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

Author
Pooja Yadav (EE23MTECH11029)
Manisha M. Malto (EE23MTECH11006)
Keshvi Dharaiya (SM23MTECH14002)
Dipesh Mishra (SM23MTECH11002)
KhomeshKumar Sahu (SM23MTECH11005)

Anonymous CVPR submission

Paper ID

Abstract

Convolutional neural networks (ConvNets) have been foundational in computer vision, but integrating them with self-supervised learning methods like masked autoencoders (MAE) has proven challenging. In this work, we propose ConvNeXt V2, a new family of ConvNets that incorporates a fully convolutional masked autoencoder framework. To address limitations in feature learning, we introduce Global Response Normalization (GRN), a novel layer that enhances feature diversity and inter-channel competition. ConvNeXt V2 demonstrates significant performance improvements in image classification and object detection tasks, showcasing the potential of combining architectural innovations with self-supervised learning.

1. Introduction

Convolutional neural networks (ConvNets) have been central to many successes in computer vision tasks, such as object recognition and image classification. However, recent developments have shown that self-supervised learning, particularly using masked autoencoders (MAE), offers significant potential by enabling models to learn from vast amounts of unlabeled data. MAE has proven highly effective for transformer-based architectures but faces challenges when applied to ConvNets due to fundamental architectural differences. Transformers, with their sequential processing nature, are well-suited for handling sparse data, where parts of the input are masked. ConvNets, on the other hand, process images through dense sliding windows, making them less compatible with the masked input paradigm used in MAE. When combined with ConvNets, MAE often yields suboptimal performance, as ConvNets are not inherently designed to work with sparse data. Therefore, this project aims to address this gap by designing an architecture that allows ConvNets to effectively leverage masked autoencoder techniques.This project aims to implement ConvNeXt V2, a novel convolutional architecture that incorporates a fully

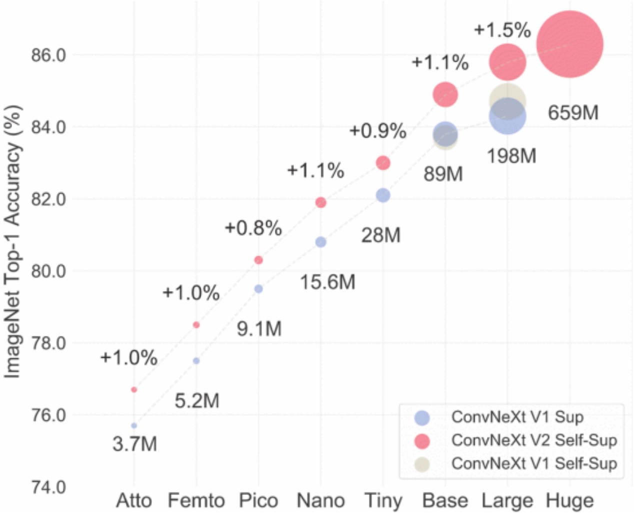


Figure 1. **Convnext v2 model scaling. The convnext v2 model, which has been pre-trained using our fully convolutional masked autoencoder framework, performs significantly better than the previous version across a wide range of model sizes**

convolutional masked autoencoder framework. ConvNeXt V2 builds upon the original ConvNeXt model by adapting its structure to handle masked input data more efficiently. Specifically, we plan to introduce sparse convolutions during the pre-training stage. Sparse convolutions will enable the network to focus on the visible, unmasked regions of the image while ignoring the masked parts, which helps prevent information leakage and improves the model's learning efficiency. In addition, to address the problem of feature collapse—where channels become inactive or redundant during training—we will introduce a novel Global Response Normalization (GRN) layer into the ConvNeXt V2 architecture. This GRN layer will enhance feature competition between channels, leading to more diverse and robust feature representations. We expect that this architectural improvement will allow ConvNeXt V2 to better capture and utilize important features across different layers of the network, particularly during self-supervised pre-training.The implementation process will involve designing the model architecture, incorporating the masked autoencoder framework, and integrating the GRN layer into each ConvNeXt block. The model will be trained on large-scale datasets such as ImageNet for classification tasks and COCO for object detection. We will also fine-tune the model on downstream tasks to evaluate its performance across a variety of benchmarks.We expect ConvNeXt V2 to deliver improved performance over existing ConvNet architectures, particularly in self-supervised learning scenarios. By co-designing the architecture with the masked autoencoder framework, we aim to bridge the gap between ConvNets and transformer-based models in terms of efficiency and accuracy in handling sparse, masked input data. The project will demonstrate that with appropriate modifications, ConvNets can effectively utilize self-supervised learning techniques and achieve state-of-the-art results in various vision tasks.

2. Literature review

The journey toward ConvNeXt V2 begins with the foundational development of convolutional neural networks (CNNs), which have been central to advancements in deep learning for visual recognition tasks. One of the first major breakthroughs in this domain was AlexNet in 2012, which demonstrated the power of deep CNNs in large-scale image recognition tasks such as ImageNet. This model was instrumental in popularizing deep learning, achieving superior performance compared to previous methods[10]. Building on AlexNet, VGGNet (2014) explored deeper networks by employing small 3x3 convolutions across many layers. While this architecture improved performance significantly, particularly in ImageNet, it also resulted in a large number of parameters, which increased computational demands [6].Further innovations came with GoogLeNet (2015), which introduced the Inception module. This architectural improvement allowed for parallel convolutions of varying sizes (1x1, 3x3, and 5x5) within the same layer, enabling multi-scale feature extraction while reducing computational costs[12]. Despite these advancements, training very deep networks led to the vanishing gradient problem. This issue was addressed by ResNet (2015), which introduced residual connections that allowed gradients to flow directly through the layers, making it possible to train networks with over 100 layers without performance degradation [6]. Building on this idea, DenseNet (2017) introduced dense connections, where each layer is connected to all subsequent layers, improving gradient flow, feature reuse, and parameter efficiency[9].As CNNs matured, the focus shifted toward self-supervised learning (SSL), which enables models to learn effective representations from unlabeled data. SSL methods such as SimCLR and MoCo utilized contrastive learning by training models to distinguish between similar and dissimilar examples, forming the foundation for future developments in masked image modeling. This approach gained traction as the success of masked language modeling (MLM) in natural language processing (NLP) inspired researchers to adapt it for visual tasks, leading to the development of masked autoencoders (MAEs). One notable framework was SimMIM, which introduced masked image modeling by randomly masking parts of an input image and training the model to predict the missing pixels. This approach demonstrated that CNNs could effectively learn representations from large-scale datasets without requiring explicit labels, reducing reliance on supervised learning[13].Masked Autoencoders (MAEs), introduced by He et al. in 2022, further enhanced this idea by utilizing an encoder-decoder structure, where the encoder processed the visible portions of the image while the decoder reconstructed the missing patches. This design enabled models to learn more generalized visual representations, which helped improve performance across various vision tasks[8]. While CNNs were being adapted to masked image modeling, Vision Transformers (ViTs) introduced a new paradigm for visual recognition. ViTs[2] treated images as sequences of patches and applied transformer models to these sequences, demonstrating that transformers could outperform CNNs when trained on large datasets like ImageNet. ViT applied self-attention mechanisms to learn long-range dependencies across the input, which marked a shift from the local receptive fields typically seen in CNNs [8][4].Building on ViTs, BEiT (2021) applied masked image modeling to transformers, showing that transformers could also learn representations by reconstructing masked patches. This extension of MAE to transformers popularized masked image modeling for self-supervised learning in vision, further challenging the dominance of CNNs in visual recognition tasks [3]. While transformers were gaining prominence, researchers explored how to integrate self-supervised learning and masked autoencoders with traditional CNNs to leverage their strengths. One significant development in this direction was Mask R-CNN, which extended Faster R-CNN by adding a branch for instance segmentation. This allowed for pixel-level object segmentation while retaining the strong detection capabilities of CNNs. Mask R-CNN showcased the versatility of CNNs in performing both detection and segmentation tasks, proving that CNNs could remain competitive with transformers[7]. In parallel, Gated Channel Transformation (GCT) introduced a lightweight method to enhance CNN performance by modeling interchannel dependencies through gated transformations. GCT improved the feature extraction process in CNNs by allowing channels to either cooperate or compete, thereby addressing feature redundancy and improving overall network efficiency. This innovation laid the groundwork for future feature normalization techniques, such as Global Response Normalization (GRN), which would be later adopted in ConvNeXt V2 [14][1]. Another significant contribution to CNN efficiency was the development of Minkowski Convolutions, which introduced sparse convolutions specifically designed for high-dimensional data such as 3D and 4D input. Minkowski Networks demonstrated the benefits of sparse convolutional operations in processing high-dimensional data efficiently, which later influenced hybrid CNN-transformer architectures and inspired the design of ConvNeXt V2's masking strategies[3][11].Combining convolutional layers with masked autoencoders, MCMAE introduced block-wise masking strategies that allowed CNNs to handle both local and global features effectively. MCMAE demonstrated that convolutional architectures could benefit from the self-supervised learning capabilities of masked autoencoders, motivating further research into the integration of MAEs with CNNs, leading toward ConvNeXt V2. Building on these innovations, ConvNeXt modernized CNNs to match the performance of Vision Transformers in image classification[5] tasks by adopting several transformer-inspired design elements, such as large kernel sizes, LayerNorm, and attention mechanisms. ConvNeXt demonstrated that CNNs, when modernized, could compete with transformers in large-scale datasets. The culmination of these developments is ConvNeXt V2, which integrates masked autoencoders (MAEs) with a modernized CNN architecture. ConvNeXt V2 co-designs CNNs to work seamlessly with MAEs by incorporating features such as Global Response Normalization (GRN), inspired by earlier innovations like GCT, to improve feature diversity and prevent feature collapse. By adopting efficient masking strategies from frameworks like MCMAE and SimMIM, ConvNeXt V2 scales effectively across various tasks, including image classification, object detection, and segmentation, while maintaining state-of-the-art performance[13][8].The development of ConvNeXt V2 represents the culmination of decades of innovations in convolutional networks, self-supervised learning, and masked autoencoders. Starting with AlexNet and VGG, CNNs laid the groundwork for deep architectures, while the rise of self-supervised learning, particularly through masked autoencoders, provided new ways to train models without relying on labeled data. Vision Transformers introduced new competition to CNNs in many domains, but by co-designing CNNs with masked autoencoders, ConvNeXt V2 represents a hybrid architecture that leverages the strengths of both paradigms, offering state-of-the-art performance across multiple vision tasks.

3. Fully Convolutional Masked Autoencoder

We propose a fully convolutional framework for masked autoencoding, where the key idea is to randomly mask visual inputs and train the model to predict the missing parts based on the visible context. This method employs a random masking strategy, where 60% of the input image is masked, and the reconstruction process is driven by the visible patches. The model is built around the ConvNeXt architecture as the encoder and employs sparse convolutions, which are inspired by techniques from large-scale 3D point cloud processing. Sparse convolutions help to efficiently pre-train the encoder by focusing only on visible parts of the image, reducing pre-training overhead and eliminating the need for masked tokens.

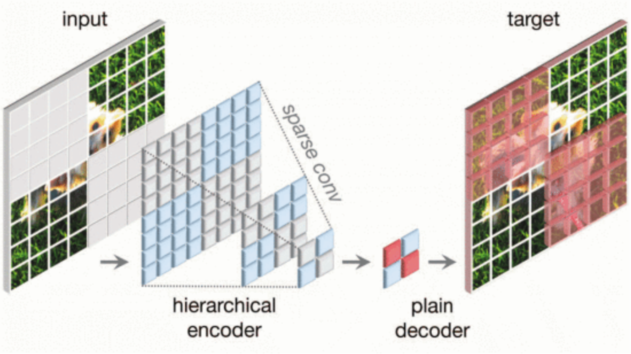
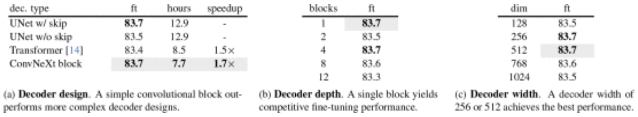


Figure 2. **Our fcmae framework. We introduce a fully convolutional masked autoencoder (fcmae). It consists of a sparse convolution-based convnext encoder and a lightweight con-v next block decoder. Overall, the architecture of our autoencoder is asymmetric. The encoder processes only the visible pixels, and the decoder reconstructs the image using the encoded pixels and mask tokens. The loss is calculated only on the masked region**

The decoder, in contrast, is a lightweight ConvNeXt block, designed to reconstruct the masked portions of the input. We utilize a mean squared error (MSE) loss, calculated only on the masked patches. This simplified yet effective approach to decoder design avoids the complexity of hierarchical decoders while maintaining strong performance in fine-tuning tasks. Through this fully convolutional masked autoencoder (FCMAE) framework, we achieved promising results, with fine-tuning performance showing a clear improvement when using sparse convolutions during pretraining. Our method is benchmarked on the ImageNet-1K dataset, yielding a significant increase in the quality of learned representations.



dec. type	ft	hours	speedup
UNet w/ skip	**83.7**	12.9	-
UNet w/o skip	83.5	12.9	-
Transformer [14]	83.4	8.5	1.5×
ConvNeXt block	**83.7**	**7.7**	**1.7×**

(a) **Decoder design**. A simple convolutional block outperforms more complex decoder designs.

blocks	ft
1	**83.7**
2	83.5
4	**83.7**
8	83.6
12	83.3

(b) **Decoder depth**. A single block yields competitive fine-tuning performance.

dim	ft
128	83.5
256	**83.7**
512	**83.7**
768	83.6
1024	83.5

(c) **Decoder width**. A decoder width of 256 or 512 achieves the best performance.

Figure 3. **Table 1. Mae decoder ablation experiments with convnext-base on imagenet-1k. We report fine-tuning (ft) accuracy (%). The pre-training schedule is 800 epochs. In the decoder design exploration, the wall-clock time is benchmarked on a 256-core tpu-v3 pod using jax. The speedup is relative to the unet decoder baseline. Our final design choices employed in the paper are marked in gray**

4. Global Response Normalization

In our research, we identify and address a key challenge in ConvNeXt models during self-supervised pretraining—feature collapse. This issue arises when neurons exhibit highly redundant activations, leading to a lack of diversity in the learned features. Specifically, we observed this problem in the multi-layer perceptron (MLP) layers of ConvNeXt blocks, where certain feature maps become inactive or saturated, significantly reducing the quality of the learned representations.

To tackle this, we introduce a novel Global Response Normalization (GRN) technique, inspired by biological mechanisms such as lateral inhibition, which promotes neuron diversity. GRN is designed to encourage feature competition across different channels, thereby increasing the contrast and selectivity of individual neurons. This ensures that each channel learns distinct and useful features, improving overall representation quality.

4.1. Mechanism of GRN

The GRN process consists of three main steps:

1. Global Feature Aggregation: First, for each feature map, we compute a global aggregation using an L2-norm operation. This step computes the norm of the spatial feature map across all spatial dimensions for each channel. Formally, for a given feature map $X \in \mathbb{R}^{H\times W\times C}$, where H and W represent the spatial dimensions and C is the number of channels, we compute:

$$G(X) = \|X_i\|_2$$

This aggregation step provides a single scalar per channel, which reflects the overall strength of the feature map in that channel.

2. Feature Normalization: After computing the global aggregation, we normalize these values to promote competition between channels. We use divisive normalization, which computes the relative importance of each channel by normalizing its aggregated value against the sum of all channel responses:

$$N(\|X_i\|_2) = \frac{\|X_i\|_2}{\sum_{j=1}^{C}\|X_j\|_2}$$

This step ensures that feature maps that are more important relative to others are given higher weight, enhancing diversity across channels by encouraging mutual inhibition.

3. Feature Calibration: In the final step, the normalized response is used to recalibrate the original feature map. The input feature map is multiplied by its corresponding normalized value and then combined with the original input in a residual fashion:

$$X_i' = \gamma \cdot (X_i \cdot N(\|X_i\|_2)) + \beta + X_i$$

Here, γ and β are learnable parameters initialized to zero, allowing GRN to initially behave as an identity function, which can adapt during training.

4.2. Implementation and Performance

The GRN unit is extremely simple to implement, requiring just a few lines of code. Despite its simplicity, GRN has a profound impact on model performance. When we incorporated GRN into the ConvNeXt architecture, the model not only mitigated the feature collapse problem but also demonstrated a notable improvement in representation quality.

We performed an extensive analysis using feature cosine distance to quantitatively validate the effectiveness of GRN. This analysis showed that GRN significantly increased the diversity of feature activations across channels compared to models without GRN. Furthermore, visualization of feature activation maps confirmed that models equipped with GRN exhibit more varied and informative activations.

4.3. Ablation Studies and Comparisons

To further understand the impact of GRN, we conducted several ablation studies:

Global Aggregation Methods: We tested different global aggregation methods and found that using the L2-norm for feature aggregation produced the best results, outperforming alternatives like global average pooling. Normalization Operators: In addition to divisive normalization, we tested other normalization techniques, such as standardization, but found that divisive normalization consistently delivered superior performance. Residual Connections: We found that adding a residual connection to the GRN block is essential for optimization, as it ensures stability during training and further improves fine-tuning performance. In these studies, GRN demonstrated a clear advantage over standard normalization methods like local response normalization (LRN), batch normalization (BN), and layer normalization (LN). Unlike LRN, which operates only on local neighborhoods, GRN contrasts features across all channels, resulting in a more global view. BN, which normalizes along the batch axis, and LN, which applies normalization across feature dimensions, also failed to achieve the same level of performance as GRN, particularly in self-supervised settings.

4.4. GRN in Pre-training and Fine-tuning

The benefits of GRN are most apparent when it is used in both pre-training and fine-tuning stages. Our experiments show that removing GRN during either phase leads to significant performance degradation, underscoring the importance of keeping GRN active throughout the training process. When GRN is applied consistently during pre-training and fine-tuning, ConvNeXt models experience a substantial boost in performance across various benchmarks.

4.5. Impact on ConvNeXt V2

By incorporating GRN, we developed ConvNeXt V2, which leverages both the architectural improvements of GRN and the benefits of masked autoencoding. ConvNeXt V2 models, ranging from small, efficient models to large, compute-intensive ones, show consistent gains over their predecessors, ConvNeXt V1. The GRN-enhanced ConvNeXt V2 models demonstrate improved performance across tasks like ImageNet classification, COCO object detection, and ADE20K segmentation.

For example, in our experiments, ConvNeXt V2 models outperformed both supervised learning baselines and other self-supervised learning methods, such as masked autoencoders (MAE) and Swin transformer models pre-trained with SimMIM. This success highlights the effectiveness of co-designing the architecture and the learning framework, particularly for self-supervised tasks.

References

[1] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3070–3079, 2019. 3

[2] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6804–6815, 2021. 2

[3] Peng Gao, Teli Ma, Hongsheng Li, Ziyi Lin, Jifeng Dai, and Yu Qiao. Convmae: Masked convolution meets masked autoencoders, 2022. 2, 3

[4] Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9224–9232, 2018. 2

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034, 2015. 3

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 2

[7] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017. 3

[8] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15979–15988, 2022. 2, 3

[9] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2017. 2

[10] Kevin Jarrett, Koray Kavukcuoglu, Marc'Aurelio Ranzato, and Yann LeCun. What is the best multi-stage architecture for object recognition? In *2009 IEEE 12th International Conference on Computer Vision*, pages 2146–2153, 2009. 2

[11] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10425–10433, 2020. 3

[12] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015. 2

[13] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: a simple framework for masked image modeling. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9643–9653, 2022. 2, 3

[14] Zongxin Yang, Linchao Zhu, Yu Wu, and Yi Yang. Gated channel transformation for visual recognition. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11791–11800, 2020. 3