Image recognition using multiple Vision Transformers in parallel having different patch sizes

A. M. Hafiz

Department of Electronics & Communication Engineering, Institute of Technology, University of Kashmir, Srinagar, J&K, 190006, India

mueedhafiz@uok.edu.in

Article Info
Page Number: 1183-1194
Publication Issue:
Vol. 71 No. 4 (2022)

Article History
Article Received: 25 March 2022
Revised: 30 April 2022
Accepted: 15 June 2022
Publication: 19 August 2022

Abstract

With the advent of Transformers which are attention-based mechanisms, many research directions have emerged. Their prowess in natural language processing tasks is well known. Extension of Transformers to computer vision is but natural. Recently, Vision Transforms (ViT's) have achieved very good results on popular image recognition datasets. However, training Transformers is a difficult process due to the need for large computational resources. Parallel processing is a well-known phenomenon present in Nature's most efficient data processors. Inspired by the same, I use a novel technique in which multiple ViT's with different patch sizes are used in parallel. This is followed by averaging the probability vectors of the ViT's for final classification. Using medium-sized ViT's I show that without going for huge scales, state-of-the-art results are achieved on popular datasets.

Keywords: Vision Transformer; ViT; Patch size; Computer Vision; Image recognition.

Introduction

The basic technique for using Transformers involves pretraining on a large dataset [1] and then finetuning on a smaller dataset [2]. Due to the Transformers' computational efficiency and scalability, it is now possible to train them with an unprecedented size e.g. having more than 100B parameters [3, 4]. Even with the ever growing number of models and datasets, no saturation in performance has been noted so far. Transformers [5, 6] have been used in various computer vision applications in the form of Vision Transformers (ViT's) [7,8,5] e.g. in image segmentation [9,10], object recognition [11], object detection [12,13], image generation [14], video understanding [15,16], text-image synthesis [17], super-resolution [18], image based question answering [19,20], etc. [21,22,23,24] and promising results have been achieved. However various issues are faced like the need for large computation resources, lower performance and excessive training. As such in computer vision, convolution based architectures have dominance [25,26,27]. Pertinently, many works have tried to combine CNN models with self attention [28,12] out of which some have been able to replace convolution holistically [29,30]. The ViT's although being efficient are not yet hardware-accelerator friendly for the reason that they use specialized attention functions. Although for large-scale computer vision tasks, traditional architectures like ResNet have been efficient [31,32,33], the state-of-the-art has improved using ViT's [8].

A recent success of ViT's in improving the state-of-the-art is shown in [8], wherein they have been used directly for image recognition with minimal modification. The authors of [8] split an image into patches after which they provide a sequence of patch embeddings as input to their ViT. Here, the image patches are used in a manner similar to tokens in NLP training with supervision. After training on datasets like ImageNet modest performance is obtained which is a little lesser than that of ResNets with similar size. The issue has been overcome by training their ViT's on larger datasets (14M-300M images) which leads to much better performance. In spite of this, works suggesting performance improvement using ViT's are rare. Taking a hint from parallel processing which is found in Nature's most efficient data-processors, I seek to augment the performance of ViT's by using them in parallel. In my work, I use multiple ViT's in parallel each having a different patch size. Prediction probabilities of the respective ViT's are averaged across the ensemble for final classification. To the best of my knowledge this is the first work to use such a parallel processing scheme in ViT based image recognition. Using an ensemble of ViT's pre-trained on the ImageNet-21k, or the JFT-300M datasets, my approach advances the state-of-the-art on multiple image recognition benchmarks. Particularly, my best ensemble reaches the accuracy of 87.92% on ImageNet, 90.74% on ImageNet-ReaL, 99.54% on CIFAR-10, 94.58% on CIFAR-100, 97.61% on Oxford-IIIT Pets, and 99.76% on Oxford Flowers-102 datasets achieving first rank on five out of these six datasets, and second rank on the remaining ImageNet dataset.

The rest of the paper is structured as follows. In Section 2 I give the background of the work. This is followed by Section 3 which discusses work related to my paper. Section 4 discusses the proposed approach and Section 5 discusses the experiments and their results. I conclude in Section 6.

Background

In this section I discuss *attention* which is the background of the architecture of Transformers.

A. Self attention

For a vector, the self attention gives the estimate of the inter-relevance of the components of a vector, e.g. word relevance in a sentence. Global information combination is used. Self attention is a fundamental unit of transformers which are attention based models. Let $\mathbf{X} \in \mathrm{R}^{n \times d}$ be a vector of *n* elements ($\mathbf{x}_1$, $\mathbf{x}_2$, ... $\mathbf{x}_n$) where *d* is the embedding dimension. Self attention captures the inter-dependency of the *n* elements in a global context using an encoder. For this I define three weight matrices viz.

Query($\mathbf{W}^Q \in \mathrm{R}^{n \times dq}$), Key ($\mathbf{W}^K \in \mathrm{R}^{n \times dk}$), and Value ($\mathbf{W}^V \in \mathrm{R}^{n \times dv}$).

Next, $\mathbf{X}$ is spread out over these matrices for obtaining $\mathbf{Q} = \mathbf{X}\mathbf{W}^Q$, $\mathbf{K} = \mathbf{X}\mathbf{W}^K$ and $\mathbf{V} = \mathbf{X}\mathbf{W}^V$. The outcome of this process $\mathbf{Z} \in \mathrm{R}^{n \times dv}$ given by the self attention layer is:

$$\boldsymbol{Z} = softmax\left(\frac{\boldsymbol{QK^T}}{\sqrt{d_q}}\right)V \qquad (1)$$

For every vector-component, self attention computes the dot product of the query and all the keys. This product is then normalized by using a softmax for inferring the attention scores. Each vector-component thus transforms into a weighted sum where the weights are the attention-map scores.

B. Masked self attention

Self attention attends to each vector-component. If the transformer [6] has to predict the next vector-component, the decoder self attention units are masked to prevent them from processing future components. This is done by multiplying the vector- components with a mask $\mathbf{M} \in \mathbf{R}^{n\times n}$, $\mathbf{M}$ being the upper triangular matrix as:

$$softmax\left(\frac{\mathbf{QK^T}}{\sqrt{d_q}} \circ \mathbf{M}\right) \qquad (2)$$

Here ° is the Hadamard product. During vector-component prediction, the future attention-map scores are nulled by this technique.

C. Multihead attention

For the derivation of intricate dependencies between vector-components, a multihead attention technique is used which comprises of several self attention units or heads. The number of heads is denoted by *h*. In the original transformer model [11], eight attention heads were used i.e. *h* was 8. Every attention head has its weight matrices $\{\mathbf{W}^{Qi}, \mathbf{W}^{Ki}, \mathbf{W}^{Vi}\}$, where $i = 0, 1, 2,\ldots, (h - 1)$. For an input $\mathbf{X}$, the outputs of *h* self attention units are combined into one multihead weight matrix $[\mathbf{Z}_0, \mathbf{Z}_1, \ldots, \mathbf{Z}_{h-1}] \in \mathbf{R}^{n\times h\times dv}$. These weights are then projected to a separate weight matrix $\mathbf{W} \in \mathbf{R}^{h.dv\times v}$.

The main difference between self attention and convolution is that in the former each weight is constantly computed, where as in the later fixed weights are used which are obtained by training. Also the self attention technique is both permutation invariant as well as input-size invariant, making it suitable for irregularity as compared to convolution. Figure 1 shows a multihead attention unit which is made up of several self attention units.

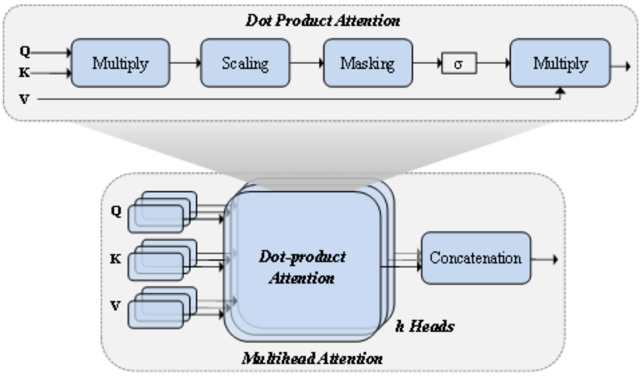

Fig. 1: Illustration of the attention mechanism used in Transformers [5]

Related Work

Transformers were introduced by [6] for machine translation tasks, and have since become state-of-the-art status for many NLP applications. Transformers are pre-trained on large datasets and finetuned for specific applications. This is done in BERT [2] which uses denoising based self-supervised pretraining. The GPT version of BERT uses language modelling pretraining [34,35,3]. A basic application of self attention for images requires that every pixel covers all other pixels. However due to the quadratic variation of computation cost with number of pixels, this scaling is not realistic. Hence an approximation is used. The work of [36] applies self attention locally for every pixel query instead of doing it globally. Local multihead attention techniques like these can completely replace convolution [37,29,30]. Sparse Transformers [38] use scaled approximation for global self attention. Attention can also be scaled by using different block sizes [39], or in the extreme only on individual axes [40,41]. Although several of these special attention-based models show good results for computer vision applications, however they require complicated engineering for efficient hardware acceleration. The ViT of [8] is adopted by me for my ensemble approach. I extract patches from an image and apply complete self attention to them. I improve the performance of the ViT by using it in an ensemble.

There has been significant interest in combination of CNNs with self-attention. This process augments the feature-maps used in image classification [42]. The same can also be achieved by subsequently processing the CNN output by self attention, as has been done in applications like object detection [43,12], video processing [28,15], image classification [28], object discovery using unsupervised learning [44], or combined text & vision applications [45,46,47].

Image GPT (iGPT) [48] uses pixel-based Transformers having resolution as well as color-space reduction. The architecture is trained in unsupervised mode followed by finetuning. It achieves a classification accuracy of 72% on ImageNet. The work of [8] takes this performance further to 88.55%. The authors of [8] achieve this feat by augmenting the ViT training data and achieve state-of-the-art results on various benchmarks. They focus on ImageNet-21k and JFT-300M datasets while using Transformers instead of ResNets. I also use the ViT's of [8] and enhance the state-of-the-art on these datasets and others by using a unique parallel approach. To the best of my knowledge this is the first work in this regard.

In the next section, I discuss the proposed approach.

Proposed Approach

The proposed approach is based on the concept of parallel processing. The parallel technique is efficient in processing volumes of data by distributing the decision making among an ensemble [49,50,51] of data-processors. Here I use this concept for Transformers. Multiple Transformers with different image patch sizes give their classification probabilities. The Transformers are trained on a large dataset and fine-tuned on smaller datasets as per [8]. Three patch sizes are used, viz. 16×16 (as in [8]), 14×14 and 18×18. Let E denote the number of ViT's used in parallel. A minimum of 2 and maximum of 3 ViT's are used. Each has a different patch size. Next the same ViT's are finetuned on the smaller datasets. The classification probability vectors for the E ViT's, are given by $\{S_1, S_2, ..., S_E\}$, where Si $=\{S_{i1}, S_{i2},..., S_{iK}\}$ with K being the number of classes. The classification probabilities are averaged over E to give final classification probability vector $\mathbf{S}_f$, as given in Eqn. (3):

$$S_f = \frac{\mathrm{S}_1 \oplus \mathrm{S}_2 \oplus ... \oplus \mathrm{S}_\mathrm{E}}{\mathrm{E}} \qquad (3)$$

where E = Number of Transformers used in parallel, and $\oplus$ denotes the element-wise addition operation.

Finally the class c of the image is decided as per the maximum in $\mathbf{S}_f$ as given in Eqn. (4) as:

$$c = \mathrm{argmax}(\mathbf{S}_f) \qquad (4)$$

Figure 2 shows the overview of the proposed technique.

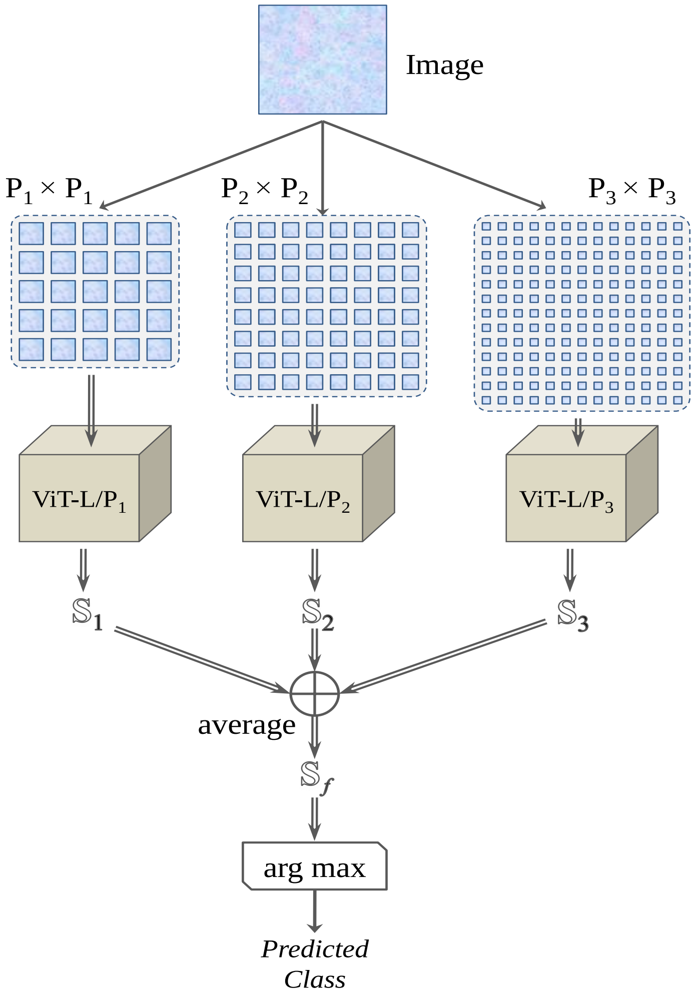

Fig. 2: Illustration of the proposed approach using an ensemble of 3 ViT's with patch sizes of (P_1, P_1), (P_2 , P_2) and (P_3, P_3) respectively. ***S*** denotes the classification probability vector.

The basic input to a Transformer is a token sequence. For handling 2D images, the image $x \in \mathbf{R}^{H\times W\times C}$ is flattened into a sequence of 2D patches $x_p \in \mathbf{R}^{\mathrm{N}\cdot(\mathrm{p}^2\mathrm{C})}$ wherein *(H, W)* is the image resolution, *C* being the number of channels, *(P, P)* being the image patch resolution, and *N* = *HW/P*^{2} being the number of resulting patches. TheTransformer uses a constant vector of size *D* in its layers to flatten the patches and map them to *D* dimensions for linear projection training (Eqn. (5)).

$$\mathbf{z_0} = \left[x_{class}; x_p^1\mathbf{E}; x_p^2\mathbf{E}; \ldots; x_p^N\mathbf{E}\right] + \mathbf{E_{pos}} \qquad \mathbf{E} \in \mathbf{R}^{\mathrm{N}\cdot(\mathrm{p}^2\mathrm{C})},\ \mathbf{E_{pos}} \in \mathbf{R}^{(\mathrm{N}+1)\times\mathrm{D}} \qquad (5)$$

Table 1: Details of the ViT used as on lines of [8]

Model	**Layers**	**Hidden size *D***	**MLP size**	**Heads**	**Params**
ViT-Large	24	1024	4096	16	307M

Like BERT’s [*class*] token, a learnable embedding is prepended to the embedded patches ($\mathbf{z}_0^0 = x_{class}$) wherein the output state of the ViT encoder **z₀** is the representation of the image *y* (Eqn. (8)).

The encoder of the Transformer [6] has alternate layers of multihead self attention (MSA), and multi layer perceptron (MLP) layers. The layernorm (LN) function is used before each block, and residual connections are used after each block (Eqns.(6),(7)). The MLP has 2 layers of the GELU non-linear function.

$$\mathbf{z}_l' = MSA\left(LN(z_{l-1})\right) + \mathbf{z}_{l-1}, \qquad l = 1\ldots\mathrm{L} \qquad (6)$$

$$\mathbf{z}_l = MLP\left(LN(\mathbf{z}_l)\right) + \mathbf{z}_l', \qquad l = 1\ldots\mathrm{L} \qquad (7)$$

$$y = LN(\mathbf{z}_L^0) \qquad (8)$$

The ViT’s are pretrained on a large dataset and finetuned on smaller task-specific datasets.

Experimentation

A. Datasets

The proposed approach uses the ILSVRC-2021 ImageNet dataset having 1k classes and 1.3M images, the superset of the same viz. ImageNet-21k with 21k classes and 14M images [52], and JFT [53] with 18k classes and 303M images. This is done on lines of [8]. The models are trained, finetuned as well as evaluated on ReaL labels [54], CIFAR-10/100 [55], Oxford-IIIT Pets [56], and Oxford Flowers-102 [57] on the lines of [8].

B. Model details

The ViT configuration of [8] is used as per BERT [2]. The “Large” model is used as per [8]. Their notation is used for the models e.g. ViT-L/16 means that the “Large” ViT variant is used with 16×16 patch size. On similar lines ViT-L/(14,16,18) means an ensemble of 3 “Large” variant ViT’s is used having 14×14, 16×16, and 18×18 patch sizes respectively. The details of the Transformers used are as per [8]. They follow the original Transformer [11] and are given in Table 1. I compare the performance of my approach with that given in [8].

Table 2: Training hyperparameters used as per [8].The models are trained with batch-size = 4096 and a learning-rate warmup of 10k steps.

Models	**Dataset**	**Epochs**	**Base LR**	**LR decay**	**Weight decay**	**Dropout**
ViT-L/*	JFT-300M	7	$4 \cdot 10^{-4}$	Linear	0.1	0.0
ViT-L/*	ImageNet-21k	30	10^{-3}	Linear	0.03	0.1
ViT-L/*	ImageNet	300	$3 \cdot 10^{-3}$	Cosine	0.3	0.1

Table 3: Hyperparameters used for fine-tuning the ViT’s as per [8]. All models have been finetuned using cosine learning rate (LR) decay, batch size = 512, no weight decay, and grad clipping with global norm = 1.

Datasets	**Steps**	**Base LR**
ImageNet	20000	{0.003, 0.01, 0.03, 0.06}
CIFAR-10	10000	{0.001, 0.003, 0.01, 0.03}
CIFAR-100	10000	{0.001, 0.003, 0.01, 0.03}
Oxford-IIIT Pets	500	{0.001, 0.003, 0.01, 0.03}
Oxford Flowers-102	500	{0.001, 0.003, 0.01, 0.03}

C. Training and finetuning

My ViT’s are trained as per [8]. I use Adam [58] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, batch size = 4096, weight decay = 0.1. Finetuning is done using SGD with momentum, and batch-size = 512. Figure 3 shows the accuracy plots for finetuning of the models having different patch sizes on

CIFAR-100.The hyperparameters used for the training of the ViT's are as per [8] and are given in Table 2. The hyperparameters used for finetuning the ViT's are as per [8] and are given in Table 3.

D. Results

My ViT's are trained on TPUv3 accelerators. One TPUv3-core-day ($\mathbf{t}_{Tcd}$) corresponds to the number of TPUv3 cores (2 per chip) used during training multiplied by training duration expressed in days. It should be noted that for the same patch size, $\mathbf{t}_{Tcd}$ for the same models is almost similar, whereas it differs if the patch size is varied. For smaller patch size (14 × 14) i.e. more tokens, the $\mathbf{t}_{Tcd}$ is slightly largerwhereas for larger patch size (18 × 18), $\mathbf{t}_{Tcd}$ is slightly lesser.

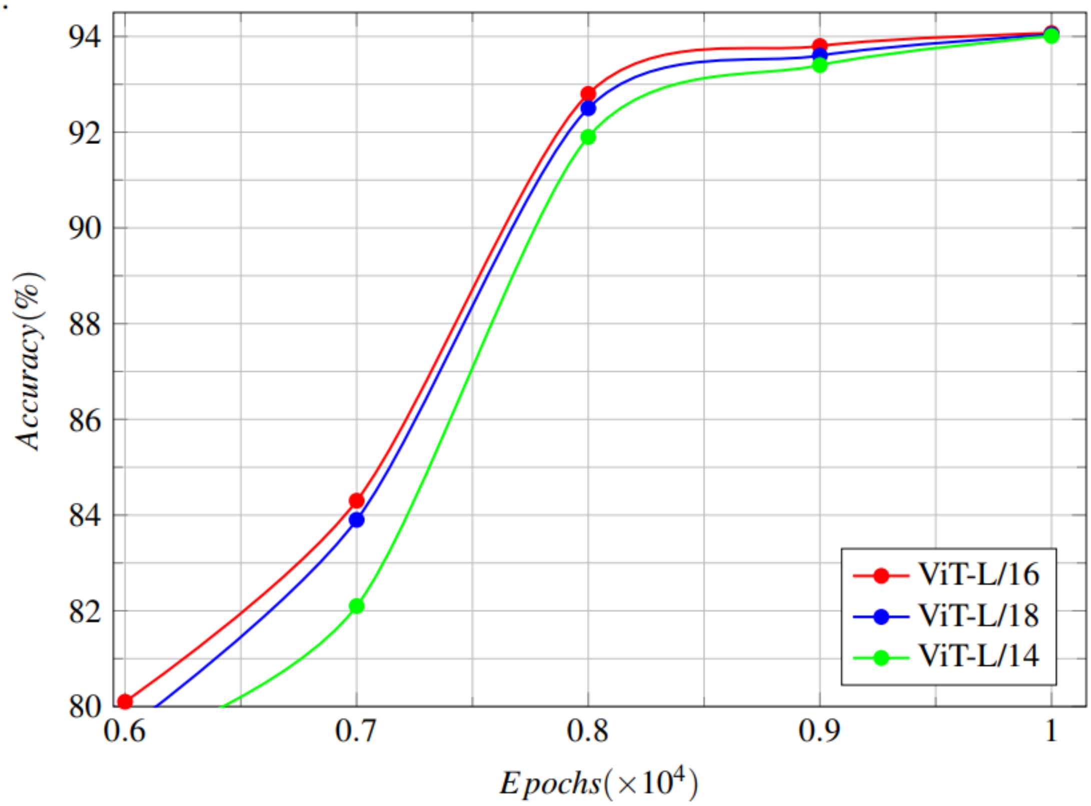

Fig. 3: Classification accuracy for finetuning on CIFAR-100 dataset for ViT-L/14, ViT-L/16 and ViT-L/18. The models are pre-trained on JFT

The performance of different ensembles on CIFAR-10, CIFAR-100, and Oxford Flowers-102, using my finetuned ViT ensembles is shown in Table 4. The performance of the ViT-L/16 of [8] is also shown. It should be noted that the best performance is obtained for my ensemble approach using 3 ViT's with respective patch sizes of 14, 16 and 18.

Table 4: Comparison of performance of variants of my ensembles against ViT-L/16 of [8], for CIFAR-10, CIFAR-100 and Oxford Flowers-102 datasets. I use models pre-trained on JFT and finetuned on the datasets given in the table. Mean and std. deviation of accuracies are reported, after taking the average over 3 finetuning runs.

Dataset	ViT-L/16 [8]	ViT-L/(16,18) Proposed	ViT-L(14,16) Proposed	ViT-L(14,18) Proposed	ViT-L(14,16,18) Proposed
CIFAR-10	99.42±0.03	99.51±0.01	99.49±0.04	99.43±0.02	**99.54**±0.05
CIFAR-100	93.90±0.05	94.41±0.05	94.32±0.06	94.23±0.04	**94.58**±0.03
Oxford-IIIT Pets	97.32±0.11	97.52±0.02	97.47±0.01	97.39±0.03	**97.61**±0.09

The performance of my ViT's is compared with that of all models mentioned in [8] in Table 5. My JFT-300M pre-trained ViT-L/(14,16,18) ensemble which uses 3 variants of ViT-L having patch sizes of 14, 16, and 18 respectively, outperforms all other state-of-the-art models on ImageNet ReaL, CIFAR-10/100, Oxford-IIIT Pets and Oxford Flowers-102. For the remaining ImageNet dataset although my proposed approach achieves 2nd rank, I am sure I would have has a better score than ViT-H/14 of [8] had I experimented with a larger ViT ensemble or with a ViT-H/(14,16,18) ensemble.

Table 5: Comparison with the state-of-the-art on notable image classification datasets. Mean and std. deviation of accuracies are reported, after taking the average over 3 finetuning runs.

Dataset	JFT ViT-L/ (14,16,18) Proposed	JFT ViT-H/14 [8]	JFT ViT-L/16 [8]	121k ViT-L/16 [8]	BiT-L ResNet152x4 [33]	Noisy Student EfficientNet-L2 [32]
ImageNet	87.92±0.02	**88.55**±0.04	87.76±0.03	85.30±0.02	87.54±0.02	88.5
ImageNet ReaL	**90.74**±0.01	90.72±0.05	90.54±0.03	88.62±0.05	90.54	90.55

CIFAR-10	**99.54**±0.05	99.50±0.06	99.42±0.03	99.15±0.03	99.37±0.06	-
CIFAR-100	**94.58**±0.03	94.55±0.04	93.90±0.05	93.25±0.05	93.51±0.08	-
Oxford-IIIT Pets	**97.61**±0.09	97.56±0.03	97.32±0.11	94.67±0.15	96.62±0.23	-
Oxford Flowers-102	**99.76**±0.01	99.68±0.02	99.74±0.00	99.61±0.02	99.63±0.03	-

Conclusion

In this paper, the efficacy of using Vision Transformers (ViT's) for image recognition tasks was demonstrated by using a novel parallel processing scheme. An overview of the attention mechanism used in Transformers was given. This was followed by a discussion of related works. Next, I introduced my approach wherein I proposed the use of multiple ViT's in parallel with different patch sizes. In particular patch sizes of (14×14), (16×16), and (18×18) were used successfully. The next step in the proposed approach involved averaging the classification probability vectors of the ViT's. I showed experimentally that using such a scheme led to state-of-the-art results on popular datasets. However, larger ViT architectures were not investigated which could reveal more information. Also, using more ViT's (above 3) with different patch sizes is a task I intend to take up in future work. One interesting research direction in this regard would be using a single ViT having internal parallel processing for multiple patch sizes.

Conflict of interest

The authors declare no conflict of interest.

Declaration of funding

This project has not received any type of funding.

References

1. J. Beal, H.-Y. Wu, D. H. Park, A. Zhai, and D. Kislyuk, "Billion-scale pretraining with vision trans- formers for multi-task visual representations," Proc. IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Jan. 2022, pp. 564–573.
2. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," Proc. 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human

Language Technologies, Volume 1 (Long and Short Papers). Minneapolis, Minnesota: Association for Computational Linguistics, 2019, pp. 4171–4186. [Online]. Available: https://www.aclweb.org/anthology/N19-1423

3. T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," 2020. [Online]. Available: https://arxiv.org/abs/2005.14165
4. D. Lepikhin, H. Lee, Y. Xu, D. Chen, O. Firat, Y. Huang, M. Krikun, N. Shazeer, and Z. Chen, "Gshard: Scaling giant models with conditional computation and automatic sharding," Proc. International Conference on Learning Representations, 2021. [Online]. Available: https://openreview.net/forum?id=qrwe7XHTmYb
5. S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in vision: A survey," 2021.
6. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, u. Kaiser, and I. Polosukhin, "Attention is all you need," Proc. 31st International Conference on Neural Information Processing Systems, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, pp. 6000–6010.
7. Y. Xu, H. Wei, M. Lin, Y. Deng, K. Sheng, M. Zhang, F. Tang, W. Dong, F. Huang, and C. Xu, "Transformers in computational visual media: A survey," Computational Visual Media, vol. 8, no. 1, pp. 33–62, 2022.
8. A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," Proc. International Conference on Learning Representations, 2021. [Online]. Available: https://openreview.net/forum?id=YicbFdNTTy
9. A. Hatamizadeh, Y. Tang, V. Nath, D. Yang, A. Myronenko, B. Landman, H. R. Roth, and D. Xu, "Unetr: Transformers for 3d medical image segmentation," Proc. IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), January 2022, pp. 574–584.
10. L. Ye, M. Rochan, Z. Liu, and Y. Wang, "Cross-modal self-attention network for referring image segmentation," Proc. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 10 494–10 503.
11. H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jegou, "Training data-efficient image transformers & distillation through attention," 2021.
12. N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," Proc. Computer Vision - ECCV 2020, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer International Publishing, 2020, pp. 213–229.
13. X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: Deformable transformers for end-to-end object detection," 2021.
14. H. Chen, Y. Wang, T. Guo, C. Xu, Y. Deng, Z. Liu, S. Ma, C. Xu, C. Xu, and W. Gao,

"Pre-trained image processing transformer," 2020.

15. C. Sun, A. Myers, C. Vondrick, K. Murphy, and C. Schmid, "Videobert: A joint model for video and language representation learning," Proc. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 7463–7472.
16. R. Girdhar, J. Joao Carreira, C. Doersch, and A. Zisserman, "Video action transformer network," Proc. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 244-253.
17. A. Ramesh, M. Pavlov, G. Goh, S. Gray, M. Chen, R. Child, V. Misra, P. Mishkin, G. Krueger, S. Agar- wal et al., "Dall-e: Creating images from text," OpenAI blog. https://openai. com/blog/dall-e, 2021.
18. F. Yang, H. Yang, J. Fu, H. Lu, and B. Guo, "Learning texture transformer network for image super- resolution," Proc. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 5790–5799.
19. 19. H. Tan and M. Bansal, "LXMERT: Learning cross-modality encoder representations from transformers," Proc. 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP- IJCNLP). Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 5100–5111. [Online]. Available: https://www.aclweb.org/anthology/D19-1514
20. 20. W. Su, X. Zhu, Y. Cao, B. Li, L. Lu, F. Wei, and J. Dai, "Vl-bert: Pre-training of generic visual-linguistic representations," Proc. International Conference on Learning Representations, 2020. [Online]. Available: https://openreview.net/forum? id=SygXPaEYvH
21. X. Wang, C. Yeshwanth, and M. Niebner, "Sceneformer: Indoor scene generation with transformers, 2021.
22. M. Kumar, D. Weissenborn, and N. Kalchbrenner, "Colorization transformer," 2021.
23. C. Doersch, A. Gupta, and A. Zisserman, "Crosstransformers: spatially-aware few-shot transfer, 2021.
24. H.-J. Ye, H. Hu, D.-C. Zhan, and F. Sha, "Few-shot learning via embedding adaptation with set-to-set functions," Proc. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 8805–8814.
25. Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.
26. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," Proc. 25th International Conference on Neural Information Processing Systems - Volume 1, ser. NIPS'12. Red Hook, NY, USA: Curran Associates Inc., 2012, pp. 1097–1105.
27. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," Proc. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Los Alamitos, CA, USA:IEEE Computer Society, jun 2016, pp. 770–778. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/CVPR.2016.90
28. 28. X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," Proc.

IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018.

29. P. Ramachandran, N. Parmar, A. Vaswani, I. Bello, A. Levskaya, and J. Shlens, "Stand-alone self-attention in vision models," Proc. Advances in Neural Information Processing Systems, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alche-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper/2019/file/3416a75f4cea9109507cacd8e2f2aefc-Paper.pdf
30. H. Zhao, J. Jia, and V. Koltun, "Exploring self-attention for image recognition," Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2020.
31. D. Mahajan, R. Girshick, V. Ramanathan, K. He, M. Paluri, Y. Li, A. Bharambe, and L. van der Maaten, "Exploring the limits of weakly supervised pretraining," Proc. European Conference on Computer Vision (ECCV), September 2018.
32. Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le, "Self-training with noisy student improves imagenet classification," Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2020.
33. A. Kolesnikov, L. Beyer, X. Zhai, J. Puigcerver, J. Yung, S. Gelly, and N. Houlsby, "Big transfer (bit): General visual representation learning," Proc. Computer Vision - ECCV 2020, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer International Publishing, 2020, pp. 491–507.
34. A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018.
35. A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language Models are Unsupervised Multitask Learners," 2019. [Online]. Available: https://openai.com/blog/better- language-models/
36. N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, N. Shazeer, A. Ku, and D. Tran, "Image transformer," Proc. 35th International Conference on Machine Learning, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. PMLR, 10-15 Jul 2018, pp. 4055–4064. [Online]. Available: https://proceedings.mlr.press /v80/parmar18a.html
37. H. Hu, Z. Zhang, Z. Xie, and S. Lin, "Local relation networks for image recognition," Proc. IEEE/CVF International Conference on Computer Vision (ICCV), October 2019.
38. R. Child, S. Gray, A. Radford, and I. Sutskever, "Generating long sequences with sparse transformers," 2019. [Online]. Available: http://arxiv.org/abs/1904.10509
39. D. Weissenborn, O. Tackstrom, and J. Uszkoreit, "Scaling autoregressive video models," in International Conference on Learning Representations, 2020. [Online]. Available: https://openreview.net/forum?id=rJgsskrFwH
40. J. Ho, N. Kalchbrenner, D. Weissenborn, and T. Salimans, "Axial attention in multidimensional transformers," 2020. [Online]. Available: https://openreview.net/forum? id= H1e 5GJBtDr
41. H. Wang, Y. Zhu, B. Green, H. Adam, A. Yuille, and L.-C. Chen, "Axial-deeplab: Stand-alone axial-attention for panoptic segmentation," Proc. Computer Vision - ECCV 2020, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer

International Publishing, 2020, pp. 108–126.

42. I. Bello, B. Zoph, A. Vaswani, J. Shlens, and Q. V. Le, “Attention augmented convolutional networks,” Proc. IEEE/CVF International Conference on Computer Vision (ICCV), October 2019.
43. H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei, “Relation networks for object detection,” Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018.
44. F. Locatello, D. Weissenborn, T. Unterthiner, A. Mahendran, G. Heigold, J. Uszkoreit, A. Dosovitskiy, and T. Kipf, “Object-centric learning with slot attention,” Proc. NeurIPS 2020, 2020. [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/ 8511df98c02ab60aea1b2356c0 13bc0f-Abstract.html
45. B. Wu, C. Xu, X. Dai, A. Wan, P. Zhang, M. Tomizuka, K. Keutzer, and P. Vajda, “Visual transformers: Token-based image representation and processing for computer vision,” 2020. [Online]. Available: https://arxiv.org/abs/2006.03677
46. J. Lu, D. Batra, D. Parikh, and S. Lee, “Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks,” 2019.
47. L. H. Li, M. Zaskar, D. Yin, C. Hsieh, and K. Chang, “Visualbert: A simple and performant baseline for vision and language,” 2019. [Online]. Available: http://arxiv.org/abs/1908.03557
48. M. Chen, A. Radford, R. Child, J. Wu, H. Jun, D. Luan, and I. Sutskever, “Generative pretraining from pixels,” Proc. 37th International Conference on Machine Learning, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13-18 Jul 2020, pp.1691–1703. [Online]. Available: https://proceedings.mlr.press/v119/chen20s.html
49. A. M. Hafiz, R. Bhat, and M. Hassaballah, “Image classification using convolutional neural network tree ensembles,” Multimedia Tools and Applications, pp. 1–18, 2022.
50. A. M. Hafiz and G. M. Bhat, “Fast training of deep networks with one-class CNNs,” Cham, pp. 409–421, 2021. Available: https://doi.org/10.1007/978-3-030-68291-0
51. A. M. Hafiz and M. Hassaballah, “Digit image recognition using an ensemble of one-versus-all deep network classifiers,” Proc. Information and Communication Technology for Competitive Strategies (ICTCS 2020), M. S. Kaiser, J. Xie, and V. S. Rathore, Eds. Singapore: Springer Singapore, 2021, pp. 445–455.
52. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” Proc. 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255.
53. C. Sun, A. Shrivastava, S. Singh, and A. Gupta, “Revisiting unreasonable effectiveness of data in deep learning era,” Proc. IEEE International Conference on Computer Vision (ICCV), Oct 2017.
54. L. Beyer, O. J. Henaff, A. Kolesnikov, X. Zhai, and A. van den Oord, “Are we done with imagenet?” 2020.
55. A. Krizhevsky, “Learning multiple layers of features from tiny images,” Master’s thesis, University of Tront, 2009.
56. O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. V. Jawahar, “Cats and dogs,” Proc. 2012

IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 3498–3505.

57. M.-E. Nilsback and A. Zisserman, “Automated flower classification over a large number of classes,” Proc. 2008 Sixth Indian Conference on Computer Vision, Graphics Image Processing, 2008, pp. 722–729.
58. D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” Proc. ICLR, 2015. [Online]. Available: http://arxiv.org/abs/1412.6980