ConvNeXt V2: Co-designing and Scaling ConvNets with Masked Autoencoders

Hà Hữu Phát 22521067
Nguyễn Hùng Phát 22521074
Nguyễn Đình Quân 22521184
Nguyễn Hồng Phát 22521072

Abstract

Trong những năm đầu thập kỷ 2020, lĩnh vực thị giác máy tính đã đạt được những bước tiến vượt bậc nhờ cải tiến kiến trúc mô hình và các phương pháp học biểu diễn, nổi bật là sự xuất hiện của các mô hình Transformer. Tuy nhiên, học có giám sát thường yêu cầu lượng lớn dữ liệu gán nhãn thủ công, trong khi học tự giám sát khai thác tốt dữ liệu chưa gán nhãn nhưng lại gặp khó khăn trong việc tối ưu hóa hiệu quả mô hình. Sự kết hợp giữa hai phương pháp này, nếu được thực hiện hiệu quả, hứa hẹn sẽ tận dụng được điểm mạnh của cả hai, nhưng hiện vẫn chưa đạt hiệu suất tối ưu. Báo cáo này tập trung nghiên cứu và đề xuất các cải tiến nhằm tối ưu hóa mô hình ConvNeXt, qua đó nâng cao hiệu suất trên các bài toán quan trọng trong thị giác máy tính như phân loại, phát hiện đối tượng và phân đoạn ảnh.

1 Introduction

Dựa trên những tiến bộ trong lĩnh vực nhận dạng hình ảnh, báo cáo này tập trung vào việc tối ưu hóa mô hình ConvNeXt nhằm cải thiện hiệu suất học biểu diễn tự giám sát. Với thiết kế được cải tiến, ConvNeXt V2 (Liu et al., 2023) được tối ưu hóa để tích hợp hiệu quả với các bộ mã hóa tự động mặt nạ (Masked Autoencoders), mang lại những cải tiến đáng kể trên các tác vụ quan trọng như phân loại ImageNet, phát hiện đối tượng COCO, và phân đoạn ADE20K. Các kết quả thực nghiệm cho thấy ConvNeXt V2 không chỉ cải thiện hiệu suất so với các mô hình ConvNet truyền thống mà còn đảm bảo hiệu quả trên nhiều mức độ phức tạp tính toán, khẳng định tính linh hoạt và ứng dụng rộng rãi trong thị giác máy tính hiện đại.

2 Related Work

2.1 Vision Transformer (ViT) và Masked Autoencoder (MAE)

Kể từ khi Transformer (Vaswani et al., 2017) đạt được thành công lớn trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP), các nhà nghiên cứu đã nhanh chóng nhận ra tiềm năng của kiến trúc này trong các bài toán thị giác máy tính (CV). Vision Transformer (Dosovitskiy et al., 2021) được giới thiệu như một ứng dụng đầu tiên của Transformer trong lĩnh vực CV, dựa trên việc chia hình ảnh thành các patch và sử dụng kỹ thuật tự chú ý để học các đặc trưng toàn cục. Điều này đã mở ra một hướng đi mới, cho phép các mô hình tận dụng sức mạnh của kiến trúc Transformer để đạt được hiệu suất vượt trội.

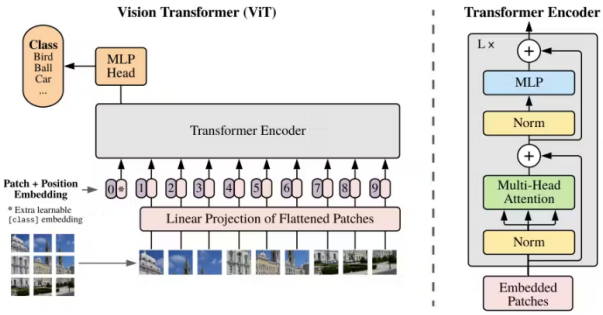

Figure 1: Kiến trúc Vision Transformer (ViT) với cơ chế mã hóa Transformer Encoder và các bước tiền xử lý.

Tuy nhiên, một trong những hạn chế lớn của ViT là yêu cầu lượng dữ liệu lớn để mô hình hoạt động tốt. Điều này khiến ViT (hình 1) gặp khó khăn trong việc áp dụng trên các tập dữ liệu nhỏ hoặc vừa. Để khắc phục hạn chế này, các nhà nghiên cứu đã áp dụng kỹ thuật *Masked Autoencoder* (MAE) - một phương pháp tiền huấn luyện được lấy cảm hứng từ lĩnh vực NLP. MAE giúp ViT học các đặc trưng tốt hơn thông qua cơ chế mã hóa và giải mã các vùng bị ẩn của hình ảnh.

Trong MAE (hình 2), một phần hình ảnh đầu vào được ẩn (masked) trước khi đưa qua mã hóa (*encoder*). Bộ mã hóa chỉ xử lý các vùng không bị ẩn, giúp giảm thiểu tính dư thừa trong dữ liệu. Sau đó, một bộ giải mã (*decoder*) được sử dụng để tái tạo lại toàn bộ hình ảnh, bao gồm cả các vùng bị ẩn. Cơ chế này không chỉ cải thiện khả năng học đặc trưng của ViT mà còn giúp mô hình hoạt động

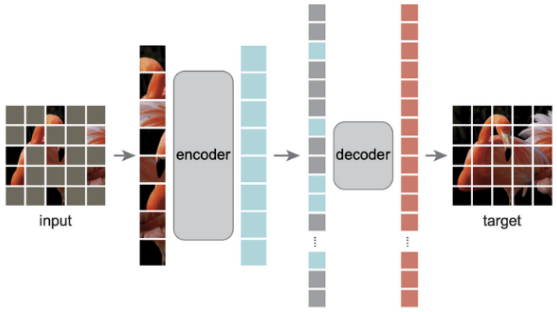

Figure 2: Cơ chế Masked Autoencoder (MAE) trong Vision Transformer.

hiệu quả hơn trên các tập dữ liệu nhỏ, nhờ việc tận dụng các đặc trưng tổng quát từ quá trình tiền huấn luyện.

Sự ra đời của MAE đã giúp ViT vượt qua những hạn chế trước đây và trở thành một trong những kiến trúc tiên phong trong lĩnh vực thị giác máy tính. Việc kết hợp ViT với các kỹ thuật như MAE tiếp tục mở ra những hướng nghiên cứu mới, đồng thời thúc đẩy sự phát triển của các mô hình dựa trên Transformer trong thị giác máy tính.

2.2 ConvNeXt V1

Sau khi ViT ra đời và đạt được kết quả tốt hơn các mạng convolution nerual network phổ biến như Resnet (He et al., 2016) ở bài toán phân loại ảnh, mạng ConvNeXt (Liu et al., 2022) ra đời để chứng minh rằng mạng tích chập vẫn không hề "lỗi thời". ConvNeXt cho thấy rằng với những cải tiến phù hợp, các mô hình tích chập vẫn có thể cạnh tranh, thậm chí vượt qua Transformer trong một số tác vụ. Điều này làm sáng tỏ tiềm năng lâu dài của các mô hình tích chập trong CV.

ConvNeXt được xây dựng dựa trên nền tảng của kiến trúc Rensnet, nhưng được hiện đại hóa và áp dụng các kĩ thuật huấn luyện của Transformer. Kiến trúc mạng ConvNeXt được giới thiệu ở Hình 3.

ConvNeXt sử dụng cấu trúc block tương tự như ResNet, nhưng thực hiện một vài sự điều chỉnh để phù hợp với các kiến trúc mạng Transformer. Ở lớp đầu tiên Stem cell, thay vì sử dụng lớp tích chập 7x7 với stride = 2 theo sau là một lớp max pooling để giảm chiều dữ liệu đi 4 lần như trong Resnet, ConvNeXt áp dụng kĩ thuật Patchify từ ViT, sử dụng các lớp tích chập không trùng lặp 4x4 với stride = 4 như hình 4.

Ở các stage tiếp theo, nếu như Resnet thực hiện giảm số chiều đi một nửa trực tiếp ở các lớp tích chập đầu tiên của Residual block bằng cách sử dụng stride = 2 (hình 5) thì ConvNeXt thực hiện tách rời hai bước tương tự như Swin Transformer (Liu et al., 2021).

Trước mỗi block, ConNeXt sử dụng một lớp tích chập 2x2, stride = 2 như hình 3 để giảm chiều dữ liệu đi một nửa. Trong mỗi block, áp dụng kĩ thuật inverted bottleneck trong các Transformer block và depthwise convolution được sử dụng trong MobileNet (Howard et al., 2017), với sự điều chỉnh thứ tự thực hiện giữa các lớp để giảm thiểu chi phí tính toán (hình 6). Thay vì thực hiện lớp tích chập trên 384 kênh, với việc đưa lớp depthwise convolution lên trước pointwise convolution, ConvNeXt chỉ thực hiện tính toán trên 96 kênh độc lập. Mỗi kênh được xử lý một cách độc lập, giúp mô hình học được các đặc trưng không gian (spatial information), sau đó sử dụng pointwise convolution để kết hợp thông tin giữa các kênh (channel information). Điều này cho phép ConvNeXt thực hiện tính toán trên chiều không gian (spatial dimension) và chiều kênh (channel dimension) độc lập với nhau, tương tự như cách thức mà ViT hoạt động: sử dụng self-attention để tính toán mối quan hệ giữa các patch (spatial information) trong các kênh một cách độc lập rồi sử dụng feed-forward network để kết hợp đặc trưng giữa các kênh.

Một ưu điểm nổi bật của kiến trúc Transformer đó là việc sử dụng self attention sẽ giúp học được các đặc trưng toàn cục thay vì bị giới hạn ở các lớp tích chập. Vận dụng ý tưởng trên, ConvNext sử dụng các lớp tích chập lớn 7x7 thay vì sử dụng các kernel nhỏ 3x3 để học được các đặc trưng ở vùng rộng hơn. Số block của mỗi stage cũng được thay đổi lần lượt thành 3,3,9,3 (tương đương với Swin Transformer là 1,1,3,1) thay vì là 3,4,6,3 như Resnet.

Ngoài ra, ConvNeXt thay thế hàm kích hoạt ReLU sử dụng trong các mạng Convnet trước đây bằng GELU được sử dụng phổ biến trong các kiến trúc Transformer hiện đại. ConvNeXt chỉ sử dụng duy nhất một hàm kích hoạt trong mỗi block thay vì là sau mỗi layer như các mạng Convnet truyền thống. Bên cạnh đó, ConvNeXt cũng thay các Batch normalization thành Layer normalization vốn được sử dụng trong các kiến trúc Transformer và cũng chỉ sử dụng duy nhất một Layer normalization. Hình 7 so sánh một block của ResNeXt, một biến thể của mạng Resnet và ConvNeXt.

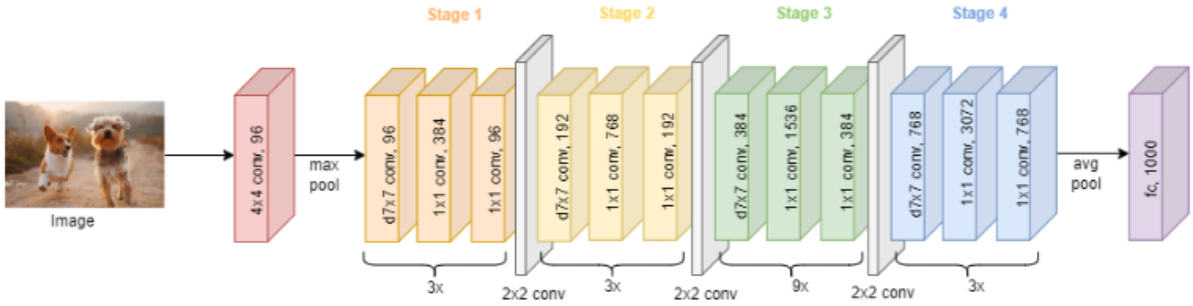

Figure 3: Kiến trúc mạng ConvNeXt

Figure 4: ConvNeXt stem cell

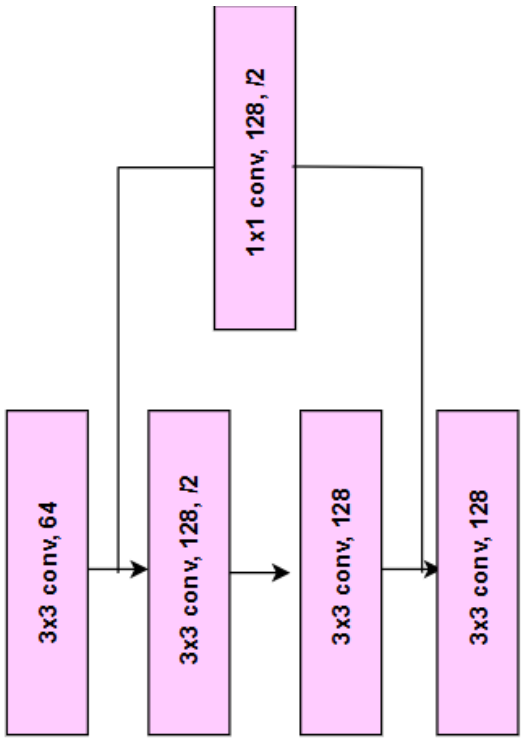

Figure 5: Resnet residual block

3 ConvNeXt V2

3.1 Fully Convolutional Masked Autoencoder (FCMAE)

Trong ConvNeXt V2, một trong những cải tiến quan trọng được giới thiệu là phương pháp *Fully Convolutional Masked Autoencoder* (FCMAE), một kỹ thuật học tự giám sát (*self-supervised learning*) được thiết kế dành riêng cho mạng nơ-ron tích chập (ConvNets).

Việc sử dụng *Masked Autoencoder* (MAE) dẫn tới chi phí tính toán tăng do phải sử dụng các masked tokens, đồng thời gây ra hiện tượng không nhất quán giữa tập huấn luyện và tập kiểm tra do tập kiểm tra không bị masked (*che*). FCMAE tận dụng hoàn toàn các lớp tích chập (*fully convolutional layers*) để xử lý và tái tạo lại dữ liệu. Kiến trúc của FCMAE bao gồm hai thành phần chính:

- **Hierarchical Encoder:** Bộ mã hóa này được thiết kế dựa trên kiến trúc mạng ConvNext V1, sử dụng các lớp tích chập rời rạc (*sparse convolution*) để trích xuất đặc trưng từ các vùng không bị che (masked) của ảnh đầu vào.
- **Plain Decoder:** Bộ giải mã đóng vai trò tái tạo lại toàn bộ ảnh từ các đặc trưng được mã hóa. Dựa trên kiến trúc mạng ConvNeXT V1, bộ giải mã có thể tái hiện các vùng bị che với độ chính xác cao.

FCMAE áp dụng quá trình masking bằng cách che đi một phần lớn các patch (vùng) trong ảnh đầu vào, với tỷ lệ lên đến 60%. Sau đó sử dụng các lớp tích chập rời rạc để chỉ xử lý trên những vùng ảnh không bị che đi. Việc chỉ chú ý vào những vùng ảnh nhìn thấy giúp giảm chi phí tính toán một cách đánh kể, đồng thời tránh việc nhìn thấy và "copy" thông tin từ các vùng bị che khuất nếu sử dụng lớp tích chập bình thường.

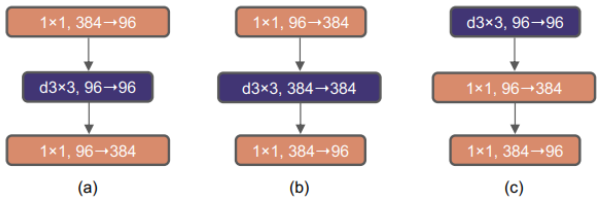

Figure 6: So sánh giữa các block. (a) sử dụng bottleneck block được giới thiệu trong mạng ResNeXt; (b) inverted bottleneck block lấy ý tưởng từ Transformer block; (c) ConvNeXt thay đổi thứ tự giữa các layer để tối ưu chi phí tính toán

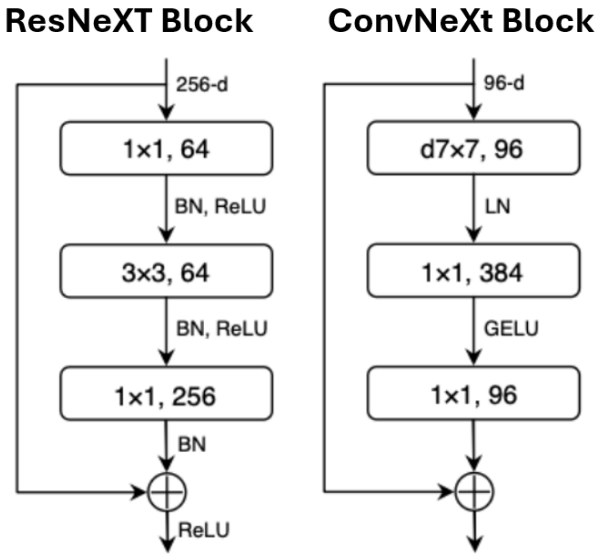

Figure 7: ResNeXt block và ConvNeXt block

So với MAE, FCMAE có ưu điểm nổi bật là giảm bớt sự phụ thuộc vào việc phải học thêm các masked tokens, từ đó tiết kiệm tài nguyên tính toán và cũng có thể dễ dàng chuyển sang lớp tích chập bình thường trên tập kiểm tra, tránh được sự không nhất quán nếu như sử dụng thêm các masked tokens. Bên cạnh đó, nhờ vào các lớp tích chập hoàn toàn, FCMAE tận dụng tốt hơn các đặc trưng không gian trong ảnh, giúp cải thiện hiệu suất tái tạo và học tập.

Kỹ thuật FCMAE không chỉ giúp ConvNeXt V2 đạt được hiệu suất vượt trội trên các bài toán phân loại và phát hiện đối tượng mà còn mở ra hướng phát triển mới cho các phương pháp học tự giám sát trong mạng tích chập. Việc thay thế Transformer bằng lớp tích chập hoàn toàn trong FCMAE đã chứng minh khả năng cải tiến đáng kể mà vẫn đảm bảo hiệu quả tính toán cao.

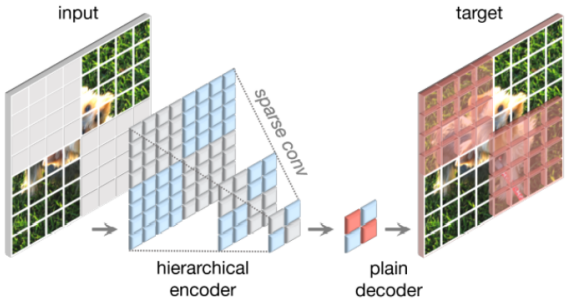

Figure 8: Kiến trúc Fully Convolutional Masked Autoencoder (FCMAE) trong ConvNeXt V2.

3.2 GRN

Trong quá trình thực nghiệm với ConvNeXt V1, một vấn đề quan trọng được phát hiện là hiện tượng "feature collapse".

Hiện tượng feature collapse: khi phân tích các kênh đặc trưng (Hình 9), chúng ta thấy nhiều kênh trong ConvNeXt V1 gần như trùng lặp hoặc bị bão hòa. Điều này dẫn đến sự thiếu đa dạng của các đặc trưng, khiến mô hình không tận dụng được tiềm năng của chúng.

Để hiểu rõ hơn hiện tượng này, chúng ta sử dụng phương pháp phân tích khoảng cách cosine giữa các đặc trưng. Phương pháp này tính toán độ tương đồng cosine giữa các vector đặc trưng để đo lường mối quan hệ giữa các kênh. Cụ thể, giá trị trung bình pair-wise cosine distance được tính như sau:

$$\frac{1}{C^2}\sum_{i}^{C}\sum_{j}^{C}\frac{1-\cos(X_i, X_j)}{2} \tag{1}$$

X_i, X_j là feature map của kênh i và j.

Khi giá trị trung bình lớn, khoảng cách giữa các đặc trưng lớn, thể hiện tính đa dạng cao. Ngược lại, giá trị nhỏ phản ánh sự dư thừa đặc trưng. Từ kết quả tại hình 10, có thể thấy khoảng cách cosine giữa các đặc trưng trong ConvNeXt V1 là khá nhỏ, dẫn đến hiện tượng feature collapse.

Để giải quyết vấn đề này, một kỹ thuật mới gọi là Global Response Normalization (GRN) được đề xuất. GRN cải thiện hiệu suất của ConvNet bằng cách:

- Tăng độ tương phản: Làm nổi bật sự khác biệt giữa các kênh.
- Tăng tính chọn lọc: Giúp mỗi kênh tập trung vào đặc trưng quan trọng nhất.

GRN gồm 3 bước chính:

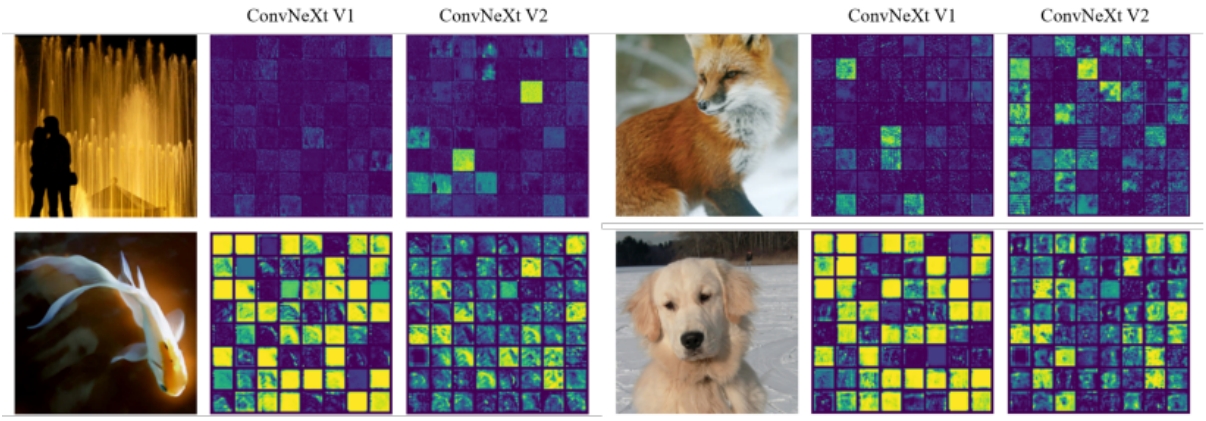

Figure 9: Feature activation visualization

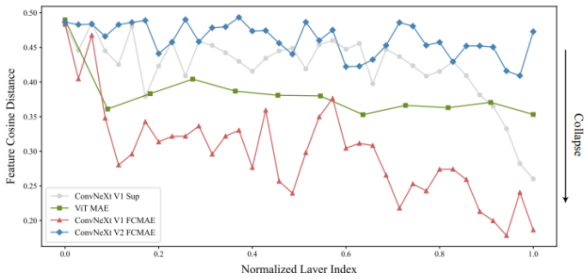

Figure 10: Feature cosine distance analysis

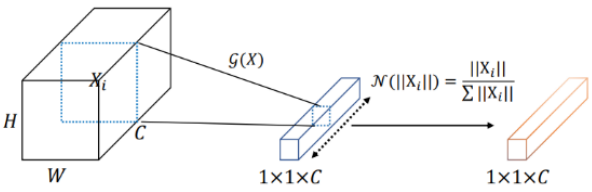

Figure 12: Feature Normalization

1. Tổng hợp đặc trưng toàn cục (Global Feature Aggregation)

Sử dụng chuẩn L2 để tổng hợp thông tin từ tất cả các vị trí không gian (chiều cao và chiều rộng) trong mỗi kênh (hình 11).

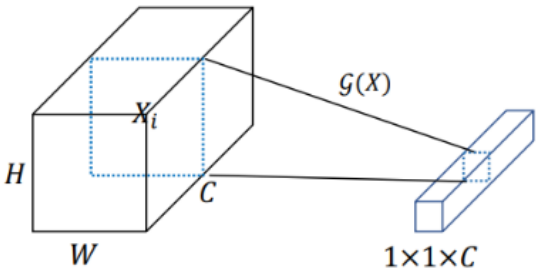

Figure 11: Global Feature Aggregation

2. Chuẩn hóa đặc trưng (Feature Normalization)

Sau khi tính tổng hợp toàn cục, GRN chuẩn hóa giá trị này bằng cách chia cho giá trị trung bình của chuẩn L2 trên các kênh. Điều này giúp điều chỉnh tầm quan trọng của từng kênh đặc trưng một cách tương đối so với các kênh khác (hình 12).

3. Điều chỉnh đặc trưng (Feature Calibration)

Cuối cùng, GRN nhân các đặc trưng đầu vào với giá trị chuẩn hóa để tăng độ tương phản giữa các kênh (hình 13). Điều này làm nổi bật đặc trưng quan trọng và giảm ảnh hưởng của các đặc trưng kém quan trọng. Đồng thời, GRN sử dụng thêm hai tham số học được (γ, β) để điều chỉnh các đặc trưng:

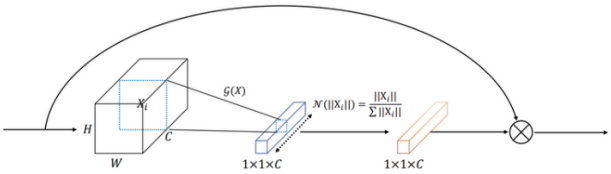

Figure 13: Feature Calibration

$$X_i = \gamma \cdot X_i \cdot N(G(X)_i) + \beta + X_i \qquad (1)$$

Sau khi sử dụng GRN ở mô hình ConvNeXt V2 chúng ta có thể quan sát thấy các đặc trưng đa dạng hơn (hình 9) và khoảng cách cosine giữa các đặc trưng được tăng lên (hình 10).

Hình 14 so sánh block giữa 2 kiến trúc ConvNeXt v1 và ConvNeXt v2, có thể thấy sự khác nhau chỉ nằm ở việc sử dụng thêm GRN.

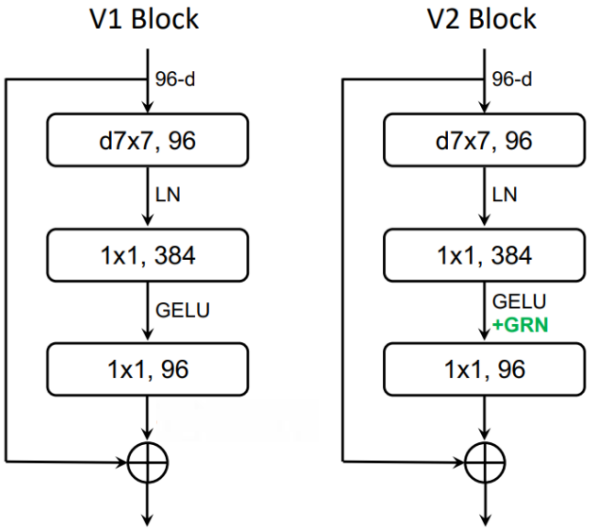

Figure 14: So sánh block giữa ConvNeXt v1 và ConvNeXt v2

4 Experiments and results

4.1 Dataset

4.1.1 Tổng quan dữ liệu

Trong bài toán này, nhóm chúng em sử dụng bộ dữ liệu trong môn Máy Học (CS114). Các hình ảnh trong dataset được đóng góp bởi toàn bộ sinh viên của lớp nhằm xây dựng tập dữ liệu thực tế và đa dạng, phục vụ cho các bài toán học máy liên quan đến nhận diện và phân loại.

Mục tiêu của nhóm chúng em là phát triển mô hình phân loại các hãng xe máy dựa trên hình ảnh, với 5 nhãn phân loại: Honda, Suzuki, Yamaha, Vinfast, Other.

4.1.2 Thông số dữ liệu

Dưới đây là số lượng mẫu dữ liệu trong từng nhãn của tập train và test.

Nhãn	Train	Test	Tổng cộng
Honda	2495	85	2580
Suzuki	3946	264	4210
Yamaha	4247	250	4497
Vinfast	2645	182	2827
Other	2012	120	2132
Tổng cộng	15345	901	16246

Ngoài ra, hai biểu đồ hình 15. và hình 16 minh họa phân phối số lượng mẫu dữ liệu trong các nhãn của tập train và test.

4.2 Results

Hình 17 thể hiện kết quả sau khi áp dụng FCMAE ở bước Encoder.

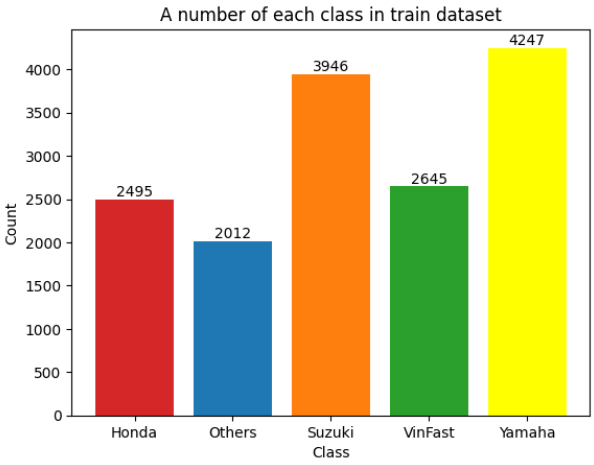

Figure 15: Dataset Train

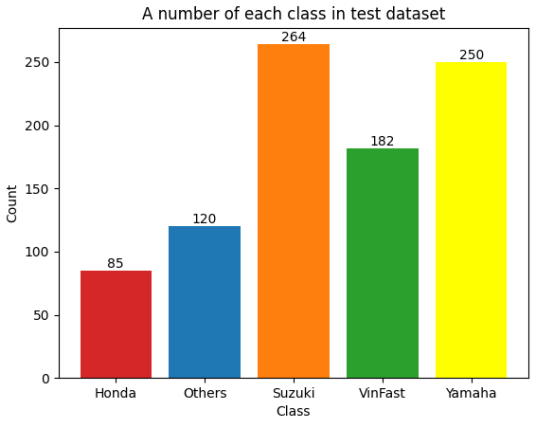

Figure 16: Dataset Test

Kết quả so sánh các mô hình với các phương pháp khác nhau được thể hiện ở bảng 1:

Backbone	Method	#Param (M)	Training Epochs	Accuracy (%)
ViT-T	MAE	5.52	50	48.44
ConvNeXt V2	FCMAE	31.9	50	**52.8**
ConvNeXt V1	FCMAE	29.7	50	49.4
ConvNeXt V2	Supervised	27.8	50	47.1
ConvNeXt V2	FCMAE without GRN	31.9	50	45.3

Table 1: Comparison of different backbones, methods, parameter counts, and their fine-tuning accuracies.

Nhận xét: Kết quả cho thấy ConvNeXt V2 với phương pháp FCMAE đạt độ chính xác cao nhất (52.8%), vượt trội so với các mô hình khác, chứng tỏ hiệu quả vượt trội của kiến trúc và phương pháp tiền huấn luyện (pre-training). So với ViT-T, ConvNeXt V1 sử dụng FCMAE cũng cho kết quả cao hơn (49.4% so với 48.44%), chỉ ra rằng ConvNeXt có tiềm năng mạnh mẽ ngay cả ở phiên bản cũ. ConvNeXt V2 với phương pháp huấn luyện có giám sát (Supervised) cho kết quả thấp hơn rõ rệt (47.1%) so với FCMAE, khẳng định lợi thế của

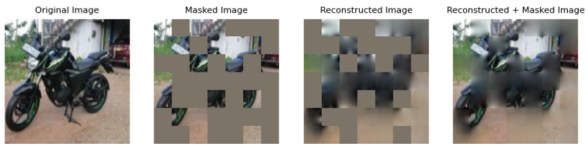

Figure 17: Kết quả áp dụng FCMAE gồm: hình ảnh gốc, hình ảnh sau khi bị che khuất, hình ảnh tái tạo (che đi những phần không cần tái tạo), hình ảnh cuối cùng (gồm hình ảnh được tái tạo + hình ảnh không cần tái tạo)

phương pháp tự giám sát trong học biểu diễn. Cuối cùng, việc loại bỏ GRN (Global Response Normalization) trong ConvNeXt V2 làm giảm hiệu suất đáng kể (45.3%), cho thấy vai trò quan trọng của GRN trong việc cải thiện chất lượng học và cạnh tranh tính năng giữa các kênh.

5 Conclusions and anwer questions

Kết luận: Trong đồ án này, nhóm chúng em đã áp dụng mô hình ConvNeXt V2 với phương pháp tiền huấn luyện FCMAE để giải quyết bài toán phân loại xe máy. Mô hình ConvNeXt V2, với thiết kế tối ưu cho tự giám sát, đã chứng minh hiệu quả vượt trội so với các mô hình khác. Các kết quả thực nghiệm cho thấy mô hình này mang lại cải thiện đáng kể trong hiệu suất, đặc biệt khi so với các phương pháp huấn luyện có giám sát, khẳng định tiềm năng mạnh mẽ của ConvNeXt trong học biểu diễn và phân loại hình ảnh. **Từ đó, có thể thấy được rằng kiến trúc mạng Convnet nếu được áp dụng các kĩ thuật huấn luyện của Transformer vẫn là "State of the art" trong các bài toán về thị giác máy tính.**

Giải đáp các câu hỏi của thầy và các nhóm về đồ án:

Câu 1: Các vấn đề mà ResNet mắc phải để có cải tiến của ConvNext?

Các vấn đề chính của ResNet mà ConvNeXt cải tiến:

1. **Khả năng học biểu diễn**: ResNet thiếu khả năng học các đặc điểm toàn cục; ConvNeXt áp dụng thiết kế giống Transformer để cải thiện điều này.
2. **Kỹ thuật huấn luyện cũ**: ResNet dùng các kỹ thuật huấn luyện cũ, ConvNeXt áp dụng kỹ thuật hiện đại tương tự Transformer như AdamW và tăng cường dữ liệu.
3. **Giảm độ phân giải**: ResNet giảm độ phân giải trong các residual block; ConvNeXt tách riêng quá trình này để giảm thiểu mất mát thông tin.
4. **Batch Normalization**: ConvNeXt thay BN bằng Layer Normalization để cải thiện sự ổn định.
5. **Hàm kích hoạt**: ConvNeXt thay ReLU bằng GELU để cải thiện sự học phi tuyến tính.
6. **Cấu trúc kiến trúc**: ConvNeXt sử dụng kernel lớn hơn (5x5, 7x7) thay vì 3x3 như ResNet.
7. **Chiều rộng và chiều sâu mạng**: ConvNeXt tối ưu chiều rộng và chiều sâu mạng hiệu quả hơn.

Câu 2: Tại sao dùng Mask Encoder thì tăng chi phí tính toán?

Các lý do chính tại sao việc sử dụng Mask Encoder làm tăng chi phí tính toán:

1. **Phức tạp trong huấn luyện**: Việc huấn luyện Mask Encoder đòi hỏi phải học thêm các masked tokens dẫn đến chi phí tăng đáng kể.
2. **Phục hồi thông tin**: Việc phục hồi các phần của dữ liệu đã bị mask đòi hỏi phải tính toán nhiều bước hơn để tái tạo lại thông tin bị mất.
3. **Tăng số lượng tham số**: Do các bước phụ trợ liên quan đến masking và phục hồi, Mask Encoder yêu cầu một mô hình phức tạp hơn với nhiều tham số hơn, làm tăng chi phí tính toán.

Do đó, Mask Encoder có thể cải thiện hiệu suất, nhưng nó làm tăng đáng kể chi phí tính toán.

Câu 3: Đối với MAE thì mask thường che bao nhiêu % ảnh là tốt?

Trong **Masked Autoencoder (MAE)**, masking khoảng **75%** của ảnh thường cho kết quả tốt nhất. Tỷ lệ này giúp mô hình học các đặc điểm mạnh mẽ từ phần không bị che, đồng thời khôi phục chính xác các phần còn lại. Masking quá ít có thể làm mô hình không học được các biểu diễn tốt từ dữ liệu bị che giấu, trong khi masking quá nhiều có thể làm mô hình trở nên khó học. Tóm lại, **75%** là tỷ lệ phổ biến và hiệu quả trong MAE, nhưng cũng cần điều chỉnh tùy theo bài toán cụ thể.

Câu 4: Lớp GRN có ảnh hưởng như thế nào đến hiệu suất mô hình ConvNeXt v2?

Lớp Global Response Normalization (GRN) cải thiện hiệu suất ConvNeXt v2 bằng cách tăng cường tính chọn lọc và mức độ tương phản giữa các kênh, giảm trùng lặp và xử lý các kênh có output không đồng đều. GRN sử dụng L2-norm và response normalization để tạo sự cạnh tranh giữa các kênh, giúp nổi bật các đặc trưng quan trọng và tối ưu hóa mô hình, cải thiện hiệu suất phân loại. Nếu không sử dụng GRN, có thể thấy hiệu suất mô hình giảm đi đáng kể (đã chứng minh ở bảng 1).

Câu 5: Cải tiến của ConvNeXt v2 so với ConvNeXt v1 là gì?

Cải tiến của ConvNeXt v2 so với ConvNeXt v1 bao gồm:

1. **Fully Convolutional Masked Autoencoder**: ConvNeXt v2 sử dụng phương pháp này để cải thiện khả năng học các đặc trưng không gian và nâng cao hiệu suất trong các tác vụ tự học không giám sát.
2. **Global Response Normalization (GRN)**: GRN được thêm vào ConvNeXt v2 để tăng cường tính chọn lọc và mức độ tương phản giữa các kênh, giúp giảm trùng lặp và tối ưu hóa đầu ra từ các kênh. Lớp GRN được đặt sau lớp MLP mở rộng chiều và thay thế LayerScale, vốn trở nên dư thừa, giúp mô hình hoạt động hiệu quả hơn và đơn giản hơn.

References

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10012–10022.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Trevor Darrell, and Saining Xie. 2023. Convnext v2: Co-designing and scaling convnets with masked autoencoders. *arXiv preprint arXiv:2301.00808*.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. 2022. A convnet for the 2020s. *arXiv preprint arXiv:2201.03545*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30.