Proceedings of the 3rd International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/5/20230542

Convolutional neural network and vision transformer for image classification

Jiaqi Lu[1]
Viterbi School of Engineering, University of Southern California, 3650 McClintock Ave, Los Angeles, CA 90089, United States of America.

jiaqilu@usc.edu

Abstract. Visual Transformer (ViT) has been a hot topic for research for the past few years after it first emerged in the field. On image recognitions, due to the amount of information ViT could retrieve from the source image, in cases it can rival the traditionally prevailing Convolutional Neural Network (CNN). Then there emerged different models based on ViT, all being built having a specific field or a flaw not addressed by original ViT in mind. In this paper these models are being tested on the same dataset along with a standard CNN to see how they perform compare to each other, and the best performing ViT model was then changed to see how there would be some possible improvements.

Keywords: Visual Transformer, Convolutional Neural Network, Image Classification, Machine Learning, Computer Vision.

1. Introduction

Visual Transformer, aka ViT, as a model adapted from NLP has shown promising results in the field of computer vision as well. Ranging from image annotation to image classification, specialized visual transformers has in many cases outperformed traditional CNN models that were used for these tasks, that performance difference comes from the difference in models' structures, with ViT using attention to learn features in the picture, with a cost of learning time, it increases accuracy for computer vision.

After Visual Transformer has shown to be successful in being implemented in computer vision, many optimizations and specialization was done on the vanilla ViT model. While maintaining the original idea of using transformer to learn images, some of them changed how the transformer works, some changed how attention was applied to the model, and some combined transformer with CNN to achieve results that were previously unattainable by both models. In this article, multiple ViT's were explored on a rather small and specific classification data set to see how they perform.

2. Related work

Visual Transformers: [1] Introduced idea of visual transformer, being the first one to implement transformer in a computer vision context. By applying self-attention to draw global dependency from input to outputs. At this point this is already a mature technique, so I would not explain further into this concept.

Simple ViT: [2] : Improvement was made on top of original ViT structure, mainly aimed at simplifying original model so training would be faster and more efficient. Changes compare to

DOI: 10.54254/2755-2721/5/20230542

original models are: global average pooling was used instead of a class token, fixed 2D sin-cos position embedding, small amount of RandAugment and Mixup.

ViT for Small Dataset: [3] Proposed changes to original ViT to fit smaller datasets. Two flaws of original ViT was pointed out by the paper. First is poor tokenization and relatively poor attention mechanism. First was solved by applying Shifted Patch Tokenization to utilize spatial relations between neighboring pixels in the tokenization process. Second was solved by introduction of Locality Self-Attention, allowing ViT to have local attentions.

DeepViT: [4] It was found that when layer count exceed 12, ViT starts to struggle to attend. The paper proposed a few changes to cope with this issue: mixing attention of each MLP head post-softmax, and re-attention to combat attention collapse at higher dimensions.

CaiT: [5] Another ViT model that focuses on improving ViT performance in greater depth. It first proposes a per-channel multiplication of the output of the residual block, second it proposes to have the patches of images attend to each other, and the CLS(class) token only attend to the patches in the last few layers.

CvT & LeViT: [6][7] Putting these two together as both combine convolution and attention together to create a model for CV. Both utilizes convolution in embedding and downsampling between stages. LeViT differs in that it used extra non-linearity in attention to replace initial absolution positional bias, also replaced layer-norm with batch-norm. Whereas CvT has extra depth-wise convolution to project input for attention.

T2T ViT: [8] T2T finds that the lack of tokenization and a inefficient backbone structure to be the limitation of ViT, it proposed tokenization of images from layer to layer, downsampling of image sequences using unfolding in the first few layers. These implementations led to overlapping of image data tokens which reduces size. It also implemented a new backbone structure inspired by CNN to increase efficiency. The whole model was developed with the purpose to better image recognition by ViT, and it has proven to be able to out-perform some more complicated CNN models (e.g. ResNet) on ImageNet dataset.

3. Method

Two models used in the paper is CNN and ViT. All models are applied as they are defined, no pretraining or extra structures were applied to any of the models used in this paper. For CNN it is just the simplest one which was defined back then by the initial paper in this field. However for ViT, apart from the original model proposed, its derivations which were introduced in Related Works section are also used, in which also introduced their difference with the original model.

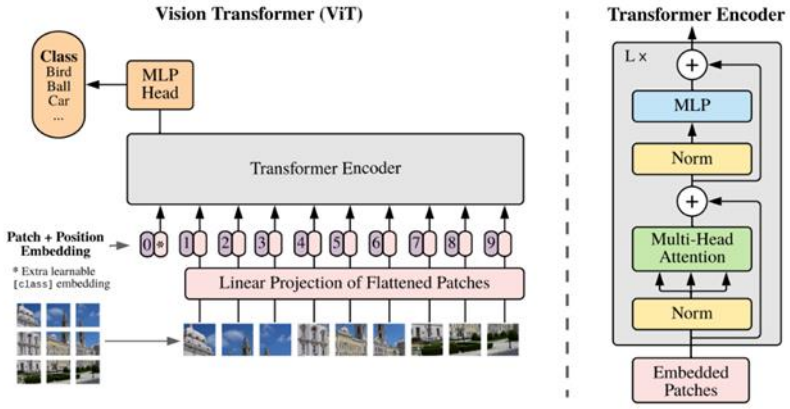

Figure 1. ViT Model Structure [1].

For CNN, a three-layer model is used. With each layer consisting of a Convolution in 2D, a BatchNorm in 2D, ReLU then Maxpool. After these three layers is a fully connected layer, a dropout

DOI: 10.54254/2755-2721/5/20230542

layer, another fully connected layer and then finally another ReLU. The loss function here is a Cross Entropy Loss function.

For ViT, since instead of a string it takes an image as input instead of the 1D sequence of token embeddings when it is used in NLP, it first reshapes the image which is represented in 3D vectors to a sequence of flattened 2D patches. Then these patches are passed through a patching and position embeddings, then all feed into a transformer encoder which structure may depend on different implementation. The transformer output is then feed into a MLP head which classifies image into different classes. In the original model, the transformer encoder consists of a norm, then a multi-head attention block, then a norm, and a final MLP.[1]

4. Experiment

For this project, multiple models were tested on the same dataset to see how they perform, and the last one was changed on some level to try to see how the performance improves. The dataset was collected by Kaggle, named "Cats vs. Dogs" dataset which contain 25000 images for classification between cats and dogs, 20000 are used for training and 5000 for validation [9]. This is a rather small dataset compare to the mainstream computer vision training datasets, but nevertheless, it was chosen for this project to see how models would perform on this rather small dataset.

Then different models mentioned in Related Works section are imported without pretraining, meaning only their basic structure was imported, thanks to Phil Wang (GitHub Username: lucidrains) who implemented them in a python package for PyTorch. All models were ran with 40 epochs to see how they perform, and they were all run on a single Nvidia A100 GPU, everything implemented in PyTorch. A standard CNN model was also implemented an ran in the same conditions to set a benchmark for the models to see how they perform compare to the traditional way of approaching this task.

Table 1. Comparison Between Different Models.

Epoch/ Model	CNN	Simple ViT	SmallViT	CaiTViT	DeepViT	CvT	LeViT	T2TViT
20 epochs								
val_loss/	0.797/	0.712/	0.734/	0.715/	0.676/	0.794/	0.783/	0.807/
val_acc	0.422	0.554	0.520	0.550	0.600	0.438	0.462	0.415
40 epochs								
val_loss/	0.830/	0.736/	0.784/	0/748/	0.707/	0.839/	0.799/	0.877/
val_acc	0.372	0.521	0.451	0.506	0.558	0.355	0.432	0.287

From Table 1, it is able to see that some models, due to the fact that they are built for different purpose than image classification on a relatively small dataset, their performance was even worse than a standard CNN model, e.g. Deep ViT, for this project never needed that deep of a layer structure. Some are around the same performance level as CNN but considering the tradeoff ViT in general makes for learning more information, the time that spent on training was not worth the performance. It was honestly surprising to see that ViT for Small Dataset did not work as well as expected on this dataset, it is better than original ViT, but still lower than CNN. Then there are also the ones that performs better than CNN. CvT and LeViT are two that combines CNN and transformer, which is not surprising for them to be better than CNN. The standout is T2T-ViT. Although with some of its structures inspired by CNN, the massive change to tokenization on the original ViT model and a backbone structure change all contributed to this model outperforming others on image classification task, as it is built for that.

After determining that T2T-ViT is the superior model for this specific task, some changes were done to it to see how its performance could be affected. As said in the paper, T2T-ViT's main work was in the T2T process part of the structure, which contains the tokenization process. This is

DOI: 10.54254/2755-2721/5/20230542

something that is quite complicated to change, but the transformer that it uses is easy to change. With the PyTorch implementation of this model, the model either takes in a pre-programmed transformer, or a set of inputs that it uses to build a transformer. Both of which are tried in the paper, with the result shown below.

With randomly tuning the model's hyperparameter, the result does seem random with no real sign of showing which direction should the parameters be changed to better the whole model. However an interesting discovery is that with Linformer [10] used as the transformer for the model, it learned faster in the first 20 epochs, but seemingly at a trade-off with the upper limit of learning accuracy.

Table 2. T2TViT vs T2TViT+Changed Transformer.

Epoch/Model	T2TViT	T2TViT + Linformer
20 epochs		
val_loss/	0.807/	0.825/
val_acc	0.415	0.390
40 epochs		
val_loss/	0.877/	0.843/
val_acc	0.287	0.350

Table 3. Comparison with CNN and T2TViT.

Epoch/Model	CNN	T2TViT
40 epochs		
val_loss/	0.830/	0.877/
val_acc	0.372	0.287
60 epochs		
val_loss/	0.840/	0.904/
val_acc	0.355	0.226
80 epochs		
val_loss/	0.846/	0.916/
val_acc	0.333	0.208
100 epochs		
val_loss/	0.856/	0.927/
val_acc	0.318	0.194

Another observation was made regarding convergence of models. CNN did converge within 100 epochs, and it capped at a certain performance. However, running T2T-ViT to 100 epochs shows some sign of convergence, but did not converge, and have surpassed CNN at around 80 epochs. This alone should be used as evidence to praise ViT for its performance, for what really matters at end of the day it where this model caps, as time and space occupancy can all be solved by simply getting more hardware.

5. Conclusion

From this project, it can be said that ViT, though on small datasets usually cannot win against CNN, should still be considered with it comes to these kinds of tasks. As shown by T2T-ViT, specialization and optimization on certain aspects of the model does provide it an edge over CNN, most notably its upper limit is higher than CNN, in an era where especially for small dataset, time and space is not the main concern. In the future, more dataset should be tested to make sure conclusion derived from this project is applicable to all, and as for T2T-ViT, more changes could be made to test each part of the model, and another direction to take is its tokenization process, to change it to fit the task more.

DOI: 10.54254/2755-2721/5/20230542

References

[1] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, et al. (2021). "An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale." arXiv.org

[2] Lucas Beyer, Xiaohua Zhai, and Alexander Kolesnikov. (2022). "Better Plain ViT Baselines for ImageNet-1k." arXiv.org

[3] Seung Hoon Lee, Seunghyun Lee, and Byung Cheol Song. (2021). "Vision Transformer for Small-Size Datasets." arXiv.org

[4] Daquan Zhou, Bingyi Kang, Xiaojie Jin, Linjie Yang, Xiaochen Lian, Zihang Jiang, Qibin Hou, and Jiashi Feng. (2021). "DeepViT: Towards Deeper Vision Transformer." arXiv.org

[5] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. (2021). "Going Deeper with Image Transformers." arXiv.org

[6] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. (2021). "CvT: Introducing Convolutions to Vision Transformers." arXiv.org

[7] Ben Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. (2021). "LeViT: a Vision Transformer in ConvNet's Clothing for Faster Inference." arXiv.org

[8] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zihang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. (2021). "Tokens-to-Token ViT: Training Vision Transformers from Scratch on ImageNet." arXiv.org

[9] Chen, Chun-Fu Richard, Quanfu Fan, and Rameswar Panda. (2021). CrossViT: Cross-Attention Multi-Scale Vision Transformer for Image Classification. In 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, pp. 347–356. doi: 10.1109/ICCV48922.2021.00041.

[10] Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. (2020). "Linformer: Self-Attention with Linear Complexity." arXiv.org