AN ABSTRACT OF THE THESIS OF

Jisoo Lee for the degree of Master of Science in Computer Science presented on July 27, 2022.

Title: Window Axial Vision Transformer for Image Classification

Abstract approved: ______________________________

Sinisa Todorovic

Currently a popular approach to image classification uses the deep Transformer architecture. In a Transformer, the attention mechanism enables the model to learn efficiently with fewer computational resources than the convolutional neural networks (CNNs). In this thesis, we study the sparse attention mechanism widely used in the Transformers developed specifically for natural language processing (NLP). We generalize these models to enable the processing of 2D images. The resulting new models specified in this thesis have fewer parameters, and as we show experimentally give good results on image classification. In particular, from our experiments, the well-known problem that the vision Transformers (ViT) lack the capability to model prior knowledge is compensated in our new models by adopting a local attention estimation. This helps our models to perform well even on small datasets. Evaluation is presented on the benchmark datasets for image classification including CIFAR-10, CIFAR-100, and ImageNet-1K. A comparison with ViT--a popular Transformer in computer vision--shows that our new models outperform ViT on all three datasets.

Window Axial Vision Transformer for Image Classification

by

Jisoo Lee

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented July 27, 2022
Commencement June 2023

Master of Science thesis of Jisoo Lee presented on July 27, 2022.

APPROVED:

Major Professor, representing Computer Science

Head of the School of Electrical Engineering and Computer Science

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Jisoo Lee, Author

ACKNOWLEDGEMENTS

I would like to express my gratitude to my advisor, Dr. Sinisa Todorovic, who has advised me since Fall 2020. Although I had to be in South Korea for a year due to the pandemic, he always supported and guided me to keep my study. I also would like to thank Dr. Fuxin Li, Dr. Stefan Lee, and Dr. Abdollah Farsoni for their feedback and advice. I am also grateful to Computer Vision Lab members, Liqiang, Nicolas, Tieqiao, and Ajay for giving me insight into this study and feedback. Thanks should go to Dr. Valerie Goodness for helping the writing of this thesis. To my friends, Sujung, Dayeon, Mal, Mark, Shahab, Kamesh, Richard, Hyunhee, Sojeong, Camila, and Pauline I would like to say thank you for supporting me mentally and inspiring me. Finally, I would like to thank my family who always makes me smile and give me the driving force behind all of this.

TABLE OF CONTENTS

LIST OF FIGURES

Figure Page

LIST OF TABLES

LIST OF ALGORITHMS

Algorithm Page

Chapter 1: Introduction

1.1 Problem Statement

Our research focuses on image classification which is a fundamental problem in computer vision. In image classification, images are assigned specific labels.

Recently, the state of the art in image classification are deep learning models. Among many deep learning approaches, convolutional neural networks (CNNs)[1] are dominant. Recently, transformer-based models have also become popular. Our research is based on Transformer models, which use self-attention. Self-attention is a way of enriching features at every image location based on their similarities to features of neighboring locations which are called spatial context. In this thesis, we use transformer-based models for image classification and present evaluations on small datasets like CIFAR-10 and CIFAR-100 to the large dataset ImageNet-1K.

1.2 Motivation

There are already many approaches to image classification that use transformer-based models. However, transformer-based models poorly encode the prior knowledge about object classes in images whereas CNN-based models are known to perform better. Thus, transformer-based models need to learn inductive bias from the data, and for that, large datasets are required to successfully encode this prior knowledge. Due to massive training data requirements, training a transformer-based model requires large computational resources. Our goal is to reduce the computational complexity of training. Also, we would like to enable models to have inductive bias so that they can perform well even with small datasets.

1.3 Contributions

There are three main contributions of our research. First, we reduce computational complexity by adopting a sparse self-attention mechanism. Since our models do not compute the full attention map as in Vision Transformer (ViT)[2], the model has fewer parameters. With fewer computations, we achieve competitive results compared to the baselines. Second, our models are enabled to have inductive bias. CNN models make the assumption that the features from the close-by locations in the image are semantically related. We enable the same modeling property for our transformer-based models by using a window-based self-attention. Third, our models are modular and can be easily integrated into larger deep architectures. We stack two types of transformer blocks with various combinations and, in this way, we make many different models. Therefore, our models can be used as a backbone for various vision tasks.

1.4 Evaluation

We evaluate the performance of models using top-1 accuracy, number of parameters and number of FLOPs. As benchmarks for image classification, we use three different datasets: CIFAR-10, CIFAR-100 and ImageNet-1K.

Chapter 2: Literature Review

In this section, we review closely related literature on image classification. Image classification is a long-standing problem in computer vision and there are many approaches from traditional computer vision algorithms to deep learning methods. Therefore, reviewing the entire computer vision algorithms is beyond our scope. In this section, we only focus on reviewing papers on deep learning architectures for image classification.

For image classification tasks, we need to extract meaningful features from images. Extracting high-level features like color, edge, and corner is an important part in many computer vision tasks. Based on these features, models reason the final class labels which correspond to images. These high-level features can be detected by traditional computer vision algorithms such as SIFT[3] and SURF[4], but neural networks have replaced these algorithms for decades. For example, people have been using neural network models since 1989 when neural nets succeeded in classifying images on MNIST which is handwritten digit classification dataset. They train these models to do specific tasks. Models receive data, such as images, as input and are trained on these datasets. After the emergence of a large amount of data, consisting of 14 million images referred to as ImageNet, various deep learning-based models have shown improved performance in image classification contests since 2010. Deep learning models are also called deep neural networks (DNNs) because they consist of a large number of neural networks. Furthermore, in addition to a huge amount of data, sufficient GPU resources made the deep learning method possible. Therefore, with the deep learning method, we no longer need to extract features manually, and the features are obtained by training the model in an end-to-end manner.

2.1 Convolutional Neural Net models

The most well-known type of neural network for processing 2D image data is the convolutional neural network. The first CNN model, named LeNet[5] was proposed by Yann LeCun in 1998 to process grayscale handwritten digit images. The model extracts features through multiple 2D convolutional layers. Specifically, 2D-shaped filters scan the entire image in a sliding window method. It converts pixel values, which are low-level features into high-level feature vectors. Recently, thanks to the acceleration of computing power and huge datasets, more complex CNN models have emerged. AlexNet[6] can process RGB images and handle more classes by utilizing GPUs. After that, VGG[7], GooGleNet[8] and ResNet[9] have suggested and many models including these continue to update the best performance in ImageNet benchmark to this day. In addition, the models for more complex tasks like object detection and image segmentation are still mainly using ResNet as a backbone for feature extraction. However, since these CNN models involve a pooling layer, it is easy to lose information about relative spatial relationships. Also, in order to collect enough contextual information, the size of the receptive field needs to be increased, which requires a large amount of computation.

2.2 Transformer models

The Transformer[10] was first suggested in Natural Language Processing (NLP) for language translation tasks. This model only relying on attention mechanisms can process a long input sequence. A self-attention mechanism in the Transformer defines the similarities between tokens regardless of their physical distance. As a result, they aggregate the global information and these interactions between tokens generate more meaningful context by enhancing the important features and fading out the other part.

To be specific, the input of a self-attention layer is a set of tokens. Each token is used as a query, key, and value. A self-attention function uses a scaled-dot product method to calculate the similarity between a query and a set of keys, that is, the output becomes the weighted sum of the values using the weight. This allows updated feature vectors to be more contextualized rather than the input vectors of the self-attention layer. This can be expressed by the following formula. After computing the dot products of the query

Q with the set of keys K, divide them by the square root of the dimension of k $\sqrt{d_k}$. Apply a softmax function to them to get the attention score, and then finally multiply them to the values V to give weights.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{2.1}$$

Since the Transformer was designed to process 1D data, the vision Transformer (ViT)[2] was proposed to process 2D image data. The input images are split into fixed-size patches, and each patch is treated as a token. The attention function computes the attention score between these tokens to find similarities. By updating the feature vector with the attention score, a meaningful relationship between each patch in the image can be identified. While CNN uses kernel for convolution computations to capture local information, ViT captures information globally within the entire image. Furthermore, the computation efficiency of ViT is superior to that of CNNs and still shows competitive performance. Figure 2.1 shows ViT architecture. Position embeddings are added to each patch and become the input of ViT along with an additional learnable class token. Finally, the class label is predicted as the value of the class token.

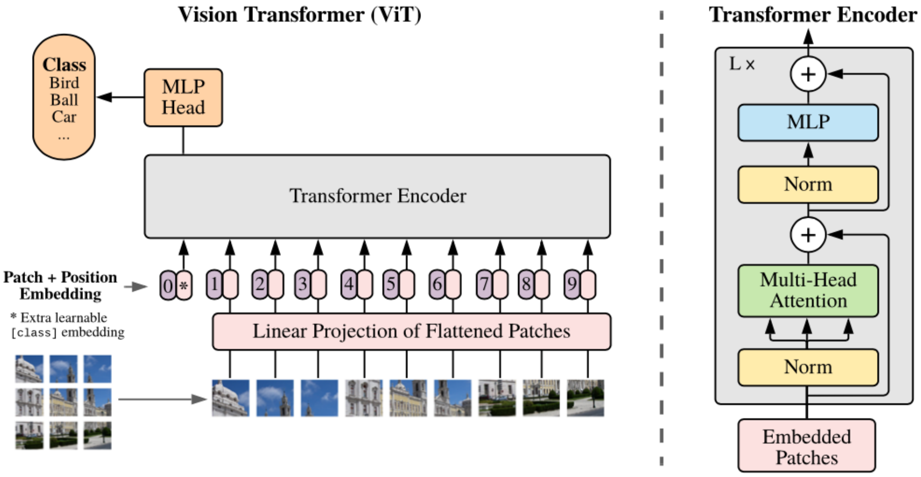

Figure 2.1: Vision Transformer architecture

However, Transformer-based models still require a lot of computations and especially when the image size is big, the computational cost is quadratically increased. This is because the self-attention function computes the attention score as much as the square of the number of tokens. To address this problem, the Swin Transformer[11] introduced a window attention mechanism. By restricting attention boundary within the windows, they compute local attention, and they exchange the information between the tokens of long distance in an implicit way using shifted windows. Whereas global attention mechanisms consider all tokens to put attention, local attention only attend within the window. Thus the Swin Transformer reduces computational complexity from quadratic to input image size to linear because they are independent from the image size.

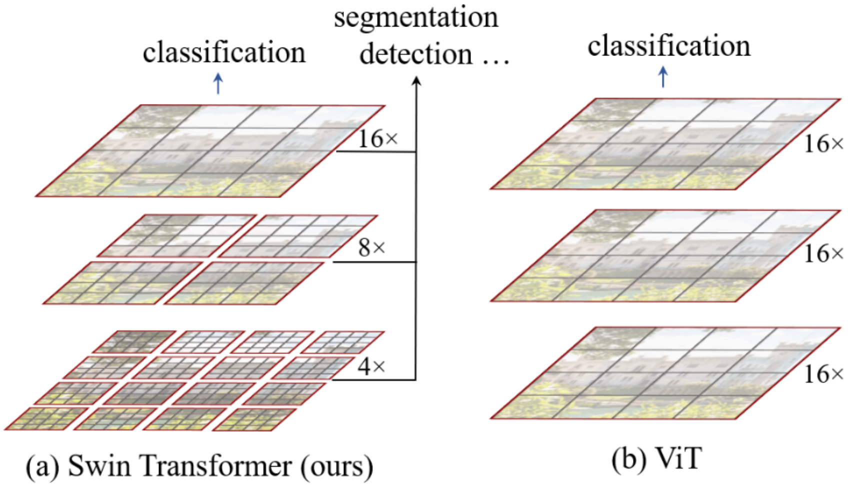

Figure 2.2: Swin Transformer architecture

In NLP, where the Transformer was first proposed, many studies have been already done on sparse transformers without obtaining the full attention map. The existing transformer has difficulty processing a longer sequence and getting the relationship between tokens, and this arises a need for sparse attention mechanisms. The Reformer[12] computed the attention score between pairs with high influence to attend. The Longformer[13] partly used a local attention mechanism using a sliding window method as well as adopting a global attention. The Bigbird[14] added a random attention which allows each token to randomly select a key set of tokens to the idea of the Longformer. However, since these research on sparse Transformers target 1D data, there is a limit to direct application to 2D image data.

The MSA Transformer[15] is one of the sparse Transformers targeting 2D data. This model which was suggested to model protein languages takes the multiple sequence alignment (MSA) as input. The input of the model is $x \in R^{M \times L}$, M is the number of sequences in the MSA and L is the number of positions in the aligned sequence. It computes attention only within each column and row rather than using the full attention layer. This constraint reduces computation cost from $O(M^2L^2)$ to $O(LM^2) + O(ML^2)$ where M is the number of rows and L is the number of columns of the attention map.

For computer vision tasks, CCNet[16] was suggested for semantic segmentation. To catch dense contextual information as well as to use GPU resources efficiently, they introduced a cross-shaped attention mechanism. Features attend to the other features which are located along a cross shape. This process is repeated twice to update the feature map. Similar to CCNet, Axial-Deeplab[17] uses the model consisting of two axial attention layers for panoptic segmentation. CSWin Transformer[18] adopted a cross-shaped window self-attention. They use the window self-attention of the Swin Transformer but they modified a square window to a long rectangle window to catch the global context.

Chapter 3: Our Approach

Our models targeting image classification tasks, take image datasets as input, and in these 2D datasets, the location of each token is more meaningful unlike the datasets in NLP or protein language. Therefore, rather than directly applying the sparse Transformer models in NLP, we developed our models by adopting the concept that the Swin Transformer uses patches within the window as tokens and the idea that the CSWin Transformer uses a cross-shaped window.

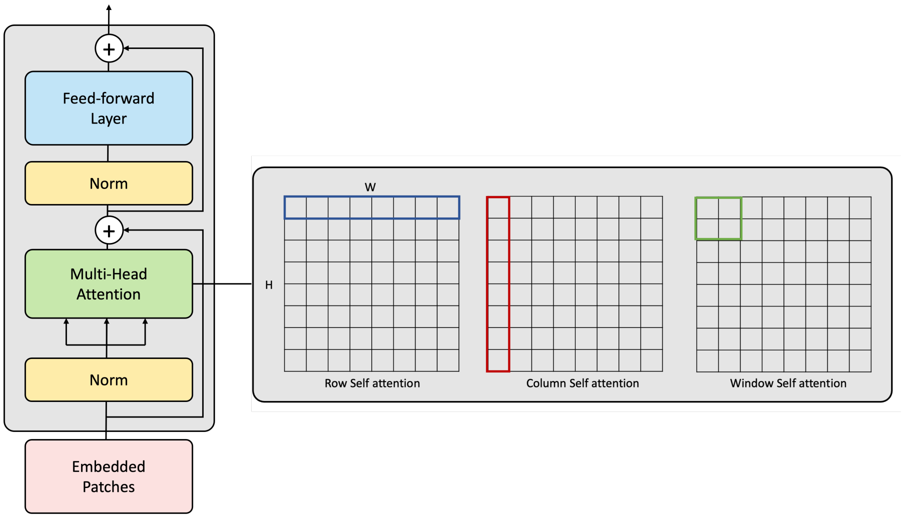

Figure 3.1: WinAxial Transformer architecture

Our model, the WinAxial Transformer consists of two types of transformer blocks: Window Transformer and Axial Transformer. A window Transformer has a window self-attention layer. An axial Transformer has a row self-attention layer and a column self-attention layer in sequential. Figure 3.1 is the architecture of the WinAxial Transformer. The main frame is the same as ViT but the self-attention layer in each Transformer block is replaced with row attention, column attention, or window attention. Each attention function uses a multi-head attention mechanism like ViT. In Transformer blocks, an attention layer is followed by a feed-forward layer consisting of two linear layers with an activation function. LayerNorm is applied before each layer and there are residual connections after each layer. For the final prediction for image classification, we adopted global average pooling instead of adding a class token.

3.1 Window self-attention

In images, features that are close to each other are typically semantically related to each other. To catch the relations locally, we adopted the window attention mechanism of the Swin Transformer. They use a fixed size window to reduce the computation compared to the full attention function, and they focus on obtaining local information. However, we don't use the idea of the shifted window in the Swin Transformer, we use the attention score obtained within non-overlapping windows. These attention scores update the features locally without interactions between windows. The computational complexity is linear with the image size, like the Swin Transformer.

3.2 Axial self-attention

We apply the axial attention mechanism, one of the sparse transformer models, to the vision task. Our axial transformer is developed by sequentially stacking row self-attention and column self-attention. These attention mechanisms have the same principle as window self-attention in that they limit the range of attention computation. Instead of a square-shaped window, strip-shaped windows are used to identify similarities within the same column and row. This enables features of patches on the same axis to interact with each other regardless of distance. By exchanging information in the row direction and the column direction through each axial self-attention layer, the models can grasp global information without the full attention map as in ViT.

Instead of stacking identical transformer blocks like ViT, we stacked window transformer blocks and axial transformer blocks in various combinations to exchange the information locally and globally. Without shifted window strategy in the Swin Transformer, we make it easier to access the patches in a long distance in fewer steps. Our feature map from these attention layers enables itself to have global information so even this can be used as an Encoder part for dense vision tasks such as object detection or segmentation. The specific model configurations will be described in detail in the next chapter.

By introducing this new model, we can reduce the computational cost from $O(W^2H^2)$ to $O(W^2H)+O(WH^2)+O(window\_size^2WH)$ which W is the width and H is the height of the input dimension.

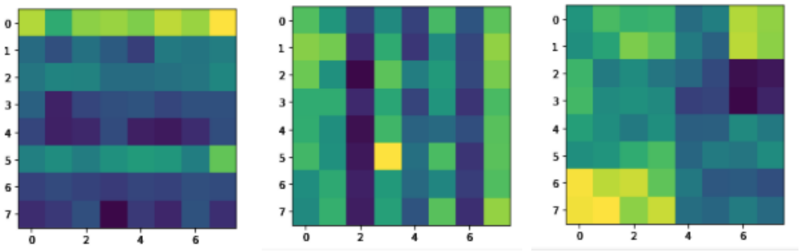

Figure 3.2: Attention map visualization

Figure 3.2. shows the attention map of the model which consists of a row, column, and window attention layers. After a row self-attention layer, they show the relationships between features along a horizontal direction. After a column self-attention layer, vertical lines are distinguished from the other vertical lines, which means they focus on the relationship along vertical lines. At the final stage, after a window self-attention, they compute the similarities within local windows. These repeated processes finally make the model find the meaningful areas in the image.

3.3 Feed Forward Network

In each transformer block, the self-attention layer is followed by a fully connected feed-forward network (FFN). This FFN consists of two linear layers with the activation function, GELU. Layer normalization is performed before the FFN as well as the multi-head self-attention layer, and the output of each layer is connected to the value before the normalization by a residual link.

3.4 Global Average Pooling

Our models use global average pooling like the Swin Transformer instead of the class token to predict the final output after the linear classifier. By eliminating the class token, the tokens consist only of image patches, therefore, the models can focus more on defining the relationship between them.

Chapter 4: Experiments

We conducted experiments on three datasets: CIFAR-10, CIFAR-100, and ImageNet-1K for image classification. CIFAR-10 dataset has 60,000 images in 10 different classes. CIFAR-100 dataset contains 600 images and has 100 classes. Each class has 500 training images and 100 test images. ImageNet-1K is a larger dataset than CIFAR-10 or CIFAR-100 but still is the subset of ImageNet-21K which is a massive dataset. ImageNet-1K has 1000 object classes and consists of 1,281,167 training images, 50,000 validation images, and 100,000 test images. Our experiments do not include any data augmentation and pre-training. On each dataset, we trained the models from scratch.

Models on each dataset were trained on 4 Nvidia Tesla V100 GPUs. For training CIFAR-10 and CIFAR-100, we set the batch size to 256 and trained 100 epochs. The resolution of input images is resized to 32x32 and the patch size is 4x4. Learning rate of the model is 0.003 and we used a cosine annealing learning rate scheduler after linear warmup. Learning rate warmup steps are 1.3K, which is about one-fifteenth of the total steps. The hidden dimension is 768 and the feed-forward network dimension is 3072, which is four times the hidden dimension. The Vanilla ViT model has 12 self-attention layers and each attention layer has 12 heads for multi-head attention. The architecture of the Swin Transformer is exactly the same as the paper[11]. In our models, window attention layers employ window size 2. For training, we adopt AdamW optimizer, and learning rate decay is set to 0.3 and set gradient clipping norm to 1. For each step, we update the learning rate and then the loss. The only different setting between CIFAR-10 and CIFAR-100 is the number of classes.

When we train ImageNet-1K, the batch size of ViT is 128 and it is set to 512 for training WinAxial Transformer. The input resolution is 224x224 and the patch size is 16x16. The patch size of the Swin Transformer is still 4x4 here. We use a cosine annealing learning rate scheduler after linear warmup for training ImageNet-1K. Warmup steps are one-fifteenth of the total training steps. The learning rate for Vanilla ViT is

0.00009375 and that of our model is 0.001. Model architectures are the same as that of training on CIFAR-10 and CIFAR-100 except for window size and patch size. As the resolution increases, the window size is set to 7. We trained for 100 epochs. In Vision Transformer paper[2], the original experiment trained the model for 300 epochs but we reduced them to one-third due to limited hardware resources. In order to train 100 epochs, it takes almost 3 days. For all training, dropout rate is set to 0.1 and it is applied to the attention maps.

Our models are various combinations of Window self-attention and Axial self-attention. We named each model using four types of self-attention layers: window, axial, window + axial, and axial + window. To represent the various combinations of two types of layers, we simply named them W, A, WA, and AW which stand for each type. Our models' names consist of these attention types and the number of them. Numbers after the characters represent the number of layers. For example, model A2WA1AW1 means it consists of the layers in this order: two axial attention layers, one window attention layer, two axial attention layers, and one window attention layer. AW3 consists of three axial and window attention layers. That means axial and window attention layers are repeated three times. As an evaluation metric, we compared top1 accuracy, flops, and the number of parameters of each model.

4.1 Results on CIFAR-10 and CIFAR-100

On CIFAR-10 and CIFAR-100, we trained our models and as baselines, we trained two ViT-based models and five Swin-based models. As ViT-based models, we use ViT-B with 12 layers and ViT-T with 8 layers. We use five Swin Transformer-based models for the comparison: Swin-B/2, Swin-B/4, Swin-T22/4, Swin-T26/4, and Swin-T48/4. The numbers after the slash are the window size. Since the original Swin Transformer uses the window size 7x7 and they use the shifted window-based mechanism, we explored the larger window size 4x4 as well as 2x2. The original Swin Transformer has four steps in its hierarchical architecture and each stage is repeated 2, 2, 6, and 2 times. The input resolution of CIFAR datasets is 32x32 whereas ImageNet-1K resolution is 224x224. Thus if we use four stages, the model should process a 1x1 token at the last stage which means the window size is bigger than the number of tokens. To resolve this problem, we modified the Swin Transformer to use two stages. Swin-T22/4 means they have two blocks for each stage. By adopting these two stages, the models process 4x4 patches at the second stage. The numbers followed by T of Swin-T26/4 and Swin-T48/4 are the numbers of blocks at each stage. When we use two stages and use more layers for each stage, the performance became better.

We construct 16 models using various combinations of the window, row, and column attention blocks for comparison with baseline models. Among our models, AW3 is the best performing model on CIFAR-10 and A2WA2 is the best on CIFAR-100. To be specific, AW3 beat the baseline ViT-T by a large margin of 7.81 with less than half of the number of parameters. It also shows better performance rather than every type of Swin Transformer. However, compared to SWIN-T22/4, SWIN-T26/4, and SWIN-T48/4 which use two stages, AW3 is better but the Swin-based models have fewer parameters. In the case of A2WA2, it achieved higher accuracy compared to ViT-based models and Swin models which have four stages but it cannot exceed the modified Swin Transformers.

method	image size	#params	GFLOPs	CIFAR-10 acc	CIFAR-100 acc
ViT-B	32 × 32	78M	5.14	72.17	53.71
ViT-T	32 × 32	52M	3.43	71.69	53.26
SWIN-B/2	32 × 32	28M	0.089	71.4	49.35
SWIN-B/4	32 × 32	28M	0.089	73.26	50.68
SWIN-T22/4	32 × 32	3M	0.003	74.55	56.94
SWIN-T26/4	32 × 32	5M	0.006	75.31	57.05
SWIN-T48/4	32 × 32	6M	0.009	75.61	**58.02**
AW3	32 × 32	23M	0.8	**79.5**	55.87
A2WA2	32 × 32	21M	0.57	78.97	**56.81**
AW1A2	32 × 32	21M	0.57	78.63	55.27
W1A3	32 × 32	22M	0.57	78.46	56.05
A3W1	32 × 32	22M	0.57	78.23	56.1
A2WA1AW1	32 × 32	21M	0.57	78.12	55.08
A2AW1	32 × 32	21M	0.57	78.01	54.94
WA1AW1A2	32 × 32	21M	0.68	77.72	55.13
WA2A1	32 × 32	18M	0.34	75.67	53.71
AW1A1	32 × 32	18M	0.34	75.34	54.18
W1A2	32 × 32	20M	0.34	75.17	55.03
A1WA1	32 × 32	18M	0.23	74.4	54.6
A2W2	32 × 32	20M	0.34	74.3	53.65
A2W1	32 × 32	20M	0.34	74.13	53.05
A1AW2	32 × 32	18M	0.23	73.74	53.61
A1WA2	32 × 32	18M	0.23	73.71	54.13

Table 4.1: Results on CIFAR-10 and CIFAR-100

4.2 Results on ImageNet-1K

We trained our models, and then we picked the best-performing models from CIFAR-10 and CIFAR-100 and the well-performing models of both datasets. For one experiment on ImageNet-1K, 3 days were needed whereas CIFAR dataset only required 2 hours.

To be specific, we conducted three comparisons: comparison between ViT-B and our models, ViT-T and our models and the Swin Transformer and our models. The table 4.2 shows the results. Compared to ViT-B, W1A3 shows similar performance with much fewer parameters and less Gflops, and it surpasses ViT-T which has similar number of parameters with margin of 3.41. However, compared to the Swin Transformer, the Swin Transformer shows a better performance with fewer parameters. This is because the Swin Transformer uses a smaller patch and utilizes a mechanism to merge windows using a hierarchical structure.

method	image size	#params	GFLOPs	acc
ViT-B/16	224 × 224	79M	16.2	67.07
ViT-T/16	224 × 224	53M	10.8	64.74
SWIN	224 × 224	28M	4.51	**71.92**
W1A3	224 × 224	58M	7.09	**68.15**
A2WA1AW1	224 × 224	53M	7.09	67.09
A2WA2	224 × 224	53M	7.09	66.5
A2AW1	224 × 224	53M	7.09	66.9

Table 4.2: Results on ImageNet-1K

Chapter 5: Discussion

Our models are designed to benefit from Transformer models as well as CNN models. Transformer-based models are superior in terms of they can capture the global context. Whereas, CNN-based models are good for having prior knowledge of the locality. To focus on local information, the Swin Transformer had been suggested but it still has limitations since it cannot directly exchange global information because they exchange the information only within a limited square-shaped window. Whereas, our models adopt various shapes of windows of self-attention layers to catch local information as well as global context.

Before starting the experiment, we assumed that this strategy could take benefit from both types of models. The window self-attention layer focuses on local area information. The axial self-attention which consists of row self-attention and column self-attention is for interaction within the axis area. Thus, it is possible to grasp the relationship from a distant point. We found the optimal value for the order and number of each attention layer by conducting experiments on models composed of various combinations. Finally, we picked the best option for each dataset.

ViT-B, the baseline model consists of 16 self-attention layers, and our models used 7 to 10 self-attention layers. Among the models, there are models whose number of parameters is almost half the baseline. Our models usually have small numbers of parameters because first, they do not compute the full attention map, and second, the number of layers is less. However, it was assumed that meaningful features could be extracted by repeatedly exchanging local and global information.

As a result of the experiment, our models showed significantly better performance than ViT models on small datasets, and they were still able to achieve similar performance with fewer parameters on the large dataset. Our models surpassed the existing ViT models because our models assumed prior knowledge whereas ViT has to learn these from the scratch. The woven structure of our models shows that the axial attention mechanism can replace the previous full attention mechanism. However, our models do not show significantly better performances when compared to the modified version of Swin Transformer baselines. On CIFAR-10 the accuracy is higher than that of Swin-based models, but the number of parameters of our model is also large. On CIFAR-100 and ImageNet-1K, the modified Swin-based models were better than our models in both performances and the number of parameters.

The distinct difference from the Swin Transformer is that our models don't adopt a hierarchical architecture that is robust to various scales. In the case of training on ImageNet-1K, different patch sizes can be the reason for the performance gap.

In our research, we compared the performance of models only for image classification tasks but, these can be extended to tasks such as object detection and instance segmentation where localization is important. For example, DETR, a transformer-based object detection model uses ResNet as a backbone, which is a CNN-based model. DETR consists of a transformer encoder and decoder and the feature extraction part is ahead of these. Our models can replace the backbone and encoder parts because they have a global context. Further research can be done for dense vision tasks.

Chapter 6: Conclusion

Our models not only reduced the number of parameters through simple modification of the existing model but also gave the Transformer prior knowledge to produce good results with small datasets. In addition, the scale of the model can be adjusted by configuring the two types of attention layers in various combinations. When we bring the transformer structure used in the NLP domain into the computer vision domain, the advantage of the structure can be adopted in a way that can utilize the spatial information of the image data. As a result, while reducing the amount of computation by using the sparse attention mechanism, it has the advantages of both a CNN with inductive bias and a Transformer that finds similar regions regardless of distance through the attention mechanism. These results show that future research can flow in the direction of taking only the strengths among the existing studies, rather than having to trade-off in performance/time, etc.

The core of deep learning technology is to quickly approximate the relationship between input and output by extracting meaningful features from an image. These features are really important because they are also used when performing dense tasks. Although our study only targets image classification, it has room to be used for higher-order tasks as a backbone. The need for these tasks, such as object detection and image segmentation, is accelerating in the industry. It is installed in many machines such as self-driving cars and robots, replacing human roles. In other words, the rapid development of deep learning research is directly related to the development of the industry.

Contributing to the development of artificial intelligence is the improvement of deep learning technology along with powerful hardware. If the problem of resource shortage is solved as in this study, deep learning-based research will increase. Not only can it be used on hardware such as mobile devices which has limited memory and resources, but also in the case of an industry with sufficient resources, more experiments will be possible.

Bibliography

[1] 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, 1989.

[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. *CoRR*, abs/2010.11929, 2020.

[3] Dawid G. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision, 20–25 September, 1999, Kerkyra, Corfu, Greece, Proceedings*, volume 2, pages 1150–1157, 1999.

[4] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3):346 – 359, 2008. Similarity Matching in Computer Vision and Multimedia.

[5] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[6] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[7] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[8] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[11] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *CoRR*, abs/2103.14030, 2021.

[12] Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. *CoRR*, abs/2001.04451, 2020.

[13] Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. *CoRR*, abs/2004.05150, 2020.

[14] Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontañón, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. Big bird: Transformers for longer sequences. *CoRR*, abs/2007.14062, 2020.

[15] Roshan M Rao, Jason Liu, Robert Verkuil, Joshua Meier, John Canny, Pieter Abbeel, Tom Sercu, and Alexander Rives. Msa transformer. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8844–8856. PMLR, 18–24 Jul 2021.

[16] Zilong Huang, Xinggang Wang, Yunchao Wei, Lichao Huang, Humphrey Shi, Wenyu Liu, and Thomas S. Huang. Ccnet: Criss-cross attention for semantic segmentation. *CoRR*, abs/1811.11721, 2018.

[17] Huiyu Wang, Yukun Zhu, Bradley Green, Hartwig Adam, Alan L. Yuille, and Liang-Chieh Chen. Axial-deeplab: Stand-alone axial-attention for panoptic segmentation. *CoRR*, abs/2003.07853, 2020.

[18] Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. Cswin transformer: A general vision transformer backbone with cross-shaped windows. *CoRR*, abs/2107.00652, 2021.

APPENDICES

Appendix A: Redundancy

This appendix is inoperable.