ChatGPT

Các thuật toán Deep Learning phân loại ảnh (2010–2025)

Tổng quan

Deep Learning đã cách mạng hóa bài toán phân loại ảnh, đặc biệt kể từ năm 2012 với sự ra đời của mạng nơ-ron tích chập (CNN) sâu. Nhiều kiến trúc mạng khác nhau đã lần lượt xuất hiện, mỗi kiến trúc lại mang đến những ý tưởng mới để cải thiện độ chính xác và hiệu quả. Dưới đây, chúng ta sẽ điểm qua **các mô hình phân loại ảnh sử dụng học sâu** nổi bật nhất từ sau năm 2010, bao gồm cả các **biến thể** của chúng, và phân loại theo từng nhóm chính. Danh sách bao gồm cả **mạng CNN truyền thống**, các **mạng hiệu quả dành cho thiết bị di động**, các kiến trúc thiết kế tự động bằng **NAS**, cũng như các mô hình **Transformer thị giác** hiện đại. Cuối cùng, bài viết cũng đề cập các **thư viện Python** phổ biến hỗ trợ sẵn những mô hình này.

Bảng dưới đây tóm tắt các mô hình chính và đặc điểm nổi bật:

Kiến trúc	Năm giới thiệu	Đặc điểm chính	Phiên bản/biến thể tiêu biểu
AlexNet	2012	CNN 8 tầng, ReLU, dropout; thắng giải ImageNet 2012	–
VGGNet	2014	CNN rất sâu (16–19 tầng) với bộ lọc 3×3 nhỏ [1]; đơn giản, hiệu năng cao	VGG-11, 13, **16**, **19**
Inception (GoogLeNet)	2014–2015	Module "Inception" với tích chập đa tỉ lệ song song; tận dụng tài nguyên hiệu quả [2]	**GoogLeNet** (Inception v1) 22 tầng [3]; Inception v2, v3 (2015); Inception v4, Inception-ResNet (2016)
ResNet	2015	Kết nối thặng dư (skip connection) giúp huấn luyện mạng cực sâu dễ dàng [4]; thắng ImageNet 2015	ResNet-18, 34, **50**, **101**, **152**; ResNet-v2 (2016, pre-activation); *Variants:* **ResNeXt** (2017, thêm "cardinality") [5]; **Wide ResNet** (2016, giảm sâu tăng rộng) [6]; **ResNet-D**, **SENet** (2017, attention); **PURe** (2025, tích chập nhân) [7]
DenseNet	2017	Mỗi tầng nối kết đầy đủ với *tất cả* tầng trước (dense connectivity), tăng cường tái sử dụng đặc trưng [8]	DenseNet-121, 169, 201, 264

Kiến trúc	Năm giới thiệu	Đặc điểm chính	Phiên bản/biến thể tiêu biểu
SqueezeNet	2016	Mạng siêu nhỏ gọn với module "Fire" (1×1 + 3×3 conv); đạt độ chính xác tương đương AlexNet nhưng số tham số chỉ bằng 1/50 [9]	SqueezeNet v1.0, v1.1
MobileNet	2017–2019	Sử dụng tích chập tách chiều sâu (depthwise separable) để giảm tính toán [10]; tối ưu cho thiết bị di động	**MobileNet v1** (2017); **v2** (2018, block "inverted residual" với bottleneck tuyến tính) [11]; **v3** (2019, tìm kiến trúc tự động NAS + NetAdapt) – gồm bản Large/Small [12]
ShuffleNet	2018	Sử dụng tích chập nhóm (grouped conv) kết hợp hoán vị kênh (channel shuffle) để giảm FLOPs [13]; tối ưu cho di động	ShuffleNet v1 (2018); **v2** (2018, cải tiến v1, đơn giản hóa kiến trúc)
NASNet	2018	Kiến trúc do *Neural Architecture Search* (NAS) tìm ra trên tập CIFAR, rồi nhân rộng lên ImageNet; đạt SOTA ImageNet 82.7% top-1 [14]	NASNet-A Large (ImageNet) & NASNet-Mobile
EfficientNet	2019	Kết hợp NAS và chiến lược *compound scaling* (phối hợp tăng đồng thời độ sâu, rộng, độ phân giải) [15] [16]; hiệu quả vượt trội so với các CNN trước đó	EfficientNet-B0 … **B7** (B7 đạt 84.3% top-1 ImageNet [16]); EfficientNetV2 (2021, tăng tốc độ huấn luyện)
Vision Transformer (ViT)	2020	Áp dụng hoàn toàn kiến trúc Transformer (self-attention) trên patch ảnh 16×16 [17]; cần tiền huấn luyện trên dữ liệu lớn, cho kết quả cạnh tranh với CNN SOTA	ViT-B/16, ViT-L/16, ViT-H/14 (các kích thước mô hình); *Variants:* DeiT (2021, huấn luyện hiệu quả dữ liệu)
Swin Transformer	2021	Transformer phân cấp (hierarchical) dùng cửa sổ trượt (*Shifted Windows*) để tự chú ý cục bộ hiệu quả [18] [19]; linh hoạt cho cả phân loại và nhận dạng đối tượng	Swin-Tiny, Small, Base, Large (các phiên bản kích thước mô hình)
ConvNeXt	2022	Kiến trúc ConvNet được thiết kế lại theo kinh nghiệm từ Transformers (kernel 7x7 lớn, bỏ tụt cực đại, chuẩn hóa LayerNorm, v.v.) để đạt hiệu năng ngang ViT	ConvNeXt-Tiny đến ConvNeXt-Large; ConvNeXt V2 (2023)

Kiến trúc	Năm giới thiệu	Đặc điểm chính	Phiên bản/biến thể tiêu biểu
Các kiến trúc mới khác	2021–2025	– MLP-Mixer, ResMLP (2021): sử dụng hoàn toàn lớp kết nối đầy đủ (MLP) để trộn đặc trưng, thử thách sự cần thiết của tích chập/attention. – **CoAtNet** (2021): kết hợp CNN và Transformer, đạt ~86% ImageNet. – **PURe** (2025): thay thế conv bằng *product units* trong block ResNet, cải thiện biểu đạt và độ hiệu quả [20]	MLP-Mixer, ResMLP, gMLP; CoAtNet-0..4; **PURe-34**, PURe-50,... PURe-272 (đạt 95% CIFAR-10 với nửa tham số ResNet1001 [21])

(Bảng: Các mô hình phân loại ảnh deep learning chính từ 2012–2025, kèm điểm nổi bật và phiên bản)

Thời kỳ đầu (2012–2014): Các CNN đột phá đầu tiên

AlexNet (2012) – Được coi là mốc khởi đầu của kỷ nguyên deep learning cho thị giác. AlexNet là mô hình CNN 8 tầng (5 tầng tích chập và 3 tầng fully-connected) do Alex Krizhevsky và cộng sự phát triển [22] . Mạng này đã chiến thắng cuộc thi ImageNet ILSVRC 2012 với sai số top-5 chỉ ~18.9%, vượt xa mô hình truyền thống trước đó [23] . Những đóng góp chính của AlexNet gồm dùng hàm kích hoạt ReLU, dropout để tránh overfit, và huấn luyện song song trên GPU. Thành công của AlexNet đã **chứng minh hiệu quả của mạng CNN sâu** trên dữ liệu hình ảnh lớn, mở đường cho hàng loạt mô hình sâu ra đời sau đó.

VGGNet (2014) – Nhóm Visual Geometry Group (Oxford) giới thiệu VGGNet nhằm nghiên cứu ảnh hưởng của **độ sâu mạng** đến độ chính xác phân loại [24] . VGGNet sử dụng cấu trúc đơn giản: chỉ các tích chập kích thước nhỏ 3×3 nhưng xếp chồng rất nhiều lớp. Kết quả cho thấy tăng độ sâu lên 16–19 lớp (**VGG16**, VGG19) đem lại cải thiện đáng kể về độ chính xác so với mô hình nông hơn [1] . Mô hình VGG đạt giải nhì phân loại và giải nhất phát hiện trong ILSVRC 2014 [25] . VGGNet tuy nhiều tham số (≈138 triệu) nhưng kiến trúc gọn gàng, trở thành mô hình nền tảng cho nhiều ứng dụng thị giác sau này. Hiện nay VGG16/19 vẫn được dùng rộng rãi làm backbone khởi tạo cho các bài toán transfer learning.

GoogLeNet/Inception (2014) – Được đề xuất bởi Christian Szegedy và các cộng sự tại Google, mạng "Inception" mang kiến trúc gọi là **GoogLeNet** (đặt theo tên Google) đã *đoạt giải nhất phân loại ILSVRC 2014* [2] . Điểm độc đáo của Inception là **module Inception**: thay vì tuần tự từng lớp, module này gồm nhiều nhánh tích chập song song (1×1, 3×3, 5×5) và pooling, cho phép trích xuất đặc trưng ở nhiều mức độ phân giải khác nhau rồi nối lại [2] . Cách thiết kế này **tận dụng hiệu quả tính toán** hơn, giúp GoogLeNet rất sâu (22 lớp) nhưng số tham số ít (chỉ ~4 triệu) [3] . Sau phiên bản đầu, Google tiếp tục phát triển **Inception v2, v3** (2015) – bổ sung Batch Normalization, factorized conv,... – và **Inception v4, Inception-ResNet** (2016) – kết hợp thêm ý tưởng residual. Các phiên bản Inception mới liên tục đạt kết quả cao trên ImageNet [26] [27] . Ngoài ra **Xception (2017)** của F. Chollet cũng là biến thể của Inception, khi thay các nhánh bằng tích chập tách kênh hoàn toàn (depthwise separable) – Xception đạt kết quả nhỉnh hơn Inception v3 trên ImageNet với cùng số tham số [28] [29] .

ResNet và các biến thể (2015–2016)

ResNet (2015) – Kaiming He và các cộng sự Microsoft đã tạo ra một cuộc cách mạng với **Residual Network (ResNet)** – mạng tích chập cực sâu nhưng **dễ huấn luyện nhờ cơ chế kết nối thặng dư** [30]. Ý tưởng chính là thêm đường tắt (*skip connection*) mang thẳng đầu vào của một vài lớp cộng vào đầu ra, giúp mạng chỉ cần học phần *chênh lệch (residual)* giữa input-output thay vì học toàn bộ hàm mong muốn [30]. Nhờ đó, ResNet giải quyết được hiện tượng suy giảm độ chính xác khi mạng quá sâu. Nhóm tác giả đã huấn luyện ResNet với độ sâu kỷ lục 152 lớp (sâu gấp 8 lần VGG) mà vẫn tăng độ chính xác, đạt lỗi top-5 3.57% và **đoạt giải nhất ILSVRC 2015** [4]. ResNet có nhiều cấu hình: thường dùng nhất là ResNet-18, 34, 50, 101 và 152 (số lớp). ResNet-50/101/152 được huấn luyện sẵn rất phổ biến làm backbone trong thị giác máy tính.

Biến thể của ResNet: Thành công của ResNet kéo theo nhiều nghiên cứu mở rộng:

- **ResNet v2 (2016)** – Cũng do Kaiming He đề xuất, sử dụng cấu trúc *pre-activation* (đưa BatchNorm+ReLU lên trước conv) giúp gradient truyền tốt hơn. ResNet-v2 được dùng trong bản mở rộng 1001 lớp trên CIFAR-10 [31].

- **Wide ResNet (WRN, 2016)** – Zagoruyko và Komodakis nhận ra ResNet rất sâu hiệu quả giảm dần, nên **giảm độ sâu, tăng độ rộng** mỗi layer để cải thiện tốc độ và độ chính xác [32]. Ví dụ WRN-16-8 (16 lớp, rộng gấp 8) cho kết quả tốt hơn ResNet-1000-lớp trên CIFAR và ImageNet với ít thời gian huấn luyện hơn [6].

- **ResNeXt (2017)** – Là biến thể của ResNet do Facebook AI (Xie et al.) đề xuất, giới thiệu thêm một siêu tham số là **"cardinality" (số nhánh tích chập song song)** trong mỗi block [33] [34]. Mạng ResNeXt tăng số nhánh song song thay vì chỉ tăng chiều rộng, cho hiệu quả vượt trội: ResNeXt-50 (32×4d) đạt độ chính xác tương đương ResNet-101 nhưng ít tham số hơn [35]. ResNeXt là nền tảng cho bài thi ImageNet 2016 (đạt hạng 2) [5].

- **SENet (2017)** – Hu et al. giới thiệu module Squeeze-and-Excitation (SE) chèn vào ResNet để học trọng số kênh (channel attention). *SENet-154* thắng ILSVRC 2017, đạt 82.7% top-1 ImageNet, tương đương NASNet cùng năm [14].

- **ResNet-D, E** – Các biến thể cải tiến kiến trúc downsampling của ResNet (như ResNet-D: dùng avg pooling trước conv 1×1) để tăng hiệu quả cho nhận dạng vật thể [36].

- **Các biến thể khác**: *Dual-Path Net (DPN, 2017)* kết hợp DenseNet + ResNet; *ECA-ResNet (2020)* tối ưu SE block; *ResNeSt (2020)* thêm split-attention trong ResNet; v.v. Gần đây nhất, **PURe (2025)** – *Product-Unit Residual Network* – thay thế lớp tích chập thứ hai trong block ResNet bằng *nhân tử sản phẩm* (product units) và bỏ hàm phi tuyến, giúp mạng học các tương tác bội số phức tạp [37] [20]. Thử nghiệm cho thấy PURe-34 đạt **80.27% top-1 trên ImageNet**, cao hơn ResNet-50/101 với ít tham số hơn, và **PURe-272 trên CIFAR-10 đạt 95.0%** tương đương ResNet-1001 nhưng kích thước chỉ bằng một nửa [7]. PURe cho thấy tiềm năng cải thiện hiệu năng và độ *robust* của ResNet truyền thống.

Mạng kết nối dày đặc và hiệu quả (2016–2018)

DenseNet (2017) – Gao Huang và cộng sự đề xuất **Dense Convolutional Network** với mục tiêu tận dụng triệt để đặc trưng. Thay vì chỉ nối đầu ra tầng $(l-1)$ vào tầng l, DenseNet **kết nối mỗi tầng với tất cả các tầng sau nó** – tức tầng l nhận đầu vào là đầu ra của *toàn bộ* các tầng trước đó [8]. Nếu

mạng có L tầng thì có tới $L(L+1)/2$ kết nối trực tiếp thay vì L như thường lệ [38]. Nhờ thiết kế này, DenseNet giải quyết tốt vấn đề mất mát gradient, khuyến khích tái sử dụng đặc trưng và giảm đáng kể số tham số so với mạng dày tương đương [39]. Trên các benchmark như CIFAR-10/100, SVHN, ImageNet, DenseNet đạt kết quả **tốt hơn ResNet** dù dùng ít tính toán hơn [40]. Các cấu hình phổ biến: DenseNet-121, 169, 201 (số lớp) – những mô hình này cũng có sẵn trong các thư viện và hay dùng trong transfer learning.

SqueezeNet (2016) – Forrest Iandola và các đồng nghiệp giới thiệu SqueezeNet với mục tiêu "**nhỏ mà có võ**": đạt độ chính xác tương đương AlexNet nhưng với số tham số *nhỏ hơn 50 lần* [9]. SqueezeNet chỉ ~1.3 triệu tham số (~0.5MB sau nén), thích hợp cho nhúng. Mô hình sáng tạo module "Fire" gồm một lớp "squeeze" 1×1 giảm kênh, sau đó là lớp "expand" gồm tích chập 1×1 và 3×3 để trích đặc trưng. Kiến trúc này giảm mạnh tham số mà vẫn giữ hiệu quả. Nhờ kích thước nhỏ, SqueezeNet được dùng trong nhiều ứng dụng di động, IoT.

ShuffleNet (2018) – Để phục vụ thiết bị di động, nhóm Megvii (Zhang et al.) đưa ra ShuffleNet – một kiến trúc *cực kỳ hiệu quả* về tính toán [41]. ShuffleNet sử dụng **tích chập nhóm** (group convolution) để giảm MACs, kết hợp với thao tác **hoán vị kênh (channel shuffle)** để trộn lẫn thông tin giữa các nhóm [41]. Kết quả, tại mức ngân sách 40 MFLOPs, ShuffleNet cho lỗi top-1 ImageNet thấp hơn MobileNet khoảng 7.8% (điểm tuyệt đối) [42]. Trên thiết bị ARM, ShuffleNet chạy **nhanh hơn AlexNet ~13 lần** với độ chính xác tương đương [43]. Phiên bản **ShuffleNet v2** (Ma et al. ECCV 2018) đưa ra một bộ hướng dẫn thiết kế mạng hiệu quả (như tránh tắc nghẽn kênh, cân bằng chi phí giữa các phần), và một kiến trúc cải tiến đơn giản hơn, hiệu năng cao hơn v1.

MobileNet (2017–2019) – Dòng MobileNet (Google) nhắm vào tối ưu mô hình trên mobile. **MobileNet v1 (2017)** đề xuất **tích chập tách biệt chiều sâu**: tách một conv chuẩn thành conv theo *từng kênh* (depthwise) + conv 1×1 (pointwise), giảm lượng tính toán và tham số đi ~8–9 lần với ít giảm độ chính xác [10]. MobileNet v1 đạt ~70% top-1 ImageNet với mô hình chỉ 4.2 triệu tham số. Họ cũng đưa vào 2 tham số *width multiplier* và *resolution multiplier* để linh hoạt đổi cân bằng tốc độ/chính xác [44]. **MobileNet v2 (2018)** tiếp tục cải tiến với **block inverted residual**: thay vì giảm dần kênh rồi tích chập, MBv2 *mở rộng kênh rồi áp dụng depthwise conv, sau đó nén lại*, và đặc biệt **bỏ kích hoạt phi tuyến ở lớp hẹp** để không làm mất thông tin [11]. Thiết kế này cải thiện cả độ chính xác lẫn khả năng trích đặc trưng [11]. MobileNet v2 đạt ~74% top-1 với 6.9M tham số, trở thành backbone phổ biến cho detector (SSDlite). **MobileNet v3 (2019)** sử dụng kỹ thuật **NAS (NetAdapt + Mạng học kiến trúc)** để tìm cấu trúc tối ưu cho phần cứng mobile [45]. Kết hợp với các module SE và thủ thuật swish activation, MobileNet v3 có hai phiên bản: Large (~75% top-1) và Small (~68%), đều nhanh và gọn hơn v2 [12]. So với MobileNet v2, v3-Large tăng 3.2% chính xác *và* giảm 15% độ trễ suy luận trên điện thoại [12].

Ngoài ra, **MNASNet (2018)** cũng là mạng do NAS tìm ra tập trung cho mobile (Mingxing Tan et al.), đạt ~75% top-1 với 4.4M tham số, là tiền thân ý tưởng cho MobileNetV3. Cùng với đó, **EfficientNet** sau này (xem phần dưới) cũng nhắm đến hiệu quả cao trên thiết bị tính toán hạn chế.

Kiến trúc do AutoML thiết kế (2018–2019)

NASNet (2017/2018) – "Network Architecture Search Network" của Google Brain (Barret Zoph et al.) là mô hình tiên phong áp dụng AutoML để thiết kế kiến trúc CNN. Thay vì con người thử nghiệm thủ công, NASNet dùng thuật toán **reinforcement learning** để tìm ra **cell tích chập tối ưu trên tập CIFAR-10**, sau đó **chuyển cell này sang ImageNet** bằng cách xếp chồng nhiều cell hơn [46] [47]. Kết quả rất ấn tượng: *NASNet-A Large* (~88M tham số) đạt **82.7% top-1 / 96.2% top-5 trên ImageNet** – cao hơn ~1.2% so với ResNeXt/SENet cùng thời, trong khi *FLOPs giảm 28%* [14]. Thậm chí một phiên bản NASNet nhỏ

hơn (NASNet-Mobile, ~5M tham số) cũng đạt 74% top-1, vượt các model human-designed tương đương kích thước ~3% [48]. NASNet chứng minh sức mạnh của AutoML trong thiết kế mạng tối ưu.

AmoebaNet (2018) – Song song với NASNet, Google Brain còn thử phương pháp tiến hóa (evolutionary algorithm) để tìm kiến trúc, cho ra AmoebaNet. Phiên bản AmoebaNet-A (466M FLOPs) đạt 82.8% top-1 ImageNet, tương đương NASNet. Dù tốn nhiều tài nguyên tìm kiếm, những mạng như NASNet, AmoebaNet đã *thiết lập kỷ lục mới* về phân loại ảnh năm 2018.

EfficientNet (2019) – Một bước tiến lớn của Google (M. Tan & Q. Le) tổng hợp cả NAS lẫn kinh nghiệm thiết kế thủ công. Đầu tiên, họ dùng NAS tìm một mô hình CNN cơ bản hiệu quả (tên là EfficientNet-B0) [49]. Sau đó, họ áp dụng chiến lược **compound scaling** – tức mở rộng đồng thời độ sâu, độ rộng và độ phân giải đầu vào theo các hệ số cố định – để tạo ra cả **hệ model EfficientNet B1–B7** lớn dần [15]. Cách scaling này giúp tận dụng tài nguyên hiệu quả hơn so với chỉ tăng một chiều. Kết quả, EfficientNet thiết lập mức SOTA mới: EfficientNet-B7 (66M tham số, 37B FLOPs) đạt **84.3% top-1 ImageNet** – cao hơn mọi CNN trước đó, *nhanh hơn 6.1× và nhỏ hơn 8.4× so với* mạng tốt nhất trước (so với SENet hoặc AmoebaNet) [16]. Ngay cả các bản nhỏ hơn như B0 (5.3M tham số) cũng cho độ chính xác vượt trội so với MobileNet cùng mức tài nguyên. EfficientNet còn cho thấy khả năng **transfer learning rất tốt**: ví dụ trên Flowers dataset đạt 98.8% với ít tham số hơn hẳn mô hình cũ [50]. Năm 2021, nhóm tác giả ra **EfficientNetV2**, cải tiến module MBConv, kết hợp thêm Fused-MBConv, và tối ưu quá trình huấn luyện (Progressive Learning) để tăng tốc độ huấn luyện **nhanh hơn ~2×** so với EfficientNet ban đầu, đồng thời cải thiện nhẹ độ chính xác.

Thời đại Transformer và các kiến trúc mới (2020–2025)

Vision Transformer (ViT, 2020) – Năm 2020, một nhóm Google Brain (Dosovitskiy et al.) đã mở ra xu hướng mới khi áp dụng thành công **Transformer** thuần túy cho phân loại ảnh. Mô hình của họ, gọi là **Vision Transformer**, chia ảnh thành các patch cố định (ví dụ 16×16) rồi đưa tuần tự qua nhiều lớp self-attention giống như xử lý chuỗi từ ngữ [51] [17]. ViT chứng minh rằng **CNN không còn là bắt buộc**: với dữ liệu huấn luyện đủ lớn (như JFT-300M) để pre-train, sau đó fine-tune, **một transformer thuần có thể đạt kết quả xuất sắc ngang hoặc trội hơn SOTA CNN** trên ImageNet và các bộ dữ liệu trung bình khác [17] [52]. Cụ thể, ViT-B/16 (86M tham số) đạt ~77.9% top-1 trên ImageNet khi train từ đầu, và tới ~85.0% nếu pre-train trên JFT rồi fine-tune – tiệm cận EfficientNet-B7 nhưng **train nhanh hơn nhiều** vì tính chất Transformer cho phép song song tốt [17]. ViT mở ra làn sóng nghiên cứu mô hình thị giác dựa trên self-attention.

Các biến thể Transformer: Sau ViT, hàng loạt mô hình dựa trên attention ra đời:

- **DeiT (2021)** – Facebook AI giới thiệu DeiT (Touvron et al.), một phiên bản ViT *huấn luyện hiệu quả trên ImageNet 1M ảnh từ đầu* (không cần data khổng lồ) bằng cách dùng mẹo data augmentation mạnh và một token giáo viên đặc biệt. DeiT-Small (22M tham số) vẫn đạt ~79.8% top-1 chỉ sau vài ngày train trên 8 GPU.

- **Swin Transformer (2021)** – Mô hình **Shifted Windows Transformer** của Microsoft (Liu et al.) khắc phục điểm yếu của ViT khi áp dụng cho ảnh độ phân giải cao. Swin chia ảnh thành các cửa sổ nhỏ và chỉ tính self-attention *bên trong từng ô*, sau mỗi tầng lại "trượt" cửa sổ để kết nối giữa các vùng [18]. Cấu trúc **phân cấp** này giúp Swin Transformer *linh hoạt như CNN*: có đầu ra nhiều mức độ phân giải (hữu ích cho detection/segmentation) và độ phức tạp tính toán tăng **tuyến tính** theo số pixel (thay vì bình phương như ViT) [18] [53]. Swin Transformer đạt **87.3% top-1 trên ImageNet-1K** với mô hình Swin-L, vượt EfficientNet và ViT cùng kích thước [19]. Đặc biệt trong

các tác vụ phát hiện và phân đoạn, Swin cũng **vượt SOTA biên độ lớn** và trở thành backbone Transformer đầu tiên thực sự **đa năng** cho thị giác [19].

- **Hybrid Convolution + Transformer:** Một số mô hình kết hợp cả CNN và Transformer để tận dụng ưu điểm đôi bên. Ví dụ **CoAtNet (2021)** của Google (Dai et al.) xếp các giai đoạn tích chập (giai đoạn đầu bắt đặc trưng cục bộ) và giai đoạn tự chú ý (giai đoạn sau nắm quan hệ rộng), đạt ~86% ImageNet với model ~81M tham số – hiệu quả hơn ViT thuần cùng quy mô. Các mô hình khác như BoTNet (2021) thay block ResNet cuối bằng multi-head attention, hoặc ConvFormer, CMT, CvT... cũng khám phá hướng lai ghép này.

- **MLP-based models (2021)** – Một trào lưu thú vị năm 2021 là thử **loại bỏ hoàn toàn cả tích chập lẫn attention**, chỉ dùng mạng *MLP thuần* để trộn lẫn thông tin theo không gian và kênh. Ví dụ **MLP-Mixer** (Tolstikhin et al., 2021) tách patch rồi áp dụng MLP theo chiều không gian (trộn các patch) và MLP theo chiều kênh luân phiên. Hay **ResMLP, gMLP** tương tự. Các mô hình MLP này đạt ~72–78% top-1 ImageNet – chưa vượt CNN/Transformer hiện tại nhưng cho thấy một hướng mới đơn giản hơn.

- **ConvNeXt (2022)** – Sau sự thống trị của Transformer, một nhóm Facebook (Liu et al.) đã *xem xét lại kiến trúc ConvNet* truyền thống và đưa ra **ConvNeXt** – một ResNet được hiện đại hóa để cạnh tranh với ViT. ConvNeXt áp dụng loạt cải tiến: sử dụng chuẩn hóa LayerNorm thay cho BatchNorm, bỏ tầng pool, tăng kernel conv lên 7×7, dùng Activation GELU, kiến trúc kiểu Swin (chia stage, không nối skip từng block)... Kết quả ConvNeXt-L đạt 87.8% top-1 (bằng Swin-L) với tốc độ suy luận nhanh hơn, cho thấy CNN vẫn có thể cạnh tranh nếu được tối ưu đúng cách. ConvNeXt đánh dấu **sự trở lại của ConvNet** trong bối cảnh Transformer.

- **Các hướng mới (2023–2025):** Nghiên cứu mô hình nền vẫn tiếp tục sôi động. Năm 2023, Meta AI công bố **ConvNeXt V2** tiếp tục cải tiến kernel và kỹ thuật huấn luyện cho ConvNeXt. Google giới thiệu **MAX-ViT** – một Transformer kết hợp attention không gian và channel theo khối, rất hiệu quả (86.5% top-1 với 212M tham số). Xu hướng xây dựng mô hình ngày càng *lớn hơn* trên tập dữ liệu *khổng lồ* cũng phổ biến (ví dụ **ViT-22B** với 22 tỷ tham số của Google vào 2022). Bên cạnh đó, các nghiên cứu như **PURe (2025)** đã nêu ở trên cho thấy những ý tưởng *đổi mới cấu trúc bên trong mạng CNN* (như dùng nhân tử tích) vẫn có khả năng đem lại cải thiện đáng kể [20]. Điều này gợi rằng sự phân hóa mô hình tương lai có thể đa dạng: *không chỉ CNN hay Transformer*, mà có thể là kết hợp hoặc những kiểu mạng mới tối ưu hơn cho thị giác.

Hỗ trợ trong các thư viện Python

Hầu hết các mô hình nổi tiếng kể trên đều được tích hợp sẵn trong các thư viện deep learning phổ biến, giúp người dùng dễ dàng sử dụng pre-trained models:

- **PyTorch (TorchVision)**: Thư viện `torchvision.models` cung cấp nhiều mô hình tiền huấn luyện trên ImageNet, bao gồm **AlexNet, VGG-16/19, ResNet-50/101/152, SqueezeNet, DenseNet-121/169/201, Inception v3, GoogLeNet, ShuffleNet v2, MobileNet v2/v3, ResNeXt-50/101, Wide ResNet-50, MNASNet, EfficientNet (B0–B7), RegNet (Y/X)**... hầu như đầy đủ các mạng đã đề cập [54] [55]. Người dùng chỉ cần gọi hàm tương ứng (vd: `models.resnet50(pretrained=True)`) là có ngay mô hình với trọng số ImageNet tải về [56] [57].

- **TensorFlow/Keras**: Tương tự, **tf.keras.applications** cũng hỗ trợ nhiều mô hình kinh điển: **Xception, VGG16/19, ResNet50/V2, ResNet101/V2, InceptionV3, Inception-ResNetV2,**

MobileNet(v1/v2), DenseNet-121/169/201, NASNetLarge/Mobile, EfficientNet-B0...B7, v.v. (tùy phiên bản TensorFlow). Chỉ với vài dòng lệnh (vd: `tf.keras.applications.EfficientNetB0(weights="imagenet")`), ta có thể tải mô hình với trọng số đã huấn luyện. Nhờ đó, các lập trình viên có thể tận dụng sức mạnh các mô hình deep learning này dễ dàng cho bài toán phân loại ảnh của mình.

- **Thư viện khác**: Ngoài ra còn có những dự án chuyên cung cấp nhiều mô hình kiến trúc hơn nữa, chẳng hạn `timm` **(PyTorch Image Models)** – tập hợp hàng trăm mô hình từ cổ điển đến SOTA do cộng đồng đóng góp (bao gồm cả các biến thể mới như ViT, Swin, ConvNeXt, v.v.). Thư viện HuggingFace Transformers cũng đã hỗ trợ các mô hình vision như ViT, Swin Transformer, BEiT,... với giao diện thống nhất.

Tóm lại, kho tàng mô hình deep learning phân loại ảnh rất phong phú và không ngừng mở rộng. Từ các CNN kinh điển cho đến Transformer hiện đại, mỗi kiến trúc đều đóng góp những tiến bộ riêng. Hiểu rõ đặc điểm và phiên bản của từng mạng giúp chúng ta lựa chọn mô hình phù hợp cho từng ứng dụng cụ thể, cũng như nắm bắt xu hướng phát triển của thị giác máy tính trong kỷ nguyên trí tuệ nhân tạo. [17] [7]

Tài liệu tham khảo:

1. Krizhevsky *et al.* (2012), *ImageNet Classification with Deep CNNs* – Advances in NIPS 25 [22].
2. Simonyan & Zisserman (2014), *Very Deep ConvNets for Large-Scale Image Recognition* – arXiv: 1409.1556 [1].
3. Szegedy *et al.* (2014), *Going Deeper with Convolutions (Inception v1/GoogLeNet)* – arXiv:1409.4842 [2].
4. Szegedy *et al.* (2015), *Rethinking the Inception Architecture for Computer Vision (Inc v2/3)* – arXiv: 1512.00567 [58].
5. Szegedy *et al.* (2016), *Inception-v4, Inception-ResNet and Impact of Residual Connections* – arXiv: 1602.07261 [26] [27].
6. Chollet (2017), *Xception: Deep Learning with Depthwise Separable Convolutions* – arXiv:1610.02357 [28] [29].
7. He *et al.* (2015), *Deep Residual Learning for Image Recognition (ResNet)* – arXiv:1512.03385 [4].
8. Zagoruyko & Komodakis (2016), *Wide Residual Networks* – arXiv:1605.07146 [32].
9. Xie *et al.* (2017), *Aggregated Residual Transformations for Deep CNNs (ResNeXt)* – arXiv:1611.05431 [34].
10. Huang *et al.* (2017), *Densely Connected Convolutional Networks (DenseNet)* – CVPR 2017 [8] [39].
11. Iandola *et al.* (2016), *SqueezeNet: AlexNet-level accuracy with 50× fewer parameters* – arXiv: 1602.07360 [9].
12. Zhang *et al.* (2018), *ShuffleNet: An Extremely Efficient CNN for Mobile* – CVPR 2018 [13].
13. Howard *et al.* (2017), *MobileNets: Efficient CNNs for Mobile Vision* – arXiv:1704.04861 [10].
14. Sandler *et al.* (2018), *MobileNetV2: Inverted Residuals and Linear Bottlenecks* – CVPR 2018 [11].
15. Howard *et al.* (2019), *Searching for MobileNetV3* – ICCV 2019 [12].
16. Zoph *et al.* (2018), *Learning Transferable Architectures for Scalable Image Recognition (NASNet)* – CVPR 2018 [14].
17. Tan & Le (2019), *EfficientNet: Rethinking Model Scaling for ConvNets* – ICML 2019 [16].
18. Dosovitskiy *et al.* (2020), *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale (ViT)* – ICLR 2021 [17].
19. Liu *et al.* (2021), *Swin Transformer: Hierarchical Vision Transformer* – ICCV 2021 [18] [19].
20. Li *et al.* (2025), *Deep Residual Learning with Product Units (PURe)* – arXiv:2505.04397 [20].

1 24 25 [1409.1556] Very Deep Convolutional Networks for Large-Scale Image Recognition
https://arxiv.org/abs/1409.1556

2 3 [1409.4842] Going Deeper with Convolutions
https://arxiv.org/abs/1409.4842

4 30 [1512.03385] Deep Residual Learning for Image Recognition
https://arxiv.org/abs/1512.03385

5 33 34 35 [1611.05431] Aggregated Residual Transformations for Deep Neural Networks
https://arxiv.org/abs/1611.05431

6 32 [1605.07146] Wide Residual Networks
https://arxiv.org/abs/1605.07146

7 20 21 31 37 [2505.04397] Deep residual learning with product units
https://www.arxiv.org/abs/2505.04397

8 38 39 40 [1608.06993] Densely Connected Convolutional Networks
https://arxiv.org/abs/1608.06993

9 [1602.07360] SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size
https://arxiv.org/abs/1602.07360

10 44 [1704.04861] MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications
https://arxiv.org/abs/1704.04861

11 [1801.04381] MobileNetV2: Inverted Residuals and Linear Bottlenecks
https://arxiv.org/abs/1801.04381

12 45 [1905.02244] Searching for MobileNetV3
https://arxiv.org/abs/1905.02244

13 41 42 43 [1707.01083] ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices
https://arxiv.org/abs/1707.01083

14 46 47 48 [1707.07012] Learning Transferable Architectures for Scalable Image Recognition
https://arxiv.org/abs/1707.07012

15 16 49 50 [1905.11946] EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks
https://arxiv.org/abs/1905.11946

17 51 52 [2010.11929] An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale
https://arxiv.org/abs/2010.11929

18 19 53 [2103.14030] Swin Transformer: Hierarchical Vision Transformer using Shifted Windows
https://arxiv.org/abs/2103.14030

22 23 ImageNet Classification with Deep Convolutional Neural Networks
https://papers.nips.cc/paper_files/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html

26 27 [1602.07261] Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning
https://arxiv.org/abs/1602.07261

28 29 [1610.02357] Xception: Deep Learning with Depthwise Separable Convolutions
https://arxiv.org/abs/1610.02357

36 ResNet-D: A Deep Learning Architecture for Improved ... - SERP AI
https://serp.ai/posts/resnet-d/

54 55 56 57 torchvision.models — Torchvision 0.11.0 documentation
https://docs.pytorch.org/vision/0.11/models.html

58 [1512.00567] Rethinking the Inception Architecture for Computer Vision
https://arxiv.org/abs/1512.00567