УДК 004.8
DOI https://doi.org/10.32782/tnv-tech.2025.2.24

ОГЛЯД СУЧАСНИХ АЛГОРИТМІВ ГЛИБИННОГО НАВЧАННЯ ДЛЯ КЛАСИФІКАЦІЇ ЗОБРАЖЕНЬ

Швець С. В. *– аспірант Приватного вищого навчального закладу «Європейський університет»*
ORCID ID: 0009-0006-0060-7760

Задача класифікації зображень є однією з актуальних проблем сучасного програмування та вже тривалий час перебуває у центрі наукових досліджень. З моменту створення алгоритмів штучних нейронних мереж (ANN – Artificial Neural Network) відбувся бурхливий розвиток і адаптація ANN під задачу класифікації зображень. Впровадження згорткових нейронних мереж (CNN) дало суттєвий поштовх розвитку цієї галузі, а згодом інтеграція механізмів уваги сприяла появі нових архітектур, що поєднують згорткові мережі з механізмом уваги. Це дозволило адаптувати механізм уваги для задач класифікації зображень та сприяло розвитку відповідних моделей. Особливу увагу нині приділяють моделям, що використовують трансформери для класифікації зображень, а також гібридним архітектурам, які поєднують згорткові нейронні мережі з трансформерами. Напрям гібридних моделей вважається одним із найперспективніших у розвитку алгоритмів класифікації. У цій статті проведено аналіз основних архітектур нейронних мереж для розв'язання задачі класифікації зображень, а також здійснено порівняння їхньої ефективності та обчислювальної складності. Також розглянуто основні тренди, які визначають напрямки дослідження і підсумовано ключові відмінності та застосування цих моделей. Наукова новизна цієї статті полягає у комплексному аналізі ефективності та обчислювальної складності сучасних архітектур класифікації зображень, зокрема трансформерних моделей та їхніх гібридних варіацій. У результаті проведеного аналізу встановлено, що гібридні архітектури, які поєднують згорткові нейронні мережі з механізмом уваги, є перспективним напрямом у задачах класифікації зображень. Здійснене порівняння різних архітектур дозволяє окреслити ключові тенденції розвитку методів класифікації зображень та їхнє застосування у практичних задачах.

Ключові слова: *класифікація зображень, згорткові нейронні мережі, трансформери, гібридні архітектури, штучні нейронні мережі, механізм уваги, обчислювальна складність, глибинне навчання.*

Shvets S. V. Review of modern deep learning algorithms for image classification

The task of image classification remains one of the most relevant challenges in modern programming and has long been a central focus of scientific research. Since the introduction of artificial neural networks (ANNs), significant progress has been made in their development and adaptation to image classification problems. The implementation of convolutional neural networks (CNNs) provided a substantial impetus for the advancement of this field. Subsequently, the integration of attention mechanisms led to the emergence of new architectures that combine CNNs with attention modules, thereby enabling attention-based models to be effectively adapted for image classification tasks. Increasing attention is now being paid to models based on transformers, as well as to hybrid architectures that fuse convolutional neural networks with transformers. The hybrid model approach is currently regarded as one of the most promising directions in the development of image classification algorithms.

This article presents a comprehensive analysis of the core neural network architectures designed for image classification, comparing their efficiency and computational complexity. It also outlines the key trends that shape ongoing research directions and summarizes the primary differences and applications of these models. The scientific novelty of this study lies in a analysis of the performance and computational complexity of modern state-of-the-art classification architectures, with particular emphasis on transformer-based and hybrid models. The analysis reveals that hybrid architectures, which integrate convolutional neural networks with attention mechanisms, represent a prospecting direction for solving image classification problems. The comparative overview of different architectures highlights the prevailing trends in the development of classification methods and their applicability to real-world tasks.

Key words: *image classification, convolutional neural networks, transformers, hybrid architectures, artificial neural networks, attention mechanism, computational complexity, deep learning.*

Вступ. Задача класифікації зображень є однією з фундаментальних у галузі комп'ютерного зору та тісно пов'язана з використанням алгоритмів на основі штучних нейронних мереж (ANN) [1]. Дослідження навколо неї стали відправною точкою для багатьох просунутих можливостей як у роботі із зображеннями, таких як розпізнавання обєктів, жестів, облич, опис сцени (scene description), так і у обробці відео – класифікація відео, відстеження об'єкта та інше [2].

Збільшення обчислювальних можливостей призвело до стрімкої еволюції і розвитку алгоритмів класифікації зображень. У цій статті ми роздивимось розвиток алгоритмів з моменту появи згорткових мереж і до гібридних моделей. Відповідно, можна виділити наступні принципові підходи:

– Згорткові мережі
– Згорткові мережі із механізмом уваги
– Трансформери
– Гібридні моделі – поєднання трансформерів та згорткових мереж.

Ми розглянемо шлях розвитку даних алгоритмів та створення мереж рівня SOTA(State Of The Art – витвір мистецтва) та розглянемо сучасні напрями розвитку гібридних моделей.

Також розглянемо безпосередньо згортки та спосіб їх взаємодії у екосистемі нейронних мереж.

Розробка нових моделей була відповіддю на виклики та недоліки існуючих на той час засобів, відповідно згорткові мережі спричинили прорив разом із появою AlexNet [3]. У цей самий момент заклався фундамент головного тестового бенчмарку роботи моделей – ImageNet, а ImageNet Challenge [4] став щорічною подією що спонукало наукову спільноту рік за роком досягати кращих результатів.

Поступове ускладнення і поглиблення моделей було обмежене технічними складнощами затухання градієнтів та слабкою здатністю до узагальнення. Відповіддю на це стало створення ResNet [5] моделей та використання механізмів уваги в згорткових мережах. Але у подальшому, поява трансформерних моделей у галузі NLP [6] корінним способом змінило перспективу розвитку нейронних мереж, і спрямувало зусилля спільноти науковців на розробку адаптованих трансформерних моделей для класифікації зображень (ViT) [7]. Відповідно із перевагами у використанні просунутих механізмів уваги трансформерів, такі моделі отримали і недоліки у вигляді затратності обробки. І відповіддю на це було створення гібридних моделей [8], які поєднують швидкість згортки на ранніх стадіях обробки зображення разом із трансформерами на більш пізніх етапах, використовуючи гнучкість механізмів уваги трансформерів. Таким чином гібридні моделі виглядають найбільш перспективними у пошуку оптимального балансу між використанням трансформерів та згорток.

Мета і завдання дослідження. Створення комплексного огляду архітектур нейронних мереж для класифікації зображень – від впровадження глибоких згорткових мереж [9] до гібридних підходів [8] (поєднання згорток і трансформерів) – з метою цілісного розуміння еволюції методів у цій сфері. Аналіз зсуву трендів у розвитку моделей від класичних згорткових нейромереж до трансформерів із механізмами уваги та оцінка перспектив їхнього подальшого розвитку.

Методи і матеріали досліджень. У цьому дослідженні використано аналітичні, порівняльні та систематизаційні методи для оцінки розвитку архітектур нейронних мереж, що використовуються для класифікації зображень.

Матеріали дослідження що було використано це оглядові статті по тематиці, аналітичні дослідження щодо розвитку нейромереж для класифікації зображень, оригінальні публікації авторів видатних мереж (SOTA) [4].

Результати досліджень. Згорткові нейронні мережі (CNN)

Згорткові нейронні мережі (CNN) є розвитком багатошарового перцептрону та використовують операцію згортки для виділення ключових ознак у вхідних даних [10]. Вхідне зображення проходить через один або декілька згорткових шарів, що дозволяє поступово витягати локальні та ієрархічні ознаки. Отримані ознаки формують вектор представлення зображення, який передається до повнозв’язних шарів для остаточної класифікації [11].

Щоб зменшити розмірність проміжних представлень та знизити обчислювальну складність, після згорткових шарів часто використовуються шари пулінг. Вони виконують операції, такі як усереднення або вибір максимуму (Max/Average Pooling) у обраній області, що допомагає зберігати найважливіші характеристики зображення, зменшуючи надлишкову інформацію [12].

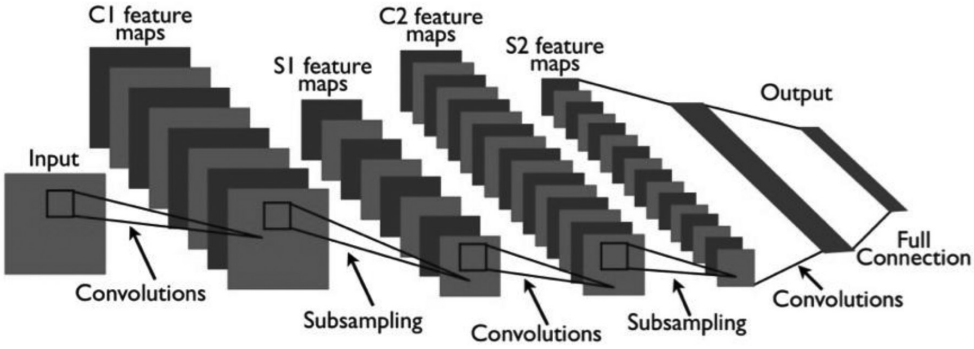

Рис. 1. Архітектура типової згорткової мережі [13]

Роздивимось кожен з архітектурних елементів більш детально.

Згортковий шар

Згортковий шар [11] використовується для виділення ключових характеристик зображення. Чим більше згорткових шарів у нейронній мережі, тим більш абстрактні ознаки вона здатна розпізнавати. Кожен згортковий шар містить певну кількість згорткових ядер (фільтрів), які виконують згорткову операцію над вхідними даними. Ядра є параметрами, що навчаються, і визначають специфічні особливості, які мережа виявляє на цьому рівні. Кількість ядер у шарі визначає кількість ознак, що будуть витягуватися мережею на даному етапі обробки.

Найчастіше використовують згорткові ядра розміром 3 × 3, 5 × 5 або 7 × 7, оскільки вони забезпечують оптимальний баланс між детальністю ознак та обчислювальною ефективністю.

Припустимо ми маємо зображення M розміром WxH, ядро K розміром k x k. Математично операцію згортки можна представити наступним виразом

$$M'(i,j)=\sum_{m,n}^{k} M(i-m,j-n)K(m,n).$$

Далі, результат згортки передається у функцію активації, яка додає нелінійність та дозволяє моделі навчатися складнішим залежностям у даних.

Функції активації

Функцій активації додають нелінійність у модель, завдяки чому нейронна мережа отримує можливість вивчати складні залежності у даних. Функції ReLu,

Leaky ReLu та його модифікації є найчастіше використовуваними та обчислювально ефективними[посилання] та допомагають уникати ефекту затухання градієнтів, проте існує великий перелік інших функцій, що також представляють цікавість в залежності від специфіки задач: sigmoid, tanh, ELU, softmax [14].

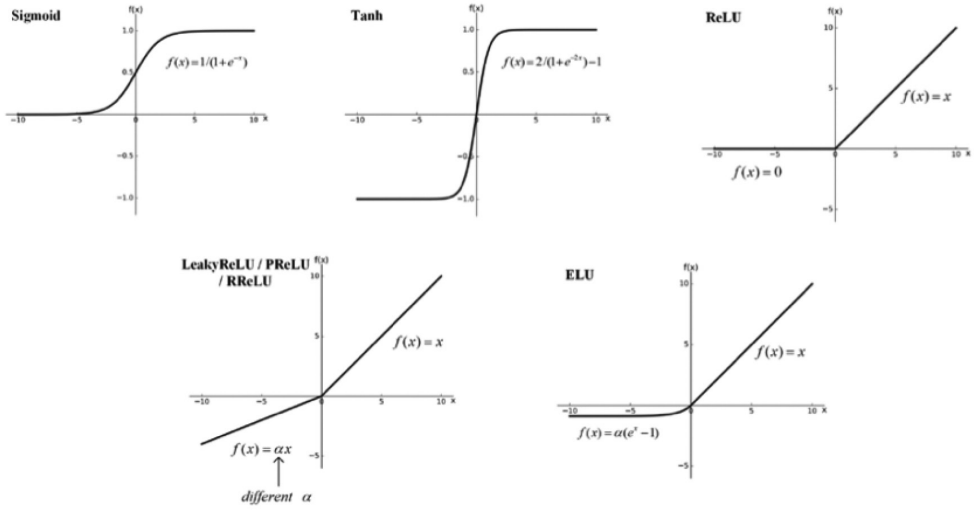

Рис. 2. Графіки типових функцій активації [15]

Не існує загального правила вибору функції активації, проте вдалий вибір може суттєво покращити результативність роботи мережі.

Пулінг

Пулінг шар отримує результат роботи функції активації та застосовує функцію пулінгу на фіксованому вікні даних, далі вікно зсувається(крок називають страйд – stride) і обраховується значення функції пулінгу на наступних даних. Цей процес виконується на всьому об'ємі даних відповідно. Пулінг використовується для зменшення розмірності та узагальнення результатів роботи попереднього шару. Крім того, він забезпечує інваріантність до зсувів, змін масштабу та поворотів, що покращує стійкість моделі до змін у вхідних даних [12].

Найчастіше використовують функції пулінгу такі як

$$F(x) = \text{Max}(x) \text{ [Max Pooling]}$$

або

$$F(x) = 1/n \cdot \text{sum}(x), \text{ де } x \text{ це рухоме вікно пулінгу.}$$

Окрім класичних стратегій пулінгу, також використовуються альтернативні підходи, такі як змішаний пулінг, стохастичний пулінг, метод просторових пірамід та інші.

Повнозв'язний шар

Повнозв'язний шар отримує вектор ознак, сформований згортковими шарами, і безпосередньо виконує задачу класифікації. Він може містити декілька прихованих шарів для подальшої обробки та інтерпретації високорівневих ознак, зазвичай побудований за архітектурою багатошарового перцептрону [1].

На виході повнозв'язний шар зазвичай використовує функцію активації sigmoid для задач бінарної класифікації або softmax для задач багатокласової

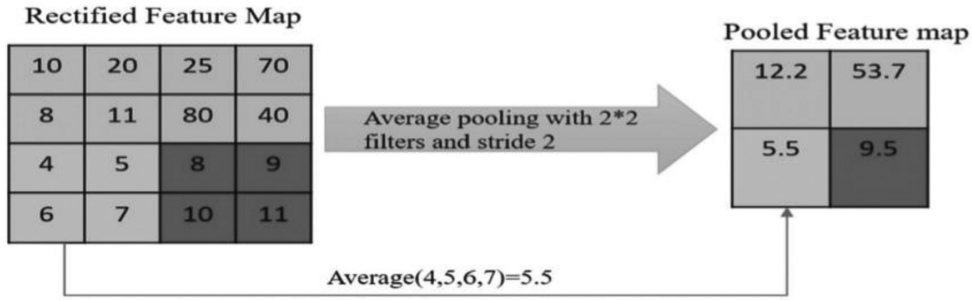

Рис. 3. Ілюстрація Avg пулінгу [12]

класифікації. Це стало загальноприйнятим стандартом у побудові класифікаційних нейромереж.

Функція втрат

Функція втрат безпосередньо впливає на ефективність навчання моделі, оскільки вона визначає відстань між прогнозованим та еталонним значенням вихідного сигналу мережі.

У задачах класифікації зазвичай використовується комбінація Softmax + Cross-Entropy, що поєднує функцію активації softmax та функцію втрат на основі крос-ентропії. Така композиція є обчислювально ефективною та більш математично стійкою до похибок [16]

$$L = -\log \frac{e^{z_{true}}}{\sum_j e^{z_j}}.$$

Загалом можна виділити наступні загальні використовувані функції

Таблиця 1

Типові функції втрат

Функція	Формула	Пояснення
L1	$L = \sum_{i=1}^{N} \left| y_i - w^T x_i - b \right|$	Середня абсолютна похибка. У нейромережах використовується рідко у чистому вигляді, але може бути застосовним для специфічних задач де в данних можуть бути викиди
L2	$L = \sum_{i=1}^{N} (y_i - w^T x_i - b)^2$	Середня квадратична похибка. Набільш використовувана формула помилки, має зручні властивості градієнта
Huber Loss	$L_\delta(y, f(x)) = \begin{cases} \frac{1}{2}(y - f(x))^2, \\ \left\|y - f(x)\right\| \le \delta \\ \delta \left\|y - f(x)\right\| - \frac{1}{2}\delta^2, \end{cases}$	Поєднує властивості L1 та L2, універсальна але ускладнює роботу через необхідність визначення параметру

Інколи є доречним використання інших функції втрат, таких як center-loss, A-Softmax, AM-Softmax [16].

Оптимізатор

Оптимізатор визначає алгоритм пошуку мінімуму цільової функції (зазвичай локального), тому його ефективність є критичною для успішного навчання нейромережі. Більшість сучасних оптимізаторів базуються на методі градієнтного спуску, обчислюючи часткові похідні першого порядку функції втрат за параметрами моделі. Це дозволяє визначити напрям зміни параметрів для їхнього оновлення [17].

Градієнти обчислюються у зворотному порядку через шари моделі за допомогою методу зворотного поширення помилки (Backpropagation). Оновлення параметрів виконується згідно з вибраним правилом оновлення. На кожному кроці оптимізатор коригує параметри всіх шарів, що забезпечує ефективну збіжність до оптимального розв'язку.

Таблиця 2

Типові оптимізатори

Оптимізатор	Назва	Пояснення
BGD	Градієнтний спуск на батчі	Обчислює градієнт зміни параметрів на всьому батчі даних. Точний, але затратний та вимагає багато памяті
SGD	Стохастичний градієнтний спуск	Випадково вибирає міні-батч і вираховує градієнт на ньому. Швидше працює але гірша збіжність
SGD з моментом	Стохастичний градієнтний спуск з моментом	Модифікація SGD що враховує момент змін від попереднього оновлення і таким чином покращує збіжність та роботу на шумних даних
AdaGrad	Адаптивний градієнтний спуск	Адаптує швидкість навчання для кожного параметра окремо відповідно до їх попередньої зміни. Застосований коли певні параметри є відокремлені (sparse)
RMSProp	Метод середнього квадрата градієнтів	Модифікація AdaGrad, використовує зважене середнє значення градієнтів для адаптації зміни швидкості навчання. Це робить його ефективним для навчання глибоких мереж
Adam	Адаптивний градієнтний спуск Adam	Поєднання AdaGrad та RMSProp. Автоматично адаптує швидкість навчання по кожному параметру. Найбільш використовуваний оптимізатор

Класичні моделі, що базуються на CNN

LeNet

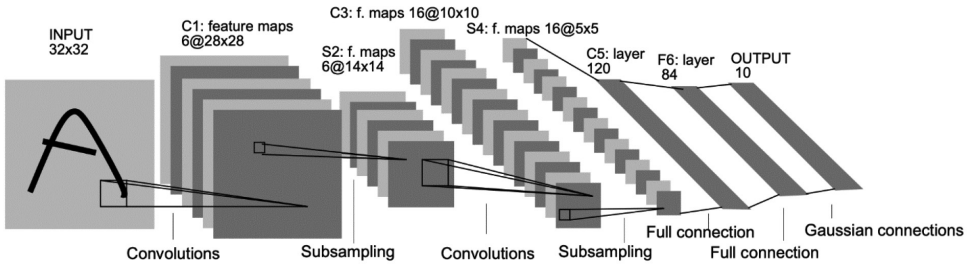

Рис. 4. Архітектура LeNet [3]

У 1998 році Ян Лекун (Yann LeCun) запропонував модель LeNet-5 [3] – на той час це була революційна архітектура, яка значно перевершувала існуючі методи розпізнавання зображень. Вона стала класичним прикладом згорткових нейромереж, концепції якої використовуються і сьогодні [18].

Архітектура LeNet-5 складається із 7 шарів та містить приблизно 60 000 параметрів. Вона поєднує три згорткові шари, кожен з яких чергується з Max Pooling-шаром, а після них слідують три повнозв'язні шари.

Хоча LeNet-5 залишається ефективною для простих наборів даних, таких як MNIST, її продуктивність на складніших задачах є недостатньою за сучасними стандартами.

AlexNet

У 2012 році Алекс Крижевський (Alex Krizhevsky) запропонував архітектуру AlexNet [19], яка стала значним розвитком LeNet завдяки збільшенню глибини як згорткових, так і повнозв'язних шарів. Модель містить приблизно 60 мільйонів параметрів, що дозволяє їй ефективніше виділяти ознаки та працювати із зображеннями більшого розміру та різноманітності.

VGGNet

У 2014 році Карен Сімоньян (Karen Simonyan) та Ендрю Зіссерман (Andrew Zisserman) запропонували архітектуру VGGNet [20], яка розвиває ідеї, закладені в AlexNet, але додає кілька важливих концептуальних змін та обмежень:

- Використання згорткових шарів фіксованого розміру 3 × 3, що покращує локальне захоплення ознак.
- Модульний дизайн – стекування згорткових шарів один за одним із чергуванням Max Pooling-шарів.
- Збільшення глибини до 16–19 шарів, що дозволяє моделі навчати складніші представлення.

Дослідники вивчили різні конфігурації глибини мережі та ступінь вкладеності згорткових модулів. Тренування проводилося на ImageNet, але отримані моделі показали високу здатність до узагальнення й на інших наборах даних, які не використовувалися під час навчання.

InceptionV1/2/3/4

Сімейство моделей Inception було запропоновано Крістіаном Сьогеді (Christian Szegedy) у 2014 році. Ця архітектура поєднує кілька ключових ідей: глибину, модульний дизайн (подібний до VGGNet) та використання різних згорткових фільтрів на одному рівні з подальшим об'єднанням їхніх результатів [9] [21] [22] [23].

Також у моделі реалізовано концепцію Network-in-Network [46] [24], що включає використання згортки 1 × 1 перед більшими згортками (3 × 3 та 5 × 5) або після них для ефективного об'єднання інформації з різних каналів.

Подальші версії моделі (InceptionV2, V3, V4) удосконалювали початкову концепцію, покращуючи ефективність навчання та обчислювальну складність.

InceptionV4 можна вважати найбільш оптимізованою версією з точки зору продуктивності та ресурсозатратності.

ResNet

Дослідження у сфері глибоких нейронних мереж показали, що збільшення кількості шарів не завжди призводить до покращення продуктивності моделі, а навпаки може її погіршити. Цей ефект не завжди вдається нівелювати методами нормалізації, зокрема Batch Normalization, що може спричинити проблеми зі збіжністю під час тренування. Це явище отримало назву деградації глибоких мереж

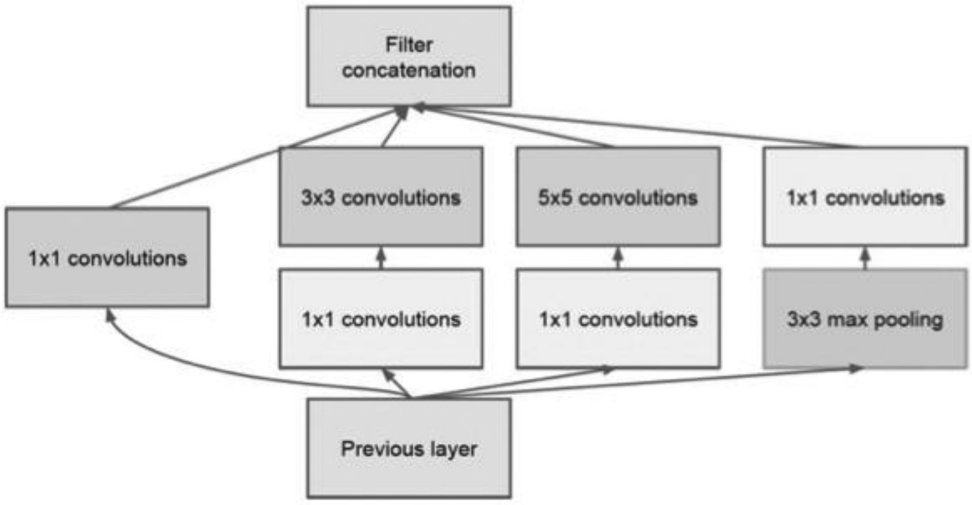

Рис. 5. Модуль InceptionV1 із зменшенням розмірності [9]

Таблиця 3

Порівняння архітектур Inception V1–V4

Архітектурні особливості	**Inception V1**	**Inception V2**	**Inception V3**	**Inception V4**
Згортки та архітектурні особливості	5 × 5 використовуються напряму 3 × 3 використовуються напряму	5 × 5 замінені на дві 3 × 3 3 × 3 розбиті на дві – 1 × 3 та 3 × 1	Факторизація згорток великого розміру Факторизація просторових згорток	Введення уніфікованих Inception та Reduction блоків
Batch Norm	не задіяно	Використовується	Використовується і додається Label Smoothing – регуляризацію для покращення узагальнення	Batch Norm + Label Smoothing
Оптимізатор	Стохастичний (SGD)	Стохастичний (SGD) + Batch Norm	RMSProp	RMSProp або Adam

(Network Degradation) [5] і певною мірою сповільнило подальший розвиток архітектур глибокого навчання.

У 2016 році Каймін Хе (Kaiming He) запропонував архітектуру ResNet [5], яка використовує обхідні (залишкові) з'єднання (shortcut connections) як рішення проблеми деградації глибоких мереж.

Основна ідея ResNet полягає у вивченні залишкової функції $F'(x)$ замість безпосереднього наближення бажаного перетворення F(x):

$$F'(x) = H(x) - x$$

замість

$$F(x) := H(x),$$

що дозволяє мережі легше передавати градієнти під час зворотного поширення помилки та покращує стабільність навчання. Схематично це виглядає наступним чином

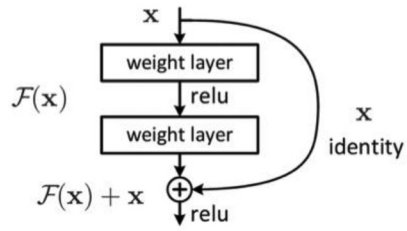

Рис. 6. Будівельний блок залишкової мережі [5]

Використання залишкових (residual) з’єднань значно покращило ефективність глибоких мереж у порівнянні із традиційними архітектурами, відкривши шлях до подальшого розвитку глибокого навчання. Сьогодні цей підхід став стандартною архітектурною особливістю, яка може бути інтегрована у різні типи нейромереж. Наприклад, ResNet-методологія була використана для покращення InceptionV4, що призвело до створення Inception-ResNet, де прямі з’єднання були додані до Inception-блоків для стабілізації навчання.

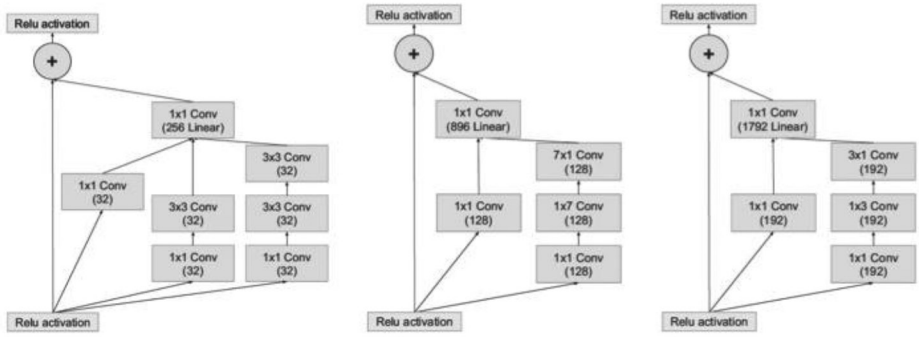

Рис. 7. Використання залишкових з’єднань у InceptionV4 [23]

DenseNet

Продовжуючи дослідження методів боротьби із затуханням градієнтів та деградацією глибоких мереж, Гао Хуанг (Gao Huang) та співавтори у 2017 році запропонували архітектуру DenseNet [25], яка є подальшим розвитком ідеї залишкових з’єднань у ResNet.

Основна концепція DenseNet полягає в тому, що кожен наступний шар отримує не лише вихід попереднього шару, а й всі попередні шари через прямі з’єднання (dense connections).

Авторами також введено параметр, відомий як темп зростання (growth rate), який визначає темп зміни кількості каналів у мережі.

Ключова відмінність від ResNet полягає в тому, що у DenseNet результати роботи всіх попередніх шарів безпосередньо доступні для кожного наступного шару через конкатенацію (concatenation), тоді як у ResNet використовується операція додавання

(element-wise addition). Це дозволяє ефективніше передавати інформацію та градієнти по мережі, що сприяє кращому навчанню глибоких моделей.

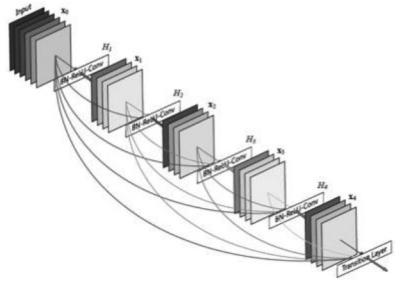

Рис. 8. Архітектура DenseNet [25]

Механізми уваги.

Використання механізму уваги у CNN

Механізм уваги є ключовим елементом розвитку нейронних мереж, і загалом цей механізм можна роздивлятись у аналогії із тим, як мозок людини обробляє інформацію. Певні деталі можуть отримати значну увагу, а деякі великі масиви можуть бути проігноровані через їх нерелевантність відносно поточного запиту [26].

У контексті згорткових нейромереж (CNN) механізм уваги реалізується за допомогою маски уваги, яка підсилює найбільш інформативні ознаки, елементи або області зображення, водночас зменшуючи вплив другорядної інформації. Це покращує якість розпізнавання та підвищує ефективність використання ресурсів моделі.

CNN моделі із механізмом уваги

Residual Attention Network (RAN)

Архітектура Residual Attention Network(RAN)[27] складається із стекованих модулів уваги, кожен із яких містить два паралельні потоки: один відповідає за вивчення ознак, а інший – за формування маски уваги. Маска множиться на вихідний сигнал, що дозволяє адаптивно підсилювати або пригнічувати окремі ознаки.

У моделі реалізовано багаторівневу увагу (multi-level attention), яка працює як на локальному, так і на глобальному рівнях, покращуючи адаптивність до різних масштабів об’єктів. Щоб уникнути затухання градієнтів і покращити збіжність під час навчання, блок уваги будується на основі залишкових (residual) з’єднань, що дозволяє ефективно передавати градієнти через глибокі шари. Використання soft attention забезпечує стабільне навчання та покращує якість класифікації.

SENet

У 2017 році Цзе Ху (Jie Hu) та співавтори запропонували підхід, що фокусується на аналізі залежностей між каналами ознак у згорткових нейромережах. Було представлено архітектуру Squeeze-and-Excitation (SE) [28], яка складається з SE-блоків, що можуть бути стековані та інтегровані в існуючі CNN-архітектури.

Кожен SE-блок адаптивно калібрує значущість каналів відповідно до ознак, отриманих після проєкції Ftr у простір ознак U, враховуючи взаємозв’язки між каналами та їхній внесок у представлення ознак. Процес реалізується у два основні етапи:

Squeeze (Fsq) – стискання просторової інформації для отримання компактного дескриптора каналу.

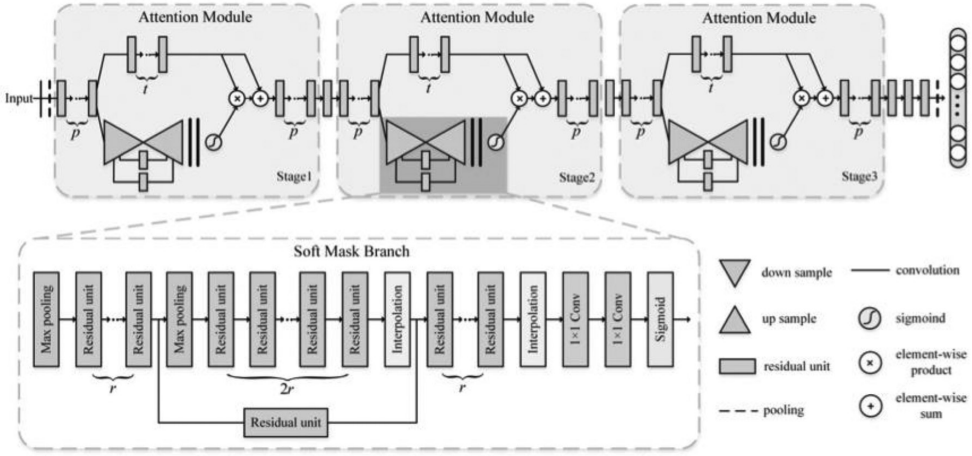

Рис. 9. Архітектура RAN [27]

Excitation (Fex) – адаптивне масштабування значень каналів шляхом навчання вагових коефіцієнтів.

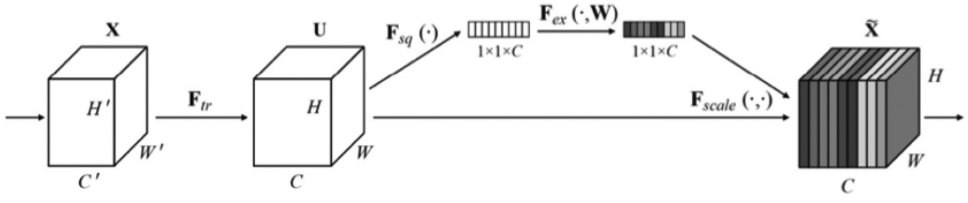

Рис. 10. Архітектура SE [28]

BAM та CBAM

Подальший розвиток архітектур механізмів уваги відбувся завдяки роботам Джонні Ву (Jongchan Woo) та Джунгіль Парка (Jungil Park). У 2018 році вони запропонували дві архітектури модулів уваги – BAM (Bottleneck Attention Module) [29] та CBAM (Convolutional Block Attention Module) [30]. Обидва підходи поєднують просторову увагу та канальну увагу, але відрізняються порядком їхнього обчислення.

BAM обчислює просторову та канальну увагу паралельно, після чого комбінує отримані карти уваги шляхом поелементного додавання (element-wise addition). Натомість CBAM використовує послідовний підхід: спочатку застосовується канальна увага, яка зважує просторові ознаки, а потім розраховується карта просторової уваги, що коригує ознаки на виході.

Обидва модулі призначені для інтеграції в існуючі CNN-мережі, покращуючи їхню продуктивність та здатність до адаптивної фільтрації важливих ознак.

Зменшенні мережі

Обчислювальна складність нейронних мереж є ключовим обмежуючим фактором у їхньому широкому впровадженні, особливо для застосувань на мобільних пристроях та ізольованих обчислювальних системах. Це спричинило появу тренду на розробку оптимізованих архітектур, які зберігають високу продуктивність, але споживають менше обчислювальних ресурсів.

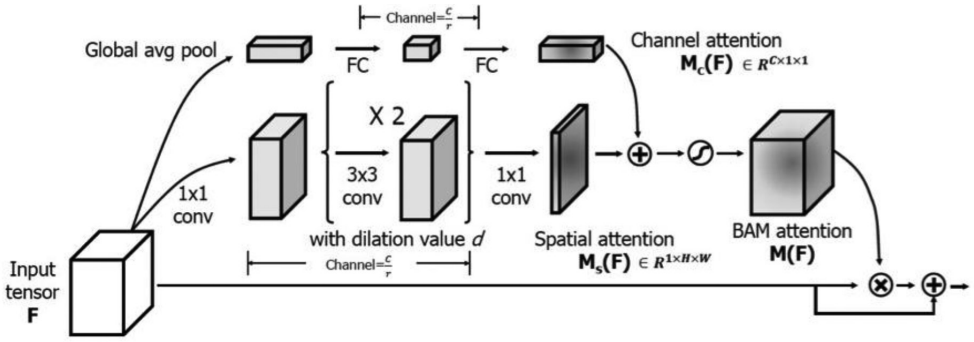

Рис. 11. Архітектура BAM [22] [29]

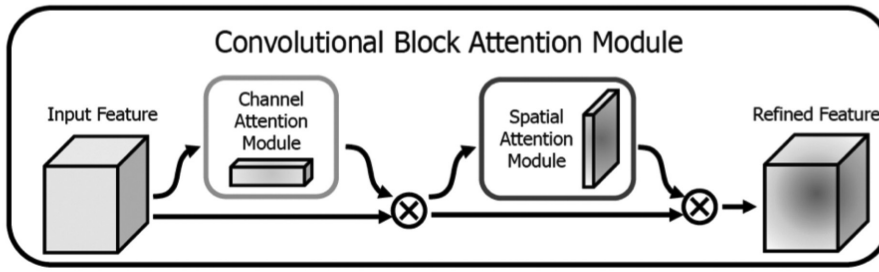

Рис. 12. Архітектура CBAM [23] [30]

SqueezeNet

У 2016 році Форрест Іандола (Forrest Iandola) та співавтори представили архітектуру SqueezeNet [31], основною особливістю якої є надзвичайно ефективне використання параметрів мережі завдяки трьом ключовим стратегіям:

(а) часткова відмова від згорток 3 × 3 на користь 1 × 1;

(б) зменшення кількості вхідних каналів до згорток 3 × 3;

(в) пізній даунсемплінг (збільшений крок згортки >1 на пізніх рівнях мережі).

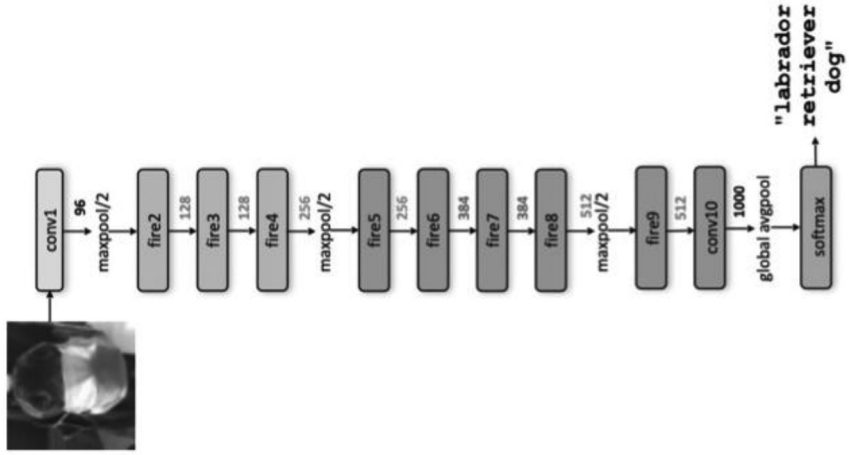

Рис. 13. Архітектура SqueezeNet [31]

Ці стратегії реалізовано в модулі Fire, який складається зі згорток 1 × 1 у шарі ущільнювача (squeeze) та комбінації згорток 1 × 1 і 3 × 3 у шарі розширювача (expand). Таким чином, просторова інформація не втрачається, але кількість каналів зменшується до кількості фільтрів у шарі ущільнювача.

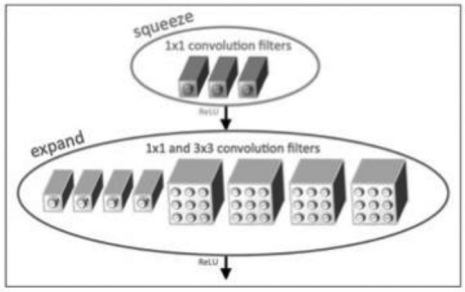

Рис. 14. Архітектура Fire модуля [31]

Для регулювання структури мережі автори пропонують три гіперпараметри – s1x1, e1x1, та e3x3, що відповідно регулює кількість згорток у шарах ущілюнювача(s1x1) та розширювача(e1x1, та e3x3 для згорток 1 × 1 та 3 × 3 відповідно). Гіперпараметр s1x1 має бути менше за (e1x1 + e3x3) щоб відповідати стратегії (б).

MobileNet

Щоб надати розробникам і дослідникам гнучкий інструмент для створення мобільних застосунків, у 2017 році Ендрю Говард (Andrew Howard) та його колеги запропонували модель MobileNet [32]. Її ключова архітектурна особливість – використання роздільних згорток (depthwise separable convolutions), що складаються з глибоких згорток (depthwise convolutions), які застосовуються до кожного каналу окремо, та точкових згорток (pointwise convolutions), які поєднують інформацію між каналами. Це значно зменшує обчислювальну складність порівняно з традиційними згортками 3 × 3 – у 8–9 разів у середньому.

Крім того, автори запровадили два гіперпараметри, α та ρ, які дозволяють масштабувати розмірність вхідного зображення ($\rho < 1$) і пропорційно змінювати кількість каналів ($\alpha < 1$), що допомагає знаходити баланс між точністю та обчислювальною складністю мережі.

У подальшому розвитку моделі, MobileNetV2 (2018) [33], було запропоновано використання так званих інвертованих вузьких блоків (Inverted Bottlenecks). На відміну від ResNet, де використовується структура стискання-розширення-стискання, тут застосовано зворотній порядок – розширення-обробка-стискання. У поєднанні з адаптованою функцією активації (ReLU6 [14]) це покращує ефективність роботи мережі у розширених просторах ознак.

У 2019 модель MobileNet отримала наступне оновлення до MobileNetV3 [34]. Було інтегровано SE-блоки (Squeeze-and-Excitation) безпосередньо всередині інвертованих вузьких з’єднань (Inverted Bottlenecks), що покращило механізм уваги на рівні каналів. Завдяки використанню нової функції активації hard-swish вдалося зменшити кількість первинних фільтрів з 32 до 16 без втрати точності:

$$h-swish(x) = x\frac{\mathrm{Re}LU6(x+3)}{6}.$$

Крім того, автори застосували Neural Architecture Search (NAS) [35] та NetAdapt [36] для автоматизованої оптимізації та точного налаштування параметрів моделі відповідно до апаратних обмежень конкретних пристроїв.

ShuffleNet

Авторами було запропоновано новітній підхід до згорткових операцій через використання групових згорток (group convolution), включаючи точкові (pointwise group convolution) та просторові згортки, а також механізм перемішування каналів (channel shuffle) [37].

Заміна звичайної згортки на групову значно зменшує обчислювальну складність, але водночас створює проблему ізоляції груп каналів, що обмежує передачу інформації між ними. Використання механізму channel shuffle усуває цей недолік, перемішуючи ознаки між групами каналів та покращуючи розповсюдження інформації в мережі. Це дозволяє суттєво зменшити обчислювальну складність моделі майже без втрати її ефективності.

Таким чином, у ShuffleNet використовується спеціальний модульний блок, основними особливостями якого є поєднання групових згорток із перемішуванням ознак між каналами, що робить модель ефективною для розгортання на пристроях із обмеженими обчислювальними ресурсами.

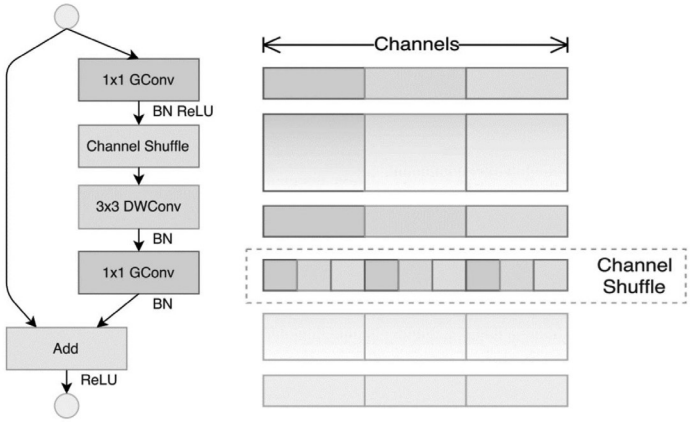

Рис. 15. Архітектура модуля ShuffleNet та перемішування каналів [15]

EfficientNet

У 2019 році М. Тан опублікував результати дослідження, в якому приділив увагу задачі масштабування згорткових мереж [38]. Модель має масштабуватися одночасно в трьох вимірах: глибина (глибина мережі), ширина (кількість каналів) і розмір вхідного зображення, що дозволяє покращити продуктивність без зайвого ускладнення. Масштабування лише одного окремого параметра не є доцільною з точки зору обчислювальної ефективності. Відповідно узгодження параметрів відбувається відповідно рівняння

$$\text{глибина} : d = \alpha^{\varphi};$$

$$\text{ширина} : w = \beta^{\varphi};$$

$$\text{роздільна здатність} : r = \gamma^{\phi};$$

$$\text{s.t. } \alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$$

$$\alpha \geq 1, \beta \geq 1, \gamma \geq 1$$

Відповідно α, β та γ це коефіцієнти, які знаходять через евристичний пошук по сітці(small grid search).

Щоб перевірити ефективність запропонованої стратегії масштабування, автори застосували її до існуючих моделей, а також запропонували нову базову архітектуру. що базується на MBConv – блоку, що поєднує зворотне вузьке з'єднання та SE блок.

Трансформери та сучасні згорткові архітектури

У 2017 році дослідники Google Brain у статті "Attention Is All You Need" [6] представили нову архітектуру – трансформер. Спочатку ця архітектура була розроблена для вирішення завдань машинного перекладу. Згодом трансформер знайшов застосування у задачах комп'ютерного зору та генеративних моделях. Безсумнівно, ця архітектура відіграла визначальну роль у розвитку глибоких нейромереж.

VisionTransformer

Архітектура VisionTransformer [7] відмовляється від класичного підходу до класифікації зображень із використанням згорток. Натомість, використовуються ембедінги патчів. Вхідне зображення розділяється на патчі розміром 16 × 16 для запропонованої моделі ViT. Кожен патч розгортається у вектор і проходить через лінійний шар, отримуючи представлення фіксованої довжини. Далі додається позиційне кодування, щоб трансформер зберігав інформацію щодо вихідної позиції патча. Також до вхідних даних додається [CLS] токен для майбутньої класифікації, який початково ініціалізується випадковими значеннями. Під час проходження через трансформер, [CLS] токен накопичує інформацію про зображення, отриману від усіх патчів. На виході це представлення використовується у повнозв'язному шарі для класифікації.

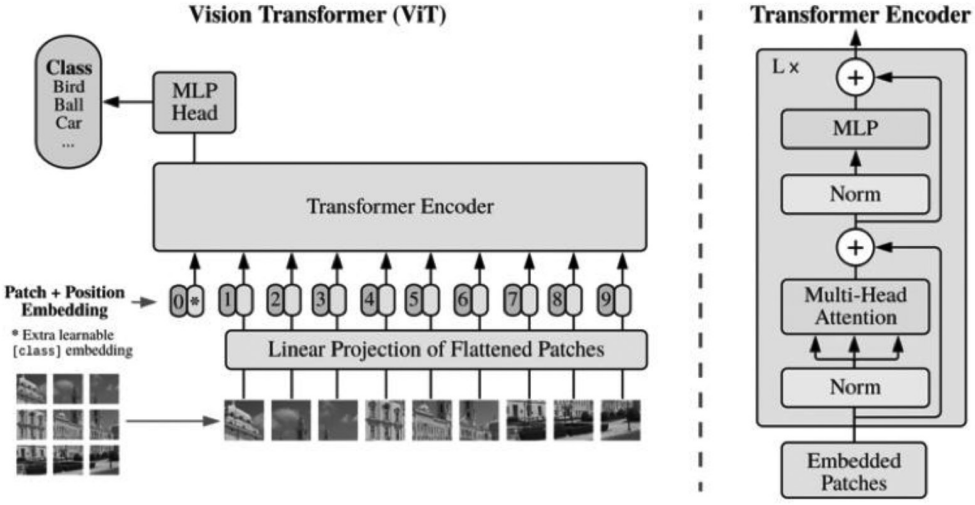

Рис. 16. Архітектура Vision Transformer [29] [7]

ConvMixer

У 2021 році Ешер Трокман опублікував резонансну роботу під назвою «Патчі – це все, що вам потрібно?» ("Patches Are All You Need?" Asher Trockman) [39], у якій поставив під сумнів ефективність використання трансформерів

у задачах класифікації зображень. Натомість він акцентував увагу на використанні представлення зображень у вигляді патчів, що також застосовується у Vision Transformer (ViT). У своїй роботі він емпірично довів, що саме подання зображення через патчі є ключовим фактором у досягненні високої ефективності роботи нейромереж. Трокман використав надзвичайно просту архітектуру, яка повністю базується на згортках, і отримав результат, співставний із моделями ViT.

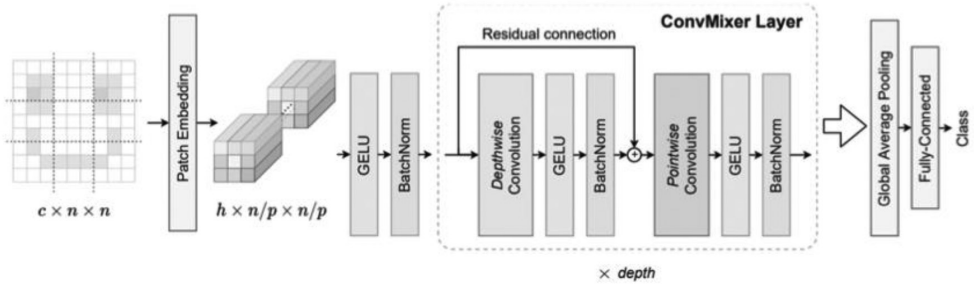

Рис. 17. Архітектура ConvMixer [36] [39]

Swin Transformer

Подальша адаптація трансформерної моделі для задач класифікації зображень була запропонована у 2021 році авторами на чолі з Зе Ліу (Ze Liu) та ін [40]. Основний підхід до розбиття зображення на патчі зберігся, але для покращення обчислювальної ефективності автори використовують локальну увагу в межах кожного вікна уваги замість глобального Self-Attention. Для забезпечення взаємодії між патчами, що не потрапили до одного локального вікна, пропонується механізм зміщених вікон: у наступному шарі вікна зміщуються на половину свого розміру вправо і вниз. Це дозволяє ефективно обмінюватися інформацією між сусідніми вікнами та інтегрувати глобальний контекст. Окрім того, автори застосували ієрархічний підхід до побудови карти ознак, поступово зменшуючи її розмірність на кожному рівні за допомогою механізму Patch Merging.

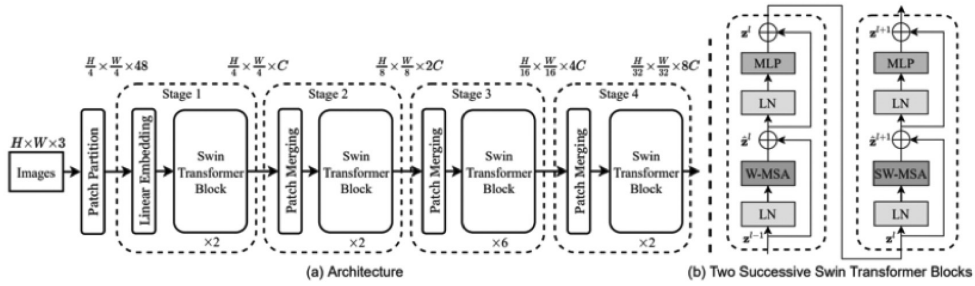

Рис. 18. Архітектура Swin Transfromer [40]

HaloNet

Подальший розвиток використання трансформерів у задачах класифікації зображень отримав новий імпульс у 2021 році завдяки роботі А. Васвані (Ashish Vaswani) та ін. [41]. Автори запропонували альтернативний підхід до локальної уваги, що дозволив значно зменшити обчислювальні витрати у порівнянні з глобальним Self-Attention. У цій моделі зображення розбивається на патчі,

а всередині кожного патчу виконується відокремлення ознак через локальний механізм уваги.

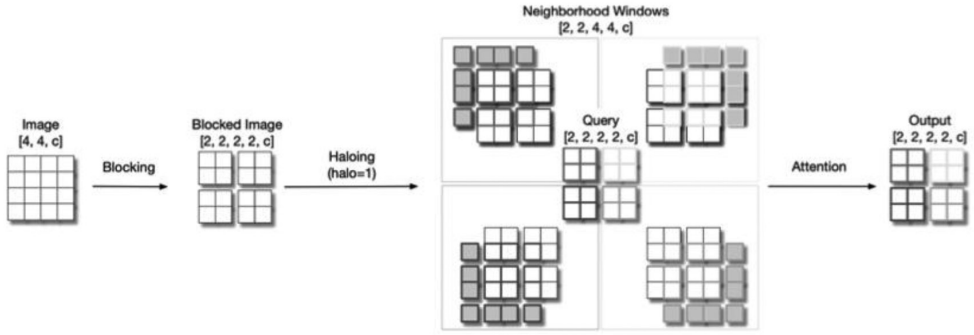

Рис. 19. Ілюстрація роботи halo уваги

Для забезпечення взаємодії між сусідніми регіонами використовується Subsampling – об’єднання сусідніх регіонів із одночасною проекцією ознак у ширший простір. Під час Subsampling також захоплюються патчі з сусідніх областей, що дозволяє моделі отримувати інформацію з ширшого контексту. Це забезпечує ієрархічне навчання, аналогічне до згорткових нейромереж (CNN), де модель поступово переходить від локальних ознак до більш абстрактних понять [41].

Окрім того, модель продемонструвала високу ефективність на зображеннях із великою роздільною здатністю. Автори назвали області, в яких обчислюється увага, «гало-регіонами» (Halo regions), що і дало назву архітектурі HaloNet.

Гібридні моделі (Згортка + Трансформер)

CoAtNet

Продовжуючи дослідження моделей трансформерів, Жиханг Даі з Google Research у 2021 році представив модель CoAtNet [8]. Її ключова особливість – поєднання згорткових мереж із модифікованим механізмом уваги трансформерів, а саме відносною увагою (Relative Attention). Основна ідея полягає у тому, що на початкових етапах використовуються згорткові мережі для виділення локальних ознак, після чого ці ознаки генералізуються на глибших рівнях завдяки механізму уваги, що дозволяє моделі будувати більш абстрактне представлення.

Дослідники провели серію експериментів щодо оптимального балансу між згортками та трансформерами, зупинившись на архітектурі, що складається з двох згорткових блоків та двох трансформерних блоків. Трансформерні блоки були доопрацьовані шляхом впровадження механізму відносної уваги, який враховує не лише самі значення ознак, а й їхнє відносне просторове розташування. Це дозволяє моделі ефективніше працювати з локальними та глобальними залежностями, надаючи більшу стійкість до змін масштабу зображень.

Завдяки такому новаторському підходу CoAtNet продемонстрував відмінні результати. Модель стала більш стійкою до змін масштабу вхідних зображень, вимагала менше даних для тренування, а її обчислювальна складність значно знизилася у порівнянні з чистими трансформерними моделями.

Набори даних

Для порівняння роботи моделей є загальноприйняті набори даних, які використовуються як бенчмарк. Разом із зростанням можливостей моделей зростали також тестові набори, через це не всі моделі мають офіційні бенчмарки на більш

складних наборах даних. Тим не менш наявність спільного знаменника у оцінці моделей є дуже корисною для порівняння моделей і правильного підходу для вибору конкретної моделі для рішення прикладних задач.

MNIST

Набір зображень рукописного тексту(цифри). Складається із 70 000 зображень розмірністю 28 × 28 пікселів.

CIFAR-10

Набір кольорових зображень розмірністю 32 × 32, у 10 класах, 6000 зображень у кожному класі.

CIFAR-100

Набір кольорових зображень розмірністю 32 × 32, у 100 класах, 600 зображень у кожному класі.

ImageNet

Один із найбільш широко використовуваних наборів зображень у машинному навчанні. Його концепція передбачає підтримку в середньому 1000 зображень на кожен клас. Класом вважається набір синонімів відповідно до концептів WordNet, і зазвичай він описується одним або кількома словами чи фразами. Відповідно, існує декілька варіантів набору ImageNet, що відрізняються за розміром і призначенням: ImageNet-1K, ImageNet-21K, ImageNet-V2 та інші [4].

Порівняння роботи моделей

Для порівняння роботи моделей дослідники використовували еталонні датасети. Щодо параметрів роботи моделей, то оцінювання проводиться шляхом порівняння точності розпізнавання класу моделлю, а також до уваги беруться величини розміру моделі (кількість параметрів) та обчислювальна складність, що виражена у BFLOPs. Данні щодо параметрів та роботи моделей взяті з публікацій авторів відповідно бібліографічних джерел.

За даними, наведеними у таблиці, можна простежити тенденцію до зростання складності архітектур глибоких нейронних мереж, що супроводжується покращенням точності розпізнавання зображень. Трансформерні моделі демонструють значний потенціал у вирішенні задач класифікації, однак їх використання залишається обмеженим через високу обчислювальну складність та значну кількість параметрів. Зважаючи на подальший розвиток архітектур та впровадження автоматизованих методів оптимізації гіперпараметрів, можна припустити що слід очікувати поступового зростання ефективності трансформерних моделей. Крім того, інтеграція трансформерів зі згортковими мережами відкриває перспективи зменшення числа параметрів та зниження обчислювальної складності моделей, зберігаючи при цьому високу точність розпізнавання.

Таким чином, мобільні моделі нейронних мереж демонструють значний потенціал, досягаючи високих показників точності при обмежених обчислювальних ресурсах, характерних для мобільних пристроїв. У цьому контексті перспективним напрямом розвитку можна вважати гібридні архітектури, які поєднують згорткові мережі з трансформерними блоками, оптимізуючи кількість параметрів та знижуючи обчислювальну складність для ефективного застосування на мобільних та граничних платформах.

Висновки. У огляді було проаналізовано еволюцію сучасних підходів до класифікації зображень – від простих згорткових моделей до великорозмірних трансформерних архітектур. Хоча згорткові мережі залишаються актуальними, принципово змінюються підходи до синтезу інформації: дедалі більше методів використовують трансформерну обробку для ефективного поєднання ознак.

Таблиця 4

Порівняння повнорозмірних моделей

Назва моделі	Рік	Глибина	Кількість параметрів	CIFAR-10	CIFAR-100	ImageNet 1k Top1	BFLOPS	Архітектура	Особливості
AlexNet	2012	8	60M			58.9		CNN	Dropout + ReLU
VGGNet	2014	16	138M			75.3		CNN	Фіксовані згортки 3 × 3, модульний дизайн
InceptionV1	2014	22	6.8M				1.45	CNN	Архітектура Inception блоку з використанням згорток різного розміру
InceptionV2	2015		11.2M			74.8	1.94	CNN	Пакетна нормалізація, зменшений розмір згортки
InceptionV3	2015	48	24M			78.8	5.73	CNN	Факторизація згорток, Label Smoothing
InceptionV4	2016					82.3		CNN	Уніфіковані Inception та Reduction блокі
ResNet	2015	50				79.26		CNN + ResNet	Залишкові з'єднання
		101				80.13			
		110		93.57					
		152				80.62			
DenseNet	2016	121 ($k = 32$)				74.98		CNN + ResNet	Кожен шар звязанний із всіма попередніми
		201 ($k = 32$)				77.48			
		250 ($k = 24$)	15.3M	94.81	80.36				
Residual Attention Network	2017	92	51.3M			80.50	10.4	CNN + Attention	Стековані модулі уваги
		92	1.9M	95.01	78.29				

Продовження таблиці 4

Назва моделі	Рік	Глибина	Кількість параметрів	CIFAR-10	CIFAR-100	ImageNet 1k Top1	BFLOPS	Архітектура	Особливості
SENet	2017	101	49.2M			81.36	8M	CNN + Attention	Увага до каналів, squeeze-and-excite
		152	146M			82.72	42M		
BAM	2018	ResNet50 + BAM	25.92M		80.00	75.98	3.94	CNN + Attention	Поєднання просторової та канальної уваги
CBAM	2018	ResNet50 + CBAM	28.09M			77.34	3.86	CNN + Attention	Поєднання просторової та канальної уваги
EfficientNetV1	2019	B0	5.3M			77.1	0.39	CNN+ NAS	Автоматизований пошук параметрів мережі
		B1	7.8M			79.1	0.70		
		B2	9.2M			80.1	1.0		
		B3	12M			81.6	1.8		
		B4	19M			82.9	4.2		
		B5	30M			83.6	9.9		
		B6	43M			84	19		
		B7	66M	98.9	91.7	84.3	37	Transformer	
ViT(-H/14)	2021	L	307M	99.42	93.90	87.77		Transformer	Модель трансформера для розпізнавання зображень
		H	632M	99.5	94.55	88.55			
Swin Transformer	2021	T	29M			81.3	4.5G	Transformer	Увага застосована у зсувних вікнах, ієрархічне розпізнавання ознак
		S	50M			86.4	8.7G		
		L	197M			87.3	103.9G		
HaloNet	2021	Base	85M			85.6		Transformer	Тренування на ImageNet-21k, дотренування на ImageNet

Закінчення таблиці 4

Назва моделі	Рік	Глибина	Кількість параметрів	CIFAR-10	CIFAR-100	ImageNet 1k Top1	BFLOPS	Архітектура	Особливості
CoAtNet	2021	CoAtNet-7	2440M			90.88	2586	CNN+ Transformer	Гібридна модель поєднання CNN та транформера
EfficientNetV2	2021	S	24M	98.7	91.5	83.2	8.8	CNN+ NAS	FixRes, Fused-MBConv, NAS
		M	55M	99	92.2	85.1	24		
		L	121M	99.1	92.3	85.7	53		
ConvMixer	2022	768/32	21.1M			80.16		CNN	Патчі + CNN
		1536/20	51.6M			81.37			

Таблиця 5

Порівняння мобільних моделей

Назва моделі	Рік	Версія	Кількість параметрів	Точність, %		
				ImageNet 1k Top1	MFLOPs	Архітектура
SqueezeNet	2016		1.25M	57.5	1700	CNN
MobileNet V1	2017		4.2M	70.6	569	CNN
MobileNet V2	2018	1	3.4M	72.0	300	CNN + Attention + NAS
		1.4	6.9M	74.7	585	
MobileNet V3	2019	1.0 Large	5.4M	75.2	219	CNN + Attention + NAS
ShuffleNet	2017		?	75.3	527	CNN

Також чітко відслідковується еволюція точності, розмірів та зростання обчислювальної складності розглянутих архітектур. Згорткові мережі досягають точності до 75–85 %, трансформерні моделі – до 80–90 %, а гібридні навіть більше 90 %.

Згорткові мережі, оптимізовані для мобільних середовищ (наприклад, MobileNet, EfficientNet), продемонстрували значний потенціал щодо зменшення обчислювальної складності (до 1–5 GFLOPs) при збереженні конкурентної точності, підтверджуючи актуальність оптимізації існуючих моделей.

Відповідно, розміри моделей досягають для згорткових мереж 5–60 млн параметрів, трансформерні моделі: до 600-800 млн+ параметрів, гібридні моделі мають дуже широкий діапазон масштабування від 25 млн параметрів (CoAtNet-0) до 2400 млн параметрів (CoAtNet-7). Подібні архітектури демонструють високу гнучкість і потенціал масштабування, що дозволяє адаптувати їх як для ресурсно-обмежених, так і для високопродуктивних серверних середовищ.

Цілком закономірно очікувати, що трансформерні та гібридні мережі пройдуть шлях оптимізації, адаптуючись як до високопродуктивних, так і до малоресурсних середовищ. Це, своєю чергою, стимулюватиме подальші дослідження щодо взаємозв'язків між гіперпараметрами трансформерних мереж, їхньою ефективністю та обчислювальними затратами. Відповідно, автоматизація пошуку оптимальної архітектури мереж залишатиметься актуальним напрямом розвитку трансформерних і гібридних моделей.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ:

1. Multilayer perceptron: architecture optimization and training / H. Ramchoun et al. *International journal of interactive multimedia and artificial intelligence*. 2016. Vol. 4, no. 1. P. 26. URL: https://doi.org/10.9781/ijimai.2016.415 (date of access: 06.05.2025).

2. Object detection with deep learning: a review / Z.-Q. Zhao et al. *IEEE transactions on neural networks and learning systems*. 2019. Vol. 30, no. 11. P. 3212–3232. URL: https://doi.org/10.1109/tnnls.2018.2876865 (date of access: 06.05.2025).

3. Gradient-based learning applied to document recognition / Y. Lecun et al. *Proceedings of the IEEE*. 1998. Vol. 86, no. 11. P. 2278–2324. URL: https://doi.org/10.1109/5.726791 (date of access: 19.05.2025).

4. ImageNet large scale visual recognition challenge / O. Russakovsky et al. *International journal of computer vision*. 2015. Vol. 115, no. 3. P. 211–252. URL: https://doi.org/10.1007/s11263-015-0816-y (date of access: 19.05.2025).

5. Deep residual learning for image recognition / K. He et al. *2016 IEEE conference on computer vision and pattern recognition (CVPR)*, Las Vegas, NV, USA, 27–30 June 2016. 2016. URL: https://doi.org/10.1109/cvpr.2016.90 (date of access: 19.05.2025).

6. Vaswani A. Attention is all you need. *NIPS'17: proceedings of the 31st international conference on neural information processing systems*. 2017. URL: https://arxiv.org/abs/1706.03762.

7. An image is worth 16x16 words: transformers for image recognition at scale / A. Dosovitskiy et al. URL: https://arxiv.org/abs/2010.11929

8. CoAtNet: marrying convolution and attention for all data sizes / Z. Dai et al. URL: https://arxiv.org/abs/2106.04803

9. Going deeper with convolutions / C. Szegedy et al. URL: https://arxiv.org/abs/1409.4842

10. Sharma N., Jain V., Mishra A. An analysis of convolutional neural networks for image classification. *Procedia computer science*. 2018. Vol. 132. P. 377–384. URL: https://doi.org/10.1016/j.procs.2018.05.198 (date of access: 19.05.2025).

11. Dumoulin V., Visin F. A guide to convolution arithmetic for deep learning. URL: https://arxiv.org/abs/1603.07285

12. A Comparison of Pooling Methods for Convolutional Neural Networks / A. Zafar et al. *Applied Sciences*. 2022. Vol. 12, no. 17. P. 8643. URL: https://doi.org/10.3390/app12178643 (date of access: 19.05.2025).

13. LeCun Y., Kavukcuoglu K., Farabet C. Convolutional networks and applications in vision. *2010 IEEE international symposium on circuits and systems – ISCAS 2010*, Paris, France, 30 May – 2 June 2010. 2010. URL: https://doi.org/10.1109/iscas.2010.5537907 (date of access: 19.05.2025).

14. Sharma S., Sharma S., Athaiya A. ACTIVATION FUNCTIONS IN NEURAL NETWORKS. *International Journal of Engineering Applied Sciences and Technology*. 2020. Vol. 04, no. 12. P. 310–316. URL: https://doi.org/10.33564/ijeast.2020.v04i12.054 (date of access: 19.05.2025).

15. Review of image classification algorithms based on convolutional neural networks / L. Chen et al. *Remote sensing*. 2021. Vol. 13, no. 22. P. 4712. URL: https://doi.org/10.3390/rs13224712 (date of access: 19.05.2025).

16. A comprehensive survey of loss functions in machine learning / Q. Wang et al. *Annals of data science*. 2020. URL: https://doi.org/10.1007/s40745-020-00253-5 (date of access: 19.05.2025).

17. Soydaner D. A comparison of optimization algorithms for deep learning. *International journal of pattern recognition and artificial intelligence*. 2020. Vol. 34, no. 13. P. 2052013. URL: https://doi.org/10.1142/s0218001420520138 (date of access: 19.05.2025).

18. Islam M. R., Matin A. Detection of COVID 19 from CT Image by The Novel LeNet-5 CNN Architecture. *2020 23rd international conference on computer and information technology (ICCIT)*, DHAKA, Bangladesh, 19–21 December 2020. 2020. URL: https://doi.org/10.1109/iccit51783.2020.9392723 (date of access: 19.05.2025).

19. Krizhevsky A., Sutskever I., Hinton G. E. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*. 2017. Vol. 60, no. 6. P. 84–90. URL: https://doi.org/10.1145/3065386 (date of access: 19.05.2025).

20. Simonyan K., Zisserman A. Very deep convolutional networks for large-scale image recognition. URL: https://arxiv.org/abs/1409.1556.

21. Rethinking the inception architecture for computer vision / C. Szegedy et al. *2016 IEEE conference on computer vision and pattern recognition (CVPR)*, Las Vegas, NV, USA, 27–30 June 2016. 2016. URL: https://doi.org/10.1109/cvpr.2016.308 (date of access: 19.05.2025).

22. Ioffe S., Szegedy C. Batch normalization: accelerating deep network training by reducing internal covariate shift. URL: https://arxiv.org/abs/1502.03167.

23. Inception-v4, inception-resnet and the impact of residual connections on learning / C. Szegedy et al. *Proceedings of the AAAI conference on artificial intelligence*. 2017. Vol. 31, no. 1. URL: https://doi.org/10.1609/aaai.v31i1.11231 (date of access: 19.05.2025).

24. Lin M., Chen Q., Yan S. Network in network. URL: https://arxiv.org/abs/1312.4400

25. Densely connected convolutional networks / G. Huang et al. *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, Honolulu, HI, 21–26 July 2017. 2017. URL: https://doi.org/10.1109/cvpr.2017.243 (date of access: 19.05.2025).

26. Brauwers G., Frasincar F. A general survey on attention mechanisms in deep learning. *IEEE transactions on knowledge and data engineering*. 2021. P. 1. URL: https://doi.org/10.1109/tkde.2021.3126456 (date of access: 19.05.2025).

27. Residual attention network for image classification / F. Wang et al. *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, Honolulu, HI, 21–26 July 2017. 2017. URL: https://doi.org/10.1109/cvpr.2017.683 (date of access: 19.05.2025).

28. Squeeze-and-Excitation networks / J. Hu et al. *IEEE transactions on pattern analysis and machine intelligence*. 2020. Vol. 42, no. 8. P. 2011–2023. URL: https://doi.org/10.1109/tpami.2019.2913372 (date of access: 19.05.2025).

29. BAM: bottleneck attention module / J. Park et al. URL: https://arxiv.org/abs/1807.06514.

30. CBAM: convolutional block attention module / S. Woo et al. *Computer vision – ECCV 2018*. Cham, 2018. P. 3–19. URL: https://doi.org/10.1007/978-3-030-01234-2_1 (date of access: 19.05.2025).

31. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size / F. N. Iandola et al. URL: https://arxiv.org/abs/1602.07360

32. MobileNets: efficient convolutional neural networks for mobile vision applications / A. G. Howard et al. URL: https://arxiv.org/abs/1704.04861

33. MobileNetV2: inverted residuals and linear bottlenecks / M. Sandler et al. URL: https://arxiv.org/abs/1801.04381

34. Searching for MobileNetV3 / A. Howard et al. *2019 IEEE/CVF international conference on computer vision (ICCV)*, Seoul, Korea (South), 27 October – 2 November 2019. 2019. URL: https://doi.org/10.1109/iccv.2019.00140 (date of access: 19.05.2025).

35. Elsken T., Metzen J. H., Hutter F. Neural architecture search. *Automated machine learning*. Cham, 2019. P. 63–77. URL: https://doi.org/10.1007/978-3-030-05318-5_3 (date of access: 19.05.2025).

36. NetAdapt: platform-aware neural network adaptation for mobile applications / T.-J. Yang et al. *Computer vision – ECCV 2018*. Cham, 2018. P. 289–304. URL: https://doi.org/10.1007/978-3-030-01249-6_18 (date of access: 19.05.2025).

37. ShuffleNet: an extremely efficient convolutional neural network for mobile devices / X. Zhang et al. *2018 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, Salt Lake City, UT, 18–23 June 2018. 2018. URL: https://doi.org/10.1109/cvpr.2018.00716 (date of access: 19.05.2025).

38. Tan M., Le Q. V. EfficientNet: rethinking model scaling for convolutional neural networks. URL: https://arxiv.org/abs/1905.11946

39. Trockman A., Kolter J. Z. Patches are all you need?. URL: https://arxiv.org/abs/2201.09792

40. Swin transformer: hierarchical vision transformer using shifted windows / Z. Liu et al. *2021 IEEE/CVF international conference on computer vision (ICCV)*, Montreal, QC, Canada, 10–17 October 2021. 2021. URL: https://doi.org/10.1109/iccv48922.2021.00986 (date of access: 19.05.2025).

41. Scaling local self-attention for parameter efficient visual backbones / A. Vaswani et al. *2021 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, Nashville, TN, USA, 20–25 June 2021. 2021. URL: https://doi.org/10.1109/cvpr46437.2021.01270 (date of access: 19.05.2025).

REFERENCES:

1. Ramchoun, H., Amine, M., Idrissi, J., Ghanou, Y., & Ettaouil, M. (2016). Multilayer perceptron: Architecture optimization and training. *International Journal of Interactive Multimedia and Artificial Intelligence, 4*(1), 26. https://doi.org/10.9781/ijimai.2016.415

2. Zhao, Z.-Q., Zheng, P., Xu, S.-T., & Wu, X. (2019). Object detection with deep learning: A review. *IEEE Transactions on Neural Networks and Learning Systems*, *30*(11), 3212–3232. https://doi.org/10.1109/tnnls.2018.2876865

3. Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE, 86*(11), 2278–2324. https://doi.org/10.1109/5.726791

4. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., & Fei-Fei, L. (2015). ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, *115*(3), 211–252. https://doi.org/10.1007/s11263-015-0816-y

5. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE conference on computer vision and pattern recognition (CVPR)*. IEEE. https://doi.org/10.1109/cvpr.2016.90

6. Vaswani, A. (2017). Attention is all you need. *NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems*. https://arxiv.org/abs/1706.03762

7. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021). *An image is worth 16x16 words: Transformers for image recognition at scale*. https://arxiv.org/abs/2010.11929

8. Dai, Z., Liu, H., Le, Q. V., & Tan, M. (2021). *CoAtNet: Marrying convolution and attention for all data sizes*. https://arxiv.org/abs/2106.04803

9. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2014). *Going deeper with convolutions*. https://arxiv.org/abs/1409.4842

10. Sharma, N., Jain, V., & Mishra, A. (2018). An analysis of convolutional neural networks for image classification. *Procedia Computer Science*, *132*, 377–384. https://doi.org/10.1016/j.procs.2018.05.198

11. Dumoulin, V., & Visin, F. (2018). *A guide to convolution arithmetic for deep learning*. https://arxiv.org/abs/1603.07285

12. Zafar, A., Aamir, M., Mohd Nawi, N., Arshad, A., Riaz, S., Alruban, A., Dutta, A. K., & Almotairi, S. (2022). A comparison of pooling methods for convolutional neural networks. *Applied Sciences*, *12*(17), 8643. https://doi.org/10.3390/app12178643

13. LeCun, Y., Kavukcuoglu, K., & Farabet, C. (2010). Convolutional networks and applications in vision. In *2010 IEEE international symposium on circuits and systems – ISCAS 2010*. IEEE. https://doi.org/10.1109/iscas.2010.5537907

14. Sharma, S., Sharma, S., & Athaiya, A. (2020). Activation functions in neural networks. *International Journal of Engineering Applied Sciences and Technology*, *04*(12), 310–316. https://doi.org/10.33564/ijeast.2020.v04i12.054

15. Chen, L., Li, S., Bai, Q., Yang, J., Jiang, S., & Miao, Y. (2021). Review of image classification algorithms based on convolutional neural networks. *Remote Sensing*, *13*(22), 4712. https://doi.org/10.3390/rs13224712

16. Wang, Q., Ma, Y., Zhao, K., & Tian, Y. (2020). A comprehensive survey of loss functions in machine learning. *Annals of Data Science*. https://doi.org/10.1007/s40745-020-00253-5

17. Soydaner, D. (2020). A comparison of optimization algorithms for deep learning. *International Journal of Pattern Recognition and Artificial Intelligence*, *34*(13), 2052013. https://doi.org/10.1142/s0218001420520138

18. Islam, M. R., & Matin, A. (2020). Detection of COVID 19 from CT Image by The Novel LeNet-5 CNN Architecture. In *2020 23rd international conference on computer and information technology (ICCIT)*. IEEE. https://doi.org/10.1109/iccit51783.2020.9392723

19. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84–90. https://doi.org/10.1145/3065386

20. Simonyan, K., & Zisserman, A. (2015). *Very deep convolutional networks for large-scale image recognition*. https://arxiv.org/abs/1409.1556

21. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *2016 IEEE conference on computer vision and pattern recognition (CVPR)*. IEEE. https://doi.org/10.1109/cvpr.2016.308

22. Ioffe, S., & Szegedy, C. (2015). *Batch normalization: Accelerating deep network training by reducing internal covariate shift*. https://arxiv.org/abs/1502.03167

23. Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. (2017). Inception-v4, inception-resnet and the impact of residual connections on learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, *31*(1). https://doi.org/10.1609/aaai.v31i1.11231

24. Lin, M., Chen, Q., & Yan, S. (2014). *Network in network*. https://arxiv.org/abs/1312.4400

25. Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)*. IEEE. https://doi.org/10.1109/cvpr.2017.243

26. Brauwers, G., & Frasincar, F. (2021). A general survey on attention mechanisms in deep learning. *IEEE Transactions on Knowledge and Data Engineering*, 1. https://doi.org/10.1109/tkde.2021.3126456

27. Wang, F., Jiang, M., Qian, C., Yang, S., Li, C., Zhang, H., Wang, X., & Tang, X. (2017). Residual attention network for image classification. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)*. IEEE. https://doi.org/10.1109/cvpr.2017.683

28. Hu, J., Shen, L., Albanie, S., Sun, G., & Wu, E. (2020). Squeeze-and-Excitation networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *42*(8), 2011–2023. https://doi.org/10.1109/tpami.2019.2913372

29. Park, J., Woo, S., Lee, J.-Y., & Kweon, I. S. (2018). *BAM: Bottleneck attention module*. https://arxiv.org/abs/1807.06514

30. Woo, S., Park, J., Lee, J.-Y., & Kweon, I. S. (2018). CBAM: Convolutional block attention module. In *Computer vision – ECCV 2018* (pp. 3–19). Springer International Publishing. https://doi.org/10.1007/978-3-030-01234-2_1

31. Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., & Keutzer, K. (2016). *SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size*. https://arxiv.org/abs/1602.07360

32. Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). *MobileNets: Efficient convolutional neural networks for mobile vision applications*. https://arxiv.org/abs/1704.04861

33. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2019). *MobileNetV2: Inverted residuals and linear bottlenecks*. https://arxiv.org/abs/1801.04381

34. Howard, A., Sandler, M., Chen, B., Wang, W., Chen, L.-C., Tan, M., Chu, G., Vasudevan, V., Zhu, Y., Pang, R., Adam, H., & Le, Q. (2019). Searching for MobileNetV3. In *2019 IEEE/CVF international conference on computer vision (ICCV)*. IEEE. https://doi.org/10.1109/iccv.2019.00140

35. Elsken, T., Metzen, J. H., & Hutter, F. (2019). Neural architecture search. In *Automated machine learning* (pp. 63–77). Springer International Publishing. https://doi.org/10.1007/978-3-030-05318-5_3

36. Yang, T.-J., Howard, A., Chen, B., Zhang, X., Go, A., Sandler, M., Sze, V., & Adam, H. (2018). NetAdapt: Platform-aware neural network adaptation for mobile applications. In *Computer vision – ECCV 2018* (pp. 289–304). Springer International Publishing. https://doi.org/10.1007/978-3-030-01249-6_18

37. Zhang, X., Zhou, X., Lin, M., & Sun, J. (2018). ShuffleNet: An extremely efficient convolutional neural network for mobile devices. In *2018 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*. IEEE. https://doi.org/10.1109/cvpr.2018.00716

38. Tan, M., & Le, Q. V. (2020). *EfficientNet: Rethinking model scaling for convolutional neural networks*. https://arxiv.org/abs/1905.11946

39. Trockman, A., & Kolter, J. Z. (2022). *Patches are all you need?* https://arxiv.org/abs/2201.09792

40. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., & Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *2021 IEEE/CVF international conference on computer vision (ICCV)*. IEEE. https://doi.org/10.1109/iccv48922.2021.00986

41. Vaswani, A., Ramachandran, P., Srinivas, A., Parmar, N., Hechtman, B., & Shlens, J. (2021). Scaling local self-attention for parameter efficient visual backbones. In *2021 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*. IEEE. https://doi.org/10.1109/cvpr46437.2021.01270