Generalizability Of Transformer-Based Models Over Small-Scale Computer Vision Tasks

Sam Kaho Tam
The Chinese University of Hong Kong
1155175983@link.cuhk.edu.hk

Abstract

Recently, vision Transformers (ViTs) have demonstrated remarkable performance on large-scale datasets. Yet, they tend to underperform for small-scale datasets when trained from scratch [1], which can cause difficulties in training on datasets for speciallized domains where data collection is expensive, and in cases when transfer learning is not effective.

In this work, we analyze and compare the performance degradation for convolution-based models like ResNet and Transformer-based models like ViT and Data Efficient Image Transformer (DeiT) on small subsets of CIFAR-10 and STL-10 datasets, and observed that vanilla ViTs, given the same input shapes and number of parameters, is less efficient and have lower accuracy on smaller datasets compared to convolution-based models.

1 Introduction

While Vision Transformers (ViTs) have demonstrated remarkable performance on large-scale datasets, they tend to underperform for small-scale datasets when trained from scratch [1]. While collecting labelled data could be expensive on speciallized domains, understanding how Transformer-based models adapt to smaller datasets can be beneficial for practical applications, especially on image sets that may share less common features with the larger datasets available, which may reduce the effects of transfer learning.

As recent variants of ViTs are developed to address this problem, we aim to explore their capabilities on generalizing over computer vision datasets of smaller scale, comparing with several non-Transformer-based models as the baseline. In particular, we would focus on the image classification task.

2 Related Work

CNN and Variants. Convolutional neural networks (CNNs) have been the standard network model throughout computer vision given their capabilities in learning and generalizing spatial features through convolutions with kernels. Res-Net [2] and U-Net [3] have demonstrated success in various computer vision tasks including image classification and semantic segmentation by utilizing skip connections.

Transformer based vision backbones. While CNN and its variants are still widely used, ViTs displayed high performance over larger datasets compared to even ResNet [4]. However, the initial versions of ViT suffered from worsened performance for relatively small datasets. According the Chen & Wang et al, this problem can be attributed to the excessive long-range attention as the amount of data decreases [5]. In this project, we aim to investigate the effectiveness of various methods on the quality of feature-extraction of the Transformer-based models like DeiT [6] and T2T-ViT [7] compared to traditional CNN-based models over small datasets.

Comparison study of different computer vision models. In the previous work of Khan et al. [8], the relatively high parametric complexity and computational cost of Transformer models have been acknowledged. Additionally, scaling studies conducted by Dosovitskiy et al. [4], Zhai et al. [9] and Bhojanapalli et al [10] also showed that the accuracy of ViT models outperforms ResNets and scales better for sufficiently large datasets, as shown in Figure 1.

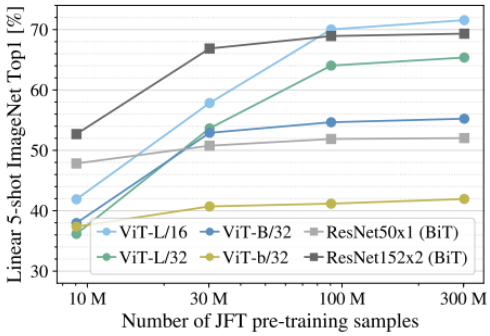

Figure 1. Scaling comparison of ViT and ResNet on large datasets. Image from [4].

3 Approach

3.1 Data

In this project, we evaluate our models on two datasets, CIFAR-10 [11] and STL-10 [12], which consists of 50k training samples of image size 32x32 and 5k training samples of image size 96x96 respectively. Both datasets have the same 10 classes and the results are thus comparable. Furthermore, to investigate the generalization power of the models on smaller datasets, we will also be training the models on class-balanced subsets of the above datasets with different proportions, ranging from 100% down to 25% of the original sizes. This would contribute to a range of sizes of training sets as shown in Table 1.

Table 1. Sizes of subsets used for training

Dataset ↓	**1.00x**	**0.90x**	**0.70x**	**0.50x**	**0.25x**
CIFAR-10	50,000	45,000	35,000	25,000	12,500
STL-10	5,000	4,500	3,500	2,500	1,250

3.2 Methodology

In this paper, we compare the performances of the models in the following aspects:

Scalability. To explore how the models generalize with small datasets, we trained the models on subsets of CIFAR-10 and STL-10 datasets, exploring different sizes of the subsets and observe the caused reduction in accuracy. The models are then compared to common convolution-based models like ResNet, trained on the same subsets.

Computational Performance. We compare the number of iterations and time resources spent to train each model to convergence on a P100 GPU. The number of iterations is obtained by multiplying the sizes of training sets with the number of epochs to convergence. We determined convergence of the models by measuring 20 epochs of stable reported train and test accuracies.

Models. We attempt to train models with similar amount of parameters for fairness. For convolution-based models, we chose ResNet-34; For transformer-based models, we chose a self-implemented version of ViT-S [4][13], DeiT-S-distillated [6][14] and Token-to-Token ViT (T2T-ViT) [7][15]. All models are adjusted to have a output dimension of 10. The specifications of the models are listed in Table 2.

Table 2. Specifications of Models Used

Model ↓	# Parameters	Input Sizes
ResNet-34	21M	64x64 and 224x224
ViT-S* (SimpleViT)	24M	64x64
DeiT-S-distillated-patch16	22M	224x224
T2T-ViT-14-wide	21M	224x224

Note that although we are using the implementation from the *knowledge-distillated* version of DeiT, in the experiments conducted in this work the DeiT model is *not* trained with the distillation loss. Thus, the DeiT-S model used should perform similarly as a reduced version of ViT-B. There are also some minor differences between the actual ViT-S and our implemented version of it (SimpleViT), so we will mark it with ViT-S* instead.

Devices. The models are mainly trained on two devices, including a P100 GPU with 16GB memory on the Kaggle platform, and a Nvidia RTX 3060-Ti GPU with 16GB memory. During evaluation on the computational efficiency and the time taken for models to converge, we calculate the training time per 1k iterations on the P100 GPU for each model, and assume that the time taken scales approximately linearly with the number of iterations or batches taken.

4 Experiments

This section presents a few quantitative experiments and results. We first dicuss the scalability over accuracy, followed by the computational performance. The results from multiple trials are manually collected into a CSV file before plotting.

4.1 Scalability

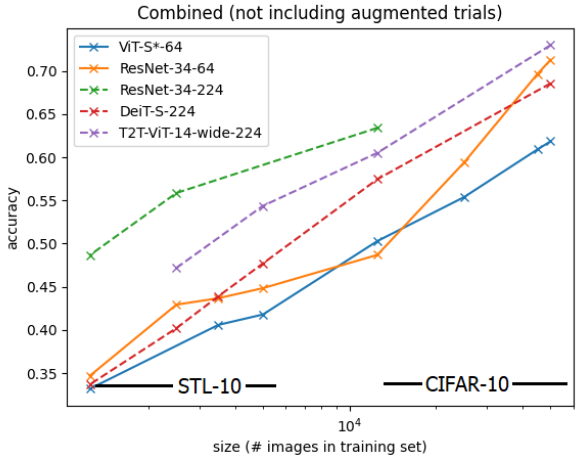

Figure 2. Scaling comparison for models trained on small data subsets.

Following Figure 2, we can observe that for models with the same input size, transformer-based models achieve significantly lower accuracy. This shows that transformers do not generalize very well on small datasets. We can also observe that this difference seem to worsen as the input size increases. This result is not unexpected, as transformer-based models relies on sufficient data for the attention layers to learn positional relations, which is not a problem for convolution-based models.

4.2 Computational Efficiency

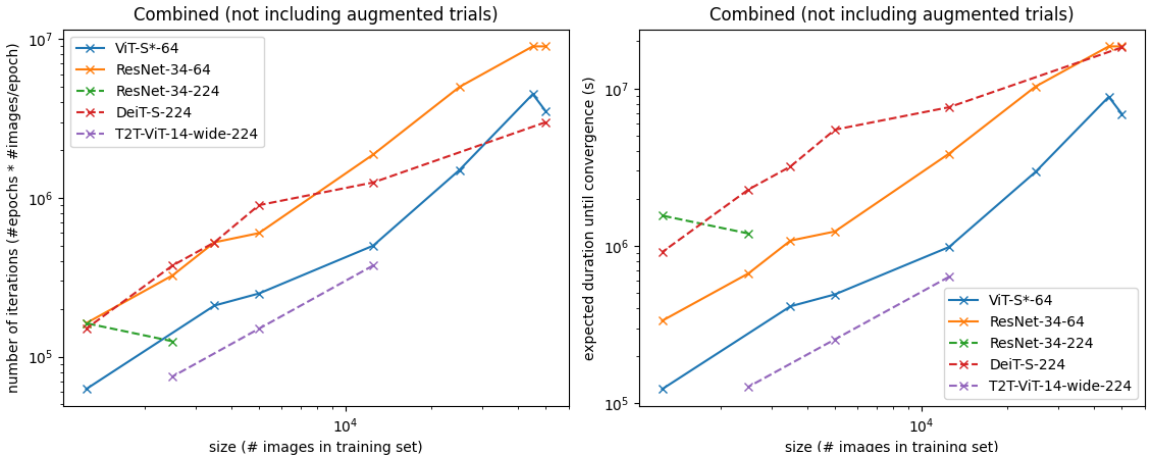

Figure 3. Comparison for models on number of iterations (left) and duration (right) taken to converge when trained with subsets of samples of different set sizes.

From the left figure of Figure 3, we can observe that transformer-based models (ViT-S*, DeiT-S and T2T-ViT-wide) in general converges with smaller number of iterations compared to convolution-based models (ResNet-34), where the number of iterations is defined by

$$\text{number of epochs to convergence} \times \text{number of training samples per epoch}$$

Note that the value obtained actually differ from the actual number of iterations by a factor of the batch size. Yet, since the batch size used is fixed to 32 in our experiments, the relative ratios of the values remains unchanged.

It may then be tempting to conclude that transformer-based models are computationally more efficient compared to convolution-based models on smaller datasets. If we compare between models with the same input sizes, then transformer-based models do take less time to converge. Yet, we cannot conclude that transformer-based models are more computationally efficient as there is still a significant difference between the accuracies of the two classes of models, which make it an unfair comparison.

However, if we instead investigate the relation of size of training subset and the *duration* of training until convergence, shown in the right figure of Figure 3, we can see that DeiT-S with input size 224x224, which have a performance (accuracy) comparable to ResNet-34 with input size 64x64 while taking significantly more time to converge.

Therefore, transformer-based models seemed to remain computationally less efficient compared to convolution-based models over significantly small datasets even when the T2T-ViT model proposed by Li et al. is more computationally efficient compared to other previous transformer-based models [7].

5 Conclusion

As we can see from previous studies, vision transformers achieve better performance for large datasets but are computationally less efficient [8], which led to the question of whether transformer-based models scale and generalize well for small-scale computer vision tasks. In our work, we compared the accuracy and computational efficiencies of convolution-based models like ResNet-34 and transformer-based models like ViT, DeiT and T2T-ViT over small subsets of training dataset of a wide range of sizes. We arrived at the conclusion that transformer models do not generalize very well for small datasets, probably due to the

difficulty in learning positional relations.

While seeing the success of vision transformers applied on large datasets and the difficulties encountered for data collection in speciallized fields, developing models that can generalize well on very small datasets may still be a future direction to work on. However, for most cases, pre-training is still a major and important way to inject extra data into transformer-based models and to improve the performances.

6 Appendix

6.1 Supporting Materials

We have attached the relevant code of our work with this assignment in a zip package in the following format:

report_1155175983.pdf

support1.zip/

plot.py: the plotting code for plotting the graphs used in this report.

data.csv: the manually-collected record of models, test accuracies, etc.

support.zip/

support/

main_code.ipynb: the main code for training the models.

requirements.txt: includes the packages that are needed for the code to run.

commands.txt: if the main code is executed on a local IDE, copy the commands in this document to the terminal of the IDE to install everything. It is advised to install it in a virtual environment to avoid conflicts with other packages.

models/

<id>_<dataset>_<model>_<input_size>/

<epoch>/

model_<timestamp>.h5: the model that can be loaded.

model_<timestamp>.h5_accs.png: accuracy history.

model_<timestamp>.h5_lrs.png: learning rate history.

model_<timestamp>.h5_details.txt: model architecture.

6.2 Cells that Users Can Modify

In *main_code.ipynb*, user can modify the following cells, following that order:

1) The 1st, 3rd and 4th cell immediately below **Main Function.**
   - Change batch size, image size**, patch size (not changed throughout experiment)
   - Change data augmentation (train_transform)
   - Change dataset and “fraction” (proportion of subset)
2) The cell that imports **torchsummary** and those below it before **Some Other Utility Function**.
   - Change the model used and directory of output.
   - Change INITIAL_LR, DECAY, GAMMA, kwargs (arguments for scheduler)

- Change LOAD_PATH (if not None, then the weights *<id>.h5* will be loaded if the model matches with the description.

3) The cell immediately after **The Training**.

- Change NUM_EPOCHS and NUM_EPOCHS_TO_SAVE.

Then, you can watch the results in the cell that follows the cell in (3). The outputs by default is in the path *models/*.

References

[1] İ. B. Akkaya, S. Kathiresan, E. Arani, and B. Zonooz, "Enhancing Performance of Vision Transformers on Small Datasets through Local Inductive Bias Incorporation," *arXiv (Cornell University)*, May 2023, doi: 10.48550/arxiv.2305.08551.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv (Cornell University)*, Dec. 2015, doi: 10.48550/arxiv.1512.03385.

[3] O. Ronneberger, P. Fischer, and T. Brox, "U-NET: Convolutional Networks for Biomedical Image Segmentation," *arXiv (Cornell University)*, May 2015, [Online]. Available: http://arxiv.org/pdf/1505.04597.pdf

[4] A. Dosovitskiy et al., "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," *arXiv (Cornell University)*, Oct. 2020, [Online]. Available: https://arxiv.org/pdf/2010.11929

[5] B. Chen, R. Wang, D. Ming, and X. Feng, "ViT-P: Rethinking Data-efficient Vision Transformers from Locality," *arXiv (Cornell University)*, Mar. 2022, doi: 10.48550/arxiv.2203.02358.

[6] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jeǵou, "Training dataefficient image transformers & distillation through attention," *HAL (Le Centre Pour La Communication Scientifique Directe)*, Jul. 2021, [Online]. Available: https://hal.science/hal-03997937

[7] Y. Li *et al.*, "Tokens-to-Token ViT: Training Vision Transformers from Scratch on ImageNet," *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2021, [Online]. Available: https://arxiv.org/pdf/2101.11986.pdf

[8] S. Khan *et al*., "Transformers in Vision: A Survey," in *ACM Computing Surveys*, vol. 54, Jan. 2022, [Online]. Available: https://arxiv.org/pdf/2101.01169.pdf

[9] C.-F. Chen, Q. Fan, and R. Panda, "CROSSVIT: Cross-Attention Multi-Scale Vision Transformer for Image Classification," *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2021, [Online]. Available: https://arxiv.org/pdf/2106.04560.pdf

[10] S. Bhojanapalli, A. Chakrabarti, D. Gläsner, D. Li, T. Unterthiner, and A. Veit, "Understanding robustness of transformers for image classification," *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Mar. 2021, doi: 10.1109/iccv48922.2021.01007.

[11] A. Krizhevsky, "Learning Multiple Layers of Features from Tiny Images," 2009. [Online]. Available: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf

[12] A. Coates, A. Y. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," *AISTATS*, vol. 15, pp. 215–223, Jun. 2011, [Online]. Available: http://cs.stanford.edu/~acoates/stl10

[13] Lucidrains, "GitHub – lucidrains/vit-pytorch" *GitHub*, Aug. 2023, [Online]. Available: https://github.com/lucidrains/vit-pytorch/blob/main/vit_pytorch/simple_vit.py

[14] Facebookresearch, "GitHub – facebookresearch/deit: Official DeiT repository," *GitHub*, Mar. 2024, [Online]. Available: https://github.com/facebookresearch/deit

[15] Yitu-Opensource, "GitHub - yitu-opensource/T2T-ViT: ICCV2021, Tokens-to-Token ViT: Training Vision Transformers from Scratch on ImageNet," *GitHub*, Mar. 2021, [Online]. https://github.com/yitu-opensource/T2T-ViT