Swin Transformer: Hierarchical Vision Transformer using Shifted Windows

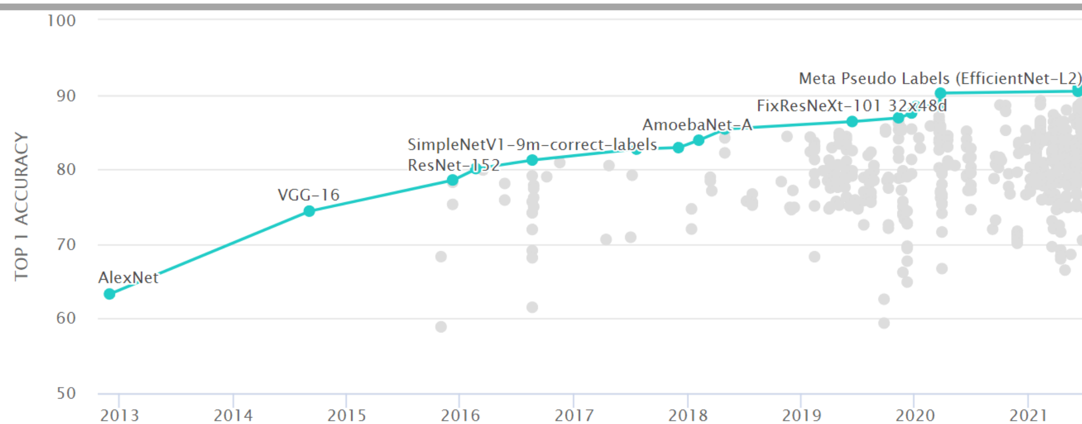
Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, Baining Guo

Proceedings of the IEEE/CVF international conference on computer vision. 2021.

경영과학연구실 김윤석

Background

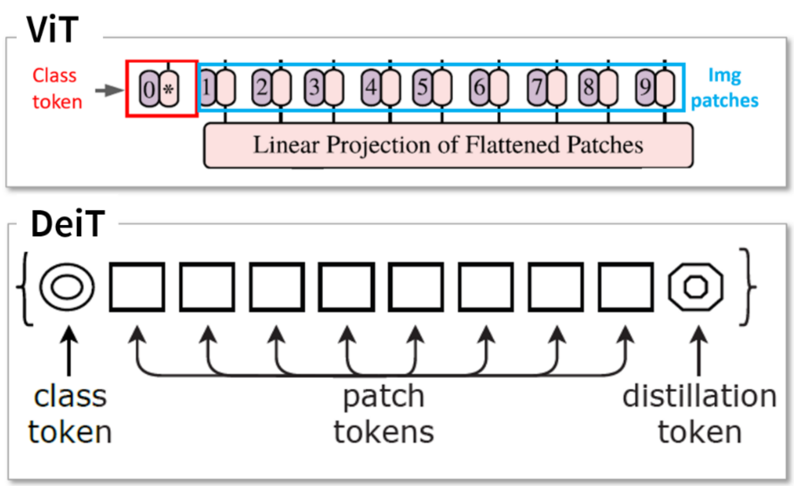
- CNNs have been the mainstay neural networks in computer vision until now.
- Starting with AlexNet, CNNs have been continually evolving.
- The various CNN architectures that have evolved are being utilized as backbone networks in various vision tasks beyond the ImageNet challenge.
- The success of Transformers in natural language processing has led to the proposal of models such as ViT and DeiT.



ImageNet challenge performance graph

Introduction

- Transformer uses word tokens as the base element, but in computer vision, the base element may vary in size.
- Fixed patch input in Vision Transformer can be difficult to understand at the pixel level.
- Vision Transformer shows high training cost and not good performance in tasks such as object detection and semantic segmentation.

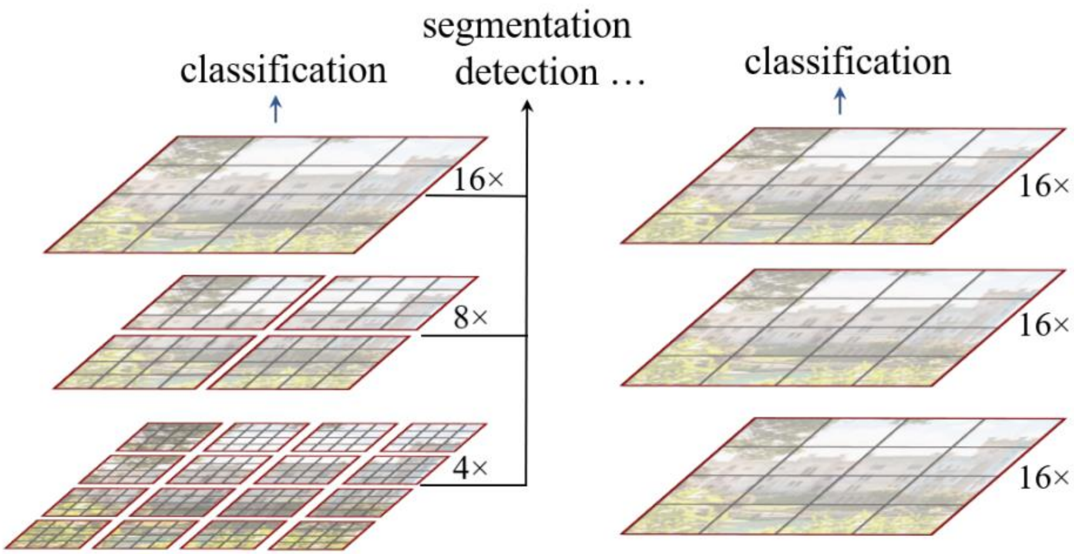


Related works

- Since the advent of AlexNet, CNN has researched and proposed more effective neural network architectures such as VGG, ResNet, DenseNet, and EfficientNet.
- With the success of Transformer, research has been conducted to apply self-attention to CNNs, but there is a problem that the model becomes heavy.
- ViT used Transformer structure for computer vision without modification and achieved impressive performance.
- Many Transformer-based architectures have been proposed to compensate for the shortcomings of ViT.

Problem statement

- They want to create Transformer with 2D data (image) in mind
- They want to create Transformer-based models that can be applied to various computer vision tasks such as CNN



(a) Swin Transformer (ours)
(b) ViT

Key idea

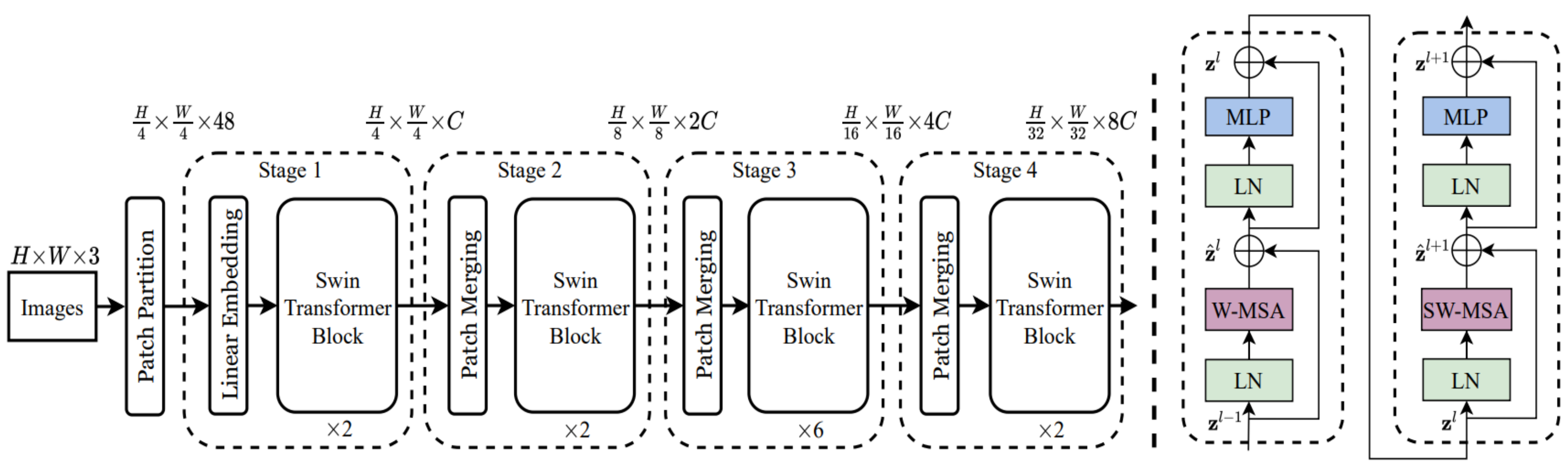
- Key idea revises Transformer structure to introduce shifted windows

Shifted window

- Shifted windows allows you to consider 2D data
- Shifted windows learns local patterns and allows them to be gradually integrated into global patterns

Overall architecture

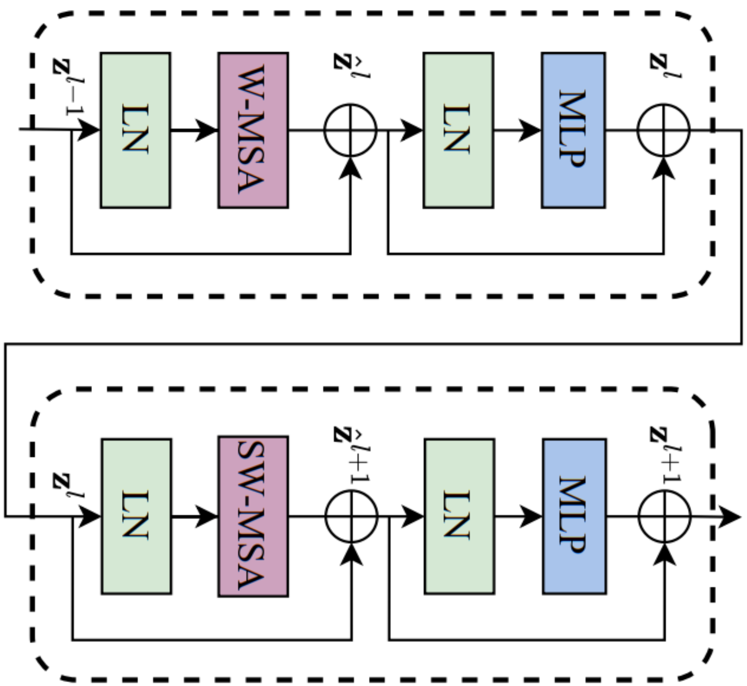
- The image is segmented into 4x4x3 patches via the patch partition layer, each of which is converted into an embedding vector.
- From Stage2, the number of channels is increased by merging four neighboring patches through Patch Merging.
- The method of increasing the number of channels in the Feature map allows it to be used as a backbone network in multiple tasks in a similar way to the CNN-based model.



(a) Architecture (b) Two Successive Swin Transformer Blocks

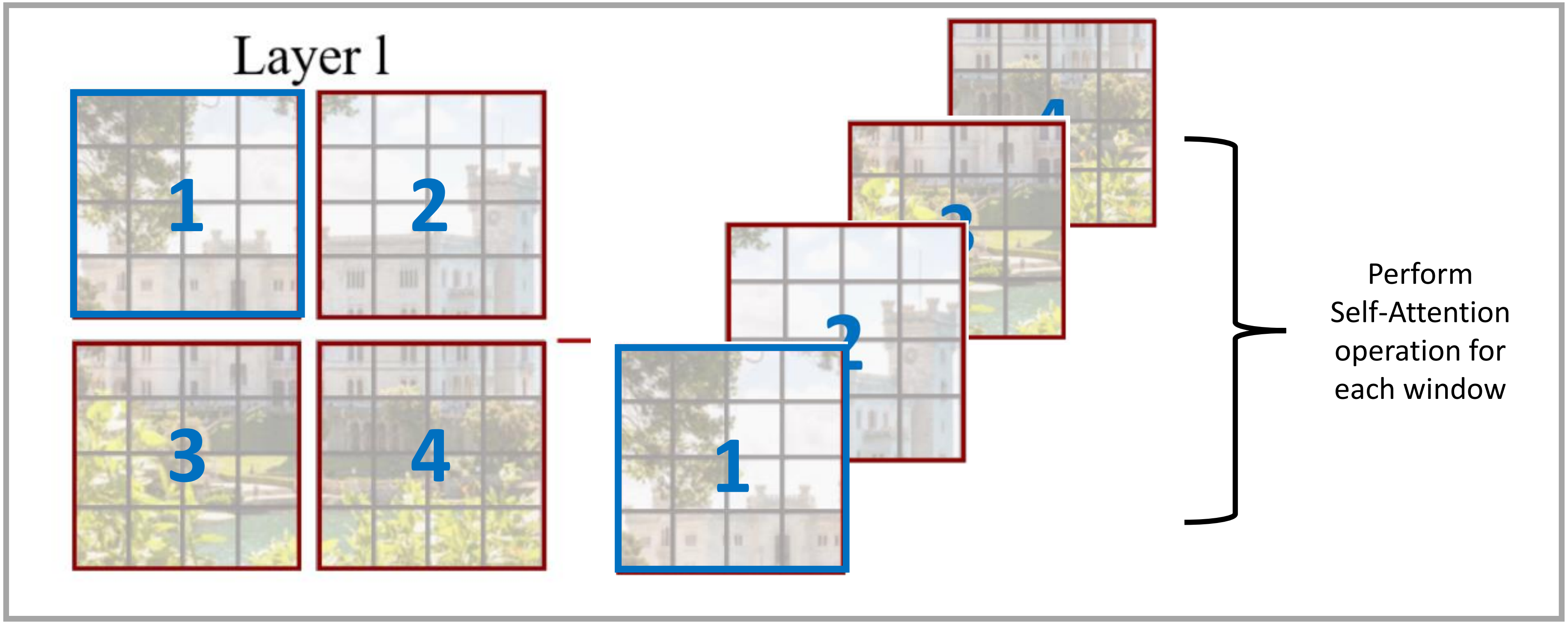
Swin Transformer block

- Swin Transformer block is configured by replacing the MSA layer with Windows-MSA layer and Shifted Windows-MSA layer
- Before entering each layer, LayerNorm layer was configured and MLP was configured as 2-layer



Window based Self-Attention (W-MSA)

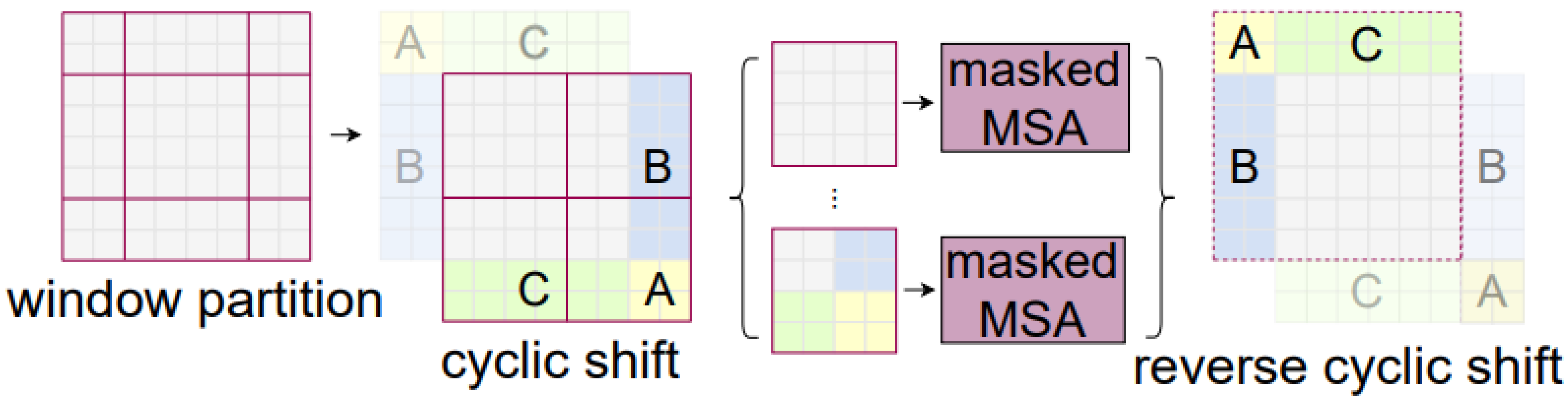
- W-MSA performs self-attention operations only on patches within Windows
- ViT performs a self-attention operation between all patches
- W-MSA enables learning of local characteristics of images



Shifted Window based Self-Attention (SW-MSA)

- The SW-MSA performs a window-based self-attention and then moves the window to the right and down 2 compartments to perform the W-MSA.
- When you move the window 2 spaces, use circular padding to adjust the window size in the window.
- SW-MSA method is introduced for connection between windows and between patches.

Circular padding method



Relative position bias

- Relative position bias is a matrix with relative position information between patches.
- The relative position bias is used because the position of the patch changes after the SW-MSA operation.
- The relative position bias plays the same role as the position embedding of ViT.

$$\text{Attention}(Q, K, V) = \text{SoftMax}(QK^T/\sqrt{d} + B)V,$$

Architecture Variants

- Swin-B is built to have similar model size and computational complexity to ViT-B/DeiT-B.
- Other Variants constructed a Swin-T with 0.25 times the parameter of Swin-B, a Swin-S with 0.5 times the parameter, and a Swin-L with 2 times the parameter of Swin-B.

- Swin-T: $C = 96$, layer numbers = $\{2, 2, 6, 2\}$
- Swin-S: $C = 96$, layer numbers =$\{2, 2, 18, 2\}$
- Swin-B: $C = 128$, layer numbers =$\{2, 2, 18, 2\}$
- Swin-L: $C = 192$, layer numbers =$\{2, 2, 18, 2\}$

Experiments

- Experiments evaluate ability as a backbone network in a variety of tasks to show goal achievement

- ImageNet-1k image classification
- COCO object detection
- ADE20K sementic segmentation
- Comparative experiments on the proposed method

Image classification on ImageNet-1K

- ImageNet-1K consists of 1,280,000 learning images and 50,000 validation images, with a total of 1,000 classes.
- In Regular Training, we outperform all models of similar size and achieve better speed-accuracy tradeoffs than CNNs.

(a) Regular ImageNet-1K trained models

method	image size	#param.	FLOPs	throughput (image / s)	ImageNet top-1 acc.
RegNetY-4G [44]	224^2	21M	4.0G	1156.7	80.0
RegNetY-8G [44]	224^2	39M	8.0G	591.6	81.7
RegNetY-16G [44]	224^2	84M	16.0G	334.7	82.9
ViT-B/16 [19]	384^2	86M	55.4G	85.9	77.9
ViT-L/16 [19]	384^2	307M	190.7G	27.3	76.5
DeiT-S [57]	224^2	22M	4.6G	940.4	79.8
DeiT-B [57]	224^2	86M	17.5G	292.3	81.8
DeiT-B [57]	384^2	86M	55.4G	85.9	83.1
Swin-T	224^2	29M	4.5G	755.2	81.3
Swin-S	224^2	50M	8.7G	436.9	83.0
Swin-B	224^2	88M	15.4G	278.1	83.5
Swin-B	384^2	88M	47.0G	84.7	84.5

(b) ImageNet-22K pre-trained models

method	image size	#param.	FLOPs	throughput (image / s)	ImageNet top-1 acc.
R-101x3 [34]	384^2	388M	204.6G	-	84.4
R-152x4 [34]	480^2	937M	840.5G	-	85.4
ViT-B/16 [19]	384^2	86M	55.4G	85.9	84.0
ViT-L/16 [19]	384^2	307M	190.7G	27.3	85.2
Swin-B	224^2	88M	15.4G	278.1	85.2
Swin-B	384^2	88M	47.0G	84.7	86.4
Swin-L	384^2	197M	103.9G	42.1	87.3

Object detection on COCO

- The object detection experiment is conducted by changing the backbone for each framework.
- Swin-T performs better than ResNet-50 and DeiT-S and ResNeXt101.

(a) Various frameworks

Method	Backbone	AP^{box}	AP^{box}_{50}	AP^{box}_{75}	#param.	FLOPs	FPS
Cascade Mask R-CNN	R-50	46.3	64.3	50.5	82M	739G	18.0
	Swin-T	**50.5**	**69.3**	**54.9**	86M	745G	15.3
ATSS	R-50	43.5	61.9	47.0	32M	205G	28.3
	Swin-T	**47.2**	**66.5**	**51.3**	36M	215G	22.3
RepPointsV2	R-50	46.5	64.6	50.3	42M	274G	13.6
	Swin-T	**50.0**	**68.5**	**54.2**	45M	283G	12.0
Sparse R-CNN	R-50	44.5	63.4	48.2	106M	166G	21.0
	Swin-T	**47.9**	**67.3**	**52.3**	110M	172G	18.4

(b) Various backbones w. Cascade Mask R-CNN

	AP^{box}	AP^{box}_{50}	AP^{box}_{75}	AP^{mask}	AP^{mask}_{50}	AP^{mask}_{75}	param	FLOPs	FPS
DeiT-S†	48.0	67.2	51.7	41.4	64.2	44.3	80M	889G	10.4
R50	46.3	64.3	50.5	40.1	61.7	43.4	82M	739G	18.0
Swin-T	**50.5**	**69.3**	**54.9**	**43.7**	**66.6**	**47.1**	86M	745G	15.3
X101-32	48.1	66.5	52.4	41.6	63.9	45.2	101M	819G	12.8
Swin-S	**51.8**	**70.4**	**56.3**	**44.7**	**67.9**	**48.5**	107M	838G	12.0
X101-64	48.3	66.4	52.3	41.7	64.0	45.1	140M	972G	10.4
Swin-B	**51.9**	**70.9**	**56.5**	**45.0**	**68.4**	**48.7**	145M	982G	11.6

Semantic Segmentation on ADE20K

- ADE20K [74] is a widely used semantic segmentation dataset containing a variety of 150 semantic categories.
- A comparative experiment is conducted by changing the backbone network to another framework.
- Swin Transformer's Variants perform well compared to similar-sized models.

ADE20K Method	Backbone	val mIoU	test score	#param.	FLOPs	FPS
DLab.v3+ [11]	ResNet-101	44.1	-	63M	1021G	16.0
DNL [65]	ResNet-101	46.0	56.2	69M	1249G	14.8
OCRNet [67]	ResNet-101	45.3	56.0	56M	923G	19.3
UperNet [63]	ResNet-101	44.9	-	86M	1029G	20.1
OCRNet [67]	HRNet-w48	45.7	-	71M	664G	12.5
DLab.v3+ [11]	ResNeSt-101	46.9	55.1	66M	1051G	11.9
DLab.v3+ [11]	ResNeSt-200	48.4	-	88M	1381G	8.1
SETR [73]	T-Large‡	50.3	61.7	308M	-	-
UperNet	DeiT-S†	44.0	-	52M	1099G	16.2
UperNet	Swin-T	46.1	-	60M	945G	18.5
UperNet	Swin-S	49.3	-	81M	1038G	15.2
UperNet	Swin-B‡	51.6	-	121M	1841G	8.7
UperNet	Swin-L‡	**53.5**	**62.8**	234M	3230G	6.2

Experiments on Shifted windows

- Experiments are conducted on the shifted window approach.
- The SW-MSA model also performed better than the W-MSA-only model.

	ImageNet		COCO		ADE20k
	top-1	top-5	AP^{box}	AP^{mask}	mIoU
w/o shifting	80.2	95.1	47.7	41.5	43.3
shifted windows	**81.3**	**95.6**	**50.5**	**43.7**	**46.1**

Experiments on Relative position bias

- It is an experiment that shows the comparison results of position embedding methods according to the results.
- Models using Relative position bias perform better than models without position encoding and models using position embedding.

	ImageNet		COCO		ADE20k
	top-1	top-5	AP^{box}	AP^{mask}	mIoU
w/o shifting	80.2	95.1	47.7	41.5	43.3
shifted windows	**81.3**	**95.6**	**50.5**	**43.7**	**46.1**
no pos.	80.1	94.9	49.2	42.6	43.8
abs. pos.	80.5	95.2	49.0	42.4	43.2
abs.+rel. pos.	81.3	95.6	50.2	43.4	44.0
rel. pos. w/o app.	79.3	94.7	48.2	41.9	44.1
rel. pos.	**81.3**	**95.6**	**50.5**	**43.7**	**46.1**

Experiments on Different self-attention methods

- It is an experiment comparing with various self-attention methods.
- Circular padding performs better than naive padding.
- The proposed SW-MSA shows that it is a more efficient model than the sliding window method.

method	MSA in a stage (ms)				Arch. (FPS)		
	S1	S2	S3	S4	T	S	B
sliding window (naive)	122.5	38.3	12.1	7.6	183	109	77
sliding window (kernel)	7.6	4.7	2.7	1.8	488	283	187
Performer [14]	4.8	2.8	1.8	1.5	638	370	241
window (w/o shifting)	2.8	1.7	1.2	0.9	770	444	280
shifted window (padding)	3.3	2.3	1.9	2.2	670	371	236
shifted window (cyclic)	3.0	1.9	1.3	1.0	755	437	278

Table 5. Real speed of different self-attention computation methods and implementations on a V100 GPU.

	Backbone	ImageNet top-1	ImageNet top-5	COCO AP^{box}	COCO AP^{mask}	ADE20k mIoU
sliding window	Swin-T	81.4	95.6	50.2	43.5	45.8
Performer [14]	Swin-T	79.0	94.2	-	-	-
shifted window	Swin-T	81.3	95.6	50.5	43.7	46.1

Table 6. Accuracy of Swin Transformer using different methods for self-attention computation on three benchmarks.

- MSA in stage(ms): Running time for MSA modules performed at each stage (step)
- Sliding window: To process an image by dividing it into fixed-sized windows so that it does not overlap
- Performer: Transformer model using kernelized attention.

Conclusion

- The Swin Transformer enables the creation of hierarchical characteristic representations.
- Swin Transformer reduces the computational complexity of ViT
- We propose a Transformer-based backbone network that can act like a CNN.
- Achieve the best performance in a variety of tasks and show the potential of Transformer-based models in Computer vision.