Review of “A ConvNet for the 2020s”

In the last decade, we saw the emerging ConvNet starting with AlexNet 2012 proceed with various model architectures for the ImageNet challenge. It has been almost 7 years since they proposed the ResNet model despite that, this model is very popular. In the year 2017, the paper “Attention is all you need” got us transformers for Natural Language Processing and then in the year 2020 Vision Transformers came into the picture which was outperforming the ConvNets in the classification tasks. The biggest challenge of ViTs was its quadratic complexity concerning the input size. Later, Swin transformers used a hybrid approach to use the sliding window strategy which was borrowed from the ConvNets. They are a major milestone work in this direction as they open up the transformers to have a generic vision backbone and all achieve state-of-the-art performance across various computer vision tasks. As many advances have been made to bring ConvNets into transformers but these attempts have come at the cost of computation as sliding window strategy can be expensive for large size images. A concurrent work ConvMixer showed that, in small-scale settings, depthwise convolution is a promising strategy. They use a smaller patch size to achieve the best results, but lose out on the throughput.

With introducing ViTs and Swin transformers, it has become a need to answer the question about the hierarchic transformer-based models (like Swin transformers) replacing ConvNets for computer vision tasks as they outperform them most of the tasks. They attribute their performance to superior scaling behaviour [1]. In this paper, the authors are re-examining the limits of ConvNets and trying to modernize them to see if they can compete with the newest architectures ViTs, Swin transformers, and hierarchical transformer-based models. As they mention their exploration is directed to the question “How do design decisions in Transformers impact ConvNets performance?”, the authors are essentially trying to figure out a way in which they can borrow the design decisions from transformers to ConvNets which can improve the performance of ConvNets. One of their primary goals is to provide an in-depth look at modernizing the ResNets and achieving state-of-the art-performance on the ImageNet dataset.

The previous attempts have tried incorporating approaches used in ConvNets into the hierarchical transformer-based models but this paper has tried incorporating the approaches from the Swin transformers into ResNet which leads to a family of ConvNeXt models which can compete with Swin transformers on most computer vision tasks like image classification, object detection, semantic segmentation. They are combining the design decisions previously used with other architectures. It is worthy to note that they provide the accuracy and GFLOPs after the changes of each step. This enables the reader to understand the importance of incorporating that design or change into the model.

The paper evaluates the model ConvNeXt on ImageNet-1K for image classification, COCO for Object detection and ADE20K for Semantic Segmentations. The reported values in the paper show that they can favourably compete with Swin transformers and shows that it outperforms them in Image Classification. They have also implemented isotropic ConvNeXt which comparable to ViT and also say that it competes with non-hierarchical models.

A detailed overview on modernizing ResNets:
The 6 major steps taken in modernizing the ResNet are setting the baseline with the training techniques similar to Swin transformers, changes in Macro design of architecture, incorporating the group convolution from ResNeXt, adding an inverted bottleneck, changing the kernel size to large, and finally some micro design changes in activation functions, normalization layers, and down-sampling layers.

The authors have trained the ResNet-50 model using the techniques used for training Swin transformers or DeiT. They used a different optimizer from the original ResNets, i.e. AdamW optimizer. They used different data

[1] Swin Transformer V2: Scaling Up Capacity and Resolution (https://arxiv.org/pdf/2111.09883.pdf)

augmentation methods like Mixup, Cutmix, RandAugment, random erasing, and some regularization schemes like Stochastic Depth and Label smoothing. The model is trained for more epochs, i.e., 300 as compared to the original, which had 90. After all, they implement these techniques; they reported it as the accuracy jumps from **76.1% to 78.8%, a jump of 2.7%** suggests that the performance difference of vision transformers could be because of the training techniques.

The next change which has been performed is in the macro design of the architecture. The computed ratio of ResNets is (3,4,6,3) whereas smaller Swin Transformers have 1:1:3:1 and bigger models have 1:1:9:1. The ratio changed to (3,3,9, s3) in such a way that the FLOPs of the model align with Swin-T. They have reported that the model accuracy improves **by 0.6% (jumps to 79.4%).** This indicates that by changing the compute ratio we can achieve better performance, therefore, conduct experiments to find the most optimal compute ratio in terms of accuracy-FLOPs trade-off. The second macro design change which has been incorporated is changing the patchify layer or how the images are processed at the beginning of the network. The paper has implemented 4x4, stride 4 with non-overlapping convolution. The increase in accuracy is **slightly (0.1%).**

The next change which was incorporated was grouped convolution inspired by ResNeXt. The main reason grouped convolution is considered is because of a better FLOPs/accuracy trade-off. A special case of group convolution where the number of groups is equal to the number of channels is called Depthwise convolution. The effect of depthwise convolution (dynamic lightweight convolution) is similar to self-attention from transformers[2]. As said earlier, the motivation to use this is to provide better accuracy/FLOPs trade-off and similar effect of self-attention as it only mixes the information in the spatial dimension. The effect brings changes in the accuracy up to **80.5% but at the cost of increased FLOPs 5.3G.** As every Transformer model has an inverted bottleneck, the authors have implemented a bottleneck where it was observed that it reduced significantly the **FLOPs to 4.6G, and accuracy was improved slightly.**

They implemented large kernel sizes of 7x7 on the motivation of self-attention having a global receptive field and Swin transformers use 7x7 window size for self-attention. The prerequisite for getting the benefit of large kernel sizes was to move the depthwise convolution layer up which is like self-attention as mentioned previously. The transformers having the multi-head attention layer before the feedforward layer inspired the motivation to shift the depthwise convolution layer. This intermediate step reduces the **FLOPs to 4.1G at cost of a decrease in accuracy** but after trying out the different sizes of kernels ranging from 3 to 11, it was observed that an increase in accuracy was saturating at 7x7 in ResNet-50 and 5x5 in ResNet-200. **It increased accuracy to 80.6%**

The last changes to incorporate to make a modernized ResNet were changing activation function, Normalization layer, and having separate downsampling layers. The activation function was changed to GeLU and the number of activation functions was also reduced. The replacement of the activation function **does not improve the accuracy** but by decreasing the number activation function does by 0.7%. The motivation to change the activation function comes from activation functions used in NLP's model like BERT and GPT-2 and to reduce the activation functions have also come from transformers who have lesser activation functions as compared to ConvNets. A ResNet has two Batch-Norm layers but transformers use lesser Layer-Norm than a usual ConvNet. By reducing the Batch-Norm layers, **accuracy surpasses the Swin-T accuracy i.e., 81.4%.** After changing Batch-Norm to Layer-Norm, only a slight increase in accuracy is seen. As Swin transformer has separate downsampling layer, therefore, a similar strategy was explored where they use 2x2 convolution with a stride of 2 but this led to diverged training, therefore several Layer-norm is introduced which helps in stabilize training.

Shortcomings of the paper

They have not provided enough reasoning to change the activation function to GeLU even though ReLU is being extensively used in ConvNets and accuracy also does not increase but decreases in ResNet-200 (as reported in

[2] Convolutions and Self-Attention: Re-interpreting Relative Positions in Pre-trained Language Models (https://arxiv.org/pdf/2106.05505.pdf)

Table 11 of Appendix). If they had decreased the activation functions first, it may explain to understand the need for the change. ConvNetXts are heavily inspired by Swin Transformers.

Some further experiments

This paper has built a base for the new novel ConvNets, which may be introduced in the upcoming years. As mentioned before, changing the activation function was not explained. I would experiment by changing the activation function back to ReLU, but keeping the number of activation functions the same. This would enable an explanation for the need to change the activation function. I could train ConvNeXt on different datasets and change the hyper-parameters provided in the original paper and study the effect of such changes. Using the backbone of ConvNeXt models for Image Super-Resolution in SRGAN's generator model, which is originally based on ResNet, we study its effect. We can explore the use of ConvNeXt backbone in Image Captioning as well[3]. Exploring more hybrid models of transformers and Convnets where we could achieve good performance in terms of accuracy, FLOPs, throughput.

[3] CPTR: FULL TRANSFORMER NETWORK FOR IMAGE CAPTIONING (https://arxiv.org/pdf/2101.10804.pd)