Swin Transformer: Hierarchical Vision Transformer using Shifted Windows_ 이은경

Column
references

keypoint : computation complexity, shifted window

발표자 : 이은경

0. Abstract

- 컴퓨터 비전의 범용 백본 역할을 할 수 있는 새로운 비전 트랜스포머(Swin Transformer)를 소개.

- NLP에서 vision으로 트랜스포머를 적응(adapting)시키는 문제는 두 domains 간의 차이에서 발생.
    - 예) visual entities의 scale의 visual entities와 텍스트의 단어와 비교하여 이미지의 픽셀 해상도가 높음.
- 이러한 차이를 해결하기 위해 논문은 representation이 **shifted windows**으로 계산되는 **계층적 트랜스포머**를 제안.
- shifted windowing scheme는 **cross-window connection을 허용**하는 동시에 self-attention 계산을 **non-overlapping local windows으로 제한함으로써 효율성** 향상.
- 이 계층 구조 아키텍처는 다양한 scales로 모델링할 수 있는 **flexibility**과 image size에 대한 linear computational **complexity**을 가짐.
- Swin Transformer의 성능은 이미지 분류(ImageNet-1K에서 86.4 top-1 정확도)와 object detection(58.7 box AP 및 51.1 mask AP on COCO test-dev) 및 semantic segmentation(53.5 mioU on ADEK on ADME 20)과 같은 dense prediction task을 포함한 광범위한 비전 과제와 호환가능
- 성능은 COCO에서는 +2.7 박스 AP 및 +2.6 마스크 AP, ADE20K에서는 +3.2 mIoU의 큰 차이로 이전 SOTA 모델을 능가하며 vision 백본으로서의 트랜스포머 기반 모델의 잠재력을 보여줌.
- 코드와 모델은 https://github.com/microsoft/Swin-Transformer에서 공개.

1. Introduction

- computer vision에서의 모델링은 오랫동안 CNN에 의해 지배되어옴.
- ImageNet 이미지 분류 문제에 대한 AlexNet [38]과 혁신적인 성능을 시작으로, CNN 아키텍처는 greater scale[29, 73], more extensive connections[33], more extensive connections[67, 17, 81]을 통해 점점 더 강력해짐.
- CNN이 다양한 vision tasks를 위한 백본 네트워크 역할을 하는 가운데, 이러한 아키텍처의 발전은 전체 분야를 광범위하게 끌어올린 성능 향상으로 이어짐.
- 반면에, 자연 언어 처리(NLP)에서의 네트워크 아키텍처의 진화는 다른 경로를 따름.
    - 보편적인 아키텍처 : Transformer[61].
    - 시퀀스 모델링 및 변환 작업을 위해 설계된 Transformer는 장거리 의존성 모델링에 attention 사용.
    - 최근 image classification[19]와 joint vision-language modeling[46]과 같은 유망한 결과를 보여줌.

- 본 논문은 Transformer 가 NLP와 vision에서 범용 백본 역할을 할 수 있도록 함.
- 언어 영역의 고성능을 시각적 영역으로 전달하는 데 있어 중요한 문제는 두 **modalities (image ↔ text)** 간의 차이로 볼 수 있음.
- 차이점
  - 1) scale
    - language Transformer 처리의 기본 요소 역할을 하는 단어 토큰과는 달리, object detection[41, 52, 53]의 시각적 요소는 **scale**에서 달라질 수 있음.
    - 기존의 Transformer 기반 모델[61, 19]에서 **토큰**은 모두 고정된 **scale이므로** vision applications에 적합하지 않음.
  - 2) 해상도
    - passages of text의 words에 비해 이미지의 픽셀 해상도가 훨씬 높음.
    - semantic segmentation : 픽셀 수준에서 dense prediction 이 필요 > self-attention의 계산 복잡성이 이미지 크기에 **quadratic계산 복잡도**를 갖기 때문에 고해상도 이미지에서는 Transformer가 다루기 어려움.
- 이러한 문제를 해결하기 위해 논문은 **hierarchical feature maps**을 구성하고 이미지 크기에 **Linear 계산 복잡도**를 갖는 범용 Transformer backbone인 Swin Transformer를 제안.

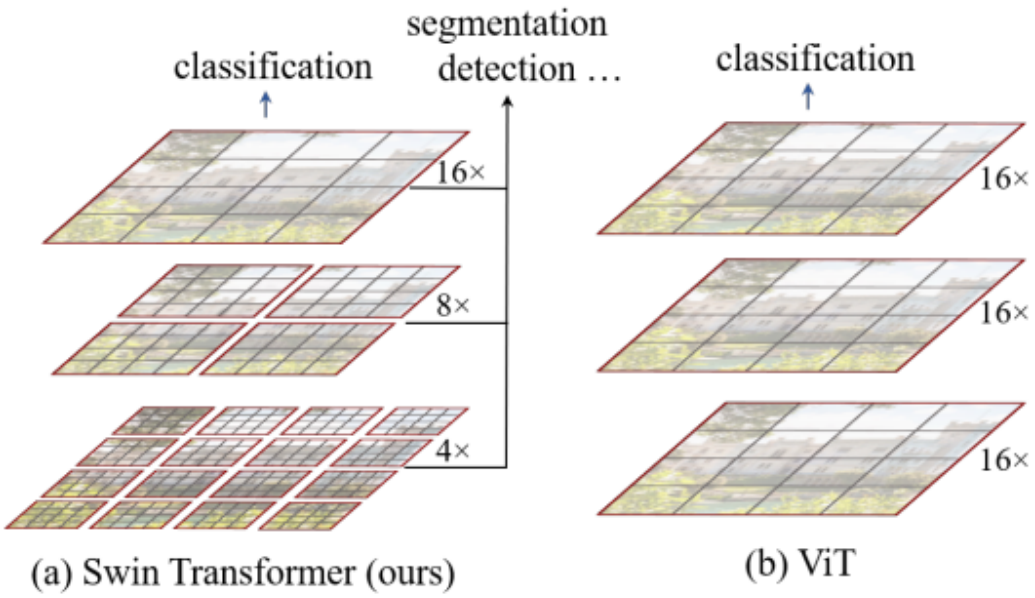
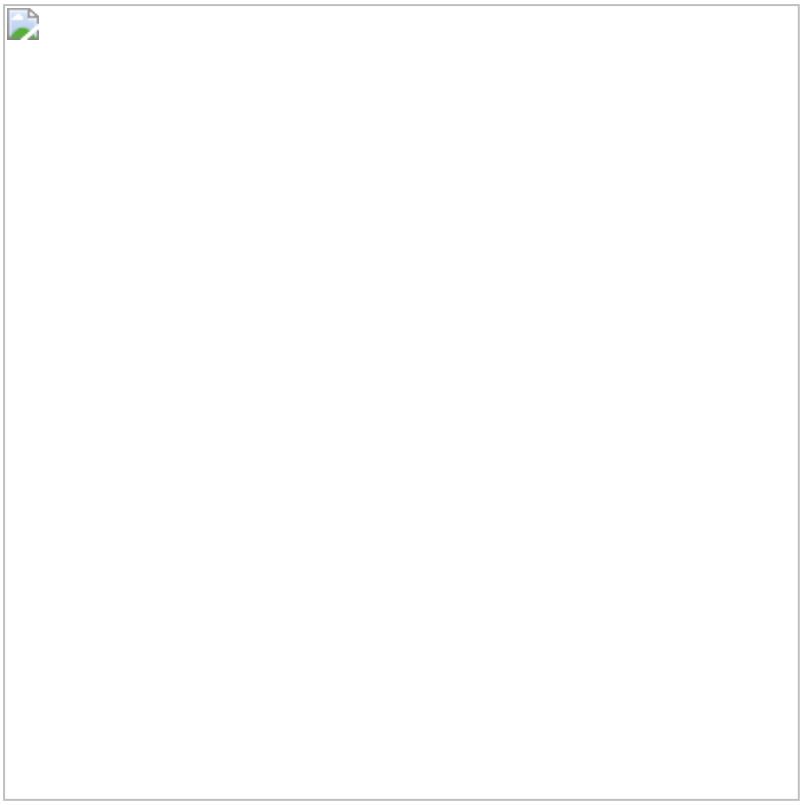


그림 1. (a) 제안된 Swin Transformer는 이미지 패치(회색)를 더 깊은 계층으로 병합하여 계층적 피 맵을 구축하며, 각 local window(빨간색) 내에서만 self-attention의 계산으로 인해 이미지 크기를 입력하기 위한 linear computation complexity을 가짐. 따라서 이미지 분류와 dense recognition tasks 모두에 범용 백본 역할을 할 수 있음. (b) 반대로, 이전 비전 Transformers[19]는 feature maps of a single low resolution을 생성하며, 전체적으로 self-attention 의 계산으로 인해 영상 크기를 입력하기 위한 quadratic computation complexity을 가짐.

- + Vit 사진을 동일한 걸 쓴 이유?

- 그림 1(a)에서 볼 수 있듯이, Swin Transformer는 작은 크기의 patch(회색)에서 시작하여 점차적으로 더 상위(deeper) 트랜스포머 계층에서 인접 patch를 병합하여 계층적 feature를 구성.
- 이러한 계층적 feature map을 통해, Swin Transformer 모델은 feature pyramid networks (FPN) [41] 또는 **U-Net** [50]과 같은 dense prediction을 위한 고급 기술을 편리하게 활용할 수 있음.
- **linear** computational complexity은 이미지를 분할하는 **non-overlapping windows**(빨간색) 내에서 locally 하게 self-attention를 계산함으로써 달성.
- 각 windows의 패치 수가 고정되므로 complexity가 **이미지 크기에 비례**.
- 이러한 장점 때문에 Swin Transformer는 single resolution의 feature map을 생성하며 **quadratic complexity**를 갖는 이전의 Transformer 기반 아키텍처[19]와는 달리 다양한 비전 작업에 대한 범용 백본으로서 적합.
- Swin Transformer의 핵심 설계 요소는 그림 2에서와 같이 연속적인 self-attention 계층 간 window partition의 이동.

- shifted window 은 이전 레이어의 windows를 브리지하여 모델링 power를 크게 향상시키는 연결을 제공(Table 4 참조).
- 이 전략은 실제 대기 시간과 관련해서도 효율적.
    - 즉, window 내의 모든 query patch는 동일한 key sets (The query and key are projection vectors in a self-attention layer)를 공유하므로 하드웨어의 메모리 액세스가 용이해짐.
- 반면, 이전의 sliding window 기반 self-attention approaches[32, 49]은 서로 다른 query pixel에 대해 서로 다른 key sets로 인해 일반 하드웨어에서 짧은 지연 시간이 발생함.
    - 논문의 실험을 통해 제안된 shifted window 접근 방식은 sliding window 방식보다 지연 시간이 훨씬 짧지만 모델링 power은 비슷하다는 것을 알 수 있음(표 5와 6 참조).
- 제안된 Swin Transformer는 이미지 분류, 객체 감지 및 의미 분할의 인식 작업에서 강력한 성능을 달성.
    - ViT/DeT[19, 60] 및 ResNe(X)t 모델[29, 67]보다 성능이 뛰어나며 세 가지 작업에서 유사한 지연 시간이 발생.
    - COCO test-dev set의 58.7 box AP 및 51.1 mask AP는 +2.7 box AP(외부 데이터가 없는 Copy-paste [25]) 및 +2.6 mask AP(DetectorRS [45])로 이전 SOTA 결과를 능가.
    - ADE20K semantic segmentation의 경우, val set에서 53.5mIoU를 얻는데, 이는 이전 state-of-the-art(SETR [78])에 비해 +3.2mIoU가 개선됨.
    - ImageNet-1K 이미지 분류에서 86.4%의 정확도를 달성.



- 컴퓨터 비전과 자연 언어 처리에 걸친 통합 아키텍처는 시각 신호와 텍스트 신호의 공동 모델링을 촉진하고 두 도메인의 모델링 지식을 더 깊이 공유할 수 있기 때문에 양쪽 모두에 도움이 될 수 있다고 생각함.
    - 논문은 다양한 비전 문제에 대한 Swin Transformer의 강력한 성능이 커뮤니티에 이러한 믿음을 더 깊이 심어주고 비전 및 언어 신호의 통합 모델링을 장려할 수 있기를 바람.

2. Related Work

▼ CNN and variants

- CNN은 컴퓨터 비전 전반에 걸쳐 표준 네트워크 모델 역할. CNN이 수십 년 동안 존재했지만 [39] AlexNet을 도입하고 나서야 CNN이 주류가 됨. 그 이후, 컴퓨터 비전의 딥러닝 파동을 더욱 촉진하기 위해 deeper and more effective convolutional neural architectures가 제안됨. VGG [51], GoogleNet [56], ResNet [29], DenseNet [33],HRNet [62] 및 EfficientNet [57].
- 이러한 아키텍처의 진보 외에도, depthwise convolution [67] 및 deformable convolution [17, 81]과 같은 개별 컨볼루션 레이어의 개선에 대한 많은 연구가 있었음.
- CNN과 그 variants는 여전히 컴퓨터 비전 애플리케이션의 주요 백본 아키텍처이지만, 우리는 시각과 언어 사이의 통합 모델링을 위한 트랜스포머와 같은 아키텍처의

강력한 잠재력을 강조. 저자는 이 작업이 몇 가지 기본적인 시각적 인식 작업에서 강력한 성과를 달성하며 모델링 전환에 기여하기를 바람.

▼ Self-attention based backbone architectures

- 또한 NLP 분야에서 Self-attention layers 과 Transformer architectures의 성공에 영감을 받아, 일부 작업은 인기 있는 ResNet에서 공간적 변환 계층의 일부 또는 전부를 대체하기 위해 Self-attention layers을 사용[32, 49, 77].
- 이 작업에서는 각 픽셀의 local windows 내에서 Self-attention 를 계산하여 최적화[32]를 촉진하고, counterpart ResNet 아키텍처보다 약간 더 나은 accuracy/FLOPs trade-offs를 달성. 그러나, costly memory access는 실제 대기 시간이 convolutional networks의 대기 시간보다 훨씬 더 커지게 함[32].
- sliding windows를 사용하는 대신 consecutive layers 간에 shift windows를 사용하여 일반 하드웨어에서 보다 효율적으로 구현할 수 있도록 제안.

▼ Self-attention/Transformers to complement CNNs

- 표준 CNN 아키텍처를 self-attention layers 또는 Transformers로 강화하는 것. self-attention layers은 distant dependencies 또는 heterogeneous interactions을 인코딩하는 기능을 제공하여 complement backbones[64, 6, 68, 22, 71, 54] 또는 head networks[31, 26]를 보완할 수 있음. 보다 최근에는 트랜스포머의 encoder-decoder 설계가 object detection and instance segmentation tasks에 적용[7, 12, 82, 55]. 본 연구에서는 기본적인 visual feature extraction을 위한 트랜스포머의 adaptation을 살펴보고 이러한 작업을 보완함.

▼ Transformer based vision backbones

- ViT(Vision Transformer)[19]와 그 후속연구[60, 69, 14, 27, 63]와 관련.
- ViT의 pioneering work는 이미지 분류를 위해 겹치지 않는 중간 크기의 이미지 패치에 Transformer 아키텍처를 직접 적용. Convolutional 네트워크와 비교하여 이미지 분류에서 놀라운 speed-accuracy trade-off를 이룸. ViT는 우수한 성능을 발휘하려면 대규모 training datasets(즉, JFT-300M)이 필요하지만 DeiT[60]는 더 작은 ImageNet-1K 데이터셋을 사용하여 ViT를 효과적으로 운영할 수 있는 여러 training strategies을 도입.
- ViT image classification 결과는 encouraging하지만, 이 아키텍처는 low-resolution feature maps과 이미지 크기에 따라 quadratic increase in complexity 때문에 dense vision tasks이나 입력 이미지 해상도의 범용 백본 네트워크로 사용하기에 적합하지 않음.
- 직접적인 upsampling 또는 deconvolution을 통한 object detection 과 semantic segmentation 의 dense vision tasks에 VIT 모델을 적용하는 몇 가지 연구가 있음

[2, 78].

- 더 나은 이미지 분류를 위해 ViT 아키텍처[69, 14, 27]를 수정하는 작업도 있음. Empirically, 이미지 분류에 관한 이러한 방법들 중에서 speed-accuracy trade-off을 달성하기 위해 Swin Transformer 아키텍처를 개발.
- 비록 논문의 연구가 특별히 분류보다는 범용 성능에 초점을 맞추고 있음에도 불구하고, 또 다른 concurrent work[63]에서는 Transformers에서 multi-resolution feature maps을 구축하기 위한 유사한 사고 방식을 살펴봄. complexity는 여전히 이미지 크기에 quadratic 한 반면, 논문의 complexity는 linear이며 또한 locally 하게 작동하여 시각적 신호의 높은 상관 관계를 모델링하는 데 도움이 됨 [35, 24, 40].
- 논문의 접근방식은 효율적이면서도 효과적이어서 COCO object detection와 ADE20K semantic segmentation 모두에서 state-of-the-art accuracy를 달성.

3. Method

3.1. Overall Architecture

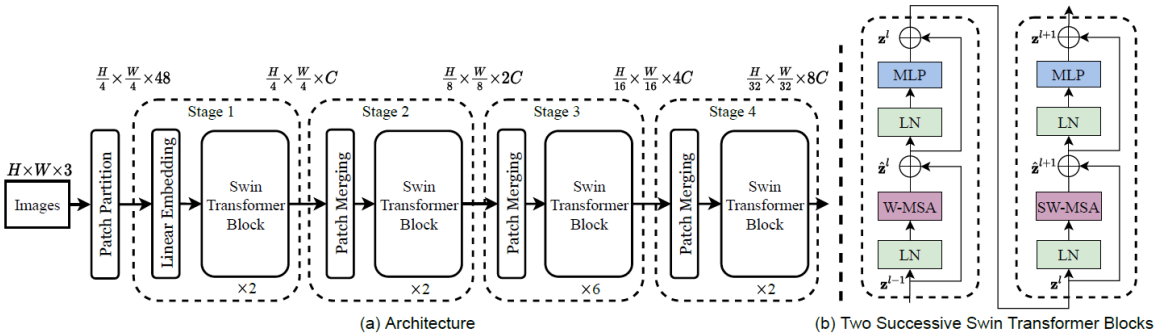


Figure 3. (a) The architecture of a Swin Transformer (Swin-T); (b) two successive Swin Transformer Blocks (notation presented with Eq. (3)). W-MSA and SW-MSA are multi-head self attention modules with regular and shifted windowing configurations, respectively.

- 그림 3에는 작은 버전(Swin-T)을 보여주는 Swin Transformer 아키텍처의 개요 소개.

  1) ViT와 같은 패치 분할 모듈에 의해 입력 RGB 이미지를 겹치지 않는 패치로 분할.

  - 각 패치는 "token"으로 처리되고 해당 feature는 raw pixel RGB 값의 연결로 설정됨. 구현 시, 4 × 4의 패치 크기를 사용하므로 각 패치의 feature dimension는 4 × 4 × 3 = 48.
  - 이 raw-valued feature에 linear embedding layer가 적용되어 arbitrary dimension(C로 표시됨)로 투영됨.

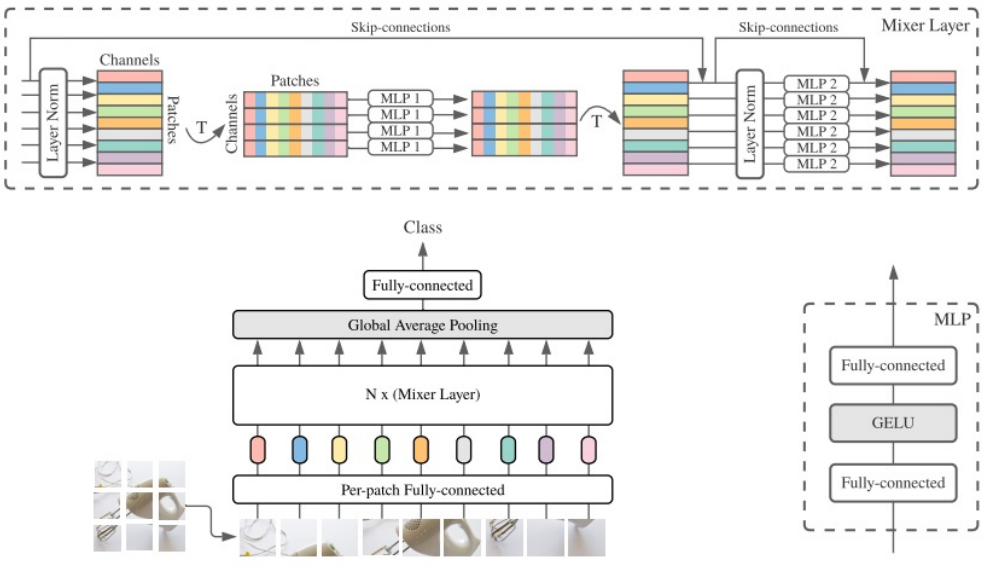
- Stage 1 : modified self-attention computation(Swin Transformer 블록)이 수정된 여러 트랜스포머 블록이 패치 토큰에 적용됨. Transformer 블록은 $(\frac{H}{4} \times \frac{W}{4})$의 토큰을

유지하며 linear embedding과 함께 사용

- 계층적 표현을 생성하기 위해 네트워크의 깊이가 깊어질수록 patch merging layers에 의해 토큰 수가 감소.
  - 첫 번째 patch merging layer는 2 × 2 인접한 patch 의 각 그룹의 특징을 연결하고 4C차원 concatenated features에 linear layer를 적용.
  - 이렇게 하면 토큰 수가 2 × 2 = 4의 배수(resolution 의 2× 다운샘플링)로 감소하고 출력 dimension이 2C로 설정됨.
  - 이후 $\frac{H}{8} \times \frac{W}{8}$resolution을 유지하면서 feature transformation을 하기 위해 Swin Transformer 블록을 적용.
    - 이 patch merging 및 feature transformation의 첫 번째 블록을 "Stage 2"로 표시.
    - $\frac{H}{16} \times \frac{W}{16}$과 $\frac{H}{32} \times \frac{W}{32}$의 output resolutions 은 각각 'Stage 3'와 'Stage 4'로 두 차례 반복.
    - 이러한 단계는 VGG [51] 및 ResNet [29]와 같은 일반적인 컨볼루션 네트워크의 것과 동일한 feature map resolutions으로 hierarchical representation을 공동으로 생성.
    - 그 결과, 제안된 아키텍처는 다양한 비전 작업을 위해 기존 방식에서 백본 네트워크를 편리하게 대체할 수 있어짐.

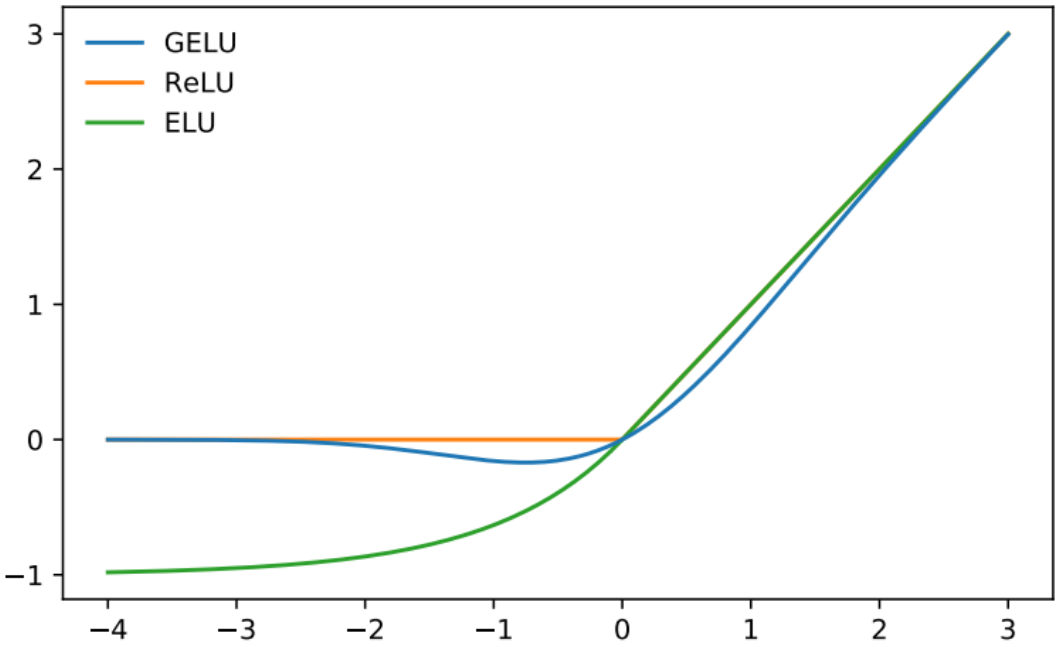
▼ MLP layer

Fully connected - GELU - Fully connected



reference : MLP Mixer

- Swin Transformer block의 Swin Transformer는 Transformer block의 표준 multi-head self attention(MSA) 모듈을 Shift window를 기반으로 하는 모듈로 교체하고(섹션 3.2 참조) 다른 레이어를 동일하게 유지.
- 그림 3(b)에서 볼 수 있듯이, Swin Transformer 블록은 shifted window 기반 MSA 모듈로 구성되고 그 다음에 GELU 비선형성(수렴 속도 빠름) 을 사이에 둔 **이단 MLP**로 구성됨.



  - 각 MSA 모듈 및 각 MLP 앞에 **LN(Layer Norm)** 레이어가 적용되고 각 모듈 뒤에 residual connection이 적용됨.

3.2. Shifted Window based Self-Attention

- 표준 Transformer 아키텍처[61]와 이미지 분류에 대한 adaptation[19]은 토큰과 다른 모든 토큰 간의 관계가 계산되는 global self-attention를 수행함.
  - global computation은 토큰 수와 관련하여 **quadratic complexity**으로 이어지며, dense prediction 또는 high-resolution 이미지를 나타내기 위해 엄청난 토큰 집합이 필요한 많은 비전 문제에 적합하지 않음.
- Self-attention in non-overlapped windows
  - 효율적인 모델링을 위해 local windows의 self-attention를 계산할 것을 제안함.
    - windows는 겹치지 않는 방식으로 이미지를 고르게 분할하도록 배열됨. 각 window에 M × M 패치가 포함되어 있다고 가정할 때, global MSA 모듈의 계산 복잡도와 h × w 패치의 이미지를 기반으로 하는 windows는

- $\Omega(MSA) = 4hwC^2 + 2(hw)^2C$ (1) global MSA
- $\Omega(WMSA) = 4hwC^2 + 2M^2hwC$ (2) Shifted Window MSA
  - 여기서 (1)은 패치 번호 hw에 quadratic이고 (2)는 M이 고정되면 후자는 linear(기본적으로 **7**로 설정됨).
  - global self-attention computation은 일반적으로 large hw에 비해 unaffordable한 반면 window based self-attention은 확장 가능.

- Shifted window partitioning in successive blocks
  - window-based self-attention module에는 window 간 연결이 부족하여 모델링 성능이 제한됨.
  - non-overlapping windows의 효율적인 계산을 유지하면서 cross-window connections을 도입하기 위해 연속적인 SwinTransformer 블록의 두 분할 구성을 번갈아 사용하는 shifted window partitioning 방식 제안.

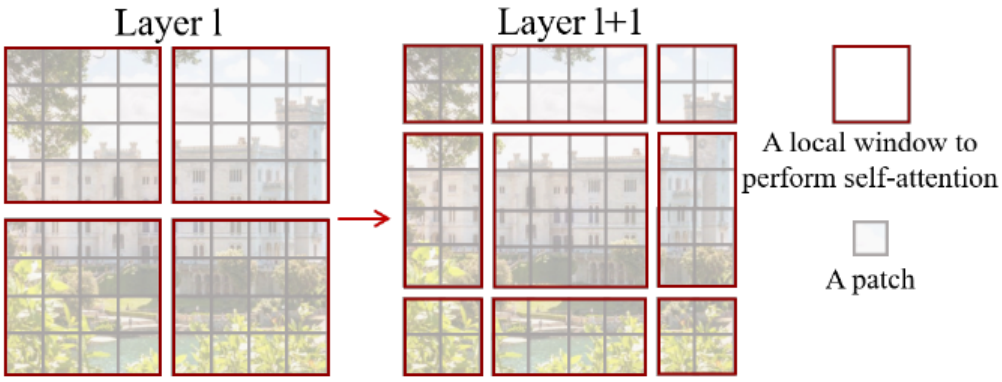


Figure 2. An illustration of the *shifted window* approach for computing self-attention in the proposed Swin Transformer architecture. In layer l (left), a regular window partitioning scheme is adopted, and self-attention is computed within each window. In the next layer $l + 1$ (right), the window partitioning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

  - 그림 2에서 볼 수 있듯이, 첫 번째 모듈은 왼쪽 상단 픽셀에서 시작되는 regular window partitioning strategy을 사용하며, 8 × 8 feature map은 크기가 4 × 4 (M = 4)인 2 × 2 windows로 균등하게 분할됨.
  - 다음 모듈은 정기적으로 분할된 windows에서 $(\lfloor \frac{M}{2} \rfloor, \lfloor \frac{M}{2} \rfloor)$픽셀로 windows를 대체하여 이전 계층에서 shifted window 구성을 채택.

        - shifted window partitioning approach를 사용하면 연속적인 Swin Transformer 블록이 다음과 같이 계산됨.
            - $\hat{z} = W - MSA(LN(z^{l-1})) + z^{l-1}$,
            - $z^{l} = MLP(LN(\hat{z}^{l})) + \hat{z}^{l}$,
            - $\hat{z}^{l+1} = SW - MSA(LN(z^{l})) + z^{l}$,
            - $z^{l+1} = MLP(LN(\hat{z}^{l+1})) + \hat{z}^{l+1}$ (3)
                - 여기서 $\hat{z}$과 z^{l} 은 각각 블록 l에 대한 (S)W-MSA 모듈과 MLP 모듈의 출력 기능을 나타냅니다. W-MSA 및 SW-MSA는 각각 일반 및 shifted window partitioning 구성을 사용한 window based multi-head self-attention를 나타냄.
    - shifted window partitioning 접근방식은 이전 계층에서 인접한 non-overlapping windows 사이의 connections을 도입하며, 표 4와 같이 image classification, object detection, and semantic segmentation,에 효과적인 것으로 확인됨.
- Efficient batch computation for shifted configuration
    - shifted window partitioning의 문제는 이동 구성의 $\lceil \frac{h}{M} \rceil \times \lceil \frac{w}{M} \rceil$ to $(\lceil \frac{h}{M} \rceil + 1) \times (\lceil \frac{w}{M} \rceil + 1)$에서 더 많은 window 가 발생하며 일부 window는 $M \times M$보다 작음.
    - naive solution은 size of $M \times M$ 보다 작은windows는 pad 하고 a attention을 계산할 때 padded values을 mask out. regular partitioning의 the number of windows가 작을 때,
        - 예) 2 × 2일 때, 이 naive solution을 사용한 증가된 계산은 상당함. (2 × 2 → 3 × 3 , 이는 9/4 = 2.25 배 더 큼.)
    - 여기서, 논문은 그림 4와 같이 top-left direction으로 cyclic-shifting함으로써 보다 more efficient batch computation approach를 제안.
    - 이동 후에는 batched window가 feature map에 인접하지 않은 여러 sub-window(A : 색 4개 조합)로 구성될 수 있으므로 masking mechanism을 사용하여 각 하위 window 내에서 self-attention computation을 제한.
    - cyclic-shift을 사용하면 batched windows의 수가 regular window partitioning의 window 수와 동일하게 유지되므로 효율적이기도 함. 이 접근법의 low latency는 표 5에 나와 있음.

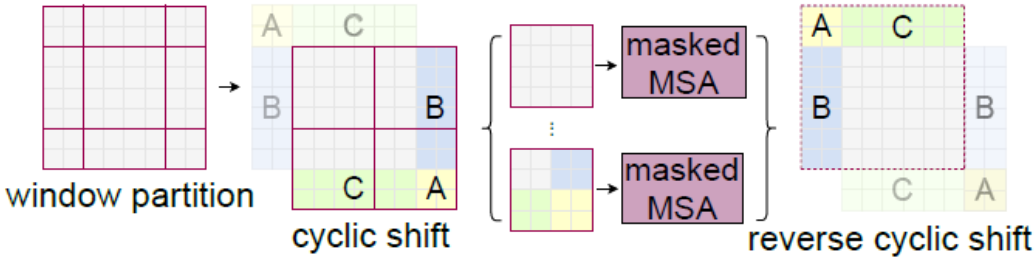


Figure 4. Illustration of an efficient batch computation approach for self-attention in shifted window partitioning.

그림 4. shifted window partitioning에서 self-attention를 기울일 수 있는 효율적인 batch computation approach을 보여줍니다.

- Relative position bias
  - self-attention computation에서는 computing similarity 에 relative position bias $B \in \mathbb{R}^{M^2 \times M^2}$를 포함시켜 [48, 1, 31, 32] $Attention(Q, K, V) = SoftMax(QK^T/\sqrt{d} + B)V$를 따름.
    - 여기서 $Q, K, V \in \mathbb{R}^{M^2 \times d}$는 쿼리, 키 및 값 매트릭스, d는 query/key dimension, M^2는 창에 있는 패치 수. 각 축을 따라 상대적인 위치가 $[-M + 1, M - 1]$범위에 있으므로 작은 크기의 바이어스 행렬 $\hat{B} \in \mathbb{R}^{(2M-1)\times(2M-1)}$의 값을 B에서 취한다.
  - 표 4에서와 같이 이러한 bias 항이 없거나 absolute position embedding을 사용하는 논문에 비해 현저한 개선을 관찰.
    - [19]에서와 같이 absolute position embedding을 입력에 추가하면 성능이 약간 저하되므로 구현에서 채택되지 않음.
  - pre-training에서 학습된 relative position bias을 사용하여 bi-cubic interpolation을 통해 다른 window size로 fine-tuning을 위한 모델을 초기화할 수도 있음[19, 60].

3.3. Architecture Variants

- ViT-B/DeiT-B와 유사한 모델 크기와 계산 복잡성을 갖도록 Swin-B라는 기본 모델을 구축.
- 또한 모델 크기 약 0.25배, 0.5배, 2배인 Swin-T, Swin-S, Swin-L을 소개.
  - Swin-T와 Swin-S의 복잡도는 각각 ResNet-50(DeiT-S)과 ResNet-101과 유사.
  - window size는 기본적으로 M = 7로 설정.

    - 모든 실험에서 각 head의 query dimension은 d = 32이고, 각 MLP의 확장 레이어는 α = 4.
    - 이러한 모델 모델의 아키텍처 하이퍼 파라미터는 다음과 같습니다.
        - 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}
- 여기서 C는 첫 번째 단계에서 숨겨진 레이어의 채널 number.
- ImageNet 이미지 분류에 대한 모델 크기, 이론적 계산 복잡성(FLOP) 및 모델 변형 처리량은 표 1.

4. Experiments

- ImageNet-1K image classification[18], COCO object detection[42] 및 ADE20K semantic segmentation[80]에 대한 실험을 수행.
- 다음에서는 먼저 제안된 Swin Transformer 아키텍처를 세 가지 작업에 대한 이전의 SOTA와 비교. 그런 다음 Swin Transformer의 중요한 디자인 요소를 완화.

4.1. Image Classification on ImageNet-1K

- Settings
    - 이미지 분류를 위해 1,000개의 클래스에서 128M개의 training 이미지와 50K valid 이미지가 포함된 ImageNet-1K[18]에서 제안된 Swin Transformer를 벤치마크함.
    - single crop에서 top-1 accuracy가 보고.
    - 다음 두 가지 교육 설정을 고려합니다.
        - Regular ImageNet-1K training.
            - 이 설정은 대부분 [60]을 따릅니다. cosine decay learning ratescheduler와 20epoch의 linear warm-up을 사용하여 300epoch에 대해 AdamW[36] optimizer를 사용.
            - 배치 크기 1024, 초기 학습 속도 0.001 및 weight decay 0.05가 사용됨.
            - [30] 및 [44]의 성능을 향상시키지 않는 반복적인 augmentation을 제외하고 [60]의 대부분의 augmentation and regularization 전략을 training에

포함시킴. 이는 ViT의 training을 안정화하는 데 있어 반복적인 augmentation이 중요한 [60]과 반대되는 점에 유의.

- ImageNet-22K에 대한 Pre-training 및 ImageNet-1K에 대한 fine-tuning 또한 14.2 million 개의 이미지와 22K 클래스를 포함하는 대규모 ImageNet-22K 데이터셋에 대한 pre-train도 실시.
- 5-epoch linear warm-up0과 함께 linear decay learning rate scheduler를 사용하여 60 epochs에 AdamW optimizer를 사용.
- 배치 크기 4096, 초기 학습률 0.001 및 weight decay 0.01이 사용. ImageNet-1K fine-tuning에서는 배치 크기 1024, 일정한 학습 속도 10^-5, weight decay 10^-8로 30 epochs 모델을 training.

- Results with regular ImageNet-1K training
  - 표 1(a)은 regular ImageNet-1K training을 사용하여 Transformer-based 와 ConvNet-based 모두를 포함한 다른 백본과 비교state-of-the-art Transformerbased.

(a) Regular ImageNet-1K trained models					
method	image size	#param.	FLOPs	throughput (image / s)	ImageNet top-1 acc.
RegNetY-4G [47]	224^2	21M	4.0G	1156.7	80.0
RegNetY-8G [47]	224^2	39M	8.0G	591.6	81.7
RegNetY-16G [47]	224^2	84M	16.0G	334.7	82.9
EffNet-B3 [57]	300^2	12M	1.8G	732.1	81.6
EffNet-B4 [57]	380^2	19M	4.2G	349.4	82.9
EffNet-B5 [57]	456^2	30M	9.9G	169.1	83.6
EffNet-B6 [57]	528^2	43M	19.0G	96.9	84.0
EffNet-B7 [57]	600^2	66M	37.0G	55.1	84.3
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 [60]	224^2	22M	4.6G	940.4	79.8
DeiT-B [60]	224^2	86M	17.5G	292.3	81.8
DeiT-B [60]	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	83.3
Swin-B	384^2	88M	47.0G	84.7	84.2
(b) ImageNet-22K pre-trained models					
method	image size	#param.	FLOPs	throughput (image / s)	ImageNet top-1 acc.
R-101x3 [37]	384^2	388M	204.6G	-	84.4
R-152x4 [37]	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	86.0
Swin-L	384^2	197M	103.9G	42.1	86.4

Table 1. Comparison of different backbones on ImageNet-1K classification. Throughput is measured using the GitHub repository of [65] and a V100 GPU, following [60].

table1. Comparison of different backbones on ImageNet-1K classification. Throughput is measured using the GitHub repository of [65] and a V100 GPU, following [60].

    - 이전의 state-of-the-art Transformerbased architecture(예: DeiT[60])와 비교했을 때, Swin Transformers는 224^2를 사용한 Dei-S(79.8%)에 비해 복잡성이 비슷한 DeiT 아키텍처를 눈에 띄게 능가.
    - 최신 ConvNet, 즉 Reg-Net[47] 및 EfficientNet[57]과 비교했을 때, Swin Transformer는 speed-accuracy trade-off를 약간 더 잘 달성.
        - RegNet [47]과 EfficientNet [57]은 철저한 아키텍처 search를 통해 확보되지만 제안된 Swin Transformer는 표준 트랜스포머에서 채택되어 추가 개선 가능성이 크다는 점에 주목.
- Results with ImageNet-22K pre-training
    - ImageNet-22K에서 대용량 Swin-B 및 Swin-L도 pretrain.
        - ImageNet-1K 영상 분류에서 fine-tuned 결과는 표 1(b).
        - Swin-B의 경우 ImageNet-22K pretrain은 처음부터 ImageNet-1K에 대한 train에 비해 1.8%~1.9% 향상.
        - ImageNet-22K pre-training에 대한 이전의 최상 결과와 비교했을 때, 논문의 모델은speed-accuracy trade-offs 측면에서 훨씬 더 나음.
        - Swin-B는 86.0%의 top-1 accuracy를 얻어 inference throughput(84.7 vs. 85.9 영상/초)이 비슷한 ViT보다 2.0% 높고 FLOP(47.0G vs. 55.4G)가 약간 낮다.
        - 대형 Swin-L 모델은 86.4%의 Top-1 정확도를 달성하여 Swin-B 모델보다 약간 우수합니다.

4.2. Object Detection on COCO

- Settings
    - Object detection and instance segmentation은 118K training, 5K validation 및 20K test-dev images가 포함된 COCO 2017에서 수행.
    - validation 세트를 사용하여 ablation study가 수행되며, test-dev 시 system-level comparison가 보고.
    - ablation study를 위해, 우리는 네 가지 일반적인 object detection frameworks: Cascade Mask R-CNN [28, 5], ATSS [76], RepPoints v2 [11], and Sparse RCNN [55] in mmdetection[9].를 고려.
        - 이 네 가지 프레임워크에 대해, 우리는 same settings: multi-scale training[7, 55](짧은 쪽이 480에서 800 사이인 반면 긴 쪽이 최대 1333 사이인 입력의 크

기 조정), Adam W[43] optimizer(초기 학습 속도 0.0001, weight decay 0.05, 배치 크기 16), 3x 스케줄(36 epoches)을 활용.

- system-level comparison를 위해, 우리는 improved HTC[8] (HTC++로 표시됨), instaboost[21], 보다 강력한 multi-scale training [6], 6x schedule(72 epoch), soft-NMS[4] 및 ImageNet-22K pre-trained model을 초기화로 채택.

- 우리는 Swin Transformer를 표준 Con-vNets(예: ResNe(X)t) 및 이전 Transformer 네트워크(예: DeiT)와 비교.
- 비교는 다른 설정이 변경되지 않은 백본만 변경하여 수행.
- Swin Transformer 및 ResNe(X)t는 hierarchical feature maps으로 인해 위의 모든 프레임워크에 직접 적용할 수 있지만 DeiT는 피쳐 맵의 단일 해상도만 생성하며 직접 적용할 수 없음.
- 공정한 비교를 위해, 우리는 deconvolution 레이어를 사용하여 DeiT에 대한 hierarchical feature maps을 구성하기 위해 [78]을 따릅니다.

- Comparison to ResNe(X)t

(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

- 표 2(a)는 네 개의 object detection frameworks에 대한 Swin-T 및 ResNet-50의 결과를 나열.
- Swin-T 아키텍처는 ResNet-50에 비해 일관된 +3.4~4.2box AP 이점을 제공하며, 모델 크기, FLOP 및 대기 시간이 약간 더 큼.

(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

- 표 2(b)는 Cascade Mask R-CNN을 사용하여 서로 다른 모델 용량에서 Swin Transformer와 ResNe(X)를 비교.
- Swin Transformer는 ResNext에 비해 +3.6 box AP 및 +3.3 mask AP의 높은 detection accuracy를 달성.
- improved HTC framework를 사용하는 52.3 box AP 및 46.0 mask AP의 상위 기준에서 Swin Transformer도 +4.1 box AP 및 +3.1 mask AP에서 높습니다(표 2(c) 참조).

(c) System-level Comparison

Method	mini-val AP^{box}	mini-val AP^{mask}	test-dev AP^{box}	test-dev AP^{mask}	#param.	FLOPs
RepPointsV2* [11]	-	-	52.1	-	-	-
GCNet* [6]	51.8	44.7	52.3	45.4	-	1041G
RelationNet++* [12]	-	-	52.7	-	-	-
SpineNet-190 [20]	52.6	-	52.8	-	164M	1885G
ResNeSt-200* [75]	52.5	-	53.3	47.1	-	-
EfficientDet-D7 [58]	54.4	-	55.1	-	77M	410G
DetectoRS* [45]	-	-	55.7	48.5	-	-
YOLOv4 P7* [3]	-	-	55.8	-	-	-
Copy-paste [25]	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.

- 추론 속도와 관련하여, ResNe(X)t는 고도로 최적화된 Cudnn 기능으로 구축된 반면, Swin-transformer 는 모두 최적화되지 않은 내장 PyTorch 기능으로 구현.

    - kernel optimization는 본 논문의 범위를 벗어남.
- Comparison to DeiT
    - Cascade Mask R-CNN Framework를 이용한 DeiT-S의 성능을 표2(b)에 나타냄.
    - Swin-T의 결과는 모델 크기가 비슷한 DeiT-S보다 +2.5 box AP와 +2.3 mask AP가 높고(86M vs 80M), 추론 속도도 상당히 빠름(15.3FPS vs 10.4FPS). DeiT의 추론 속도가 낮은 것은 주로 입력 영상 크기에 대한 quadratic complexity 때문.
- Comparison to previous state-of-the-art
    - 표 2(c)는 best results를 이전 state-ofthe-art models와 비교.
    - 논문의 베스트 모델은 COCO test-dev에서 58.7 box AP 및 51.1 mask AP를 달성하여 +2.7 box AP(외부 데이터가 없는 [25] Copy-paste) 및 +2.6 mask AP(DetectorRS [45])로 이전 최고의 결과를 능가.

4.3. Semantic Segmentation on ADE20K

- Settings
    - ADE20K[80]는 널리 사용되는 semantic segmentation dataset으로, 150개의 semantic categories를 포괄. 총 25K개의 이미지를 보유하고 있으며, training 20K개, validation 2K개, testing 3K개.
    - 논문은 높은 효율성을 위한 기본 프레임워크로UperNet [66] in mmseg [15]을 활용.
    - 자세한 내용은 부록 참조.

- Results

ADE20K Method	Backbone	val mIoU	test score	#param.	FLOPs	FPS
DANet [22]	ResNet-101	45.2	-	69M	1119G	15.2
DLab.v3+ [10]	ResNet-101	44.1	-	63M	1021G	16.0
ACNet [23]	ResNet-101	45.9	38.5	-		
DNL [68]	ResNet-101	46.0	56.2	69M	1249G	14.8
OCRNet [70]	ResNet-101	45.3	56.0	56M	923G	19.3
UperNet [66]	ResNet-101	44.9	-	86M	1029G	20.1
OCRNet [70]	HRNet-w48	45.7	-	71M	664G	12.5
DLab.v3+ [10]	ResNeSt-101	46.9	55.1	66M	1051G	11.9
DLab.v3+ [10]	ResNeSt-200	48.4	-	88M	1381G	8.1
SETR [78]	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.

- 표 3은 different method/backbone pairs에 대한 mIoU, 모델 크기(#param), FLOP 및 FPS를 보여줌.
- 이 결과 비슷한 연산비용으로 Swin-S가 DeiT-S보다 +5.3mIoU(49.3 대 44.0) 높은 것으로 나타났다.
- 또한 ResNet-101보다 +4.4mIoU 높고 +2.4m입니다.ResNeSt-101[75]보다 높은 IoU.
- ImageNet-22K pre-training이 적용된 Swin-L 모델은 기존 최고 모델보다 +3.2mIoU(모델 크기가 더 큰 SETR [78])를 능가하는 53.5mIoU를 달성.

4.4. Ablation Study

이 섹션에서는 ImageNet-1K image classification, COCO object detection Cascade Mask R-CNN, semantic segmentation 시 ADE20K UperNet을 사용하여 제안된 Swin Transformer에서 중요한 설계 요소를 단순화.

- Shifted windows

- 세 가지 작업에 대한 Shifted window 접근법의 Ablations이 표 4에 보고.
- Swin-T Shifted windows partitioning은 ImageNet-1K의 경우 +1.1% top-1 accuracy, COCO의 경우 +2.8 box AP/+2.2 mask AP, AD20K의 경우 +2.8 mioU만큼 각 단계에서 single window partitioning 은 우수.
- 결과는 preceding layers에서 Shifted windows을 사용하여 windows 간 연결을 구축하는 것의 효과를 나타냄.
- 표 5와 같이 이동 창에 의한 latency overhead도 작습니다.

- 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).

- 표 4는 position embedding approaches간의 차이를 보여줌. relative position bias가 있는 Swin-T는 ImageNet-1K에서 +1.2%/+0.8%의 top-1 accuracy, COCO에서 +1.3/+1.3 mask AP에서 +1.3/+1.3 mask AP에서 +2.9mioU 및 relation to those without position encoding and with absolute position embedding에서 +2.3/+2.9mioU를 산출.
- 또한 absolute position embedding을 포함하면 영상 분류 정확도(+0.4%)가 향상되지만, object detection and semantic segmentation(COCO의 경우 0.2 box/mask AP, ADE20K의 경우 -0.6mIoU)에 해를 미칩니다.
- 최근image classification의 ViT/DeiT models의 abandon translation invariance은 오랫동안 시각적 모델링에 중요한 것으로 입증되었지만, 논문은 translation invariance을 장려하는 inductive bias이 general-purpose visual modeling, 특히

object detection and semantic segmentation의 dense prediction tasks에 여전히 선호된다는 것을 발견.

- Different self-attention methods

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 [13]	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.

  - 다양한 self-attention computation과 구현의 실제 속도를 표 5에 비교.
  - 논문의 cyclic implementation은 특히 deeper stages에서 naive padding보다 하드웨어 효율성이 더 높음.
  - 전체적으로 Swin-T, Swin-S, Swin-B에서 각각 13%, 18%의 속도 향상.
  - 제안된 shifted windows 접근 방식을 기반으로 구축된 self-attention modules은 4개의 네트워크 단계에서 sliding windows보다 각각 40.8배/2.5배, 20.2배/2.5배, 9.3배/2.1배, 7.6배/1.8배 더 효율적.
  - 전체적으로 shifted windows에 구축된 Swin Transformer 아키텍처는 Swin-T, Swin-S, Swin-B용 sliding windows에 구축된 변형 모델보다 각각 4.1/1.5, 4.0/1.5, 3.6/1.5배 더 빠름.

	Backbone	ImageNet		COCO		ADE20k
		top-1	top-5	AP^{box}	AP^{mask}	mIoU
sliding window	Swin-T	81.4	95.6	50.2	43.5	45.8
Performer [13]	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.

  - 표 6은 세 가지 작업에 대한 정확성을 비교하여 시각적 모델링에 있어 similarly accurate을 보여줌.

- 가장 빠른 트랜스포머 아키텍처 중 하나인 Performer [13]에 비해([59] 참조), 제안된 shifted windows attention computation 과 전체 Swin Transformer 아키텍처는 약간 빠르며(표 5 참조), Swin-T를 사용하는 ImageNet-1K에 비해 +2.3%의 상위 1위 정확도를 달성(표 6 참조).

5. Conclusion

- 이 논문에서는 hierarchical feature representation을 생산하고 입력 이미지 크기에 대한 linear computational complexity을 갖는 새로운 비전 Transformer인 Swin Transformer를 소개.
- Swin Transformer는 COCO Object detection와 관련하여 SOTA를 달성.
- ADE20K semantic segmentation, 이전 SOTA를 훨씬 능가.
- 다양한 비전 문제에 대한 Swin Transformer의 강력한 성능이 vision and language signals의 통일된 모델링을 촉진하기를 바람.
- Swin Transformer의 핵심 요소로서, shifted window based self-attention이 비전 문제에 효과적이고 효율적인 것으로 입증되었으며, 자연어 처리에서도 활용도를 조사할 수 있기를 기대.

A1. Detailed Architectures

	downsp. rate (output size)	Swin-T	Swin-S	Swin-B	Swin-L
stage 1	4× (56×56)	concat 4×4, 96-d, LN	concat 4×4, 96-d, LN	concat 4×4, 128-d, LN	concat 4×4, 192-d, LN
		[win. sz. 7×7, dim 96, head 3] × 2	[win. sz. 7×7, dim 96, head 3] × 2	[win. sz. 7×7, dim 128, head 4] × 2	[win. sz. 7×7, dim 192, head 6] × 2
stage 2	8× (28×28)	concat 2×2, 192-d , LN	concat 2×2, 192-d , LN	concat 2×2, 256-d , LN	concat 2×2, 384-d , LN
		[win. sz. 7×7, dim 192, head 6] × 2	[win. sz. 7×7, dim 192, head 6] × 2	[win. sz. 7×7, dim 256, head 8] × 2	[win. sz. 7×7, dim 384, head 12] × 2
stage 3	16× (14×14)	concat 2×2, 384-d , LN	concat 2×2, 384-d , LN	concat 2×2, 512-d , LN	concat 2×2, 768-d , LN
		[win. sz. 7×7, dim 384, head 12] × 6	[win. sz. 7×7, dim 384, head 12] × 18	[win. sz. 7×7, dim 512, head 16] × 18	[win. sz. 7×7, dim 768, head 24] × 18
stage 4	32× (7×7)	concat 2×2, 768-d , LN	concat 2×2, 768-d , LN	concat 2×2, 1024-d , LN	concat 2×2, 1536-d , LN
		[win. sz. 7×7, dim 768, head 24] × 2	[win. sz. 7×7, dim 768, head 24] × 2	[win. sz. 7×7, dim 1024, head 32] × 2	[win. sz. 7×7, dim 1536, head 48] × 2

Table 7. Detailed architecture specifications.

- 자세한 아키텍처 사양은 모든 아키텍처에 대해 224×224의 입력 이미지 크기를 가정하는 표 7에 나와 있음.
- “Concat $n \times n$”은 패치에서$n \times n$이웃 피쳐의 연결을 나타냄. 이 작업을 수행하면 feature map의 다운샘플링 속도가 n.

- "96-d"는 출력 dim이 96인 Linear layer를 나타냄. "win. sz. 7 × 7"은 window size가 7 × 7인 multi-head self-attention (MSA) module을 나타냄.

A2. Detailed Experimental Settings

A2.1. Image classification on ImageNet-1K

- image classification는 마지막 단계의 출력 피쳐 맵에 global average pooling layer를 적용한 다음 linear classifier를 적용하여 수행.
- 이 전략이 ViT[19]와 DeiT[60]에서와 같이 additional class 토큰을 사용하는 것만큼 정확하다고 생각함.
- 평가에서 single crop를 사용한 top-1 accuracy 가 보고.

- Regular ImageNet-1K training
  - 대부분의 training settings은 [60]을 따름.
  - 모든 모델 변형에 대해 기본 입력 이미지 해상도224^2를 채택. 384^2와 같은 다른 해상도의 경우 GPU 소비를 줄이기 위해 처음부터 교육하는 대신 224^2 해상도로 교육된 모델을 fine-tune.
- 224^2 input 으로 처음부터 training할 때, 20개의 epochs of linear warm-up이 있는 cosine decay learning rate scheduler를 사용하여 300개 epoch에 AdamW[36] 최적화 사용.
- batch size 1024, initial learning rate 0.001, weight decay 0.05 및 max norm 1의 gradient clipping이 사용됨.
- RandAugment [16], Mixup [74], Cutmix [72], random erasing [79], stochastic depth [34]를 포함하여 대부분의 augmentation 및 정규화 전략을 training에 포함하지만 repeated augmentation[30] 및 Exponential Moving Average[44]은 포함하지 않음.
- 이는 ViT 훈련을 안정화하기 위해 repeated augmentation가 중요하다는 점과 반대되는 것에 유의.
- 대형 모델(예: 각각 0.2; 0.3; 0.5), SWin-T, SWin-S 및 SWin-B의 경우 0.5)에 stochastic depth augmentation가 사용된다.
- 분해능이 더 큰 입력에 대한 미세 조정을 위해, 확률적 깊이 비율을 0.1로 설정하는 것을 제외하고, 10-5의 일정한 학습 속도, 10-8의 체중 감소, 첫 번째 단계와 동일한 데이터

확대 및 정규화의 30개 에포크에 대해 Adam W[36] 최적화 장치를 사용한다.

- ImageNet-22K pre-training
  - 또한 14.2 million 개의 이미지와 22K 클래스를 포함하는 대규모 ImageNet-22K 데이터셋에 대해 pre-training.
  - training은 두 단계로 진행.
    - 224^2입력의 첫 번째 단계에서는 5 epochs linear warm-up을 사용하는 linear warm-up scheduler를 사용하여 60epochs에 AdamW optimizer를 사용.
    - 배치 크기 4096, 초기 학습률 0.001 및 weight decay 0.01이 사용.
    - $224^2/384^2$ 입력으로 ImageNet-1K finetuning의 두 번째 단계에서는 배치 크기 1024, 일정한 학습 속도 10^-5, weight decay 10^-8의 30epoch 모델을 training.

A2.2. Object detection on COCO

- ablation study의 경우, 네 가지 일반적인 개체 탐지 프레임워크인 Cascade Mask R-CNN [28, 5], ATSS [76], RepPoints v2 [11], and Sparse RCNN [55] in mmdetection [9]을 고려합니다.
- 이 네 가지 프레임워크에 대해, 우리는 동일한 설정을 활용:multi-scale training [7, 55] (짧은 쪽이 480에서 800 사이인 반면 긴 쪽이 최대 1333 사이인 입력의 크기 조정), Adam W[43] optimizer(초기 학습률 0.0001, weight decay 0.05, 배치 크기 16), 3x schedule(learning rate decayed 36 epochs 27과 33에서 10배 증가).
- 시스템 수준 비교를 위해, 우리는 향상된HTC[8] (HTC++로 표시), instaboost [21], 보다 강력한 multi-scale training[6] (짧은 쪽이 400~1400 사이인 반면 긴 쪽이 최대 1600까지), 6x schedule(72 epochs(63~69에 학습 속도가 0.1배 감소), soft-NMS [4], 마지막 단계의 출력과 ImageNet-22K pre-trained 모델을 초기화.
- 모든 Swin Transformer 모델에 대해 0.2의 비율로 stochastic depth를 채택

A2.3. Semantic segmentation on ADE20K

- ADE20K[80]는 널리 사용되는 semantic segmentation dataset으로, 150개의 semantic categories를 포괄
- 총 25K개의 이미지를 보유하고 있으며, training 20K개, validation 2K개, testing 3K개.
- 높은 효율성을 위한 기본 프레임워크로 UperNet[66] in mm segmentation[15]을 활용합니다.

- training 에서는 initial learning rate이 6 × 10-5, weight decay가 0.01, linear learning rate decay를 사용하는 scheduler 및 1,500회 반복의 linear warmup을 사용하는 AdamW[43] 최적화기를 사용.
- 모델은 GPU당 이미지 2개가 포함된 8개의 GPU에서 160K회 반복 training 을 받음.
- augmentations의 경우, random horizontal flipping의 mmsegmentation, [0.5, 2.0] 비율 범위 내 random re-scaling 및 random photometric distortion의 기본 설정을 채택.
- 모든 SwinTransformer 모델에 0.2의 Stochastic depth가 적용.
- Swin-T, Swin-S는 512×512의 입력으로 이전 접근 방식에 따라 standard setting에 대한 trained. ‡ 가 있는 Swin-B 및 Swin-L은 이 두 모델이 ImageNet-22K에서 pre-trained되었으며 640×640의 입력으로 trained되었음을 나타냄.
- inference에서는 training 에 사용되는 resolutions의 [0.5, 0.75, 1.0, 1.25, 1.5, 1.75]× 을 사용한 multi-scale test가 사용.
- test scores를 보고할 때, common practice에 따라 training images and validation images이 모두 training 에 사용[68].

A3. More Experiments

A3.1. Image classification with different input size

	Swin-T		Swin-S		Swin-B	
input size	top-1 acc	throughput (image / s)	top-1 acc	throughput (image / s)	top-1 acc	throughput (image / s)
224^2	81.3	755.2	83.0	436.9	83.3	278.1
256^2	81.6	580.9	83.4	336.7	83.7	208.1
320^2	82.1	342.0	83.7	198.2	84.0	132.0
384^2	82.2	219.5	83.9	127.6	84.2	84.7

Table 8. Swin Transformers with different input image size on ImageNet-1K classification.

- 표 8은 224^2부터 384^2까지의 다양한 입력 이미지 크기를 가진 Swin Transformer의 성능을 보여줍니다. 일반적으로 입력 resolution이 클수록 top-1 accuracy는 향상되지만 추론 속도는 느려짐.

A3.2. Different Optimizers for ResNe(X)t on COCO

Backbone	Optimizer	AP^{box}	AP^{box}_{50}	AP^{box}_{75}	AP^{mask}	AP^{mask}_{50}	AP^{mask}_{75}
R50	SGD	45.0	62.9	48.8	38.5	59.9	41.4
	AdamW	46.3	64.3	50.5	40.1	61.7	43.4
X101-32x4d	SGD	47.8	65.9	51.9	40.4	62.9	43.5
	AdamW	48.1	66.5	52.4	41.6	63.9	45.2
X101-64x4d	SGD	48.8	66.9	53.0	41.4	63.9	44.7
	AdamW	48.3	66.4	52.3	41.7	64.0	45.1

Table 9. Comparison of the SGD and AdamW optimizers for ResNe(X)t backbones on COCO object detection using the Cascade Mask R-CNN framework.

- 표 9는 COCO 객체 감지 시 ResNe(X)t 백본의 AdamW 및 SGD 최적화기를 비교.
- 이 비교에는 Cascade Mask R-CNN 프레임워크가 사용. SGD는 Cas-cade Mask R-CNN 프레임워크의 기본 optimizer로 사용되지만, 일반적으로 특히 작은 백본의 경우 AdamW optimizer로 교체하여 정확도가 향상되는 것을 관찰.
- 따라서 제안된 Swin Transformer 아키텍처와 비교할 때 AdamW for ResNe(X)t 백본을 사용.

Q. 기존 선행논문인 ViT에 있는 [CLS] Token 은 어디?