Patches OR Attention

Project Report

Zakariae EL ASRI
MSC AI at CentraleSupelec
zakariae.elasri@student-cs.fr

Nicolas GREVET
MSC AI at CentraleSupelec
nicolas.grevet@student-cs.fr

Our code is available at https://colab.research.google.com/drive/

1. Abstract

For many years, the mainstream architecture in computer vision was CNNs, until the time when *Vision Transformer (ViT)*, a transformer-based model shown promising performance. On later works, it was improved to outperform CNNs in many vision tasks.

Where image resolutions are very large, the quadratic computation complexity of self-attention was a major bottleneck for vision. To tackle this problem, ViTs introduced the use of patch embeddings, which group together small regions of the image into single input features. This raises the idea that gains of vision transformers are due, in part, to patch representation as input. The question is to determine which factor is more important, the patch representation or the self-attention?

In this sense, a paper recently appeared in review for ICLR 2022 **[1]**, presents a new idea in computer vision. The authors present a new architecture named ConvMixer that destroy the pyramid architecture on CNNs and replace it by an isotropic one using patches.

The paper show that the new architecture outperforms ViT, for similar parameter counts and dataset sizes.

In this project, we try to reimplement a ConvMixer on a new dataset. Then, we verify if the ConvMixers performs better than a Transformer-base model on this new dataset.

2. Introduction

After the first appearance of Transformer architectures by *Vaswani et al* **[10]**, they became the state-of-the-art for natural language processing tasks. However, its applications to computer vision was limited. While ConvNets were the state-of-the-art for many years, several works have tried to apply attention in conjunction with ConvNets, or to replace certain components keeping the overall structure in place.

In 2021, *Dosovitskiy et al* **[2]** introduced ViT, the first pure Transformer for Vision. At this time, ViT attained excellent results in image classification compared to state-of-the-art convolutional networks. Later, it was improved to outperform CNNs in many vision tasks on later works such as DeiT **[8]** and Swin transformers **[3]**.

This was a major event in computer vision field. Then, a debate on this subject has emerged within scientific community. We can distinguish two major topics: How transformers can came to such high performances? and will Transformers replace ConvNets ?

When the later consist of asking if the ConvNets are limited and will be replaced in computer vision tasks by transformers like RNN in Natural language processing. Or they are necessary, in this case, either they will remain e the go-to model for computer vision or they will be merged with transformed to make a strong architecture. In this context, *Matsoukas et al* **[6]**

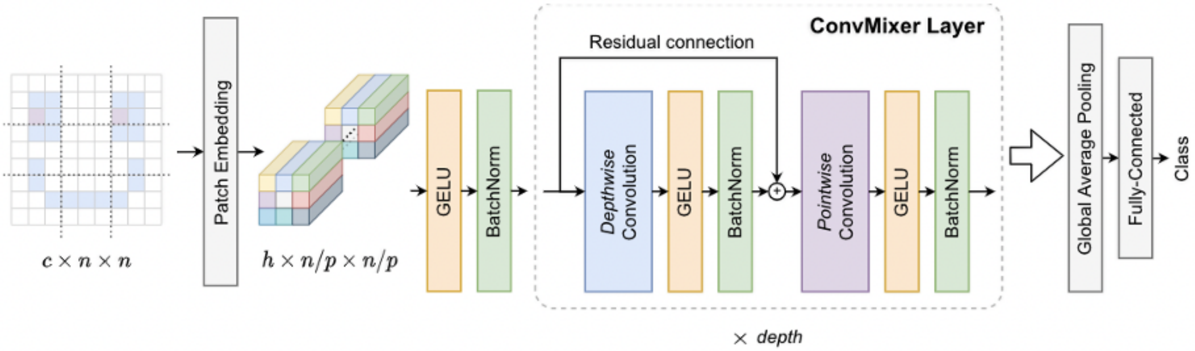

Figure 1: ConvMixer uses "tensor layout" patch embeddings to preserve locality, and then applies d copies of a simple fully-convolutional block consisting of *large-kernel* depthwise convolution followed by pointwise convolution, before finishing with global pooling and a simple linear classifier.

Time to Replace CNNs with] conclude that vanilla transformers can reliably replace CNNs on medical image tasks with little effort. When *Tolstikhin et al* **[7]** suggest a simple architecture based exclusively on multi-layer perceptrons (MLPs) to show that neither ConvNets neither ViT are necessary for vision tasks.

The former asks about the factors that yield transformer-based architecture to perform well on computer vision tasks. In this context, our work will explore the question of whether, fundamentally, the strong performance of vision transformers may result more from this patch-based representation than from the Transformer architecture itself.

3. Related Work

Given the central role of ConvNets in computer vision breakthroughs, and the remarkable results of Transformer-based architectures. This raises fundamental questions on whether factor impacts more the performance and how can this work with Convolutional architectures.

Isotropic architectures: ViT had inspired a new paradigm of "isotropic" architectures that have the same size and shape throughout the network. This architecture uses patch embeddings for the first layer.

Tolstikhin et al **[7]** were inspired by ViT and replaced the transformer-encoder blocks and the self-attention with MLPs across different dimensions and was quite performant.

Convolution in Transformers: to generalize ViT performance on image classification to other computer vision tasks (identification, segmentation), *Liu et al.* **[3]** have presented a hierarchical Transformer that reintroduced several ConvNet priors. Swin Transformer achieves the state-of-the-art performance on COCO object detection and ADE20K semantic segmentation, significantly surpassing previous best methods. Recently, *Liu et al.* **[4]** affirmed that the performance of Swin was due to inherent inductive biases of convolutions. They introduced ConvNeXt, a new model full convolutional that compete favorably with Transformers in terms of accuracy and scalability.

Attention in ConvNets: *Touvron et al.* **[9]** introduced a full patch-based ConvNet with isotropic structure. They add an attention-based pooling on top of the trunk. They demonstrated its interest on several computer vision tasks: classification, segmentation, detection.

4. ConvMixer : The model

This model destroys the historical triangular architecture of ConvNets that increases feature sizes and decreases resolution. Instead, It use an isotropic architecture similar to transformers, where the main computations are performed with convolutions instead of self-attention.

The architecture is very simple. It has a patch embedding stage followed by repeated convolutional blocks.

Patch embedding summarizes a **p×p patch** into an embedded **vector** of dimensions **e**. This is implemented by a single convolution with kernel **size p, stride p,** and **h output channels**, followed by a non-linearity. This trick will convert the **(n×n)** image into features of shape **(h × n/p × n/p)** The convolution and pooling of a typical ConvNet are replaced by repeated ConvMixer blocks.

In the final stage, features are flattened via global average pooling, and inference is made using a softmax classifier

All the operations of ConvMixer block can be implemented using only activations, BachNorma, and convolutions. So, it is just a ConvNet with some specific architectural hyper-parameters:

- Large downsampling in the initial layer but no more in the main bottleneck
- Isotropic architecture with same resolutions
- Large kernel sizes,

5. Experiment :

5.1. Dataset:

The Imagenette dataset consists of a subset of 10 classes from Imagenet (tench, English springer, cassette player, chain saw, church, French horn, garbage truck, gas pump, golf ball, parachute). It contains two versions: '320 px' and '160 px' and have a 70/30 train/valid split.

In this project, we use the 160px version, the dataset contains 13.394 images. Splitted on 9.469 on training set and 3925 on validation set (~ 70/30 split). Each class contains between 1244 and 1350 images (with a 70/30 train/valid split).

Imagenette is very similar to Imagenet, but much less expensive to deal with.

5.2. Training setup:

We trained ConvMixers on Imagenette-160 classification.

For data augmentation, we used standard techniques such as Random Horizontal Flip and Random Resized Crop.

We used a kernel size and a patch size of 7. The depth of the ConvMixer layer is set to 12. The number of hidden dimensions is 256. Of course, deeper networks take longer to converge while wider networks converge faster, so we had to adjust these dimensions accordingly.

We also focused on the Table 4 of the original paper **[1]** to determine the best settings (page 12, An investigation of ConvMixer design parameters h, d, p, k and weight decay on CIFAR-10).

Like other isotropic architecture, we used GeLu (Gaussian error linear units) for activation.

Batch size is one of the most important hyperparameters to tune in modern deep learning systems. Practitioners often want to use a larger batch size to train their model as it allows computational speedups from the parallelism of GPUs. However, it is well known that too large of a batch size will lead to poor generalization. So, we decided to use a batch size of 16 after few experiments.

Also, we used an optimization technique, the learning rate scheduling. Instead of using a fixed learning rate, we use a learning rate scheduler, which change the learning rate after every batch of training. There are many strategies for varying the learning rate during training, and the one we used is called the "One Cycle Learning Rate Policy", which involves starting with a low learning rate, gradually increasing it batch-by-batch to a high learning rate for about 30% of epochs, then gradually decreasing it to a very low value for the remaining epochs. As optimizer, we used the AdamW **[5]** which uses techniques like momentum and adaptive learning rates for faster training.

We also trained the ViT and ResNet9 architectures on Imagenette-160 with the same pipeline of data transformation and model optimization.

For the Vit, we used 6 transform layers with 8 heads in the Multi-Head Attention block, and a patch size of 4. We also used weight decay, which is yet another regularization technique which prevents the weights from becoming too large by adding an additional term to the loss function.

5.3 Results:

From the plots of the convergence of the losses vs. the number of epochs (max 30), in the three cases, it's clear from the trend that our model isn't overfitting to the training data just yet.

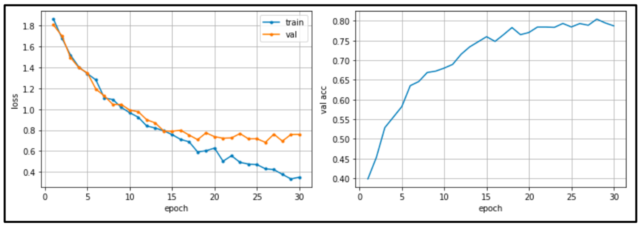

Figure 2: Loss on Train and Val sets & Accuracy on Validation_set for the ConvMixer

With a reminder that our objective was not to perform the best accuracy but to compare the three models to see that patches are a major factor in such architectures, we see that the ConvMixer model is clearly performing better with the same hyperparameters settings and with the same range in the number of parameters (between 0.8M and 1.6M).

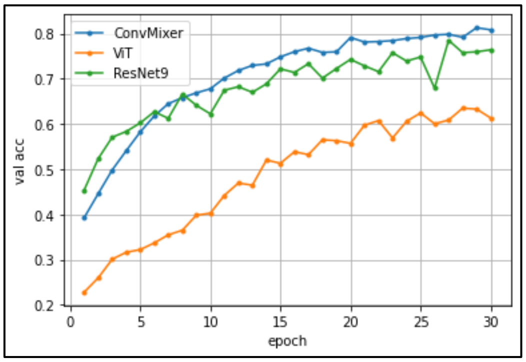

Figure 2: Comparaison of accuracies on Validation_set for the 3 models

6. Conclusion

In this project, we discovered the ConvMixer, a full ConvNet based on patch embedding in an isotropic architecture. This model seems competitive with the state-of-the-art models, it can serve as a baseline for future models, conceptually simple and performant.

Can we conclude that Patches are all we need? No! this work shows that isotropic architectures with simple patch embedding are a powerful template for deep learning. Other works showed the increasingly power of pure ConvNet models in vision tasks with the right combination of conv methods. When others examined the performance of a hybrid-model combining convolutional and transformer blocks.

At this stage, we are far from concluding either if Convolution is all we need, or Attention is all we need, or if combination is that we need.

7. References

[1] Anonymous authors, **Patches Are All You Need?** *Under review as a conference paper at ICLR* 2022

[2] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. **An image is worth 16x16 words: Transformers for image recognition at scale.** *arXiv:2010.11929*, 2020.

[3] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. **Swin transformer: Hierarchical vision transformer using shifted windows.** *arXiv:2103.14030,* 2021.

[4] Z. Liu, H. Mao, C.Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie. **A ConvNet for the 2020s.** *arXiv:2201.03545v1*, 2022.

[5] I. Loshchilov, and F. Hutter, **DECOUPLED WEIGHT DECAY REGULARIZATION.** *arXiv:1711.05101v3,* 2019

[6] C. Matsoukas, J. F. Haslum, M. Soderberg, and K. Smith. **Is it Time to Replace CNNs with Transformers for Medical Images?** *arXiv:2108.09038v1,* 2021.

[7] I. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, D. Keysers, J. Uszkoreit, M. Lucic, et al. **Mlp-mixer: An all-mlp architecture for vision.** *arXiv:2105.01601*, 2021

[8] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou. **Training data-efficient image transformers & distillation through attention.** *arXiv:2012.12877,* 2020

[9] H. Touvron, M. Cord, A. El-Nouby, P. Bojanowski, A, Joulin, G. Synnaeve, and H. Jégou. **Augmenting Convolutional networks with attention-based aggregation.** *arXiv:2112.13692v1,* 2021.

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. **Attention is all you need**. In *NIPS*, 2017