Week11_예습과제_김도희

Swin Transformer: Hierarchical Vision Transformer using Shifted Windows

> 💡 Swin Transformer라는 새로운 ViT를 제안하며 이를 범용 백본 모델로 사용할 수 있다. **hierarchical Transformer**를 제안하며, 이는 Shifted windows를 사용하여 표현을 계산한다.
> 다양한 크기를 모델링할 수 있는 유연성을 제공하며, 이미지 크기에 대해 선형 계산 복잡도를 가지며 다양한 작업에서 좋은 성능을 보여 준다. Transformer 기반 모델이 비전 백본으로서 가질 수 있는 잠재력을 보여줬다.

1. Introduction

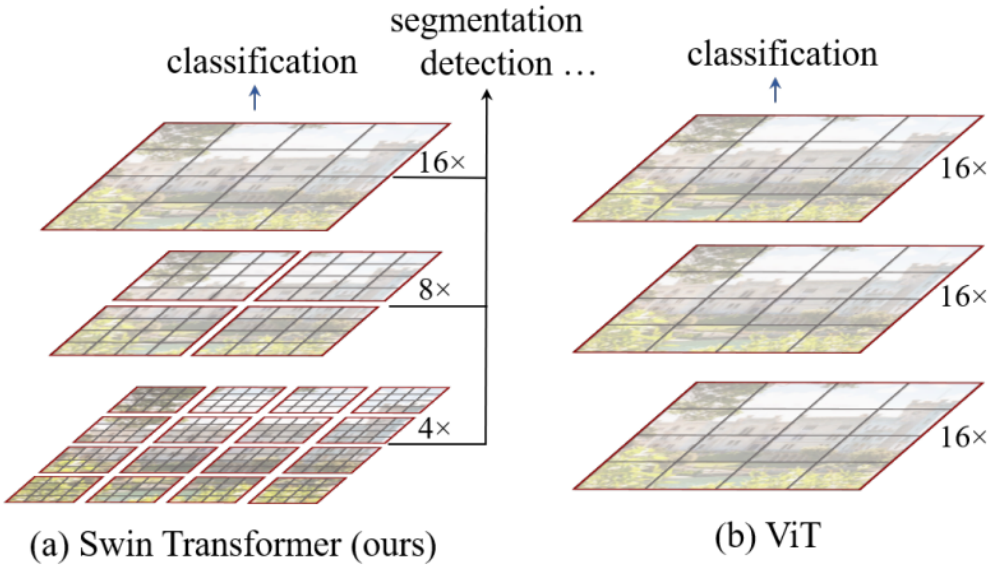
CV 분야: CNN + NLP 분야: Transformer

→ Goal : Transformer의 높은 성능을 컴퓨터 비전에서의 범용 백본으로 확장

❕ 스케일 문제 발생 즉, 토큰과 달리 이미지에서 픽셀은 크기가 달라질 수 있다.

❕ 텍스트에서의 단어보다 이미지의 픽셀이 훨씬 높은 해상도를 가진다

→ **Swin Transformer** 라는 범용 모델 제안



- Swin Transformer는 hierarchical feature maps을 생성하며, 이미지 크기에 대해 선형 계산 복잡도
- Figure 1(a)에서 보여지는 것처럼, Swin Transformer는 작은 크기의 패치에서 시작하여, 더 깊은 Transformer 레이어에서 이웃 패치를 점진적으로 병합하며 계층적 표현을 구성합니다.

💡shifted window

: 연속적인 Self Attention 레이어 사이에서 윈도우의 분할 방식을 이동시키는 것

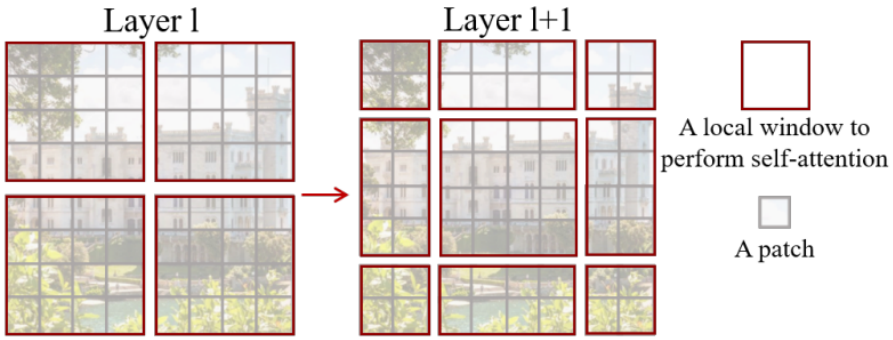


Figure 2. An illustration of the *shifted window* approach for computing self-attention in the proposed Swin Transformer architecture. In layer l (left), a regular window partitioning scheme is adopted, and self-attention is computed within each window. In the next layer $l + 1$ (right), the window partitioning 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.

Swin Transformer는 이미지 분류, 객체 탐지 및 의미론적 분할과 같은 인식 작업에서 강력한 성능을 보여줬다.

2. Related Work

CNN and variants

VGG [52], GoogleNet [57], ResNet [30], DenseNet [34], HRNet [65], EfficientNet [58]과 같은 점점 더 깊고 효과적인 합성곱 신경망 아키텍처들이 등장하며 컴퓨터 비전에서 딥러닝 열풍.

이러한 아키텍처적 발전과 더불어 개별 합성곱 레이어를 개선하려는 연구도 활발히 진행되었다.

CNN과 그 변형들은 여전히 컴퓨터 비전 애플리케이션의 주요 백본 아키텍처로 자리 잡고 있지만, Transformer와 같은 새로운 아키텍처가 비전과 언어를 통합적으로 모델링할 가능성을 보여주며 강력한 잠재력을 발휘하고 있다.

Self-attention based backbone architectures

Transformer 아키텍처의 성공에 영감을 받아 공간적 합성곱 레이어의 일부 또는 전부를 Self-Attention Layer로 대체하려고 했지만 메모리 접근이 비용이 많이 드는 특성 때문에 실제 지연(latency)이 합성곱 네트워크에 비해 훨씬 커지는 문제가 발생

→ 본 논문은 슬라이딩 윈도우 방식 대신 연속적인 레이어 간에 윈도우를 이동하는 방식을 제안하여 일반 하드웨어에서도 더 효율적으로 구현할 수 있도록 했다.

Self-attention/Transformers to complement CNNs

다른 연구에서는 표준 CNN 아키텍처를 Self-Attention 레이어나 Transformer로 보완하려고 하였다. Self-Attention 레이어는 백본이나 헤드 네트워크에서 장거리 의존성을 인코딩하거나 이질적인 상호작용을 모델링하는 데 사용되었다.

→ Transformer를 기본적인 시각적 특징 추출에 적응시키는 방법에 대해 제안

Transformer based vision backbones

ViT의 이미지 분류 결과는 상당하지만, 단일 해상도 특성 맵과 이미지 크기에 따라 제곱으로 증가하는 복잡도 때문에 고해상도 이미지 입력이나 고밀도 비전 작업을 위한 범용 백본 네트워크로는 적합하지 않다.

→ 본 연구의 접근 방식은 효율적이면서도 효과적이며, COCO 객체 탐지와 ADE20K 의미론적 분할에서 SOTA을 달성.

3. Method

Swin Transformer는 입력 이미지를 비중첩 패치로 분할하여 계층적 특성 맵을 생성

3.1. Overall Architecture

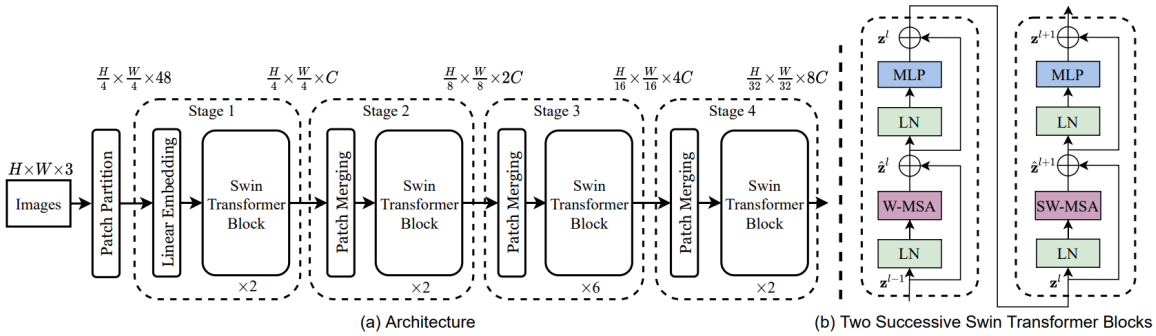


Figure 3. (a) The architecture of a Swin Transformer (Swin-T); (b) two successive Swin Transformer Blocks (notation presented with Eq. (3)). W-MSA and SW-MSA are multi-head self attention modules with regular and shifted windowing configurations, respectively.

- RGB 이미지를 겹치지 않는 패치들로 분할하는 패치 분할 모듈을 먼저 수행
- 각 패치는 "토큰(token)"으로 취급되며, 해당 특성은 원본 픽셀 RGB 값의 concatenation으로 설정
- 원시 특성에 선형 임베딩 레이어를 적용하여 임의의 차원(C)으로 projection
- Transformer 블록을 적용한다. Transformer 블록은 토큰 수($H/4 \times W/4$)를 유지

- hierarchical representation을 생성하기 위해, 네트워크가 깊어질수록 패치 병합 레이어를 통해 토큰 수를 줄인다.
- Swin Transformer 블록이 적용되어 특성이 변환되며, 해상도는 $H/8 \times W/8$으로 유지됩니다. 이 첫 번째 블록은 "Stage 2"이다. 이 절차는 두 번 더 반복되어 "Stage 3"과 "Stage 4"를 생성하며, 출력 해상도는 각각 $H/16 \times W/16$과 $H/32 \times W/32$가 된다.

3.2. Shifted Window based Self-Attention

기존의 Transformer는 토큰 간의 모든 관계를 계산하므로, 토큰 수에 대해 제곱 복잡도를 초래하는 무제가 발생한다. 이는 고밀도 예측을 요구하거나 고해상도 이미지를 표현하기 위해 대규모 토큰 세트를 필요로 하는 많은 비전 문제에 적합하지 않다.

Self-attention in non-overlapped windows

효율적인 모델링을 위해, 지역적 윈도우 내에서 Self attention을 계산하는 방식을 제안한다. 윈도우는 이미지의 비중첩 방식으로 균등하게 배치한다.

$$\Omega(MSA) = 4hwC^2 + 2(hw)^2C$$
$$\Omega(W-MSA) = 4hwC^2 + 2M^2hwC$$

각 윈도우가 $M \times M$ 패치를 포함한다고 가정하면, 높이 h 너비 w를 가진 이미지에서의 계산 복잡도는 위와 같다.

Shifted window partitioning in successive blocks

윈도우 기반 Self Attention 모듈은 윈도우 간의 연결이 부족하여 모델링 능력이 제한될 수 있다. 이를 해결하기 위해, 윈도우의 분할 방식을 이동시키는
shifted window 접근법을 제안했다. 이 방법은 연속적인 Swin Transformer 블록 간의 효율적인 계산을 유지하면서 윈도우 간 연결을 돕는다.

연속적인 Swin Transformer 블록은 다음과 같이 계산된다:

$$
\begin{aligned}
\hat{\mathbf{z}}^{l} &= \text{W-MSA}\left(\text{LN}\left(\mathbf{z}^{l-1}\right)\right) + \mathbf{z}^{l-1}, \\
\mathbf{z}^{l} &= \text{MLP}\left(\text{LN}\left(\hat{\mathbf{z}}^{l}\right)\right) + \hat{\mathbf{z}}^{l}, \\
\hat{\mathbf{z}}^{l+1} &= \text{SW-MSA}\left(\text{LN}\left(\mathbf{z}^{l}\right)\right) + \mathbf{z}^{l}, \\
\mathbf{z}^{l+1} &= \text{MLP}\left(\text{LN}\left(\hat{\mathbf{z}}^{l+1}\right)\right) + \hat{\mathbf{z}}^{l+1}, \qquad (3)
\end{aligned}
$$

Efficient batch computation for shifted configuration

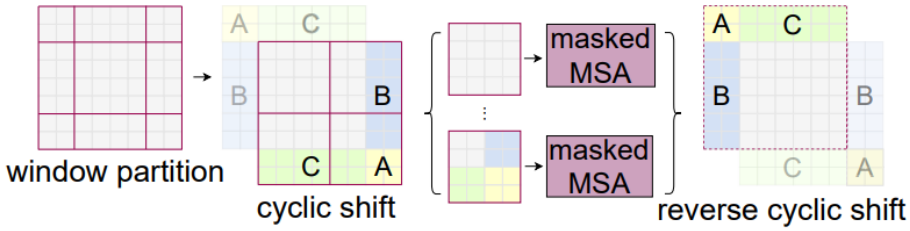


Figure 4. Illustration of an efficient batch computation approach for self-attention in shifted window partitioning.

이동된 윈도우는 더 많은 윈도우를 생성하므로, 작은 윈도우를 패딩한 후 마스크를 사용해 계산하는데 이는 계산량이 증가한다. 이를 해결하기 위해
cyclic shift 방식을 사용하여, 윈도우 크기를 일정하게 유지하면서 효율성을 확보한다.

Relative position bias

relative position bias을 추가하여 유의미한 성능 향상

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

3.3. Architecture Variants

- 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 =$\{2, 2, 18, 2\}$

4. Experiments

4.1. Image Classification on ImageNet-1K

- Setting
  - ImageNet-1K는 1,000개의 클래스에서 128만 개의 학습 이미지와 50,000개의 검증 이미지를 포함.
  - single crop에서의 Top-1 정확도

(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].

- Swin Transformer는 DeiT와 EfficientNet보다 우수한 성능-속도 Trade-off를 달성.
- ImageNet-22K 사전학습을 통해 더 높은 성능을 얻었다.

4.2. Object Detection on COCO

- Setting
  - 객체 탐지 및 인스턴스 분할 실험은 COCO 2017 데이터셋에서 수행.

- COCO 2017은 118,000개의 학습 이미지, 5,000개의 검증 이미지, 20,000개의 테스트-데브(test-dev) 이미지를 포함

(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	-

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.

- Cascade Mask R-CNN, ATSS 등 다양한 프레임워크에서 ResNet과 DeiT를 능가하는 성능
- Swin-L은 이전 최고 모델 대비 +2.7 box AP와 +2.6 mask AP의 성능 향상을 달성했다.

4.3. Semantic Segmentation on ADE20K

- Setting
  - 150개의 의미론적 카테고리를 포함한 데이터셋으로, 총 25,000개의 이미지가 있으며, 이 중 20,000개는 학습용, 2,000개는 검증용, 3,000개는 테스트용
  - AdamW 최적화기를 사용해 160K iterations 동안 진행했으며, 다중 스케일 학습과 Stochastic Depth를 적용

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.

- Swin-S는 DeiT-S 대비 +5.3 mIoU 더 높은 성능을 기록했다.

- Swin-L은 53.5 mIoU를 기록하며 이전 최첨단 모델보다 +3.2 mIoU 더 높았다.

4.4. Ablation Study

	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).

shifted window 접근법과 다양한 위치 임베딩 방법에 대한 ablation study 결과

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

다음은 다양한 self-attention 계산 방법에 대한 실제 속도를 비교한 표

	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

- Shifted Window와 relative position bias이 모델 성능 향상에 기여함을 확인

5. Conclusion

- Swin Transformer는 효율성과 성능을 동시에 제공하는 계층적 Vision Transformer로, 이미지 크기에 대해 선형 계산 복잡도를 가진다.
- 객체 탐지, 의미론적 분할 등에서 SOTA을 달성하며, 비전과 언어 신호의 통합 모델링을 촉진
- Shifted-Window 기반 Self-Attention은 비전 문제뿐만 아니라 NLP에서도 잠재력을 가진다.