ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Нейромережеві моделі для розпізнавання зображень»

на здобуття освітнього ступеня бакалавра
зі спеціальності 123 – Комп'ютерна інженерія
(код, найменування спеціальності)
освітньо-професійної програми Комп'ютерна інженерія
(назва)

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

_____ *Юрій ЛЕВЧЕНКО*
(підпис) *Ім'я, ПРІЗВИЩЕ здобувача*

Виконав: здобувач вищої освіти гр. КІД-41
Юрій ЛЕВЧЕНКО
Ім'я, ПРІЗВИЩЕ
Керівник: Олександр ЗУБ
науковий ступінь, вчене звання Ім'я, ПРІЗВИЩЕ
Рецензент: _____
науковий ступінь, вчене звання Ім'я, ПРІЗВИЩЕ

Київ 2025

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут Інформаційних технологій

Кафедра Комп'ютерної інженерії

Ступінь вищої освіти бакалавр

Спеціальність 123 – Комп'ютерна інженерія

Освітньо-професійна програма Комп'ютерна інженерія

ЗАТВЕРДЖУЮ
Завідувач кафедри
Комп'ютерної інженерії
Наталія ЛАЩЕВСЬКА
«_____» _____ 2025р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Левченко Юрій Олегович

(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Нейромережеві моделі для розпізнавання зображень
керівник кваліфікаційної роботи Олександр ЗУБ,
(Ім'я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)
затверджені наказом Державного університету інформаційно-комунікаційних технологій від «24» лютого 2025р. № 56
2. Строк подання кваліфікаційної роботи «3» червня 2024р.
3. Вихідні дані до кваліфікаційної роботи: науково-технічна література; архітектури нейронних мереж(CNN, RNN, Transformer, ViT, ResNet, Inception, EfficientNet); набори даних(MNIST, CIFAR-10, CIFAR-100, ImageNet); метрики якості(Top-1 Accuracy, Top-5 Accuracy, FLOPs, Params (Parameters), Inference Time)
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Огляд існуючих нейромережевих моделей, призначених для розпізнавання зображень
2. Релевантні критерії для здійснення об'єктивного порівняльного аналізу моделей
3. Сильні та слабкі сторони кожної моделі з точки зору їхньої архітектури, принципів роботи, переваг, обмежень та потенційних застосувань
4. Рекомендації щодо вибору оптимальної нейромережевої моделі для конкретних задач розпізнавання зображень
5. Перелік ілюстративного матеріалу: *презентація*
6. Дата видачі завдання«24» лютого 2025р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Збір даних	25.02-24.03.2025	Виконано
2	Обробка матеріалу	23.03-05.04.2025	Виконано
3	Розробка теоретичної частини	06.04-10.04.2025	Виконано
4	Розробка моделі мережі	11.04-20.04.2025	Виконано
5	Висновки за результатами аналізу	21.04-06.05.2025	Виконано
6	Оформлення пояснювальної записки	07.05-08.05.2025	Виконано
7	Розробка презентації	09.05-11.05.2025	Виконано
8	Передзахист	14.05-17.05.2025	Виконано
9	Здача в деканат	27.05-03.06.2025	Виконано

Здобувач вищої освіти _____ Юрій ЛЕВЧЕНКО
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

Керівник кваліфікаційної роботи _____ Олександр ЗУБ
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра: 61 стор., 9 рис., 10 табл., 30 джерел.

Мета роботи – порівняльний аналіз різних архітектур нейромережевих моделей для задачі розпізнавання зображень.

Об'єкт дослідження – процес розпізнавання зображень за допомогою нейромережевих моделей.

Предмет дослідження – нейромережеві моделі, що використовуються для розпізнавання зображень, включаючи їхні архітектури, алгоритми навчання, параметри та можливості застосування.

Короткий зміст роботи: Кваліфікаційна робота присвячена аналізу нейромережевих моделей для задачі розпізнавання зображень. В роботі проведено огляд існуючих типів нейромережевих моделей, таких як згорткові, рекурентні та трансформери, та порівняно їхні архітектури, переваги та недоліки. Проаналізовано застосування різних моделей для конкретних задач розпізнавання. Результати роботи можуть бути використані для вибору оптимальної моделі при розробці систем розпізнавання зображень, для надання практичних рекомендацій розробникам систем розпізнавання зображень, дослідникам та інженерам при виборі оптимальної нейромережевої моделі для конкретних застосувань. Це сприятиме підвищенню ефективності та надійності систем розпізнавання зображень у різних галузях, таких як медицина (автоматичний аналіз медичних зображень), безпека (системи розпізнавання облич), автономні транспортні засоби (розпізнавання дорожніх знаків та об'єктів) та багато інших.

Ключові слова: нейронні мережі, розпізнавання зображень, згорткові нейронні мережі, CNN, рекурентні нейронні мережі, RNN, трансформери, машинне навчання, комп'ютерний зір, порівняльний аналіз.

ЗМІСТ

ВСТУП.....9
1 Огляд літератури.....11
1.1 Загальні відомості про нейронні мережі та їх застосування в розпізнаванні зображень.....11
1.2 Огляд існуючих нейромережевих моделей.....11
1.2.1 Згорткові нейронні мережі (Convolutional Neural Networks, CNN).....12
1.2.2 Рекурентні нейронні мережі (Recurrent Neural Networks, RNN).....14
1.2.3 Трансформери (Transformers).....16
1.3 Огляд наборів даних для розпізнавання зображень.....21
1.3.1 MNIST (Modified National Institute of Standards and Technology database) 22
1.3.2 CIFAR (Canadian Institute For Advanced Research).....23
1.3.3 ImageNet.....25
1.4 Аналіз ключових параметрів архітектур нейронних мереж для розпізнавання зображень.....27
1.4.1 Глибина мережі.....27
1.4.2 Ширина мережі.....29
1.4.3 Типи шарів.....31
1.4.4 З'єднання між шарами.....32
1.4.5 Інші параметри.....34
1.5 Порівняльний аналіз існуючих підходів.....35
1.5.1 Порівняння нейромережевих моделей.....36
1.5.2 Порівняння підходів до навчання.....38
2 Методологія дослідження.....40
2.1 Опис критеріїв вибору моделей для аналізу.....40
2.2 Опис критеріїв вибору наборів даних для аналізу.....41
2.3 Опис методів аналізу та порівняння моделей.....43
2.4 Опис процедури перенавчання моделей (fine-tuning).....45
3 Аналіз нейромережевих моделей для розпізнавання зображень.....49
3.1 Детальний аналіз обраних моделей.....49
3.1.1 ResNet (Residual Network).....50
3.1.2 Inception.....53
3.1.3 EfficientNet.....56
3.1.4 Vision Transformer (ViT).....61
3.2 Порівняльний аналіз моделей на основі визначених критеріїв.....63

3.3 Аналіз впливу параметрів моделей на ефективність розпізнавання зображень.....67
3.4 Аналіз можливостей застосування кожної моделі для різних задач розпізнавання зображень..... 70
Висновки.....72
Список Використаних джерел..... 74
Додаток А.....77
Додаток Б..... 80

ВСТУП

У сучасному світі розпізнавання зображень перетворилося з вузькоспеціалізованої області досліджень на фундаментальну технологію, що пронизує практично всі аспекти нашого життя. Від медичної діагностики та систем безпеки до автономних транспортних засобів і розважальних платформ, можливості автоматичного аналізу та інтерпретації візуальної інформації відкривають безпрецедентні перспективи для автоматизації, оптимізації та інновацій. З розвитком цифрових технологій та експоненційним зростанням обсягів візуальних даних, ефективне та точне розпізнавання зображень стає критично важливим для вилучення корисної інформації та прийняття обґрунтованих рішень.

Нейромережеві моделі, натхненні принципами функціонування людського мозку, стали потужним інструментом для вирішення різноманітних задач розпізнавання зображень. Їхня здатність автоматично вивчати складні закономірності та абстракції з великих обсягів даних дозволяє досягати вражаючих результатів у таких завданнях, як класифікація об'єктів, виявлення облич, сегментація зображень та багато інших. Глибоке навчання, підгалузь машинного навчання, що лежить в основі сучасних нейромережевих моделей, зробило революцію в галузі комп'ютерного зору, забезпечивши значний прогрес у порівнянні з традиційними методами.

Однак, незважаючи на значні досягнення, існують невід'ємні недоліки, пов'язані з розпізнаванням складних зображень у реальних умовах. Варіативність об'єктів, зміни в освітленні, шум, перешкоди та інші фактори можуть суттєво ускладнити процес розпізнавання. Крім того, розробка та навчання ефективних нейромережевих моделей вимагає значних обчислювальних ресурсів та великих обсягів розмічених даних, що також є серйозною проблемою. Тому, глибокий аналіз та ретельне порівняння ефективності різних нейромережевих моделей для розпізнавання зображень залишається актуальною науково-практичною задачею,

спрямованою на подолання цих викликів та подальший розвиток технології.

Метою даної роботи є проведення всебічного аналізу ефективності різних нейромережевих моделей, що застосовуються для вирішення задачі розпізнавання зображень. Дослідження спрямоване на виявлення сильних і слабких сторін кожної моделі, визначення факторів, що впливають на їхню продуктивність, та формування рекомендацій щодо їхнього вибору для конкретних застосувань.

1 Огляд літератури

1.1 Загальні відомості про нейронні мережі та їх застосування в розпізнаванні зображень

Нейронні мережі є обчислювальними системами, натхненними структурою та функціонуванням людського мозку. Вони складаються з взаємоз'язаних вузлів, або "нейронів", організованих у шари, які обробляють інформацію шляхом передачі сигналів між собою.

У контексті розпізнавання зображень, нейронні мережі використовуються для автоматичного вилучення та аналізу візуальних ознак з зображень. Шляхом навчання на великих наборах даних, вони можуть навчитися розпізнавати складні закономірності та абстракції, що дозволяє їм класифікувати зображення, виявляти об'єкти, сегментувати зображення та виконувати інші завдання, пов'язані з аналізом візуальної інформації.

Основні переваги нейронних мереж у порівнянні з традиційними методами розпізнавання зображень включають їхню здатність до автоматичного вилучення ознак, високу точність та можливість обробки великих обсягів даних. Однак, вони також мають певні недоліки, такі як висока обчислювальна складність, потреба у великих наборах даних для навчання та складність інтерпретації їхніх рішень.

1.2 Огляд існуючих нейромережевих моделей

Різноманітність завдань та структур даних зумовила розробку спеціалізованих нейромережевих архітектур, кожна з яких оптимізована для ефективної обробки певних типів інформації та вирішення відповідних

прикладних задач.

Для аналізу просторових даних, таких як зображення, де ключове значення мають локальні та глобальні просторові ознаки, були розроблені специфічні архітектури. Водночас, для обробки послідовних даних, де важливим є збереження та використання інформації про часовий порядок елементів, застосовуються моделі, здатні підтримувати внутрішній стан. Останні розробки у сфері архітектур зосередилися на впровадженні механізмів уваги, що дозволило підвищити ефективність обробки довгих послідовностей та паралелізувати обчислення. Далі буде представлено детальний опис основних представників цих архітектур, їхні структурні особливості та типові сфери застосування.

1.2.1 Згорткові нейронні мережі (Convolutional Neural Networks, CNN)

Згорткові нейронні мережі, або CNN, тривалий час залишалися найбільш поширеним і ефективним типом нейромережевих моделей у царині розпізнавання зображень. Їхня виняткова здатність до обробки візуальної інформації зробила їх фундаментом для величезної кількості застосувань у комп'ютерному зорі.

Основна сила CNN полягає в їхньому унікальному механізмі – операції згортки. Ця операція дозволяє мережі автоматично вилучати просторові ознаки безпосередньо із зображень. Завдяки згортці, модель може ефективно виявляти різноманітні візуальні патерни, такі як краї, кути, текстури та інші важливі елементи, причому незалежно від їхнього конкретного положення на зображенні. Це дає CNN величезну перевагу, оскільки вони здатні розпізнавати об'єкти, навіть якщо вони зміщені, повернуті або знаходяться в різних частинах зображення. Саме тому згорткові мережі ідеально підходять для вирішення таких складних завдань, як класифікація зображень (визначення, що зображено на знімку), виявлення об'єктів (локалізація та ідентифікація об'єктів у кадрі) та сегментація зображень (розділення зображення на смислові області).

Типова архітектура CNN, що представлена на рисунку 1.1, складається з кількох ключових типів шарів, кожен з яких виконує свою важливу функцію у

процесі обробки зображень.

Наріжним каменем є шари згортки (convolutional layers). Вони застосовують набір спеціальних фільтрів, які сканують вхідне зображення, подібно до того, як око людини виділяє певні деталі. Кожен фільтр "шукає" певний патерн або ознаку, а результатом його роботи є так звані карти ознак (feature maps), що підсвічують місця, де ця ознака була виявлена на зображенні.

Наступними у ланцюжку часто йдуть шари пулінгу (pooling layers). Їхнє основне завдання – зменшити розмірність цих карт ознак. Це не тільки значно знижує обчислювальну складність моделі, але й робить мережу більш стійкою до невеликих зсувів, поворотів або деформацій об'єктів на зображенні. Шар пулінгу, по суті, узагальнює інформацію, залишаючи лише найважливіші активації.

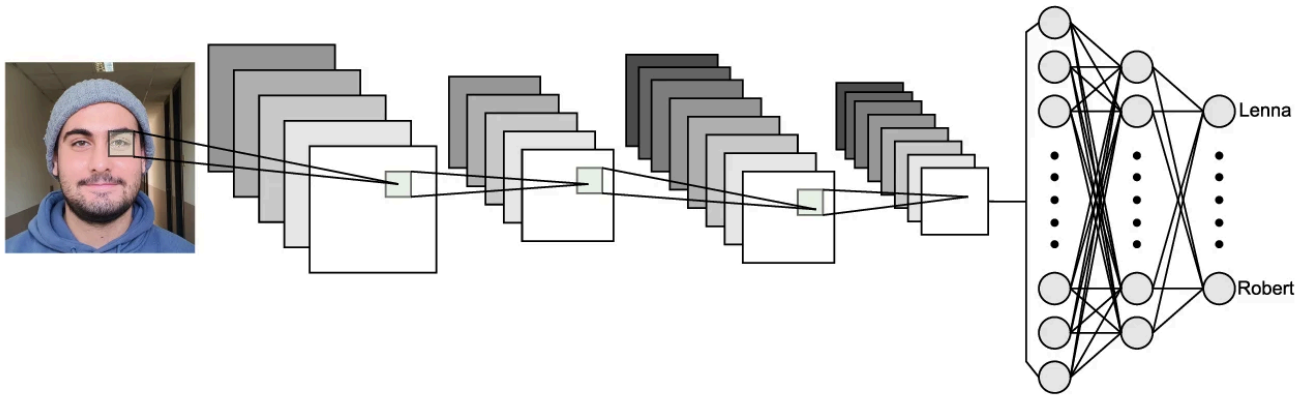


Рисунок 1.1 – Архітектура Convolutional Neural Network

Нарешті, після того, як згорткові та пулінгові шари вилучили високорівневі, абстрактні ознаки з зображення, ці дані передаються до повністю зв'язаних шарів (fully connected layers). Ці шари функціонують як традиційна нейронна мережа, яка приймає вилучені ознаки та використовує їх для виконання кінцевого завдання, наприклад, для класифікації зображення у певну категорію.

Переваги CNN:

- ефективне вилучення просторових ознак;

- інваріантність до зсуву, масштабу та повороту;
- висока точність у багатьох задачах розпізнавання зображень.

Недоліки CNN:

- висока обчислювальна складність для великих зображень;
- потребують великих наборів даних для навчання;
- можуть бути чутливими до невеликих змін у зображенні.

1.2.2 Рекурентні нейронні мережі (Recurrent Neural Networks, RNN)

Рекурентні нейронні мережі, або RNN, є потужним інструментом у світі штучного інтелекту, спеціально розробленим для ефективної роботи з послідовностями даних. Це робить їх ідеальними для обробки різноманітних інформаційних потоків, таких як відеоматеріали, де послідовність кадрів має вирішальне значення, а також часові ряди зображень, де важливо відстежувати зміни у візуальних даних з плином часу.

Центральним елементом архітектури RNN є їхній механізм рекурентності. Це дозволяє мережі зберігати та використовувати інформацію про попередні кадри або часові кроки. Завдяки цьому рекурентному зв'язку, мережа набуває здатності "пам'ятати" попередні вхідні дані, що є критично важливим для розуміння контексту та динаміки послідовності. Ця інформація з минулого активно використовується для обробки поточного входу, дозволяючи моделі формувати більш повне та осмислене представлення даних.

З огляду на цю унікальну здатність враховувати часову динаміку, RNN виявляються надзвичайно корисними для широкого спектру завдань, де контекст та послідовність відіграють ключову роль. Наприклад, вони чудово підходять для розпізнавання жестів, де рух тіла розвивається у часі, або для аналізу відеопотоку,

де потрібно виявляти події чи зміни у реальному часі.

Архітектура RNN побудована на концепції рекурентних шарів. Як зображено на рисунку 1.2, ці шари відповідають за послідовну обробку вхідних даних, крок за кроком переходячи від одного елемента послідовності до іншого. Ключовою особливістю кожного рекурентного шару є його зворотний зв'язок. Саме цей зворотний зв'язок дозволяє інформації "перетікати" з одного часового кроку на наступний, формуючи так званий "прихований стан" або "пам'ять" мережі. Цей прихований стан інкапсулює всю релевантну інформацію, яку мережа накопичила з попередніх елементів послідовності.

Хоча базові RNN є ефективними, вони стикаються з певними викликами, такими як проблема "зникаючого градієнта", що ускладнює навчання глибоких мереж та захоплення довгострокових залежностей. Для вирішення цих проблем були розроблені більш досконалі варіанти.

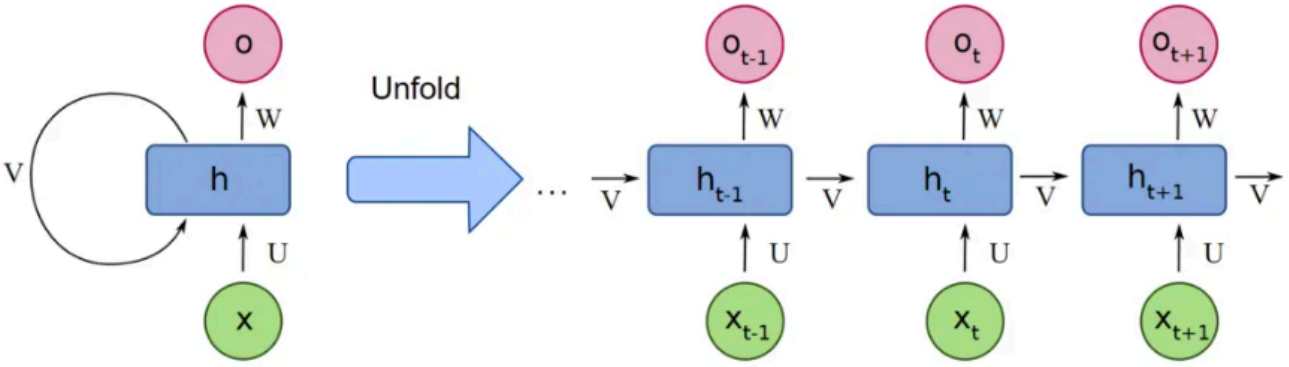


Рисунок 1.2 – Архітектура Recurrent Neural Network

Одним з найвідоміших прикладів є LSTM (Long Short-Term Memory). Ці мережі були спеціально створені для подолання проблеми зникаючого градієнта, що дозволяє їм ефективно навчатися на довгих послідовностях. LSTM досягають цього завдяки використанню так званих вентилів (gates) – складних механізмів, які контролюють потік інформації через мережу. Ці вентилі вирішують, яка інформація повинна бути збережена в пам'яті, яка повинна бути оновлена, а яка

повинна бути забута, забезпечуючи таким чином більш стабільне та ефективне навчання.

Іншим популярним та дещо спрощеним варіантом є GRU (Gated Recurrent Unit). GRU можна вважати більш "легкою" версією LSTM, оскільки вони мають меншу кількість параметрів завдяки об'єднанню деяких вентилів. Незважаючи на свою простоту, GRU також надзвичайно ефективно обробляють послідовності та успішно вирішують проблему зникаючого градієнта, часто демонструючи порівнянну продуктивність з LSTM у багатьох завданнях.

Таким чином, RNN та їхні модифікації, такі як LSTM і GRU, залишаються фундаментальними компонентами у арсеналі фахівців зі штучного інтелекту для роботи з послідовними даними, дозволяючи моделям "пам'ятати" та "розуміти" динаміку інформації у часі.

Переваги RNN:

- здатність обробляти послідовності даних;
- можливість моделювати часову динаміку.

Недоліки RNN:

- схильність до проблеми зникаючого градієнту;
- складність паралелізації обчислень;
- можуть бути менш ефективними для обробки великих зображень, ніж CNN.

1.2.3 Трансформери (Transformers)

Трансформери, безперечно, є одним з найбільш значущих та революційних проривів у галузі штучного інтелекту за останні роки. Ця архітектура, яка початково здійснила справжню революцію в обробці природної мови (NLP),

з'явилася порівняно недавно, але вже встигла кардинально змінити підходи до вирішення численних складних завдань. Їхня унікальність і неперевершена ефективність полягає у використанні особливого, інноваційного механізму, відомого як механізм уваги (attention).

Цей механізм є справжнім серцем Трансформерів, дозволяючи їм не просто обробляти інформацію послідовно, а динамічно визначати та зважувати важливість різних частин вхідних даних. Подібно до того, як людина фокусується на ключових деталях, ігноруючи другорядні, механізм уваги дає змогу моделі зосереджуватися на найбільш релевантних елементах під час кожного етапу обробки. Це дозволяє їм виявляти складні взаємозв'язки та довгострокові залежності, які були недоступні для попередніх архітектур.

Завдяки цій безпрецедентній інновації, Трансформери не лише досягли феноменальних результатів у сфері мовних моделей, але й перетнули традиційні межі, продемонструвавши вражаючі можливості у багатьох інших складних завданнях. Зокрема, вони здійснили справжній переворот у комп'ютерному зорі, де показали виняткову продуктивність, зокрема в таких критично важливих застосуваннях, як розпізнавання зображень. Їхня здатність "розуміти" візуальний контекст та виділяти ключові ознаки зробила їх незамінним інструментом у сучасному AI.

За своєю суттю, архітектура Трансформера складається з двох основних частин: кодувальника (encoder) та декодувальника (decoder). Кодувальник відповідає за обробку вхідної інформації, перетворюючи її на спеціальне векторне представлення. Потім декодувальник використовує це представлення, щоб генерувати відповідні вихідні дані. Ключовим компонентом Трансформера є шар уваги (attention layer), який відіграє вирішальну роль. Саме цей шар обчислює ступінь важливості між кожною парою елементів у вхідній послідовності, дозволяючи моделі "розуміти" контекст і взаємозв'язки даних.

З огляду на їхню ефективність, було розроблено низку архітектур на основі Трансформерів, які успішно застосовуються у комп'ютерному зорі. Наприклад, на рисунку 1.3, що наведено далі, зображено як ViT (Vision Transformer) адаптує

архітектуру Трансформера. Основна ідея полягає в тому, щоб розглядати зображення не як сітку пікселів, а як послідовність "візуальних слів" або "патчів".

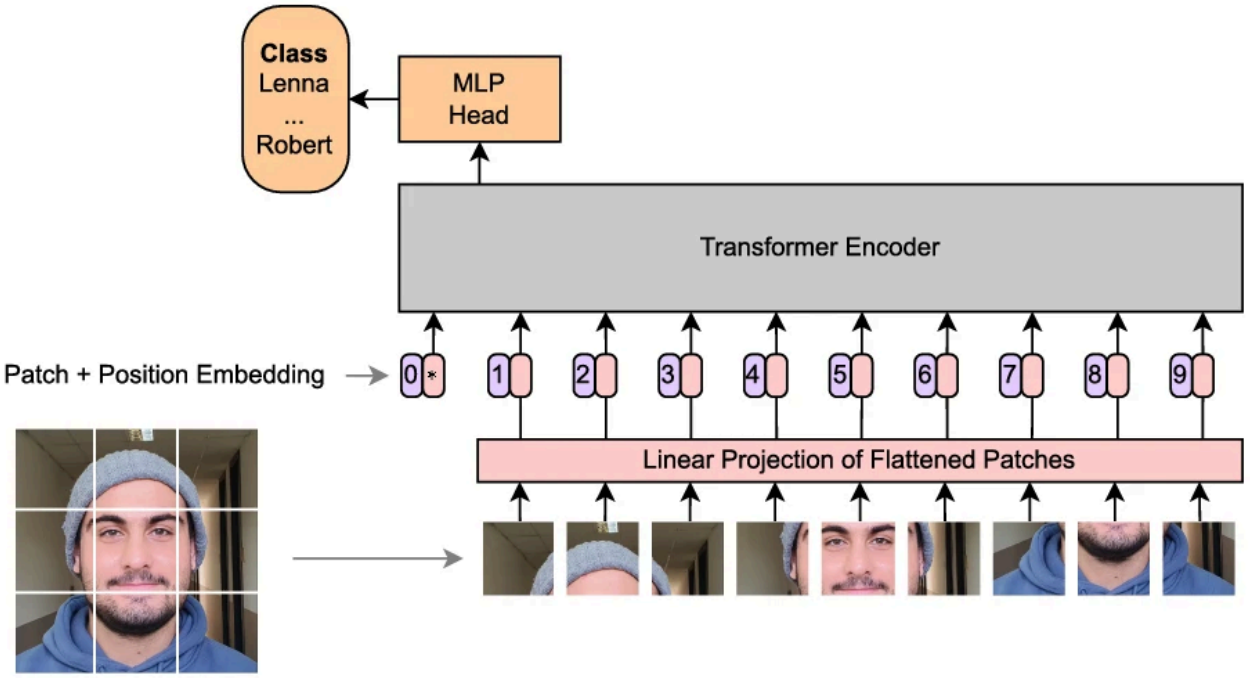


Рисунок 1.3 – Архітектура Vision Transformer

Робота кодувальника у ViT починається зі "розрізання" зображення на фіксовані патчі. Замість того, щоб подавати все зображення цілком, ViT ділить його на невеликі, непересічні квадрати, наприклад, 16×16 пікселів. Кожен такий патч розглядається як окремий "токен" або "слово".

Далі, кожен такий патч лінійно проектується у вектор певного розміру. Це схоже на створення вбудовування для кожного візуального патчу. Тобто, замість семантичного вбудовування слів, ми отримуємо візуальне вбудовування патчів.

До цих патчевих вбудовувань, як і в класичному Трансформері, додається позиційне кодування. Це критично важливо для зображень, оскільки положення патча на зображенні несе важливу інформацію про його розташування та сусідство. Без позиційного кодування модель не знала б, де знаходиться кожен патч, і це призвело б до втрати просторової інформації.

Особливим елементом у ViT є додавання класифікаційного токена (CLS token) на початку послідовності патчів. Цей спеціальний токен не відповідає

жодному патчу зображення, але його кінцеве представлення після проходження через кодувальник буде використовуватися для виконання завдання класифікації зображення. Модель вчиться агрегувати всю необхідну інформацію про зображення в цей один токен.

Після цих початкових кроків, підготовлена послідовність, а саме: класифікаційний токен, вбудовування патчів, позиційне кодування – подається на вхід багатошаровому кодувальнику Трансформера, арітектуру якого представлено на рисунку 1.4. Кожен шар цього кодувальника ідентичний шарам кодувальника класичного Трансформера і включає Multi-Head Self-Attention та Position-wise Feed-Forward Network.

Завдяки Багатоголовій самоувазі (Multi-Head Self-Attention) модель обчислює взаємозв'язки між різними патчами зображення. Вона може зрозуміти, які патчі є важливими для розуміння об'єктів або сцен на зображенні, і як вони співвідносяться один з одним, незалежно від їхнього фізичного розташування. Наприклад, якщо на зображенні є обличчя, механізм уваги може пов'язати патчі, що містять очі, ніс і рот, навіть якщо вони знаходяться на різних ділянках зображення. Це дозволяє ViT захоплювати глобальні залежності та контекст у зображенні, на відміну від згорткових мереж, які здебільшого фокусуються на локальних ознаках.

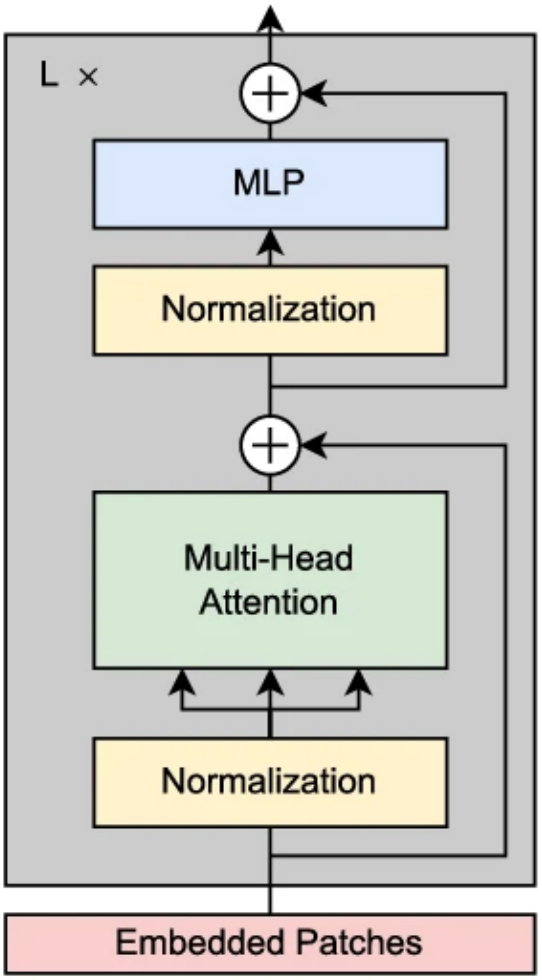


Рисунок 1.4 – Архітектура Transformer Encoder

Як і в класичному Трансформері, Позиційно-орієнтована пряма нейронна мережа (Position-wise Feed-Forward Network) додатково обробляє контекстуалізовані представлення кожного патча, додаючи нелінійність та поглиблюючи вивчені ознаки.

Залишкові з'єднання та нормалізація шарів забезпечують стабільність та ефективність навчання глибоких мереж.

Після проходження через всі шари кодувальника, вихідне представлення класифікаційного токена (CLS token) використовується для подачі на вхід простому класифікаційному шару (зазвичай багатошаровому персептрону), який видає прогноз класу зображення. Інші вихідні вектори патчів також містять багату інформацію і можуть бути використані для інших завдань, таких як виявлення об'єктів або сегментація.

Переваги Transformer:

- здатність моделювати глобальні залежності між елементами вхідних даних;
- висока паралельність обчислень;
- досягнення найсучасніших результатів у багатьох задачах комп'ютерного зору.

Недоліки Transformer:

- висока обчислювальна складність, особливо для великих зображень;
- потребують дуже великих наборів даних для навчання.

1.3 Огляд наборів даних для розпізнавання зображень

Навчання та тестування сучасних нейромережевих моделей, орієнтованих на розпізнавання зображень, неможливе без використання великих, якісно розмічених наборів даних. Такі датасети є основою для ефективного навчання моделей, оскільки забезпечують достатню кількість прикладів для узагальнення, виявлення закономірностей та підвищення точності класифікації. У сфері комп'ютерного зору сформувався перелік авторитетних та широко застосовуваних наборів даних, які стали де-факто стандартами для оцінювання продуктивності різних архітектур нейромереж. Вони використовуються як у наукових дослідженнях, так і в прикладних розробках, дозволяючи здійснювати об'єктивне порівняння між різними моделями за єдиними критеріями.

1.3.1 MNIST (Modified National Institute of Standards and Technology database)

Датасет MNIST є одним з найбільш знакових і широко використовуваних наборів даних у галузі машинного навчання, особливо коли йдеться про розробку та тестування алгоритмів, пов'язаних з розпізнаванням зображень. Його створили з конкретною метою – слугувати стандартизованим полігоном для всебічного тестування різноманітних алгоритмів класифікації, особливо тих, що застосовуються для вирішення задачі оптичного розпізнавання рукописних цифр. Фактично, MNIST став своєрідним "привіт, світ" для багатьох дослідників та ентузіастів, які починають свій шлях у світ нейронних мереж та глибокого навчання.

Походження MNIST сягає корінням двох оригінальних, дещо відмінних наборів даних, які були розроблені під егідою Національного інституту стандартів і технологій США (NIST). У цих початкових версіях були зібрані рукописні цифри, які писали як працівники американського Бюро перепису населення, так і школярі. Однак, щоб забезпечити кращу узагальненість і, що не менш важливо, створити справедливу платформу для порівняння продуктивності різних моделей, ці оригінальні зображення пройшли ретельну попередню обробку. Вони були нормалізовані за розміром і центровані таким чином, щоб кожна цифра знаходилася в центрі зображення, а її розмір був уніфікований. Всі зображення були приведені до єдиного, фіксованого формату – 28×28 пікселів, що значно спростило їхню подальшу обробку та аналіз будь-якими алгоритмами машинного навчання. Ця стандартизація дозволила дослідникам по всьому світу використовувати ідентичні вхідні дані, забезпечуючи можливість прямого та об'єктивного зіставлення результатів.

Переваги MNIST:

- набір добре структурований і легко завантажується через популярні бібліотеки, такі як TensorFlow, PyTorch тощо;

- служить точкою входу для початківців у галузі комп’ютерного зору та машинного навчання;
- дозволяє швидко перевіряти ефективність алгоритмів без потреби в обробці складних реальних даних.

Недоліки MNIST:

- більшість глибоких нейронних мереж досягають дуже високої точності (понад 99%), що знижує його цінність як тестового набору для складних моделей;
- лише цифри, обмежена кількість варіацій почерку порівняно з реальними сценаріями.

1.3.2 CIFAR (Canadian Institute For Advanced Research)

Комплекси даних CIFAR є знаковими та активно застосовуються в галузі комп'ютерного зору, зокрема в контексті класифікації зображень. Ці набори даних представляють собою суттєво складніший виклик порівняно з такими еталонами, як MNIST, що вимагає розробки та застосування більш складних моделей для досягнення високих показників точності та узагальненості.

Найбільш відомими та поширеними варіантами цієї серії є CIFAR-10 та CIFAR-100. Кожен з цих наборів містить 60 000 кольорових зображень, уніфікованих за розміром до 32×32 пікселів. Розподіл цих зображень передбачає 50 000 екземплярів для використання у фазі навчання моделі, тоді як решта 10 000 екземплярів призначені для об'єктивної оцінки її продуктивності на тестовій вибірці.

CIFAR-10 структурований на 10 дискретних класів об'єктів. До цих класів належать літаки, автомобілі, птахи, коти, собаки, жаби, коні, олені, кораблі та вантажівки. Ця категоризація дозволяє моделям навчатися розрізняти базові

візуальні концепції.

На відміну від цього, CIFAR-100 є значно більш деталізованим, представляючи 100 унікальних класів зображень. Ці класи додатково організовані у 20 суперкласів, що надає можливість проводити класифікацію як на дрібному (fine-grained), так і на грубому (coarse-grained) рівні. Така ієрархічна структура робить CIFAR-100 особливо цінним для дослідження моделей, здатних до багатомасштабного розуміння візуальної інформації.

Зображення, що входять до наборів CIFAR, характеризуються високою реалістичністю та походять з різноманітних джерел, охоплюючи широкий спектр типових об'єктів реального світу. Важливою особливістю є наявність трьох колірних каналів (RGB) для кожного зображення, що дозволяє нейронним мережам опрацьовувати повноцінну кольорову інформацію, яка є більш наближеною до реального візуального середовища, ніж чорно-білі зображення.

Ці набори даних були створені з метою надання науковцям та розробникам у сфері глибокого навчання стандартизованого та зручного еталону. Вони слугують для всебічного тестування, калібрування та об'єктивного порівняння продуктивності різних нейромережевих архітектур, особливо згорткових нейронних мереж (CNN), у задачах класифікації зображень. Вимогливість CIFAR до моделей сприяє розробці більш стійких, точних та узагальнювальних алгоритмів комп'ютерного зору.

Переваги CIFAR:

- CIFAR-10 і CIFAR-100 містять реалістичні зображення об’єктів (літаків, автомобілів, птахів, котів, собак, тощо), що набагато ближче до реальних задач класифікації;
- це дозволяє розробляти моделі, які враховують колірну інформацію, що є важливою для реальних додатків;
- кожен клас має однакову кількість зображень, що дозволяє уникнути перекосу при навчанні;

- набір є стандартом для порівняння ефективності різних архітектур нейромереж, особливо згорткових мереж (CNN).

Недоліки CIFAR:

- 32×32 пікселі – дуже низька роздільна здатність. Через це об'єкти на зображеннях мають мало деталей, що ускладнює точну класифікацію навіть для людини;
- через обрізані зображення та фіксований розмір, об’єкти іноді виглядають не повністю, або мають незвичний ракурс;
- у порівнянні з великими наборами даних, як-от ImageNet, CIFAR охоплює обмежену кількість об'єктів та сценаріїв.

1.3.3 ImageNet

ImageNet – це один з наймасштабніших і найвпливовіших наборів даних у сфері комп’ютерного зору, що значною мірою сприяв стрімкому розвитку глибокого навчання, особливо у задачах класифікації, детекції та локалізації об’єктів на зображеннях. Проєкт ImageNet був ініційований у 2009 році дослідниками зі Стенфордського та Принстонського університетів, і з того часу він став ключовим ресурсом для досліджень у галузі машинного навчання.

ImageNet містить понад 14 мільйонів зображень, які вручну анотовані відповідно до ієрархії понять із структури WordNet. Більшість зображень мають кілька класів, які відповідають об’єктам у кадрі, а багато з них також мають мітки з координатами об’єктів (bounding boxes). Одним із найвідоміших піднаборів є ILSVRC (ImageNet Large Scale Visual Recognition Challenge), який включає 1 000 категорій об’єктів та понад 1,2 мільйона навчальних зображень. Саме на цьому піднаборі проходили щорічні змагання з розпізнавання зображень, які відіграли важливу роль у перевороті галузі після появи глибоких згорткових мереж

(зокрема, після перемоги архітектури AlexNet у 2012 році).

ImageNet є критично важливим еталоном для порівняння різних моделей, оскільки він містить велику кількість різноманітних зображень реального світу, складних за змістом, варіативністю ракурсів, освітленням, фоном і перекриттям об’єктів. Це дозволяє проводити глибоку оцінку якості нейромережевих рішень у реалістичних умовах.

Переваги ImageNet:

- понад 14 мільйонів зображень, серед яких понад 1,2 мільйона мають точні мітки категорій. Це дозволяє навчати дуже глибокі моделі, які здатні узагальнювати з великих обсягів інформації;
- набір охоплює широкий спектр категорій (тварини, техніка, предмети побуту, сцени тощо), що робить моделі, навчені на ImageNet, добре пристосованими до вирішення багатьох задач за межами набору;
- піднабір ILSVRC став загальноприйнятим еталоном для порівняння моделей, що забезпечує репрезентативну оцінку їх ефективності;
- ієрархічна структура міток (на основі WordNet) дозволяє більш гнучко організовувати та аналізувати класи, а також будувати багаторівневі класифікатори.

Недоліки ImageNet:

- через великий обсяг зображень та велику кількість класів навчання моделей потребує значних обчислювальних потужностей і часу;
- набір даних переважно орієнтований на класифікацію окремих об’єктів, і тому не завжди підходить для спеціалізованих задач, таких як медична діагностика, супутникові зображення чи виявлення аномалій;
- набір формувався з відкритих джерел в Інтернеті, що могло призвести

до дисбалансу в представленні культур, предметів або умов зйомки;

- хоча частина міток вручну перевірена, все ж деякі зображення мають помилкові або надто узагальнені мітки, що може впливати на якість навчання.

1.4 Аналіз ключових параметрів архітектур нейронних мереж для розпізнавання зображень

При розробці та застосуванні нейромережевих архітектур для задач розпізнавання зображень критично важливим є розуміння та оптимізація їхніх внутрішніх параметрів. Ці параметри безпосередньо впливають на здатність моделі до навчання, її узагальнювальну спроможність та обчислювальну ефективність. Глибина архітектури, представництво її ширини, розподіл кількості параметрів та тип активаційних функцій є фундаментальними характеристиками, які детермінують функціональність та продуктивність нейронної мережі.

1.4.1 Глибина мережі

Глибина нейронної мережі є одним із фундаментальних архітектурних параметрів, що визначається кількістю послідовних шарів, через які дані проходять від вхідного до вихідного рівня. Ця характеристика безпосередньо впливає на здатність моделі до вивчення та представлення складних ієрархічних ознак у вхідних даних.

На початкових етапах обробки, в перших шарах глибокої мережі, відбувається вилучення простих, низькорівневих ознак. Це можуть бути елементарні візуальні патерни, такі як краї, кути або базові градієнти в зображеннях. Ці шари функціонують як своєрідні "детектори" базових структур, які присутні майже в будь-якому візуальному об'єкті.

Проте, в міру просування даних крізь подальші, глибші шари мережі, відбувається поступове формування більш абстрактних і складних представлень. Інформація з попередніх шарів комбінується та трансформується, дозволяючи мережі виявляти такі елементи, як текстури, складніші форми або навіть цілі об'єкти. Кожен наступний шар будує своє розуміння світу на основі ознак, витягнутих попереднім шаром, створюючи багатошарову ієрархію абстракцій. Ця здатність до ієрархічного вивчення ознак є критично важливою для ефективного розпізнавання складних візуальних сцен та об'єктів.

Однак, зі збільшенням глибини мережі виникають певні виклики. Глибокі архітектури, незважаючи на їхню потенційну потужність, складніше навчати та оптимізувати. Вони схильні до проблем, таких як зникаючий або вибуховий градієнт, що ускладнює ефективне поширення помилок у процесі зворотного поширення. Крім того, функціонування та навчання таких мереж вимагає значно більше обчислювальних ресурсів, зокрема пам'яті та процесорного часу, що обмежує їхнє застосування на пристроях з обмеженими можливостями.

Переваги збільшення глибини:

- більш глибокі моделі можуть вивчати складні багаторівневі патерни, що підвищує точність класифікації;
- глибока мережа здатна виділити високорівневі абстракції, корисні для розпізнавання різноманітних об’єктів.

Недоліки збільшення глибини:

- зі збільшенням числа шарів зростає ризик “згасання градієнту” під час навчання;
- потрібні значні обчислювальні ресурси: час тренування, пам’ять GPU, енергія;
- глибокі мережі складніше налагоджувати та підбирати

гіперпараметри.

1.4.2 Ширина мережі

Ширина нейронної мережі є ключовим архітектурним параметром, який визначається кількістю нейронів у кожному прихованому шарі або, у контексті згорткових нейронних мереж, кількістю фільтрів (каналів), що застосовуються у згортковому шарі. Ця характеристика безпосередньо впливає на ємність моделі, тобто її здатність до вивчення та представлення складних функціональних залежностей у даних. Збільшення ширини дозволяє мережі оперувати з більшою кількістю внутрішніх представлень та вивчати більш різноманітні та численні ознаки на кожному рівні абстракції.

Широкі мережі потенційно здатні захоплювати більше інформації та формувати багатші внутрішні представлення даних. Кожен додатковий нейрон або фільтр у шарі розширює "простір" ознак, які можуть бути виявлені та трансформовані. Це особливо важливо для завдань з високою варіативністю вхідних даних або з великою кількістю релевантних, але тонких деталей, які необхідно розрізняти. Наприклад, у згорткових мережах більша кількість фільтрів у шарі дозволяє вивчати ширший спектр візуальних патернів, таких як різні орієнтації країв, текстури або локальні кольорові комбінації, що може покращити розпізнавання об'єктів у складних сценах.

Однак, як і у випадку з глибиною мережі, надмірне розширення її ширини несе в собі певні наслідності. Збільшення кількості нейронів або фільтрів призводить до значного збільшення кількості параметрів, які мережа повинна оптимізувати під час навчання. Це, своєю чергою, вимагає більших обсягів навчальних даних для адекватного навчання та уникнення перенавчання (overfitting), коли модель починає "запам'ятовувати" навчальні приклади замість того, щоб узагальнювати їх. Крім того, більша кількість параметрів прямо пропорційно впливає на інтенсивність обчислювальних ресурсів, необхідних для навчання та інференції (виконання прогнозів), що може стати критичним

фактором для розгортання моделей на пристроях з обмеженими можливостями або у сценаріях реального часу.

Переваги ширини мережі:

- більша кількість нейронів/фільтрів дозволяє мережі вивчати більш складні та численні ознаки;
- дозволяє моделі формувати багатші та різноманітніші внутрішні представлення вхідних даних;
- у задачах з високою варіативністю вхідних даних або з великою кількістю дрібних деталей широкі мережі можуть краще виявляти ці нюанси;
- у певних випадках збільшення ширини може призвести до покращення метрик продуктивності, таких як точність класифікації, якщо модель має достатньо даних для навчання.

Недоліки ширини мережі:

- збільшення кількості параметрів веде до зростання загальної кількості ваг та зміщень, які необхідно оптимізувати;
- моделі з надмірною кількістю параметрів більш схильні до перенавчання на навчальних даних, що знижує їхню узагальнювальну здатність на невидимих даних;
- вимагає значно більше пам'яті та процесорного часу як під час навчання, так і під час інференції;
- оптимізація моделі з великою кількістю параметрів може займати більше часу, оскільки простір пошуку оптимальних ваг значно зростає;

− збільшення ширини може ускладнити інтерпретацію внутрішніх механізмів роботи мережі через зростання складності взаємозв'язків.

1.4.3 Типи шарів

У нейронних мережах, призначених для обробки та аналізу візуальної інформації, різні типи шарів виконують специфічні та взаємодоповнюючі функції, що дозволяє системі послідовно вилучати та обробляти дедалі складніші ознаки із зображень. Кожен шар є невід'ємною частиною архітектури, сприяючи глибокому розумінню візуального контенту.

Згорткові шари (Convolutional layers) є фундаментальним елементом у більшості сучасних архітектур комп'ютерного зору, що по праву вважаються їхньою основою. Їхня ключова функція полягає у застосуванні набору спеціальних фільтрів, які здатні автоматично вивчати локальні патерни у вхідному зображенні. Це можуть бути базові візуальні елементи, такі як чіткі краї, різноманітні текстури або градієнти, що формують первинне представлення зображення. Завдяки цій операції, згорткові шари ефективно виявляють ознаки, які є важливими для подальшого аналізу.

Після згорткових шарів часто застосовуються шари пулінгу (Pooling layers). Їхнє основне призначення – це зменшення просторової розмірності карт ознак, які були згенеровані попередніми шарами. Це досягається шляхом агрегації інформації в певних регіонах, наприклад, через операції max-pooling (вибір максимального значення) або average-pooling (обчислення середнього значення). Таке зменшення розмірності має подвійну перевагу: воно значно підвищує інваріантність моделі до невеликих трансформацій (як-от незначні зсуви чи повороти об'єктів на зображенні), роблячи мережу більш стійкою, а також суттєво знижує обчислювальне навантаження, що є критично важливим для глибоких архітектур.

У контексті аналізу послідовних даних, таких як відеопотоки або серії зображень, рекурентні шари (Recurrent layers) відіграють важливу роль, переважно

знаходячи своє застосування у відеоаналітиці. Їхня унікальна особливість полягає у здатності зберігати контекст або "пам'ять" про попередні кадри чи часові патчі. Це дозволяє мережі розуміти динаміку подій та взаємозв'язки між послідовними візуальними даними, що є незамінним для задач, де часова залежність має вирішальне значення.

Шари уваги (Attention layers) є відносно новим, але надзвичайно потужним компонентом, що набув особливої актуальності у трансформерних архітектурах. Ці шари дозволяють мережі динамічно фокусуватися на найважливіших регіонах зображення або на найрелевантніших елементах у послідовності. Механізм уваги обчислює вагові коефіцієнти для різних частин вхідних даних, даючи моделі можливість "зважувати" їхню важливість. Це є ключовим компонентом для моделювання глобальних залежностей, оскільки мережа може ефективно пов'язувати віддалені елементи вхідної інформації, що значно покращує розуміння контексту та ефективність розпізнавання.

1.4.4 З'єднання між шарами

У архітектурі нейронних мереж, особливо тих, що застосовуються для розпізнавання зображень, спосіб, у який окремі шари об'єднуються та взаємодіють між собою, відіграє критично важливу роль. Цей аспект архітектурного дизайну суттєво впливає на швидкість збіжності процесу навчання та на здатність моделей ефективно вивчати складні залежності у вхідних даних, визначаючи загальну продуктивність та стійкість системи.

Послідовні мережі представляють собою найбазовішу та інтуїтивно зрозумілу конфігурацію, де кожен шар з'єднаний виключно з попереднім у лінійному, строго послідовному порядку. Ця структура характеризується надзвичайною простотою реалізації та розуміння. Проте, при значному збільшенні глибини таких мереж, вони стають схильними до добре відомих проблем, таких як затухання або вибух градієнту. Ці явища значно ускладнюють або унеможливлюють ефективне навчання глибоких моделей, оскільки інформація

про помилку не може належним чином поширюватися по всій мережі.

На противагу лінійній структурі, концепція паралельних гілок передбачає, що модель містить кілька незалежних шляхів обробки даних, які функціонують одночасно. Цей підхід, яскравим прикладом якого є архітектура Inception, дозволяє нейронній мережі одночасно вивчати патерни та ознаки різних масштабів або з використанням різних фільтрів. Кожна гілка може бути оптимізована для вилучення певного типу інформації, а їхні виходи згодом об'єднуються, що збагачує загальне представлення даних та підвищує адаптивність моделі до різноманітних візуальних варіацій.

Механізм залишкових з'єднань (residual connections) являє собою значний прорив у дизайні глибоких нейронних мереж. Ця інновація полягає у створенні обхідних шляхів, які дозволяють передавати необроблений сигнал з попередніх шарів безпосередньо до глибших шарів, оминаючи один або декілька проміжних обчислювальних блоків. Такий підхід значно полегшує проходження градієнтів через дуже глибоку архітектуру, ефективно долаючи проблеми затухання градієнта та дозволяючи будувати мережі з сотнями шарів, які раніше були неможливі для ефективного навчання. Залишкові з'єднання дозволяють моделі легше вивчати зміни від вхідного сигналу, а не повністю нові представлення.

Концепція щільних з'єднань (dense connections) є подальшим розвитком ідеї багаторазового використання ознак. У такій архітектурі кожен шар приймає на вхід не лише вихід безпосередньо попереднього шару, але й виходи всіх попередніх шарів у даному блоці. Це забезпечує надзвичайно ефективний потік інформації по всій мережі, оскільки кожен шар має доступ до сирих та оброблених ознак з усіх попередніх етапів. Такий інтенсивний обмін інформацією зменшує потребу в надмірній кількості фільтрів у кожному окремому шарі, оскільки ознаки можуть бути повторно використані, що призводить до більш компактних та ефективних моделей.

1.4.5 Інші параметри

Окрім фундаментальної архітектурної структури та типів шарів, на загальну продуктивність нейронної мережі у задачах розпізнавання зображень впливає низка додаткових параметрів, які відіграють вирішальну роль у процесі навчання та здатності моделі до узагальнення. Точна конфігурація цих гіперпараметрів є критично важливою для досягнення оптимальних результатів.

Розмір та кількість фільтрів у згорткових шарах є одними з першочергових факторів. Ці параметри безпосередньо визначають ступінь деталізації вилучених ознак. Малі фільтри (наприклад, 3×3 пікселі) зазвичай захоплюють локальні, дрібні патерни, тоді як більші фільтри можуть виявляти більш глобальні та абстрактні ознаки. Кількість фільтрів, своєю чергою, визначає кількість різних патернів, які шар здатен одночасно вивчати. Оптимальний вибір цих розмірів і кількості дозволяє мережі ефективно витягувати релевантну інформацію на різних масштабах.

Функції активації, такі як ReLU (Rectified Linear Unit), Leaky ReLU або Swish, вводять нелінійність у модель, що є абсолютно необхідним для її здатності вивчати складні, нелінійні залежності у даних. Окрім забезпечення нелінійності, вибір конкретної функції активації може суттєво впливати на швидкість збіжності процесу навчання та здатність градієнтів ефективно поширюватися по мережі. Наприклад, ReLU допомагає уникнути проблеми зникаючого градієнта, тоді як Leaky ReLU намагається вирішити проблему "вмираючих" нейронів.

Методи регуляризації, серед яких найпоширенішими є Dropout, Batch Normalization та Stochastic Depth, є невід'ємною частиною сучасного тренувального процесу. Їхня основна мета – запобігти перенавчанню (overfitting) моделі на навчальних даних та стабілізувати процес навчання. Dropout, наприклад, випадково "відключає" частину нейронів під час кожного кроку навчання, змушуючи мережу вчитися більш надійним представленням. Batch Normalization нормалізує входи шарів, що прискорює навчання та робить його більш стійким до ініціалізації параметрів.

Стратегії оптимізації, такі як Adam або SGD (Stochastic Gradient Descent) із моментом, разом з вибором швидкості навчання (learning rate), визначають динаміку зміни ваг нейронної мережі під час ітерацій навчання. Оптимізатор керує тим, як саме ваги оновлюються на основі обчислених градієнтів, а швидкість навчання контролює розмір кроку, який робить оптимізатор у напрямку мінімізації функції втрат. Неправильний вибір цих параметрів може призвести до повільної збіжності, осциляцій або навіть розбіжності процесу навчання.

Таким чином, оптимальний підбір усіх цих параметрів, у поєднанні з правильним та обґрунтованим дизайном архітектури нейронної мережі, є фундаментальним для досягнення високих результатів у складних та вимогливих задачах розпізнавання зображень. Це комплексний процес, що часто вимагає емпіричного налаштування та глибокого розуміння взаємодії між різними компонентами системи.

1.5 Порівняльний аналіз існуючих підходів

Метою цього аналізу є не лише вивчення, а й глибинне виявлення сильних та слабких сторін кожного розглянутого підходу до архітектур нейронних мереж. Це передбачає детальний розгляд їхніх функціональних особливостей, що дозволить всебічно оцінити, де саме проявляються їхні ключові переваги у певних контекстах, і водночас ідентифікувати притаманні їм недоліки та обмеження. Фінальним етапом цього ретельного дослідження є обґрунтування вибору конкретних моделей для подальшого, більш глибокого аналізу. Такий підхід забезпечує системність та цілеспрямованість у дослідженні, гарантуючи, що подальші етапи роботи будуть базуватися на ретельному розумінні фундаментальних характеристик кожної архітектури.

1.5.1 Порівняння нейромережевих моделей

Згорткові нейронні мережі (CNN) традиційно вважаються одними з найбільш ефективних архітектур для завдань, де першочергове значення мають просторові ознаки. Це включає широкий спектр застосувань, таких як класифікація об'єктів, коли необхідно ідентифікувати об'єкти на зображенні, виявлення об'єктів, що передбачає локалізацію та ідентифікацію декількох об'єктів, та сегментація зображень, де кожен піксель зображення класифікується відповідно до об'єкта, до якого він належить. CNN чудово справляються з вилученням ієрархічних ознак із зображень, починаючи від простих елементів, таких як краї та кути, і закінчуючи складними текстурами та формами. Важливою перевагою цих мереж є їхня інваріантність до невеликих зсувів, поворотів та масштабування об'єктів на зображенні, що робить їх стійкими до варіацій у вхідних даних. Однак, CNN можуть бути обчислювально складними при роботі з дуже великими зображеннями через високу кількість параметрів та операцій згортки, а також потребують значних обсягів розмічених даних для ефективного навчання та узагальнення.

Рекурентні нейронні мережі (RNN) є особливо корисними для задач, які вимагають врахування часової динаміки у послідовностях зображень або відео. Вони знаходять своє застосування у таких сферах, як розпізнавання жестів, де послідовність рухів має вирішальне значення, або аналіз відеопотоку, що передбачає розуміння подій, які розгортаються з часом. Основна перевага RNN полягає в їхній здатності обробляти послідовності даних та моделювати залежності між послідовними кадрами, ефективно "пам'ятаючи" попередній контекст. Незважаючи на це, RNN схильні до так званої проблеми зникаючого градієнту, що значно ускладнює навчання глибоких мереж та захоплення довгострокових залежностей. Крім того, у порівнянні з CNN, вони менш ефективні для безпосередньої обробки великих окремих зображень, оскільки їхня архітектура оптимізована під послідовний ввід.

Архітектури Transformer представляють собою одне з найбільш

інноваційних та потужних досягнень у комп'ютерному зорі останнього часу, забезпечуючи найсучасніші результати у багатьох складних задачах. Їхня виняткова продуктивність зумовлена механізмом уваги (attention mechanism), який дозволяє моделі моделювати глобальні залежності між будь-якими частинами зображення, незалежно від їхнього просторового розташування. Це дає Transformer'ам можливість охоплювати контекст цілого зображення, на відміну від локального фокусу CNN. Додатковою значною перевагою є їхня висока паралельність обчислень, що дозволяє ефективно використовувати сучасні обчислювальні ресурси, такі як GPU та TPU, значно прискорюючи процеси навчання та інференції. Однак, незважаючи на їхню потужність, Transformer'и є найбільш обчислювально складними моделями серед розглянутих, вимагаючи значних ресурсів для тренування. Крім того, вони, як правило, потребують найбільших наборів даних для навчання, щоб повністю реалізувати свій потенціал та уникнути перенавчання.

Таблиця 1.1 – Порівняльна характеристик нейронних мереж

Характеристика	**Згорткові нейронні мережі (CNN)**	**Рекурентні нейронні мережі (RNN)**	**Трансформери (Transformers)**
Основний принцип	Згортка, виявлення локальних ознак	Рекурентність, обробка послідовностей	Механізм уваги, паралельна обробка
Типові дані	Зображення, сіткові дані	Послідовності (текст, мова, часові ряди)	Послідовності (текст, мова, часові ряди)
Паралелізація навчання	Висока	Низька (послідовна обробка)	Висока (паралельна обробка)
Довгострокові залежності	Обмежена здатність	Моделювання (з LSTM/GRU), але з обмеженнями	Дуже хороша, незалежно від відстані
Обчислювальна складність	Лінійна за розміром зображення	Лінійна за довжиною послідовності	Квадратична за довжиною послідовності (O(N2))
Основна перевага	Ефективні для комп'ютерного зору	Обробка послідовностей з пам'яттю	Паралелізація, потужне моделювання залежностей
Основний недолік	Слабкі для послідовностей	Повільне навчання, проблема градієнта (у простих RNN)	Високі обчислювальні витрати для дуже довгих послідовностей

1.5.2 Порівняння підходів до навчання

У контексті навчання нейромережевих моделей для розпізнавання зображень, існують три основні парадигми навчання, кожна з яких застосовується залежно від наявності та типу доступних даних.

Навчання з учителем (Supervised Learning) є домінуючим та найбільш поширеним підходом у сфері розпізнавання зображень. Його ефективність базується на доступності розмічених даних, де кожному вхідному зображенню однозначно відповідає правильна мітка або бажаний вихід. Це може бути назва об'єкта, його категорія, або ж координати обмежувальної рамки. Процес навчання моделі полягає у мінімізації функції втрат, яка кількісно відображає різницю між передбаченими виходами моделі та фактичними (істинними) мітками. Завдяки цьому постійному порівнянню та корекції, модель поступово адаптує свої внутрішні параметри, щоб якомога точніше відображати складні залежності між вхідними зображеннями та їхніми відповідними мітками. Це дозволяє їй узагальнювати знання та робити точні прогнози на нових, раніше не бачених даних.

На противагу навчанню з учителем, навчання без учителя (Unsupervised Learning) застосовується для тренування моделей на нерозмічених даних, тобто даних, для яких відсутні явні правильні відповіді чи мітки. Метою цього підходу є надання моделям здатності самостійно виявляти приховані закономірності, структуру та внутрішні зв'язки в даних без будь-яких зовнішніх вказівок. Наприклад, модель може кластеризувати схожі зображення разом або навчитися стискати дані, зберігаючи їхню сутність. Цей підхід виявляється особливо корисним для таких задач, як вилучення ознак, де модель вчиться формувати інформативні представлення вхідних даних, або для зменшення розмірності даних, що допомагає спростити подальшу обробку та аналіз. Він дозволяє використовувати величезні обсяги доступних нерозмічених даних для попереднього навчання моделей, які потім можуть бути донавчані з меншою кількістю розмічених даних.

Навчання з підкріпленням (Reinforcement Learning) є унікальною парадигмою, яка зосереджена на навчанні агентів приймати оптимальні рішення в динамічному середовищі. У цьому підході агент взаємодіє зі своїм оточенням, виконуючи певні дії. За кожну виконану дію агент отримує винагороду (позитивний сигнал) або покарання (негативний сигнал), що слугує зворотним зв'язком. Метою навчання є вироблення такої стратегії поведінки, яка дозволить агенту максимізувати сукупну винагороду протягом тривалого періоду. Цей підхід є надзвичайно ефективним для задач, що вимагають прийняття послідовності рішень, де кожна наступна дія залежить від попередніх станів середовища. Яскравим прикладом застосування навчання з підкріпленням у контексті візуальної інформації є навігація робота, де агент (робот) повинен, спираючись на дані з камер та сенсорів, самостійно навчатися, як переміщуватися у просторі, уникаючи перешкод та досягаючи цілей.

2 Методологія дослідження

2.1 Опис критеріїв вибору моделей для аналізу

Вибір нейромережевих моделей для аналізу в межах даного дослідження здійснювався на основі низки чітко визначених критеріїв. Враховуючи значну різноманітність існуючих архітектур, основною метою є забезпечення репрезентативного охоплення сучасних підходів до розпізнавання зображень, а також можливість об'єктивного порівняння моделей за сукупністю характеристик.

Для забезпечення різноманітності у підходах до обробки візуальної інформації до аналізу включено моделі трьох типів архітектур: згорткові нейронні мережі (CNN), що є класичним та перевіреним підходом до вилучення просторових ознак; трансформери (Vision Transformers) – сучасні архітектури, що використовують механізм уваги та демонструють високу точність на складних задачах; та гібридні моделі або варіації CNN, які поєднують глибоку оптимізацію архітектури з ефективним масштабуванням.

Обрані моделі активно використовуються у наукових дослідженнях і входять до складу найрезультативніших систем у задачах комп'ютерного зору, зокрема в змаганнях ImageNet Large Scale Visual Recognition Challenge (ILSVRC). Такий вибір гарантує, що аналіз охоплює найефективніші та найвивченіші архітектури.

Практичне використання моделей у дослідженні вимагало наявності реалізацій у відкритих бібліотеках (TensorFlow, PyTorch), а також якісної документації, що забезпечує точне відтворення архітектур та коректну інтерпретацію результатів.

Враховувалися показники точності моделей на стандартних наборах даних, таких як ImageNet, CIFAR-10 і MNIST. До аналізу включалися лише ті архітектури, які досягли конкурентоспроможних або найвищих результатів у відповідних тестах.

Обиралися моделі з різним рівнем обчислювальної складності – від легких у використанні ResNet до потужних, але ресурсоємних ViT. Це дозволяє оцінити компроміс між точністю та продуктивністю у контексті практичного застосування.

Особливу увагу приділено архітектурам, які легко масштабуються – як по глибині, так і по ширині, з можливістю адаптації до різних розмірів вхідних зображень і задач. Яскравим прикладом є EfficientNet, в якій масштабування оптимізовано автоматично.

2.2 Опис критеріїв вибору наборів даних для аналізу

Для об'єктивного порівняння нейромережевих моделей розпізнавання зображень вибір відповідних наборів даних є критично важливим. Ці набори повинні забезпечувати достатню кількість зображень для ефективного навчання моделей та представляти різні рівні складності завдань, що дозволить всебічно оцінити продуктивність архітектур на основі вже наявних емпіричних даних.

При виборі наборів даних, чиї результати підлягають аналізу, враховувалася різноманітність складності, що дозволяє всебічно оцінити продуктивність моделей у різних умовах. Перевага надавалася наборам, які широко використовуються в науковій спільноті, що забезпечує можливість прямого порівняння результатів з іншими дослідженнями та релевантність висновків. Наявність попередньо оброблених версій датасетів є важливою для мінімізації впливу людського фактору на опубліковані результати та спрощення їхнього порівняльного аналізу. Висока якість анотацій була пріоритетною для коректного навчання моделей, чиї показники аналізуються, та мінімізації шуму в даних, що є особливо важливим для завдань навчання з учителем, де точність розмітки критично впливає на кінцеву продуктивність. Підтримка наборів даних у таких бібліотеках, як TensorFlow Datasets та PyTorch torchvision, значно спрощує процес їх завантаження та попередньої обробки, що дозволяє ефективно проводити власні

експерименти та забезпечує можливість порівняння результатів з роботами інших дослідників.

Використання наборів даних різного масштабу та змісту дозволяє оцінити здатність моделей до генералізації – збереження високої точності на нових, невідомих даних, що є ключовим показником для практичного застосування.

Для аналізу результатів було обрано три загальновизнані датасети: MNIST, що є простим набором чорно-білих рукописних цифр; CIFAR-10, що представляє набір кольорових зображень середньої складності з десятьма категоріями; та ImageNet, великий набір складних реалістичних зображень із тисячами класів.

Ці обрані датасети є загальноприйнятими еталонами в галузі комп'ютерного зору та широко використовуються для тестування і порівняння більшості сучасних моделей. Це забезпечує релевантність отриманих результатів, доступних у наукових джерелах, та дозволяє зіставити їх з іншими дослідженнями. Для MNIST, CIFAR-10 та ImageNet доступні стандартизовані версії, які включають підготовлені навчальні, валідаційні та тестові множини. Це мінімізує вплив людського фактору на проведені експерименти та спрощує процес порівняльного аналізу опублікованих даних. Усі три набори характеризуються високою якістю анотацій, що дозволяє моделям коректно навчатися та мінімізує шум у даних. Цей аспект є особливо важливим для завдань навчання з учителем, де точність розмітки критично впливає на кінцеву продуктивність моделей, чиї результати підлягають дослідженню. Наявність вбудованої підтримки у таких бібліотеках, як TensorFlow Datasets та PyTorch torchvision, значно полегшує завантаження та попередню обробку даних для дослідників, які проводили первинні експерименти. Такий вибір наборів даних дозволяє реалізувати повноцінний багаторівневий аналіз нейромережевих моделей у контексті завдань розпізнавання зображень, спираючись на всебічну сукупність вже існуючих результатів.

2.3 Опис методів аналізу та порівняння моделей

З метою об'єктивного оцінювання ефективності обраних нейромережевих моделей для розпізнавання зображень застосовується комплексний підхід, що поєднує глибокий структурний аналіз архітектури, всебічний огляд опублікованих емпіричних тестувань продуктивності та ретельне порівняльне узагальнення зібраних даних. Цей підхід дає змогу сформувати обґрунтовані висновки, спираючись на консолідований досвід наукової спільноти.

Методи аналізу охоплюють низку взаємодоповнюючих етапів: детальний аналіз архітектурних характеристик моделей, ревізію доступних результатів емпіричних тестувань на різних наборах даних, оцінку ефективності використання ресурсів на основі звітних даних, порівняльний аналіз усіх отриманих результатів та, нарешті, якісну інтерпретацію виявлених закономірностей.

Аналіз архітектурних характеристик моделей дозволяє всебічно зрозуміти внутрішню організацію кожної архітектури та пояснити причини її поведінки при обробці різних типів зображень. В рамках цього аналізу ретельно розглядаються ключові структурні особливості кожної моделі, включаючи її глибину (тобто кількість шарів), специфічні типи використаних шарів (таких як згорткові, рекурентні шари або шари уваги), наявність залишкових або щільних з'єднань, загальна кількість параметрів та їх обчислювальна складність, що вимірюється в FLOPs, а також обсяг пам'яті, що споживається графічним процесором (GPU memory footprint), і, звичайно, їхня здатність до масштабування.

Опубліковані дані тестування моделей аналізуються на трьох рівнях складності наборів даних: MNIST, CIFAR-10 та ImageNet. Це дозволяє порівнювати моделі не лише за показниками точності, але й за зафіксованими даними щодо швидкості та надійності їх роботи. Для кожного набору даних аналізуються такі метрики, як точність класифікації (accuracy), що відображає частку правильно класифікованих зображень, а також показники Top-1 / Top-5 accuracy, які вказують на частоту, з якою правильна відповідь є першою або

входить до перших п’яти прогнозів моделі. Крім того, оцінюються зафіксовані в джерелах час навчання та час інференсу (для inference-батчів фіксованого розміру), а також досліджується стабільність результатів у різних умовах, таких як вплив шуму, перекручень або зміни освітлення.

Зокрема, fine-tuning моделей здійснюється в режимі повного перенавчання без заморожування шарів, що дозволяє оцінити початкову здатність архітектур швидко адаптуватися до нових задач на обмежених датасетах, таких як CIFAR-10 та MNIST. Навчання проводиться протягом однієї епохи, що забезпечує уніфікацію експериментальних умов і знижує вплив тривалості тренування на результати.

Такий підхід демонструє ефективність початкової адаптації моделей, критично важливої для прикладних задач із обмеженими ресурсами та часом, та дозволяє об'єктивно порівняти продуктивність різних архітектур за початковий період навчання.

Оцінка ефективності використання ресурсів, що ґрунтується на опублікованих даних, дозволяє зробити висновки щодо практичної доцільності впровадження моделей у реальних системах з обмеженими ресурсами. Для кожної моделі аналізуються такі зафіксовані показники, як середній обсяг оперативної пам’яті, що використовується, середній час інференсу одного зображення (як на CPU, так і на GPU), загальна кількість параметрів та відповідні обчислювальні витрати.

Порівняльний аналіз усіх зібраних результатів дозволяє об’єктивно обрати найбільш ефективну модель для конкретного застосування. Отримані дані узагальнюються у вигляді порівняльних таблиць та діаграм, що наочно відображають співвідношення між точністю, споживаними ресурсами та часом роботи моделей. Також формуються зведені переваги та недоліки моделей у контексті різних завдань, і, за необхідності, складається їх рейтинг за ключовими метриками.

Додатково, за допомогою якісної інтерпретації результатів здійснюється аналіз відповідності між теоретичними очікуваннями, що ґрунтуються на

архітектурі моделі, та емпіричними результатами, отриманими іншими дослідниками. Це дозволяє з’ясувати глибинні причини, чому певна модель показала кращі або гірші результати в конкретних умовах, розкриваючи неочевидні аспекти її функціонування.

Наведені методи формують основу для системного та комплексного аналізу нейромережевих моделей, що дозволяє зробити обґрунтовані висновки та запропонувати практичні рекомендації для подальшого використання цих архітектур.

2.4 Опис процедури перенавчання моделей (fine-tuning)

У межах експериментальної частини дослідження було реалізовано процедуру перенавчання популярних нейромережевих архітектур на двох датасетах – MNIST та CIFAR-10. Для цього було використано мову програмування Python та бібліотеки torch, torchvision, timm, thop, а також середовище розробки Visual Studio Code. Уся реалізація виконувалась локально на комп’ютері з підтримкою графічного прискорення за допомогою технології CUDA, що значно пришвидшувало процес навчання моделей. Запуск здійснювався в межах ізольованого середовища venv, у якому були встановлені всі необхідні залежності, що забезпечувало стабільну роботу коду без конфліктів версій. На рисунку 2.1 зображено приклад інтерфейсу середовища VS Code з відкритим файлом основного скрипту навчання моделей.

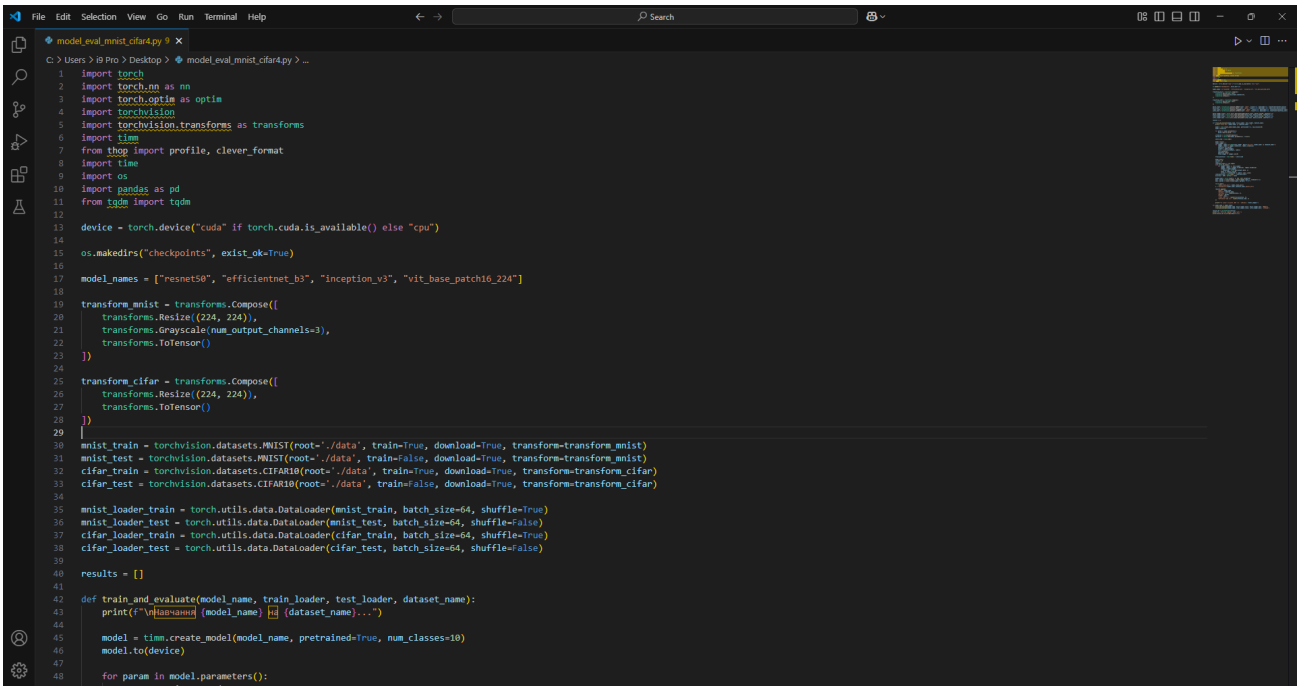


Рисунок 2.1 – Інтерфейс Visual Studio Code з відкритим кодом перенавчання моделей

У процесі реалізації було обрано чотири попередньо натреновані моделі: ResNet-50, EfficientNet-B3, Inception v3 та ViT-B/16. Усі вони були завантажені з бібліотеки timm з параметром pretrained=True. Обраний підхід тонкого налаштування (fine-tuning) усієї моделі був зумовлений ефективністю та швидкістю адаптації до нових даних, оскільки навчання моделей з нуля вимагало б значно більших обчислювальних ресурсів і часу, а також великої кількості навчальних зразків. Fine-tuning дозволяє використати вже набуті модельні ознаки під час попереднього навчання на великих датасетах, що суттєво покращує якість і стабільність навчання на відносно невеликих наборах, таких як MNIST та CIFAR-10. Крім того, цей підхід дає змогу порівнювати різні архітектури у зіставних умовах, оцінюючи їх здатність до адаптації без додаткових факторів, пов’язаних із різною швидкістю або якістю початкового навчання.

Навчання кожної моделі відбувалося протягом лише однієї епохи. Цього було достатньо для оцінки базових адаптаційних можливостей архітектури без глибокої оптимізації. В якості функції втрат застосовувалась CrossEntropyLoss, що дозволяло обчислювати відхилення прогнозованих класів від очікуваних.

Оновлення ваг здійснювалось за допомогою оптимізатора Adam зі сталою швидкістю навчання. Запуск скрипта, що наведено у рисунку 2.2, здійснювався через стандартний термінал Windows (cmd), у якому можна було спостерігати повідомлення про перебіг навчання та результати кожної моделі.

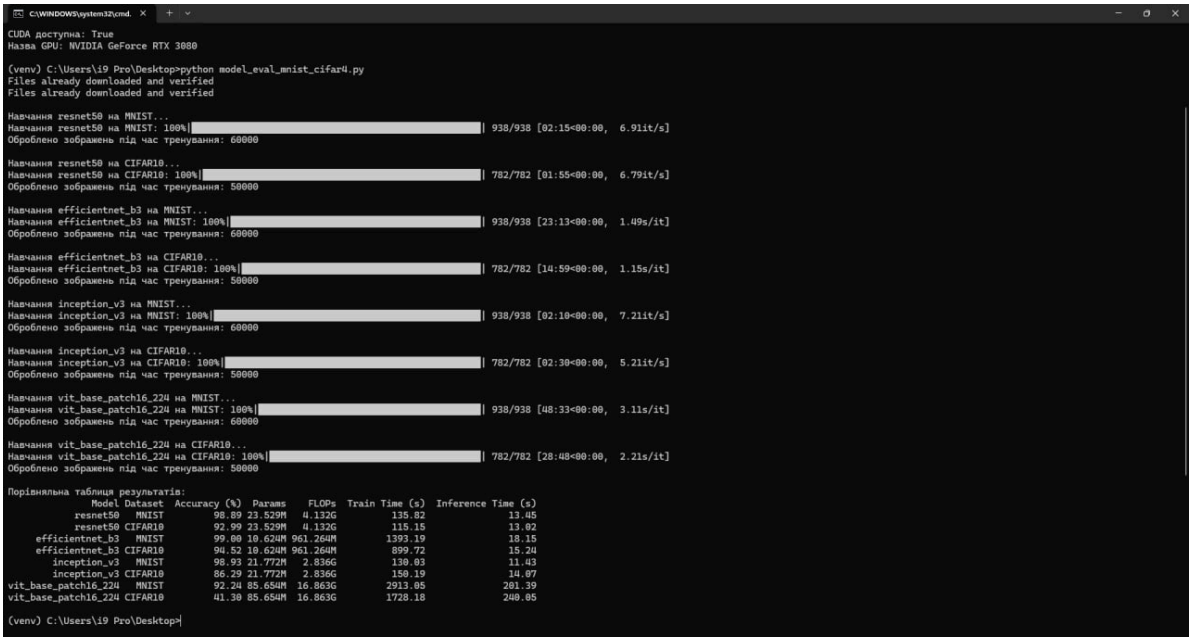


Рисунок 2.2 – Приклад запуску коду перенавчання моделей у терміналі з виводом результатів

Кожна модель тренувалась окремо для кожного з двох датасетів, після чого проводилось тестування та оцінка точності класифікації. Також вимірювався час тренування, час інференсу, кількість параметрів та обчислювальні витрати (FLOPs) за допомогою бібліотеки THoP. Результати зберігались у таблицю та виводились у консоль після завершення навчання.

Моделі зберігались у форматі .pth після кожного запуску в окрему директорію checkpoints. Таким чином, у подальшому їх можна повторно використовувати без необхідності повторного тренування. З метою стандартизації входів усі зображення були змінені до розміру 224x224, що відповідає розміру, очікуваному попередньо натренованими моделями. Для датасету MNIST також застосовувалась конверсія зображень у трьохканальний формат.

Усі етапи експерименту реалізовано максимально уніфіковано, що дало змогу отримати репрезентативні результати порівняння моделей в однакових умовах.

3 Аналіз нейромережевих моделей для розпізнавання зображень

3.1 Детальний аналіз обраних моделей

Методологія дослідження базуватиметься на кількох взаємопов'язаних підходах, що дозволять отримати повну картину продуктивності та архітектурних особливостей моделей.

По-перше, буде проведено аналіз архітектурних характеристик кожної моделі. Це включатиме детальний розгляд таких аспектів, як глибина мережі (кількість шарів), типи використаних шарів (наприклад, згорткові, рекурентні, шари уваги), наявність специфічних з'єднань (залишкових або щільних), загальна кількість параметрів, обчислювальна складність (виміряна в FLOPs), обсяг пам'яті, що споживається графічним процесором (GPU memory footprint), а також здатність моделі до масштабування. Розуміння цих структурних особливостей дозволить пояснити принципи роботи моделей та інтерпретувати опубліковані дані про їхню продуктивність.

По-друге, буде здійснено аналіз опублікованих емпіричних даних щодо продуктивності моделей на різних наборах даних, що представляють різні рівні складності. Ці дані, як правило, були отримані в результаті тестування на трьох визначених датасетах: MNIST (простий набір рукописних цифр), CIFAR-10 (середньої складності набір кольорових зображень) та ImageNet (великий набір складних реалістичних зображень). Для кожного набору даних будуть аналізуватися та порівнюватися вже доступні ключові метрики, зокрема: точність класифікації (accuracy), що відображає частку правильно розпізнаних зображень; Top-1 та Top-5 accuracy, які вказують на частоту, з якою правильна відповідь знаходиться на першій позиції або входить до п'яти найкращих варіантів; час навчання моделі; час інференсу (прогнозування) для фіксованих батчів; а також стабільність результатів, оцінена впливом шуму, перекручень та змін освітлення.

Дані для цього аналізу будуть отримані з рецензованих наукових публікацій та загальнодоступних бенчмарків.

По-третє, важливим етапом стане оцінка ефективності використання ресурсів на основі вже існуючих даних. Цей аналіз є критичним для розуміння практичної доцільності впровадження моделей у реальні системи з обмеженими обчислювальними можливостями. Будуть проаналізовані опубліковані показники, такі як середній обсяг оперативної пам'яті, що використовується моделлю, середній час інференсу одного зображення на центральному процесорі (CPU) та графічному процесорі (GPU), а також загальна кількість параметрів та обчислювальні витрати.

Нарешті, буде проведено порівняльний аналіз консолідованих результатів та їх якісна інтерпретація. Зібрані дані будуть узагальнені у вигляді порівняльних таблиць та діаграм, що візуалізують співвідношення між точністю, споживаними ресурсами та часом роботи моделей. Будуть виділені зведені переваги та недоліки кожної моделі в контексті різних завдань розпізнавання, а також сформований рейтинг моделей за ключовими метриками. Якісна інтерпретація дозволить встановити відповідність між теоретичними очікуваннями, що базуються на архітектурі моделі, та емпіричними результатами, з'ясувавши причини, за якими певна модель показала кращі або гірші результати в конкретних умовах.

Застосування вищезазначених методів створить основу для системного та глибокого аналізу нейромережевих моделей шляхом агрегації та інтерпретації вже існуючих даних, що дозволить зробити обґрунтовані висновки та запропонувати практичні рекомендації щодо їх подальшого використання.

3.1.1 ResNet (Residual Network)

ResNet представляють собою визначний клас згорткових нейронних мереж, фундаментальною інновацією якого є запровадження механізму залишкових блоків (residual blocks). Цей архітектурний елемент, який можна побачити на рисунку 3.1, відіграв ключову роль у вирішенні однієї з найскладніших проблем

глибокого навчання, а саме – ефективного тренування надзвичайно глибоких моделей.

Кожен залишковий блок реалізує функцію, що дозволяє передавати вхідні ознаки безпосередньо до виходу блоку, паралельно з їхнім перетворенням через традиційні шари. Формально це виражається як

$$y=F(x,\{Wi\})+x, \tag{3.1}$$

де x позначає вхідні ознаки для блоку, F є непаралельною гілкою, яка здійснює низку трансформацій (часто згорткових), а додавання x до виходу гілки F формує так зване "залишкове з'єднання". Цей простий, але потужний механізм дозволяє мережі ефективно навчатися як ідентичній функції (тобто просто передавати вхідні дані), так і залишковим відображенням, що представляють відхилення від ідентичності. Вплив цього додавання x є критично важливим для запобігання згасанню градієнтів – явища, при якому градієнти стають надто малими під час поширення назад через багато шарів, унеможливлюючи ефективне оновлення ваг у початкових шарах.

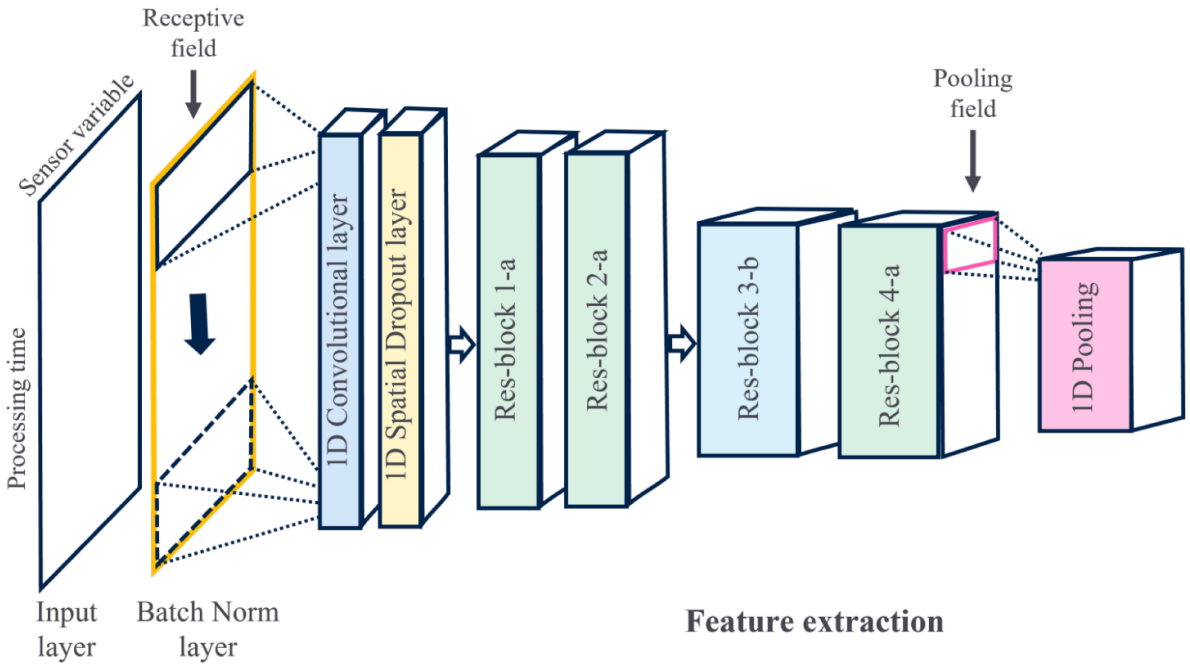


Рисунок 3.1 – Архітектура ResNet

ResNet розроблено в декількох конфігураціях, що відрізняються глибиною мережі, надаючи гнучкість у виборі моделі залежно від складності завдання та доступних обчислювальних ресурсів. Серед найпоширеніших варіантів – ResNet-50, ResNet-101 та ResNet-152, які, відповідно, включають 50, 101 та 152 шари. Збільшення глибини мережі зазвичай корелює зі здатністю вивчати більш складні та абстрактні ознаки, що є особливо цінним для великомасштабних завдань розпізнавання зображень.

Основний механізм навчання ResNet полягає саме в усуненні проблеми зникаючого градієнта. Завдяки реалізації залишкових з'єднань, інформація про градієнти може передаватися через значну кількість шарів без їхнього суттєвого зменшення. Це забезпечує стабільність процесу тренування навіть для надзвичайно глибоких архітектур, що раніше було значною перешкодою. Таким чином, ResNet дозволяє ефективно збільшувати глибину мережі, розширюючи її здатність до вивчення все більш комплексних ознак, одночасно зберігаючи високу стабільність та ефективність тренування.

Таблиця 3.1 – Продуктивність та ресурсоємність Residual Network

Модель	**Кількість шарів**	**Параметри (млн)**	**FLOPs (G) @224×224**	**Top-1 точність (%)**	**Top-5 точність (%)**	**Інференс (ms/зобр., GPU)**
ResNet-50	50	25.6	~4.1	~76.0	~93.0	~10
ResNet-101	101	44.5	~7.6	~77.8	~94.0	~14
ResNet-152	152	60.2	~11.3	~78.5	~94.1	~20

Переваги:

- можливість глибокого навчання без деградації точності;
- сильна інваріантність до масштабування та зсувів об’єктів;
- широко підтримується у фреймворках, добре документована.

Недоліки:

- потребує значних обчислювальних ресурсів (GPU-пам’ять, обчислювальні одиниці);
- може бути надмірно великою для застосунків з обмеженою пам’яттю чи енергією;
- для задач з дуже високою роздільністю зображень класичний ResNet потребує модифікацій (наприклад, зміна стратегії пулингу або фрагментації).

3.1.2 Inception

Архітектура Inception, що зображена на рисунку 3.2, базується на новаторській ідеї, що дозволяє нейронній мережі ефективно обробляти візуальні ознаки на різних масштабах одночасно, не призводячи при цьому до значного зростання обчислювальної складності. Цей принцип реалізовано шляхом паралельного застосування фільтрів різного розміру, таких як 1×1, 3×3, 5×5, а також пулінгу, в межах одного унікального "Inception-блоку". Це інтелектуальне поєднання дозволяє мережі динамічно обирати оптимальні розміри фільтрів для вилучення найбільш релевантних ознак з вхідних даних.

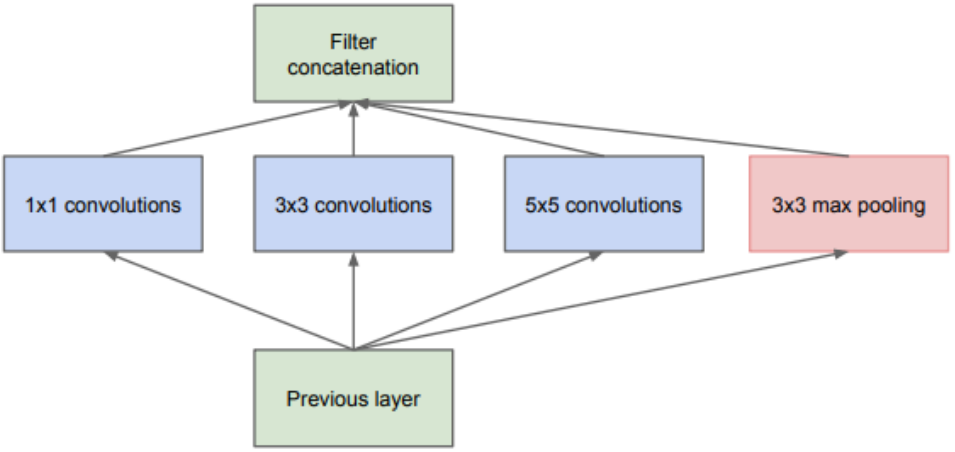


Рисунок 3.2 – Базовий варіант архітектури Inception

На концептуальному рівні, Inception-блок складається з чотирьох паралельних гілок, кожна з яких виконує свою специфічну функцію обробки інформації. Перша гілка зазвичай використовує 1×1 згортки, що є ефективним методом для зменшення розмірності каналів і контролю обчислювальних витрат. Друга та третя гілки застосовують послідовність 1×1 згорток, за якими слідують більші фільтри – 3×3 та 5×5 згортки відповідно. Цей підхід дозволяє захоплювати ознаки різного просторового масштабу. Четверта гілка включає 3×3 пулінг, після якого також застосовуються 1×1 згортки для трансформації каналів. Після обробки в усіх паралельних гілках, їхні вихідні результати конкатенуються (об'єднуються) по канальному виміру, створюючи багатомасштабне представлення ознак, готове для подальшої обробки в наступних шарах мережі.

Еволюція архітектури Inception пройшла кілька ключових етапів, кожен з яких привносив значні вдосконалення. Першою реалізацією стала GoogleNet, або Inception v1, представлена у 2014 році. Ця версія, що містила 22 шари (без урахування пулінгу), вже тоді демонструвала вражаючі результати, досягнувши 68% Top-1 та 88% Top-5 точності на широко відомому датасеті ImageNet, при цьому оперуючи відносно скромною кількістю параметрів – близько 6.8 мільйона.

Наступним значним кроком стала Inception v3, випущена у 2015 році. Ця версія внесла низку ключових вдосконалень, спрямованих на оптимізацію обчислювальної ефективності та підвищення точності. Серед найважливіших змін варто виділити факторизацію великих згорток, зокрема заміну однієї 5×5 згортки на дві послідовні 3×3 згортки, що дозволило суттєво зменшити кількість операцій з плаваючою комою (FLOPs). Також було впроваджено використання асиметричних згорток, таких як 1×7 та 7×1, для певних важливих шарів, що сприяло ефективнішому вилученню ознак. Критичним елементом стало повсюдне застосування Batch Normalization після кожного шару, що значно прискорило та стабілізувало процес навчання мережі. В результаті Inception v3 досягла приблизно 78% Top-1 та 94% Top-5 точності, при цьому кількість параметрів зросла до близько 24 мільйонів, а обчислювальні витрати становили приблизно 5.7 GFLOPs.

Подальший розвиток призвів до появи Inception v4 та Inception-ResNet у 2016 році. Ці архітектури представляли собою інтеграцію ідей залишкових з'єднань, що були успішно застосовані в мережах ResNet, безпосередньо в Inception-модулі (звідси й назва Inception-ResNet). Була покращена загальна архітектура, що стосувалася кращої збалансованості паралельних гілок та додавання спеціальних шляхів для градієнтів, що сприяло більш стабільному та ефективному навчанню дуже глибоких мереж. Ці версії демонстрували ще вищі показники продуктивності, досягаючи приблизно 80% Top-1 та 95% Top-5 точності, з кількістю параметрів близько 42 мільйонів та обчислювальними витратами близько 12 GFLOPs. Ця еволюція ілюструє постійний пошук балансу між складністю архітектури, її обчислювальною ефективністю та, звичайно, результуючою точністю розпізнавання.

Таблиця 3.2 – Продуктивність та ресурсоємність Inception

Модель	**Кількість шарів**	**Параметри (млн)**	**FLOPs (G) @224×224**	**Top-1 точність (%)**	**Top-5 точність (%)**	**Інференс (ms/зобр., GPU)**
GoogleNet	~22	~6.8	~1.5	~68.0	~88.0	~7
Inception v3	~48	~24	~5.7	~78.0	~94.0	~15
Inception v4	~75	~42	~12.0	~80.0	~95.0	~25

Переваги:

- висока точність при помірному зростанні обчислювальних витрат (порівняно з глибокими ResNet);
- гнучкість щодо масштабу ознак завдяки паралельним фільтрам різних розмірів;
- покращена стабільність навчання завдяки Batch Normalization.

Недоліки:

- архітектура стає складнішою з кожною версією, що ускладнює інтерпретацію;
- висока складність обчислень у останніх версіях (v4) робить їх менш придатними для обмежених пристроїв;
- порівняно велика кількість параметрів у v3/v4 може призводити до потреби в більшій пам’яті.

3.1.3 EfficientNet

Архітектура EfficientNet формалізує підхід до оптимізації нейромереж шляхом впровадження принципу складеного масштабування (Compound Scaling). Замість ізольованого збільшення окремих параметрів, таких як глибина або ширина, ця модель масштабується у трьох напрямках одночасно – глибина, ширина та просторовий розмір вхідних даних. Такий підхід дозволяє досягти більшої продуктивності без суттєвого зростання обчислювального навантаження.

В основі цієї концепції лежить прагнення до збереження балансу між кількістю шарів (глибиною) та кількістю каналів (шириною) у межах кожного шару. Надмірне збільшення лише глибини призводить до деградації градієнтів і надмірної спеціалізації, тоді як збільшення лише ширини – до надлишкової надмірності та роздутості параметрів. EfficientNet, натомість, використовує єдиний масштабувальний коефіцієнт, який дозволяє одночасно збільшувати глибину і ширину в оптимальних пропорціях, з урахуванням доступного обчислювального бюджету. Це дозволяє уникнути перенасичення або втрати представницької здатності моделі та забезпечує високий рівень узагальнення.

У центрі цієї архітектури лежить ефективна реалізація блоку MBConv, що являє собою модифікований інверсний вузький резидуальний блок із глибинною згорткою. Завдяки своїй будові MBConv забезпечує високу пропускну здатність за обмежених ресурсів, що є критичним чинником при розгортанні моделей у

продуктивному середовищі. Його використання дозволяє суттєво зменшити кількість параметрів і обчислювальні витрати без істотної втрати точності. Така ефективність досягається завдяки поєднанню розширення каналів, глибинної згортки та операції згортки 1×1 із подальшою нелінійністю та нормалізацією, що забезпечує гнучкість трансформації ознак на кожному етапі обробки.

Базова модель EfficientNet-B0, як представник даного підходу, демонструє баланс між складністю архітектури та її продуктивністю. Вона слугує основою для всієї лінійки моделей і закладає фундамент для подальшого масштабування з урахуванням обчислювальних обмежень або вимог до точності. У більш загальному випадку це дозволяє досліднику або інженеру обрати конфігурацію, яка найкраще відповідає контексту завдання, не жертвуючи при цьому узагальнюючою здатністю моделі. Практичне застосування моделей EfficientNet охоплює широкий спектр задач, де постає потреба в обробці візуальної інформації при обмеженому доступі до ресурсів. Зокрема, модель B0 може бути розгорнута на системах реального часу, де критичною є латентність, або у вбудованих пристроях, що мають обмеження щодо енергоспоживання. Завдяки компактності та високій продуктивності, ці моделі виявляються придатними для мобільної аналітики, відеоспостереження, медичного скринінгу з використанням зображень низької якості, а також для адаптивних систем, що працюють в умовах змінного або непередбачуваного навантаження. На рисунку 3.3 подано архітектуру EfficientNet-B0 із MBConv як базовими будівельними блоками.

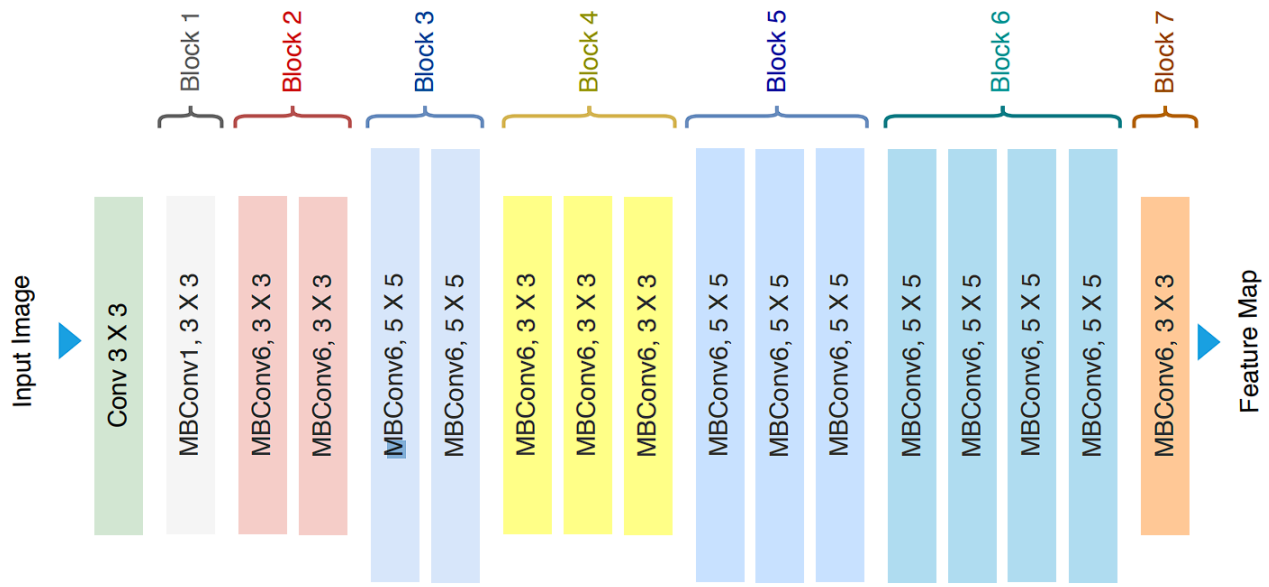


Рисунок 3.3 – Архітектура EfficientNet-B0

Практичне застосування EfficientNet охоплює широкий спектр задач із обробки візуальної інформації, особливо там, де обмежені ресурси. Модель B0, завдяки компактності і високій ефективності, ідеально підходить для систем реального часу з критичними вимогами до латентності, мобільних пристроїв та IoT, а також для медичного скринінгу з низькою роздільною здатністю і відеоспостереження. У той же час, більш масштабні версії серії EfficientNet призначені для серверних середовищ і задач, де пріоритетом є максимальна точність, що досягається при значних обчислювальних витратах.

Поступове збільшення складності моделей від B0 до B7 відображає пропорційне масштабування всіх трьох вимірів – глибини, ширини та роздільної здатності вхідних зображень. Це дозволяє адаптувати моделі до різноманітних обчислювальних середовищ та вимог точності, зберігаючи при цьому єдність архітектури, що спрощує їх інтеграцію в існуючі пайплайни машинного навчання.

У таблиці нижче наведено короткий опис характеристик моделей EfficientNet від B0 до B7, що ілюструє еволюцію параметрів і можливостей кожної конфігурації.

Таблиця 3.3 – Опис характеристик моделей EfficientNet серії B0–B7

Модель	Характеристика
EfficientNet-B0	Компактна модель з невеликою кількістю параметрів; оптимальна для мобільних пристроїв.
EfficientNet-B1	Трохи ширша та глибша, приймає зображення вищої роздільності (240×240).
EfficientNet-B2	Краща точність, при цьому зберігає прийнятну швидкодію (260×260 вхід).
EfficientNet-B3	Збалансована продуктивність: хороша точність і помірні обчислювальні витрати (300×300).
EfficientNet-B4	Помітне покращення точності, підходить для серверних рішень (380×380).
EfficientNet-B5	Висока точність (83.6 % Top-1); вимагає більше пам’яті та ресурсів (456×456).
EfficientNet-B6	Дуже глибока модель з точністю до 84 %; роздільність зображення – 528×528.
EfficientNet-B7	Найпотужніша модель серії; точність до 84.3 %, але й найвищі обчислювальні витрати (600×600).

Вказані розміри вхідних зображень та рівень точності дозволяють оцінити доцільність використання конкретної моделі залежно від умов застосування, включаючи обмеження на обчислювальні ресурси, пам’ять та енергоспоживання.

Спільне масштабування, реалізоване в EfficientNet, забезпечує значно ефективніше використання обчислювальних ресурсів, ніж підходи, які передбачають окреме збільшення лише глибини або лише ширини мережі. Ця оптимізація робить серію EfficientNet надзвичайно універсальною.

Зокрема, моделі EfficientNet-B0 та EfficientNet-B1 є оптимальним вибором для застосувань, де діють суворі обмеження за енергоспоживанням та пам'яттю, таких як мобільні пристрої та системи Інтернету речей (IoT). Їхня компактність та висока ефективність дозволяють інтегрувати їх у широке коло вбудованих систем.

Натомість, більш потужні моделі, такі як EfficientNet-B4–B7, призначені для використання в серверних середовищах або при розробці продуктів, де ключовим пріоритетом є досягнення максимальної точності, і при цьому доступне потужне апаратне забезпечення. Ці конфігурації здатні обробляти складніші дані та забезпечувати найвищий рівень продуктивності в задачах розпізнавання зображень.

Таблиця 3.4 – Продуктивність та ресурсоємність EfficientNet

Модель	Кількість шарів*	Параметри (млн)	FLOPs (G)	Top-1 точність (%)	Top-5 точність (%)	Інференс (ms/зобр., GPU)
EfficientNet-B0	~237	5.3	0.39	77.1	93.3	~8
EfficientNet-B1	~270	7.8	0.70	79.1	94.2	~10
EfficientNet-B2	~309	9.2	1.00	80.1	94.9	~12
EfficientNet-B3	~344	12.0	1.80	81.6	95.3	~15
EfficientNet-B4	~475	19.0	4.20	82.9	95.9	~20
EfficientNet-B5	~597	30.0	9.90	83.6	96.2	~30
EfficientNet-B6	~669	43.0	19.0	84.0	96.4	~40
EfficientNet-B7	~813	66.0	37.0	84.3	96.6	~55

Переваги:

- висока точність при меншій кількості параметрів, ніж у конкурентів (ResNet, Inception);
- можна підібрати модель під будь-які обчислювальні ресурси (від смартфона до серверного кластера);
- енергоефективність для мобільних і вбудованих систем;
- масштабованість (чим більше ресурсів, тим більшу модель можна використати для вищої точності).

Недоліки:

- моделі B4–B7 потребують більшої роздільної здатності вхідних зображень, що збільшує навантаження на пам’ять;
- B6 і B7 вимагають потужних GPU для ефективної роботи й довше навчаються;
- не універсальні для нестандартних задач, наприклад, для послідовних даних або мультимодальних входів (де краще підходять трансформери).

3.1.4 Vision Transformer (ViT)

Vision Transformer (ViT) є революційною архітектурою, яка адаптує принцип роботи традиційного трансформера, що спочатку був розроблений для обробки послідовностей у задачах природної мови, до сфери комп'ютерного зору. Цей підхід кардинально змінює парадигму обробки зображень, відмовляючись від класичних згорткових операцій на користь механізмів уваги.

Фундаментальний принцип роботи ViT полягає в тому, що вхідне зображення спочатку сегментується на серію неперекривних патчів (фрагментів) фіксованого розміру, наприклад, 16×16 пікселів. Кожен з цих патчів потім лінійно перетворюється у вектор фіксованої розмірності. Цей етап є аналогом перетворення слів у токени у завданнях обробки природної мови, дозволяючи моделі сприймати частини зображення як елементи послідовності. Для збереження критично важливої інформації про просторове розташування кожного патча в оригінальному зображенні, до кожного лінійно трансформованого вектора додається позиційне вбудовування (positional embedding). Це дозволяє моделі розрізняти ієрархію та взаємне розміщення елементів зображення, що є життєво важливим для розуміння візуального контексту.

Після етапу лінійного вбудовування патчів, центральним елементом архітектури стає механізм самоуваги (Self-Attention). ViT застосовує декілька шарів цього механізму, що надає моделі унікальну здатність моделювати глобальні залежності та взаємозв'язки між різними частинами зображення, незалежно від їхньої віддаленості. На відміну від згорткових мереж, які обмежені локальним рецептивним полем, самоувага дозволяє кожному патчу "бачити" та інтегрувати інформацію з усіх інших патчів зображення одночасно. Кожен шар уваги обчислює три ключові матриці: запитів (Q), ключів (K) та значень (V). Взаємодія між цими матрицями дозволяє визначити, наскільки сильно кожен патч "уважить" інші патчі в послідовності. Математично цей процес описується формулою:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \quad (3.2)$$

де $\frac{QK^T}{\sqrt{d_k}}$ представляє собою матрицю оцінок сумісності (схожості) між запитами та ключами, яка потім нормалізується за допомогою функції Softmax. Результат множиться на матрицю значень, агрегуючи інформацію з релевантних частин зображення. Цей механізм дозволяє ViT ефективно вивчати складні просторові взаємозв'язки та контекст, що є його ключовою перевагою у порівнянні з традиційними згортковими мережами для певних типів завдань.

Таблиця 3.5 – Продуктивність та ресурсоємність ViT

Модель	**Кількість шарів**	**Параметри (млн)**	**FLOPs (G) @224×224**	**Top-1 точність (%)**	**Top-5 точність (%)**	**Інференс (ms/зобр., GPU)**
ViT-B/16	12	86	~55	~77.9	~93.2	~20
ViT-L/16	24	307	~180	~76.5–79.8 *	~94.0	~45–50

Переваги:

- можливість моделювати глобальні залежності між патчами, що особливо корисно для задач із розпізнаванням складних структур;
- висока паралельність обчислень у кожному шарі, що прискорює тренування на сучасних GPU;
- гнучкість архітектури для додавання нових модулів уваги та гібридних комбінацій з CNN.

Недоліки:

- висока обчислювальна складність (значні FLOPs) і велика кількість параметрів, що потребує великих обсягів даних для тренування;

- чутливість до розміру батчу та довжини послідовності патчів, що може ускладнити адаптацію до дуже великих зображень;
- потреба в масштабному переднавчанні на великих наборах даних (наприклад, JFT-300M або ImageNet21k).

3.2 Порівняльний аналіз моделей на основі визначених критеріїв

У цьому підрозділі представлено узагальнене порівняння архітектур ResNet, Inception, EfficientNet та Vision Transformer з урахуванням не лише формальних метрик, таких як точність, кількість параметрів та обчислювальні витрати, а й практично важливих аспектів: швидкість інференсу, гнучкість до перенавчання, стабільність навчання, підтримка в інструментах та здатність до роботи в умовах обмежених ресурсів. Такий розширений аналіз дозволяє більш об'єктивно оцінити універсальність моделей у прикладних задачах комп’ютерного зору. Проведене дослідження виявило суттєві відмінності між архітектурами не лише за ефективністю, а й за рівнем адаптивності до реальних сценаріїв використання, що дає змогу сформулювати виважені рекомендації щодо їх вибору.

У таблиці 3.6 розглядалися такі аспекти, як: точність, кількість параметрів та обчислювальні витрати. Тести було проведено на базі ImageNet.

Таблиця 3.6 – Продуктивність та ресурсоємність моделей ResNet, Inception, EfficientNet, ViT на ImageNet.

Модель	**Top-1 Acc. (%)**	**Параметри (млн)**	**FLOPs (G)**	**Інференс ≈ ms/зобр.**	**Стійкість***
ResNet-50	76.0	25.6	4.1	10	висока
Inception v3	78.0	24	5.7	15	середня
EfficientNet-B3	81.6	12	1.8	8	висока
ViT-B/16	77.9	86	55	20	низька

У контексті точності класифікації, найвищі показники продемонструвала модель EfficientNet-B3, досягнувши 81.6%. Це значно перевершує результати інших архітектур, зокрема Vision Transformer (77.9%), Inception v3 (78%) та ResNet-50 (76%). Виняткова точність EfficientNet обумовлена ефективністю принципу збалансованого масштабування всіх параметрів моделі, що забезпечує оптимальне співвідношення між глибиною, шириною та роздільною здатністю.

Щодо обчислювальних витрат та кількості параметрів, ViT-B/16 виявився найбільш ресурсно вимогливою архітектурою, демонструючи найбільшу обчислювальну складність з 55 G FLOPs та 86 мільйонами параметрів. На противагу цьому, EfficientNet-B3 демонструє найкращий баланс між високою точністю та низькими обчислювальними витратами, маючи лише 1.8 G FLOPs та 12 мільйонів параметрів. Моделі ResNet-50 та Inception v3 знаходяться в середині спектра за цими показниками, пропонуючи помірні вимоги до ресурсів при конкурентній точності.

Аналіз швидкості інференсу (часу, необхідного для обробки одного зображення) виявив, що найшвидшим є EfficientNet-B3, який виконує інференс приблизно за 8 мілісекунд. За ним слідує ResNet-50 з часом близько 10 мілісекунд, Inception v3 з показником близько 15 мілісекунд, тоді як ViT-B/16 є найповільнішим, витрачаючи приблизно 20 мілісекунд на інференс. Ці дані підкреслюють високу операційну ефективність EfficientNet-B3, що робить його привабливим для застосувань, де критична швидкість обробки.

У аспекті стійкості до шуму та трансформацій, згорткові нейронні мережі (ResNet, EfficientNet) проявили вищу інваріантність до простих зсувів, додавання шуму та змін освітлення. Це пояснюється їхньою внутрішньою архітектурою, яка природно враховує локальні просторові залежності. Натомість, архітектура трансформера (ViT) виявилася більш чутливою до порушень структури патчів, оскільки механізми уваги, що покладаються на глобальні залежності між цими патчами, можуть бути значно порушені при таких трансформаціях.

Порівняльна таблиця 3.7 демонструє продуктивність чотирьох нейромережевих архітектур, попередньо навчених на ImageNet, після їх

донавчання (fine-tuning) на датасетах CIFAR-10 та MNIST. Основні характеристики включають точність класифікації, кількість параметрів, обчислювальну складність (FLOPs), а також час навчання та інференсу. Усі моделі навчалися в режимі повного перенавчання, без заморожування шарів.

Таблиця 3.7 – Оцінка ефективності моделей на спрощених задачах класифікації (CIFAR-10 та MNIST).

Model	Dataset	Accuracy (%)	Params	FLOPs	Train Time (s)	Inference Time (s)
ResNet-50	MNIST	98.89	23.529M	4.132G	135.82	13.45
	CIFAR10	92.99	23.529M	4.132G	115.15	13.02
EfficientNet-B3	MNIST	99.00	10.624M	961.264M	1393.19	18.15
	CIFAR10	94.52	10.624M	961.264M	899.72	15.24
Inception v3	MNIST	98.93	21.772M	2.836G	130.03	11.43
	CIFAR10	86.29	21.772M	2.836G	150.19	14.07
ViT-B/16	MNIST	92.24	85.654M	16.863G	2913.05	201.39
	CIFAR10	41.30	85.654M	16.863G	1728.18	240.05

Навчання здійснювалося протягом лише однієї епохи, що було зроблено свідомо з метою уніфікації експериментальних умов і зменшення залежності результатів від кількості епох або динаміки переобучення. Такий підхід дозволяє краще оцінити початкову здатність архітектур до швидкого пристосування до нової задачі, а також об’єктивно порівняти ефективність fine-tuning на ранньому етапі адаптації. Він також наочно демонструє, які моделі можуть досягти високої продуктивності за мінімального часу навчання – критично важливо для прикладних сценаріїв, де швидкість і ресурси обмежені.

Порівняно з продуктивністю моделей на ImageNet, спостерігається суттєве зниження точності на CIFAR-10 у випадку ViT, що свідчить про його залежність від великих обсягів даних і складніших структур. У той же час ResNet50, EfficientNet-B3 та InceptionV3 демонструють стабільну й високу точність на обох датасетах, особливо на MNIST, де результат перевищує 98%, що відповідає їхній здатності узагальнювати простіші візуальні патерни.

Застосування fine-tuning замість повноцінного навчання з нуля обумовлено кількома критичними факторами. По-перше, ImageNet забезпечує надійно

попередньо навчені представлення, які суттєво пришвидшують та стабілізують процес адаптації до нових, менш масштабних датасетів. По-друге, навчання моделей з нуля вимагає значно більше даних та ресурсів, особливо у випадку трансформерної архітектури, такої як ViT. Fine-tuning дозволяє уникнути перенавчання, ефективно використовує існуючі знання мережі та досягає високої продуктивності за помірного обчислювального навантаження.

Таким чином, fine-tuning виступає доцільним методом адаптації моделей для задач класифікації з обмеженим обсягом навчальних даних, дозволяючи провести об’єктивне та ефективне порівняння різних архітектур при збереженні їхньої вихідної структури та потенціалу, отриманого на базі великого датасету ImageNet.

На основі отриманих даних після тестування моделей, далі буде наведено узагальнювальну таблицю 3.8 з якісною оцінкою ключових характеристик кожної з архітектур.

Таблиця 3.8 – Характеристика ResNet-50, EfficientNet-B3, Inception-v3 та ViT-B/16 за ключовими критеріями прикладної ефективності

Критерій	ResNet-50	EfficientNet-B3	Inception-v3	ViT-B/16
Стабільність при навчанні	Дуже висока	Середня (вимагає тонкого налаштування)	Середня (більш вибаглива архітектура)	Низька (без pretrain нестабільна)
Гнучкість до transfer learning	Відмінна (дуже багато pretrain-застосувань)	Висока, але менше підтримки	Середня (рідше використовується в нових задачах)	Середня (висока лише з великим pretrain)
Підтримка у фреймворках	Абсолютна (PyTorch, TensorFlow, ONNX тощо)	Висока, але менше прикладів	Добра, але дещо застаріла	Обмежена для новачків, висока в дослідницьких
Стійкість до обмежених даних	Висока	Середня	Середня	Низька (потребує багато даних)
Складність інтеграції	Низька (простий у впровадженні)	Середня	Вища (паралельні шари, складні блоки)	Висока (позиційні ембеддинги, patchify тощо)

ResNet-50 демонструє дуже високу стабільність при навчанні, що особливо важливо, якщо модель навчається з нуля або донавчається на складних даних. Вона добре переноситься на нові задачі завдяки широкій доступності попередньо натренованих ваг – це спрощує transfer learning, навіть для нестандартних доменів. На відміну від деяких більш складних моделей, ResNet-50 підтримується практично у всіх фреймворках і бібліотеках, має величезну кількість прикладів коду, що робить її дуже зручною у використанні як для досвідчених розробників, так і для початківців.

Інші архітектури мають свої сильні сторони, але менш універсальні. EfficientNet-B3, наприклад, показує хорошу ефективність на мобільних пристроях і вимагає менше параметрів, проте чутливий до налаштувань, складніший у тренуванні і має менше загальнодоступних прикладів. Inception-v3 вже вважається дещо застарілим, менш популярний у сучасних дослідженнях. ViT-B/16 (візуальний трансформер) теоретично потужний, але нестабільний без великої кількості даних і потребує значних обчислювальних ресурсів, що обмежує його універсальність.

З огляду на всі ці аспекти, ResNet-50 – найзбалансованіший і найуніверсальніший вибір. Вона поєднує стабільність, простоту, підтримку в спільноті та високу якість результатів у найрізноманітніших задачах комп’ютерного зору, від класифікації до сегментації, що й обґрунтовує її вибір як базової моделі для широкого спектра застосувань.

3.3 Аналіз впливу параметрів моделей на ефективність розпізнавання зображень

Глибоке розуміння впливу ключових параметрів нейромережевих моделей, таких як глибина, ширина, роздільна здатність вхідних даних та розмір патчів, є фундаментальним для оптимізації їхньої продуктивності в задачах класифікації

зображень. Зміна кожного з цих вимірів викликає складні взаємодії, що вимагають ретельного аналізу для досягнення бажаного балансу між точністю та обчислювальною ефективністю.

Масштабування глибини мережі, тобто збільшення кількості шарів, є одним з перших і найбільш очевидних шляхів підвищення її здатності до вивчення складних ознак. Наприклад, перехід від архітектури ResNet-50 до ResNet-101 забезпечує приріст Top-1 точності приблизно на 2%. Однак, таке поліпшення супроводжується значним збільшенням ресурсоємності: кількість параметрів зростає приблизно на 19 мільйонів, а обчислювальні витрати, виміряні в FLOPs, збільшуються майже вдвічі. Подальше поглиблення мережі, як у випадку переходу до ResNet-152, додає лише незначний приріст точності, що не перевищує 0.5%, при цьому витрати ресурсів зростають ще приблизно на 50%. З цього випливає, що зростання глибини мережі ефективне лише до певного порога, після якого прибуток у точності стає настільки незначним, що не виправдовує істотне зростання обчислювальної складності та ресурсних вимог.

Ширина мережі, що визначається числом фільтрів у шарі, також відіграє критичну роль. У контексті архітектури EfficientNet, де масштабування ширини інтегровано в загальний принцип Compound Scaling, збільшення ширини (наприклад, від моделі B0 до B3) може дати значний приріст точності, близько 4.5%. Це досягається при відносно помірному збільшенні FLOPs (приблизно до 1.8 G) та кількості параметрів (до 12 мільйонів). Проте, подальше масштабування ширини в більших конфігураціях (B4–B7) призводить до зменшення відносного приросту точності (менше 1%), при цьому значно зростають обчислювальні витрати. Це свідчить про те, що масштабування ширини є ефективним способом поліпшення продуктивності на початкових етапах, але його ефективність знижується при надмірному розширенні.

Роздільна здатність вхідних зображень є ще одним фундаментальним параметром, що впливає на здатність моделі розпізнавати дрібні деталі. У серії EfficientNet-B0–B7 передбачено використання вхідних зображень різної роздільної здатності (від 224×224 до 600×600 пікселів). Підвищення роздільної здатності

позитивно впливає на точність, забезпечуючи приріст приблизно на 1–2% на кожному кроці. Однак, слід враховувати, що збільшення роздільної здатності призводить до пропорційного зростання FLOPs, що відповідає квадрату розміру зображення. У випадку Vision Transformer, збільшення числа патчів (що є еквівалентом зменшення їх розміру) також підвищує точність, але це вимагає значно більше пам'яті та збільшує довжину послідовності, що призводить до зростання обчислень у механізмі самоуваги. Отже, ключовим аспектом при виборі налаштувань моделі для конкретного завдання є пошук оптимального балансу між бажаною роздільною здатністю та доступними обчислювальними ресурсами.

Для архітектур на основі трансформерів, таких як ViT, розмір патчів безпосередньо впливає на деталізацію, яку може сприймати модель. Порівняння ViT-B/16 (патчі 16×16) та ViT-B/32 (патчі 32×32) демонструє, що використання менших патчів (16×16) забезпечує приблизно на 1.5% вищу Top-1 точність. Проте, це досягається ціною подвоєння кількості патчів, що відповідно збільшує обчислювальні витрати механізму уваги. Подальше зменшення розміру патчів, наприклад до 8×8, може ще більше підвищити точність, але при цьому суттєво зростають вимоги до ресурсів, зокрема FLOPs та споживання пам'яті. Це підтверджує, що менші патчі забезпечують кращу деталізацію, але призводять до значно важчих розрахунків.

Крім архітектурних параметрів, налаштування регуляризації та процедур навчання також мають критичний вплив на кінцеві метрики та стійкість моделей. Такі техніки, як Dropout або Stochastic Depth, що застосовуються в ResNet та Inception-ResNet, суттєво покращують узагальнюючу здатність моделей, допомагаючи запобігти перенавчанню, хоча можуть дещо зменшувати початкову швидкість збіжності. Методи аугментації даних, зокрема CutMix та MixUp, значно підвищують стійкість моделей до шуму та також є ефективними засобами боротьби з перенавчанням, особливо на менших датасетах, таких як CIFAR-10 та MNIST. Загалом, ретельне налаштування процедур навчання є невід'ємною частиною оптимізації продуктивності та стабільності моделей.

3.4 Аналіз можливостей застосування кожної моделі для різних задач розпізнавання зображень

Враховуючи особливості архітектур, продуктивність та ресурсні вимоги обраних моделей, було розроблено таблицю їх оптимальних сфер застосування.

Таблиця 3.8 – Сфери застосування моделей

Задача	ResNet	Inception	EfficientNet	ViT
Класифікація об'єктів	Широко застосовується, особливо ResNet-50 для баланс між точністю та швидкістю.	Підходить для складних задач з різномасштабними ознаками.	Ідеально для мобільних застосунків (B0–B1) та серверних (B3).	Потужний для складних сценаріїв із глобальними залежностями.
Виявлення об'єктів (Object Detection)	Використовується як "backbone" в Faster R-CNN, Mask R-CNN.	Інтегрується в SSD та YOLO як екстрактор ознак.	Використовується в lightweight детекторах, наприклад EfficientDet.	Використовується в DetR, Deformable DETR для точного локалювання.
Сегментація зображень	Застосуваний у U-Net-подібних архітектурах та DeepLab.	Підходить для multi-scale сегментації через Inception-модулі.	Використовується у мобільних варіантах сегментаторів (EfficientSeg).	Експериментальні підходи на базі Transformer, наприклад Segmenter.
Аналіз медичних зображень	Широко застосовується для рентгенів, УЗД, КТ (ResNet-101).	Використовується для виявлення меланоми, пухлин.	B0/B1 застосовують у точкових портативних пристроях.	Перспективний для 3D-зображень (глобальні зв'язки між зрізами).
Автономний транспорт	Backbone у perception stack (розпізнавання дорожніх знаків).	Підходить для виявлення різномасштабних об'єктів на дорозі.	B3 для edge-пристроїв у транспорті.	Використовується у дослідженнях для точного усвідомлення сцени.
Аналіз супутникових знімків	Добре працює з великими зображеннями після адаптації.	Корисний завдяки multi-scale Inception-модулю.	Застосовують B1–B3 для моніторингу змін.	Використовують в експериментах з великими патчами для виявлення структур.
Розпізнавання жестів та відео-аналітика	Менш придатний, бо не враховує часові зв'язки.	Аналогічно, надає просторові ознаки.	Використовують у поєднанні з ConvLSTM.	Поєднання з Video Transformer для аналізу послідовностей.

ResNet є чудовим вибором для класичної класифікації, детекції об'єктів та медичної візуалізації. Для більшості прикладних задач оптимальним рішенням буде ResNet-50, оскільки він забезпечує хороший баланс між точністю та швидкістю. Якщо ж ваша задача вимагає підвищеної точності, варто розглянути ResNet-101 або ResNet-152.

Для завдань, що потребують аналізу ознак різних масштабів, наприклад, коли потрібно одночасно розпізнавати великі та дрібні деталі, або для multi-scale сегментації, підійде архітектура Inception. Якщо ви маєте середні ресурси, обирайте Inception v3. Для потужних серверів краще використовувати Inception.

Якщо вам потрібна модель для мобільних та вбудованих систем, або для швидкої класифікації на edge-пристроях, зверніть увагу на EfficientNet. Для обмежених ресурсів ідеально підійдуть EfficientNet-B0 або B1, тоді як B2 та B3 будуть ефективними для задач із середніми ресурсами та високими вимогами до швидкості.

Нарешті, для дослідницьких задач, що працюють з високою роздільною здатністю, глобальним контекстом зображення та складними сценами, варто використовувати Vision Transformer (ViT). Зокрема, ViT-B/16 на серверних GPU рекомендовано для завдань, де критична висока точність і є великі обсяги даних для навчання.

Висновки

У межах проведеного дослідження було здійснено глибокий аналіз чотирьох сучасних архітектур нейронних мереж, що використовуються для задач розпізнавання зображень: ResNet, Inception, EfficientNet та Vision Transformer. Робота охоплювала як порівняння структурних особливостей цих моделей, так і дослідження їхньої практичної ефективності у різних сценаріях.

Особливу увагу було приділено архітектурним характеристикам, які безпосередньо впливають на точність, стабільність і обчислювальні витрати під час навчання та інференсу. Наприклад, ResNet завдяки залишковим зв’язкам дозволяє будувати глибокі моделі без втрати градієнту, демонструючи при цьому високу стабільність і адаптивність. Inception завдяки паралельному аналізу ознак на різних масштабах забезпечує високу гнучкість у задачах, де важливі як локальні, так і глобальні ознаки. EfficientNet реалізує ідею спільного масштабування параметрів моделі, досягаючи відмінного співвідношення між точністю та ресурсною ефективністю. Vision Transformer, у свою чергу, вносить нову парадигму в аналіз зображень, працюючи з глобальними залежностями через механізм самоуваги, однак вимагає значних обсягів даних і ресурсів для ефективного навчання.

Оцінка моделей також враховувала вплив архітектурних параметрів на продуктивність. Було встановлено, що збільшення глибини моделей, наприклад у ResNet або EfficientNet, дає покращення точності до певної межі, після чого приріст стає мінімальним при значному зростанні обчислювальної вартості. Ширина моделей та роздільність вхідних зображень, як показано на прикладі EfficientNet, відіграють ключову роль у досягненні оптимального балансу між точністю та швидкістю. У випадку трансформерів, зокрема ViT, важливим параметром є розмір патчів, який визначає рівень деталізації та ресурсні вимоги.

Зважаючи на результати порівняльного аналізу, модель ResNet-50 було обґрунтовано визначено як найбільш універсальну для широкого спектра завдань.

Вона забезпечує стабільний рівень точності, має помірну обчислювальну складність та добре піддається перенавчанню, навіть при обмежених ресурсах. Її широка підтримка у фреймворках, наявність попередньо навчених ваг і можливість легкої інтеграції в складніші архітектури (наприклад, у ролі backbone) робить її практичним вибором у якості базової моделі для комп’ютерного зору.

Таким чином, результати цього дослідження демонструють, що вибір нейромережевої архітектури має ґрунтуватися на поєднанні таких факторів, як технічні можливості, доступ до обчислювальних ресурсів, тип задачі та вимоги до точності. Надані висновки можуть бути використані як орієнтир для прийняття зважених рішень при впровадженні систем машинного зору в реальні застосунки, де необхідна оптимізація між точністю, швидкодією та складністю реалізації.

Список Використаних джерел

1. He, K., Zhang, X., Ren, S., & Sun, J. Deep Residual Learning for Image Recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016. P. 770–778.
2. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. Going Deeper with Convolutions. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2015. P. 1–9.
3. Tan, M., & Le, Q. V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. Proceedings of the 36th International Conference on Machine Learning (ICML). 2019. P. 6105–6114.
4. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, A., Uszkoreit, J., & Houlsby, N. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. International Conference on Learning Representations (ICLR). 2021. URL: https://arxiv.org/abs/2010.11929 (дата звернення: 2025.03.17).
5. Krizhevsky, A., Sutskever, I., & Hinton, G. E. ImageNet Classification with Deep Convolutional Neural Networks. Advances in Neural Information Processing Systems (NIPS). 2012. Vol. 25. P. 1097–1105.
6. Simonyan, K., & Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. International Conference on Learning Representations (ICLR). 2015. URL: https://arxiv.org/abs/1409.1556 (дата звернення: 2025.04.03).
7. Goodfellow, I., Bengio, Y., & Courville, A. Deep Learning. Cambridge, MA : MIT Press, 2016. 800 p.
8. LeCun, Y., Bengio, Y., & Hinton, G. Deep learning. Nature. 2015. Vol. 521, № 7553. P. 436–444.
9. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. Attention Is All You Need. Advances in Neural

Information Processing Systems (NIPS). 2017. Vol. 30. P. 5998–6008.

10. Ronneberger, O., Fischer, P., & Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI). 2015. P. 234–241.
11. Long, J., Shelhamer, E., & Darrell, T. Fully Convolutional Networks for Semantic Segmentation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2015. P. 3431–3440.
12. 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 (CVPR). 2016. P. 779–788.
13. Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv preprint arXiv:1704.04861. 2017. URL: https://arxiv.org/abs/1704.04861 (дата звернення: 2025.02.28).
14. Chollet, F. Xception: Deep Learning with Depthwise Separable Convolutions. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017. P. 1251–1258.
15. Abadi, M., Barham, P., Chen, J., et al. TensorFlow: A System for Large-Scale Machine Learning. Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16). 2016. P. 265–283.
16. Paszke, A., Gross, S., Massa, F., et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. Advances in Neural Information Processing Systems (NIPS). 2019. Vol. 32. P. 8024–8035.
17. Chollet, F. Keras. 2015. URL: https://keras.io/ (дата звернення: 2025.03.05).
18. Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2009. P. 248–255.
19. Russakovsky, O., Deng, J., Su, H., et al. ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision. 2015. Vol. 115, № 3. P. 211–252.

20. Sutton, R. S., & Barto, A. G. Reinforcement Learning: An Introduction. 2nd ed. Cambridge, MA : MIT Press, 2018. 550 p.
21. Schmidhuber, J. Deep Learning in Neural Networks: An Overview. Neural Networks. 2015. Vol. 61. P. 85–117.
22. Rumelhart, D. E., Hinton, G. E., & Williams, R. J. Learning representations by back-propagating errors. Nature. 1986. Vol. 323, № 6088. P. 533–536.
23. Hochreiter, S., & Schmidhuber, J. Long Short-Term Memory. Neural Computation. 1997. Vol. 9, № 8. P. 1735–1780.
24. Graves, A., & Schmidhuber, J. Framewise phoneme classification with Bidirectional LSTM and other neural networks. Neural Networks. 2005. Vol. 18, № 5-6. P. 602–610.
25. Kingma, D. P., & Ba, J. Adam: A Method for Stochastic Optimization. International Conference on Learning Representations (ICLR). 2015. URL: https://arxiv.org/abs/1412.6980 (дата звернення: 2025.04.22).
26. Zeiler, M. D., & Fergus, R. Visualizing and Understanding Convolutional Networks. European Conference on Computer Vision (ECCV). 2014. P. 818–833.
27. Dalal, N., & Triggs, B. Histograms of Oriented Gradients for Human Detection. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). 2005. Vol. 1. P. 886–893.
28. LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. Backpropagation Applied to Handwritten Zip Code Recognition. Neural Computation. 1989. Vol. 1, № 4. P. 541–551.
29. Viola, P., & Jones, M. Rapid Object Detection using a Boosted Cascade of Simple Features. Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). 2001. Vol. 1. P. I-511–I-518.
30. NVIDIA Developer. NVIDIA Deep Learning SDK Documentation. NVIDIA. URL: https://developer.nvidia.com/deep-learning-documentation (дата звернення: 2025.05.01).

Додаток А

Код програми

```
import torch
import torch.nn as nn
import torch.optim as optim
import torchvision
import torchvision.transforms as transforms
import timm
from thop import profile, clever_format
import time
import os
import pandas as pd
from tqdm import tqdm

device = torch.device("cuda" if torch.cuda.is_available() else "cpu")

os.makedirs("checkpoints", exist_ok=True)

model_names = ["resnet50", "efficientnet_b3", "inception_v3",
"vit_base_patch16_224"]

transform_mnist = transforms.Compose([
    transforms.Resize((224, 224)),
    transforms.Grayscale(num_output_channels=3),
    transforms.ToTensor()
])

transform_cifar = transforms.Compose([
    transforms.Resize((224, 224)),
    transforms.ToTensor()
```

```
])

mnist_train = torchvision.datasets.MNIST(root='./data', train=True, download=True,
transform=transform_mnist)
mnist_test = torchvision.datasets.MNIST(root='./data', train=False, download=True,
transform=transform_mnist)
cifar_train = torchvision.datasets.CIFAR10(root='./data', train=True, download=True,
transform=transform_cifar)
cifar_test = torchvision.datasets.CIFAR10(root='./data', train=False, download=True,
transform=transform_cifar)

mnist_loader_train = torch.utils.data.DataLoader(mnist_train, batch_size=64,
shuffle=True)
mnist_loader_test = torch.utils.data.DataLoader(mnist_test, batch_size=64,
shuffle=False)
cifar_loader_train = torch.utils.data.DataLoader(cifar_train, batch_size=64,
shuffle=True)
cifar_loader_test = torch.utils.data.DataLoader(cifar_test, batch_size=64, shuffle=False)

results = []

def train_and_evaluate(model_name, train_loader, test_loader, dataset_name):
    print(f"\nНавчання {model_name} на {dataset_name}...")

    model = timm.create_model(model_name, pretrained=True, num_classes=10)
    model.to(device)

    for param in model.parameters():
        param.requires_grad = True
```

```
    criterion = nn.CrossEntropyLoss()
    optimizer = optim.Adam(model.parameters(), lr=1e-3)

    start_time = time.time()

    model.train()
    total_images = 0
    for images, labels in tqdm(train_loader, desc=f"Навчання {model_name} на
{dataset_name}"):
        images, labels = images.to(device), labels.to(device)
        optimizer.zero_grad()
        outputs = model(images)
        loss = criterion(outputs, labels)
        loss.backward()
        optimizer.step()
        total_images += images.size(0)

    train_duration = time.time() - start_time

    model.eval()
    correct = 0
    total = 0
    inference_start = time.time()
    with torch.no_grad():
        for images, labels in test_loader:
            images, labels = images.to(device), labels.to(device)
            outputs = model(images)
            _, predicted = torch.max(outputs.data, 1)
            total += labels.size(0)
            correct += (predicted == labels).sum().item()
```

```
    inference_time = time.time() - inference_start
    accuracy = 100 * correct / total

    dummy_input = torch.randn(1, 3, 224, 224).to(device)
    macs, params = profile(model, inputs=(dummy_input,), verbose=False)
    macs, params = clever_format([macs, params], "%.3f")

    torch.save({
        'model_state_dict': model.state_dict()
    }, f"checkpoints/{model_name}_{dataset_name}_epoch1.pth")

    results.append({
        "Model": model_name,
        "Dataset": dataset_name,
        "Accuracy (%)": round(accuracy, 2),
        "Params": params,
        "FLOPs": macs,
        "Train Time (s)": round(train_duration, 2),
        "Inference Time (s)": round(inference_time, 2)
    })

    print(f"Оброблено зображень під час тренування: {total_images}")

for model_name in model_names:
    train_and_evaluate(model_name, mnist_loader_train, mnist_loader_test, "MNIST")
    train_and_evaluate(model_name, cifar_loader_train, cifar_loader_test, "CIFAR10")

results_df = pd.DataFrame(results)
print("\nПорівняльна таблиця результатів:")
print(results_df.to_string(index=False))
```

Додаток Б

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

(Презентація)

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут Інформаційних технологій
Кафедра Комп'ютерної інженерії

КВАЛІФІКАЦІЙНА РОБОТА
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВРА

на тему: «Нейромережеві моделі для розпізнавання зображень»

Виконав студент групи КІД-41
Левченко Юрій Олегович

Керівник кваліфікаційної роботи:
Зуб Олександр Вікторович

Київ – 2025

Мета роботи – порівняльний аналіз різних архітектур нейромережевих моделей для задачі розпізнавання зображень.

Об`єкт дослідження – процес розпізнавання зображень за допомогою нейромережевих моделей.

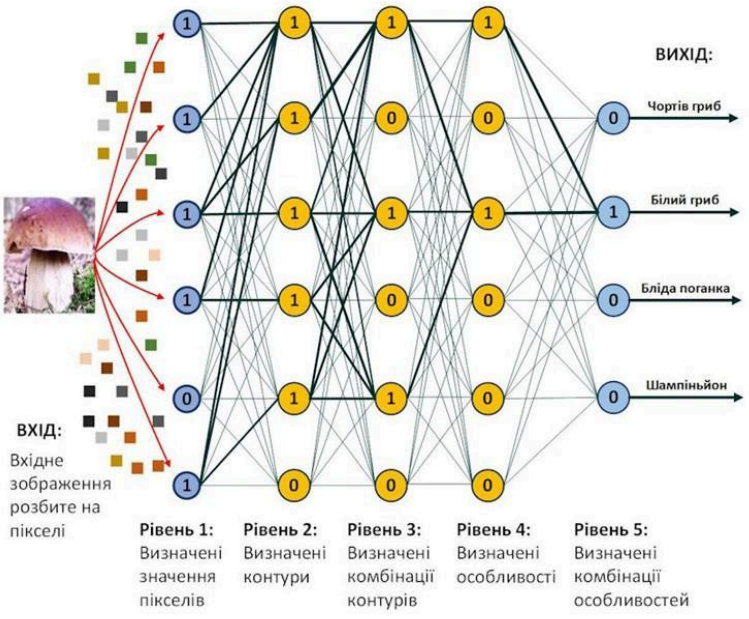
Предмет дослідження – нейромережеві моделі, що використовуються для розпізнавання зображень, включаючи їхні архітектури, алгоритми навчання, параметри та можливості застосування.

Актуальність обраної теми

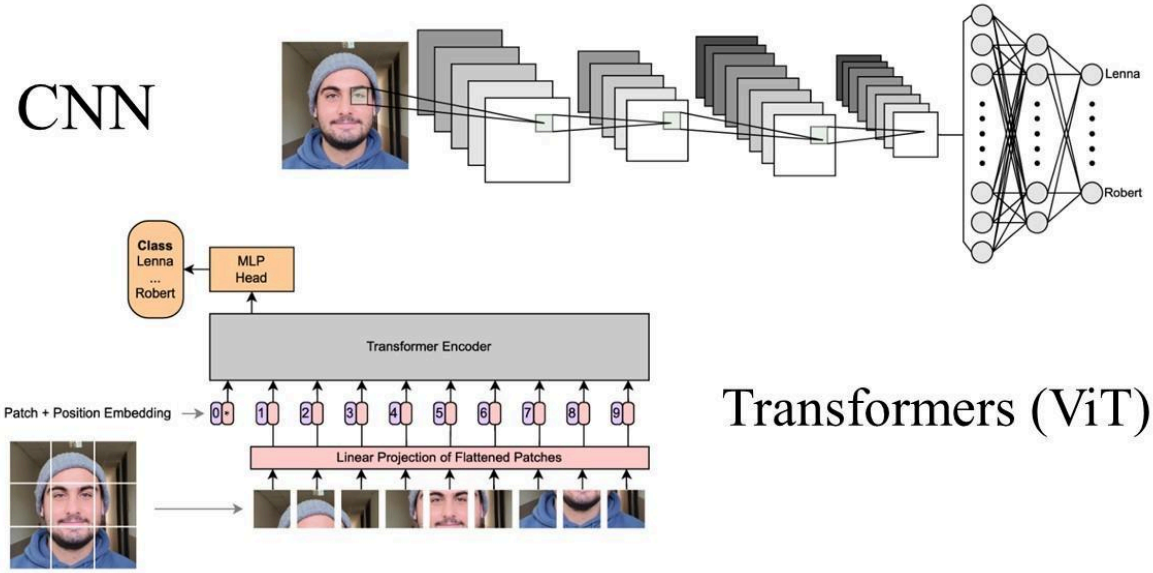
Нині розпізнавання зображень перетворилося з вузькоспеціалізованої області досліджень на фундаментальну технологію, що пронизує практично всі аспекти нашого життя. З розвитком цифрових технологій та експоненційним зростанням обсягів візуальних даних, ефективне та точне розпізнавання зображень стає критично важливим для вилучення корисної інформації та прийняття обґрунтованих рішень.

Однак, незважаючи на значні досягнення, існують невід'ємні недоліки, пов'язані з розпізнаванням складних зображень у реальних умовах. Тому, глибокий аналіз та ретельне порівняння ефективності різних нейромережевих моделей для розпізнавання зображень залишається актуальною науково-практичною задачею

Приклад роботи нейронної мережі



Обрані архітектури



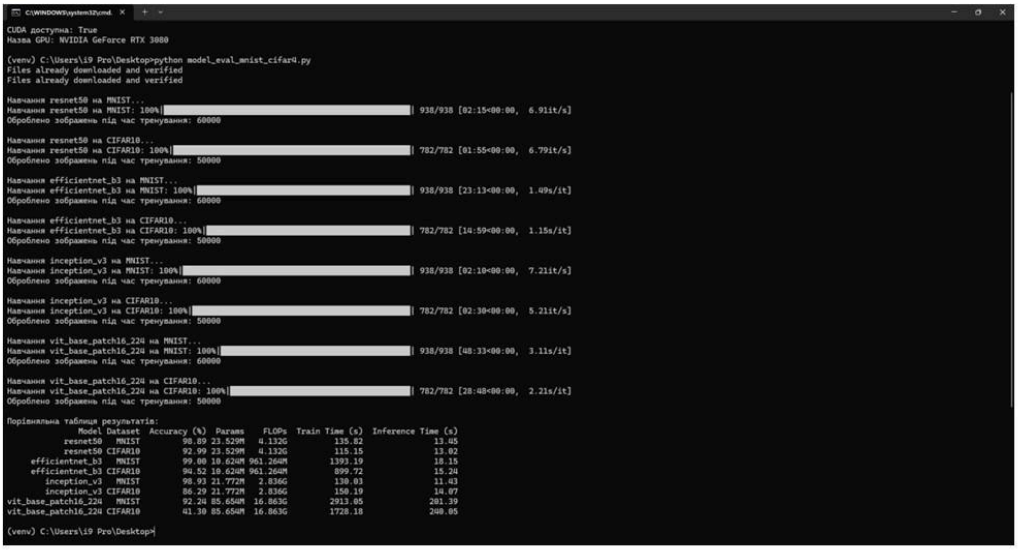
5

Обрані моделі

Модель	Глибина	Ширина	Складність навчання	Гнучкість для перенавчання
ResNet-50	50 шарів (залишкові блоки)	64–2048 фільтрів	Легка. Стабільне навчання, добре масштабується	Висока. Підходить для багатьох задач, легко адаптується
EfficientNet-B3	~30 шарів (MBConv-блоки, Swish-активації)	40–384 каналів (авто-масштабовані)	Помірна. Потребує правильного налаштування навчання	Добра. Але може вимагати тоншого тюнінгу при перенавчанні
INCEPTION v3	12 encoder-блоків (Transformer layers)	768 embedding size, 12 attention heads	Висока. Потребує великих даних і ресурсів	Добра з переднавченими вагами, але нестабільна “з нуля”
ViT-B/16	~48 шарів (паралельні гілки, Inception-блоки)	192–2048 фільтрів	Помірна. Складна архітектура, потребує часу на налаштування	Середня. Менш зручна для перенавчання, ніж ResNet

6

Приклад запуску коду у терміналі з виводом результатів



Порівняння моделей

Модель	Top-1 Acc. (%)	Параметри (млн)	FLOPs (G)	Інференс ≈ ms/зобр.	Стійкість*
ResNet-50	76.0	25.6	4.1	10	висока
Inception v3	78.0	24	5.7	15	середня
EfficientNet-B3	81.6	12	1.8	8	висока
ViT-B/16	77.9	86	55	20	низька

Модель	Dataset	Top-1 Acc. (%)	Параметри (млн)	FLOPs	Train Time (s)	Inference Time (s)
ResNet-50	MNIST	98.89	23.529	4.132G	135.82	13.45
	CIFAR10	92.99	23.529	4.132G	115.15	13.02
EfficientNet-B3	MNIST	99.00	10.624	961.264M	1393.19	18.15
	CIFAR10	94.52	10.624	961.264M	899.72	15.24
Inception v3	MNIST	98.93	21.772	2.836G	130.03	11.43
	CIFAR10	86.29	21.772	2.836G	150.19	14.07
ViT-B/16	MNIST	92.24	85.654	16.863G	2913.05	201.39
	CIFAR10	41.30	85.654	16.863G	1728.18	240.05

Рекомендації

Оптимальні сфери застосування:

- *ResNet.* Універсальне рішення для класифікації та детекції об'єктів, особливо на серверних та edge-системах;
- *Inception.* Кращий вибір для multi-scale задач та сегментації, коли потрібно аналізувати як дрібні, так і великі деталі;
- *EfficientNet.* Ідеальний для мобільних і вбудованих систем та середніх серверів;
- *ViT.* Підходить для дослідницьких задач із високо роздільними зображеннями та достатніми обчислювальними ресурсами.

ВИСНОВКИ

У ході дослідження було проведено аналіз ефективності чотирьох ключових архітектур нейронних мереж у задачах розпізнавання зображень: ResNet, Inception, EfficientNet та Vision Transformer.

Вибір конкретної нейромережевої моделі базується на балансі між точністю, ресурсними обмеженнями та специфікою завдання.

Результати цього дослідження можуть бути використані для прийняття обґрунтованих рішень при розробці систем розпізнавання зображень в різноманітних прикладних сценаріях.

Модель ResNet-50 є доцільним вибором як універсальна базова архітектура для широкого кола задач комп'ютерного зору. Вона забезпечує збалансоване поєднання точності, обчислювальної ефективності та стійкості до змін у даних.