Literature Review of
A ConvNet for the 2020s

arXiv link

by Ashmit Khandelwal

The main goal of the paper is to develop a new family of ConvNets, dubbed ConvNeXts, which would compete with Vision Transformers in terms of accuracy and scalability in vision tasks. Introduced in 2020, Vision Transformers (ViTs) and Hierarchical Transformers (eg: Swin Transformers) have superseded traditional ResNets by a significant margin, and have become a generic vision backbone.

ViTs have very little resemblance to ConvNets, except for an initial patch extraction layer, and use a global attention design. This absence of image specific inductive-bias is why ViTs generally struggle with vision tasks. To counteract this, Hierarchical Transformers like the Swin Transformer were developed. These reintroduced the sliding window strategy used in ConvNets, and saw rapid adoption as a generic vision backbone.

The paper highlights this fact, stating that the convolution sliding window strategy intrinsic to visual processing still remains useful. ConvNets already possess image-specific inductive biases such as translational equivalence, and are also computationally simpler. With this, the authors attempt to identify the design choices that make up Transformers (specifically, the Swin Transformer), and explore how these can be used to modernize the ConvNet. The design choices adopted by the authors as a result of their experiments are not novel. These decisions have all been researched separately as ConvNets developed. However, they have not been studied collectively with the aim of bridging the gap and competing with Transformers.

First, the authors attempted to adopt the training techniques used in the Swin Transformer into the original ResNet-50. They used the AdamW optimizer, and increased the epoch count from 90 to 3000. They also used data augmentation techniques such as Mixup, Cutmix, RandAugment, and Random Erasing, and regularization strategies such as Stochastic Depth and Label Smoothing. This modified ResNet-50's accuracy increased from 76.1% to 78.8% on ImageNet-1K. This model was used as a baseline for further modifications.

Next they moved onto Macro design, and changed the model's blocks per stage count from (3, 4, 6, 3) to (3, 3, 9, 3), matching Swin-T's stage ratio. This change increased the accuracy to 79.4%, and was adopted into the model. Then, the model's stem was changed to perform non-overlapping convolutions with kernel-size: 4 and stride: 4. This is opposed to an overlapping convolution with kernel-size: 7 and stride: 2. The model's stem becomes akin to the patch extraction layer in Swin-T. The accuracy increases slightly to 79.5%.

The authors proceeded with the use of ResNeXt's idea of "use more groups, expand width". Grouped convolution, in the form of depthwise convolution, is applied on the model. This reduces the network's FLOPs, so the network's width is increased from 64 to 96. This matches Swin-T's width, and raises the

accuracy to 80.5%. Next, the authors note that an important design in Transformer blocks is the presence of a hidden layer which is 4 times wider than the input layer. This design is akin to an inverted bottleneck with an expansion ratio of 4. Using inverted bottlenecks as the residual blocks (see Fig 3a, 3b) results in a slightly increased 80.6% accuracy.

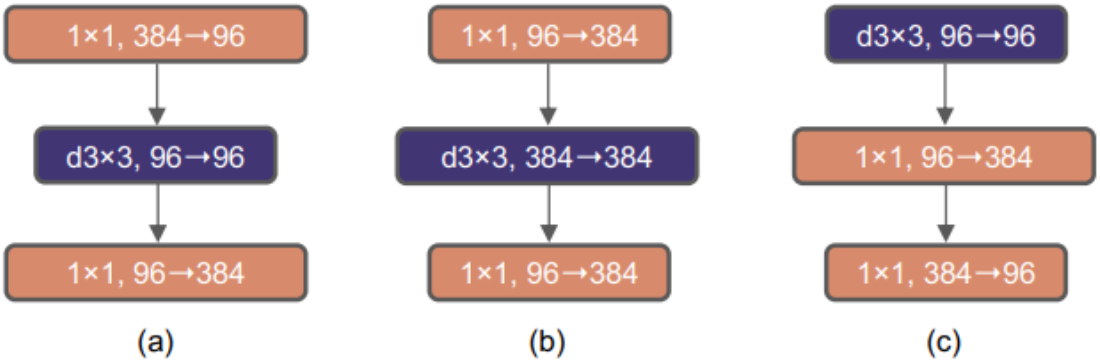

Figure 3. **Block modifications and resulted specifications. (a)** is a ResNeXt block; in **(b)** we create an inverted bottleneck block and in **(c)** the position of the spatial depthwise conv layer is moved up.

Figure 3

After this, increased kernel sizes are experimented with. To do this, the depthwise convolution layers must be moved to the start of the residual layer. This is done because large kernels are computationally complex, so it would be inefficient to apply them on 386 channels (see Fig 3b). Moving it to the start means there are only 96 channels to convolve on (see Fig 3c). Doing this reduces the FLOPs, as well as the accuracy to 79.9%. Finally the kernel size is increased, and it is observed that the network performance saturates at size 7x7. The final accuracy is the same as the inverted bottleneck step, at 80.6%.

Then, the authors swap out the ReLU activation function with GELU. Their reasoning for this is that GELU is used in ViTs. This change keeps the accuracy at 80.6%. They also replicate the Transformer block style and eliminate all GELU layers from the residual block except for one between the 1 × 1 layers. This improves the accuracy to 81.3%, matching Swin-T. Drawing again from Transformer architecture, the Batch Normalization layers are replaced with the simpler Layer Normalization layers. The number of normalization layers is also reduced to only one layer per residual block. This increases the accuracy to 81.5%.

Finally, using separate downsampling layers, along with Layer Normalization before each of them, raises the accuracy to 82.0%. Applying all the design changes performed in the experiments to ResNet yields the ConvNeXt model, which supersedes Swin Transformers.

The strengths of this paper lie in its step-by-step approach. The authors have laid out the training, macro, and micro changes they have experimented with on ResNet-50 (*Section 2*). There is a clear

path of experiments that modernize ResNet into ConvNeXt. Each of them is explained in detail, and reasoning is given for applying them.
The ResNet model is modified either to match Swin Transformers' design, or be analogous to them. The modifications are then tested on how they affect the accuracy. If the accuracy increases, they are incorporated, else further experiments are performed to boost the accuracy.

The authors have also provided details on how they have trained ConvNeXt (*Section 3.1 and Table 5*), including the data augmentation procedures, the regularizers, as well as the hyperparameters. This enables the readers of the paper to easily implement and test the model without having to worry about fine-tuning of the training procedure.

Another important strength of this paper is the fact ConvNeXts surpass Swin Transformers while still having a similar number of FLOPs. The latter's throughput is comparable to or exceeds that of the former, and show good scalability (*Section 3.2*).

A distinct weakness of this paper is that it is too focused on changing the ConvNet architecture and making it as similar to Transformer design as possible. Although this focus is somewhat justified, given that Hierarchical Transformers have superseded ConvNets and become generic vision backbones, there are cases where the ConvNet design is changed just for the sake of bringing it closer to Transformers.

The replacement of ReLU with GELU (*Section 2.6*), and the increase of kernel sizes to 7x7 (*Section 2.5*) highlight this issue. GELU is computationally more complex than ReLU. The authors also mention in *Section 2.5* that small kernel sizes, such as 3x3, have efficient hardware implementations on GPUs. These changes do not produce any change in the accuracy or the FLOPs of the ConvNeXt model. Yet, these are incorporated into ConvNext, with the reason simply being that Transformers use them too.

The paper minimally focuses on Transformer design decisions that would negatively affect ConvNets if incorporated, with the exception of separate downsampling layers. It can be built upon by performing more experiments on the differences/similarities between ConvNets' and Transformers' architectures. A general research direction would be to experiment and incorporate the design characteristics of one Machine Learning model into not only ConvNets, but any other network/model.