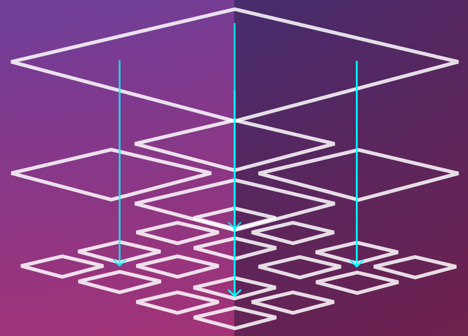


Swin Transformer

Hierarchical Vision Transformer using Shifted Windows

Su Hyung Choi

Korea University

Research Intern @ Computer Vision Lab

Dept. of Computer Science & Engineering

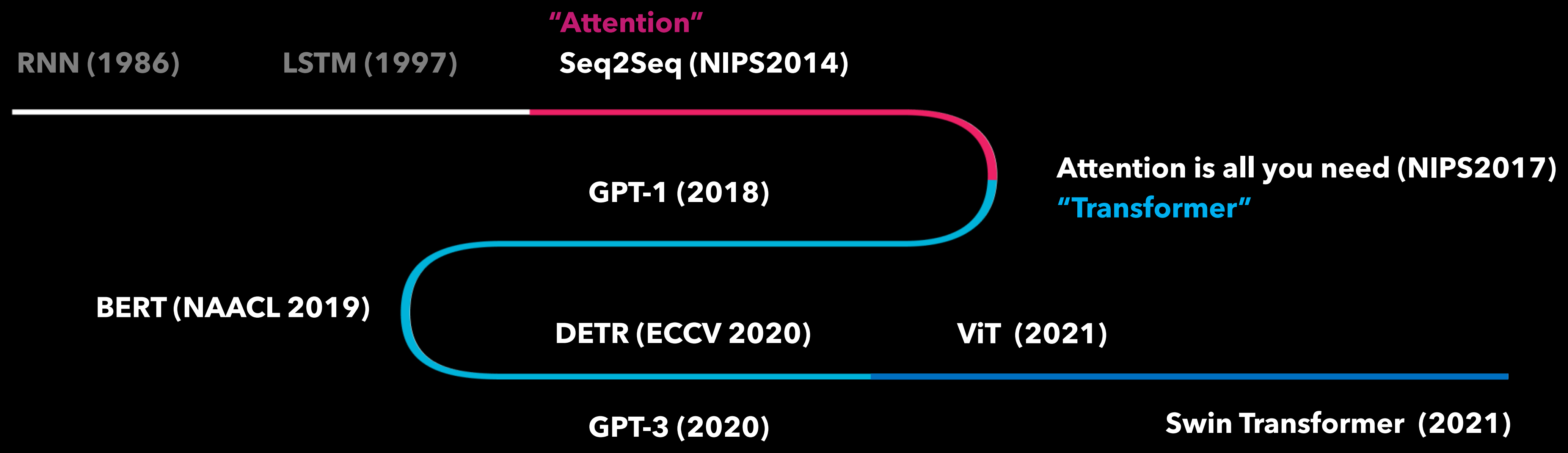
Submitted on Jan 6, 2022

Authors {v-zeliu1,v-yutlin,yuecao,hanhu,v-yixwe,zhez,stevelin,bainguo}@microsoft.com

Computer Vision Laboratory

Paper Review

Attention is now what the real SOTA needs.



Swin Transformer (2021 ICCV Award)

Image Classification: 90.17 top-6 accuracy on ImageNet-1K (SwinV2-G) 2022.1.6 Present

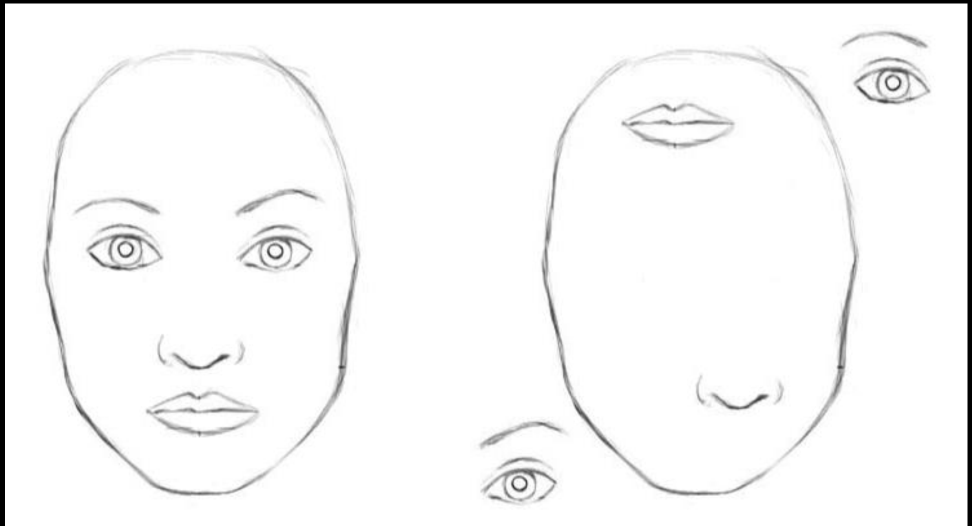
Object Detection: 63.1 top-1 box AP on COCO test-dev (SwinV2-G) 2022.1.6 Present

Semantic Segmentation: 59.9 top-1 mlouU on ADE20K (SwinV2-G) 2022.1.6 Present

Why Transformer Anyway?

"Everyone wants a universal model to solve different tasks with accuracy and speed" → JUST LIKE MLP !!

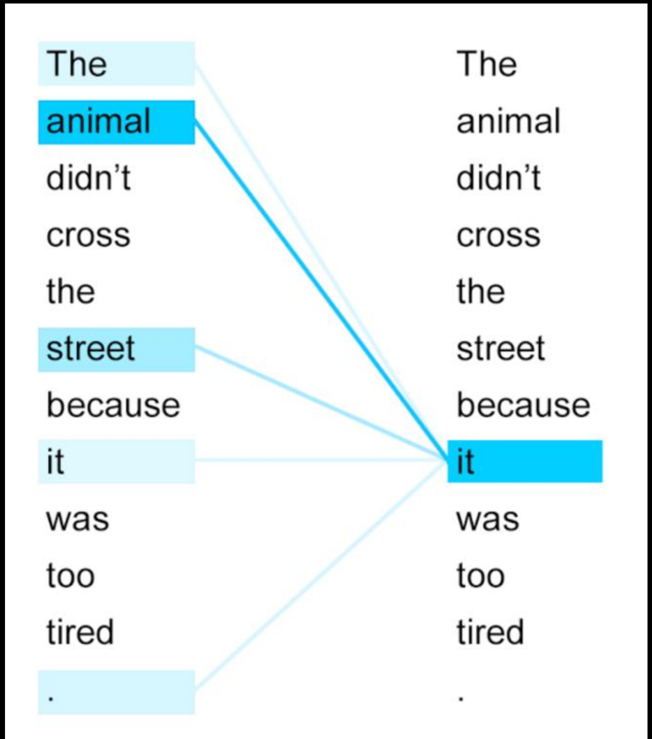
→ Transformer models are universal approximators of sequence-to-sequence functions.



For a CNN, both of these pictures are almost same. CNN does not encode the relative position of different features.

Large receptive fields are required in order to track **long-range dependencies** within an image.

[Self-Attention] The Core of the Transformer



Self-Attention on NLP

Attention between Patches on ViT

Basically, a self-attention layer updates **each component** of a sequence by **aggregating global information from the complete input sequence.**

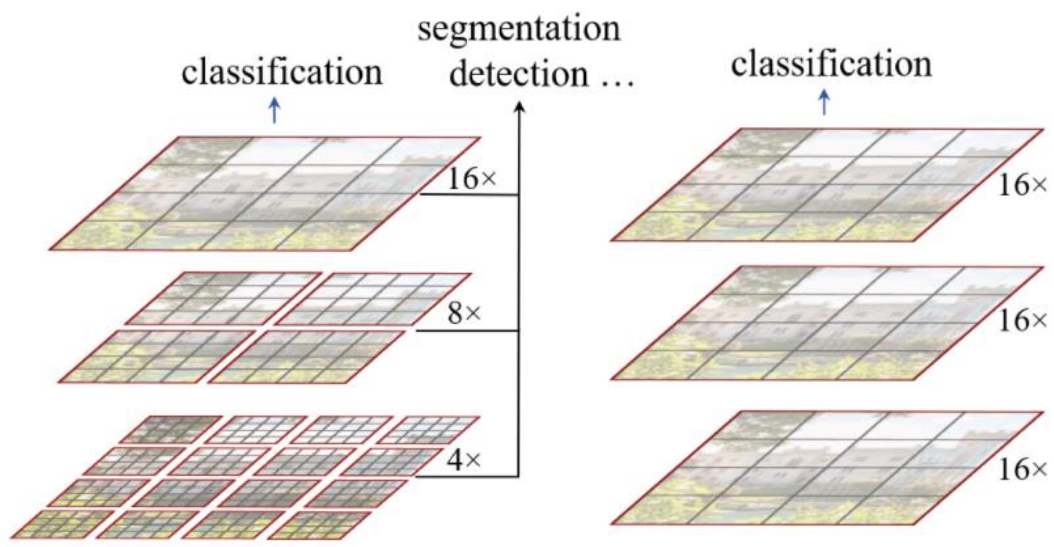
The Problem of Vision Transformer

Previous Problem

- ViT suggested for classification
- No Well - Featured for Image compared to Text
  - Resolution
  - The scale of visual entities
- Computation Complexity rises as increases of Tokens

Solution

Adapt Local Window to Model



(a) Swin Transformer (ours)

(b) ViT

The Problem of Vision Transformer

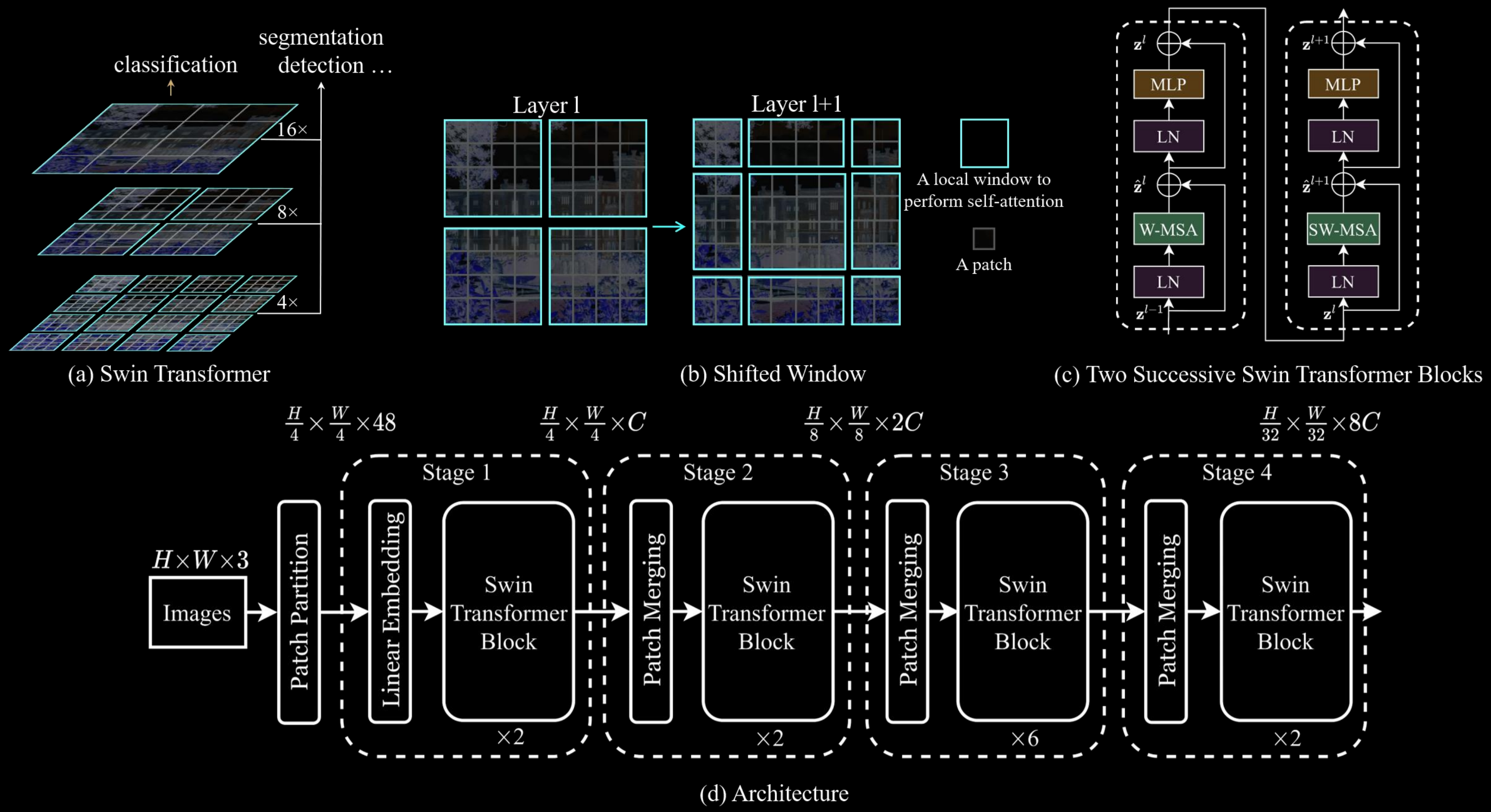
Computation Complexity rises as increases of Tokens

Swin Transformer

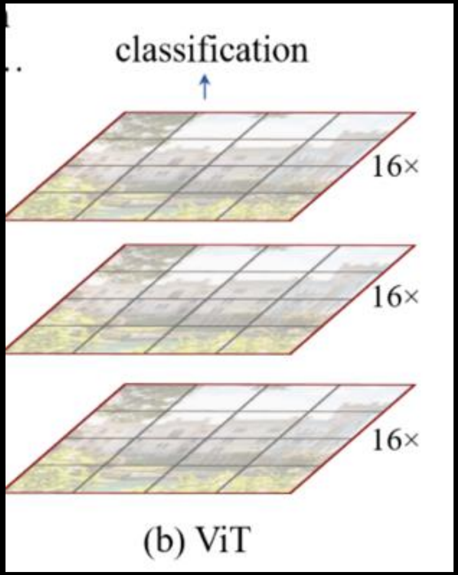
Purpose

- Suggest the BackBone for Multi Tasks
- Suggest the method of Transformer to adapt the Images
- Suggest the method having less computation than ViT

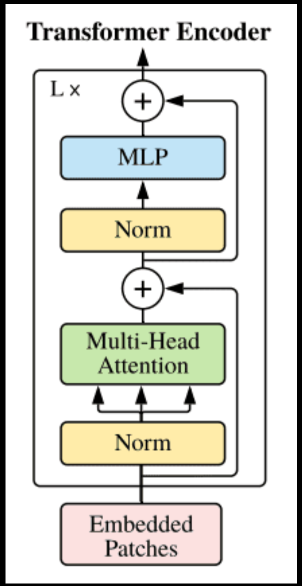
Vision Transformer vs Swin Transformer



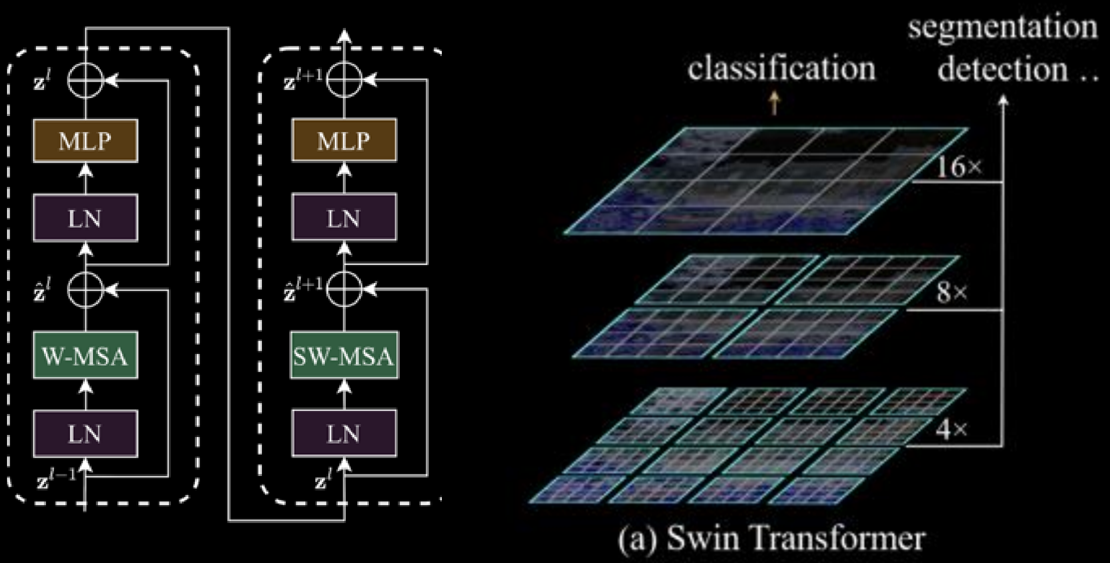
Vision Transformer vs Swin Transformer



(b) ViT



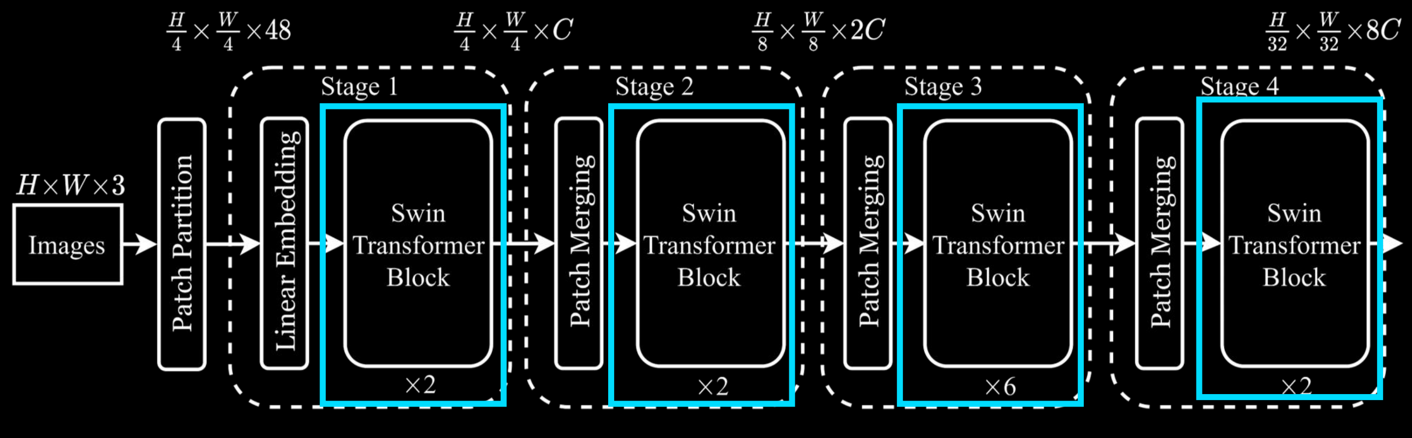
$$\Omega(\text{MSA}) = 4hwC^2 + 2(hw)^2C$$



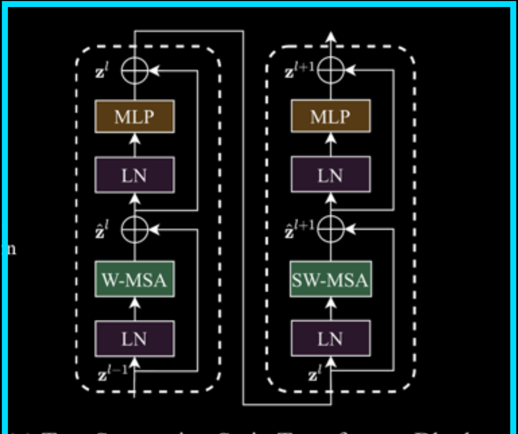
(a) Swin Transformer

$$\Omega(\text{W-MSA}) = 4hwC^2 + 2M^2hwC$$

Swin Transformer Architecture



(d) Architecture

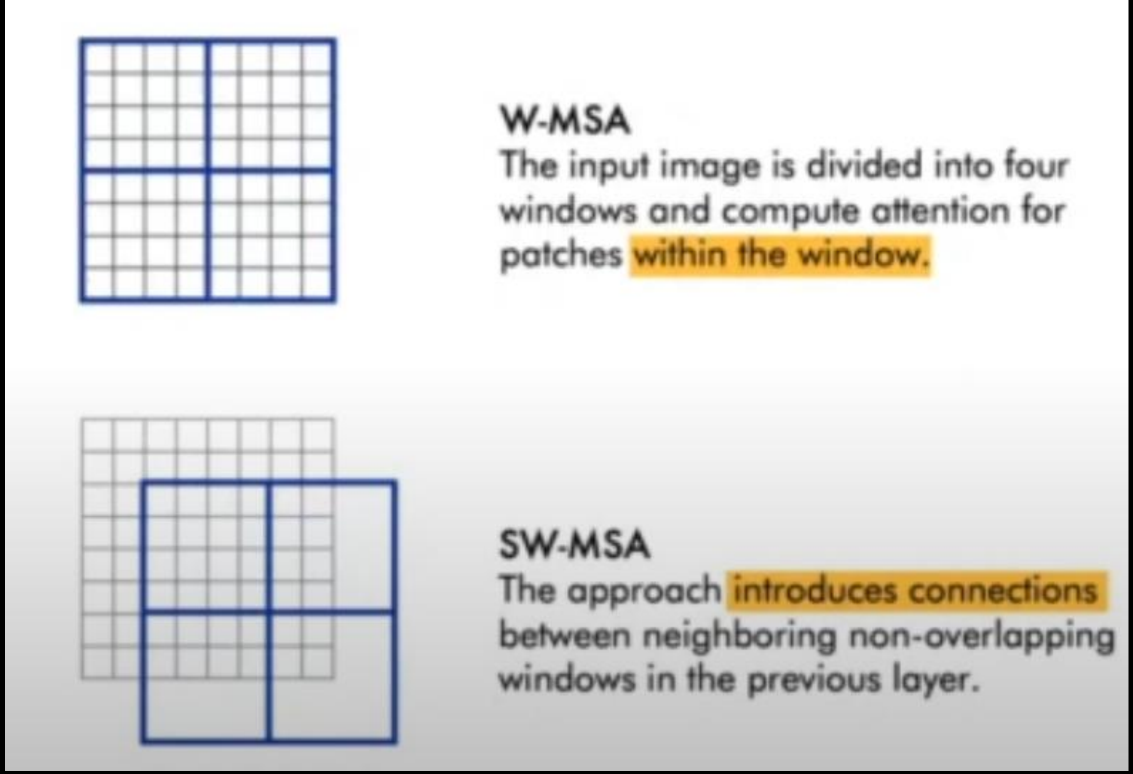


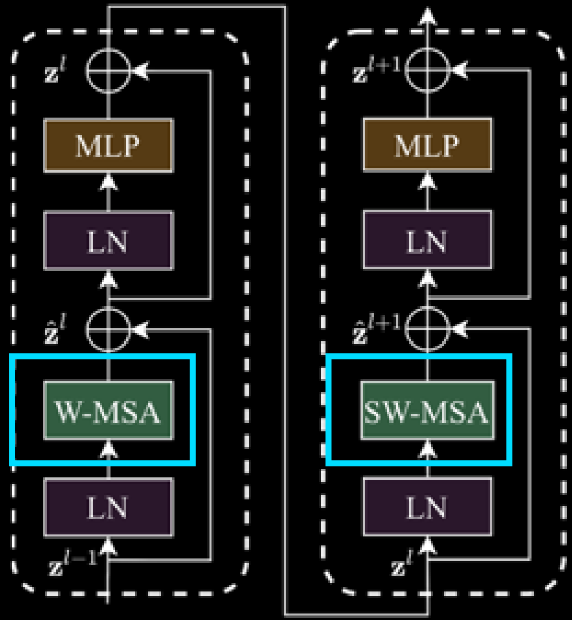
(c) Two Successive Swin Transformer Blocks

W-MSA (Window Multi-head Self Attention)

SW-MSA (Shifted Window Multi-head self Attention)

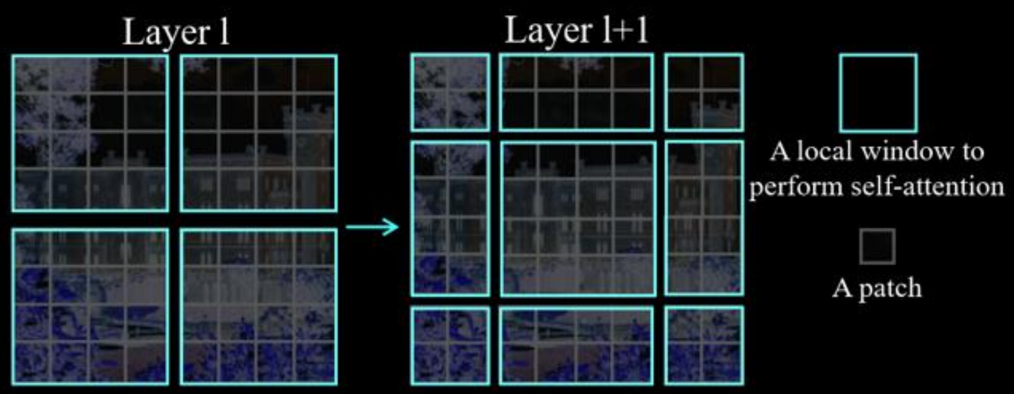
Swin Transformer Block





(c) Two Successive Swin Transformer Blocks

Shifted Window based Self-Attention



(b) Shifted Window

Self-attention is applied on each patch, here referred to as windows. In layer 1 (left), a regular window partitioning shceme is adapted, and self-attention within each window.

Then, the windows are shifted, resulting in a new window configuration to apply self attention again. This allows the creation of connections between windows while maintaining the computation efficiency of this windowed architecture.

Cyclic-Shift

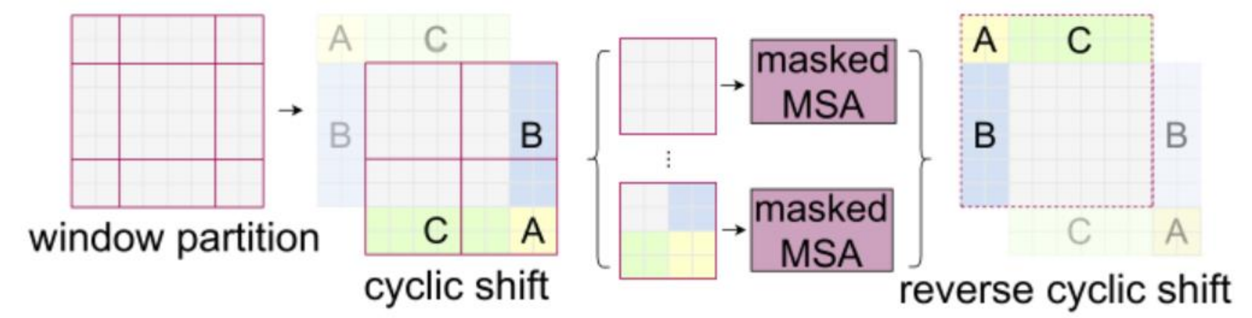


Figure 4: Illustration of an efficient batch computation approach for self-attention in shifted window partitioning.

The paper propose an efficient batch computation approach by cyclic-shifting toward the top-left direction.

After this shift, a batched window may be composed of several sub-windows that are not adjacent in the feature map, so a masking mechanism is employed to limit self-attention computation to within each sub-window.

Experiments

Setting

- Experiment with ImageNet 1K for Classification
- Used ImageNet-22K for pre-train
- Experiment with COCO 2017 for Object Detection
- Experiment with ADE20K for Semantic Segmentation

- Required various Augmentation and Regularization -> INDUCTIVE BIAS
- EMA and repeated augmentation was not required compared at Vit
- Used AdamW optimizer

Experiments

Vision Tasks

- Classification
- Object Detection
- Semantic Segmentation

Others

- Relative Positional Bias
- Shifted Window
- Speed Comparison for different Window Adaption
- Comparison between Sliding Window and Shifted Window

Experiments

Baseline Models

- **CNN based Models**
  - RegNetY
  - EfficientNet

- **ViT based Models**
  - Vit
  - DeiT

Experiment [Classification]

(a) Regular ImageNet-1K trained models

method	image size	#param.	FLOPs	throughput (image / s)	ImageNet top-1 acc.
RegNetY-4G [48]	224^2	21M	4.0G	1156.7	80.0
RegNetY-8G [48]	224^2	39M	8.0G	591.6	81.7
RegNetY-16G [48]	224^2	84M	16.0G	334.7	82.9
EffNet-B3 [58]	300^2	12M	1.8G	732.1	81.6
EffNet-B4 [58]	380^2	19M	4.2G	349.4	82.9
EffNet-B5 [58]	456^2	30M	9.9G	169.1	83.6
EffNet-B6 [58]	528^2	43M	19.0G	96.9	84.0
EffNet-B7 [58]	600^2	66M	37.0G	55.1	84.3
ViT-B/16 [20]	384^2	86M	55.4G	85.9	77.9
ViT-L/16 [20]	384^2	307M	190.7G	27.3	76.5
DeiT-S [63]	224^2	22M	4.6G	940.4	79.8
DeiT-B [63]	224^2	86M	17.5G	292.3	81.8
DeiT-B [63]	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 [38]	384^2	388M	204.6G	-	84.4
R-152x4 [38]	480^2	937M	840.5G	-	85.4
ViT-B/16 [20]	384^2	86M	55.4G	85.9	84.0
ViT-L/16 [20]	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

Table 1. Comparison of different backbones on ImageNet-1K classification. Throughput is measured using the GitHub repository of [68] and a V100 GPU, following [63].

ImageNet 1K

- DeiT vs Swin
- CNN based Models vs Swin
  - Less Trade-off between Performance and Training Time

ImageNet 22K

- ViT vs Swin
  - Better Performance with less Params

Rank	Model	Top 1 Accuracy	Top 5 Accuracy	Number of params	Extra Training Data	Paper	Code	Result	Year	Tags
1	**CoAtNet-7**	90.88%		2440M	✓	CoAtNet: Marrying Convolution and Attention for All Data Sizes			2021	CNN Conv+Transformer JFT-3B
2	**ViT-G/14**	90.45%		1843M	✓	Scaling Vision Transformers			2021	Transformer JFT-3B
3	**CoAtNet-6**	90.45%		1470M	✓	CoAtNet: Marrying Convolution and Attention for All Data Sizes			2021	Conv+Transformer JFT-3B
4	**ViT-MoE-15B** (Every-2)	90.35%		14700M	✓	Scaling Vision with Sparse Mixture of Experts			2021	Transformer JFT-3B
5	**Meta Pseudo Labels** (EfficientNet-L2)	90.2%	98.8%	480M	✓	Meta Pseudo Labels			2021	EfficientNet JFT-300M
6	**SwinV2-G**	90.17%			✓	Swin Transformer V2: Scaling Up Capacity and Resolution			2021	

Experiment
[Object Detection]

Based on Frameworks

- ResNet-50 vs Swin
  - + 3.4 ~ 4.2 box AP

Based on Backbone

- CNN based Model vs Swin
- **Dominating Leaderboard (+ SwinV2) – 2022.1.6 Present [COCO – test Dev]**

Rank	Model	box AP	AP50	AP75	APS	APM	APL	Extra Training Data	Paper	Code	Result	Year	Tags
1	SwinV2-G (HTC++)	63.1						✓	Swin Transformer V2: Scaling Up Capacity and Resolution			2021	Swin-Transformer
2	Florence-CoSwin-H	62.4						✓	Florence: A New Foundation Model for Computer Vision			2021	Swin-Transformer
3	GLIP (Swin-L, multi-scale)	61.5	79.5	67.7	45.3	64.9	75.0	✓	Grounded Language-Image Pre-training			2021	multiscale, Vision Language, Dynamic Head, BERT-Base

(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

(c) System-level Comparison

Method	mini-val AP^{box}	mini-val AP^{mask}	test-dev AP^{box}	test-dev AP^{mask}	#param.	FLOPs
RepPointsV2* [12]	-	-	52.1	-	-	-
GCNet* [7]	51.8	44.7	52.3	45.4	-	1041G
RelationNet++* [13]	-	-	52.7	-	-	-
SpineNet-190 [21]	52.6	-	52.8	-	164M	1885G
ResNeSt-200* [78]	52.5	-	53.3	47.1	-	-
EfficientDet-D7 [59]	54.4	-	55.1	-	77M	410G
DetectoRS* [46]	-	-	55.7	48.5	-	-
YOLOv4 P7* [4]	-	-	55.8	-	-	-
Copy-paste [26]	55.9	47.2	56.0	47.4	185M	1440G
X101-64 (HTC++)	52.3	46.0	-	-	155M	1033G
Swin-B (HTC++)	56.4	49.1	-	-	160M	1043G
Swin-L (HTC++)	57.1	49.5	57.7	50.2	284M	1470G
Swin-L (HTC++)*	**58.0**	**50.4**	**58.7**	**51.1**	284M	-

Table 2. Results on COCO object detection and instance segmentation. †denotes that additional decovolution layers are used to produce hierarchical feature maps. * indicates multi-scale testing.

Experiment
[Semantic Segmentation]

ADE20K Method	Backbone	val mIoU	test score	#param.	FLOPs	FPS
DANet [23]	ResNet-101	45.2	-	69M	1119G	15.2
DLab.v3+ [11]	ResNet-101	44.1	-	63M	1021G	16.0
ACNet [24]	ResNet-101	45.9	38.5	-		
DNL [71]	ResNet-101	46.0	56.2	69M	1249G	14.8
OCRNet [73]	ResNet-101	45.3	56.0	56M	923G	19.3
UperNet [69]	ResNet-101	44.9	-	86M	1029G	20.1
OCRNet [73]	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 [81]	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

Table 3. Results of semantic segmentation on the ADE20K val and test set. † indicates additional deconvolution layers are used to produce hierarchical feature maps. ‡ indicates that the model is pre-trained on ImageNet-22K.

SETR + T-Large vs UperNet + Swin-L

- Better Performance with LESS PARAMS

ALL MOST DOMINAT LEADERBOARD on ADE20K Datasets (1-4 and 6-7) – 2022.1.6 Present

Rank	Model	Validation mIoU	Test Score	Extra Training Data	Paper	Code	Result	Year	Tags
1	**SwinV2-G** (UperNet)	59.9		✓	Swin Transformer V2: Scaling Up Capacity and Resolution			2021	Swin-Transformer
2	**SeMask** (SeMask Swin-L MSFaPN-Mask2Former)	58.2		✓	SeMask: Semantically Masked Transformers for Semantic Segmentation			2021	Swin-Transformer
3	**SeMask** (SeMask Swin-L FaPN-Mask2Former)	58.0		✓	SeMask: Semantically Masked Transformers for Semantic Segmentation			2021	Swin-Transformer

Simple Code Review

GitHub

[@JonyChoi](#)

[Computer-Vision-Paper-Reviews](#)

Conclusion

- This paper presents Swin Transformer, a new vision Transformer which produces a hierarchical feature representation and has linear computational complexity with respect to input image size.

- As a key element of Swin Transformer, the shifted window self-attention is shown to be effective and efficient on vision problems.

- Using a similar architecture for both NLP and computer vision could significantly accelerate the research process.

Thank You

Su Hyung Choi
CVLAB Intern