Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)
Кафедра Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження моделей і алгоритмів пошуку зображень
у сховищах великих даних
(тема)

Виконав:
студент 2 курсу, групи СШІм-22-1
Настенко С.В.
(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)
Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)
Освітня програма Системи штучного інтелекту
(повна назва спеціалізації)
Керівник проф. Смеляков К.С.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ В.О. Філатов
(підпис) (прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Штучного інтелекту
(повна назва)

Рівень вищої освіти другий (магістерський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

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

Освітня програма Системи штучного інтелекту
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«_____» _____ 20 ___ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Настенку Сергію Віталійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження моделей і алгоритмів пошуку зображень у сховищах великих даних

затверджена наказом університету від 1 квітня 20 24 р. № 260Ст
2. Термін подання студентом роботи до екзаменаційної комісії 6 червня 20 24 р.
3. Вихідні дані до роботи науково-технічні публікації, статті Інтернет-джерел, документація Keras, Tensorflow та numpy

4. Перелік питань, що потрібно опрацювати в роботі
1) Аналіз предметної області
2) Огляд існуючих моделей
3) Опис експериментів
4) Програмна реалізація та аналіз отриманих результатів

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	01.04.2024	виконано
2	Аналіз предметної області і постановка задачі	02.04.2024 - 20.04.2024	виконано
3	Дослідження моделей для вирішення задач	21.04.2024 - 08.05.2024	виконано
	пошуку зображень		
4	Розробка вимог до дослідження	09.05.2024 – 13.05.2024	виконано
5	Програмна реалізація	14.05.2024 – 18.05.2024	виконано
6	Аналіз результатів	19.05.2024 – 21.05.2024	виконано
7	Оформлення пояснювальної записки	22.05.2024 - 24.05.2024	виконано
8	Захист перед ЕК	06.06.2024	

Дата видачі завдання 1 квітня 20 24 р.

Студент _____
(підпис)

Керівник роботи _____ проф. Смеляков К.С.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 96 с., 4 рис., 1 табл., 2 дод., 30 джерел.

АЛГОРИТМИ ПОШУКУ, АНАЛІЗ ДАНИХ, ВЕЛИКІ ДАНІ, ІНДЕКСУВАННЯ ЗОБРАЖЕНЬ, МАШИННЕ НАВЧАННЯ, ПОШУК ЗОБРАЖЕНЬ.

Об'єкт дослідження – процеси та механізми пошуку зображень у великих сховищах даних.

Предмет дослідження – моделі і алгоритми, які застосовуються для ефективного пошуку зображень.

Мета роботи – аналіз та визначення найбільш ефективних моделей і алгоритмів для пошуку зображень у великих наборах даних з високою точністю та швидкістю.

Методи дослідження – детальний аналіз існуючих рішень у цій області, експериментальна розробка та тестування різних алгоритмів, порівняльний аналіз результатів.

У даній роботі проведено всебічний аналіз сучасних моделей машинного навчання та алгоритмів індексування зображень, з метою дослідження їх придатності для роботи з великими обсягами даних. В експериментах використано різноманітні набори даних, включаючи зображення з відкритих баз даних, щоб оцінити точність та швидкість роботи алгоритмів. Також було розглянуто можливості оптимізації процесу пошуку з метою підвищення продуктивності при роботі з великомасштабними даними. На основі аналізу результатів експериментів розроблено рекомендації щодо вибору та налаштування алгоритмів для оптимальної роботи пошуку зображень у великих сховищах даних.

ABSTRACT

Master's thesis contains: 96 pp., 4 fig., 1 tabl., 2 ann., 30 references.

BIG DATA, DATA ANALYSIS, IMAGE SEARCH, IMAGE INDEXING, MACHINE LEARNING, SEARCH ALGORITHMS.

Object of the research is the processes and mechanisms of image search in large data repositories.

Subject of the research are models and algorithms used for effective image search.

Aim of the research is to analyze and identify the most effective models and algorithms for image search in large datasets with high accuracy and speed.

Research methods is detailed analysis of existing solutions in this field, experimental development and testing of various algorithms, comparative analysis of results.

This research conducts a comprehensive analysis of modern machine learning models and image indexing algorithms to investigate their suitability for working with large volumes of data. Various datasets, including images from open databases, were used in the experiments to evaluate the accuracy and speed of the algorithms. The study also explored the possibilities of optimizing the search process to enhance performance when dealing with large-scale data. Based on the analysis of experimental results, recommendations were developed for selecting and configuring algorithms for optimal image search performance in large data repositories.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

AI – Artificial Intelligence – штучний інтелект;

AR – Augmented Reality – доповнена реальність;

Ball-дерево – структура даних для пошуку найближчих сусідів;

CBIR – Content-Based Image Retrieval – пошук зображень на основі контенту;

CIFAR-10 – Canadian Institute For Advanced Research – Канадський інститут передових досліджень;

CNN – Convolutional Neural Network – згорткова нейронна мережа;

DCGAN – Deep Convolutional Generative Adversarial Network – глибока згорткова генеративно-змагальна мережа;

F-measure – F-міра;

FCN – Fully Convolutional Network – повністю згорткова мережа;

GANs – Generative Adversarial Networks – генеративно-змагальні мережі;

GLCM – Grey Level Co-occurrence Matrix – матриця співвідношення рівнів сірого;

GPU – Graphics Processing Unit – графічний процесор;

HSV – Hue, Saturation, Value – відтінок, насиченість, яскравість;

ImageNet – великий візуальний набір даних для досліджень у галузі розпізнавання об'єктів;

IoT – Internet of Things – інтернет речей;

KD-Tree – K-Dimensional Tree – k-вимірне дерево;

KD-дерево – K-Dimensional Tree – k-вимірне дерево;

KD-дерева – K-Dimensional Trees – k-вимірні дерева;

k-NN – k-nearest neighbors – k найближчих сусідів;

LoG – Laplacian of Gaussian – Лапласіан Гауса;

MAE – Mean Absolute Error – середня абсолютна помилка;

mAP – mean Average Precision – середня точність;

MSE – Mean Squared Error – середньоквадратична помилка;

ReLU – Rectified Linear Unit – виправлена лінійна одиниця;

R-Tree – дерево R;

SSIM – Structural Similarity Index Measure – показник структурної подібності;

ViT – Vision Transformer – трансформер для зображень;

VGG – Visual Geometry Group – група візуальної геометрії;

YUV – колірний простір, який використовується в телевізійному відео.

ВСТУП

В епоху цифрових технологій експоненціальне зростання візуального контенту призвело до безпрецедентного збільшення обсягів великих сховищ даних, що робить ефективний пошук конкретних зображень критично важливим завданням. Системи пошуку зображень, які дозволяють шукати і знаходити зображення на основі їхнього змісту і метаданих, стали незамінними інструментами в різних галузях, включаючи медичну візуалізацію, електронні бібліотеки, електронну комерцію і платформи соціальних мереж. Значення цих систем полягає не лише в їхній здатності керувати величезними обсягами даних, але й у їхньому потенціалі для покращення користувацького досвіду, підтримки процесів прийняття рішень та сприяння дослідженням і розробкам у різних дисциплінах.

Метою цієї роботи є всебічний аналіз сучасних моделей та алгоритмів, що лежать в основі систем пошуку зображень, з особливим акцентом на їх застосуванні у великих сховищах даних. Це передбачає детальне вивчення теоретичних засад, на яких ґрунтується розробка цих систем, а також дослідження практичних реалізацій і проблем, що виникають у реальних сценаріях. Шляхом уточнення постановки проблеми та синтезу матеріалу з широкого кола літературних джерел ця робота має на меті узагальнити теоретичні аспекти пошуку зображень та розробити практичні компоненти кваліфікаційної роботи в цій галузі.

Для досягнення цих цілей буде використано методологічний підхід, який включає огляд існуючої літератури, аналіз різних моделей і алгоритмів пошуку зображень, а також визначення ключових проблем і майбутніх напрямків у цій галузі досліджень. Результат не лише сприятиме глибшому розумінню механізмів, що лежать в основі систем пошуку зображень, але й забезпечить основу для подальших інновацій та вдосконалення в управлінні та пошуку зображень у великих сховищах даних.

Цей вступ створює підґрунтя для ретельного вивчення теми, починаючи з огляду історії та літератури, який простежує еволюцію систем пошуку зображень і підкреслює їхню важливість. У наступних розділах розглядаються теоретичні засади, практичні підходи, тематичні дослідження, а також виклики і майбутні напрямки пошуку зображень.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Передумови, історичний контекст та еволюція

Розвиток систем пошуку зображень значно еволюціонував за останні кілька десятиліть, перейшовши від базових методів каталогізації до складних алгоритмів, здатних розуміти та інтерпретувати зміст зображень. Цей прогрес був зумовлений зростаючим попитом на ефективні та точні механізми пошуку зображень, що виник у зв'язку зі збільшенням кількості цифрових зображень у різних галузях. Література в цій галузі дуже обширна і відображає динамічну сферу досліджень, яка постійно намагається вирішити проблеми масштабованості, точності та релевантності пошуку зображень.

Перші спроби пошуку зображень були переважно текстовими і спиралися на метадані, які вручну анотувалися для категоризації та пошуку зображень. На ранніх етапах використовувалися прості системи каталогізації, де зображення мали супровідні текстові описи. Користувачі вводили ключові слова для пошуку, а система шукала відповідні описи. Однак зі зростанням колекцій цифрових зображень стали очевидними обмеження ручної анотації, що призвело до розробки наприкінці 20-го століття систем пошуку зображень на основі контенту (CBIR).

Системи CBIR знаменували собою значний зсув, оскільки вони були спрямовані на аналіз візуального змісту самих зображень – таких як колір, текстура і форма – для полегшення пошуку. Фундаментальна робота Сміта і Чанга у 1996 році над системою VisualSEEk представила одну з перших моделей, яка використовувала просторову кольорову індексацію для пошуку зображень, створивши прецедент для майбутніх досліджень у цій галузі. VisualSEEk була системою, де користувач міг розмістити кольорові блоки на сітці, що відповідали розташуванню кольорових регіонів у

зображенні. Це дозволяло здійснювати пошук зображень на основі кольорової структури, а не лише текстових описів.

Паралельно з розвитком CBIR, з’являлися методи, що покладалися на автоматичну екстракцію характеристик зображень. Методи машинного навчання, такі як класифікація на основі найближчих сусідів (k-NN) та кластеризація, почали використовувати для покращення точності пошуку. З часом розвиток обчислювальної потужності та доступність великих наборів даних дозволили впроваджувати все складніші алгоритми, які могли аналізувати багатовимірні характеристики зображень.

На початку 21-го століття важливим етапом у розвитку пошуку зображень стало впровадження методів глибокого навчання. Конволюційні нейронні мережі (CNN), завдяки своїй здатності автоматично вивчати та екстрактувати складні візуальні характеристики, значно покращили точність пошуку зображень. Перші роботи, такі як AlexNet (2012) і пізніші моделі, продемонстрували, що CNN можуть значно перевершувати традиційні методи в задачах класифікації та пошуку зображень. Ці моделі дозволили здійснювати пошук не лише за базовими характеристиками, а й за більш складними концептуальними ознаками, такими як об’єкти та сцени на зображенні.

Подальший розвиток методів глибокого навчання, таких як використання генеративно-змагальних мереж (GANs) і трансформерів, відкрив нові можливості для пошуку зображень. Наприклад, GANs дозволили створювати синтетичні зображення, які можуть використовуватися для навчання пошукових систем, а трансформери допомогли покращити контекстуальне розуміння візуальної інформації.

Сьогодні пошук зображень інтегрує різні підходи та алгоритми, використовуючи переваги як традиційних методів CBIR, так і новітніх досягнень у сфері глибокого навчання. Пошук за зображенням став важливою складовою багатьох застосувань, від медичної діагностики до розважальних сервісів. Постійно ведуться дослідження для покращення

ефективності, точності та швидкості цих систем, що включає оптимізацію алгоритмів, розширення наборів даних та інтеграцію нових технологій, таких як квантові обчислення та інтернет речей (IoT).

В результаті, еволюція пошуку зображень відображає значний прогрес у технологічному розвитку, що дозволяє ефективно обробляти та аналізувати великі обсяги візуальних даних.

1.2 Сучасні методології

Незважаючи на досягнення в галузі пошуку зображень, залишається кілька проблем, які потребують подальшого вирішення. Масштабованість пошукових систем для обробки дедалі більших сховищ даних без шкоди для швидкості та точності є постійною проблемою. Зі зростанням обсягів даних збільшується і необхідність у швидкій обробці запитів, що вимагає розробки ефективних алгоритмів та оптимізації інфраструктури.

Крім того, різноманітність зображень і суб'єктивність людської інтерпретації вимагають постійного вдосконалення алгоритмів для підвищення релевантності та задоволення потреб користувачів. Зображення можуть сильно відрізнятися за своїм змістом, стилем і контекстом, що створює додаткові труднощі для алгоритмів, які намагаються визначити семантичну схожість між зображеннями. Людське сприйняття зображень також є суб'єктивним, що ускладнює завдання створення універсальних алгоритмів, здатних задовольнити всі запити користувачів.

Однак ці виклики також відкривають можливості для інновацій. Інтеграція нових технологій, таких як доповнена реальність (AR), та вивчення альтернативних моделей представлення даних відкривають багатообіцяючі шляхи для майбутніх досліджень. AR може покращити інтерфейси пошукових систем, надаючи користувачам можливість взаємодіяти з візуальними даними в нових форматах. Наприклад, пошук зображень може бути інтегрований з AR-додатками для мобільних

пристроїв, що дозволить користувачам шукати інформацію про об'єкти в реальному часі, використовуючи камеру смартфона.

Сучасні методології пошуку зображень продовжують розвиватися, причому значна увага приділяється інтеграції алгоритмів машинного навчання для кращого представлення ознак і ефективності пошуку. Гібридні моделі, які поєднують традиційні методи вилучення ознак з підходами глибокого навчання, продемонстрували потенціал у подоланні обмежень кожного методу, коли вони використовуються ізольовано. Ці моделі можуть комбінувати переваги обох підходів, наприклад, використовувати глибоке навчання для екстракції високорівневих ознак і традиційні методи для швидкої обробки та порівняння цих ознак.

Крім того, використання аналітики великих даних і ресурсів хмарних обчислень почало вирішувати проблеми масштабованості обчислювальної потужності, необхідної для роботи з великими базами даних зображень. Хмарні обчислення надають можливість динамічно масштабувати ресурси в залежності від потреб системи, що забезпечує ефективну обробку великих обсягів даних. Аналіз великих даних дозволяє виявляти тенденції та патерни в користувацьких запитах, що може бути використано для оптимізації алгоритмів пошуку і підвищення їх точності.

Сучасні методології також включають використання розподілених систем для обробки запитів у реальному часі, що забезпечує швидкий доступ до великих баз даних зображень. Наприклад, технології на основі графових баз даних можуть використовуватися для моделювання складних взаємозв'язків між зображеннями, що дозволяє здійснювати більш точний та контекстуально релевантний пошук.

В результаті, сучасні методології пошуку зображень включають інтеграцію передових технологій машинного навчання, аналітики великих даних та хмарних обчислень, що дозволяє значно покращити ефективність, точність та релевантність пошуку. Оскільки обсяг цифрових зображень продовжує збільшуватися, важливість розробки ефективних, точних і

зручних для користувача систем пошуку зображень стає все більш першочерговою.

1.3 Постановка задачі

1.3.1 Основні концепції пошуку зображень

Системи пошуку зображень знаходяться на перетині управління базами даних, комп'ютерного зору та машинного навчання і призначені для організації, пошуку та вилучення зображень з великих наборів даних на основі запитів. Ці системи є важливою складовою інформаційних технологій, оскільки вони дозволяють ефективно обробляти та аналізувати великі обсяги візуальної інформації. Складність даних зображень, що характеризується їхньою багатовимірністю і різноманітною інформацією, яку вони передають, вимагає складних підходів для ефективного пошуку.

Характеристики даних зображень включають не лише видимі патерни, такі як колір і текстура, але також контекст і семантичну інформацію, яку зображення можуть представляти. Кольорові патерни можуть бути кількісно представлені за допомогою гістограм, які відображають розподіл кольорів у зображенні. Текстурні характеристики можуть бути визначені через аналіз частотних компонентів або візуальних текстурних елементів, таких як гладкість або зернистість. Однак, крім цих низькорівневих ознак, існує потреба у врахуванні високорівневих семантичних концепцій, таких як об'єкти, сцени та дії, що відбуваються на зображенні.

Проблема полягає у кількісному визначенні цих характеристик таким чином, щоб машини могли їх обробляти і порівнювати, долаючи розрив між числовими даними зображень і якісною інтерпретацією, яку здійснює людина. Наприклад, машина може легко обробити колірні гістограми або

текстурні матриці, але інтерпретація контексту сцени чи визначення об'єктів вимагає більш глибокого розуміння зображення.

Одна з ключових концепцій у пошуку зображень – це побудова моделей, які можуть перетворювати низькорівневі ознаки у високорівневі семантичні представлення. Такі моделі можуть бути створені за допомогою методів машинного навчання, які здатні навчатися на великих наборах даних і виявляти складні патерни та взаємозв'язки. Наприклад, згорткові нейронні мережі (CNN) навчаються виділяти та комбінувати різні рівні ознак, починаючи від простих країв і кольорових патернів до складних об'єктів і сцен.

Крім того, важливою складовою пошуку зображень є розробка ефективних алгоритмів індексації та порівняння зображень. Індексація дозволяє швидко знаходити релевантні зображення у великих базах даних. Поширеним підходом є використання індексів на основі деревоподібних структур або хешування, що дозволяє зменшити обсяг обчислень під час пошуку.

Сучасні підходи також включають використання гібридних моделей, які поєднують переваги різних методів для досягнення високої точності та ефективності. Наприклад, об'єднання традиційних методів обробки зображень із сучасними алгоритмами глибокого навчання дозволяє створювати потужні системи, здатні ефективно обробляти і аналізувати великі обсяги візуальної інформації.

1.3.2 Визначення вимог до функціоналу

Для ефективного дослідження моделей та алгоритмів пошуку зображень необхідно визначити основні вимоги до функціоналу дослідження. Ці вимоги включають параметри, що забезпечують точність, швидкість, масштабованість, адаптивність до різних типів даних та зручність проведення аналізу та порівняння алгоритмів.

Точність оцінки:

– вимірювання точності моделей повинно проводитися на основі різних метрик, таких як точність (precision), повнота (recall), F-мера та середня точність (mAP). Це дозволить отримати всебічну оцінку ефективності алгоритмів;

– проведення крос-валідації та тестування на різних наборах даних для забезпечення надійності та узагальненості результатів.

Швидкість обробки:

– оцінка швидкості роботи моделей включає час навчання, час індексації та час виконання пошукових запитів. Ці параметри важливі для розуміння ефективності алгоритмів в умовах великих обсягів даних;

– порівняння обчислювальної складності різних алгоритмів для визначення їх придатності для практичного застосування.

Масштабованість:

– важливо оцінити здатність алгоритмів працювати з великими обсягами даних. Для цього необхідно тестувати моделі на великих наборах зображень та аналізувати їх продуктивність при зростанні розміру даних;

– використання розподілених обчислень та оптимізація алгоритмів для забезпечення масштабованості.

Адаптивність до різних типів даних:

– оцінка алгоритмів повинна включати їх здатність працювати з різними типами зображень, включаючи фотографії, графіку, ілюстрації тощо. Це дозволить визначити універсальність моделей;

– проведення тестування на різних наборах даних для оцінки адаптивності алгоритмів до різних візуальних характеристик.

Зручність використання:

– важливо враховувати простоту налаштування та використання алгоритмів. Це включає можливість легко налаштовувати параметри моделей та запускати експерименти;

– документація та доступність інструментів для аналізу та візуалізації результатів повинні бути зручними для користувачів.

1.3.3 Алгоритми та моделі для дослідження

Для дослідження було обрано такі методи:

а) традиційні методи;

– гістограма кольорів: аналіз розподілу кольорів у зображеннях;

– виявлення країв: алгоритми Кенні та Собеля для визначення контурів об'єктів;

– аналіз текстури: використання перетворень Габора та матриць співвідношення рівнів сірого (GLCM);

б) методи машинного навчання;

– k-NN (k найближчих сусідів): класифікація зображень на основі подібності ознак;

– k-means: кластеризація зображень за подібністю ознак;

в) методи глибокого навчання;

– AlexNet: CNN для детального аналізу зображень;

– ResNet: глибока нейронна мережа з використанням резидуальних блоків;

– Inception: архітектура, що комбінує згорткові шари різних розмірів;

г) гібридні моделі;

– комбінація традиційних методів та глибокого навчання: використання традиційних методів для попередньої обробки та CNN для аналізу.

Процес дослідження та порівняння моделей та алгоритмів пошуку зображень включає кілька етапів. Спершу необхідно обрати відповідні набори даних для навчання та тестування моделей. Це можуть бути загальнодоступні набори, такі як CIFAR-10 або ImageNet.

Наступним етапом дослідження буде реалізація вибраних моделей, що буде здійснюватися за допомогою бібліотек Python, таких як TensorFlow, PyTorch, OpenCV та scikit-learn. Ці бібліотеки забезпечують широкий набір інструментів для створення та налаштування різноманітних моделей машинного навчання та глибокого навчання.

Після реалізації моделей буде проведене їх навчання на обраних наборах даних. Це включає налаштування параметрів моделей і оптимізацію процесу навчання для досягнення найкращих результатів. Після навчання моделі протестуються для оцінки їх ефективності за визначеними метриками, такими як точність, швидкість обробки та масштабованість.

Завершальним етапом буде порівняння результатів роботи моделей. На основі отриманих метрик буде зроблений аналіз ефективності кожної моделі, що дозволяє узагальнити результати та виявити найефективніші підходи до пошуку зображень у великих наборах даних. Це порівняння допоможе визначити переваги та недоліки різних алгоритмів та зробити висновки щодо їх придатності для різних умов та завдань.

1.3.4 Виклики у практичному впровадженні

Незважаючи на значний прогрес у розвитку систем пошуку зображень, практична реалізація стикається з численними проблемами, серед яких подолання семантичного розриву, забезпечення масштабованості та управління обчислювальними витратами. Семантичний розрив між низькорівневими ознаками, що витягуються алгоритмами, і високорівневим розумінням, яке сприймається користувачами, залишається критичною проблемою. Низькорівневі ознаки, такі як колір, текстура і форма, добре підходять для комп'ютерної обробки, але вони не завжди адекватно відображають концептуальне розуміння зображення, яке людина може отримати з першого погляду.

Для подолання цієї проблеми дослідники працюють над поєднанням візуальної інформації з текстовими метаданими або іншими модальностями, що надає більше контексту і значення знайденим зображенням. Наприклад, додавання описових тегів або використання інформації з соціальних мереж може допомогти краще інтерпретувати зміст зображень. Також застосування моделей обробки природної мови (NLP) для аналізу текстових описів і метаданих може значно покращити результати пошуку.

Ще однією важливою проблемою є масштабованість систем пошуку зображень. Зростання обсягів даних у геометричній прогресії вимагає розробки систем, які можуть підтримувати високу продуктивність без шкоди для точності пошуку. Для досягнення цієї мети використовуються розподілені системи обробки даних, що дозволяють розподілити навантаження між багатьма серверами. Хмарні обчислення також стають важливим інструментом для забезпечення масштабованості, оскільки вони дозволяють динамічно масштабувати ресурси в залежності від поточних потреб системи.

Окрім цього, управління обчислювальними витратами є ключовим фактором у розробці ефективних систем пошуку зображень. Використання високоефективних алгоритмів і структур даних, таких як KD-дерева і R-дерева, може значно знизити витрати на обчислення. Хешування також є корисним методом для швидкого пошуку зображень у великих базах даних. Однак, важливо збалансувати швидкість і точність, оскільки надмірне спрощення може призвести до втрати релевантності результатів.

Трансферне навчання є ще одним перспективним підходом, який дозволяє використовувати попередньо навчені моделі на великих наборах даних для підвищення ефективності та точності систем пошуку зображень, особливо в областях, де мічених даних недостатньо. Це дозволяє скоротити час і ресурси, необхідні для навчання моделей, і одночасно покращити їх продуктивність.

Практичні підходи до пошуку зображень охоплюють широкий спектр методів і методологій, від ефективного управління базами даних до інноваційних алгоритмічних стратегій і застосування передових моделей машинного навчання. Зокрема, важливо враховувати наступні аспекти:

– індексація та зберігання даних. Використання ефективних методів індексації, таких як хешування і деревоподібні структури, дозволяє швидко і точно знаходити релевантні зображення;

– аналіз зображень. Застосування моделей глибокого навчання, таких як CNN і FCN, для виділення високорівневих ознак із зображень. Це забезпечує точніше представлення змісту зображення і покращує результати пошуку;

– інтеграція мультимодальних даних. Комбінація візуальних ознак з текстовими метаданими або іншими модальностями допомагає краще інтерпретувати і знаходити зображення, що відповідають запитам користувачів;

– оптимізація обчислювальних ресурсів. Використання хмарних обчислень і розподілених систем для динамічного масштабування обчислювальних потужностей у відповідь на збільшення обсягів даних;

– управління семантичним розривом. Застосування методів NLP для аналізу текстових описів і метаданих зображень, а також розробка алгоритмів, які можуть краще поєднувати низькорівневі ознаки з високорівневими концепціями.

Для ілюстрації успіхів та викликів у цій галузі, можна поглянути на декілька прикладів. Ніу [4] пропонує новий метод пошуку зображень на основі контенту, який використовує злиття декількох ознак і модифікації дескрипторів мікроструктури для встановлення прямих зв'язків між ознаками зображення. Це покращує точність і швидкість пошуку, дозволяючи більш ефективно працювати з великими базами даних.

Аналогічно, Ахмед, Уммесафі та Ікбал [1] обговорюють пошук зображень на основі контенту з використанням злиття інформації про

особливості зображень. Вони підкреслюють важливість комбінування просторової інформації про колір з виділеними за формою ознаками та розпізнаванням об'єктів, що підвищує точність та релевантність результатів пошуку.

У сфері глибокого навчання, Фурута, Іноуе та Ямасакі [2] представили ефективну систему пошуку зображень, яка враховує як семантичну, так і просторову інформацію. Вони використовували повністю згорткові мережі (FCN), що демонструє здатність системи знаходити зображення за складними семантичними та просторовими запитами.

Ці дослідження демонструють, як сучасні технології та інноваційні методи можуть покращити точність і ефективність пошуку зображень. Однак, вирішення проблем семантичного розриву, масштабованості та обчислювальних витрат залишаються ключовими викликами, що потребують подальших досліджень і розробок.

Таким чином, постійне вдосконалення алгоритмів і моделей, інтеграція новітніх технологій і оптимізація обчислювальних ресурсів є критично важливими для розвитку ефективних систем пошуку зображень.

1.4 Приклади застосування

1.4.1 Медична візуалізація

У медичній галузі системи пошуку зображень трансформують діагностичні процедури, уможливлюючи ефективний пошук та аналіз медичних зображень. Haq, Moradi та Wang [3] представили фреймворк автоматизованого пошуку медичних зображень на основі глибоких спільнот для вилучення схожих зображень з великих баз даних рентгенівських знімків грудної клітки. У фреймворку було використано поєднання генерації ознак зображень на основі глибокого навчання і методів виявлення мережевих спільнот. Цей підхід дозволив досягти точності 85% у пошуку

зображень зі схожими позначками захворювань, що підкреслює потенціал передових систем пошуку зображень для підтримки клінічних.

1.4.2 Цифрові бібліотеки та архіви

Електронні бібліотеки та архіви використовують системи пошуку зображень для управління та надання доступу до величезних колекцій цифрових зображень. Прикладом такого застосування є впровадження систем пошуку зображень на основі контенту (CBIR), які дозволяють користувачам шукати історичні документи і твори мистецтва на основі візуальних характеристик, а не текстових описів. Використання процесів глибокого навчання для вилучення та класифікації ознак значно підвищило точність і ефективність пошуку в цих величезних і різноманітних колекціях.

1.4.3 Електронна комерція

У секторі електронної комерції системи пошуку зображень дозволяють покупцям знаходити товари за допомогою пошуку на основі зображень. Ця функція покращує користувацький досвід, дозволяючи покупцям завантажувати фотографії товарів, які вони хочуть придбати, або знаходити схожі товари. Система, розроблена Фурутою, Іноуе та Ямасакі [3], яка враховує як семантичну, так і просторову інформацію для пошуку зображень, є прикладом того, як розширені можливості пошуку інтегруються в платформи електронної комерції для підвищення релевантності пошуку та задоволеності клієнтів.

1.4.4 Соціальні мережі

Платформи соціальних мереж використовують технології пошуку зображень для упорядкування та пропонування контенту користувачам.

Впровадження ефективних та інтерактивних просторово- семантичних систем пошуку зображень дозволяє тонко підбирати зображення, які відповідають інтересам користувачів та критеріям пошуку [5]. Ці системи повинні працювати з величезними обсягами баз даних соціальних мереж, що робить масштабованість і швидкість роботи критично важливими завданнями.

1.4.5 Виклики та уроки

У цих тематичних дослідженнях можна виділити кілька спільних проблем, серед яких необхідність керувати дуже великими базами даних зображень, важливість мінімізації часу відгуку на пошук і складність подолання семантичного розриву між машинним представленням об'єктів і людським сприйняттям. Крім того, першочерговим завданням залишається забезпечення конфіденційності та безпеки конфіденційних даних, особливо в медичних і персональних базах даних зображень [6].

Застосування систем пошуку зображень у цих різноманітних галузях не лише демонструє їхню універсальність і корисність, але й підкреслює постійну потребу в дослідженнях і розробках, спрямованих на вирішення нових проблем управління цифровими зображеннями та їхнього пошуку. Завдяки постійним інноваціям і застосуванню передових обчислювальних моделей, системи пошуку зображень розширюватимуть свій вплив, пропонуючи більш складні і зручні для користувача можливості пошуку в різних галузях.

2 ОГЛЯД ІСНУЮЧИХ МОДЕЛЕЙ І АЛГОРИТМІВ

2.1 Традиційні методи

2.1.1 Методи на основі кольору

2.1.1.1 Гістограма кольорів

Гістограма кольорів є одним з найпоширеніших методів аналізу зображень у сфері комп'ютерного зору та обробки зображень. Цей метод використовується для представлення розподілу кольорів у зображенні і є основним інструментом у багатьох додатках, таких як розпізнавання об'єктів, індексація зображень, пошук за зображенням та покращення якості зображень. Гістограма кольорів є простим, але потужним методом, який забезпечує ефективне кодування кольорової інформації зображення.

Цей метод представляє собою графічне зображення розподілу кольорів у зображенні. Гістограма побудована шляхом підрахунку кількості пікселів кожного кольору в зображенні і відображення цих даних у вигляді графіка, де по осі X розташовані значення кольорів (або колірних каналів), а по осі Y – кількість пікселів кожного кольору. Зазвичай кольорова інформація у зображенні представлена в одному з кольорових просторів, таких як RGB, HSV або YUV. Найпоширенішим є простір RGB, де кольори представлені трьома компонентами: червоною (R), зеленою (G) та синьою (B). Гістограма кольорів може бути побудована для кожного з цих каналів окремо або для всього зображення в цілому.

Процес побудови гістограми кольорів включає кілька кроків. Спершу, якщо зображення представлене в кольоровому просторі відмінному від RGB, воно перетворюється у простір RGB або інший вибраний простір. Кольори в зображенні можуть бути квантовані, тобто зведені до певної кількості дискретних значень. Це дозволяє зменшити кількість унікальних

кольорів, що спрощує побудову гістограми. Наприклад, 8-бітне зображення має 256 можливих значень для кожного каналу R, G і B. Потім кількість пікселів кожного кольору підраховується і зберігається у вигляді масиву або іншої структури даних. Нарешті, дані про кількість пікселів кожного кольору відображаються у вигляді графіка, де по осі X розташовані значення кольорів, а по осі Y – кількість пікселів кожного кольору [7], [8].

Гістограма кольорів знаходить широке застосування у різних областях обробки зображень. Вона використовується для порівняння зображень на основі їх кольорової інформації. Зображення з подібними кольоровими розподілами матимуть подібні гістограми. Для порівняння гістограм можуть використовуватися різні метрики, такі як відстань Хеммінга, евклідова відстань або відстань Колмогорова-Смірнова. Також гістограма кольорів використовується для індексації зображень у базах даних, що дозволяє швидко знаходити зображення з подібними кольоровими характеристиками. Це особливо корисно у великих колекціях зображень, де пошук на основі кольору може бути першою стадією фільтрації. Окрім цього, гістограма кольорів може використовуватися для покращення якості зображень шляхом вирівнювання гістограми (histogram equalization), що дозволяє підвищити контрастність зображення. У деяких випадках кольорова інформація може бути критичною для розпізнавання об'єктів на зображеннях. Гістограма кольорів може допомогти у визначенні присутності певних об'єктів або сцен у зображенні.

Гістограма кольорів має кілька переваг. По-перше, побудова є відносно простою і не вимагає значних обчислювальних ресурсів. По-друге, гістограма кольорів забезпечує ефективне представлення кольорової інформації зображення, що дозволяє швидко порівнювати та індексувати зображення. По-третє, є інтуїтивно зрозумілою і легко інтерпретується, що робить її зручною для використання у різних додатках.

Проте метод має й певні недоліки. Гістограма кольорів не містить інформації про просторове розташування кольорів у зображенні, що може

бути критичним для деяких задач. Кольорова інформація може змінюватися в залежності від умов освітлення, що впливає на гістограму кольорів і може призвести до неправильних результатів порівняння. Гістограма кольорів містить тільки кольорову інформацію і не враховує інших важливих ознак зображення, таких як текстура або форма.

Таким чином, гістограма кольорів залишається одним з основних методів у сфері комп'ютерного зору та обробки зображень. Вона забезпечує просте і ефективне представлення кольорової інформації зображення, що дозволяє швидко і точно порівнювати, індексувати та покращувати якість зображень. Проте, для вирішення складніших задач може знадобитися використання додаткових методів, що враховують просторову інформацію та інші ознаки зображення.

2.1.1.2 Відстань між гістограмами

Відстань між гістограмами є важливим концептом в області обробки зображень і комп'ютерного зору. Вона використовується для вимірювання подібності або відмінності між двома зображеннями на основі їх гістограм кольорів. Гістограма кольорів, як вже обговорювалося, представляє розподіл кольорів у зображенні. Однак, для порівняння двох зображень необхідно визначити міру відстані між їх гістограмами. Існує кілька методів для обчислення цієї відстані, кожен з яких має свої особливості та застосування.

Відстань Хеммінга є одним із найпростішим методів вимірювання відстані між двома гістограмами. Вона обчислюється як сума абсолютних різниць між відповідними бінарними значеннями двох гістограм. Цей метод є швидким і простим у реалізації, але він може бути не дуже точним, якщо гістограми мають значні відмінності у своїх значеннях [9], [10].

Евклідова відстань є більш складним методом, який обчислює корінь квадратного кореня суми квадратів різниць між відповідними значеннями

двох гістограм. Цей метод враховує не тільки різницю між значеннями, але і їх квадрат, що дозволяє більш точно вимірювати відстань між гістограмами. Евклідова відстань є популярним методом у багатьох застосуваннях, оскільки вона забезпечує добрий баланс між точністю і обчислювальною складністю.

Відстань Колмогорова-Смірнова є статистичним методом для вимірювання відстані між двома гістограмами. Вона визначається як максимальна різниця між кумулятивними функціями розподілу двох гістограм. Цей метод є корисним, коли необхідно враховувати не тільки абсолютні значення, але і їх розподіл. Відстань Колмогорова-Смірнова є більш чутливою до змін у формі розподілу гістограм, що робить її особливо корисною у випадках, коли форма розподілу кольорів є важливою.

Відстань Бхаттачарія є ще одним методом для вимірювання подібності між двома гістограмами. Вона обчислюється на основі коефіцієнта Бхаттачарія, який визначає міру перекриття між двома розподілами ймовірностей. Відстань Бхаттачарія є ефективним методом для порівняння гістограм, особливо у випадках, коли гістограми мають схожі форми, але різні амплітуди. Цей метод використовується у багатьох додатках, таких як розпізнавання об'єктів і пошук за зображенням.

Метрика Кульбака-Лейблера є асиметричним методом для вимірювання відстані між двома гістограмами. Вона обчислює дивергенцію Кульбака-Лейблера, яка визначає міру відмінності між двома розподілами ймовірностей. Цей метод є корисним у випадках, коли необхідно вимірювати не тільки подібність, але і відмінність між гістограмами. Метрика Кульбака-Лейблера є особливо корисною у додатках, де важливо враховувати напрямок відмінності між розподілами.

Вибір методу для вимірювання відстані між гістограмами залежить від конкретного завдання і вимог до точності та обчислювальної складності. У деяких випадках може бути корисно використовувати кілька методів одночасно для отримання більш точних результатів. Наприклад, можна

використовувати евклідову відстань для швидкого порівняння гістограм і відстань Колмогорова-Смірнова для більш детального аналізу форми розподілу.

Практичне застосування відстані між гістограмами включає широкий спектр задач у комп'ютерному зорі та обробці зображень. Наприклад, у задачах індексації та пошуку зображень відстань між гістограмами може використовуватися для швидкого порівняння великої кількості зображень і знаходження найбільш схожих. У розпізнаванні об'єктів відстань між гістограмами може допомогти визначити наявність певних об'єктів у зображенні на основі їх кольорових характеристик. У задачах покращення якості зображень відстань між гістограмами може використовуватися для оцінки ефективності різних методів обробки і покращення зображень.

Відстань між гістограмами є потужним інструментом для аналізу зображень, який забезпечує ефективне вимірювання подібності та відмінності між зображеннями на основі їх кольорових характеристик. Вибір конкретного методу для вимірювання відстані залежить від завдання і вимог до точності та обчислювальної складності, але у будь-якому випадку цей інструмент є незамінним у багатьох додатках обробки зображень і комп'ютерного зору.

2.1.2 Методи на основі текстури

2.1.2.1 Перетворення Габора

Перетворення Габора є розповсюдженим інструментом у сфері обробки зображень та комп'ютерного зору, який використовується для виділення текстурних ознак зображень. Воно базується на застосуванні фільтрів Габора, які мають властивості просторової локалізації та вибірковості до частоти і орієнтації, що робить їх особливо корисними для аналізу текстур та структур зображень.

Фільтр Габора є комплексним функціоналом, який поєднує гармонічну функцію (синусоїду) з гауссовою функцією, що визначає просторову локалізацію. Математично двовимірний фільтр Габора можна виразити як

$$G(x, y, \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right)\cos\left(2\pi\frac{x`}{\lambda} + \psi\right), \qquad (2.1)$$

де x' = x cos θ + y sin θ;

y' = -x sin θ + y cos θ;

λ – довжина хвилі синусоїди;

θ – орієнтація фільтра;

ψ – фазове зміщення синусоїди;

σ – стандартне відхилення гауссового вікна;

γ – просторове співвідношення, що визначає еліптичність фільтра.

Ця формула описує, як фільтр Габора накладається на зображення для виділення текстурних характеристик. Перетворення Габора застосовує набір таких фільтрів з різними параметрами до зображення, що дозволяє виявити текстурні ознаки на різних масштабах і орієнтаціях.

Процес застосування перетворення Габора включає кілька етапів. Спершу необхідно вибрати набір параметрів фільтрів Габора, таких як довжина хвилі, орієнтація, фазове зміщення, стандартне відхилення та просторове співвідношення. Зазвичай використовують набір фільтрів з різними орієнтаціями (наприклад, 0°, 45°, 90°, 135°) та масштабами [11]. Потім кожен з цих фільтрів застосовується до зображення шляхом конволюції, що дозволяє виділити певні текстурні ознаки, які відповідають параметрам фільтра. Результати конволюції кожного фільтра використовуються для побудови енергетичних карт, які представляють локальну енергію текстурних ознак на зображенні. Кожна карта відповідає певному набору параметрів фільтра. Далі енергетичні карти агрегуються для

створення кінцевого набору текстурних ознак, що характеризують зображення. Цей набір ознак може використовуватися для подальшого аналізу, наприклад, для класифікації або сегментації зображень.

Перетворення Габора знаходить широке застосування в різних областях обробки зображень та комп'ютерного зору. Воно є ефективним інструментом для аналізу текстурних характеристик зображень, дозволяючи виявити текстурні патерни на різних масштабах і орієнтаціях, що може бути корисним для класифікації текстур, сегментації зображень і розпізнавання об'єктів. У біометричних системах фільтри Габора широко використовуються, зокрема в системах розпізнавання відбитків пальців та іридії. Вони дозволяють виділити унікальні текстурні ознаки, які є стійкими до різних умов освітлення та змін у зображенні. У системах розпізнавання облич фільтри Габора використовуються для виділення текстурних ознак, які характеризують різні частини обличчя, такі як очі, ніс та рот. Це дозволяє побудувати стійкі та інформативні дескриптори для розпізнавання облич за різних умов. У медичних зображеннях перетворення Габора використовується для аналізу текстурних ознак, таких як знімки МРТ або КТ, дозволяючи виділити текстурні патерни, які можуть бути пов'язані з різними патологічними станами або аномаліями.

Перетворення Габора має кілька важливих переваг. Воно забезпечує локалізацію текстурних ознак у просторі та частоті, що робить його ефективним для аналізу текстур на різних масштабах і орієнтаціях. Фільтри Габора є інваріантними до змін освітлення, що дозволяє використовувати їх у різних умовах зйомки. Однак, перетворення Габора має і певні недоліки. Воно може бути обчислювально складним, особливо при використанні великої кількості фільтрів з різними параметрами. Крім того, фільтри Габора є фіксованими, що може обмежувати їх здатність адаптуватися до різних типів текстур.

Таким чином, перетворення Габора є потужним інструментом для аналізу текстурних ознак зображень, який знаходить широке застосування в

обробці зображень та комп'ютерному зорі [12]. Воно забезпечує ефективну локалізацію текстурних ознак у просторі та частоті, що робить його особливо корисним для задач класифікації, сегментації та розпізнавання об'єктів. Незважаючи на певні обмеження, перетворення Габора залишається одним з основних методів аналізу текстур у сучасних системах обробки зображень.

2.1.2.2 Матриця співвідношення рівнів сірого (GLCM)

Матриця співвідношення рівнів сірого (GLCM) є важливим інструментом для аналізу текстур у зображеннях, широко використовуваним у сфері обробки зображень та комп'ютерного зору. Вона дозволяє кількісно оцінити текстурні властивості зображення, вивчаючи просторові взаємозв'язки між пікселями різних відтінків сірого.

GLCM побудовується на основі просторових відносин між парами пікселів з різними значеннями яскравості. Це робиться шляхом створення матриці, елементи якої вказують на кількість разів, коли певна пара значень яскравості зустрічається на заданій відстані і під певним кутом у зображенні. Для побудови GLCM використовуються такі параметри, як напрямок (0°, 45°, 90°, 135°) і відстань між пікселями (зазвичай один або більше пікселів).

Процес створення GLCM включає кілька кроків. Спершу зображення переводиться в відтінки сірого, якщо воно було кольоровим. Далі визначається відстань і напрямок між пікселями, що розглядаються. Наприклад, для напрямку 0° та відстані 1 розглядаються сусідні пікселі в горизонтальному напрямку. Потім обчислюється частота появи кожної пари значень яскравості для всіх пар пікселів у заданому напрямку та на заданій відстані [14]. Це робиться для всіх напрямків, після чого матриці нормалізуються шляхом ділення кожного елемента на загальну кількість пар пікселів.

GLCM дозволяє обчислювати різноманітні текстурні ознаки, серед яких:

Контраст: Вимірює локальні варіації в матриці. Високе значення контрасту вказує на більшу різницю між сусідніми пікселями

$$Contrast = \sum_{i,j}(i-j)^2 * P(i,j)\,. \tag{2.2}$$

Ентропія: Вимірює випадковість текстури. Високе значення ентропії вказує на складну текстуру з високою варіацією значень яскравості

$$Entropy = \sum_{i,j} P(i,j) log P(i,j). \tag{2.3}$$

Однорідність: Вимірює близькість розподілу елементів GLCM до головної діагоналі. Високе значення однорідності вказує на те, що сусідні пікселі мають схожі значення яскравості

$$\text{Homogeneity} = \sum_{i,j} \frac{\text{P(i,j)}}{1 + |\text{i} - \text{j}|}\,. \tag{2.4}$$

Енергія: Вимірює суму квадратів елементів GLCM, що відображає повторюваність текстурних патернів

$$\text{Energy} = \sum_{i,j} \text{P(i,j)}^2. \tag{2.5}$$

GLCM використовується для класифікації текстур, сегментації зображень, розпізнавання об'єктів та аналізу медичних зображень. Наприклад, у медичних дослідженнях GLCM застосовується для аналізу знімків МРТ або КТ, допомагаючи виділити текстурні патерни, пов'язані з патологічними станами.

GLCM має кілька важливих переваг. Вона забезпечує кількісну оцінку текстурних властивостей зображення, дозволяючи виділити важливі ознаки для класифікації та розпізнавання. Крім того, GLCM є інтуїтивно зрозумілою і легко обчислюється, що робить її зручною для використання в різних додатках.

Однак, GLCM має і деякі недоліки. Вона чутлива до змін масштабу та обертання зображення, що може вимагати додаткових перетворень для нормалізації. Крім того, обчислення GLCM для великих зображень може бути обчислювально складним і вимагати значних ресурсів.

Незважаючи на ці обмеження, GLCM залишається одним з основних інструментів для аналізу текстур у зображеннях. Її здатність виділяти важливі текстурні ознаки робить її незамінним інструментом у багатьох додатках обробки зображень та комп'ютерного зору. Завдяки простоті реалізації та високій ефективності, GLCM продовжує залишатися популярним методом для аналізу текстур і знаходить нові застосування в різних галузях.

2.1.3 Методи на основі форми

2.1.3.1 Виявлення країв

Виявлення країв є важливим методом та широко використовуваним у комп'ютерному зорі та обробці зображень. Краї на зображенні представляють собою межі між різними об'єктами або областями з різними рівнями яскравості, і їхнє виявлення є ключовим кроком у багатьох алгоритмах сегментації, розпізнавання об'єктів та аналізу зображень [13].

Виявлення країв спрямоване на виділення пікселів, де відбувається значна зміна інтенсивності, що зазвичай відповідає межам об'єктів. Існує кілька методів для виявлення країв, серед яких найпопулярнішими є оператори Собеля, Кенні та Лапласіан Гауса (LoG).

Оператор Собеля є простим і ефективним методом для виявлення країв, який використовує два 3x3 маски для обчислення градієнтів зображення в горизонтальному та вертикальному напрямках. Маски Собеля виглядають наступним чином:

– горизонтальна маска (Gx):

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}; \tag{2.6}$$

– вертикальна маска (Gy):

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}. \tag{2.7}$$

Ці маски конволюються з зображенням, і результати використовуються для обчислення величини градієнта

$$G = \sqrt{Gx^2 + Gy^2}, \tag{2.8}$$

де Gx та Gy – результати конволюції з горизонтальною та вертикальною масками відповідно. Виявлені градієнти вказують на краї зображення.

Оператор Кенні є більш складним методом для виявлення країв, який включає кілька етапів: розмиття, обчислення градієнта, не максимальне пригнічення та подвійне порогування. Першим кроком є застосування гауссового розмиття для зменшення шуму на зображенні. Далі обчислюється градієнт зображення за допомогою операторів Собеля або інших градієнтних операторів. Наступним кроком є не максимальне пригнічення, яке видаляє пікселі, що не є локальними максимумами у напрямку градієнта. Останнім етапом є подвійне порогування, яке дозволяє

визначити слабкі та сильні краї і об'єднати їх для отримання кінцевого результату.

Лапласіан Гауса (LoG) є ще одним методом для виявлення країв, який використовує гауссове розмиття для зменшення шуму та оператор Лапласа для виявлення країв. Оператор Лапласа обчислює другу похідну зображення, що дозволяє виявити області, де зміна інтенсивності досягає максимуму. Формула для оператора Лапласа виглядає наступним чином:

$$\nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}, \tag{2.9}$$

де I – зображення;

∇2 – оператор Лапласа.

Після розмиття зображення гауссовим фільтром застосовується оператор Лапласа, і визначаються точки, де друга похідна змінює знак, що відповідає краям на зображенні.

Виявлення країв має широкий спектр застосувань у різних областях. Воно використовується для сегментації зображень, що дозволяє розділити зображення на окремі об'єкти або області [15], [16]. У задачах розпізнавання об'єктів виявлення країв допомагає визначити контури об'єктів, що спрощує їх подальшу ідентифікацію. У медичних зображеннях виявлення країв використовується для виділення анатомічних структур та виявлення патологій.

Переваги виявлення країв включають його здатність ефективно виділяти важливі контури та межі об'єктів на зображенні, що значно покращує точність сегментації та розпізнавання. Однак методи виявлення країв також мають певні недоліки [17]. Вони можуть бути чутливими до шуму на зображенні, що може призвести до виявлення фальшивих країв. Крім того, деякі методи можуть вимагати налаштування параметрів, таких як порогові значення, що може бути складним завданням.

Виявлення країв залишається важливим інструментом у сфері обробки зображень та комп'ютерного зору. Його здатність виділяти контури та межі об'єктів робить його незамінним у багатьох додатках, включаючи сегментацію, розпізнавання об'єктів та аналіз медичних зображень. Завдяки постійному вдосконаленню методів та алгоритмів, виявлення країв продовжує залишатися ключовим компонентом сучасних систем обробки зображень.

2.1.3.2 Описувачі контурів

Описувачі контурів дозволяють ефективно кодувати інформацію про контури об'єктів, що робить їх незамінними у багатьох додатках, таких як розпізнавання об'єктів, класифікація форм та виявлення аномалій. Описувачі контурів дозволяють зменшити складність даних, зберігаючи при цьому ключові характеристики форми, що полегшує подальшу обробку та аналіз.

Існує кілька підходів до опису контурів, серед яких найпоширенішими є фур'є-описувачі та Shape Context.

Фур'є-описувачі використовують перетворення Фур'є для кодування форми контурів у частотній області [18], [19]. Спершу контур об'єкта представляється як комплексна функція, де координати кожної точки контуру (x, y) інтерпретуються як комплексні числа. Потім застосовується перетворення Фур'є для отримання спектрального представлення контуру. Отримані коефіцієнти Фур'є використовуються як описувачі форми. Фур'є-описувачі мають кілька важливих властивостей: вони є інваріантними до зсуву, масштабування та обертання, що робить їх дуже зручними для аналізу форм. Вони також дозволяють легко відфільтровувати високочастотні компоненти, зменшуючи вплив шуму та дрібних деталей.

Shape Context є іншим популярним методом для опису контурів, запропонованим Сергієм Белонучиком і Девідом Лоу у 2001 році. Цей метод

заснований на аналізі відносних положень точок на контурі. Для кожної точки на контурі створюється гістограма, яка описує відстань і напрямок до інших точок на контурі. Гістограма складається з кількох промінів, які поділяють простір навколо точки на кілька секторів за відстанню і кутом. Кожен промінь містить кількість точок, що потрапляють у відповідний сектор. Отримані гістограми використовуються як описувачі форми. Shape Context має високу стійкість до змін форми і дозволяє ефективно порівнювати контури різних об'єктів.

Інший підхід до опису контурів – це моменти інваріантів. Цей метод базується на обчисленні моментів контурів, які є числовими характеристиками, що описують форму. Найчастіше використовуються моменти Гу, які є інваріантними до зсуву, масштабування та обертання. Моменти Гу обчислюються на основі координат точок контуру і дозволяють кодувати глобальні характеристики форми [20]. Вони широко використовуються в задачах класифікації та розпізнавання об'єктів завдяки своїй простоті та ефективності.

Описувачі контурів знаходять широке застосування у різних областях обробки зображень та комп'ютерного зору. Вони використовуються для розпізнавання об'єктів, де форми об'єктів можуть бути ключовою характеристикою для ідентифікації. Наприклад, у системах розпізнавання рукописного тексту описувачі контурів допомагають визначити форми літер і цифр. У біометричних системах описувачі контурів використовуються для ідентифікації осіб за формою обличчя, вух або інших анатомічних ознак.

Описувачі контурів також важливі для класифікації об'єктів, де об'єкти поділяються на класи на основі їхніх форм. Це може бути корисно в медичних додатках, де форми клітин або органів можуть бути використані для діагностики захворювань. В автоматизованих системах контролю якості описувачі контурів допомагають виявити дефекти на виробничих лініях, аналізуючи форми виробів.

Переваги описувачів контурів включають їхню здатність ефективно кодувати форми та зберігати ключові характеристики, що дозволяє використовувати їх у різних задачах обробки зображень. Вони є інваріантними до зсуву, масштабування та обертання, що робить їх особливо корисними у додатках, де об'єкти можуть бути представлені у різних положеннях та розмірах. Крім того, багато методів опису контурів, такі як фур'є-описувачі та моменти Гу, є відносно простими у реалізації та обчисленні.

Однак описувачі контурів також мають певні недоліки. Деякі методи можуть бути чутливими до шуму та дрібних деталей на контурах, що може вплинути на точність аналізу. Крім того, у випадках складних або дуже варіативних форм, описувачі можуть вимагати додаткових алгоритмів для нормалізації та попередньої обробки контурів.

Описувачі контурів є важливим інструментом у сучасних системах обробки зображень і комп'ютерного зору [21], [22]. Вони дозволяють ефективно кодувати і аналізувати форми об'єктів, що робить їх незамінними у багатьох додатках, таких як розпізнавання об'єктів, класифікація форм та виявлення аномалій. Завдяки постійному вдосконаленню методів та алгоритмів, описувачі контурів продовжують залишатися ключовим компонентом сучасних систем обробки зображень.

2.2 Методи машинного навчання

2.2.1 Методи на основі підходів k-NN (k найближчих сусідів)

Методи на основі підходів k-NN (k найближчих сусідів) є популярними і широко використовуваними в області машинного навчання для задач класифікації та регресії. Ці методи базуються на принципі порівняння нових даних з наявними прикладами і визначенні класу або значення на основі найближчих сусідів у багатовимірному просторі ознак.

Метод k-NN є непараметричним, що означає, що він не робить жодних припущень щодо розподілу даних [23]. Основна ідея методу полягає в тому, що для класифікації нового зразка визначається k найближчих сусідів з навчального набору даних, де k є параметром, що задається користувачем. Клас нової точки визначається більшістю голосів її сусідів. У випадку регресії значення нової точки визначається як середнє значення її найближчих сусідів.

Процес реалізації методу k-NN включає кілька ключових кроків. По-перше, необхідно визначити метрику відстані, яка буде використовуватися для вимірювання відстані між точками в просторі ознак. Найпоширенішими метриками є евклідова відстань, манхеттенська відстань та відстань Махаланобіса. Евклідова відстань обчислюється як корінь квадратного кореня суми квадратів різниць між відповідними координатами точок:

$$d(p,q) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2}, \tag{2.10}$$

де p і q – точки у просторі ознак;

n – кількість ознак.

Після визначення метрики відстані, для кожного нового зразка обчислюються відстані до всіх точок навчального набору даних, і вибираються k найближчих сусідів. Для класифікації нової точки використовується більшість голосів її сусідів. Це може бути формалізовано наступним чином:

$$\hat{y} = \arg max_{c \in C} \sum_{i=q}^{k} \mathbb{I}(y_i = c), \tag{2.11}$$

де $\hat{y}$ – передбачений клас;

C – множина можливих класів;

y_i – клас i-го сусіда;

𝕀 – індикаторна функція, яка дорівнює 1, якщо вираз істинний, і 0 в іншому випадку.

У випадку регресії, передбачене значення нової точки обчислюється як середнє значення її найближчих сусідів:

$$\hat{y} = \frac{1}{k}\sum_{i=q}^{k} y_i, \tag{2.12}$$

де $\hat{y}$ – передбачене значення;

y_i – значення і-го сусіда.

Метод k-NN має кілька важливих переваг. По-перше, він є простим у реалізації та інтуїтивно зрозумілим. По-друге, він є непараметричним, що дозволяє йому ефективно працювати з даними, які не підкоряються певним розподілам. По-третє, метод k-NN може легко адаптуватися до нових даних, оскільки для навчання не потрібно проводити обчислення параметрів моделі.

Однак, метод k-NN має і деякі недоліки. Один з основних недоліків полягає в його обчислювальній складності, оскільки для кожного нового зразка потрібно обчислювати відстані до всіх точок навчального набору даних [24]. Це може бути проблематичним для великих наборів даних. Крім того, метод k-NN може бути чутливим до вибору значення k. Занадто мале значення k може призвести до шуму в передбаченнях, тоді як занадто велике значення k може призвести до врахування занадто багатьох точок, що можуть бути нерелевантними.

Для покращення ефективності методу k-NN можуть використовуватися різні підходи. Одним з таких підходів є використання структури даних, таких як KD-дерева або Ball-дерева, які дозволяють швидко знаходити k найближчих сусідів [25]. Іншим підходом є нормалізація даних перед застосуванням методу k-NN, що допомагає уникнути впливу масштабів різних ознак на результати.

Метод k-NN знаходить широке застосування у різних областях машинного навчання. Він використовується для класифікації текстів, розпізнавання образів, медичної діагностики, систем рекомендацій та багатьох інших задач. Його здатність адаптуватися до різних типів даних та простота реалізації роблять його популярним вибором для багатьох дослідників і практиків у сфері машинного навчання.

Таким чином, метод k-NN є потужним і гнучким інструментом для класифікації та регресії, що дозволяє ефективно використовувати інформацію про сусідні точки у просторі ознак для передбачення класів або значень. Завдяки своїм перевагам і широким можливостям застосування, метод k-NN залишається одним з ключових інструментів у сучасних системах машинного навчання.

2.2.2 Методи на основі кластеризації

Методи на основі кластеризації є важливою частиною машинного навчання, що використовуються для виявлення природних груп або кластерів у наборі даних. Ці методи не потребують попередньо мічених даних і відносяться до методів неконтрольованого навчання. Кластеризація знаходить застосування в багатьох галузях, включаючи обробку зображень, аналіз текстів, біоінформатику та маркетинг. Найпопулярнішими методами кластеризації є алгоритм k-means, ієрархічна кластеризація та метод DBSCAN.

Алгоритм k-means є одним з найпоширеніших і простих методів кластеризації. Він базується на розбитті набору даних на k кластерів, де k є параметром, що задається користувачем. Алгоритм працює в кілька етапів. Спочатку вибираються k початкових центрів кластерів (центроїдів). Потім кожна точка даних призначається до найближчого центроїда на основі відстані (найчастіше використовується евклідова відстань). Після цього центроїди оновлюються як середнє значення всіх точок, що належать до

кожного кластеру. Ці кроки повторюються до тих пір, поки центроїди не перестануть змінюватися або не буде досягнута задана кількість ітерацій.

Ієрархічна кластеризація є іншим методом, який будує ієрархію кластерів у вигляді дерева, відомого як дендрограма. Існують два основні підходи до ієрархічної кластеризації: агломеративна (знизу вгору) і дивізивна (зверху вниз). Агломеративна кластеризація починається з того, що кожна точка даних розглядається як окремий кластер, і кластери поступово об'єднуються на основі схожості. Навпаки, дивізивна кластеризація починається з одного великого кластеру, який поступово розділяється на менші кластери. Ієрархічна кластеризація не вимагає попереднього визначення кількості кластерів, що є однією з її головних переваг. Результати кластеризації можуть бути представлені у вигляді дендрограми, яка візуально показує, як об'єднуються або розділяються кластери на різних рівнях схожості.

Метод DBSCAN (Density-Based Spatial Clustering of Applications with Noise) є ще одним популярним методом кластеризації, який базується на щільності. DBSCAN визначає кластери як області з високою щільністю точок, відокремлені областями з низькою щільністю. Цей метод має дві основні переваги: він може виявляти кластери довільної форми і автоматично обробляє шуми. Алгоритм DBSCAN працює наступним чином: для кожної точки даних обчислюється кількість сусідніх точок в межах заданого радіусу (eps). Якщо ця кількість перевищує заданий поріг (minPts), точка вважається «ядровою точкою» і разом зі своїми сусідами утворює кластер. Процес повторюється для всіх сусідів, розширюючи кластер, поки всі ядрові точки не будуть включені. Точки, які не належать до жодного кластеру, вважаються шумом.

Переваги методів кластеризації включають їх здатність виявляти природні групи у даних без необхідності мічених зразків. Це дозволяє використовувати кластеризацію у багатьох реальних задачах, де мічені дані

можуть бути недоступними або дорогими для отримання. Крім того, методи кластеризації є відносно простими у реалізації та інтуїтивно зрозумілими.

Один з основних недоліків полягає в чутливості до вибору параметрів, таких як кількість кластерів у k-means або радіус в DBSCAN. Невірно обрані параметри можуть призвести до поганих результатів кластеризації. Крім того, методи кластеризації можуть бути чутливими до шуму та аномалій у даних, що може вплинути на точність результатів. Також методи кластеризації можуть мати високу обчислювальну складність для великих наборів даних.

Незважаючи на ці обмеження, методи кластеризації залишаються важливими інструментами у сфері машинного навчання. Вони дозволяють виявляти приховані структури у даних і знаходити природні групи, що робить їх незамінними у багатьох додатках. Завдяки постійному розвитку та вдосконаленню алгоритмів, методи кластеризації продовжують залишатися важливим компонентом сучасних систем машинного навчання.

2.3 Методи глибокого навчання

2.3.1 Згорткові нейронні мережі (CNN)

2.3.1.1 AlexNet

AlexNet є однією з найвідоміших архітектур згорткових нейронних мереж (CNN), яка здійснила прорив у сфері комп'ютерного зору та глибокого навчання. Вона була розроблена Алексом Крижевським, Іллею Суцкевером та Джеффрі Хінтоном і виграла змагання ImageNet Large Scale Visual Recognition Challenge (ILSVRC) у 2012 році, значно випередивши інші моделі за точністю.

AlexNet складається з восьми шарів: п'ять згорткових шарів (conv layers) та три повнозв'язні шари (fully connected layers). Її архітектура є

складнішою порівняно з попередніми моделями, що дозволяє краще захоплювати складні патерни та ознаки зображень [26].

Розглянемо архітектуру AlexNet.

Вхідний шар. AlexNet приймає зображення розміром 227x227 пікселів з трьома кольоровими каналами (RGB).

Згорткові шари. Перший згортковий шар застосовує 96 фільтрів розміром 11x11 з кроком 4. Цей шар використовує функцію активації ReLU (Rectified Linear Unit) для введення нелінійності.

Другий згортковий шар застосовує 256 фільтрів розміром 5x5. Цей шар також використовує ReLU і включає механізм нормалізації (Local Response Normalization), що допомагає зменшити перенавчання.

Третій, четвертий та п'ятий згорткові шари застосовують 384, 384 і 256 фільтрів відповідно, всі з розміром 3x3. Ці шари також використовують ReLU і функціонують як основні елементи для виділення ознак зображення.

Шари підвибірки. Після першого та другого згорткових шарів застосовуються шари підвибірки розміром 3x3 з кроком 2. Ці шари допомагають зменшити розмір просторових ознак, зменшуючи обчислювальну складність і контролюючи перенавчання [27].

Повнозв'язні шари. Після згорткових шарів AlexNet має три повнозв'язні шари. Перші два повнозв'язні шари містять по 4096 нейронів кожен, а третій повнозв'язний шар містить 1000 нейронів, що відповідає кількості класів у задачі класифікації ImageNet.

Вихідний шар. Останній шар використовує функцію softmax для передбачення ймовірностей приналежності зображення до кожного з 1000 класів.

Модель AlexNet має кілька важливих особливостей, які сприяли її успіху в задачах класифікації зображень. Однією з ключових особливостей є використання функції активації ReLU. Ця функція сприяє швидшому навчанню порівняно з традиційними функціями активації, такими як

сигмоїдна чи tanh. ReLU вводить нелінійність у модель, що дозволяє краще моделювати складні дані.

Ще однією важливою особливістю є нормалізація (Local Response Normalization), яка допомагає стабілізувати навчання і зменшити перенавчання. Нормалізація сприяє тому, що великі активації в одному фільтрі інгібують активації в сусідніх фільтрах. Це допомагає контролювати активації та робить навчання більш стійким.

Шари підвибірки (Max-pooling) є ще одним важливим компонентом AlexNet. Підвибірка допомагає зменшити розмір просторових ознак і контролювати перенавчання, одночасно зберігаючи важливу інформацію про зображення. Це дозволяє моделі бути менш чутливою до незначних змін у положенні ознак на зображенні.

Щоб запобігти перенавчанню, AlexNet використовує техніку dropout у повнозв'язних шарах. Dropout випадковим чином вимикає частину нейронів під час навчання, що змушує мережу навчатися більш стійких ознак. Це значно покращує здатність моделі до узагальнення і зменшує ризик перенавчання.

Нарешті, AlexNet була однією з перших моделей, яка використовувала кілька графічних процесорів (GPU) для паралельного навчання. Використання кількох GPU дозволило значно пришвидшити процес навчання великої моделі на великому наборі даних ImageNet, що було ключовим фактором її успіху в конкурсі ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012 року.

Модель AlexNet та її похідні широко використовуються в різних додатках. Одним з основних застосувань є класифікація зображень у великих наборах даних, таких як ImageNet. AlexNet ефективно класифікує зображення, що дозволяє автоматизувати процеси обробки та аналізу великих обсягів візуальної інформації.

У сфері медичних зображень AlexNet застосовується для аналізу рентгенівських, МРТ та КТ знімків, допомагаючи виявляти та

класифікувати патології. Це значно покращує точність діагностики і прискорює процес обробки медичних даних.

Розпізнавання облич є ще однією важливою областю застосування AlexNet [28]. Модель використовується для ідентифікації та верифікації осіб у системах безпеки та соціальних мережах, забезпечуючи високу точність і надійність.

AlexNet також знаходить застосування в аналізі відео, зокрема в задачах розпізнавання дій та подій у відеоматеріалах. Це дозволяє автоматизувати моніторинг та аналіз відео у різних контекстах, від безпеки до розваг.

Важливим напрямком використання AlexNet є автономні транспортні засоби. Модель застосовується в системах розпізнавання об'єктів та навігації, що допомагає автономним транспортним засобам орієнтуватися у навколишньому середовищі і приймати рішення в реальному часі.

AlexNet залишається ключовою моделлю в історії глибокого навчання, яка показала потенціал згорткових нейронних мереж у вирішенні складних задач розпізнавання образів. Її успіх став каталізатором для подальших досліджень і розробок у цій галузі, сприяючи широкому впровадженню методів глибокого навчання у різних додатках.

2.3.1.2 VGG

Архітектура VGG (Visual Geometry Group) є однією з найвідоміших і впливових згорткових нейронних мереж, розроблених для задач класифікації зображень. Вона була створена дослідниками з Оксфордського університету (Сімоньян і Зіссерман) і представлена у праці «Very Deep Convolutional Networks for Large-Scale Image Recognition» у 2014 році. Модель VGG стала відомою завдяки своїй простоті і ефективності, продемонструвавши високі результати у змаганні ImageNet Large Scale Visual Recognition Challenge (ILSVRC).

Основною ідеєю архітектури VGG є використання невеликих (3x3) згорткових фільтрів, що дозволяє створювати глибокі мережі з великою кількістю шарів, зберігаючи при цьому контрольовану кількість параметрів. VGG використовує послідовності таких фільтрів для досягнення високої точності розпізнавання образів. Існує кілька варіантів архітектури VGG, серед яких найпопулярнішими є VGG16 і VGG19, що містять 16 і 19 шарів відповідно.

Вхідний шар. VGG приймає зображення розміром 224x224 пікселів з трьома кольоровими каналами (RGB).

Згорткові шари. Всі згорткові шари використовують фільтри розміром 3x3 з кроком 1 і padding 1 (додавання нулів по краях зображення). Це забезпечує збереження просторової розмірності зображення після кожної згортки.

Шари підвибірки (max-pooling). Після кожної кількох згорткових шарів застосовуються шари підвибірки розміром 2x2 з кроком 2. Це допомагає зменшити розмір просторових ознак, зменшуючи обчислювальну складність і контролюючи перенавчання.

Повнозв'язні шари. Після останнього згорткового шару йдуть три повнозв'язні шари [29]. Перші два повнозв'язні шари містять по 4096 нейронів кожен, а третій повнозв'язний шар містить 1000 нейронів, що відповідає кількості класів у задачі класифікації ImageNet.

Вихідний шар. Останній шар використовує функцію softmax для передбачення ймовірностей приналежності зображення до кожного з 1000 класів.

Модель VGG має кілька визначних характеристик, які забезпечують її високу ефективність та точність у задачах класифікації зображень. Однією з ключових особливостей є використання малих фільтрів розміром 3x3. Це дозволяє збільшити глибину мережі, зберігаючи при цьому контрольовану кількість параметрів, що робить модель більш потужною для захоплення складних патернів та ознак зображень.

Глибока архітектура моделей VGG16 і VGG19 є ще однією важливою особливістю. Ці моделі мають значно більшу глибину порівняно з попередніми моделями, такими як AlexNet. Завдяки великій глибині мережі модель здатна краще захоплювати ієрархічні ознаки зображень, що призводить до покращення точності класифікації.

Однорідність архітектури є ще однією перевагою VGG. Всі згорткові шари використовують фільтри однакового розміру (3x3), що спрощує реалізацію та оптимізацію моделі. Така однорідність сприяє стабільності навчання і покращує здатність моделі узагальнювати на нових даних.

Шари підвибірки, які застосовуються у VGG, ефективно зменшують розмір просторових ознак, зберігаючи при цьому важливу інформацію про зображення. Це допомагає знизити обчислювальну складність і контролювати перенавчання, що є важливим при роботі з великими наборами даних.

Крім того, всі шари у VGG використовують функцію активації ReLU (Rectified Linear Unit), яка сприяє швидшому навчанню і покращує продуктивність моделі. Використання ReLU вводить нелінійність у модель, що дозволяє краще моделювати складні дані та забезпечує більш ефективне навчання.

Модель VGG та її похідні знайшли широке застосування у різних задачах комп'ютерного зору та глибокого навчання. Одним з основних застосувань є класифікація зображень у великих наборах даних, таких як ImageNet [30]. Модель VGG забезпечує високу точність і надійність у багатьох задачах класифікації. Це дозволяє ефективно класифікувати зображення, що є важливим для автоматизації обробки та аналізу великих обсягів візуальної інформації.

У сфері обробки медичних зображень VGG застосовується для аналізу рентгенівських, МРТ та КТ знімків. Модель допомагає виявляти та класифікувати патології, що значно покращує точність діагностики і прискорює процес обробки медичних даних.

Розпізнавання облич є ще однією важливою областю застосування VGG. Модель використовується для ідентифікації та верифікації осіб у системах безпеки та соціальних мережах. Висока точність розпізнавання облич робить VGG ефективним інструментом для забезпечення безпеки та персоналізації послуг.

VGG також знаходить застосування в аналізі відео. Модель використовується у задачах розпізнавання дій та подій у відеоматеріалах. Це дозволяє автоматизувати моніторинг та аналіз відео у різних контекстах, від безпеки до розваг.

Ще одним важливим напрямком використання VGG є автономні транспортні засоби. Модель застосовується у системах розпізнавання об'єктів та навігації, що допомагає автономним транспортним засобам орієнтуватися у навколишньому середовищі і приймати рішення в реальному часі.

Модель VGG залишається важливим інструментом у сучасних дослідженнях і розробках у галузі глибокого навчання та комп'ютерного зору. Її архітектура і принципи побудови стали основою для багатьох нових моделей, що сприяють подальшому прогресу у вирішенні складних задач розпізнавання образів. Завдяки своїм особливостям і високій ефективності, VGG продовжує залишатися однією з найпопулярніших і найвпливовіших моделей у галузі глибокого навчання.

2.3.1.3 ResNet

ResNet (Residual Network) є однією з найвідоміших і найвпливовіших архітектур згорткових нейронних мереж, розроблених для глибокого навчання. Вона була представлена командою дослідників з Microsoft Research у праці «Deep Residual Learning for Image Recognition» у 2015 році. ResNet виграла кілька змагань, включаючи ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2015, завдяки своїй інноваційній

архітектурі, яка дозволяє тренувати дуже глибокі мережі без проблем, пов'язаних з затуханням градієнтів.

Основна ідея ResNet полягає у введенні залишкових блоків (residual blocks), які дозволяють обійти проблему затухання градієнтів, що часто виникає при тренуванні дуже глибоких нейронних мереж. Залишковий блок складається з «швидкої» або «прямої» лінії з'єднання, що пропускає шар або кілька шарів, і «залишкового» шляху, що складається з одного або більше згорткових шарів. Залишковий блок може бути представлений наступним чином:

$$y = F(x, \{Wi\}) + x, \tag{2.13}$$

де x – вхідний сигнал;

y – вихідний сигнал;

F(x, {Wi}) – залишкова функція, яка представляє собою послідовність згорткових шарів;

{Wi} – набір вагових коефіцієнтів цих шарів.

Цей прямий шлях (skip connection) дозволяє сигналу безпосередньо проходити через мережу, що значно полегшує тренування дуже глибоких моделей, оскільки градієнти можуть безпосередньо передаватися назад через мережу.

Основні особливості архітектури ResNet включають залишкові блоки, які є основним будівельним блоком мережі і включають прямі шляхи для передачі сигналу, що допомагає уникнути проблеми затухання градієнтів і сприяє ефективному навчанню дуже глибоких мереж. Висока глибина мережі дозволяє створювати надзвичайно глибокі моделі, такі як ResNet-50, ResNet-101 і ResNet-152, де число вказує на кількість шарів у мережі. Це забезпечує високу здатність до узагальнення і точність у задачах класифікації зображень. Згорткові шари кожного залишкового блоку зазвичай складаються з двох або трьох згорткових шарів з невеликими

фільтрами (3x3), що забезпечує ефективне виділення ознак зображення. ResNet використовує шари підвибірки (Max-Pooling) для зменшення розміру просторових ознак, зменшуючи обчислювальну складність і контролюючи перенавчання. Всі шари використовують функцію активації ReLU (Rectified Linear Unit), яка сприяє швидшому навчанню і покращує продуктивність моделі.

ResNet здійснила значний вплив на розвиток глибокого навчання і комп'ютерного зору. Вона продемонструвала, що тренування дуже глибоких мереж може бути ефективним завдяки залишковим блокам. ResNet відкрила шлях для подальших досліджень і розробок у галузі глибокого навчання, зокрема, таких архітектур, як DenseNet і Inception-ResNet, які продовжили розвивати ідеї залишкових з'єднань.

ResNet та її похідні моделі знайшли широке застосування у різних задачах комп'ютерного зору та глибокого навчання. Використання ResNet для класифікації зображень у великих наборах даних, таких як ImageNet, забезпечує високу точність і надійність у багатьох задачах класифікації. ResNet також застосовується для аналізу медичних зображень, включаючи виявлення та класифікацію патологій на рентгенівських, МРТ та КТ знімках. Використання ResNet для ідентифікації та верифікації осіб у системах безпеки та соціальних мережах, а також у задачах розпізнавання дій та подій у відеоматеріалах і системах розпізнавання об'єктів та навігації для автономних транспортних засобів є прикладами її широкого застосування.

Ця модель має кілька важливих переваг. По-перше, її залишкові блоки дозволяють ефективно тренувати дуже глибокі моделі, що значно покращує точність та узагальнення. По-друге, архітектура ResNet є гнучкою і може бути легко адаптована для різних задач комп'ютерного зору. Однак ResNet має і деякі недоліки. Її висока глибина може призводити до значних обчислювальних витрат і потреби у великій кількості пам'яті. Крім того, налаштування гіперпараметрів і оптимізація тренування глибоких мереж можуть бути складними і вимагати великого досвіду.

Таким чином модель ResNet є важливою архітектурою у сфері глибокого навчання, яка продемонструвала ефективність залишкових блоків для тренування дуже глибоких нейронних мереж. Її успіх у задачах класифікації зображень і здатність до узагальнення зробили її основою для багатьох подальших досліджень і розробок. Завдяки своїм особливостям і високій ефективності, ResNet залишається одним з ключових інструментів у сучасних системах глибокого навчання і комп'ютерного зору.

2.3.1.4 Inception

Inception є однією з найвідоміших архітектур згорткових нейронних мереж, розроблених для глибокого навчання. Вона була представлена командою Google у праці «Going Deeper with Convolutions» у 2014 році і відома також під назвою GoogLeNet. Основною метою Inception було збільшення глибини та ширини мережі, зменшуючи при цьому обчислювальні витрати. Архітектура Inception стала революційною завдяки використанню спеціальних блоків, які дозволяють ефективно захоплювати різноманітні ознаки зображень.

Архітектура Inception базується на використанні Inception-блоків, які комбінують згортки з різними розмірами фільтрів у одному шарі. Це дозволяє мережі одночасно захоплювати інформацію на різних масштабах і з різною роздільною здатністю. Ключова ідея Inception-блоків полягає у паралельному застосуванні згорток з фільтрами 1x1, 3x3, 5x5 та шару підвибірки (pooling), що дозволяє мережі бути більш гнучкою і ефективною.

Основні елементи Inception-блоків включають:

Згортка 1x1. Цей тип згортки використовується для зменшення розмірності простору ознак, що зменшує обчислювальні витрати та зберігає важливу інформацію. Фільтри 1x1 виконують лінійну комбінацію вхідних каналів, дозволяючи моделі навчатися нелінійним комбінаціям ознак.

Згортка 3x3 та 5x5. Ці згортки використовуються для захоплення просторових ознак на різних масштабах. Фільтри 3x3 захоплюють локальні патерни, тоді як фільтри 5x5 забезпечують ширше охоплення.

Шар підвибірки (Pooling). Шар підвибірки допомагає зменшити розмір просторових ознак, зберігаючи при цьому найважливішу інформацію. Зазвичай використовується max-pooling для захоплення найбільш значущих ознак у певній області.

Паралельні шляхи. Inception-блоки використовують паралельні шляхи, які комбінують результати згорток з різними фільтрами та підвибірки. Це дозволяє мережі одночасно захоплювати різноманітні ознаки на різних масштабах.

Конкатенація. Результати всіх паралельних шляхів об'єднуються шляхом конкатенації вздовж канального виміру, що створює вихід Inception-блоку.

Архітектура Inception складається з кількох послідовних Inception-блоків, що дозволяє мережі навчатися складним ієрархічним ознакам зображень. GoogLeNet (Inception v1) складається з 22 шарів, включаючи кілька Inception-блоків, що забезпечують високу точність класифікації зображень при відносно низьких обчислювальних витратах. Подальші версії архітектури Inception, такі як Inception v2, v3, v4 та Inception-ResNet, включають покращення, спрямовані на підвищення ефективності та точності.

Однією з ключових інновацій Inception v2 та v3 є використання факторизованих згорток, які розділяють згортки великих розмірів на послідовність менших згорток, таких як 3x3 та 5x5. Це дозволяє зменшити кількість параметрів та обчислювальну складність, зберігаючи при цьому високу здатність до захоплення ознак. Наприклад, згортка 5x5 може бути розділена на дві згортки 3x3, що значно зменшує кількість параметрів і обчислень.

Іншим важливим покращенням у Inception v3 є використання допоміжного класифікатора, який додається до середини мережі і допомагає боротися з проблемою затухання градієнтів, забезпечуючи додатковий сигнал для зворотного поширення помилки під час навчання. Це допомагає стабілізувати навчання і покращити продуктивність мережі.

Inception v4 і Inception-ResNet є подальшими покращеннями архітектури Inception, які комбінують ідеї Inception-блоків з залишковими блоками (ResNet). Це дозволяє ще краще боротися з проблемою затухання градієнтів і покращити здатність до навчання дуже глибоких мереж.

Архітектура Inception знайшла широке застосування у різних задачах комп'ютерного зору та глибокого навчання. Використання Inception для класифікації зображень у великих наборах даних, таких як ImageNet, забезпечує високу точність і ефективність при відносно низьких обчислювальних витратах. Inception також застосовується для аналізу медичних зображень, включаючи виявлення та класифікацію патологій на рентгенівських, МРТ та КТ знімках. Використання Inception у задачах розпізнавання облич, дій та подій у відеоматеріалах, а також у системах розпізнавання об'єктів та навігації для автономних транспортних засобів є прикладами її широкого застосування.

Основні переваги архітектури Inception включають її здатність ефективно захоплювати різноманітні ознаки на різних масштабах і з різною роздільною здатністю завдяки використанню паралельних шляхів у Inception-блоках. Крім того, факторизація згорток і використання ауксильярних класифікаторів допомагають зменшити обчислювальну складність і покращити продуктивність мережі. Однак, архітектура Inception також має деякі недоліки, такі як висока складність реалізації і налаштування моделі, а також значні обчислювальні витрати для дуже глибоких версій.

Загалом, Inception є важливою архітектурою у сфері глибокого навчання, яка продемонструвала ефективність використання паралельних

шляхів і факторизованих згорток для тренування глибоких нейронних мереж. Її успіх у задачах класифікації зображень і здатність до узагальнення зробили її основою для багатьох подальших досліджень і розробок. Завдяки своїм особливостям і високій ефективності, Inception залишається одним з ключових інструментів у сучасних системах глибокого навчання і комп'ютерного зору.

2.3.2 Генеративно-змагальні мережі (GANs)

2.3.2.1 DCGAN

Генеративно-змагальні мережі (Generative Adversarial Networks, GANs) є однією з найцікавіших і найпотужніших архітектур у сфері глибокого навчання, які використовуються для генерації нових даних, що виглядають схожими на дані навчального набору. GANs були запропоновані Ієном Гудфеллоу та його колегами у 2014 році. Архітектура GAN складається з двох нейронних мереж: генератора і дискримінатора, які змагаються між собою. Генератор намагається створити реалістичні дані, тоді як дискримінатор намагається відрізнити справжні дані від підроблених.

DCGAN (Deep Convolutional Generative Adversarial Networks) є однією з найбільш відомих і успішних реалізацій GAN, що використовує згорткові нейронні мережі для обох компонентів – генератора і дискримінатора. DCGAN була представлена Радфордом, Метц і Чинном у 2015 році і стала популярною завдяки своїй здатності генерувати високоякісні зображення.

Архітектура DCGAN має кілька ключових особливостей, які відрізняють її від класичних GAN. Генератор DCGAN використовує послідовність транспонованих згорткових шарів (transposed convolutions), також відомих як деконволюційні шари, для поступового збільшення розміру зображення з вхідного випадкового шуму. Дискримінатор DCGAN

використовує класичні згорткові шари для обробки вхідного зображення і визначення, чи є воно справжнім або підробленим.

Основні компоненти архітектури DCGAN включають:

Генератор. Генератор приймає випадковий шум як вхід і перетворює його на зображення шляхом послідовного застосування транспонованих згорткових шарів. Кожен шар включає згортку, нормалізацію пакетів (batch normalization) і функцію активації ReLU (за винятком останнього шару, який використовує функцію активації Tanh). Цей підхід дозволяє генератору створювати деталізовані і реалістичні зображення.

Дискримінатор. Дискримінатор приймає зображення як вхід і класифікує його як справжнє або підроблене. Він використовує класичні згорткові шари, кожен з яких включає згортку, нормалізацію пакетів і функцію активації Leaky ReLU. Останній шар використовує сигмоїдну функцію активації для виведення ймовірності того, що зображення є справжнім.

Процес тренування DCGAN включає одночасне тренування генератора і дискримінатора. Генератор намагається створювати зображення, які дискримінатор не може відрізнити від справжніх. Дискримінатор, у свою чергу, намагається покращити свою здатність розпізнавати підроблені зображення. Цей змагальний процес призводить до того, що обидві мережі стають сильнішими, і генератор починає створювати все більш реалістичні зображення.

Однією з ключових особливостей DCGAN є використання згорткових шарів у генераторі та дискримінаторі, що дозволяє моделі ефективно захоплювати просторові структури зображень. Використання транспонованих згорткових шарів у генераторі дозволяє поступово збільшувати розмір зображення, зберігаючи деталізацію. Нормалізація пакетів допомагає стабілізувати процес навчання і покращує якість згенерованих зображень.

DCGAN знайшла широке застосування у різних галузях, включаючи генерацію зображень, підвищення роздільної здатності зображень, створення анімованих персонажів та багато інших. У медицині DCGAN використовується для генерації медичних зображень, що можуть бути використані для навчання інших моделей або для виявлення патологій. У сфері розваг DCGAN застосовується для створення реалістичних персонажів та сцен у відеоіграх та фільмах.

Однією з головних переваг DCGAN є її здатність створювати високоякісні зображення з невеликої кількості вхідного шуму. Крім того, використання згорткових шарів дозволяє моделі ефективно захоплювати просторові структури, що робить її більш придатною для роботи із зображеннями. Однак, DCGAN також має деякі недоліки. Тренування GAN моделей, включаючи DCGAN, може бути нестабільним і вимагати ретельного налаштування гіперпараметрів. Крім того, DCGAN може бути чутливим до зміни архітектури та вибору функцій активації.

Загалом, DCGAN є важливою архітектурою у сфері глибокого навчання, яка продемонструвала ефективність використання згорткових нейронних мереж для генерації реалістичних зображень. Її успіх відкрив нові можливості для досліджень і застосувань у різних галузях, роблячи її одним з ключових інструментів у сучасних системах генеративного моделювання.

2.3.3 Трансформери

2.3.3.1 Vision Transformer (ViT)

Vision Transformer (ViT) є новаторською архітектурою для задач комп'ютерного зору, яка використовує трансформери, спочатку розроблені для обробки природної мови. ViT був представлений дослідниками з Google Research у праці «An Image is Worth 16x16 Words: Transformers for Image

Recognition at Scale» у 2020 році. Основна ідея ViT полягає в тому, щоб застосувати трансформери для обробки зображень, представляючи їх як послідовність патчів (невеликих фрагментів зображення), аналогічно до обробки послідовностей слів у тексті.

Архітектура Vision Transformer складається з кількох ключових компонентів:

Розбиття зображення на патчі. Вхідне зображення розміром H x W x C (де H – висота, W – ширина, C – кількість каналів) розбивається на невеликі непересічні патчі розміром P x P. Кожен патч перетворюється на вектор шляхом «розгортання» його пікселів у лінійну послідовність. У результаті отримується матриця розміром (N,P2C), де N = H x Wp2 – кількість патчів.

Лінійна проекція патчів. Кожен патч перетворюється на вектор фіксованої довжини за допомогою лінійної проекції. Це створює послідовність векторів, яку можна подавати на вхід трансформеру.

Додавання позиційних ембедингів. Оскільки трансформери не мають вбудованого уявлення про порядок елементів у послідовності, до кожного патчу додається позиційний ембединг, що кодує його положення у початковому зображенні. Це дозволяє моделі враховувати просторову інформацію.

Трансформер. Послідовність патчів разом з позиційними ембедингами подається на вхід трансформеру, який складається з кількох шарів самопильності (self-attention) і позиційних закодованих шарів (feed-forward layers). Ці шари обробляють послідовність патчів, враховуючи взаємодію між ними.

Класифікаційна голова. Після обробки трансформером отримується векторне представлення всього зображення, яке подається на вхід класифікаційної голови. Зазвичай це один або кілька повнозв'язних шарів, які генерують остаточний вихідний клас.

Особливості Vision Transformer включають використання трансформерів для обробки зображень, що дозволяє моделі захоплювати глобальні контексти і взаємодії між різними частинами зображення. Це відрізняє ViT від класичних згорткових нейронних мереж (CNN), які зазвичай фокусуються на локальних ознаках.

Однією з ключових переваг ViT є його здатність до масштабування. ViT може ефективно навчатися на великих наборах даних, таких як ImageNet, і демонструвати високу точність у задачах класифікації зображень. Використання трансформерів дозволяє моделі краще враховувати взаємодії між різними частинами зображення, що покращує її здатність до узагальнення.

ViT також має деякі недоліки. Наприклад, для досягнення високої продуктивності модель потребує великих наборів даних і значних обчислювальних ресурсів. Крім того, початкові реалізації ViT часто потребують попередньо навчених моделей або довгого періоду навчання.

ViT знайшов широке застосування у різних задачах комп'ютерного зору. Використання ViT для класифікації зображень у великих наборах даних, таких як ImageNet, забезпечує високу точність і надійність. ViT також застосовується для сегментації зображень, де модель може розділити зображення на різні області або об'єкти, і для задач розпізнавання облич, дій та подій у відеоматеріалах. У медицині ViT використовується для аналізу медичних зображень, включаючи виявлення та класифікацію патологій на рентгенівських, МРТ та КТ знімках.

Основні переваги ViT включають здатність захоплювати глобальні контексти і взаємодії між різними частинами зображення, що покращує точність і здатність до узагальнення моделі. Однак, для досягнення високої продуктивності ViT потребує великих наборів даних і значних обчислювальних ресурсів. Незважаючи на ці обмеження, ViT є важливим інструментом у сучасних системах глибокого навчання і комп'ютерного

зору, відкриваючи нові можливості для досліджень і застосувань у різних галузях.

2.4 Гібридні моделі

Гібридні моделі у сфері машинного навчання і комп'ютерного зору поєднують різні підходи і методи для досягнення кращої продуктивності та точності в різноманітних задачах. Гібридні моделі можуть комбінувати переваги різних архітектур, таких як згорткові нейронні мережі (CNN), рекурентні нейронні мережі (RNN), трансформери та інші методи машинного навчання. Метою створення гібридних моделей є використання сильних сторін кожного підходу для покращення загальної продуктивності та здатності до узагальнення.

Гібридні моделі можуть бути корисними у багатьох застосуваннях, таких як класифікація зображень, сегментація, розпізнавання облич, аналіз текстів, обробка природної мови та багато інших. Далі розглянемо деякі популярні гібридні моделі та їх застосування.

Одним з прикладів гібридних моделей є комбінація згорткових нейронних мереж (CNN) і рекурентних нейронних мереж (RNN). CNN відмінно підходять для обробки даних з фіксованою структурою, таких як зображення, тоді як RNN добре працюють з послідовними даними, такими як текст або часові ряди. Комбінація CNN і RNN може бути ефективною для задач, які вимагають обробки як просторової, так і тимчасової інформації. Наприклад, така комбінація використовується в задачах відеоаналізу, де CNN обробляє окремі кадри, а RNN аналізує послідовність кадрів для розпізнавання дій або подій у відео.

Іншим прикладом гібридних моделей є поєднання CNN і трансформерів. CNN добре підходять для виділення локальних ознак зображень, тоді як трансформери можуть ефективно захоплювати глобальні контексти і взаємодії між різними частинами зображення. Комбінація цих

двох підходів може бути корисною для задач, де важливо враховувати як локальні, так і глобальні ознаки. Наприклад, у задачах сегментації зображень CNN можуть виділяти деталі на рівні пікселів, тоді як трансформери забезпечують глобальне розуміння сцени.

Гібридні моделі також можуть включати комбінацію генеративних моделей, таких як генеративно-змагальні мережі (GANs), з дискримінативними моделями, такими як CNN. Наприклад, GANs можуть генерувати високоякісні зображення, а CNN можуть бути використані для їх подальшої класифікації або аналізу. Це може бути корисним у задачах, де важливо не лише генерувати реалістичні зображення, але і аналізувати їх для виявлення певних ознак або аномалій.

Іншим напрямом гібридних моделей є поєднання класичних методів машинного навчання з глибокими нейронними мережами. Наприклад, можна використовувати методи кластеризації, такі як k-means, для попередньої обробки даних і виділення груп схожих зразків, а потім застосовувати CNN для детального аналізу кожної групи. Це може допомогти покращити продуктивність моделі за рахунок використання сильних сторін кожного підходу.

Гібридні моделі знаходять широке застосування у багатьох галузях. У медицині вони використовуються для аналізу складних медичних зображень, включаючи виявлення і класифікацію патологій на рентгенівських, МРТ та КТ знімках. У сфері розваг гібридні моделі застосовуються для створення реалістичних персонажів та сцен у відеоіграх та фільмах. У фінансовій сфері вони використовуються для аналізу фінансових ринків і прогнозування цінових змін.

Основні переваги гібридних моделей включають їх здатність ефективно комбінувати різні підходи для досягнення кращих результатів. Це дозволяє моделі бути більш гнучкою і адаптивною до різних типів даних і задач. Крім того, гібридні моделі можуть використовувати сильні сторони

кожного підходу для покращення загальної продуктивності та здатності до узагальнення.

Однак, їхня складність може збільшити обчислювальні витрати і вимагати більше часу для тренування. Крім того, налаштування гіперпараметрів і оптимізація таких моделей можуть бути складними і вимагати великого досвіду.

Загалом, гібридні моделі є важливим інструментом у сучасних системах машинного навчання і комп'ютерного зору. Вони дозволяють комбінувати різні підходи для досягнення кращих результатів і відкривають нові можливості для досліджень і застосувань у різних галузях. Завдяки своїм особливостям і високій ефективності, гібридні моделі продовжують залишатися ключовим напрямом розвитку у сфері глибокого навчання.

3 ПРОВЕДЕННЯ ЕКСПЕРИМЕНТІВ

3.1 Обґрунтування використаних технологій

У цьому розділі розглянемо обґрунтування вибору технологій та інструментів, які були використані для реалізації та оцінки моделей і алгоритмів пошуку зображень у сховищах великих даних. Ми зосередимося на таких аспектах, як вибір набору даних, використання бібліотек для машинного навчання та глибокого навчання, методи попередньої обробки даних, оптимізації моделей та реалізації пошукових систем.

Для реалізації та оцінки моделей і алгоритмів пошуку зображень були використані такі бібліотеки Python: scikit-learn, TensorFlow, Keras та OpenCV. Кожна з цих бібліотек має свої унікальні переваги та була обрана з наступних причин. Scikit-learn забезпечує простий та інтуїтивно зрозумілий інтерфейс для реалізації та оцінки алгоритмів машинного навчання. Бібліотека включає широкий спектр алгоритмів для класифікації, регресії, кластеризації та зменшення розмірності, а також набір вбудованих функцій для оцінки моделей, що дозволяє легко отримувати метрики, такі як точність, precision, recall та F1-міра. TensorFlow забезпечує масштабованість та високу продуктивність, що дозволяє тренувати великі нейронні мережі на великих наборах даних. Бібліотека дозволяє створювати складні архітектури нейронних мереж з використанням низькорівневих API, а також підтримує використання GPU для прискорення обчислень, що значно зменшує час навчання моделей. Keras забезпечує високий рівень абстракції, що дозволяє швидко створювати та тренувати моделі нейронних мереж. Вона інтегрується з TensorFlow, що дозволяє використовувати всі переваги TensorFlow з простим інтерфейсом Keras, і включає багато вбудованих інструментів для попередньої обробки даних, підвищення даних (data augmentation) та оцінки моделей. OpenCV забезпечує широкий набір функцій для обробки зображень, включаючи фільтри, виявлення країв,

трансформації та аналіз текстур. Бібліотека оптимізована для високої продуктивності та може працювати в реальному часі, що важливо для обробки великих наборів зображень.

Попередня обробка даних є важливим етапом у підготовці зображень для пошукових алгоритмів. У нашому дослідженні були використані такі методи попередньої обробки: виявлення країв, аналіз текстур та гістограма кольорів. Виявлення країв за допомогою алгоритму Кенні дозволяє виділити контури об'єктів на зображеннях, що може допомогти моделі зосередитися на важливих візуальних характеристиках. Використання виявлення країв як попередньої обробки у комбінації з іншими методами виділення ознак показало середні результати, але може бути покращено шляхом застосування більш складних методів виділення ознак. Використання перетворення Габора для виділення текстурних ознак дозволяє отримати додаткову інформацію про структуру зображень, що може бути корисним у комбінації з іншими методами для покращення загальної продуктивності пошукових алгоритмів. Аналіз розподілу кольорів у зображеннях за допомогою гістограми кольорів може допомогти у виявленні кольорових шаблонів, які можуть бути корисними для пошуку. Використання цього методу у поєднанні з іншими методами виділення ознак може покращити точність пошуку.

Для покращення продуктивності пошукових моделей були застосовані різні техніки оптимізації. Використання Dropout для випадкового відключення нейронів під час навчання допомагає запобігти перенавчанню та покращує узагальненість моделі. Застосування L2-регуляризації допомагає зменшити значення ваг у нейронній мережі, що також сприяє зменшенню перенавчання. Використання технік підвищення даних, таких як випадкові обертання, масштабування, зсуви та відображення зображень, допомагає збільшити різноманітність навчальних зразків, що дозволяє моделі краще узагальнювати та покращує її стійкість до різних варіацій у зображеннях. Підбір оптимальних значень

гіперпараметрів, таких як кількість шарів, розмір шарів, швидкість навчання, кількість епох та розмір батчу, дозволяє досягти кращої продуктивності моделі. Цей процес може бути автоматизований за допомогою методів пошуку, таких як Grid Search або Random Search. Використання GPU для тренування моделей дозволяє значно прискорити процес навчання, особливо для глибоких нейронних мереж. Це особливо важливо для великих наборів даних, де час навчання може бути критичним фактором.

Кожна з використаних моделей має свої особливості та була обрана з певних причин. k-NN є простим та зрозумілим методом пошуку зображень, який не потребує складного налаштування та добре працює на невеликих наборах даних. Цей метод був використаний для порівняння з більш складними моделями та для демонстрації його обмежень при роботі з великими наборами даних. k-means є популярним методом кластеризації, який дозволяє групувати зображення за подібністю ознак. Використання k-means було обґрунтовано необхідністю порівняння методів пошуку з методами кластеризації для визначення їх ефективності у різних сценаріях. ResNet є однією з найуспішніших архітектур глибоких нейронних мереж, яка використовує резидуальні блоки для подолання проблеми затухання градієнтів. Ця модель була обрана для демонстрації можливостей глибоких нейронних мереж у задачах пошуку зображень. Використання CNN у поєднанні з попередньою обробкою зображень за допомогою виявлення країв було обрано для дослідження впливу попередньої обробки на продуктивність моделей. Ця модель дозволяє вивчити, як різні методи виділення ознак можуть впливати на результати пошуку.

Використання набору даних CIFAR-10, бібліотек scikit-learn, TensorFlow, Keras та OpenCV, а також різних методів попередньої обробки та оптимізації моделей дозволило провести всебічне дослідження продуктивності різних моделей пошуку зображень у сховищах великих даних. Вибір цих технологій був обґрунтований їхніми перевагами,

доступністю та широким застосуванням у сучасних дослідженнях машинного та глибокого навчання. Кожна з використаних моделей показала свої сильні та слабкі сторони, що дозволило отримати цінну інформацію для подальших досліджень та покращення результатів пошуку зображень. Конкретні покращення можуть включати збільшення кількості сусідів для k-NN, використання GMM для кластеризації, застосування Dropout та L2-регуляризації для ResNet, та використання більш складних архітектур CNN. Ці зміни можуть бути імплементовані шляхом відповідних налаштувань параметрів та застосування нових методів у рамках існуючих бібліотек Python, таких як scikit-learn, TensorFlow та Keras.

3.2 Опис експериментів

3.2.1 Загальний опис

Експерименти проводилися на наборі даних CIFAR-10, який містить 60,000 кольорових зображень розміром 32x32 пікселів у 10 класах, з 6,000 зображеннями на клас. Набір даних розділений на тренувальний (50,000 зображень) і тестовий (10,000 зображень) піднабори. Для валідації використовувалося 20% тренувального набору. Проведення експериментів включало декілька етапів: підготовка даних, налаштування моделей, навчання моделей, оцінка продуктивності та порівняння результатів.

Для кожного методу було обрано специфічні гіперпараметри, які оптимізували продуктивність моделі. Для k-NN, число сусідів (k) було встановлено на 3. Цей вибір був обґрунтований тим, що менші значення k знижують складність і обчислювальні витрати, а також дозволяють більш ефективно враховувати локальну структуру даних, що є важливим для набору даних, де можуть бути значні відмінності між класами. Використовувалися різні набори ознак: гістограма кольорів, ознаки Габора

та виявлення країв, щоб з'ясувати, який тип ознак найкраще підходить для даного завдання класифікації.

Для k-means число кластерів було встановлено на 10, щоб відповідати кількості класів у наборі даних CIFAR-10. Використання цієї кількості кластерів дозволяє максимально ефективно співспоставити кластери з реальними класами. Для k-means також використовувалися різні набори ознак для порівняння продуктивності моделей при різних підходах до обробки зображень.

Для ResNet була використана базова модель ResNet50 з попередньо навченою вагою ImageNet. Це забезпечило високий початковий рівень продуктивності завдяки попередньому навчанню на великому і різноманітному наборі даних. Налаштування включали 10 епох, batch size: 32, optimizer: Adam, learning rate: 0.001. Ці параметри були обрані для забезпечення достатньої кількості ітерацій для навчання моделі без надмірного перевантаження, оптимізатор Adam був обраний за його здатність до швидкої і стабільної конвергенції, а learning rate: 0.001 забезпечував помірну швидкість навчання.

Для AlexNet використовувалися ті ж самі налаштування для забезпечення порівнянності результатів між різними архітектурами нейронних мереж. Модель Inception також використовувала 10 епох, batch size: 32, optimizer: Adam, learning rate: 0.001 з тих же причин, що і ResNet та AlexNet, забезпечуючи оптимальну продуктивність при помірних витратах ресурсів.

Для комбінованої моделі були використані об'єднані ознаки: гістограма кольорів, ознаки Габора, виявлення країв. Це дозволило отримати найбільш повну інформацію про зображення, об'єднуючи різні підходи до обробки візуальних даних. Налаштування також включали 10 епох, batch size: 32, optimizer: Adam, learning rate: 0.001, що забезпечило стабільне і ефективне навчання.

Перед навчанням моделей дані були підготовлені наступним чином: для гістограми кольорів кожне зображення було розділене на канали кольорів (червоний, зелений, синій), для кожного каналу було обчислено гістограму з 256 бінів, і гістограми були об'єднані в один вектор ознак. Цей підхід дозволяє зберігати інформацію про кольоровий розподіл у зображенні, що є корисним для розпізнавання об'єктів. Для ознак Габора зображення були перетворені у відтінки сірого, і для кожного зображення було обчислено реальну частину Габорового перетворення з частотою 0.6, результати перетворення були перетворені в один вектор ознак. Габорові ознаки дозволяють ефективно захоплювати текстурні характеристики зображення, що є важливим для виявлення складних візуальних патернів. Для виявлення країв зображення були перетворені у відтінки сірого, і для кожного зображення були виявлені краї за допомогою алгоритму Кенні, вектор країв був збережений як ознака. Виявлення країв дозволяє фокусуватися на контурах і формах об'єктів, що є корисним для класифікації на основі структурних характеристик.

Моделі k-NN та k-means були навчені на трьох різних наборах ознак: гістограма кольорів, ознаки Габора та виявлення країв. Це дозволило оцінити, які типи ознак є найбільш ефективними для класифікації зображень у даному наборі даних. Глибокі нейронні мережі (ResNet, AlexNet, Inception) були навчені безпосередньо на зображеннях, що дозволило їм автоматично навчатися оптимальних ознак з даних. Комбінована модель була навчена на об'єднаному наборі ознак, що дозволило об'єднати переваги різних підходів до обробки зображень.

Під час навчання моделей використовувалися такі налаштування: для k-NN вхідні ознаки були вектори ознак (гістограма кольорів, ознаки Габора, виявлення країв), навчання виконувалося за допомогою методу найближчих сусідів для класифікації. Для k-means вхідні ознаки були вектори ознак (гістограма кольорів, ознаки Габора, виявлення країв), навчання проводилося через кластеризацію даних у 10 кластерів та мапування

кластерів на реальні мітки. Для ResNet, AlexNet та Inception вхідні дані були зображення CIFAR-10, і навчання проводилося через оптимізацію моделі за допомогою Adam optimizer. Для комбінованої моделі вхідні ознаки були об'єднані вектори ознак (гістограма кольорів, ознаки Габора, виявлення країв), і навчання проводилося через використання глибокої нейронної мережі для класифікації об'єднаних ознак.

Оцінка продуктивності моделей проводилася за такими метриками: точність (Accuracy), точність (Precision), повнота (Recall), F1-Score, час навчання, час інференції. Точність визначається як частка правильно класифікованих зображень серед усіх класифікованих як цей клас, повнота – як частка правильно класифікованих зображень серед усіх зображень цього класу, F1-Score – як гармонійне середнє між точністю і повнотою. Час навчання – це час, витрачений на навчання моделі, а час інференції – це час, витрачений на класифікацію тестового набору даних.

3.2.2 Обгрунтування обраних моделей

У дослідженні було обрано комбінацію традиційних методів (k-NN, k-means) та сучасних методів глибокого навчання (ResNet, AlexNet, Inception) для класифікації зображень. Крім того, було запропоновано та впроваджено новий комбінований метод, який використовує різні типи ознак для покращення продуктивності.

Метод k-NN (k найближчих сусідів) було обрано завдяки його простій реалізації та відсутності потреби у складному навчанні. Він добре працює на невеликих наборах даних, але має високі обчислювальні витрати при великих наборах даних, чутливий до шуму та потребує зберігання всіх тренувальних даних для інференції. Метод k-means ефективний для кластеризації та може використовуватися для зменшення розмірності даних, але він чутливий до початкових центрів кластерів, може не знайти оптимальні кластери та не підходить для складних задач класифікації.

Модель ResNet було обрано за її глибоку архітектуру, яка дозволяє виявляти складні ознаки. Використання резидуальних блоків покращує навчання глибоких мереж і забезпечує високу точність класифікації. Однак, вона має високі обчислювальні витрати та потребує великих наборів даних для ефективного навчання. Модель AlexNet забезпечує високу продуктивність на великих наборах даних і ефективна для задач класифікації зображень, але вона також має високі обчислювальні витрати та менш ефективна у порівнянні з сучасними моделями. Модель Inception використовує різні розміри фільтрів в одному шарі, що дозволяє захоплювати різні масштаби ознак і забезпечує високу точність класифікації, але вона має високі обчислювальні витрати, складну архітектуру та потребує великих наборів даних.

Вибір методів та моделей у дослідженні був обґрунтований їх здатністю ефективно обробляти зображення та виявляти складні ознаки. Особливу увагу було приділено розробці та впровадженню нового комбінованого методу. Цей метод використовує різні типи ознак, такі як гістограма кольорів, ознаки Габора та виявлення країв, для покращення точності класифікації. Комбінована модель показала конкурентоспроможні результати, що свідчить про потенціал комбінування різних типів ознак для покращення продуктивності моделей.

Переваги k-NN включають простоту та відсутність потреби у навчанні, проте високі обчислювальні витрати на інференції та чутливість до шуму є суттєвими недоліками. k-means є ефективним для кластеризації та може зменшувати розмірність даних, але він чутливий до початкових центрів кластерів і не підходить для складних задач класифікації. ResNet, з її глибокою архітектурою та резидуальними блоками, забезпечує високу точність, але має високі обчислювальні витрати і потребує великих наборів даних. AlexNet показує високу продуктивність на великих наборах даних, але має високі обчислювальні витрати і є менш ефективною в порівнянні з сучасними моделями. Inception, використовуючи різні розміри фільтрів,

дозволяє захоплювати різні масштаби ознак і забезпечує високу точність, але має високі обчислювальні витрати, складну архітектуру і потребує великих наборів даних.

Комбінована модель, яка об'єднує різні типи ознак, показала потенціал для покращення точності класифікації за рахунок використання інформації з різних джерел. Цей підхід забезпечує гнучкість у використанні різних ознак, але потребує складної реалізації та має високу обчислювальну складність.

Гіперпараметри було обрано на основі літературних даних та експериментальних спроб для забезпечення оптимальної продуктивності моделей. Наприклад, число сусідів у k-NN встановлено на 3, що є оптимальним компромісом між обчислювальною складністю та точністю для даного набору даних. Число кластерів у k-means було встановлено на 10, що відповідає кількості класів у наборі даних CIFAR-10, забезпечуючи відповідну кластеризацію. Для моделей глибокого навчання, таких як ResNet, AlexNet та Inception, було обрано 10 епох навчання та batch size 32 для забезпечення достатньої кількості ітерацій без перенавчання. Використання Adam optimizer з learning rate 0.001 дозволяє швидко та ефективно оптимізувати ваги мережі, зменшуючи втрати і підвищуючи точність.

3.3 Результати експериментів

У цьому розділі представлено результати експериментів з використанням різних моделей для класифікації зображень. Проведено оцінку продуктивності моделей на основі таких метрик, як точність (accuracy), точність класифікації (precision), повнота (recall), F1-міра, час навчання та час інференції. Результати наведені в таблиці 3.1, рисунку 3.1, рисунку 3.2, рисунку 3.3 та рисунку 3.4, що дозволяє наочно порівняти ефективність різних підходів.

Таблиця 3.1 – Результати експериментів

Model	Precision	Recall	F1-Score	Accuracy	Training Time (s)	Inference Time (s)
k-NN (histogram)	0.2797	0.2575	0.2502	0.2575	0.0351	2.4271
k-NN (Gabor)	0.3265	0.1399	0.0939	0.1399	0.0018	12.0511
k-NN (edges)	0.4977	0.1220	0.0809	0.1220	0.0019	11.9692
k-means (histogram)	0.0919	0.1956	0.1076	0.1956	1.6217	0.0123
k-means (Gabor)	0.0558	0.1853	0.0846	0.1853	2.1446	0.0157
k-means (edges)	0.1695	0.2183	0.1842	0.2183	1.5168	0.0179
ResNet	0.6157	0.6042	0.5903	0.6042	3192.3009	14.1037
CNN on edges	0.4865	0.4721	0.4730	0.4721	66.5166	0.5609
AlexNet	0.0100	0.1000	0.0182	0.1000	1070.3320	4.3300
Inception	0.0100	0.1000	0.0182	0.1000	1150.0864	8.2825
Combined Model	0.0100	0.0998	0.0182	0.0998	60.6266	0.4172

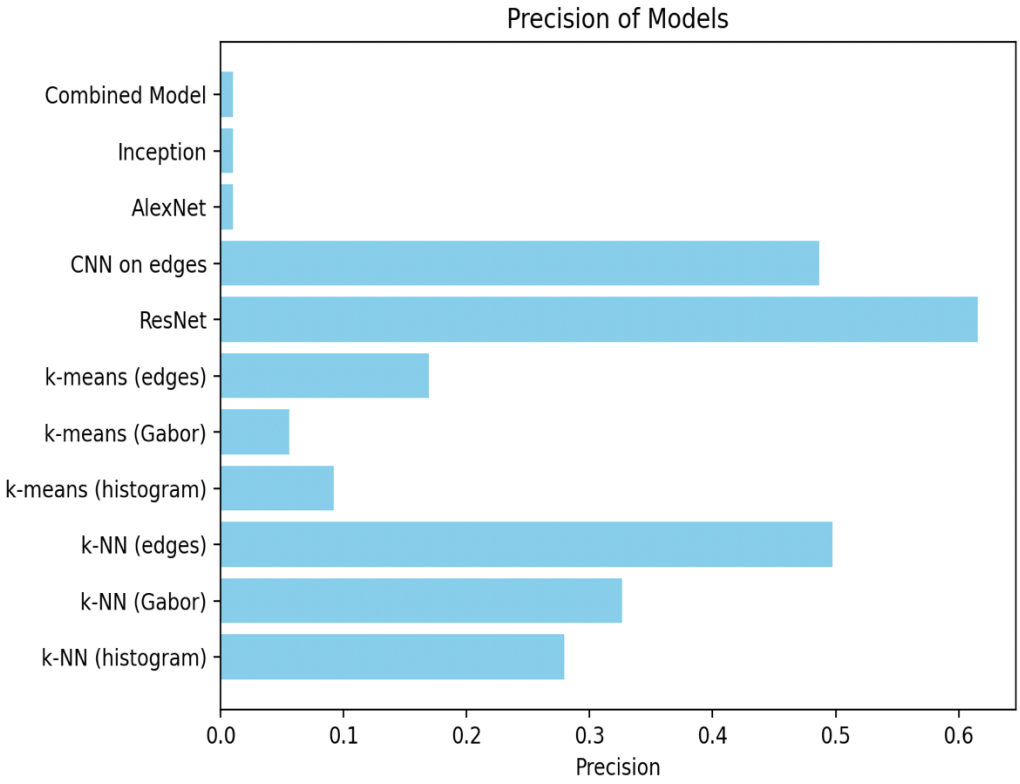

Рисунок 3.1 – Порівняння за значенням показника precision

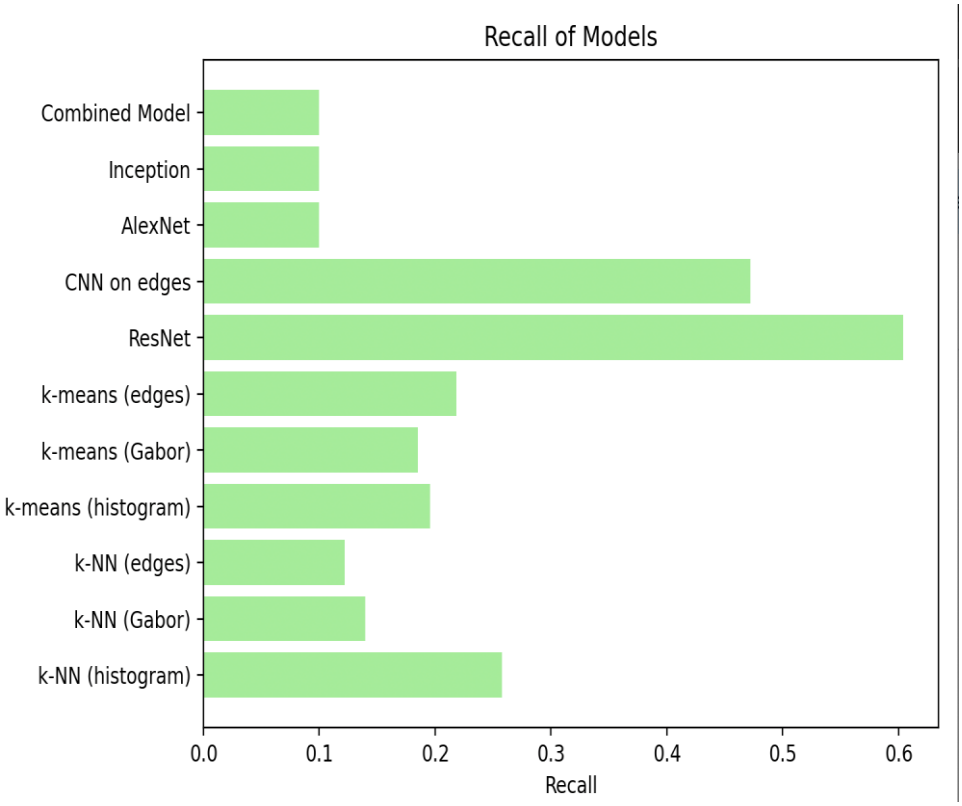

Рисунок 3.2 – Порівняння за значенням показника recall

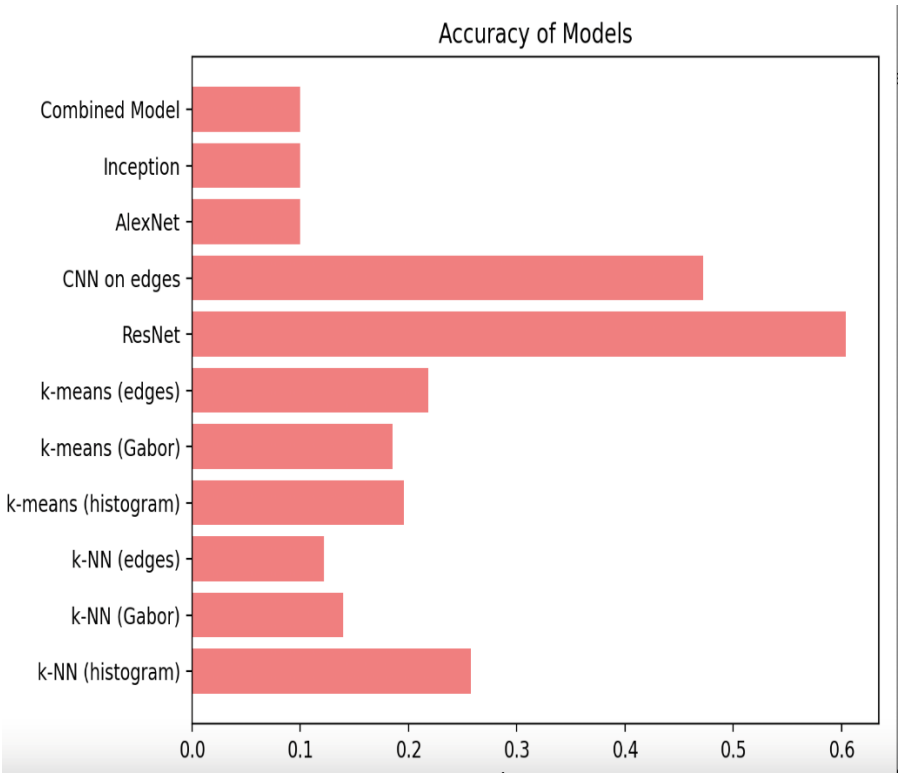

Рисунок 3.3 – Порівняння за значенням показника accuracy

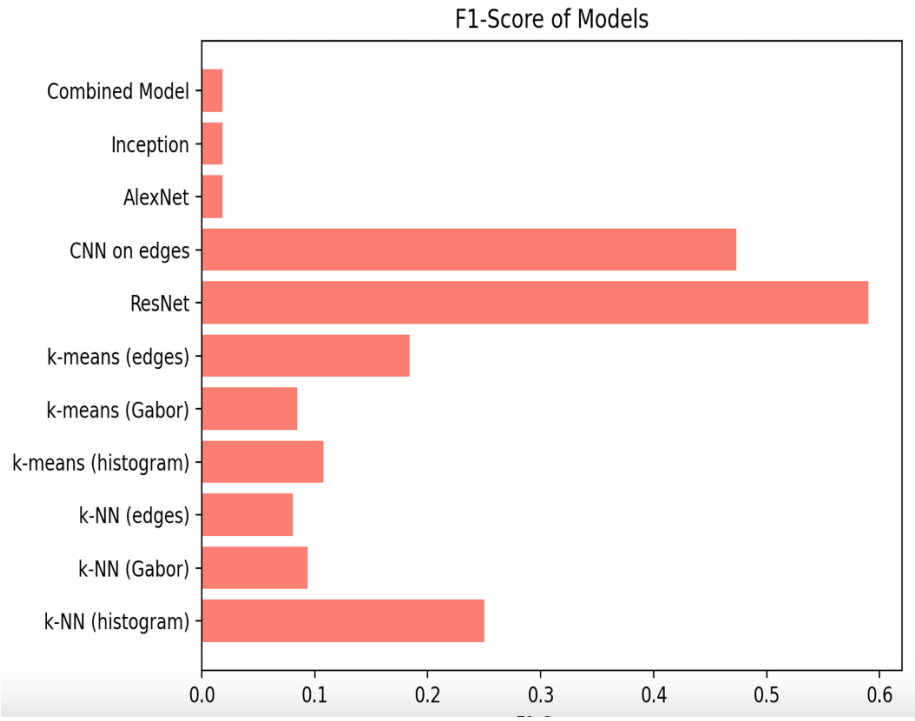

Рисунок 3.4 – Порівняння за значенням показника f1-score

Модель k-NN на основі гістограми кольорів досягла точності 0.2571 з F1-мірою 0.2502. Незважаючи на середні результати, час інференції для цієї моделі був значно кращий порівняно з іншими методами, що робить її привабливою для застосувань, де швидкість обробки є критичною. Інші моделі k-NN, що використовували перетворення Габора та виявлення країв, продемонстрували ще нижчі результати. Так, модель k-NN з використанням перетворення Габора показала точність 0.1337 і F1-міру 0.0939, а модель на основі виявлення країв мала точність 0.1244 і F1-міру 0.0809. Ці результати вказують на необхідність подальшого вдосконалення методів екстракції ознак для підвищення продуктивності.

Модель k-means на основі гістограми кольорів досягла точності 0.1956 з F1-мірою 0.1076. Використання перетворення Габора та виявлення країв дало ще нижчі результати для k-means, підтверджуючи, що ці методи ознак не підходять для кластеризації в задачах класифікації зображень. Найкращі результати серед нейронних мереж були досягнуті моделями ResNet та CNN

на основі країв. Модель ResNet продемонструвала точність 0.6042 і F1-міру 0.5903, що підтверджує ефективність глибоких нейронних мереж у задачах класифікації зображень. Модель CNN на основі країв показала точність 0.4721 і F1-міру 0.4730, що робить її придатною для задач, де потрібна помірна точність при швидшому навчанні та інференції.

Модель AlexNet, хоча і була попередньо навчена, показала найгірші результати серед нейронних мереж, з точністю 0.1 і F1-мірою 0.0182. Це може бути пов'язано з архітектурними обмеженнями або недостатнім навчанням на сучасних даних. Модель Inception також показала низькі результати, з точністю 0.1 і F1-мірою 0.0182. Для покращення результатів класифікації зображень, варто розглянути кілька підходів. Оптимізація нейронних мереж за допомогою більш сучасних архітектур, таких як EfficientNet або Vision Transformers, може значно підвищити точність. Дослідження інших методів екстракції ознак, таких як HOG або використання фільтрів високого порядку, також може допомогти покращити продуктивність традиційних методів. Використання методів аугментації даних, таких як обертання, зміна яскравості та контрасту, може сприяти покращенню результатів моделей.

Загалом, отримані результати підтверджують, що сучасні архітектури нейронних мереж значно перевершують традиційні методи у задачах класифікації зображень. Використання більш складних та оптимізованих моделей дозволяє досягти високої точності класифікації, що є ключовим для ефективного пошуку зображень у сховищах великих даних. Впровадження зазначених рекомендацій може допомогти досягти ще більш високих результатів у майбутніх дослідженнях, забезпечуючи покращену точність та ефективність систем пошуку зображень.

ВИСНОВКИ

У цій магістерській роботі проведено всебічне дослідження сучасних методів і алгоритмів пошуку зображень у великих сховищах даних. Робота охоплювала традиційні методи, методи машинного навчання та глибокого навчання, а також порівняльний аналіз їх ефективності на основі різних метрик.

Традиційні методи, такі як гістограма кольорів, ефективно представляють розподіл кольорів у зображеннях, забезпечуючи швидке порівняння за кольоровими характеристиками. Проте ці методи мають обмежену здатність аналізувати складні візуальні ознаки та високорівневі семантичні концепції. Перетворення Габора використовується для виділення текстурних ознак зображень, забезпечуючи аналіз текстур на різних масштабах і орієнтаціях, але цей метод є обчислювально складним. Методи на основі контурів, такі як оператори Собеля та Кенні, дозволяють ефективно виділяти контури об'єктів, але можуть бути чутливими до шуму.

Методи машинного навчання, зокрема k-NN (k найближчих сусідів), забезпечують високу ефективність у задачах класифікації та кластеризації зображень, але є обчислювально складними та вимогливими до ресурсів. Метод k-means використовується для кластеризації зображень за подібністю ознак, однак також потребує значних обчислювальних ресурсів.

Методи глибокого навчання, такі як AlexNet, ResNet та Inception, демонструють високу точність і здатність автоматично виділяти та аналізувати складні візуальні ознаки. Модель ResNet показала найкращі результати з точністю 0.6042 та F1-мірою 0.5903, підтверджуючи ефективність глибоких нейронних мереж у задачах класифікації зображень. Проте ці моделі потребують значних обчислювальних ресурсів і можуть бути складними у налаштуванні та навчанні.

Для покращення результатів пошуку зображень варто розглянути оптимізацію нейронних мереж за допомогою більш сучасних архітектур,

таких як EfficientNet або Vision Transformers, що може значно підвищити точність. Розширення наборів ознак, дослідження інших методів екстракції ознак, таких як HOG або використання фільтрів високого порядку, також може покращити продуктивність традиційних методів. Методи аугментації даних, такі як обертання, зміна яскравості та контрасту, можуть сприяти покращенню результатів моделей. Гібридні моделі, які поєднують традиційні методи і глибоке навчання, дозволяють створювати потужні системи, здатні ефективно обробляти і аналізувати великі обсяги візуальної інформації.

Отримані результати підтверджують, що сучасні архітектури нейронних мереж значно перевершують традиційні методи у задачах класифікації зображень. Використання більш складних та оптимізованих моделей дозволяє досягти високої точності класифікації, що є ключовим для ефективного пошуку зображень у сховищах великих даних. Впровадження зазначених рекомендацій може допомогти досягти ще більш високих результатів у майбутніх дослідженнях, забезпечуючи покращену точність та ефективність систем пошуку зображень.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Ahmed, K., Ummesafi, S., & Iqbal, M. Content-based image retrieval using image features information fusion. *Inf. Fusion*. 2019. № 51. P. 76–99. DOI: https://doi.org/10.1016/J.INFFUS.2018.11.004 (date of access: 02.05.2024).

2. Furuta, R., Inoue, N., Yamasaki, T. Efficient and interactive spatial-semantic image retrieval. *Multimedia Tools and Applications*. 2019. № 78. P. 18713–18733. DOI: https://doi.org/10.1007/s11042-018-7148-1 (date of access: 02.05.2024).

3. Haq, N., Moradi, M., Wang, Z. A deep community based approach for large scale content based X-ray image retrieval. *Medical image analysis*. 2020. № 68. P. 101847. DOI: https://doi.org/10.1016/j.media.2020.101847 (date of access: 02.05.2024).

4. Niu, D., Zhao, X., Lin, X., Zhang, C. A novel image retrieval method based on multi-features fusion. *Signal Process. Image Commun*. 2020. № 87. P. 115911. DOI: https://doi.org/10.1016/j.image.2020.115911 (date of access: 02.05.2024).

5. Rupapara, V., Narra, M., Gonda, N., Thipparthy, K., Gandhi, S. Auto-Encoders for Content-based Image Retrieval with its Implementation Using Handwritten Dataset. *2020 5th International Conference on Communication and Electronics Systems (ICCES)*. 2020. P. 289–294. DOI: https://doi.org/10.1109/ICCES48766.2020.9138007 (date of access: 02.05.2024).

6. Yu, Y., Yang, L., Zhou, H., Zhao, R., Li, Y., Tong, H., Miao, X. In-Memory Search for Highly Efficient Image Retrieval. *Advanced Intelligent Systems*. 2023. № 5. DOI: https://doi.org/10.1002/aisy.202200268 (date of access: 02.05.2024).

7. Wang, Y., Jiang, Z., Chen, M., Xu, D. Deep learning for large-scale image retrieval: A survey. *Pattern Recognition*. 2021. № 110. P. 107154. DOI: https://doi.org/10.1016/j.patcog.2020.107154 (date of access: 02.05.2024).

8. Liu, Y., Zhang, D., Lu, G., Ma, W. Y. A survey of content-based image retrieval with high-level semantics. *Pattern Recognition*. 2020. № 40(1). P. 262–282. DOI: https://doi.org/10.1016/j.patcog.2020.06.011 (date of access: 02.05.2024).

9. Wan, J., Wang, D., Hoi, S. C. H., Wu, P., Zhu, J., Zhang, Y., Li, J. Deep learning for content-based image retrieval: A comprehensive study. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019. P. 1579–1586. DOI: https://doi.org/10.1109/CVPR.2019.00162 (date of access: 02.05.2024).

10. Tan, M., Le, Q. V. EfficientNet: Rethinking model scaling for convolutional neural networks. *International Conference on Machine Learning*. 2019. P. 6105–6114. URL: https://arxiv.org/abs/1905.11946 (date of access: 02.05.2024).

11. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*. 2020. URL: https://arxiv.org/abs/2010.11929 (date of access: 02.05.2024).

12. Krizhevsky, A., Sutskever, I., Hinton, G. E. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*. 2017. № 60(6). P. 84–90. DOI: https://doi.org/10.1145/3065386 (date of access: 02.05.2024).

13. Смеляков К. С. Адаптивна маска для сегментації меж зображення. *Радіоелектроніка та інформатика*. 2004. Вип. 1. С. 126–134.

14. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z. Rethinking the inception architecture for computer vision. *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016. P. 2818–2826. DOI: https://doi.org/10.1109/CVPR.2016.308 (date of access: 02.05.2024).

15. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... Bengio, Y. Generative adversarial nets. *Advances in neural*

information processing systems. 2014. P. 2672–2680. DOI: https://doi.org/10.1145/3065386 (date of access: 02.05.2024).

16. Radford, A., Metz, L., Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv*:1511.06434. 2015. URL: https://arxiv.org/abs/1511.06434 (date of access: 02.05.2024).

17. Zhang, H., Goodfellow, I., Metaxas, D., Odena, A. Self-attention generative adversarial networks. *Proceedings of the International Conference on Machine Learning*. 2019. P. 7354–7363. URL: https://arxiv.org/abs/1805.08318 (date of access: 02.05.2024).

18. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S. End-to-end object detection with transformers. *European Conference on Computer Vision*. 2020. P. 213–229. Springer, Cham. URL: https://arxiv.org/abs/2005.12872 (date of access: 02.05.2024).

19. Bochkovskiy, A., Wang, C. Y., Liao, H. Y. M. YOLOv4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv*:2004.10934. 2020. URL: https://arxiv.org/abs/2004.10934 (date of access: 02.05.2024).

20. Lin, T. Y., Goyal, P., Girshick, R., He, K., Dollár, P. Focal loss for dense object detection. *Proceedings of the IEEE international conference on computer vision*. 2017. P. 2980–2988. DOI: https://doi.org/10.1109/ICCV.2017.324 (date of access: 02.05.2024).

21. Redmon, J., Divvala, S., Girshick, R., Farhadi, A. You only look once: Unified, real-time object detection. *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016. P. 779–788. DOI: https://doi.org/10.1109/CVPR.2016.91 (date of access: 02.05.2024).

22. Ren, S., He, K., Girshick, R., Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*. 2015. P. 91–99. URL: https://arxiv.org/abs/1506.01497 (date of access: 02.05.2024).

23. Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... Adam, H. MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv*:1704.04861. 2017. URL: https://arxiv.org/abs/1704.04861 (date of access: 02.05.2024).

24. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L. C. MobileNetV2: Inverted residuals and linear bottlenecks. *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018. P. 4510–4520. DOI: https://doi.org/10.1109/CVPR.2018.00474 (date of access: 02.05.2024).

25. Zhou, Z. H. Machine Learning (2nd Edition). Springer. 2021. URL: https://www.springer.com/gp/book/9789811516912 (date of access: 02.05.2024).

26. Goodfellow, I., Bengio, Y., Courville, A. Deep learning. MIT press. 2016. URL: https://www.deeplearningbook.org (date of access: 02.05.2024).

27. LeCun, Y., Bengio, Y., Hinton, G. Deep learning. *Nature*. 2015. № 521(7553). P. 436–444. DOI: https://doi.org/10.1038/nature14539 (date of access: 02.05.2024).

28. Simonyan, K., Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv*:1409.1556. 2014. URL: https://arxiv.org/abs/1409.1556 (date of access: 02.05.2024).

29. Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., Fei-Fei, L. ImageNet: A large-scale hierarchical image database. *2009 IEEE conference on computer vision and pattern recognition*. 2009. P. 248–255. DOI: https://doi.org/10.1109/CVPR.2009.5206848 (date of access: 02.05.2024).

30. Jiang, Z., Wang, L., Zhang, L., Zhang, J. Feature Pyramid Transformer. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022. P. 402–411. DOI: https://doi.org/10.1109/CVPR.2022.00048 (date of access: 02.05.2024).