



Vision Transformer và So sánh với CNN

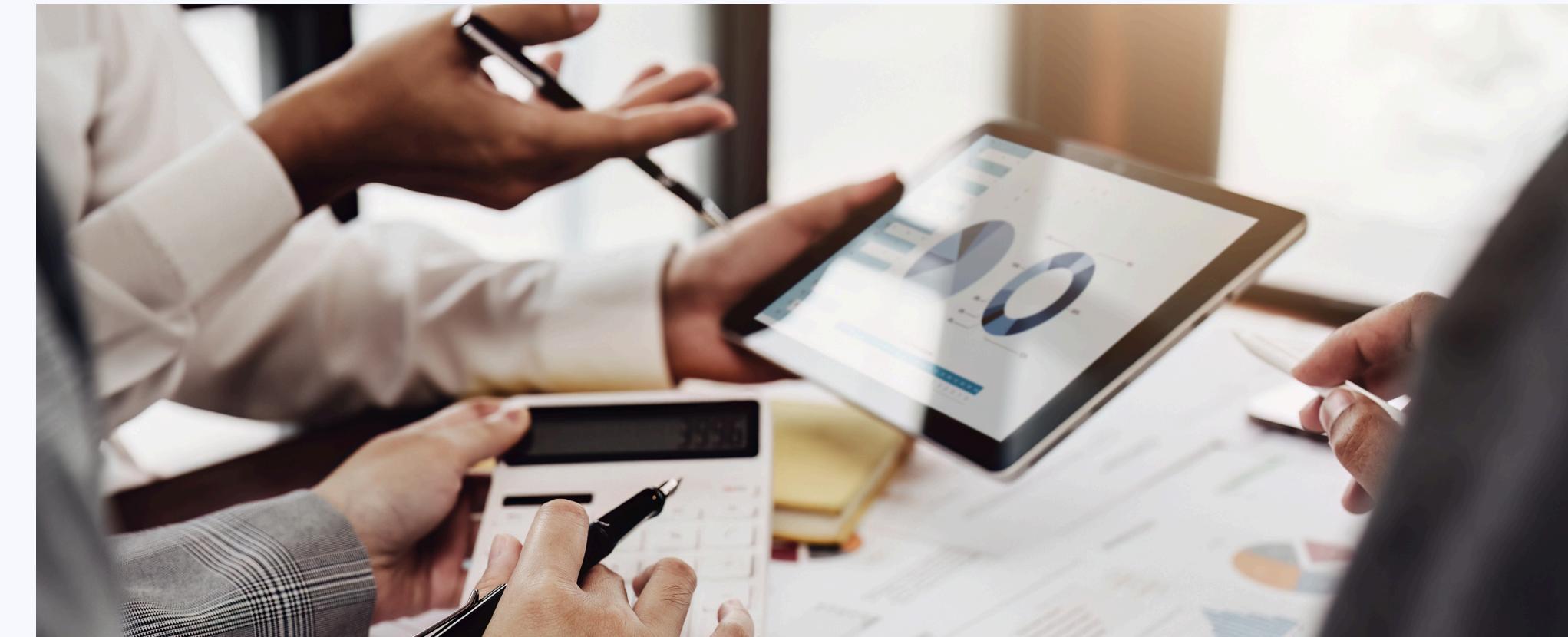
Phạm Duy Long

CS338.P22

Mục lục

-
- 01 Giới thiệu: Bối cảnh và động lực nghiên cứu.
- 02 Từ Transformer trong NLP đến Vision Transformer (ViT).
- 03 Kiến trúc Vision Transformer (ViT).
So sánh ViT và CNN.
- 04 Thực nghiệm và Kết quả trên CIFAR-100.

- 05 Hỏi & Đáp



Giới thiệu - Bối cảnh

CNN là chuẩn vàng cho các tác vụ thị giác như phân loại ảnh và phát hiện đối tượng nhờ khả năng học đặc trưng cục bộ và bất biến tịnh tiến.

Ví dụ: LeNet, AlexNet, VGG, ResNet.

Transformer đã cách mạng hóa NLP với khả năng học các mối quan hệ dài thông qua Self-Attention.

ViT khám phá việc áp dụng Self-Attention lên hình ảnh

Từ Transformer trong NLP đến Vision Transformer (ViT)



Tổng quan về Transformer trong NLP:

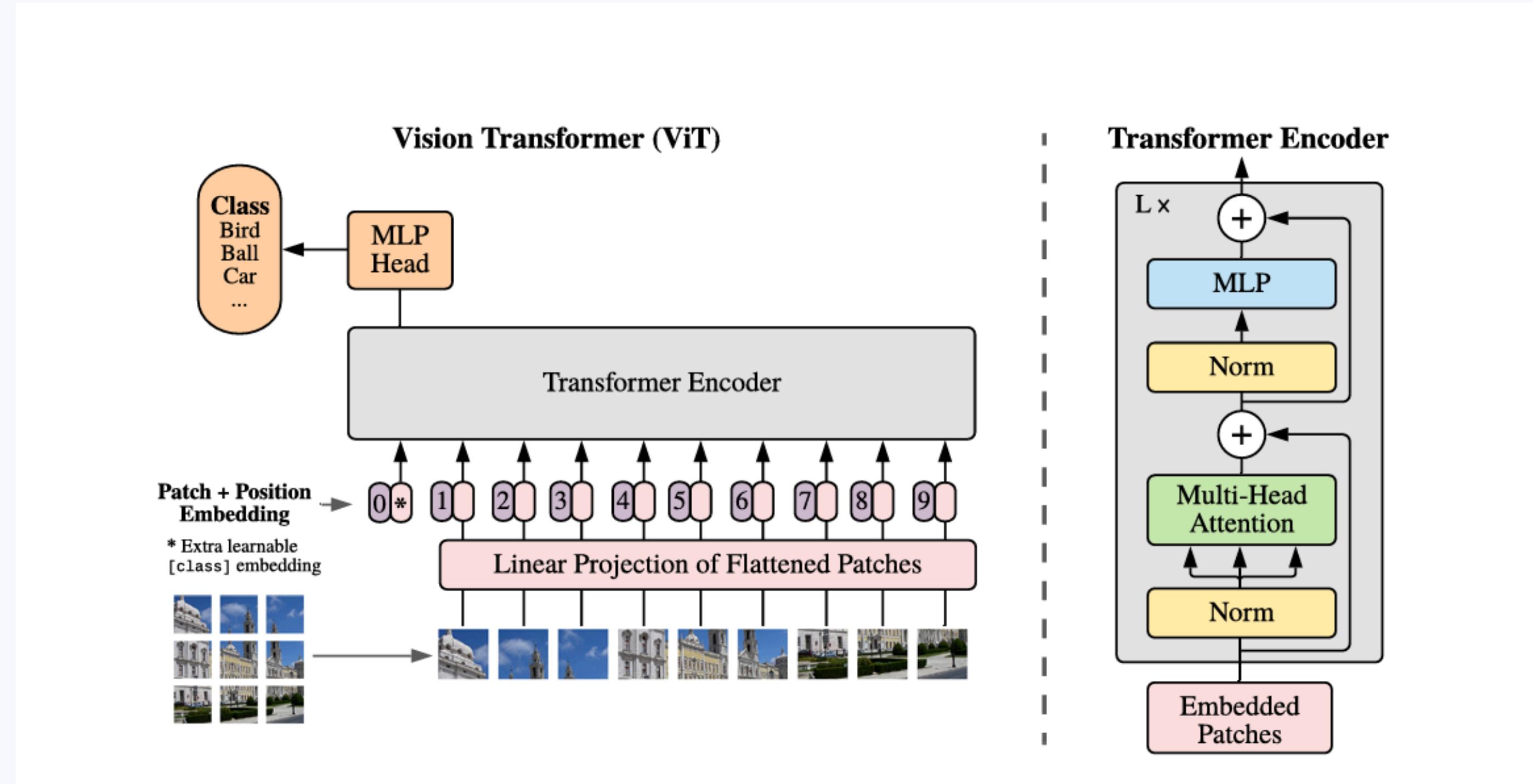
- Transformer xử lý một chuỗi các "token" (từ) đã được nhúng (embedded).
- Cơ chế cốt lõi là Self-Attention cho phép mỗi token "chú ý" đến các token khác trong chuỗi để hiểu ngữ cảnh.

Ý tưởng cốt lõi của ViT:

- Thay vì xử lý ảnh như một ma trận pixel 2D, ViT coi ảnh như một chuỗi các "patch" (miếng nhỏ) cố định.
- Mỗi patch được xem như một "token" (tương tự như một từ trong NLP).
- Các patch này sau đó được tuyến tính hóa và nhúng (linear embedding) thành các vector, rồi được đưa vào một kiến trúc Transformer Encoder tiêu chuẩn.

Mục tiêu:

Chứng minh rằng sự phụ thuộc vào CNNs là không cần thiết, và một Transformer thuần túy có thể hoạt động rất tốt trên các tác vụ phân loại ảnh khi được huấn luyện trên lượng dữ liệu lớn.



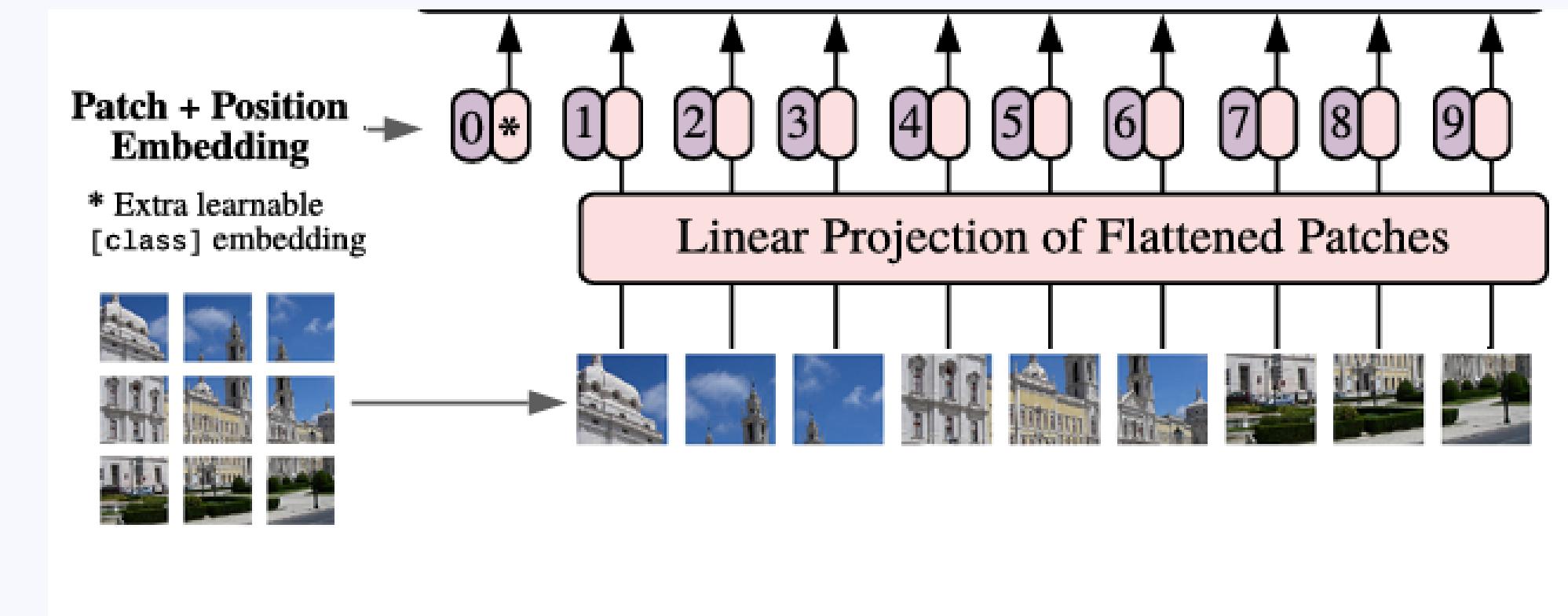
Kiến trúc Vision Transformer (ViT)

- Tổng quan

- 01 **Image Patching:** Chia ảnh thành các miếng nhỏ
- 02 **Linear Embedding:** Chuyển đổi các patch thành vector.
- 03 **Class Token:** Một token đặc biệt để tổng hợp thông tin phân loại.
- 04 **Positional Embedding:** Mã hóa thông tin vị trí của các patch.

- 05 **Transformer Encoder:** Chuỗi các khối Multi-Head Self-Attention và MLP.
- 06 **MLP Head:** Lớp phân loại cuối cùng.

Kiến trúc ViT - Chi tiết 1: Patch Embedding & Positional Embedding



$x \in \mathbb{R}^{H \times W \times C}$ Trong đó H là chiều cao, W là chiều rộng và C là số kênh màu.

- Chia ảnh thành các Patch: Ảnh được chia thành N patch (miếng nhỏ không chồng lấn), mỗi patch có kích thước (P, P) pixel.
- Số lượng patch được tính bằng công thức: $N = \frac{H \cdot W}{P^2}$.
- Mỗi patch (có kích thước $P \times P \times C$) sẽ được làm phẳng (flatten) thành một vector 1 chiều có $P^2 \cdot C$ phần tử, ký hiệu là $x_p^i \in \mathbb{R}^{P^2 \cdot C}$.

Kiến trúc ViT - Chi tiết 1: Patch Embedding & Positional Embedding

Tạo chuỗi đầu vào cho Transformer: Chuỗi đầu vào \mathbf{z}_0 cho Transformer Encoder được hình thành bằng cách kết hợp một Class Token \mathbf{x}_{class} với các patch embedding đã được chiếu tuyến tính $\mathbf{x}_p^i \mathbf{E}$, sau đó cộng thêm Positional Embedding \mathbf{E}_{pos} để giữ thông tin vị trí của các patch.

$$\mathbf{z}_0 = [\mathbf{x}_{class}; \mathbf{x}_p^1 \mathbf{E}; \mathbf{x}_p^2 \mathbf{E}; \dots; \mathbf{x}_p^N \mathbf{E}] + \mathbf{E}_{pos}$$

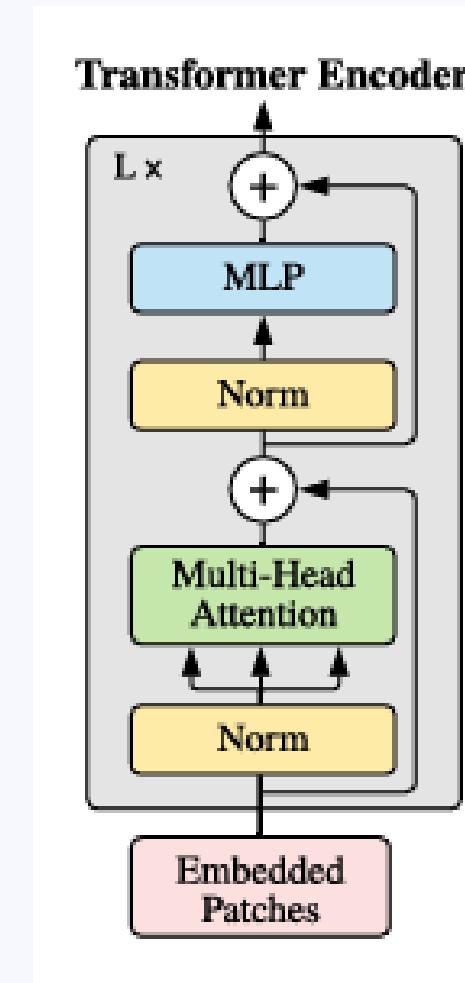
Trong đó:

- $\mathbf{E} \in \mathbb{R}^{(P^2 \cdot C) \times D}$ là ma trận chiếu tuyến tính (trainable linear projection).
- $\mathbf{x}_p^i \mathbf{E}$ là patch embedding thứ i.

Class Token (xclass): Embedding học được, tổng hợp thông tin phân loại.

Positional Embedding (Epos): Embedding học được, cộng vào patch embeddings, giữ thông tin vị trí.

Kiến trúc ViT - Chi tiết 2: Transformer Encoder



Transformer Encoder:

- Khối chính của Transformer gốc.
- Gồm Multi-Head Self-Attention (MSA) và MLP blocks.
- Áp dụng Layer Normalization (LN) trước, Residual Connections sau mỗi khối.

Kiến trúc ViT - Chi tiết 2: Transformer Encoder

$$z'_l = \text{MSA}(\text{LN}(z_{l-1})) + z_{l-1}$$

$$z_l = \text{MLP}(\text{LN}(z'_l)) + z'_l$$

Trong đó: l=1...L (chỉ số lớp); z_{l-1} (đầu vào); **LN** (LayerNorm); **MSA** (Multi-Head Self-Attention); **MLP** (Multi-Layer Perceptron).

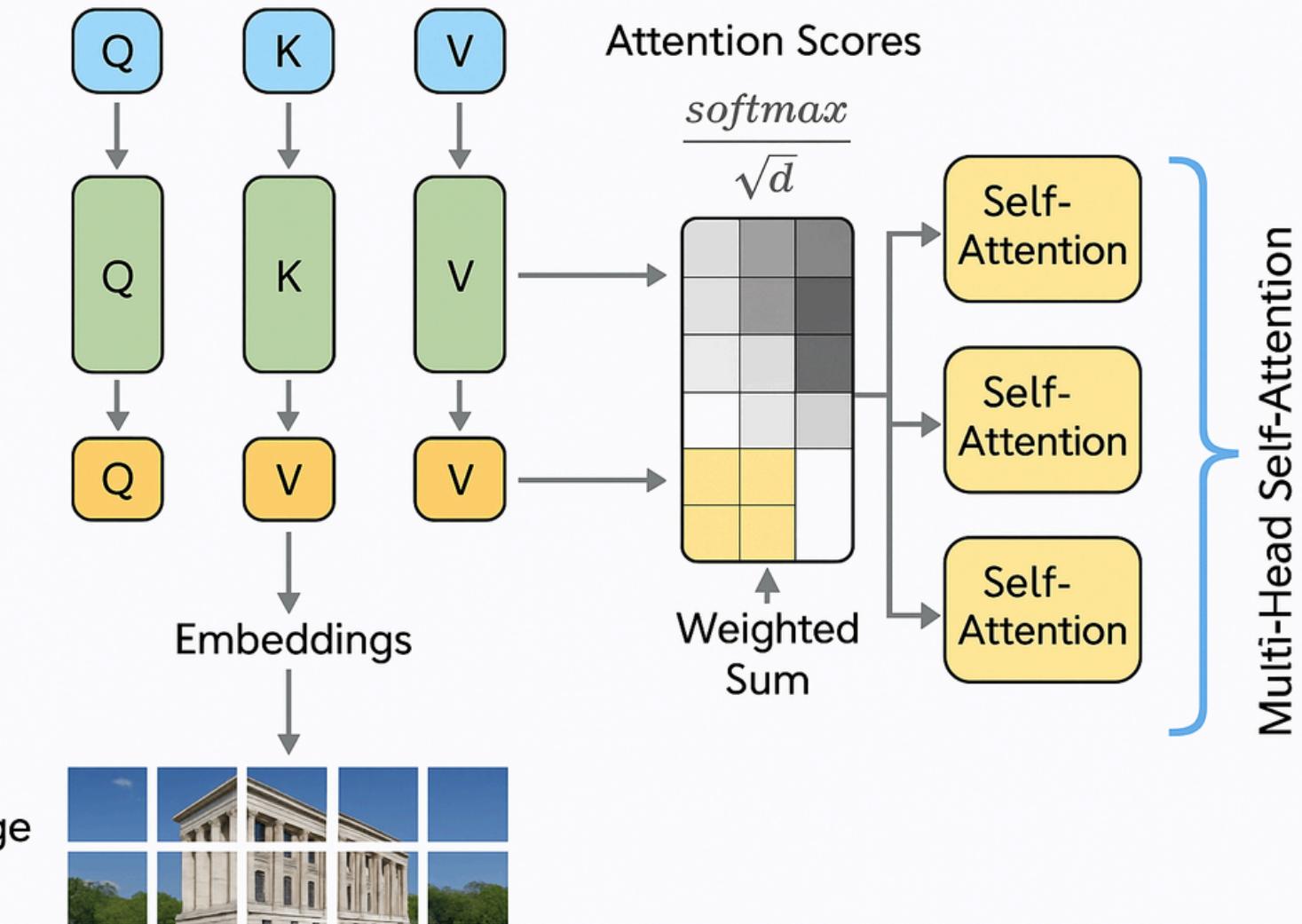
Cơ chế Multi-Head Self-Attention (MSA) Bản chất Self-Attention (SA):

Bản chất Self-Attention (SA):

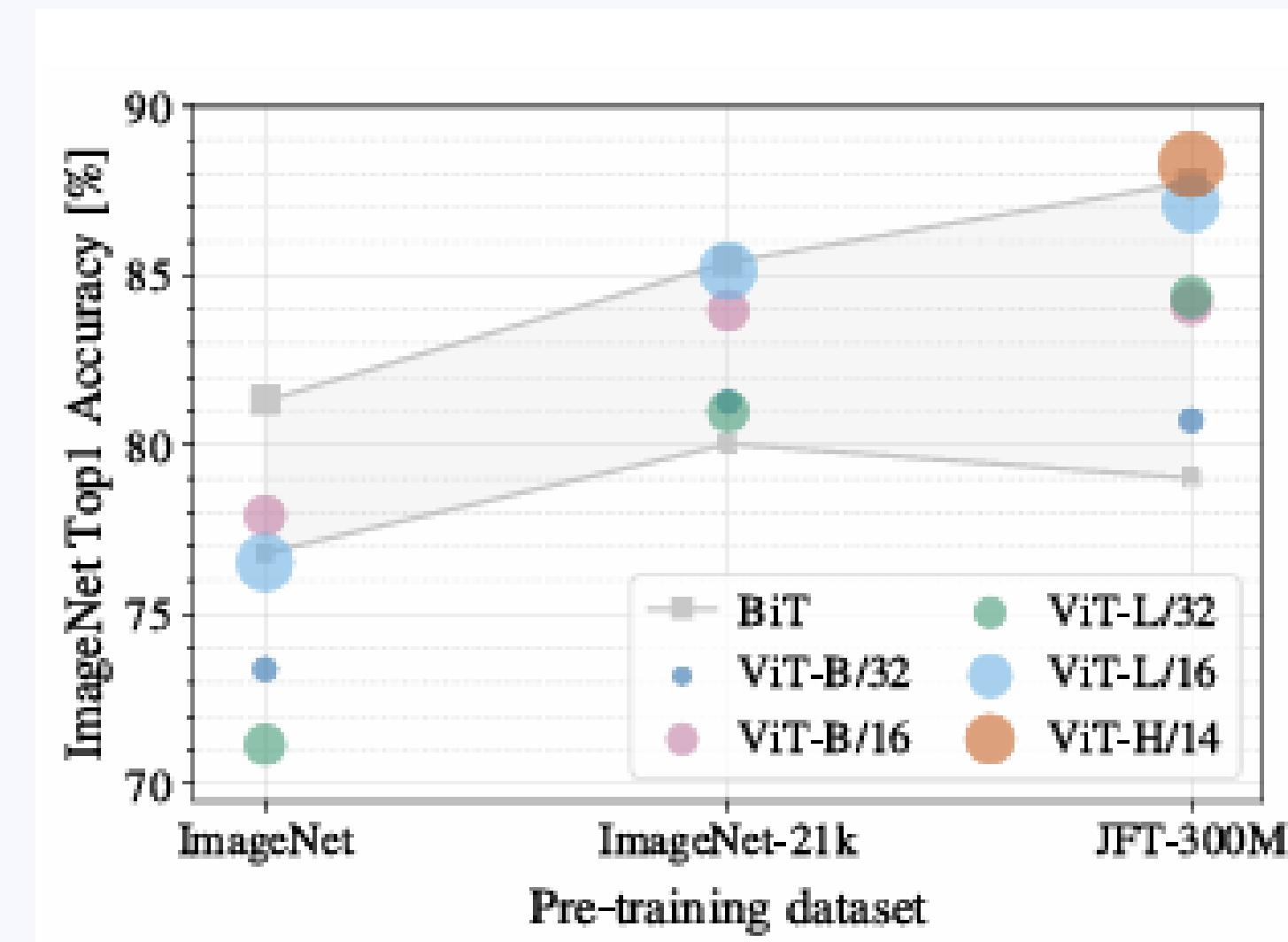
- Input $z \in \mathbb{R}^{N \times D}$
- Mỗi token chiếu thành Query (q), Key (k), Value (v).
- Công thức $[q, k, v] = zU_{qkv}$
- Tính Attention Scores : $A = \text{softmax}\left(\frac{QK^T}{\sqrt{D_h}}\right)$
- Tính đầu ra : $SA(z) = AV$

Multi-Head Self-Attention (MSA):

- k phép SA song song ("heads").
- Đầu ra các head nối lại, chiếu tuyến tính.
- Công thức: $MSA(z) = [SA_1(z); \dots; SA_k(z)]U_{msa}$
- Ý nghĩa: Chú ý các khía cạnh khác nhau của thông tin.



So sánh ViT và CNN - Yêu cầu dữ liệu & Khả năng mở rộng



Yêu cầu dữ liệu:

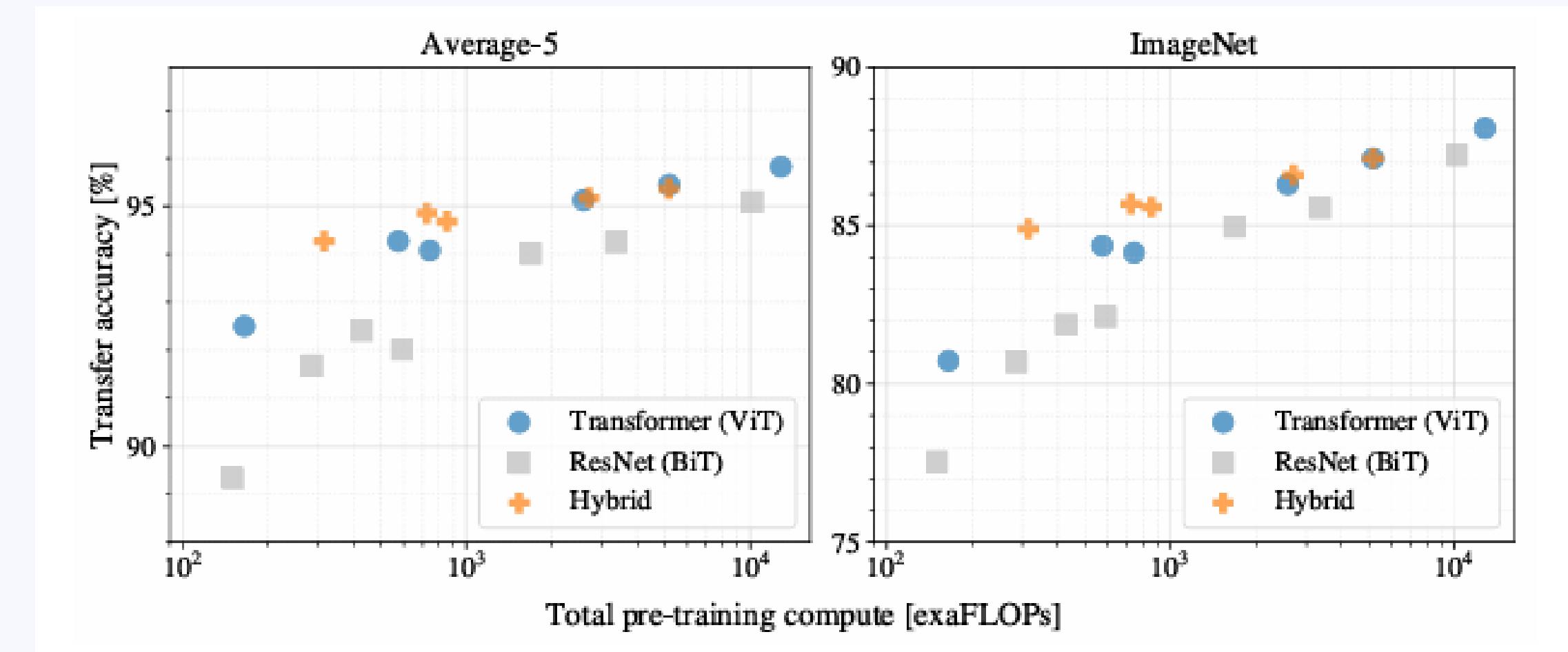
CNN: Tốt với dữ liệu vừa/nhỏ (ImageNet-1k).

ViT: Kém hơn CNN khi huấn luyện từ đầu trên dữ liệu nhỏ.

- “Khi được huấn luyện trên các bộ dữ liệu cỡ trung bình... mô hình chỉ đạt độ chính xác ở mức khiêm tốn...”
- “Tuy nhiên, bức tranh sẽ khác nếu các mô hình được huấn luyện trên những bộ dữ liệu lớn hơn... huấn luyện với quy mô lớn vượt trội hơn hẳn so với lợi thế từ inductive bias.”

Kết luận: ViT cần pre-trained trên datasets cực lớn (ImageNet-21k, JFT-300M).

So sánh ViT và CNN - Yêu cầu dữ liệu & Khả năng mở rộng



Khả năng mở rộng (Scalability):

CNN: Hiệu suất bão hòa.

ViT: Mở rộng vượt trội, tiếp tục cải thiện với dữ liệu lớn hơn.

- "Vision Transformers appear not to saturate within the range tried..." (Trang 8)
- Hiệu suất tương đương/tốt hơn CNN với chi phí pre-training thấp hơn.

So sánh ViT và CNN - Khác biệt cốt lõi

- "Sự khác biệt cốt lõi nằm ở Inductive Bias (thiên kiến quy nạp) – những giả định mà mô hình đã có về dữ liệu. CNN có các bias mạnh như Locality (chỉ xử lý vùng cục bộ) và Translation Equivariance (bất biến tịnh tiến – một vật thể dịch chuyển vẫn được nhận diện). Điều này giúp CNN học hiệu quả hơn với ít dữ liệu."
- "Ngược lại, ViT có rất ít thiên kiến về ảnh. Nó học các Global Relationships (mối quan hệ toàn cục) ngay từ lớp đầu tiên thông qua Self-Attention. Thông tin vị trí được học qua Positional Embedding chứ không phải tự nhiên."

Thực nghiệm và Kết quả trên CIFAR-100 - Thiết lập

Bộ dữ liệu: CIFAR-100 (100 lớp, 32×32 RGB).

- Tiền xử lý ViT: Resize 224×224, Data Augmentation, Chuẩn hóa.
- Tiền xử lý CNN: Giữ 32×32, Data Augmentation, Chuẩn hóa.

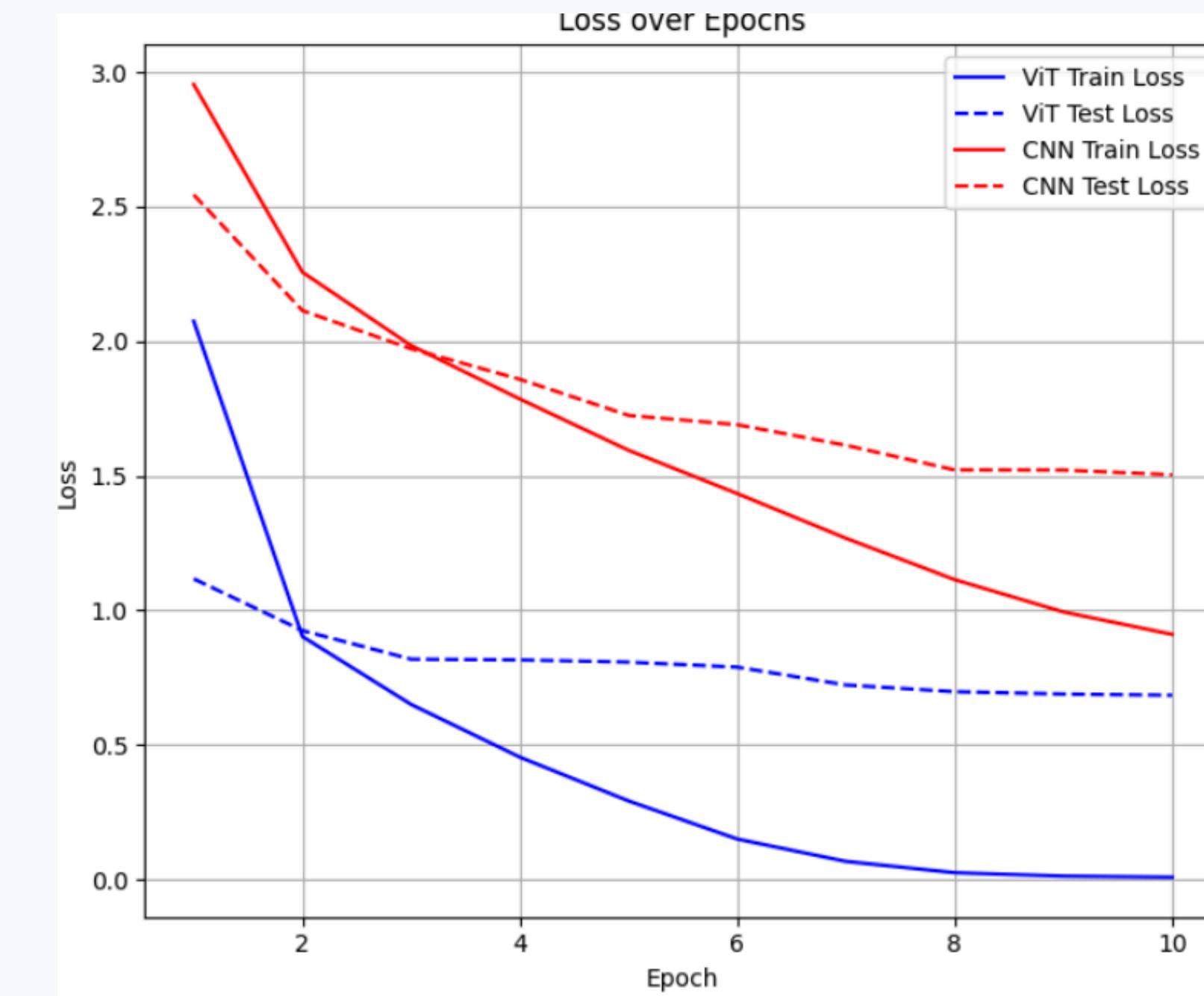
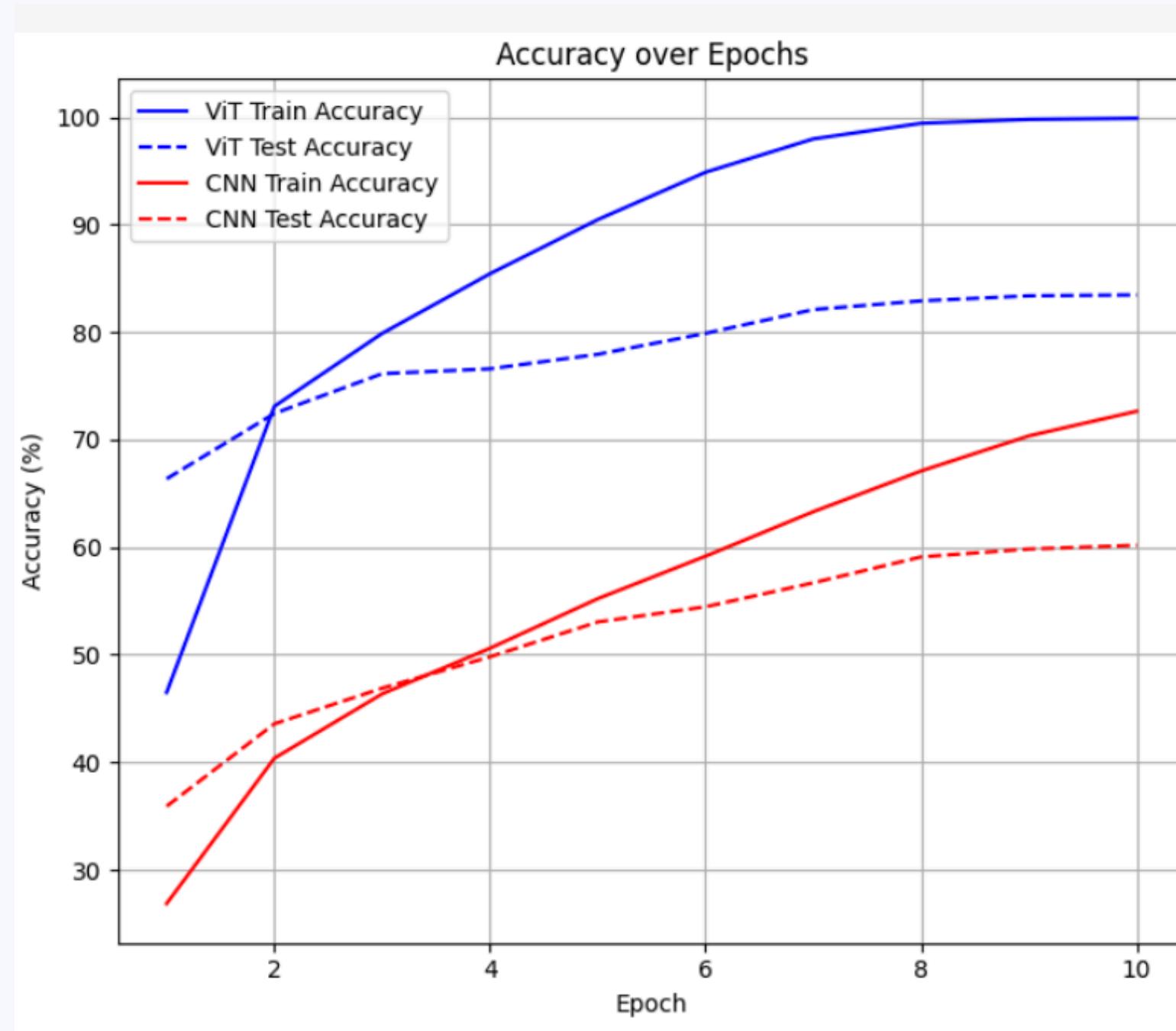
Mô hình:

- ViT: vit_tiny_patch16_224 (pretrained=True, fine-tune).
- CNN: ResNet18 (pretrained=True, fine-tune).

Huấn luyện:

- Loss: Cross-Entropy Loss.
- Optimizer: Adam.
- LR Scheduler: Cosine Annealing LR.
- Số Epoch: [Số epoch bạn đã chạy, ví dụ: 10]
- Thiết bị: GPU (Google Colab).

Thực nghiệm và Kết quả trên CIFAR-100 - Phân tích



Thực nghiệm và Kết quả trên CIFAR-100 - Phân tích

WARNING:matplotlib.image:Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.41051102..1.5126765].

--- Dự đoán của Vision Transformer ---

Thực tế (Ground Truth): mountain forest seal mushroom



ViT Dự đoán: road kangaroo seal mushroom

--- Dự đoán của ResNet18 (CNN) ---

Thực tế (Ground Truth): mountain forest seal mushroom

WARNING:matplotlib.image:Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.41051102..1.5126765].



CNN Dự đoán: train forest otter mushroom

Tài liệu tham khảo

Bài báo gốc Vision Transformer (ViT):

- Dosovitskiy, A., et al. (2020). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. ICLR 2021.
- [Link tới arXiv <https://arxiv.org/abs/2010.11929>]

Bài báo gốc Transformer:

- Vaswani, A., et al. (2017). Attention Is All You Need. NeurIPS 2017.
- [Link tới arXiv : <https://arxiv.org/abs/1706.03762>]

Thank You