An Image Is Worth 393 Areas: Training image Transformers with Area-Attention

Osher Tidhar, 200944940
Tel-Aviv University
tidharosher@mail.tau.ac.il

Yoav Kurtz, 304820582
Tel-Aviv University
yoavkurtz@mail.tau.ac.il

Abstract - *recently, attention only neural network models have shown promising results for image related tasks. While being the standard for natural language processing, the Transformers successfully were shifted to the realm of computer vision. In this work, we investigate the effects of modifying the vision Transformer architecture by replacing the self-attention blocks with the novel area-attention mechanism. We test and compare several area-attention configurations and show that by utilizing data-efficient training methods the area-attention Transformer can achieve superior results, compared to the original model, without adding a single parameter.*

1 Introduction

Attentional mechanisms [2], have significantly boosted the accuracy on a variety of deep learning tasks. They allow the model to selectively focus on specific pieces of information. This can be a word in a sentence for neural machine translation or a region of pixels in image related tasks such as captioning [19].

Area-attention (AA), Li et al. [12], is a general attention mechanism enabling the model to attend a group of items in the memory that are structurally adjacent. According to this method, each unit for attention calculation is an area that can contain one or more items. Previously only considered for NLP related tasks, experiments have indicated that AA outperforms regular attention on both machine translation and image captioning. The writers of [12] tested the effects of AA on LSTM [14] and Transformers [25] (encoder-decoder) architectures.

Self-attention-based models, such as the Transformers, have revolutionized natural language processing due to their computational efficiency and scalability. Motivated by this success, researchers have proposed a hybrid architecture transplanting Transformer ingredients to convnets to perform vision tasks [3]. The vision Transformer (ViT) introduced by Dosovitskiy et al. [7] proposes an architecture that is directly based on the original Transformer. Unlike the image captioning frame-work proposed in [19], the ViT uses raw image patches as input to perform classification.

Although showing state-of-the-art results in image classification benchmarks, the model presented in the paper was trained on a very large private dataset (JFT-300M [20], 300 millions images). When trained on public datasets such as ImageNet [6], using the same training scheme, ViT does not generalize as well.

To this end, Touvron et al. [24] presented a data-efficient training scheme. Using improvements included in the torch image models (timm) library [26], they reported large improvements over previous results. More specifically, the writers trained a ViT that is competitive with convnets having a similar number of parameters and efficiency. Their method, which depends on strong data augmentations, made it possible to train vision Transformer models without the need for huge datasets. Consequently this enables researchers, without access to extremely strong hardware, to achieve plausible results.

Motivated by these latest achievements, we attempt to further improve the ViT performance when training it from scratch on a publicly available dataset. As seen in [7], a major advantage of the self-attention mechanism over CNNs (LeCun et al. [11]) is the fact that even the earliest layers of the Transformer are able to get "global" information. Whereas CNN models incorporate global information only at later layers, as the receptive field increases. In the ViT case this translates to the ability to "give" attention to every patch of the image, at any of the layers. Under the assumption that the ViT's strength comes from the ability to attend multiple areas in the memory, we chose to improve the model architecture by strengthening its attention mechanism. This is done by replacing some of the self-attention layers with the innovative area-attention introduced by Li et al. [12].

We modify the ViT architecture by replacing the first multi-head self-attention layers with multi-head area-attention layers. Each item processed in the AA layer is comprised of several embedding vectors that are spatially adjacent. Due to the way the areas are calculated, this improvement does not add any parameters to the model and its effect on runtime is minimal.

In this work we use the architectures presented by Touvron et al. [24] as a baseline, and compare their model's classification accuracy to our AA ViT models. We follow their data-efficient training scheme. Several AA configurations are tested and it is shown that on the dataset used, simply by only modifying the first Transformer layers, our model present superior results.

2 Related Work

Transformers in Vision: Transformers [25], which rely on the self-attention mechanism, were proposed for machine translation. This method has achieved state-of-the-art results in many NLP tasks, and therefeore has rapidly become the model of choice for NLP problems, replacing older recurrent neural network models such as the long short-term memory (LSTM).

Following their success in NLP, using Transformers for vision tasks became a new research direction that was explored heavily recently; Vision Transformers (ViT) [7], proposed for image classification, is no doubt, the most successful application of Transformers for Computer Vision.

Besides image classification, the application of Transformers in vision was explored in other tasks:

- object detection: Facebook's Detection Transformers (DETR) [3], was the first object detection framework that successfully used Transformer as the main building blocks in the pipeline. It uses a hybrid model architecture, which combines the convolutional neural network (CNNs) with Transformer.

  other object detection works: [5], [21], [31].

- video processing: [33], [30].

- image generation: [16].

Other Attention Mechanisms: The first attempt to go beyond the standart attention mechanism was changing the use of softmax as the attention activation function. Instead, sigmoid has been used to allow multiple items to be attended [22], [18]. However, sigmoid activation alone does not enforce the constraint for attended items to be structurally adjacent. Previous works have proposed several methods for capturing structures in attention calculation. For example, Kim et al. [9] used a conditional random field to directly model the dependency between items, which allows multiple "cliques" of items to be attended to at the same time. Niculae et al. [15] approached the problem, from a different angle, by using regularizers to encourage attention to be placed onto contiguous segments. In image captioning tasks, previous works showed that it is beneficial to attend to semantic regions or concepts on an image ([17], [1], [13], [27]).

Compared to these works, area attention mechanism does not require to train a special network or use an additional loss to capture structures, and can be entirely parameter free. It allows a model to attend to information at a varying granularity, which can be at the input layer or in the latent space.

Challenges in ViT: ViT performs very well on large-scale datasets such as Imagenet-1k, ImageNet-21k and JFT-300M [20]. However, it achieves inferior performance compared with CNNs when trained from scratch on a midsize dataset (e.g., ImageNet).

To this end, Several works have been tried to address this problem by slightly modify the ViT architecture:

- Li Yuan et al. [28] claimed that the simple tokenization of input images in ViT fails to model the important local structure (e.g., edges, lines) among neighboring pixels and proposed to overcome it mainly by recursively aggregating neighboring Tokens into one Token (Tokens-to-Token), such that local structure presented by surrounding tokens can be modeled.

- In the second part of [24], the writers apply a Knowledge Distillation [29],[8] to improve the original ViT by adding a distillation-token along with the class-token, such that on the output of the network its objective is to reproduce the label predicted by the teacher, instead of true label.

These 2 works, in a similar way to ours, focus on the architecture design. However, differently from all other works, we attempt to go one step further in ViT's approach to attend multiple areas in the memory, and suggest to improve its results by strengthening its self-attention mechanism.

3 Methods

3.1 Vision Transformer

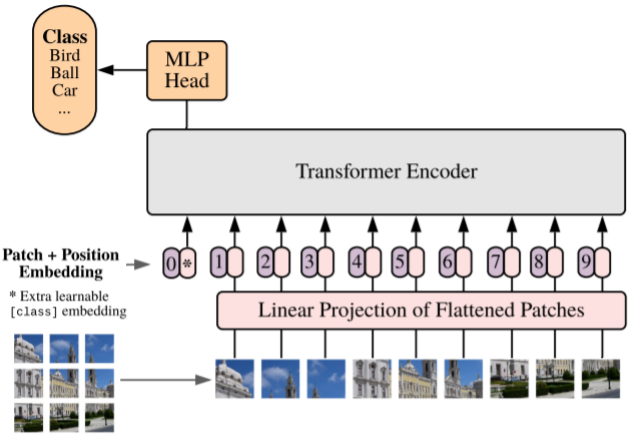

Figure 1: Overview of the ViT model as given in [7]. The image is split into fixed size patches. The patches are embedded and a position imbedding is added. Classification is done by adding an extra token to the input and forwarding the the Transformer output for this token to an MLP

Vaswani et al. [25] proposed an encoder-decoder architecture for processing a 1D sequence of tokens. The ViT is comprised of the encoder section, and similarly receives an arbitrary length 1D sequence as an input. See Figure 1. To handle 2-dimensional images, the fixed size image is decomposed into a batch of N patches of fixed size of 16×16 pixels. The input sequence length is therefore $\frac{W}{16}\mathrm{x}\frac{H}{16}$ for image $I \in \mathbb{R}^{H\mathrm{x}W\mathrm{x}C}$. Each 2D patch is projected with a trainable linear layer to form a latent vector with dimension d. This is referred to as the embedding vector. The Transformer block described here is invariant to the order of the patch embeddings, and thus does not consider their relative position. To this end, positional information is incorporated as trainable positional

embeddings and added to the embedding vectors. The later are then fed to the stack of Transformer blocks.

In order to perform classification using a Transformer encoder, an extra token is appended to the input embedding vector sequence. Referred to as the class token, this is a trainable vector which is forwarded through the Transformer. The model's output corresponding to the class vector is passed to a MLP which outputs a probability distribution over the classes. The ViT processes the $N+1$ input tokens and is trained in a supervised manner using negative log likelihood loss on the MLP output.

The Transformer encoder [25] consists of L blocks, each is built of multiheaded self-attention (MSA) and MLP blocks. Within every block, layer normalization is applied before the MSA and the MLP. Residual connections are placed after the MSA and the MLP. See Figure 1 in [7].

3.2 Area Attention

Attention mechanisms are typically designed to focus on individual items in the entire memory, where each item defines the granularity of what the model can attend to. The fixed granularity limits the model from modeling complex attention that might be needed for the tasks at hand. The writers of [12] suggested to solve this limitation by applying attention on *areas* instead of items. An area is an aggregation of a varying number of items. When an area contains a single item then it is equivalent to regular attention mechanism. Therefore area-attention is in fact a generalization of the original mechanism.

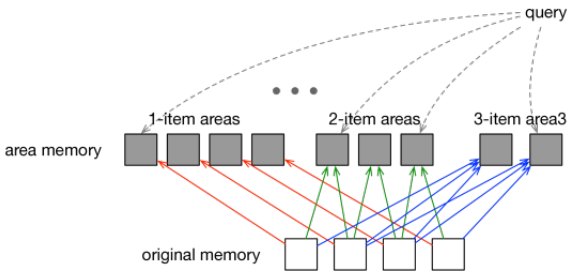

Figure 2: One dimensional area-attention as given in [12]. The input to the layer is a sequence with length 4. For maximal area of 3 we get 9 different areas.

When the memory consists of a sequence of items, a 1-dimensional structure, an area is a range of items that are sequentially (or temporally) adjacent. When the memory contains a grid of items, a 2-dimensional structure (as in image-related tasks), an area can be any rectangular region in the grid. The number of areas is set according to the maximal area size. In figure 2 for example, the maximal size is 2 which leads to 9 different areas.

Given query **Q**, key **K** and value **V** area-attention works the same as other attention mechanisms. The output of the attention is the weighted sum of a set of l value vectors ($\mathbf{V} \in \mathbb{R}^{l\mathrm{X}d}$). For a sequence of **N** query vectors ($\mathbf{Q} \in \mathbb{R}^{N\mathrm{X}d}$), it produces an output matrix:

$$Attention(Q,K,V) = SoftMax(QK^T/\sqrt{d})V \qquad (1)$$

Each area is identified by its key and value, these are functions of the keys and values of the items that comprise the area. The writers suggest two methods of representing the area by its items, one that involves learnable parameters and another that does not. Due to the fact that the first method did not prove to be better, we used the second one - equations 3 and 4 from [12].

The computation of K and V is done efficiently by using a summed area table. This optimization technique is based on the integral image [23] concept and allows asymptotic runtime of $\mathcal{O}(MA)$. Where M is the sequence length (number of tokens) and A is the maximal area size.

3.3 Area Attention Encoder

We suggest a simple yet effective modification to the ViT model: for the first L_{AA} encoder blocks, the MSA layer is replaced with multiheaded area-attention. The rest $L - L_{AA}$ blocks remain unchanged. See figure 3.

1D or 2D area-attention: The image patches can be placed on a two dimensional grid, making the 2D area-attention the natural choice. After adding another extra token, the class embedding, this is no longer the case as the latter has no spatial meaning. Therefore our implementation uses one-dimensional area-attention.

The area-attention layer operates as follows: the input $Z \in \mathbb{R}^{(N+1)\mathrm{X}d}$ (N+1 input vectors, each with dimension d) is linearly mapped to $Q,K,V \in \mathbb{R}^{(N+1)\mathrm{X}d_h}$. As suggested in [24], we keep $d_h = d$. We then calculate the keys and values for each of the areas. The total number of areas N_{areas} is:

$$N_{areas} = (N+1-A)A + (A+1)A/2 \qquad (2)$$

Where A is the maximal area size. The K,V calculation for each area is as follows:

$$K^{(i)}_{area} = \frac{1}{|r_i|}\sum_{j=1}^{|r_i|} K_{i,j} \quad \forall 1 \le i \le N_{areas} \qquad (3)$$

$$V^{(i)}_{area} = \sum_{j=1}^{|r_i|} V_{i,j} \quad \forall 1 \le i \le N_{areas} \qquad (4)$$

where $|r_i|$ is the size of the area r_i

The attention weights are then calculated using $Q_{area}, K_{area}, V_{area}$ using equation 1. As in regular MSA, the model runs k self-area-attention operations, in parallel, and project their concatenated outputs. To keep compute and number of parameters constant when changing k, is set to d_h/k.

It should be noted that calculating the keys and values using equations 3&4 is parameter free. Therefore, our proposed method does not add any weights that should be back-propagated.

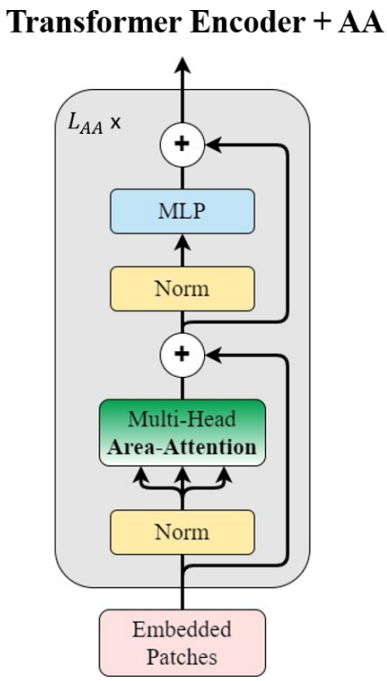

Figure 3: The basic Transformer encoder block, with the area-attention layer

Training scheme: We follow the training strategies presented by Touvron et al.[24] . Compared to models that integrate more priors (such as convolutions), Transformers require a larger amount of data. Therefore, in order to train with smaller datasets, the writers rely on extensive data augmentation. The following augmentations flows are used: Rand-Augment [4] and random erasing [32]. Mixup and cutmix are also used.

4 Data

CIFAR-10 [10] dataset is used for evaluating our modified architecture. The data consists of 60000 32x32 color images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images. The motivation for choosing this small dataset is that we wished to test if it is possible to learn a reasonable Transformer on CIFAR-10 only. In order to compare our models with the ones presented in [24], we followed their experiment settings. The 32x32 CIFAR-10 images were resized to ImageNet size (224x224), and the same augmentations were used. See figure 4.

Figure 4: Examples of training data. For each pair, the left is the original CIFAR-10 image, and the right is the same image after preprocessing. Not true to scale.

5 Experiments

Above all, we would like to clarify that the purpose of our experiment is not trying to achieve the best results possible on CIFAR-10. Rather, we wish to examine how our model performs relatively to the models presented in [24].

5.1 Setup

Dataset: As mentioned earlier, all of the following experiments are on CIFAR-10. This dataset was significantly easier to test and develop due to its low running time.

Model Variants: All of the ViT models we used in our experiments are based on those used in [24]. Their architecture design was based on a thinner version of the ViT-Base architecture (depicted in Table 1), which was proposed by Dosovitskiy et al. [7]. The writers of [24] introduced two smaller models originally named DeiT-S and DeiT-Ti, (for simplicity we named them ViT-Small and ViT-Tiny). For these models they changed the number of attention heads and the embedding dimension, while keeping the dimension of each attention head constant (equal to 64). We based on these two small models for both the baseline we compare our results to, and for our modified ViT models, which we described earlier in section 3.3. The configurations we used for these four model variants are summarized in Table 2.

The motivation to create these smaller models was to have a lower parameter count, and a faster throughput, meaning more images processed per second on a GPU.

An important point to see in Table 2, is that we don't pay any additional cost in regards of the number of parameters by replacing the MSA layer self-attention with multiheaded area-attention, as it remains the same.

General Configurations: First of all, in a similar way to the implementation in [12] we decided to set $L_{AA} = 2$. Meaning, replacing the MSA layer with multiheaded area-attention only in the first 2 encoder blocks. Secondly, after checking several sizes of image patches, we decided to follow the same image patch break down of an input image as in ViT, as described in section 3.1. Meaning, image patches of size of 16×16 pixels. Therefore, since we used input images of size 224x224, the sequence length is:

$$M = N + 1 = \frac{224\text{x}224}{16\text{x}16} + 1 = 197 \tag{5}$$

Number of areas: To calculate the number of these areas, according to equation 2, one needs to first set A, the maximal area size allowed. As we will further elaborate in the next subsection, we configured A across different values. By setting $A = 2$, we get that, using equations 2 and 5, the total number of areas is: $N_{areas} = 393$.

Training Parameters: For all experiments we used both Nvidia T4 and K80 GPUs. We used on both google colab and on Google Cloud Platform (GCP). The total number of epochs in the triaining of all the models was 300.

	embedding	#heads	#layers	#params	training resolution
ViT-Base	768	12	12	86M	224

Table 1: the ViT-Base architecture which was proposed in [7].

	embedding	#heads	#layers	#AA layers	#params	training resolution
ViT-Small	384	6	12	N/A	22M	224
ViT-Small + AA	384	6	12	2	22M	224
ViT-Tiny	192	3	12	N/A	5M	224
ViT-Tiny + AA	192	3	12	2	5M	224

Table 2: Model variants we used for our experiments. The only parameters that vary across models are the embedding dimension and the number of heads, and we keep the dimension per head constant (equal to 64).

5.2 Comparison to Data-Efficent Image Transformers (DeiT)

5.2.1 Accuracy

The experiments are devided into two parts:

- Evaluating the effect of changing the values of A, the maximal area size allowed, on the performence of the ViT-Small+AA model. We were interested to see how the combined number of adjacent items into one area will affect the accuray of our model.
- Comparing ViT-Tiny+AA and ViT-Small+AA models, both with $A = 2$, to their baseline models (ViT-Tiny and ViT-Small, respectively).

ViT-Small model results: Figure 5 displays the top-1 test accuracy over 300 training epoch for ViT-Small model. Table 3 shows the test accuracies and test loss values for this model at the last (300th) training epoch. As seen from both sources, between the values $A = 2, 3, 4$, the one with $A = 2$ achieved the highest accuracy.

Comparing to the baseline model, among all different maximal area size configurations for ViT-Small+AA model, the one with $A = 2$ indeed manages to achieves better accuracy than its baseline, the ViT-Small model, while other configuratios achieves inferior accuracy.

ViT-Tiny model results: Table 4 shows the test accuracies and test loss values for this model at the last (300th) training epoch. After acknowledging (from ViT-Small model) that $A = 2$ achieves best accuracy among all A configurations, we tested our AA model we only for $A = 2$, for which it also manages to achieve better accuracy than its baseline, the ViT-Tiny model.

5.2.2 Training Time

During the training phase, there was a training time difference between our model and the baseline: on Nvidia T4 GPU, the ViT-Small+AA model with $A = 2$ takes 50 seconds more (4:30min vs 5:20min) per one epoch over 50,000 CIFAR-10 training images.

model	maximal area size allowed (only for AA)	top-1 accuracy	top-5 accuracy	loss
ViT-Small + AA	**2**	**92.19**	**99.68**	**0.38**
ViT-Small + AA	**3**	**89.32**	**99.51**	**0.473**
ViT-Small + AA	**4**	**85.67**	**98.29**	**0.577**
ViT-Small	N/A	90.6	99.47	0.411

Table 3: Accuracy results on the CIFAR-10 Testset for ViT-Small model

model	maximal area size allowed (only for AA)	top-1 accuracy	top-5 accuracy	loss
ViT-Tiny + AA	**2**	**86.14**	**9.52**	**0.557**
ViT-Tiny	N/A	85.49	399.49	0.576

Table 4: Accuracy results on the CIFAR-10 Testset for ViT-Tiny model

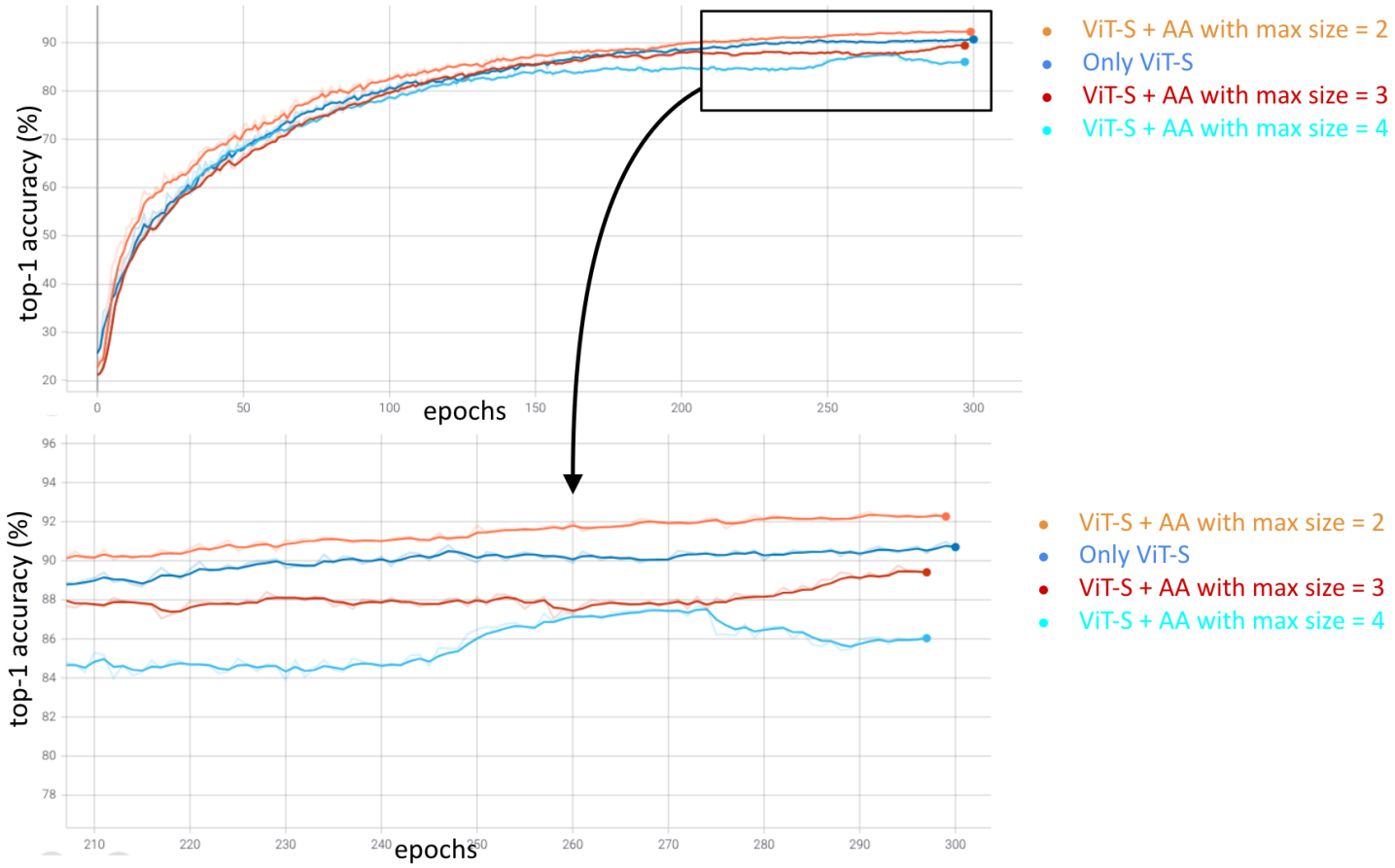

Figure 5: Accuracy graphs for ViT-Small models with different configurations: The upper graph is the accuracy along all training epochs. The bottom graph is the accuracy along last ±90 training epochs

6 Conclusion

In this work, we evaluated the effect of modifying the attention mechanism of a Transformer encoder, when used for image classification. Our experiments reveal that by replacing few of the model's MSA layers with its area-attention counterpart, the new architecture was able to achieve better result on the CIFAR10 dataset. This improvement was evident in both the tiny and small models. Testing multiple area-attention configurations showed that the outcome is sensitive to the maximal area size, as only one out of the three sizes tested led to an improvement.

While earlier works showed that area attention can improve the Transformers accuracy when using feature maps as input, it is shown that the ability to attend to information at varying granularity is also beneficial when the model uses images as input.

Possible future extensions:

- Exploring how Transformer that incorporates area-attention performs on object detection tasks and other vision tasks.
- Evaluating the effect of increasing L_{AA}, the number of area attention layers.
- Evaluating the effect of adding a distillation-token, as described in the second part of [24].
- Evaluating the effect of area key calculations that are different from equation 3, such as equation 9 in [12]. Evaluating the effect of other area representations.

7 Appendix

- Our modified ViT model is based on implementation from : `https://github.com/lucidrains/vit-pytorch`.
- The area-attention layer is implemented in pytorch in `https://github.com/mikomel/area-attention`.
- Training scheme is based on the data-efficient flow presented in [24] and implemented in `https://github.com/facebookresearch/deit`.
- Our additions and training code is available at `https://github.com/YoavKurtz/ViT-AA`.

References

[1] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering, 2018.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural Machine Translation by Jointly

Learning to Align and Translate. *arXiv e-prints*, page arXiv:1409.0473, September 2014.

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers, 2020.

[4] Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Practical data augmentation with no separate search. *CoRR*, abs/1909.13719, 2019.

[5] Zhigang Dai, Bolun Cai, Yugeng Lin, and Junying Chen. Up-detr: Unsupervised pre-training for object detection with transformers, 2020.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[7] 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. *arXiv preprint arXiv:2010.11929*, 2020.

[8] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015.

[9] Yoon Kim, Carl Denton, Luong Hoang, and Alexander M. Rush. Structured attention networks, 2017.

[10] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[11] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, Dec 1989.

[12] Yang Li, Lukasz Kaiser, Samy Bengio, and Si Si. Area attention. *CoRR*, abs/1810.10126, 2018.

[13] Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. Neural baby talk, 2018.

[14] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. *CoRR*, abs/1508.04025, 2015.

[15] Vlad Niculae and Mathieu Blondel. A regularized framework for sparse and structured neural attention, 2019.

[16] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Łukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer, 2018.

[17] Marco Pedersoli, Thomas Lucas, Cordelia Schmid, and Jakob Verbeek. Areas of attention for image captioning, 2017.

[18] Marek Rei and Anders Søgaard. Jointly learning to label sentences and tokens, 2018.

[19] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[20] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era, 2017.

[21] Zhiqing Sun, Shengcao Cao, Yiming Yang, and Kris Kitani. Rethinking transformer-based set prediction for object detection, 2020.

[22] Sheng syun Shen and Hung yi Lee. Neural attention models for sequence classification: Analysis and application to key term extraction and dialogue act detection, 2016.

[23] Richard Szeliski. *Computer Vision: Algorithms and Applications*. Springer-Verlag, Berlin, Heidelberg, 1st edition, 2010.

[24] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *arXiv e-prints*, page arXiv:2012.12877, December 2020.

[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. *arXiv e-prints*, page arXiv:1706.03762, June 2017.

[26] Ross Wightman. Pytorch image models. https://github.com/rwightman/pytorch-image-models, 2019.

[27] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. Image captioning with semantic attention, 2016.

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

[29] Li Yuan, Francis E. H. Tay, Guilin Li, Tao Wang, and Jiashi Feng. Revisit knowledge distillation: a teacher-free framework, 2020.

[30] Yanhong Zeng, Jianlong Fu, and Hongyang Chao. Learning joint spatial-temporal transformations for video inpainting, 2020.

[31] Minghang Zheng, Peng Gao, Xiaogang Wang, Hongsheng Li, and Hao Dong. End-to-end object detection with adaptive clustering transformer, 2020.

[32] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation, 2017.

[33] Luowei Zhou, Yingbo Zhou, Jason J. Corso, Richard Socher, and Caiming Xiong. End-to-end dense video captioning with masked transformer, 2018.