VISION TRANSFORMERS VS. CONVOLUTIONAL NEURAL NETWORKS

Shawn Diaz

ABSTRACT

Convolutional neural networks (CNN) have been known for some time to be effective in computer vision applications. However, transformers have recently been shown to also be effective at the same tasks. Using three differently sized transformers and CNNs, comparisons were made between these two styles of neural network architectures. Evaluation was done on three publicly available datasets: CIFAR-10, CIFAR-100, and MNIST Fashion. After training and doing evaluations, it was determined that the vision transformer networks showed competitive performance when compared to their similarly sized CNN counterpart. Furthermore, with the more difficult datasets, the vision transformer networks outperformed the CNN networks. Therefore, it was concluded that vision transformers are at least on a par with CNNs, and show promise for tasks in the field of computer vision.

1. INTRODUCTION

Transformers have been found to be useful in natural language processing (NLP) tasks due to their ability to attend to previous relevant tokens, or words, through an attention-based encoder-decoder architecture [1]. The encoder works by mapping an input sequence into a continuous representation. The decoder then uses the continuous representation and previous outputs to generate an output sequence. Because of the how the attention mechanism works, transformers have been able to solve the short-term memory issues which affected recurrent neural networks, including long short-term memory networks. After the success of Transformers, it was shown that these same mechanisms could also work well with imagine processing tasks.

A Vision Transformer (ViT) is a Transformer that was created for use in applications such as image recognition [2]. ViT networks work by reshaping an image into a sequence of fixed-size patches, which are linearly embedded. Then positional embeddings are prepended to the resulting sequence of vectors, before being served as input to the Tranformer encoder. The encoder used in the ViT architecture is identical to the encoder used in a standard Transformer. However, no decoder is used in the ViT architecture. The output of the encoder is then used as input into a multilayer perceptron head, before being fed into a classifier.

Because a ViT can effectively do the same tasks as a Convolutional Neural Network (CNN), it would be interesting to do comparisons between these two approaches. In this experiment, comparisons were made between three differently sized, yet relatively comparable, versions of each approach. From smallest to largest, these sizes include: tiny, small, and base. Further details about how the sizes of each network were determined are discussed in section 3. Finally, evaluation of each model was done using each of three publicly available datasets: CIFAR-10, CIFAR-100 [3], and MNIST Fashion [4].

2. RELATED WORKS

In the last few years, a lot of focus has been put on research pertaining to transformers and their ability to work with image classification problems. Dosovitskiy et al. [2] announced their ViT network architecture as a Transformer capable of classifying images, similar to what CNNs had previously been thought to be the best at. After training and evaluating their model, they found that the self-attention mechanism allowed for the entirety of the input images to be attended to early on in the lowest layers. This confirmed that the model was able to integrate and use information globally. Furthermore, they found that their larger models only performed better than state-of-the-art CNN architectures when trained on large datasets. Possibly indicating that ViT networks need large amounts of data to scale well.

Wang et al. [5] presented improvements to a previous Pyramid Vision Transformer (PVTv1) by incorporating three new designs. First, they added a linear complexity attention layer. Then they added an overlapping patch embedding. Lastly, they added a convolutional feed-forward network. These additions to their baseline model, called PVTv2, reduced the computational complexity and showed significant improvements to vision classification tasks. They did an ablation study and found that all three designs improved the model performance, while reducing computation overhead and parameter number. Their largest model achieved 83.8% accuracy on ImageNet.

Li et al. [6] showcased a type of Vision Transformer called Separable Vision Transformer (SepViT). SepViT was developed to address the large computational requirements of ViT networks. Their network architecture uses a window-base self-attention, which is more computationally efficient than the full-attention machanism found in ViT. It then uses a depthwise self-attention module to capture local features within each window, and pointwise self-attention for building connections between windows. Alongside these optimizations, they developeda window token embedding, which models relationships between windows with little or no performance cost. Lastly, they use a grouped self-attention to capture more context across windows and improve performance. With these optimizations to performance, SepViT achieved 84.0% accuracy on ImageNet-1K, while decreasing latency by 40%.

Ranftl et al. [7] introduced a Vision Transformer architecture for dense prediction tasks. They accomplished this by assembling image-like representations at various resolutions to create full-resolution predictions. They found the transformer was able to process the images at a relatively high resolution. Furthermore, the global receptive field of the transformer gave their architecture finer-grained predictions. In their tests using monocular depth estimation, they found an improvement of up to 28% when compared with a state-of-the-art fully-convolutional network.

Yuan et al. [8] recognized one limitation of Vision Transformers to be that they perform worse than CNNs on midsize datasets, such as ImageNet, without pretraining. They found this to be because of two reasons: 1) the simple way in which ViT tokenizes the input images fails to represent edges and lines of neighboring pixels, leading to poor efficiency; 2) the attention design of ViT creates limited feature richness with smaller computation budgets. To fix these issues, they presented the Tokens-To-Token Vision Transformer (T2T-ViT), which uses a layer-wise Tokens-to-Token (T2T) transformation. This module progressively tokenizes images to tokens, iteratively reducing the length of tokens. They also used different ViT backbone to reduce redundancy and improve feature richness. Thus, their model achieved an accuracy of 83.3% when trained on ImageNet.

Caron et al. [9] question the new possibilities with Vision Transformers in the context of self-supervised learning. They do this by detailing a self-supervised method called DINO, which uses a self-distillation process with no labels. The self-distillation process works by passing two different randomly transformed versions of an image as input to student and teacher networks. The student network propagates gradients through a stop-gradient operator. The teacher network parameters are updated using an exponential moving average of the student network parameters. The similarities of the outputs of the networks are then measured using a cross-entropy loss. Their model was trained on ImageNet and achieved 80.1% accuracy.

Raghu et al. [10] did an analysis on the differences in performance between Vision Transformers and Convolutional Neural Networks on image classification tasks. After doing their analysis, they found that there were clear differences between the internal structures of ViT networks and those of CNNs. From an in-depth analysis on self-attention and the strength of skip connections, they found that earlier global features and stronger representation propagation were factors in explaining how ViT networks can offer performance similar to state-of-the-art CNNs. They also found that CNN properties, such as local information aggregation at lower layers, are important to ViT networks when without pretraining. Lastly, they found larger ViT models were able to develop stronger representations through the use of larger datasets for pretraining.

He et al. [11] demonstrated dozens of procedure refinements to Convolutional Neural Networks through an empirical process. They implemented and analyzed various refinements such as using label smoothing, knowledge distillation, cosine learning rate, large-batch training, low-precision training, among other things. They found that when the modifications they tested were used in conjunction, their model showed significantly higher accuracy. They also concluded that their improved pre-trained models showed strong advantages when used in transfer learning, which help with object detection and semantic segmentation.

Mingxing Tan and Quoc V. Le [12] produced a study on model scaling in order to identify a way to balance network depth, width, and resolution to improve model performance. They were able to accomplish this task by proposing a new scaling method for CNNs that uniformly scales all dimensions of depth/width/resolution through the use of a compound coefficient. They implemented this method on a baseline network to create a family of models they called EfficientNets. One of their models, EfficientNet-B7, achieved 84.3% accuracy on ImageNet, while being multiples faster and and smaller than the previous best CNN architecture.

Liu et al [13] attempt to modernize CNNs by investigating the practices with training CNNs, and the design decisions of Transformers, to formulate a new architecture called ConvNeXt. They started with a baseline ResNet model and trained the model with an improved procedure. Through their testing, they discovered several components that contribute to the performance of Transformers, which could be implemented in CNNs. In their design considerations, they focused on the stage compute ratio, and the ”stem cell” structure. They changed the stage compute ratio by adjusting the number of blocks in each stage of the ResNet more closely follow the stage compute ratios of Swin Transformers. Furthermore, they adjusted the stem cell design to a ”patchify” design, as is used in Swin Transformers, for downsampling purposes. After implementing these changes to their baseline ResNet architecture, their ConvNeXt model achieved 87.8% accuracy on ImageNet.

3. METHOD

The Vision Transformer architecture that was used for the experiment is an implementation of the ViT model created by Dosovitskiy et al. [2] for image classification tasks. The ViT model consists of several Transformer blocks, which contain a multihead attention layer as a self-attention mechanism. This gets applied to a sequence of patches that were input to the network, before outputting a tensor which gets processed by a classifier head with a softmax; This outputs a probability distribution over the classes of the dataset. The ViT networks were found to converge much slower than the CNN networks. Therefore, training was done over 50 epochs for the ViT networks. Gaussian error linear unit (GELU) was used for the activation function since it has been shown to perform better than ReLU; GELU nonlinearity weights inputs by value, rather than by their sign as in ReLU [14]. For the optimizer, AdamW was used with a 0.001 learning rate. AdamW is an implementation of the Adam algorithm with weight decay [15]. Finally, sparse categorical cross entropy was used for the loss function.

The CNN architecture that was used consist of either 5, 9, or 15 layers for tiny, small, and base sizes, respectively. In all three networks, there are a series of convolutional layers with differing number of filters, followed by a batch normalization layer, pooling layer, a dropout layer, and finally a fully connected layer. The CNNs converged quicker than the ViT networks. Therefore, 15 epochs was enough for all three sizes. ReLu was used for the activation function throughout all of the CNNs. Sparse categorical cross entropy was used for the loss function. Lastly, RMSprop was used for the optimizer due to it showing better performance than Adam with the tests that were run.

Since comparisons were made between differently sized networks, it was important to ensure each comparison was fair. In order to ensure that the comparisons were fair, adjustments were made to each network in such a way that each pair of models that were compared contained a relatively similar number of parameters. For the Vision Transformer networks, size was determined by adjusting the number of Transformer blocks in the network, and by adjusting the dimensions of the multilayer perceptron (MLP) head units; The MLP heads contain the dense layers of the final classifier for the network. For the CNN networks, size was determined by adjusting the number of layers and the final dense layer dimension. The models and their parameters can be seen in Figure 1 and Figure 2.

Model	Layers	Dense Dim	Parameters
CNN-Tiny	5	512	6M
CNN-Small	9	1024	12M
CNN-Base	15	2048	24M

Table 1. **CNN Network configurations.** ”Layers” is the number of layers with trainable parameters present in the network. ”Dense Dim” is the dimensions of the dense fully connected layer. ”Parameters” is the total number of parameters in the network.

Model	Blocks	MLP Dims	Parameters
ViT-Tiny	4	512, 1024	6M
ViT-Small	6	1024, 2048	12M
ViT-Base	8	2048, 2048	24M

Table 2. **ViT Network configurations.** ”Blocks” is the number of Transformer blocks present in the network. ”MLP Dims” is the dimensions of the dense layers in the final classifier. ”Parameters” is the total number of parameters in the network.

4. EXPERIMENTS AND RESULTS

For the experiment, three different datasets were used: CIFAR-10, CIFAR-100, and Fashion MNIST. CIFAR-10 is comprised of 60,000 32x32 colour images in 10 classes, giving 6,000 images per class. The 10 classes include: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. Each class contains 5,000 training images and 1,000 randomly selected testing images. CIFAR-100 is similar to CIFAR-10, except it consists of 100 classes containing 600 images each; 500 of those are training imagines, and 100 are testing images. The dataset contains 20 superclasses, which each contain several classes. Examples of some of the superclasses in the dataset include: aquatic mammals, fish, flowers, food containers, fruit and vegetables, household electrical devices, household furniture, and insects. Lastly, Fashion MNIST was also used for testing. Fashion MNIST consists of 70,000 examples of 28x28 greyscale images, labelled from 10 classes. 10,000 of these make up the test set, while the remaining 60,000 make up the training set. These classes include: T-shirt/top, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, bag, and Ankle boot.

Model	Accuracy (%)	Loss
ViT-Tiny	50.39	1.86
ViT-Small	50.40	1.88
ViT-Base	51.91	1.92
CNN-Tiny	32.28	3.01
CNN-Small	42.32	2.62
CNN-Base	43.74	2.66

Table 3.
Comparisons between the accuracy and loss metrics of the different sized networks after training and testing on the CIFAR-100 dataset.

After training each model, evaluation was done and comparisons were made between each size of model. When looking at the performance of the Vision Transformer between the three different sized networks, there was a relatively uniform accuracy. For each of the datasets, the accuracy did not change much when changing network size. Whereas, the CNNs performed better with an increase in size. For instance, the tiny CNN yielded an accuracy of 32.28% on the CIFAR-100 dataset, but the small CNN produced an accuracy of 42.32%. On the same dataset, the tiny ViT network had an accuracy of 50.39%, and the small ViT an similar accuracy of 50.40%. These results can be seen in Table 3. The reason for this could be that the ViT networks need more data to take advantage of the increase in network size. With the Fashion MNIST dataset, seen in Table 4, the CNNs did not change much between network sizes. This could be explained by the relatively low difficulty for CNNs, as well as ViT networks, to learn that particular dataset. So in the case of Fashion MNIST, there would need to be further and more advanced optimizations, such as regularization, to see further improvements. The CIFAR-10 dataset showed similar metrics. Presented in Table 5, the ViT network performed very similarly across all network sizes. And the CNNs showed steady improvements as network size increased. This demonstrates that ViT networks are able to perform very well with enough data. Furthermore, smaller ViT networks are almost as performant when compared to larger ViT networks.

Model	Accuracy (%)	Loss
ViT-Tiny	91.94	0.21
ViT-Small	92.35	0.21
ViT-Base	91.56	0.24
CNN-Tiny	91.87	0.31
CNN-Small	89.79	0.46
CNN-Base	91.59	0.40

Table 4.
Comparisons between the accuracy and loss metrics of the different sized networks after training and testing on the Fashion-MNIST dataset.

Model	Accuracy (%)	Loss
CNN-Tiny	66.88	1.17
CNN-Small	74.67	1.05
CNN-Base	78.13	0.95
ViT-Tiny	79.98	0.58
ViT-Small	80.69	0.57
ViT-Base	80.25	0.58

Table 5.
Comparisons between the accuracy and loss metrics of the different sized networks after training and testing on the CIFAR-10 dataset.

In Figure 1 and Figure 2, the learning curves for the base sized networks can be seen. The CNNs converged much more quickly in training than the ViT networks. However, their overall performance was worse. The ViT networks took longer to reach convergence, but showed noticeably better performance; This was especially true on the more challenging CIFAR-100 dataset. The main takeaway from these comparisons is that the ViT networks perform least as well as the CNN networks, if not noticeably better. And they also scale better with larger datasets.

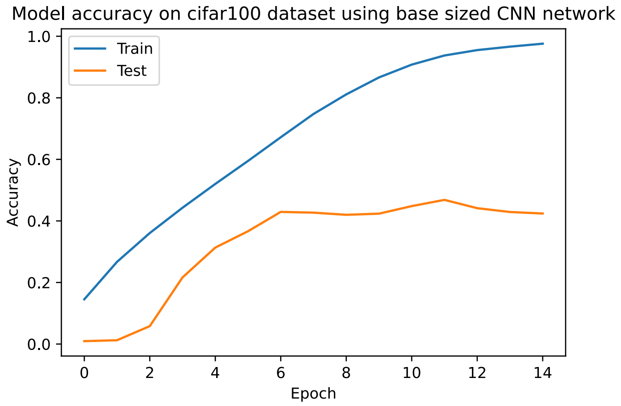

Fig. 1. A comparison between the learning curve of the base CNN training and testing on the CIFAR-100 dataset. The model converges relatively quick, at around 15 epochs.

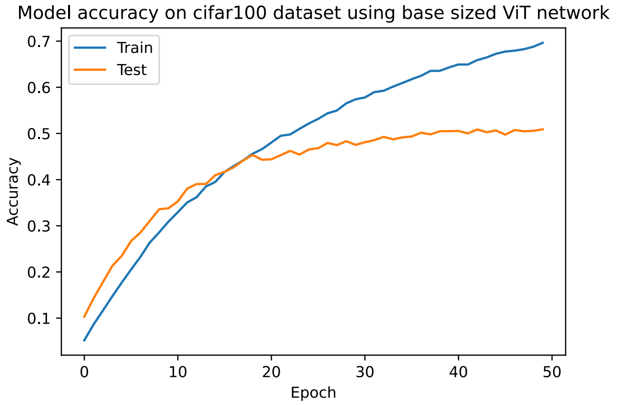

Fig. 2. A comparison between the learning curve of the base ViT network training and testing on the CIFAR-100 dataset. The model converges at around 50 epochs.

5. CONCLUSION

The comparisons between Vision Transformers and Convolutional Neural Networks showcase the potential that Transformers have in competition with CNNs for image recognition tasks. The recently discovered abilities of Transformers in image processing tasks is promising for the future; By evaluating differently sized Transformers, and making comparisons with similarly sized CNNs, the claimed competitive performance of ViT networks was confirmed. Furthermore, it was demonstrated that ViT networks scale well with larger datasets. Whereas, CNNs might have the advantage on smaller datasets and in computationally restricted situations. In conclusion, Vision Transformers are capable of performing well at image classification tasks, especially ones with large amounts of data and computational power to work with.

6. REFERENCES

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin, “Attention is all you need,” 2017.

[2] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby, “An image is worth 16x16 words: Transformers for image recognition at scale,” 2020.

[3] Alex Krizhevsky, “Learning multiple layers of features from tiny images,” 2009.

[4] Han Xiao, Kashif Rasul, and Roland Vollgraf, “Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms,” 2017.

[5] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao, “Pvtv2: Improved baselines with pyramid vision transformer,” 2021.

[6] Wei Li, Xing Wang, Xin Xia, Jie Wu, Xuefeng Xiao, Min Zheng, and Shiping Wen, “Sepvit: Separable vision transformer,” 2022.

[7] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun, “Vision transformers for dense prediction,” 2021.

[8] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zihang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan, “Tokens-to-token vit: Training vision transformers from scratch on imagenet,” 2021.

[9] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin, “Emerging properties in self-supervised vision transformers,” 2021.

[10] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy, “Do vision transformers see like convolutional neural networks?,” 2021.

[11] Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li, “Bag of tricks for image classification with convolutional neural networks,” 2018.

[12] Mingxing Tan and Quoc V. Le, “Efficientnet: Rethinking model scaling for convolutional neural networks,” 2019.

[13] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie, “A convnet for the 2020s,” 2022.

[14] Dan Hendrycks and Kevin Gimpel, “Gaussian error linear units (gelus),” 2016.

[15] Ilya Loshchilov and Frank Hutter, “Decoupled weight decay regularization,” 2017.