних мереж і таких методів, як розширення даних і вилучення даних. Його реалізація використовувала паралельні обчислення з використанням середовища з декількома графічними процесорами та процесорами для завдання збільшення даних. AlexNet проклав шлях для нових і більш складних архітектур, таких як VGG Net [7], GoogLeNet [8], ResNet [9], ResNeXt [10], які можуть використовувати паралельні та розподілені обчислювальні ресурси для навчання та тестування. Реалізація попередньо навченої моделі на основі цих архітектур вимагає розширених апаратних ресурсів, таких як центральний процесор, графічний процесор, пам'ять і довший час навчання.

З удосконаленням фреймворків глибокого навчання процес проектування, побудови та перевірки попередньо навчених моделей на основі добре налагоджених існуючих архітектур стає простішим. Але через обмеження ресурсів наші моделі базуються на базових типах архітектури, які вимагають менше обчислювальних ресурсів і, отже, мають коротший час навчання.

2.5. Фреймворки глибокого навчання

Фреймворки глибокого навчання надають інструменти для проектування, побудови, тестування, перевірки та повторного використання нейронних мереж за допомогою різних високорівневих інтерфейсів прикладного програмування (API). Високорівневі API роблять реалізацію легшою, простішою та швидшою, оскільки складність базових математичних концепцій абстрагується. З удосконаленням апаратних архітектур центрального та графічного процесорів інфраструктури глибокого навчання можуть використовувати переваги своїх можливостей розпаралелювання для навчання, тестування та перевірки швидше.

TensorFlow — один із найпопулярніших фреймворків глибокого навчання. Він розроблений дослідниками Google, які працюють у Google Brain Team. Він був розроблений для виконання на гетерогенних системах без жодних змін у вихідному коді. Його було розроблено на C++ і Python, а підтримуваними мовами інтерфейсу є C, C++, Python, Java, Go, JavaScript. Його можна виконувати на різноманітних системах, починаючи від мобільних пристроїв і закінчуючи системами з кількома графічними процесорами та тензорними процесорами (GPU/TPU). Він підтримує як згорточні нейронні мережі, так і рекурентні нейронні мережі.

Cognitive Toolkit (CNTK) — це інструментарій із відкритим кодом для розподіленого глибокого навчання, розроблений Microsoft. CNTK написаний на C++ і забезпечує інтерфейс на C++, C#, Python, Java та BrainScript. Він

підтримує обчислення на центральному процесорі та кількох графічних процесорах і нейронних мережах, таких як згорткові нейронні мережі, рекурентні нейронні мережі та мережі довготривалої короткочасної пам'яті. Нейронні мережі представлені у вигляді серії обчислювальних кроків через спрямований граф.

Torch — це структура глибокого навчання, яка підтримує широкий спектр машинних алгоритмів і алгоритмів глибокого навчання. Він забезпечує кращу продуктивність порівняно з іншими фреймворками глибокого навчання завдяки швидкій мові сценаріїв LuaJIT та її реалізації C/CUDA. Він визначає пріоритетність обчислень на графічному процесорі та підтримує згорткові нейронні мережі та рекурентні нейронні мережі.

PyTorch — це платформа глибокого навчання, розроблена Facebook, яка зосереджена на простоті використання, функціональності та швидкості в середовищі виконання C++. Він підтримує розподілене навчання та асинхронне виконання операцій і однорангову комунікацію за допомогою Python і C++. Він забезпечує хорошу продуктивність завдяки підтримці кількох графічних процесорів і CUDA 10.0 і 11.1, а також підтримує згорткові нейронні мережі та рекурентні нейронні мережі.

Caffe — це структура глибокого навчання, розроблена Berkeley Vision and Learning Center. Він підтримує згорткові нейронні мережі та рекурентні нейронні мережі, але не підтримує мережі глибоких переконань. Базовий код написаний на C++ і забезпечує інтерфейс на Python і MATLAB для навчання та розгортання різних архітектур нейронних мереж.

DL4j — це популярна структура глибокого навчання серед спільноти Java. Він розроблений SkyMind. У поєднанні з Apache Hadoop і Spark DL4j може виконуватися в розподіленому середовищі для швидшого навчання. Він написаний на Java і має сумісність з мовами JVM, такими як Scala, Clojure і Kotlin. Усі бібліотеки є відкритими. Він забезпечує підтримку реалізацій із кількома GPU, а також підтримує CNN і RNN.

Для реалізації нашого проекту ми використали TensorFlow як платформу для створення, тестування та розгортання наших моделей ML і DL. Основною причиною цього вибору є інтуїтивно зрозумілий набір API, які прості у використанні, а також велика кількість документації та підтримки, які надає спільнота TensorFlow. Він також надає багатий набір інструментів для попередньої обробки даних, оцінки та візуалізації моделі, які спрощують процес навчання, що, з нашого досвіду та точки зору, було ключовим фактором для нашого рішення.

Висновки до розділу

Другий розділ присвячений детальному аналізу моделей і методів машинного навчання, що використовуються для ідентифікації елементів на зображеннях. Основну увагу зосереджено на вивченні існуючих алгоритмів, архітектур та інструментів, які дозволяють ефективно вирішувати цю задачу.

На початку розглянуто особливості елементів Віденської класифікації зображень, які є важливою основою для формалізації завдань маркування. Аналіз показав, що стандарти класифікації сприяють уніфікації підходів до обробки даних, що значно полегшує інтеграцію алгоритмів машинного навчання.

Далі було досліджено методи контрольованого машинного навчання. Розглянуто класичні алгоритми, такі як машина опорних векторів (SVM), дерева рішень і випадковий ліс, а також наївний Байєс. Ці методи демонструють високу ефективність для задач класифікації з відносно невеликими обсягами даних та є простими у впровадженні. Однак їх можливості часто обмежені складністю об'єктів і високорівневою взаємодією елементів на зображеннях. Також було досліджено сучасні фреймворки глибокого навчання, серед яких TensorFlow, PyTorch та Keras, які забезпечують потужні інструменти для швидкої розробки та оптимізації моделей. Вони не лише спрощують реалізацію складних архітектур, але й

підтримують інтеграцію з апаратним прискоренням, що робить їх надзвичайно ефективними.

У підсумку, проведений аналіз показав, що сучасні моделі і методи машинного навчання є надзвичайно гнучкими та ефективними інструментами для ідентифікації елементів на зображеннях. Цей розділ створює основу для практичного використання досліджених методів у реальних задачах маркування та класифікації зображень.

РОЗДІЛ 3. ЗАСТОСУВАННЯ МЕТОДОЛОГІЇ ТА МОДЕЛЕЙ МАШИННОГО НАВЧАННЯ ДЛЯ ІДЕНТИФІКАЦІЇ ЕЛЕМЕНТІВ ЗОБРАЖЕНЬ

3.1. Представлення методології

Великі набори даних є необхідною умовою для навчання моделей глибокого та машинного навчання. Доступність нормалізованих наборів даних є рідкістю, і такі набори даних не завжди є загальнодоступними для завантаження та тестування. Дані зазвичай надходять неструктурованими та в необробленому форматі, який як такий не можна використовувати для навчання та тестування. Щоб підготувати набір даних для експериментальних цілей, потрібна попередня обробка даних. Результатом такого підходу є створення нового набору даних для навчання та експериментування з різними моделями машинного та глибокого навчання. Теоретично цей крок простий і зрозумілий, але на практиці він досить складний. Якість результатів сильно залежить від якості набору даних. Тому виконання коригувальних ітерацій етапу попередньої обробки до досягнення бажаних результатів є обов'язковим.

Набір даних, який використовується для наших досліджень і експериментів, є відкритим набором даних торгових марок EUIPO. Він надає дані з 1996 по 2020 роки. Детальний аналіз завантаженого набору даних виконується для кращого розуміння складності проблеми. Під час процесу аналізу ми визначили, що набір даних як такий не можна використовувати та що він вимагає виконання дій з попередньої обробки даних, таких як очищення та курація, перш ніж підготувати остаточний набір даних.

Вихідні дані надходять у необробленому форматі XML. Віденські коди, які будуть використовуватися як етикетки, вбудовані як бібліографічна інформація торгової марки. Після визначення важливих елементів у XML ми вибрали підмножину елементів для створення нашого набору даних у

форматах JSON і CSV. Набір даних зображень доступний у форматах JPG і TIF. Цей крок дуже важливий, оскільки якість даних на вході впливає на результати на виході.

3.1.1. Контрольні імітаційні експерименти

Для вирішення постановки проблеми, ми використовуємо контрольовані експерименти як ефективний метод досягнення цієї мети. Алгоритми, що використовуються для навчання моделей, і кількісні дані, отримані в результаті контрольованого експерименту, можна вважати залежними та незалежними змінними, які сильно залежать від параметрів статичної моделі. І те, і інше можна виміряти за допомогою попередньо визначених показників ефективності, і це може допомогти нам відповісти на запитання дослідження. Контрольовані експерименти дозволяють нам зосередитися на зв'язку між залежними та незалежними змінними та розробити чітко визначену гіпотезу. За допомогою чітко визначених контрольованих експериментів ми можемо визначити, які невеликі зміни в незалежних змінних впливають на кінцевий результат. Щоб досягти бажаних результатів, повторення контрольованих експериментів з різними змінними дає з часом дорогий процес виконання, що є одним із головних недоліків цієї методології.

Вибір відповідних моделей глибокого та машинного навчання для проектування, побудови та виконання контрольованих експериментів для нашого тестового прикладу є дуже простим через мічений набір даних, який ми використовуємо. Для наших контрольованих експериментів ми вибрали Convolutional Neural Networks, Long Short-Term Memory і Gated Recurrent Unit для створення моделей глибокого навчання, а також Support Vector Machines, Decision Trees, Random Forests і Naive Bayes для створення моделей машинного навчання.

Проведення контрольованих експериментів на основі вибраних моделей є складним завданням через велику кількість змінних, які необхідно

відстежувати. Для збору дійсних і надійних даних, які допоможуть нам прийняти або відхилити нашу гіпотезу, ці випробування вимагають збалансованого набору даних на вході та точного виконання шляхом зміни параметрів і налаштувань в однаковому порядку для всіх моделей.

У домені торгових марок існує не так багато загальнодоступних наборів даних. Загальнодоступні набори даних зазвичай надходять у необробленому форматі, що містить лише файли зображень, а останні публікації містять зображення, пов'язані з бібліографічними даними. Останні потребують додаткової попередньої обробки даних для створення дійсного набору даних для тестування. У нашому випадку на етапі попередньої обробки ми визначили, що маємо справу з дуже незбалансованим набором даних, де присутність кількох концептуальних кодів Vienna є більшою порівняно з іншими. Для класів із меншою кількістю концептуальних кодів Vienna ми реалізували такі методи розширення даних, як масштабування, доповнення, обертання, обрізання та перевертання, але цього підходу було недостатньо, щоб збалансувати набір даних. Таким чином, створення збалансованого набору даних для порівняльного аналізу стало основною мотивацією для нового підходу.

Відібрані моделі були навчені за допомогою набору даних, що складається з 14 500 зображень, позначених категорією Vienna. Підмножини поділяються на 70% набору для навчання, 20% набору для тестування та 10% набору для перевірки. Повний цикл створення моделі, навчання, оцінки, оновлення гіперпараметрів повторювався для кожної зміни гіперпараметра.

3.1.2. Результати та методи оцінювання

Після кожної ітерації експерименту ми експортували результати та збирали вихідну інформацію для інтерпретації наших висновків. Спираючись на ці висновки, ми намагаємося відповісти на запитання дослідження та представити, які моделі глибокого навчання ефективніші для цілей класифікації зображень товарних знаків із зображеннями однієї мітки та

кількох класів із використанням міток або концептуальних кодів Відня з Віденської класифікації.

Результати експерименту оцінюються за допомогою визначених показників оцінки, таких як зважена точність, зважена точність, зважене пригадування та зважена оцінка F1. Загальний час навчання вимірюється та контролюється для кожного контрольованого циклу виконання експерименту, який використовується для побудови графіків та функцій візуалізації.

3.1.3. Надійність і валідність

Відтворення контрольованих експериментів з використанням того самого методу, налаштувань середовища та створення набору даних із використанням того самого підходу та точних умов, як описано в нашій дисертації, має дозволити будь-кому досягти тих самих результатів.

Метод описано чітко, але існує можлива проблема з надійністю, яка може легко виникнути під час збору даних, аналізу та етапу попередньої обробки, що може спричинити помилковий набір даних і, отже, різні результати. Витягнуті зображення торгових марок теоретично повинні мати унікальні ідентифікаційні номери або унікальні імена файлів зображень і повинні бути пов'язані з етикетками або кодами Віденської класифікації. З нашого аналізу ми дійшли висновку, що зображення не мають унікальних ідентифікаторів і не всі зображення пов'язані з кодами Vienna. Ця проблема впливає на створення набору даних, який поширюється, створюючи неправильні результати.

Контрольовані експерименти виконуються з використанням тих самих змінних і того самого випадкового початкового значення під час створення набору для навчання, тестування та перевірки. Для створення різних наборів ми використали бібліотеку scikit-learn і випадкове початкове значення 42 для всіх моделей. Це дозволяє нам мати той самий набір для навчання, тестування та перевірки, поки ми оновлюємо значення інших гіперпараметрів

у наших моделях. Цей підхід гарантує, що отримані бали є результатом зміни моделі, а не результатом різного випадкового навчання, тестування чи підтвердження для кожної ітерації навчання. Завдяки використанню випадкового початкового числа ми забезпечуємо надійність контрольованих експериментів, надійну оцінку й оцінку ефективності різних моделей. Усі моделі глибокого та машинного навчання, розроблені в наших контрольованих експериментах, використовують однакові експериментальні параметри та гіперпараметри.

Стосовно оцінки валідності та того, як вона пов'язана з нашим дипломним проектом, у наступному розділі ми обговоримо різні виміри, такі як валідність конструкту, внутрішня валідність і зовнішня валідність.

У нашому дослідженні було виявлено загрози внутрішній валідності, які пов'язані з попередньою обробкою даних. Під час етапу попередньої обробки даних під час вилучення ідентифікатора файлу зображення та пов'язаних кодів Vienna з XML-файлів ми виявили, що ідентифікатор зображення не є унікальним і, отже, впливає на якість набору даних. Це може призвести до того, що неправильні ідентифікатори файлів зображень будуть пов'язані з неправильними кодами Vienna.

Зовнішня валідність дослідження полягає в першу чергу в тому, чи можна застосувати методологію та контрольовані експерименти дисертації до інших проблем у предметній області та чи може включення експертів у даній галузі виправдати загальність результатів. Оскільки немає загальнодоступних наборів даних для тестування наших моделей, ми не можемо стверджувати, наскільки добре моделі можуть узагальнювати набори даних торговельних марок. Однак моделі, які використовуються в наших контрольованих експериментах, є стандартними моделями, які використовуються для класифікації зображень за кількома класами, продуктивність яких може бути добре узагальнена для подібних випадків використання, як ми досягли з нашими результатами.

3.2. Попередня обробка даних

Набір даних, який використовується для навчання моделей і для оцінки продуктивності вибраних алгоритмів, базується на товарних знаках Європейського Союзу (EUTM), які є загальнодоступними для завантаження з веб-сайту Офісу інтелектуальної власності Європейського Союзу. Набір даних містить файли зображень (JPG і TIF) і файли XML у необробленому форматі. Набір даних потребує попередньої обробки, очищення та перетворення даних, щоб використовувати його для навчання та тестування. Загальна кількість завантажених торгових марок становить понад 1,9 мільйона, що охоплює роки з 1996 по 2020 рік. У таблиці 3.1 нижче наведено типи торгових марок, включених до відкритого набору даних EUTM.

Таблиця 3.1.

Огляд відкритого набору даних EUTM

Trademark type	Count
3D	9,991
Color	1,005
Figurative	800,089
Pattern	75
Position	202
Sound	377
Hologram	13
Word mark	1,103,089
Total	1,914,841

Типи торговельних марок, які містять зображення та пов'язані коди Vienna у файлах XML, можуть бути включені в наш набір даних. З цієї причини для нашого дослідження ми вирішили включити лише фігуративні, кольорові та шаблонні типи товарних знаків для створення нашого унікального набору даних. Такі типи торгових марок, як положення, звук, голограма та слово, були виключені та не є частиною дослідження. Загальна

кількість записів у нашому наборі даних після початкового етапу відбору становить понад 800 000 позначених файлів зображень.

Для отримання необхідних елементів даних із XML-файлів, таких як номер програми, назва файлу зображення, тип файлу та коди Vienna, розроблено програму Java, яка створює вихідні файли у форматі JSON або CSV (рис. 3.1). Мова програмування Java використовується завдяки існуючим бібліотекам, які мають розширені функції та які легко використовувати для обробки та перевірки файлів XML. Після детального аналізу та багатьох тестів ми зрозуміли, що набір даних містить неунікальні назви файлів зображень, які під час створення унікального набору даних зображень існуючі зображення були замінені новими зображеннями з такою самою назвою файлу, і тому зображення асоціюються з неправильними концептуальними кодами Vienna.

```
<Transaction xmlns:schemaLocation="http://euipo.europa.eu/trademark/data http://euipo.europa.eu/schemas/trademark/EM-IM-TradeMark-V3-4.xsd"
  xmlns="http://euipo.europa.eu/trademark/data">
  <TransactionHeader>
    <SenderDetails>
      <RequestProducerDateTime>2020-07-24T12:15:10</RequestProducerDateTime>
    </SenderDetails>
  </TransactionHeader>
  <TradeMarkTransactionBody>
    <TransactionContentDetails>
      <TransactionIdentifier>018221003</TransactionIdentifier>
      <TransactionCode>EM-Trade Mark</TransactionCode>
      <TransactionData>
        <TradeMarkDetails>
          <TradeMark operationCode="Insert">
            <RegistrationOfficeCode>EM</RegistrationOfficeCode>
            <ApplicationNumber>018221003</ApplicationNumber>
            <ApplicationDate>2020-04-03</ApplicationDate>
            <RegistrationDate>2020-07-22</RegistrationDate>
            <ApplicationLanguageCode>de</ApplicationLanguageCode>
            <SecondLanguageCode>en</SecondLanguageCode>
            <ExpiryDate>2030-04-03</ExpiryDate>
            <MarkCurrentStatusCode status="1">1</MarkCurrentStatusCode>
            <MarkCurrentStatusDate>2020-07-24</MarkCurrentStatusDate>
            <KindMark>Individual</KindMark>
            <MarkFeature>Figurative</MarkFeature>
            <TradeMarkDistinctivenessIndicator>false</TradeMarkDistinctivenessIndicator>
            <WordMarkSpecification>
              <MarkVerbalElementText>qbees</MarkVerbalElementText>
            </WordMarkSpecification>
            <MarkImageDetails>
              <MarkImage>
                <MarkImageIdentifier identifierKindCode="EM">000162105466</MarkImageIdentifier>
                <MarkImageFileFormat>JPG</MarkImageFileFormat>
                <MarkImageURI>file://018/221/018221003/000162105466.JPG</MarkImageURI>
                <MarkImageCategory>
                  <CategoryCodeDetails>
                    <CategoryCode>27.05.22</CategoryCode>
                    <CategoryCode>27.99.02</CategoryCode>
                    <CategoryCode>27.99.17</CategoryCode>
                  </CategoryCodeDetails>
                </MarkImageCategory>
              </MarkImage>
            </MarkImageDetails>
          </TradeMark>
        </TradeMarkDetails>
      </TransactionData>
    </TransactionContentDetails>
  </TradeMarkTransactionBody>
</Transaction>
```

```
{
  "UID": "018221003-000162105466.JPG",
  "appNumber": "018221003",
  "fileName": "000162105466",
  "fileExtension": ".JPG",
  "viennaCodes": ["270522", "279902", "279917"]
}
```

Рис. 3.1. Вилучення даних з XML у JSON

Для точного вилучення створюється унікальний ідентифікатор зображення (UID) шляхом об'єднання номера програми з назвою файлу зображення, щоб відповідати частині рядка абсолютного шляху до файлу. Тоді той самий UID використовується як угода про іменування файлів зображень.

XML-файли, опубліковані EUIPO, використовують спеціальну реалізацію схеми на основі стандарту BOIB ST.66 для обробки, публікації та обміну всіма типами торгових марок [79]. Схема XSD, що перевіряє XML-файли, є загальнодоступною на веб-сайті відкритих даних EUIPO [27]. Схему було завантажено та включено до програми Java. Перед початком обробки кожен файл XML перевіряється на відповідність схемі. Перевірка даних важлива, щоб переконатися, що всі елементи та їхній вміст дійсні.

Під час попередньої обробки набору даних ми виявили, що XML-файли містять елементи й атрибути, видалені зі схеми внаслідок змін у регулюванні торговельних марок, тому потрібна була зміна схеми. Застосування цих змін було важливим, оскільки наша мета полягала в тому, щоб не втратити жодної прийнятної торгової марки на етапі попередньої обробки. Оскільки коди Відня є обов'язковими елементами в схемі, це означає, що в наборі даних є зображення, які не класифікуються за жодним із кодів Віденської класифікації. Таким чином, будь-які фігурні, кольорові чи візерунки товарні знаки, які не містили кодів Vienna, були видалені з остаточного набору даних. Крім того, будь-яка торгова марка, класифікована з кодом категорії Vienna 98 (інші марки) та 99 (занадто поганий логотип), відкидається та видаляється з набору даних, оскільки відсутність кількості репрезентативних зображень впливала на баланс набору даних.

Під час першої ітерації попередньої обробки даних загальна кількість зображень торгових марок становила 775 154, а загальна кількість унікальних кодів Vienna становила 1 513. Унікальні коди Vienna представляють список кодів, якими можна позначити зображення. Після виконання перехресної перевірки даних між набором даних і фактичними зображеннями ми

зрозуміли, що багато зображень відсутні у вихідних zip-файлах, і тому у нас були прогалини в перехресній перевірці даних. Ми оновили програму Java, яка виконує вилучення даних, і повторили попередню обробку даних із новими правилами вилучення, і загальну кількість доступних зображень торгових марок було зменшено до 716 496.

Щоб мати більш детальне уявлення про наш набір даних, ми згенерували статистичні дані на етапі попередньої обробки даних. Вилучення та візуалізація всіх кодів Vienna для кожної торгової марки, згрупованих за роками, виконується для кращого розуміння розподілу даних у нашому наборі даних перед ініціалізацією процесу навчання. На рисунку 3.2 нижче показано точний розподіл кодів Vienna (включаючи три ієрархічні рівні) для кожної категорії, починаючи з класу 1 до 29 для всього набору даних.

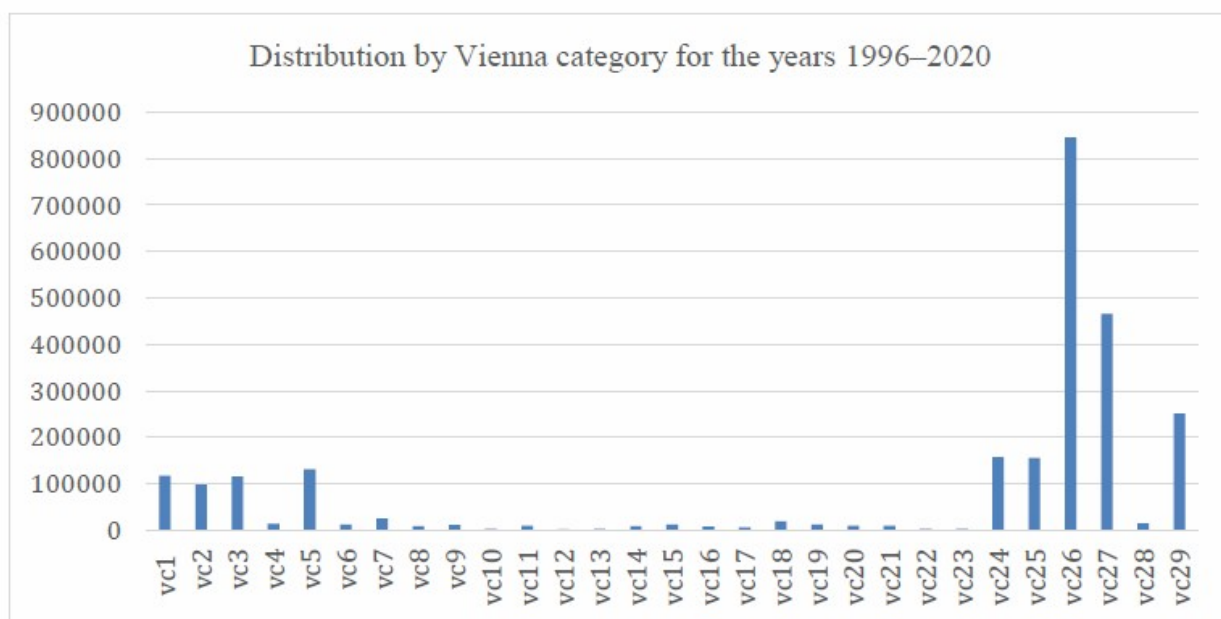


Рис. 3.2. Розподіл даних за категоріями

Щоб зменшити розрив між класами, ми досліджували методи збільшення даних, щоб збільшити кількість зображень у класах із низьким рівнем представлення, але все одно цього було недостатньо, щоб зменшити розрив, і, звичайно, не був правильним підходом до збалансування набору даних, оскільки велика кількість зображення не були доповнені. Розподіл

кількості зображень за категорією Vienna дуже незбалансований, і тому алгоритми глибокого навчання не можуть вивчати шаблони з даних без впливу більших класів та ігнорування шаблонів менших класів [80].

Початковий підхід до вирішення постановки проблеми полягав у проведенні контрольованого експерименту на основі підходу класифікації зображень із кількома мітками з наявним незбалансованим набором даних. Ми розробили моделі та виконали повний цикл контрольованих експериментів, але отримані результати були дуже поганими, незважаючи на налаштування гіперпараметрів. Зокрема, моделі машинного навчання з великими наборами даних погано масштабуються, і це негативно впливає на загальний час навчання.

3.3. Критерії відбору зображення за концептуальними кодами

Завдяки новому підходу до створення нового порівняльного аналізу та збалансованого набору даних ми визначили новий процес вибору критеріїв зображення. Для кожного вихідного класу або категорії Vienna ми вирішили вибрати 500 зображень на основі:

- вибір домінуючої категорії Vienna або кодів першого рівня зі списку кодів Vienna, якими маркується зображення торгової марки, наприклад:

а) 26 .11.01, 26.11.08, 29.01.01 -> домінуюча категорія 26

б) 03.01.15, 03.11.24 -> домінуюча категорія 03

с) 01.15.17 -> домінуюча категорія 01

- скинути зображення, де домінуючу категорію неможливо виділити через рівномірний розподіл категорій у списку кодів Vienna, якими позначено зображення торгової марки.

Процес, описаний вище, призвів до створення набору даних порівняльного аналізу, що складається з 14 500 зображень, і кожне зображення пов'язано з однією остаточною категорією Vienna. За допомогою цього підходу проблема перетворюється з класифікації зображень за

кількома мітками в класифікацію зображень за кількома мітками. На рисунку 3.3 нижче представлено збалансований набір даних, що складається з 500 зображень на категорію Vienna із загальною кількістю 14 500 зображень для нового набору даних порівняльного аналізу.

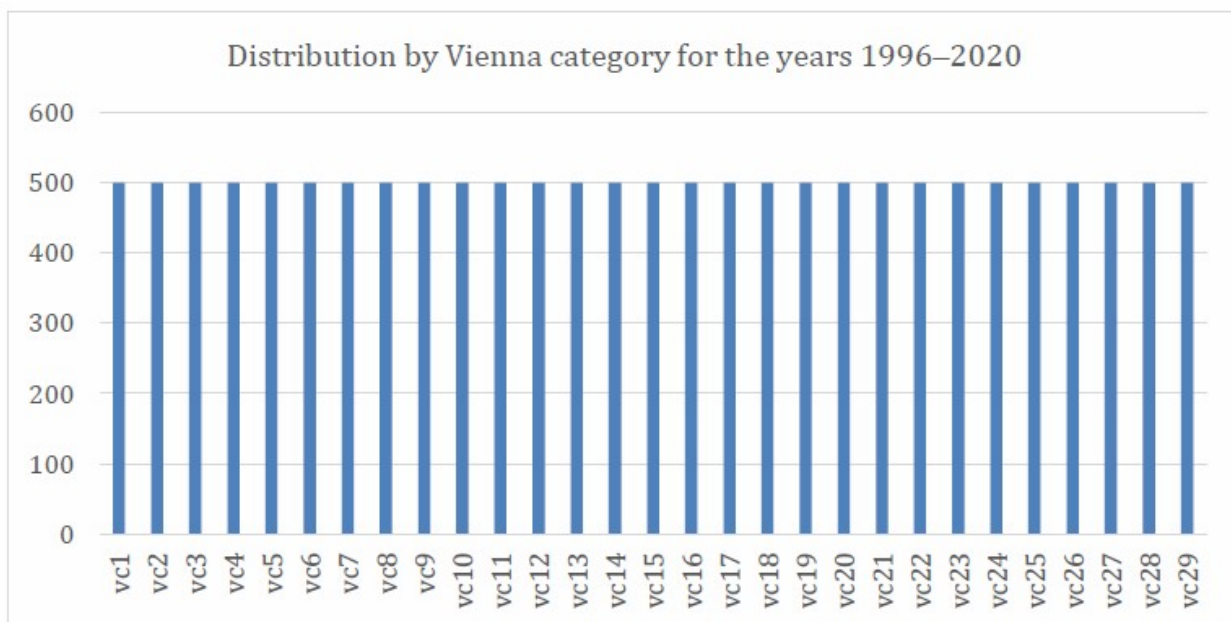


Рис. 3.3. Розподіл даних за категоріями за 1996-2020 роки для набору даних із 14 500 зображень

3.4. Метрики оцінювання результатів машинного навчання

Вибір правильної моделі для певного завдання дуже важливий для досягнення хороших результатів, а хороші результати можна інтерпретувати як хорошу продуктивність. Ефективні можливості вибраної моделі можна виміряти за допомогою точності, часу навчання, кількості параметрів, кількості функцій або лінійності. Якщо нашою основною метою є підрахувати кількість правильно прогнозованих результатів, тоді ми зосередимося на вимірюванні точності чи рівня помилок. У випадку, коли нашою метою є низький час навчання, слід виміряти час навчання та кількість параметрів, оскільки більше параметрів означає більше годин навчання. Щоб оцінити, наскільки добре працює модель, було введено кілька

показників класифікації, таких як точність, точність, запам'ятовування та оцінка F1. Щоб мати ширше та чіткіше розуміння під час оцінки загальної продуктивності даної моделі, ми використовуємо матрицю невідповідностей (рис. 3.4).

	P' (Predicted)	N' (Predicted)
P (Actual)	True Positive	False Negative
N (Actual)	False Positive	True Negative

Рис. 3.4. Матриця невідповідностей

Матриця невідповідностей представляє чотири комбінації міток, які є фактичними класами проти прогнозованих класів.

Таблиця 3.2.

Показники класифікації

Metric	Formula	Description
Accuracy	$\frac{TP + TN}{TP + FP + FN + TN}$	Overall model performance
Precision	$\frac{TP}{TP + FP}$	Degree of the positive predictions value accuracy
Recall	$\frac{TP}{TP + FN}$	Coverage of an actual positive sample (true positive or hit rate)
F1	$\frac{2TP}{2TP + FP + FN}$	Hybrid metric useful for unbalanced classes. Scores are marked between 0.0 and 1.0 where 1.0 is the best score.

Інші загальні показники, які використовуються для вимірювання продуктивності певної моделі глибокого навчання, перераховані в таблиці 3.2, надавши показники, які ми хочемо оцінити, математичні формули та опис.

3.5. Представлення результатів та їх аналіз

Цей розділ охоплює аналіз результатів, отриманих у результаті виконання контрольованих експериментів для кожної моделі, наприклад згорткової нейронної мережі, довгострокової пам'яті, рекурентної одиниці, опорних векторних машин, дерев рішень, випадкових лісів і простого Байєса.

3.5.1. Результати глибокого навчання

У таблиці 3.3 наведено огляд списку гіперпараметрів, які використовуються в моделях для виконання контрольованих експериментів. Для кожної моделі ми використовували однаковий розмір набору даних, який містить 14 500 зображень і розмір зображення 28x28.

Таблиця 3.3.

Гіперпараметри глибокого навчання

Parameter	Description
Activation function	tanh, relu, sigmoid
Output activation function	softmax
Loss function	Categorical cross entropy
Optimizer	Adam, SGD
Learning rate	0.001
Dataset size	14,500
Image size	28x28
Batch size	256

3.5.2. Згорткові нейронні мережі

Для згорткових нейронних мереж ми побудували шість моделей з ідентичною архітектурою, де перші три використовують Адама як алгоритм оптимізації, а решта три використовують Стохастичний градієнтний спуск (SGD) як алгоритм оптимізації. Структура архітектури моделей CNN складається з шести згорткових шарів і двох повністю з'єднаних шарів. Згорткові шари мають 32, 32, 64, 64, 128, 128 фільтрів, розмір ядра 3x3 і одну

з функцій активації, наприклад tanh, relu або sigmoid. Ініціалізація ваги досягається за допомогою ініціалізатора HeUniform у всіх згорткових шарах і першому щільному шарі. Усі шари, крім шостого й останнього, супроводжуються пакетною нормалізацією, що впливає на швидкість навчання, стабільність і полегшує роботу з кількома функціями активації. За другим, четвертим і шостим шарами йде максимальний шар об'єднання розміром 2x2. Для запобігання переобладнанню дропаут вводиться після шостого згорткового шару зі значенням дропауту 0,3. Шар випадання випадковим чином видаляє 30% нейронів. Усі згорткові шари сплющуються перед переходом до останніх двох щільних шарів.

Layer (type)	Output Shape	Param #
conv2d_22 (Conv2D)	(None, 28, 28, 32)	896
batch_normalization_29 (Batch Normalization)	(None, 28, 28, 32)	128
conv2d_23 (Conv2D)	(None, 28, 28, 32)	9248
batch_normalization_30 (Batch Normalization)	(None, 28, 28, 32)	128
max_pooling2d_11 (Max Pooling)	(None, 14, 14, 32)	0
conv2d_24 (Conv2D)	(None, 14, 14, 64)	18496
batch_normalization_31 (Batch Normalization)	(None, 14, 14, 64)	256
conv2d_25 (Conv2D)	(None, 14, 14, 64)	36928
batch_normalization_32 (Batch Normalization)	(None, 14, 14, 64)	256
max_pooling2d_12 (Max Pooling)	(None, 7, 7, 64)	0
dropout_8 (Dropout)	(None, 7, 7, 64)	0
flatten_6 (Flatten)	(None, 3136)	0
dense_12 (Dense)	(None, 128)	401536
batch_normalization_33 (Batch Normalization)	(None, 128)	512
dense_13 (Dense)	(None, 29)	3741

Total params: 472,125
 Trainable params: 471,485
 Non-trainable params: 640

Рис. 3.5. Інформація по моделі CNN_1

Останній вихідний шар — це щільний шар із 29 одиницями, які відповідають 29 категоріям, і softmax як функція активації. Для завдань класифікації зображень із кількома класами використовується функція активації softmax, оскільки вона містить численні межі рішень і повертає

розподіл ймовірностей за взаємовиключними вихідними класами. Моделі скомпільовані з використанням Адама та SGD як оптимізації зі швидкістю навчання, встановленою на 0,001, і функцією втрат встановлено категоріальну крос-ентропію. Набір даних було поділено на 70% набору для навчання, 20% набору для тестування та 10 % набору для перевірки. Архітектура базової моделі CNN візуалізується на рисунку 3.5.

З точки зору загальної продуктивності, усі моделі CNN, скомпільовані за допомогою Адама як алгоритму оптимізації та швидкості навчання, встановленої на 0,001, мають чудову продуктивність для завдання багатокласової класифікації зображень. Моделі тренуються протягом 100 epoch із терпінням ранньої зупинки, встановленим на 5, що означає, що метрика втрат відстежується, і якщо вона не зменшується протягом п'яти послідовних epoch, ми зупиняємо навчання, щоб уникнути переобладнання. Таблиця 3.4 надає список результатів набору даних навчання для кожної моделі CNN.

Таблиця 3.4.

Результати на основі моделей CNN

Model	Activation function	Optimizer	Accuracy	Precision	Recall	F1	Training time (sec)
CNN_1	tanh	Adam	0.9965	0.9730	0.9250	0.9484	1954
CNN_2	relu	Adam	0.9995	0.9957	0.9911	0.9934	1894
CNN_3	sigmoid	Adam	0.9996	0.9960	0.9925	0.9942	2174
CNN_4	tanh	SDG	0.9664	0.8937	0.0297	0.0575	3943
CNN_5	relu	SDG	0.9664	0.9034	0.0290	0.0562	3747
CNN_6	sigmoid	SDG	0.9661	0.8707	0.0220	0.0430	3804

Перша модель CNN_1 досягає точності 99,65%, точності 97,30%, запам'ятовування 92,50% і оцінки F1 94,84%. Друга модель, CNN_2, отримує показник точності 99,95%, показник точності 99,57%, показник запам'ятовування 99,34% і показник F1 99,34%. І третя модель, CNN_3, має

показник точності 99,96%, показник точності 99,60%, показник запам'ятовування 99,25 і показник F1 99,42, що робить її найефективнішою моделлю.

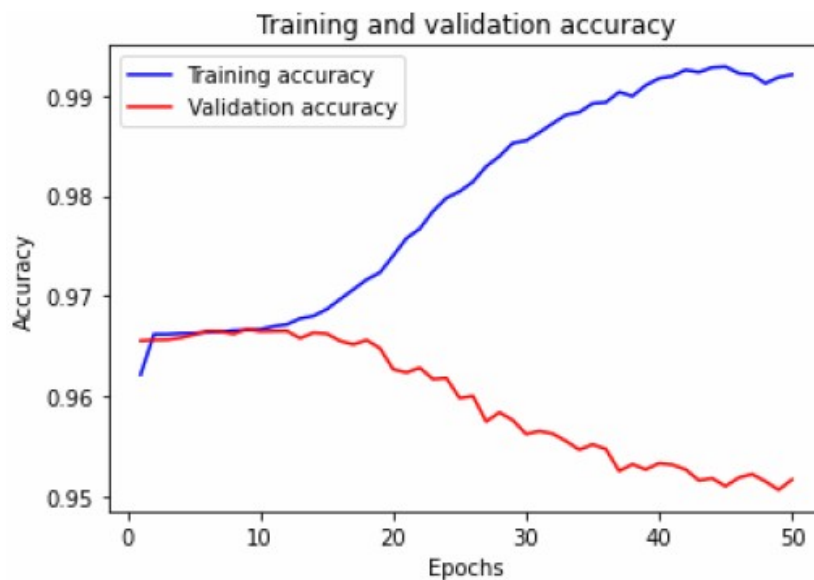


Рис. 3.6. Точність навчання та перевірки – модель CNN_3

Рисунок 3.6 вище ілюструє точність навчання та точність перевірки. Точність навчання та перевірки збігається між епохами 2–14, що є поворотним моментом, коли точність навчання починає зростати, а точність перевірки – знижуватися із загальним запасом ~3%. Це вказує на те, що параметри, обрані для моделі, не мають проблеми переобладнання. З базовою моделлю CNN результати навчання та точності перевірки дуже хороші, ми бачимо, що найнижче значення втрати перевірки становить 95,60% порівняно з найвищою точністю навчання, яка становить 99,96%.

На рисунку 3.7 показано, що втрати під час навчання для моделі CNN_3 починаються дуже високо, і вони демонструють бажане зменшення, що означає, що оптимізатор Адама здатний направляти модель для оновлення вагових матриць і значень зсуву в правильному напрямку. Тоді як втрати підтвердження діють у протилежному напрямку порівняно з втратами під час навчання.



Рис. 3.7. Втрати під час навчання та підтвердження – модель CNN_3

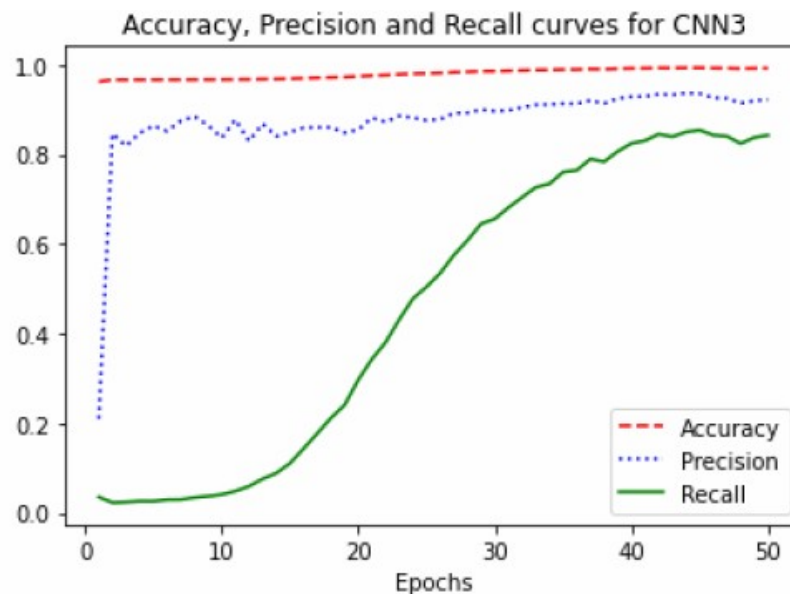


Рис. 3.8. Криві точності, прецизійності та відкликання - модель CNN_3

На рисунку 3.8 ми ілюструємо еволюцію точності, точності та збільшення кривої запам'ятовування для найефективнішої моделі CNN_3 на основі архітектури CNN. Точність має постійне невелике збільшення протягом усього часу навчання і зображена пунктирною червоною лінією. Точність починається приблизно з 20% і протягом кількох епох досягає пікового зростання приблизно на 82%, а потім з повільним збільшенням протягом усього часу навчання зростає до 99,60%, і це представлено синьою

пунктирною лінією. Згадайте, що еволюція має дуже низький старт і протягом 12 епох вона залишається близько ~2%.

3.5.3. Модель на основі архітектури LSTM

Моделі, побудовані на основі архітектури LSTM, мають однакову структуру архітектури. Вони складаються з двох шарів LSTM з 64, 64, 58, 29 одиницями, з формою введення (28, 84) і функціями активації, такими як \tanh , relu та sigmoid для шарів LSTM. За кожним шаром LSTM іде шар випадання та пакетна нормалізація для запобігання переобладнанню зі значенням випадання 0,2. Шари видалення випадковим чином видаляють 20% нейронів, а пакетна нормалізація впливає на швидкість навчання, стабільність і полегшує роботу з кількома функціями активації. Перш ніж перейти до повністю з'єднаних шарів, ми зводимо всі шари LSTM. Останні два шари є повністю пов'язаними або відомими як щільні шари, де перший щільний шар має 58 одиниць і налаштований на використання однієї з функцій активації, наприклад \tanh , relu або sigmoid . Останній повністю підключений рівень є вихідним рівнем з 29 одиницями, що відповідають 29 категоріям Vienna, і функцією активації є softmax . Моделі скомпільовано з використанням Адама як оптимізатора за замовчуванням зі швидкістю навчання, встановленою на 0,001, і функцією втрат встановлено категоріальну крос-ентропію.

На рисунку 3.9 вище показано збільшення точності еволюційного навчання та втрати підтвердження порівняно з втратою навчання та зниженням точності підтвердження для базової моделі LSTM. Крива, пов'язана з точністю навчання та втратою, представлена червоним і зеленим кольором, а також пунктирною лінією та суцільною лінією. Криві, пов'язані з точністю перевірки та втратою перевірки, представлені синім і блакитним кольорами, а також пунктирною та суцільною лінією.

Оскільки на рисунку 3.9 точність навчання та точність перевірки збігаються, ми візуалізуємо на масштабованому малюнку як точність

навчання, так і точність перевірки на рисунку 3.10. Точність перевірки, від початку до кінця навчання має стабільний хід зі зниженням лише на 3% порівняно з точністю навчання.

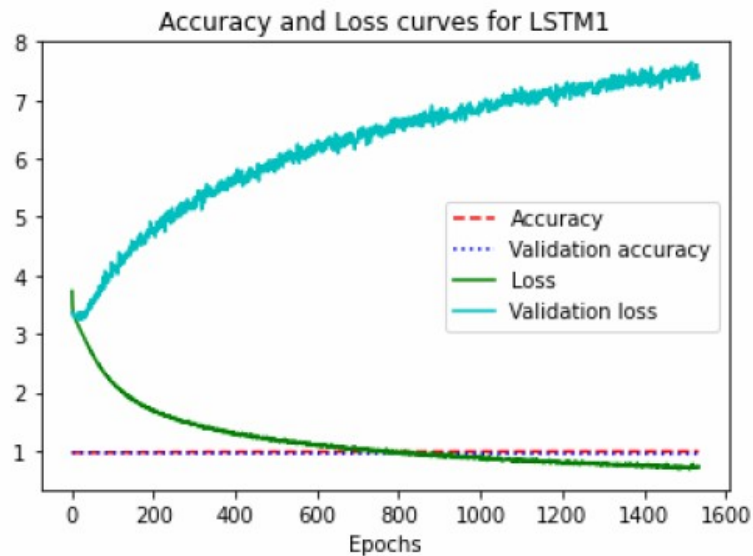


Рис. 3.9. Криві точності та втрат - модель LSTM

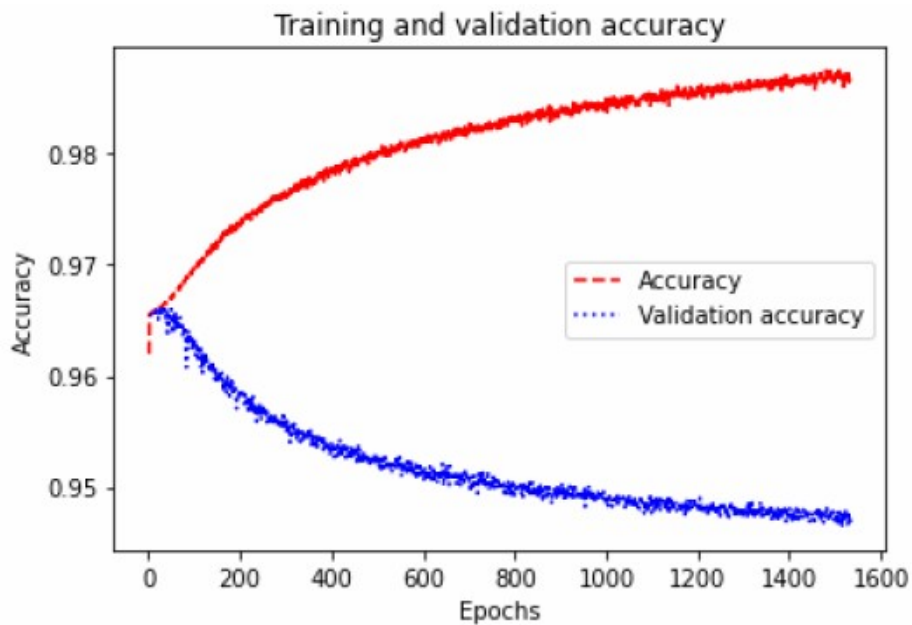


Рис. 3.10. Втрати під час навчання та підтвердження – модель LSTM

3.5.4. Рекурентна мережа

Друга партія моделей, створених на основі архітектур RNN, — це моделі Gated Recurrent Unit. Вони побудовані з використанням архітектури

GRU і мають однакову структуру архітектури. Вони складаються з двох рівнів GRU з 64, 64, 58, 29 одиницями, з формою введення (28, 84) і функціями активації, такими як \tanh , relu та sigmoid для рівнів GRU відповідно до перевіреної моделі. За кожним шаром GRU слідує шар вилучення та пакетна нормалізація для запобігання переобладнанню зі значенням вилучення 0,2. Шари вилучення випадковим чином видаляють 20% нейронів, а пакетна нормалізація впливає на швидкість навчання, стабільність і полегшує роботу з кількома функціями активації. Перш ніж перейти до повністю пов'язаних шарів, ми зводимо всі шари GRU. Останні два шари є повністю пов'язаними або відомими як щільні шари, де перший щільний шар має 58 одиниць і налаштований на використання однієї з функцій активації, наприклад \tanh , relu або sigmoid . Останній повністю підключений рівень є вихідним рівнем з 29 одиницями, що відповідають 29 категоріям Vienna, і функцією активації є softmax . Моделі скомпільовані з використанням Адама як оптимізатора за замовчуванням зі швидкістю навчання, встановленою на 0,001, і функцією втрат встановлено категориальну перехресну ентропію, яка використовується для завдання класифікації зображень із кількома класами.

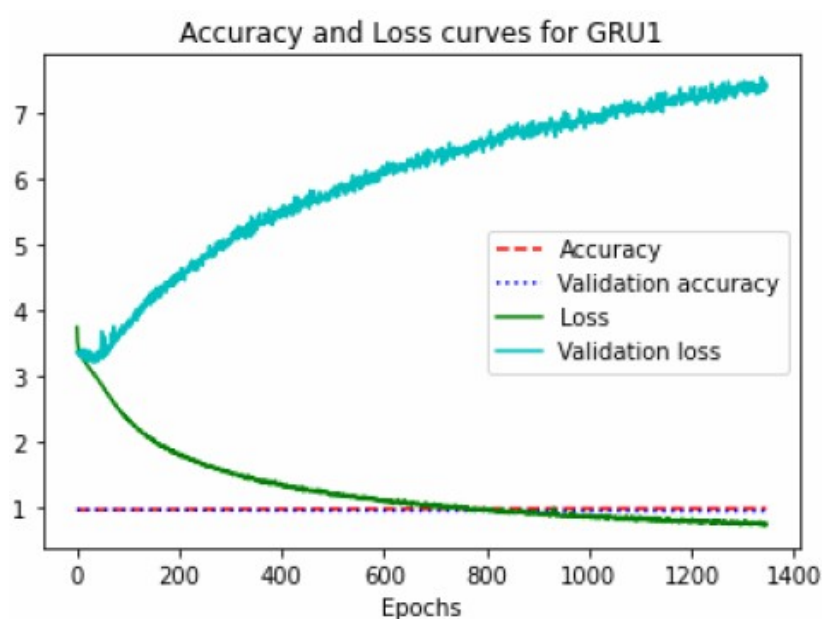


Рис. 3.11. Криві точності та втрат – модель GRU

На рисунку 3.11 показано збільшення точності еволюційного навчання та втрати підтвердження порівняно з втратою навчання та зниженням точності підтвердження для базової моделі GRU. Крива, пов'язана з точністю навчання та втратою, представлена червоним і зеленим кольором, а також пунктирною лінією та суцільною лінією. Криві, пов'язані з точністю перевірки та втратою перевірки, представлені синім і блакитним кольорами, а також пунктирною та суцільною лініями.

Оскільки на рисунку 3.11 точність навчання та точність перевірки збігаються, ми візуалізуємо на масштабованому малюнку як точність навчання, так і точність перевірки (див. рис. 3.12). Точність перевірки, від початку до кінця навчання має стабільний хід зі зниженням лише на 4% порівняно з точністю навчання.

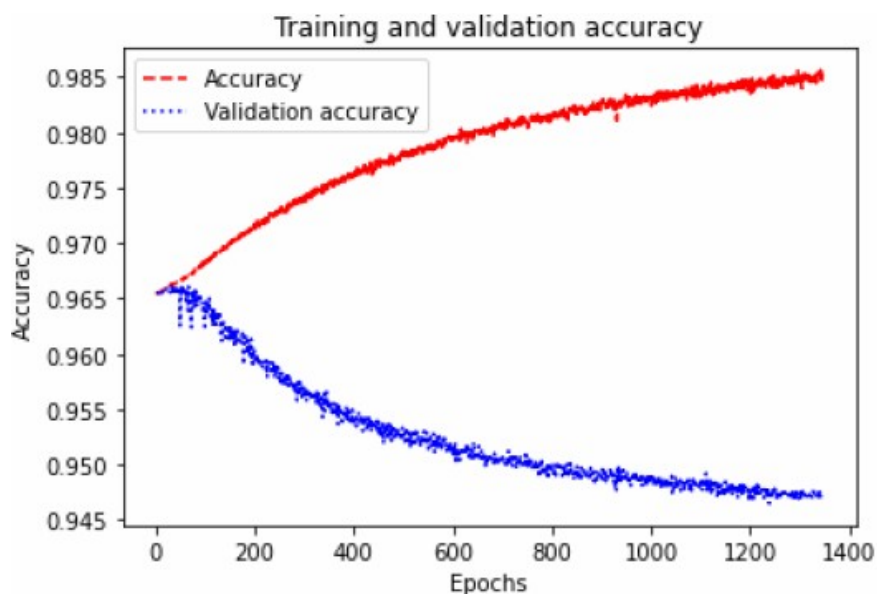


Рис. 3.12. Втрати під час навчання та підтвердження – модель GRU

З результатів, наданих для кожної моделі, ми можемо зробити висновок, що моделі CNN не тільки забезпечують вищу точність для прогнозованих класів, але також забезпечують більш відповідні прогнози. LSTM і GRU мають трохи нижчу точність передбачення, незважаючи на те, що обидва досягають 99% ступеня точності.

Доведено, що для загального часу навчання LSTM має втрату продуктивності, оскільки через конструкцію комірки пам'яті він працює повільніше порівняно з GRU, який має простіший дизайн. У наших зразках моделі CNN мають дещо складнішу архітектуру, оскільки вони мають більше прихованих рівнів порівняно з LSTM і GRU, але з кращим загальним часом навчання.

Результати цього дослідження чітко вказують на те, що моделі глибокого навчання перевершують моделі машинного навчання принаймні для завдання багатокласової класифікації зображення торгової марки. Зокрема, моделі CNN досягають найкращих результатів, за ними йдуть моделі LSTM, які займають друге місце, а потім моделі GRU, які займають третє місце.

Висновки до розділу

Третій розділ зосереджено на практичному застосуванні методології та моделей машинного навчання для ідентифікації елементів на зображеннях. В ньому висвітлено ключові етапи впровадження розроблених підходів, проведення експериментів та аналіз отриманих результатів. Представлена методологія включає всі аспекти роботи з моделями машинного навчання: від попередньої обробки даних до оцінювання надійності та валідності результатів. Контрольні імітаційні експерименти підтвердили ефективність запропонованих методів, дозволяючи оцінити їх продуктивність у реальних умовах. Результати показали, що глибоке навчання, зокрема згорткові нейронні мережі (CNN), продемонструвало найвищу точність в ідентифікації елементів на зображеннях. Модель на основі архітектури LSTM, попри свою орієнтацію на обробку послідовностей, успішно адаптована для задач зображень, що вказує на її потенціал у суміжних застосуваннях. Рекурентні нейронні мережі також показали прийнятні результати, проте їх ефективність була нижчою у порівнянні з CNN.

ВИСНОВКИ

У магістерській роботі було виконано комплексне дослідження застосування методів машинного навчання для ідентифікації та маркування елементів на зображеннях. Проведена робота охоплює широкий спектр аспектів від аналізу предметної області до впровадження та оцінювання практичних результатів.

Перший розділ заклав фундамент для дослідження, розкриваючи основні принципи штучного інтелекту та машинного навчання. Проаналізовано тенденції розвитку в цій галузі, зокрема прогрес у розробці методів обробки зображень, таких як згорткові нейронні мережі (CNNs). Було окреслено актуальну проблему: розпізнавання та ідентифікація об'єктів на зображеннях із високою точністю, що є критичним для багатьох прикладних задач.

Другий розділ присвячено детальному аналізу моделей машинного навчання. Розглянуто як класичні алгоритми (машина опорних векторів, дерева рішень, наївний Байєс), так і сучасні архітектури глибокого навчання (VGG Net, GoogLeNet, ResNet). Дослідження показало, що класичні методи можуть бути ефективними для простих задач, тоді як глибокі нейронні мережі демонструють високу продуктивність у складних сценаріях. Також оцінено переваги та недоліки різних фреймворків, які спрощують реалізацію та оптимізацію моделей.

У третьому розділі представлено результати практичного застосування розроблених підходів. Зокрема, попередня обробка даних та критерії їх відбору дозволили підвищити якість навчальних вибірок. Контрольні експерименти продемонстрували високу точність моделей, особливо згорткових нейронних мереж. Було успішно адаптовано модель на основі LSTM, що показало потенціал рекурентних архітектур у задачах обробки зображень. Результати були проаналізовані за допомогою метрик точності,

повноти та F-міри, що забезпечило об'єктивну оцінку ефективності запропонованих методів.

Таким чином, у роботі досягнуто основної мети – розроблено та проаналізовано методи машинного навчання для ідентифікації елементів на зображеннях. Результати дослідження можуть бути використані у прикладних сферах, таких як комп'ютерний зір, автоматизовані системи контролю та цифрова обробка даних. Робота також закладає перспективи для подальшого вдосконалення моделей, включаючи інтеграцію різних підходів і оптимізацію для обробки великих обсягів даних.

ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. J. McCarthy, "Artificial intelligence, logic and formalizing common sense," in *Philosophical logic and artificial intelligence*, Springer, 1989, pp. 161–190.
2. R. Pfeifer and F. Iida, "Embodied artificial intelligence: Trends and challenges," in *Embodied artificial intelligence*, Springer, 2004, pp. 1–26.
3. S. Pouyanfar et al., "A survey on deep learning: Algorithms, techniques, and applications," *ACM Comput. Surv.*, vol. 51, no. 5, pp. 1–36, 2018.
4. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Adv. Neural Inf. Process. Syst.*, vol. 25, pp. 1097–1105, 2012.
5. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of the 3rd International Conference on Learning Representations*, 2015, pp. 1–14.
6. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
7. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
8. Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.
9. LeCun, Y., Bengio, Y., & Hinton, G. (2015). "Deep Learning." *Nature*, 521(7553), 436–444.
10. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). "ImageNet Classification with Deep Convolutional Neural Networks." *Advances in Neural Information Processing Systems (NIPS)*.
11. Simonyan, K., & Zisserman, A. (2015). "Very Deep Convolutional Networks for Large-Scale Image Recognition." *International Conference on Learning Representations (ICLR)*.

- 12.He, K., Zhang, X., Ren, S., & Sun, J. (2016). "Deep Residual Learning for Image Recognition." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
- 13.Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). "You Only Look Once: Unified, Real-Time Object Detection." CVPR.
- 14.Girshick, R. (2015). "Fast R-CNN." ICCV.
- 15.Ren, S., He, K., Girshick, R., & Sun, J. (2015). "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks." NIPS.
- 16.Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). "Densely Connected Convolutional Networks." CVPR.
- 17.Szegedy, C., Liu, W., Jia, Y., et al. (2015). "Going Deeper with Convolutions." CVPR.
- 18.Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollar, P. (2017). "Focal Loss for Dense Object Detection." ICCV.
- 19.Zhao, Z. Q., Zheng, P., Xu, S. T., & Wu, X. (2019). "Object Detection with Deep Learning: A Review." IEEE Transactions on Neural Networks and Learning Systems.
- 20.Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2010). "The Pascal Visual Object Classes (VOC) Challenge." IJCV.
- 21.Russakovsky, O., Deng, J., Su, H., et al. (2015). "ImageNet Large Scale Visual Recognition Challenge." IJCV.
- 22.Long, J., Shelhamer, E., & Darrell, T. (2015). "Fully Convolutional Networks for Semantic Segmentation." CVPR.
- 23.Shelhamer, E., Long, J., & Darrell, T. (2017). "Fully Convolutional Networks for Semantic Segmentation." IEEE Transactions on Pattern Analysis and Machine Intelligence.
- 24.Ronneberger, O., Fischer, P., & Brox, T. (2015). "U-Net: Convolutional Networks for Biomedical Image Segmentation." MICCAI.

25. Badrinarayanan, V., Kendall, A., & Cipolla, R. (2017). "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation." *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
26. Mask R-CNN by He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). *ICCV*.
27. Lin, M., Chen, Q., & Yan, S. (2014). "Network In Network." *ICLR*.
28. Dosovitskiy, A., Beyer, L., Kolesnikov, A., et al. (2020). "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale." *ICLR*.
29. Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2017). "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs." *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
30. Lin, T.-Y., Dollar, P., Girshick, R., et al. (2014). "Microsoft COCO: Common Objects in Context." *ECCV*.
31. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). "ImageNet: A Large-Scale Hierarchical Image Database." *CVPR*.
32. Zeiler, M. D., & Fergus, R. (2014). "Visualizing and Understanding Convolutional Networks." *ECCV*.
33. Zhou, B., Khosla, A., Lapedriza, A., et al. (2016). "Learning Deep Features for Discriminative Localization." *CVPR*.
34. Xie, S., Girshick, R., Dollar, P., Tu, Z., & He, K. (2017). "Aggregated Residual Transformations for Deep Neural Networks." *CVPR*.
35. Kingma, D. P., & Ba, J. (2015). "Adam: A Method for Stochastic Optimization." *ICLR*.
36. Chollet, F. (2017). "Xception: Deep Learning with Depthwise Separable Convolutions." *CVPR*.
37. Carion, N., Massa, F., Synnaeve, G., et al. (2020). "End-to-End Object Detection with Transformers." *ECCV*.
38. Tan, M., & Le, Q. (2019). "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks." *ICML*.

39. Dosovitskiy, A., et al. (2021). "Vision Transformers (ViT)." ICLR.
40. Johnson, J., Krishna, R., Stark, M., et al. (2016). "Image Retrieval Using Scene Graphs." CVPR.
41. Radford, A., Kim, J. W., Hallacy, C., et al. (2021). "Learning Transferable Visual Models From Natural Language Supervision." ICML.
42. Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). "Image Quality Assessment: From Error Visibility to Structural Similarity." IEEE Transactions on Image Processing.
43. Olah, C., Mordvintsev, A., & Schubert, L. (2017). "Feature Visualization." Distill.
44. Brown, T. B., et al. (2020). "Language Models are Few-Shot Learners." NeurIPS.
45. N. Mohanty, T. M. Rath, A. Lee, and R. Manmatha, "Learning shapes for image classification and retrieval," in International Conference on Image and Video Retrieval, 2005, pp. 589–598.
46. Luo, W., Li, Y., Urtasun, R., & Zemel, R. (2016). "Understanding the Effective Receptive Field in Deep Convolutional Neural Networks." NIPS.
47. Xu, K., Ba, J., Kiros, R., et al. (2015). "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention." ICML.
48. S. Madhavan and M. T. Jones, "Deep learning architectures." <https://developer.ibm.com/technologies/artificial-intelligence/articles/cc-machine-learning-deep-learning-architectures>
49. X. Li and X. Wu, "Constructing long short-term memory based deep recurrent neural networks for large vocabulary speech recognition," in 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2015, pp. 4520–4524, doi: 10.1109/ICASSP.2015.7178826.