Swin Transformer: Hierarchical Vision Transformer using Shifted Windows

건국대학교
컴퓨터공학과

김동환

2024.07.08

목 차

Swin Transformer: Hierarchical Vision Transformer using Shifted Windows

0. Abstract

배경

- NLP에서 성공을 거둔 Transformer를 CV에 도입
- ViT는 이미지 변동성, 큰 해상도를 처리하지 못함

  -> 중간 해상도, Image Classification에만 사용가능

목적

- 큰 스케일을 처리하고, 다양한 CV tasks 적용가능한 ViT

Swin Transformer: Hierarchical Vision Transformer using Shifted Windows

1. Introduction

Swin Transformer

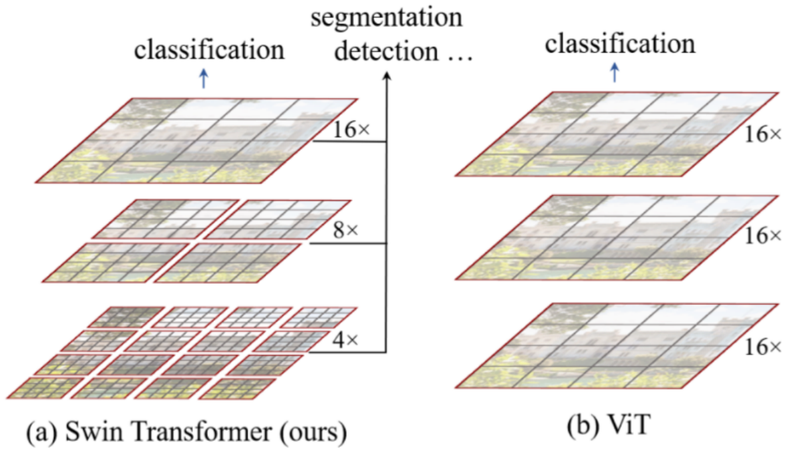


Figure 1. (a) The proposed Swin Transformer builds hierarchical feature maps by merging image patches (shown in gray) in deeper layers and has linear computation complexity to input image size due to computation of self-attention only within each local window (shown in red). It can thus serve as a general-purpose backbone for both image classification and dense recognition tasks. (b) In contrast, previous vision Transformers [20] produce feature maps of a single low resolution and have quadratic computation complexity to input image size due to computation of self-attention globally.

연구 개요

ViT의 O(N^2) 계산 복잡도를 개선하기 위하여
계층적 feature maps을 이용해

O(N)의 계산 복잡도를 가진
Swin Transformer를 제작

Shifted Windows Transformer

Swin Transformer: Hierarchical Vision Transformer using Shifted Windows

1. Introduction

Swin Transformer

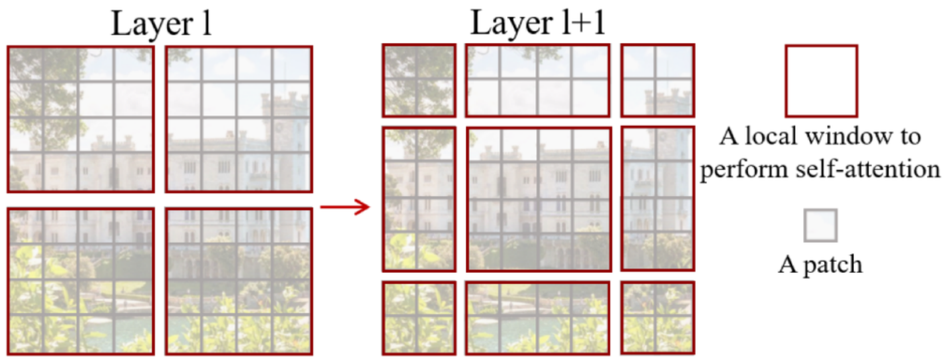


Figure 2. An illustration of the *shifted window* approach for computing self-attention in the proposed Swin Transformer architecture. In layer l (left), a regular window partitioning scheme is adopted, and self-attention is computed within each window. In the next layer $l + 1$ (right), the window partitioning is shifted, resulting in new windows. The self-attention computation in the new windows crosses the boundaries of the previous windows in layer l, providing connections among them.

연구 개요

Window 내에서만
Self-Attention을 계산하여 연산량을 줄이고

Window Shift 를 통해
Window 간 Connection을 계산

Swin Transformer: Hierarchical Vision Transformer using Shifted Windows

2. Related Work

CNN and variants

CV에서 CNN 기반 아키텍쳐는 표준이 되어 왔음

MS Team은 NLP, CV 공통으로 적용가능한 모델에 주목

Self-attention based backbone architectures

ResNet의 CNN layer를 Self-Attention layer로 대체 하는 연구

약간 향상된 성능을 보이나 매우 큰 computation cost

Self-attention/Transformers to complement CNNs

CNN 기반 모델에 Self-Attention layer를 추가

Transformer based vision backbones

구글에서 발표한 ViT

CNN 대비 훌륭한 성능을 보이나, image classification에만 적용가능하고 대규모 dataset 필요

-> O(N^2)의 계산 복잡도 때문

Swin Transformer: Hierarchical Vision Transformer using Shifted Windows

3. Method

Swin Transformer

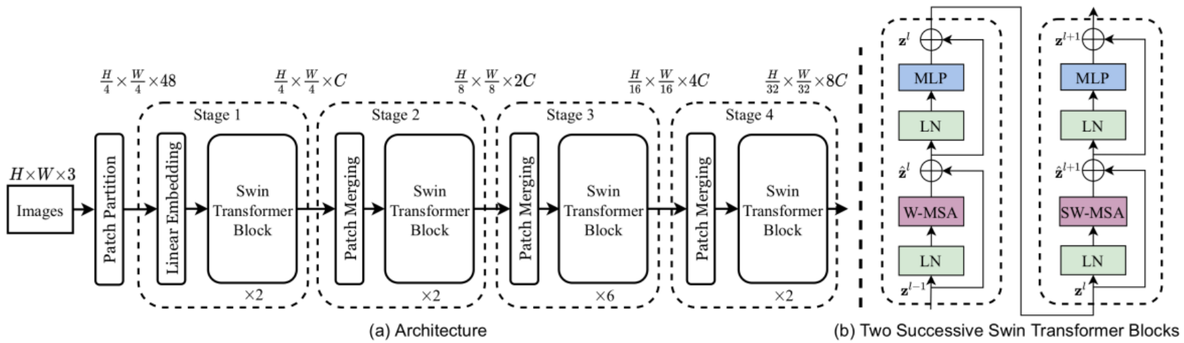


Figure 3. (a) The architecture of a Swin Transformer (Swin-T); (b) two successive Swin Transformer Blocks (notation presented with Eq. (3)). W-MSA and SW-MSA are multi-head self attention modules with regular and shifted windowing configurations, respectively.

ViT의 아키텍쳐

Transformer를 변형한 Swin Transformer Block을 사용

Swin Transformer: Hierarchical Vision Transformer using Shifted Windows

3.1 Overall Architecture

Swin Transformer

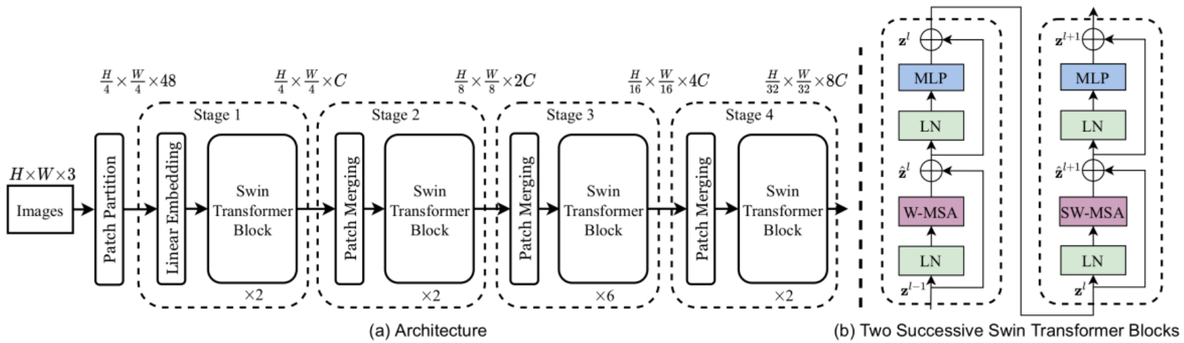


Figure 3. (a) The architecture of a Swin Transformer (Swin-T); (b) two successive Swin Transformer Blocks (notation presented with Eq. (3)). W-MSA and SW-MSA are multi-head self attention modules with regular and shifted windowing configurations, respectively.

Patch Projection

RGB image를 patch로 나누는
Patch Projection 수행

Swin Transformer: Hierarchical Vision Transformer using Shifted Windows

3.1 Overall Architecture

Swin Transformer

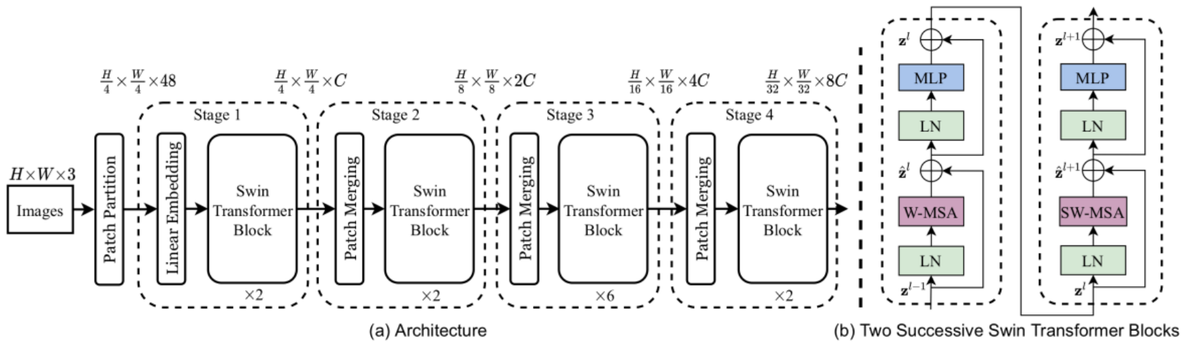


Figure 3. (a) The architecture of a Swin Transformer (Swin-T); (b) two successive Swin Transformer Blocks (notation presented with Eq. (3)). W-MSA and SW-MSA are multi-head self attention modules with regular and shifted windowing configurations, respectively.

Stage 1

Linear Embedding을 적용하여
임의 차원 C로 Projection

Swin Transformer block을 적용하여
Self-Attention 계산

Swin Transformer: Hierarchical Vision Transformer using Shifted Windows

3.1 Overall Architecture

Swin Transformer

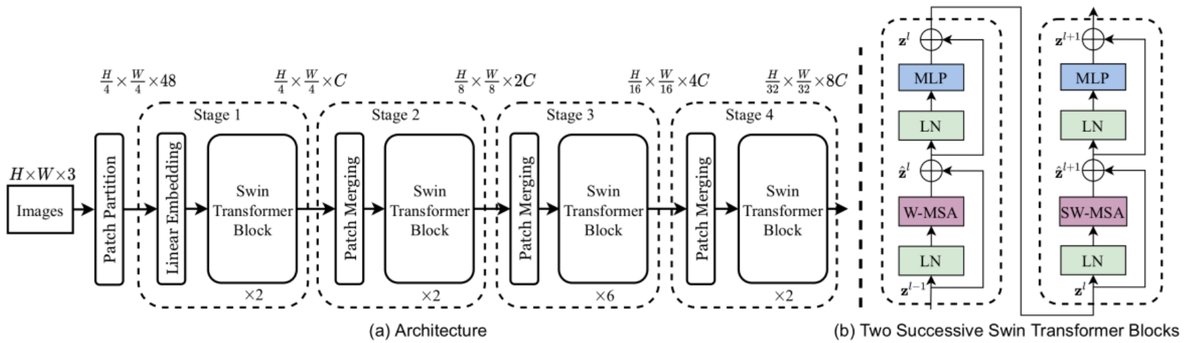


Figure 3. (a) The architecture of a Swin Transformer (Swin-T); (b) two successive Swin Transformer Blocks (notation presented with Eq. (3)). W-MSA and SW-MSA are multi-head self attention modules with regular and shifted windowing configurations, respectively.

Stage 2,3,4

Patch Merging을 적용하여
인접 patch merge -> token 수 4배 감소, output 차원 2배 증가

Swin Transformer block을 적용하여
Self-Attention 계산

Swin Transformer: Hierarchical Vision Transformer using Shifted Windows

3.1 Overall Architecture

Swin Transformer

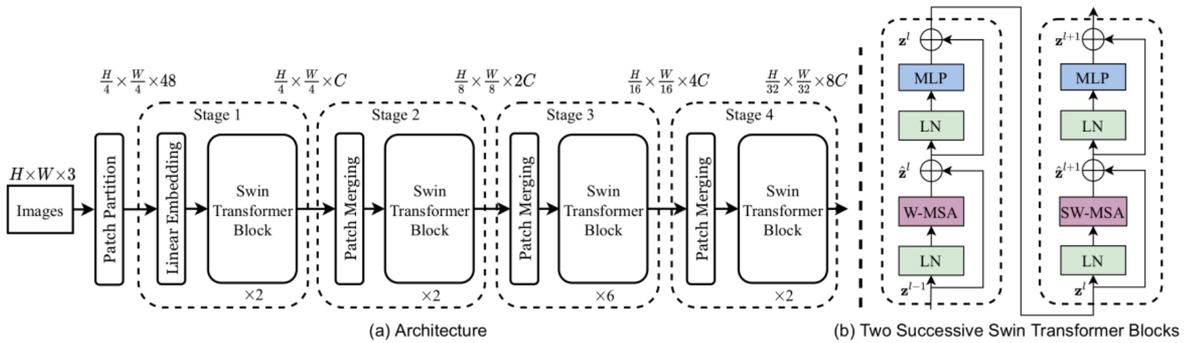


Figure 3. (a) The architecture of a Swin Transformer (Swin-T); (b) two successive Swin Transformer Blocks (notation presented with Eq. (3)). W-MSA and SW-MSA are multi-head self attention modules with regular and shifted windowing configurations, respectively.

Swin Transformer Block

기존 Transformer의 MSA를
shifted window 기반 MSA로 바꿈

W-MSA : Window based MSA
SW-MSA : Shifted Window based MSA

Swin Transformer: Hierarchical Vision Transformer using Shifted Windows

3.2 Shifted Window based Self-Attention

Shifted Window based Self-Attention

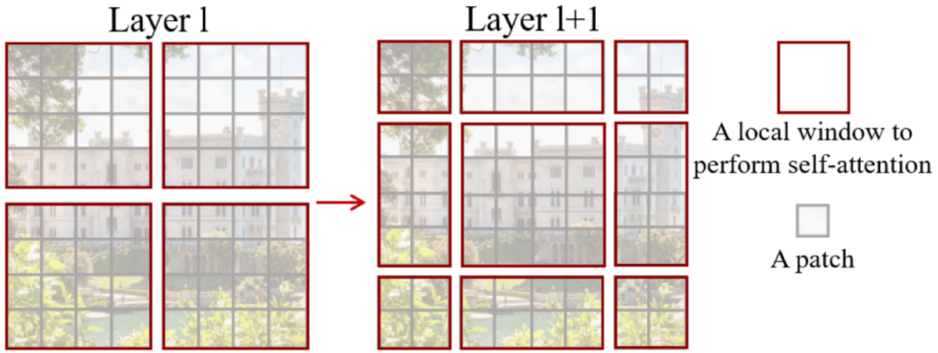


Figure 2. An illustration of the *shifted window* approach for computing self-attention in the proposed Swin Transformer architecture. In layer l (left), a regular window partitioning scheme is adopted, and self-attention is computed within each window. In the next layer $l+1$ (right), the window partitioning is shifted, resulting in new windows. The self-attention computation in the new windows crosses the boundaries of the previous windows in layer l, providing connections among them.

local한 Self-Attention 계산

$$\Omega(\text{MSA}) = 4hwC^2 + 2(hw)^2C, \quad (1)$$

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

패치의 수인 hw에 선형의 계산 복잡도

Window 간 connection 도입하기 위하여
Shifted Window Partitioning 이용

이전 레이어에서 규칙적으로 분할된 윈도우를
(⌊M/2⌋, ⌊M/2⌋) 픽셀만큼 이동시켜 윈도우 구성 방식을 변경

Swin Transformer: Hierarchical Vision Transformer using Shifted Windows

3.2 Shifted Window based Self-Attention

Swin Transformer

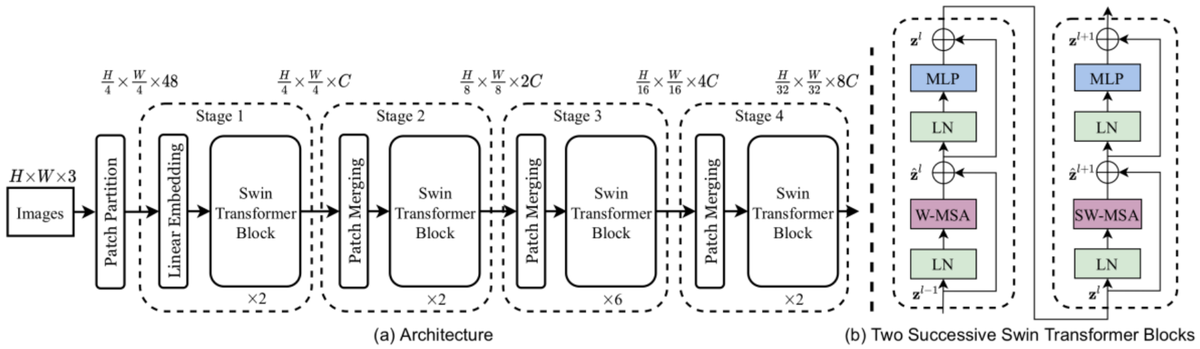


Figure 3. (a) The architecture of a Swin Transformer (Swin-T); (b) two successive Swin Transformer Blocks (notation presented with Eq. (3)). W-MSA and SW-MSA are multi-head self attention modules with regular and shifted windowing configurations, respectively.

Swin의 동작 방식

$$\hat{\mathbf{z}}^l = \text{W-MSA}\left(\text{LN}\left(\mathbf{z}^{l-1}\right)\right) + \mathbf{z}^{l-1},$$
$$\mathbf{z}^l = \text{MLP}\left(\text{LN}\left(\hat{\mathbf{z}}^l\right)\right) + \hat{\mathbf{z}}^l,$$
$$\hat{\mathbf{z}}^{l+1} = \text{SW-MSA}\left(\text{LN}\left(\mathbf{z}^l\right)\right) + \mathbf{z}^l,$$
$$\mathbf{z}^{l+1} = \text{MLP}\left(\text{LN}\left(\hat{\mathbf{z}}^{l+1}\right)\right) + \hat{\mathbf{z}}^{l+1},$$

MSA, MLP를 번갈아가며 수행
LN과 residual connection은 지속적으로 계산

Swin Transformer: Hierarchical Vision Transformer using Shifted Windows

Efficient batch computation for shifted configuration

shifed window partitiong에서 주된 문제는
window 개수가 증가한다는 것

Shifted Window Partition

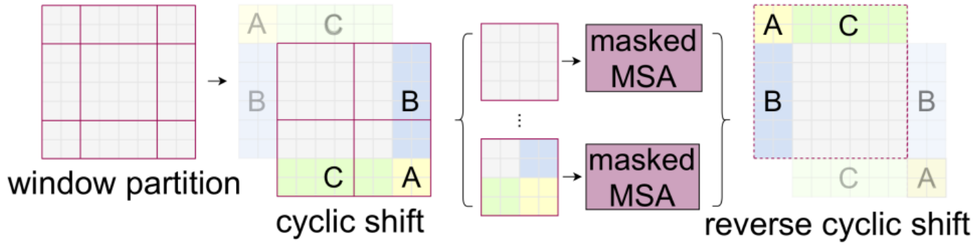


Figure 4. Illustration of an efficient batch computation approach for self-attention in shifted window partitioning.

윈도우 개수가 증가

원래 구성에서는 ⌈H/M⌉×⌈W/M⌉ 개의 window가 필요
Shifted 구성에서는 (⌈HM⌉+1)×(⌈WM⌉+1) 윈도우가 필요

Swin Transformer: Hierarchical Vision Transformer using Shifted Windows

Efficient batch computation for shifted configuration

shifed window partitiong에서 주된 문제는
window 개수가 증가한다는 것

Shifted Window Partition

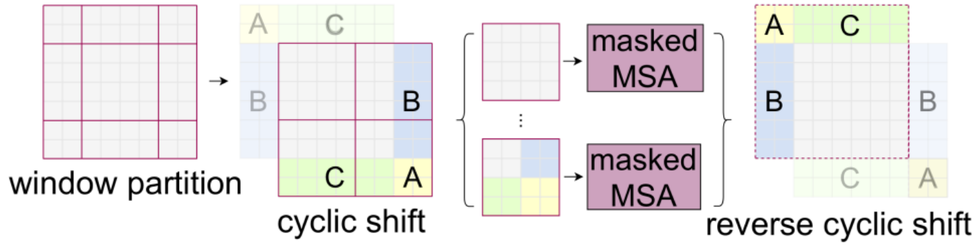


Figure 4. Illustration of an efficient batch computation approach for self-attention in shifted window partitioning.

cyclic-shifting 방식

masking 기법을 활용하여
3 x 3 window가 아닌 2 x 2 window 처럼 처리가능

Swin Transformer: Hierarchical Vision Transformer using Shifted Windows

Efficient batch computation for shifted configuration

shifed window partitiong에서 주된 문제는
window 개수가 증가한다는 것

Shifted Window Partition

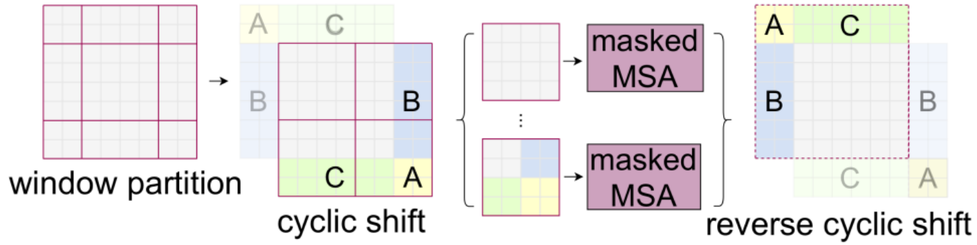
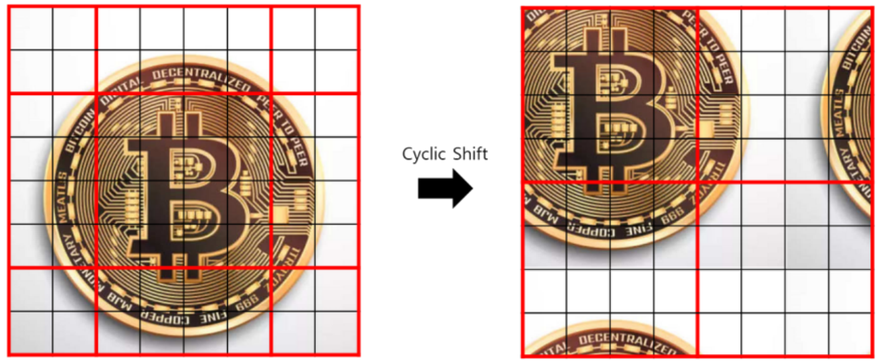


Figure 4. Illustration of an efficient batch computation approach for self-attention in shifted window partitioning.

cyclic-shifting 방식



Swin Transformer: Hierarchical Vision Transformer using Shifted Windows

Relative position bias

상대 위치 편향 도입

Relative position bias

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

Attention 계산에
Relative Position Bias인 B를 도입

-> 미사용, absolute position bias 대비 성능 향상

Swin Transformer: Hierarchical Vision Transformer using Shifted Windows

4. Experiments

(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

ImageNet dataset을 이용한
Image Classification task 비교

기존 SOTA 모델(CNN,ViT)를
능가하는 성능

pre-training 했을때
더 우수한 성능

Swin Transformer: Hierarchical Vision Transformer using Shifted Windows

4. Experiments

(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		test-dev		#param.	FLOPs
	AP^{box}	AP^{mask}	AP^{box}	AP^{mask}		
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	-

COCO dataset을 이용한
Object Detection Task

기존 SOTA 모델(CNN,ViT)를
능가하는 성능

ViT(DeiT-S)보다 높은 FPS
-> O(N^2), O(N) 계산 복잡도에서
기인한 차이

Swin Transformer: Hierarchical Vision Transformer using Shifted Windows

4. Experiments

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.

ADE 20K dataset을 이용한
Semantic Segmentation

기존 SOTA 모델(CNN,ViT)를
능가하는 성능

ViT(DeiT)보다
높은 정확도, 큰 FPS

Swin Transformer: Hierarchical Vision Transformer using Shifted Windows

4. Experiments

Ablation Study

Window Shifted 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.

sliding window, shifted window 중
shifted window가 더 빠름

또한 padding 보다 cyclic이 더 큰 FPS

Relative Position Bias

	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**

Table 4. Ablation study on the *shifted windows* approach and different position embedding methods on three benchmarks, using the Swin-T architecture. w/o shifting: all self-attention modules adopt regular window partitioning, without *shifting*; abs. pos.: absolute position embedding term of ViT; rel. pos.: the default settings with an additional relative position bias term (see Eq. (4)); app.: the first scaled dot-product term in Eq. (4).

Relative Position Bias가
유의미한 성능 향상

Swin Transformer: Hierarchical Vision Transformer using Shifted Windows

5. Conclusion

결론 - 1

O(N^2) 시간 복잡도 가진 ViT를 개선한 O(N) 시간 복잡도의 Swin Transformer 도입

결론 - 2

Window 기반 local Self-Attention 계산과 Shifted Window를 이용한 Window 간 connection을 도입

결론 - 3

작은 시간 복잡도를 기반으로 다양한 CV Tasks에 적용가능

추후 연구과제

CV, NLP 분야를 하나로 묶는 통합 모델링

Swin Transformer: Hierarchical Vision Transformer using Shifted Windows