An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale

(ViT)

Samsung Software Developer Community
Korea Vision & Robotics
HoChan Jeong

2023.09.23

Contents

1. Background

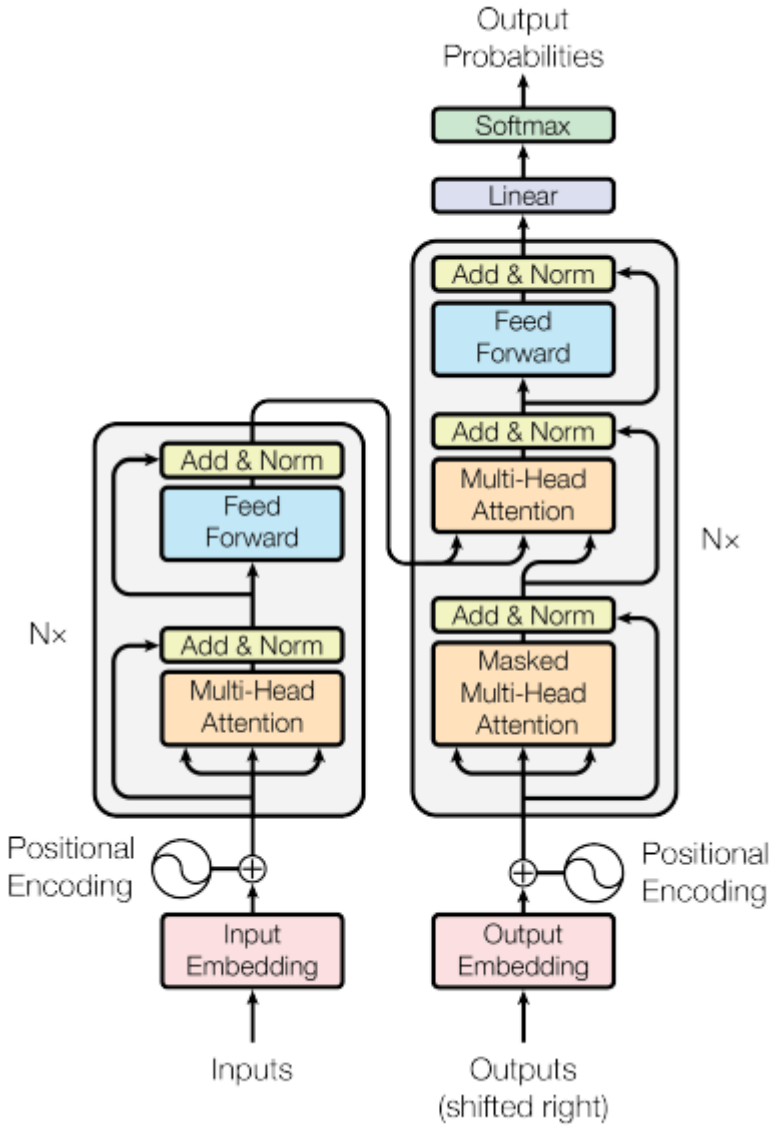
2017 – '**Attention All You Need**'

: 해당 논문에서 제안한 `Transformer`가 이후 NLP 분야에서 사실상의 Base Model 성장

- Transformer의 Attention (self attention + cross attention) 방법을 통해서
기존 RNN 구조의 문제점 해결

- 효율적인 연산을 통해서 GPT, Bert 같은 큰 규모의 모델
이를 fine-tune 해서 다양한 task에 활용

- 사실상 Transformer 이후, NLP 분야에서는 Transformer라는 패러다임이 정착

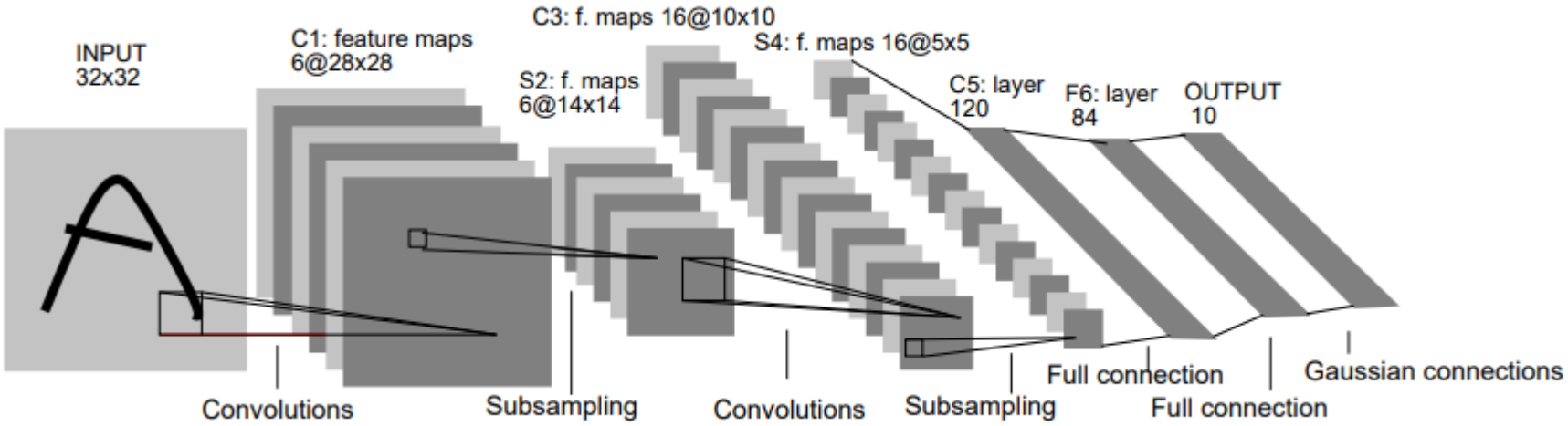


1. Background

간단하게 복습...

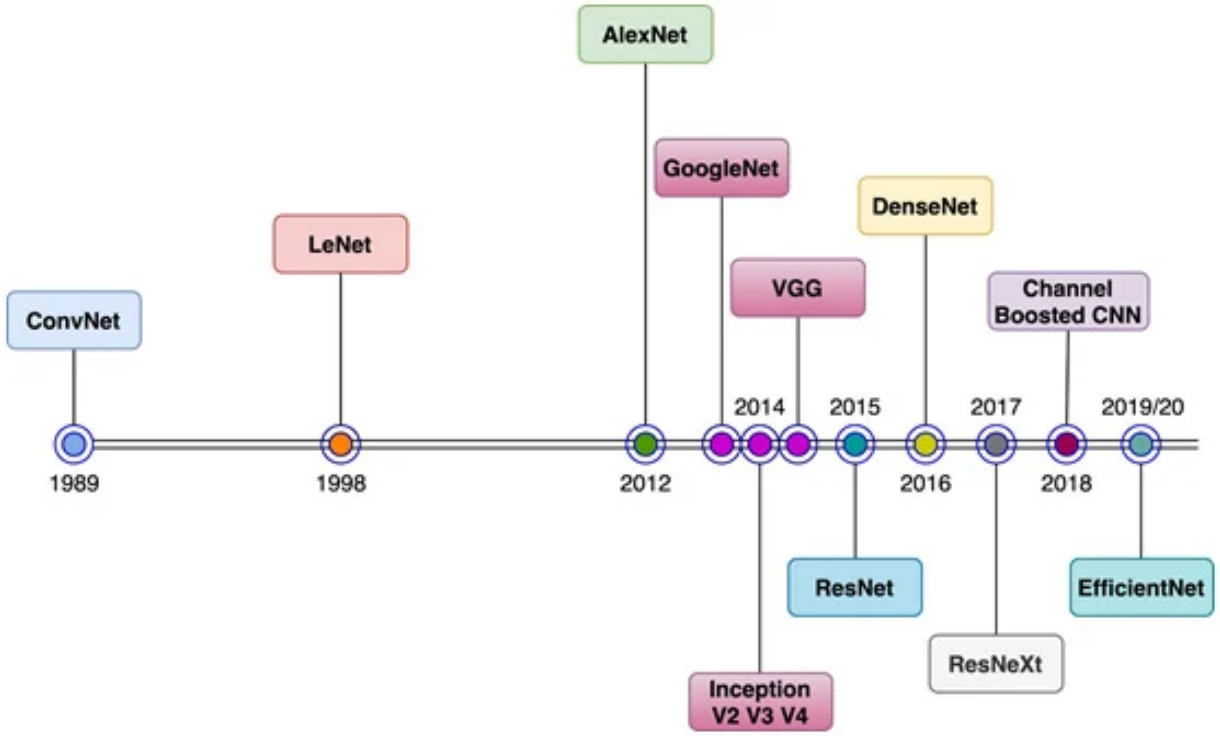
기존 CV의 지배적인 방법

Computer Vision



Convolution Neural Networks (CNN)

1. Background



- AlexNet을 시작으로, ResNet, DenseNet, EfficientNet 모두 CNN 구조를 Base로

- 해당 방법들의 단점, 모델의 깊이가 깊어지면 성능향상의 한계점, 포화지점이 있음

NLP에서 attention 기법은 모델의 size를 늘려도 성능포화의 모습을 보이지 않았음

-> **따라서 이를 cv에 적용해 볼 수는 없을까**?

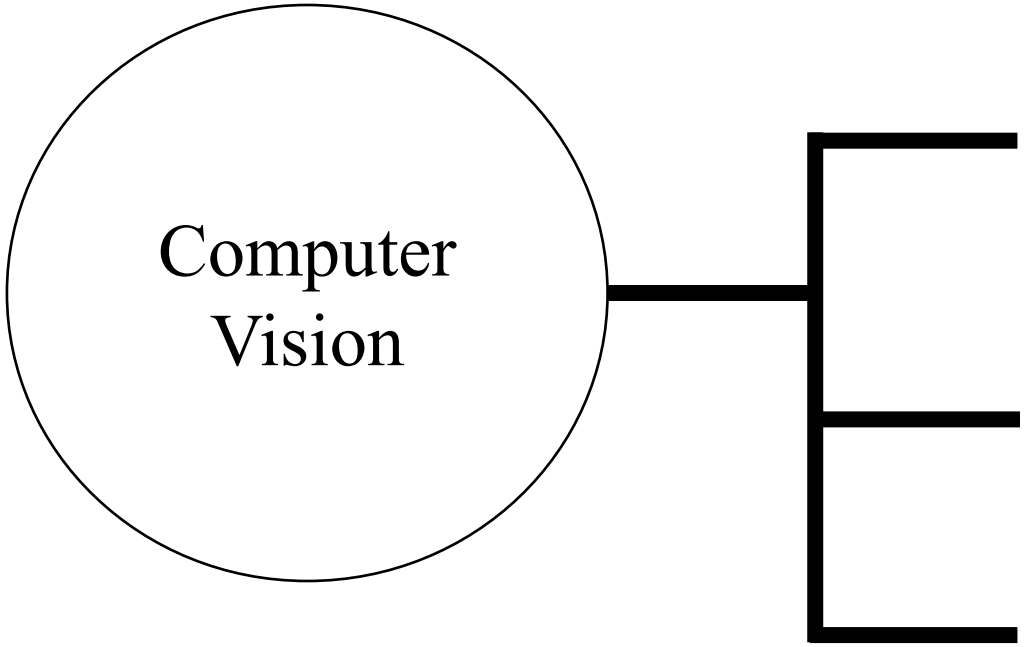
1. Background

Transformer (NLP)

Computer Vision

- NLP에서의 Transformer의 성공에 힘입어,
CV 분야에서도 Transformer를 적용하려는 시도가 계속 되었다

1. Background



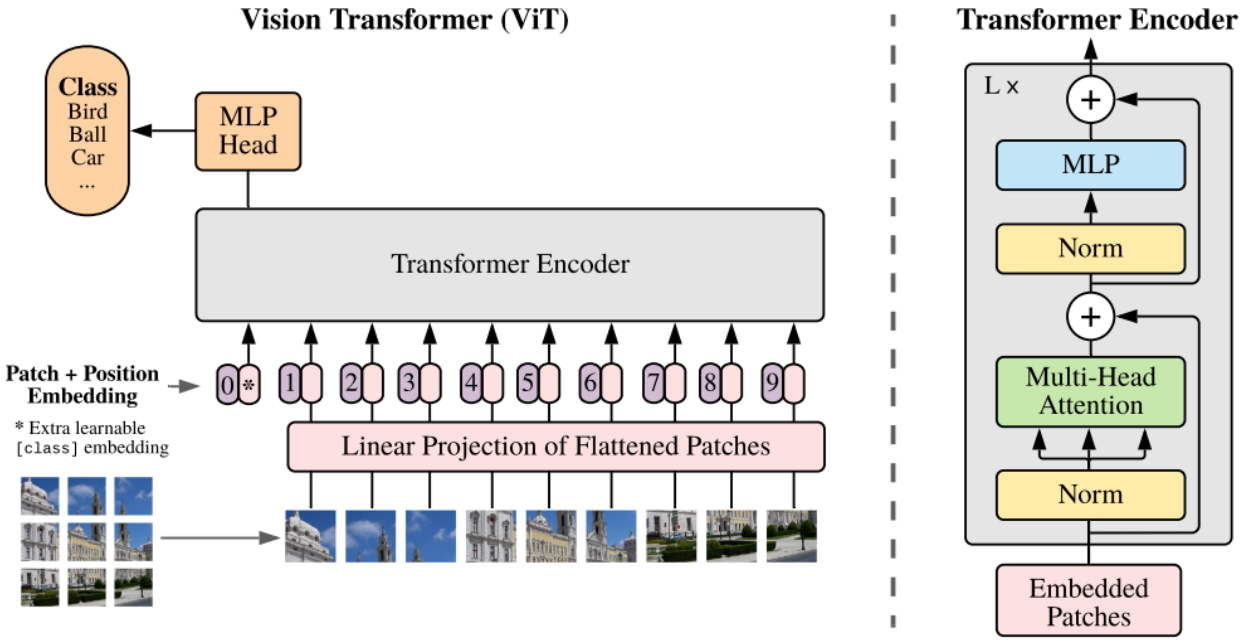
이미지의 모든 pixel에 대해 attention 수행
: 저해상도 이미지는 가능, 고해상도 : 기하급수적인 연산량
(하드웨어 혁명 없이 사실상 불가능 방법)

Convolution 연산을 Attention으로 대체해서 근접한 이웃에 대해서만 Attention을 수행 (사용하기 어려움- CNN 대체)

CNN의 output을 Attention 하는 후처리 방법
(CNN과 결합)

ViT 등장 이전 attention 방법을 CV에 적용한 방법들

1. Background



- 오늘 살펴볼 ViT의 경우 input 이미지를 고정된 patch size로 split
- 그리고 이렇게 나눈 이미지를 flatten하여 linear projection 해주고
- transformer 와 같이 position embedding 해준다음
- Transformer Encoder와 똑같은 구조를 활용해서 Attention 수행
-이후 Classification의 경우 해당 값을 MLP Head를 통과하여 분류

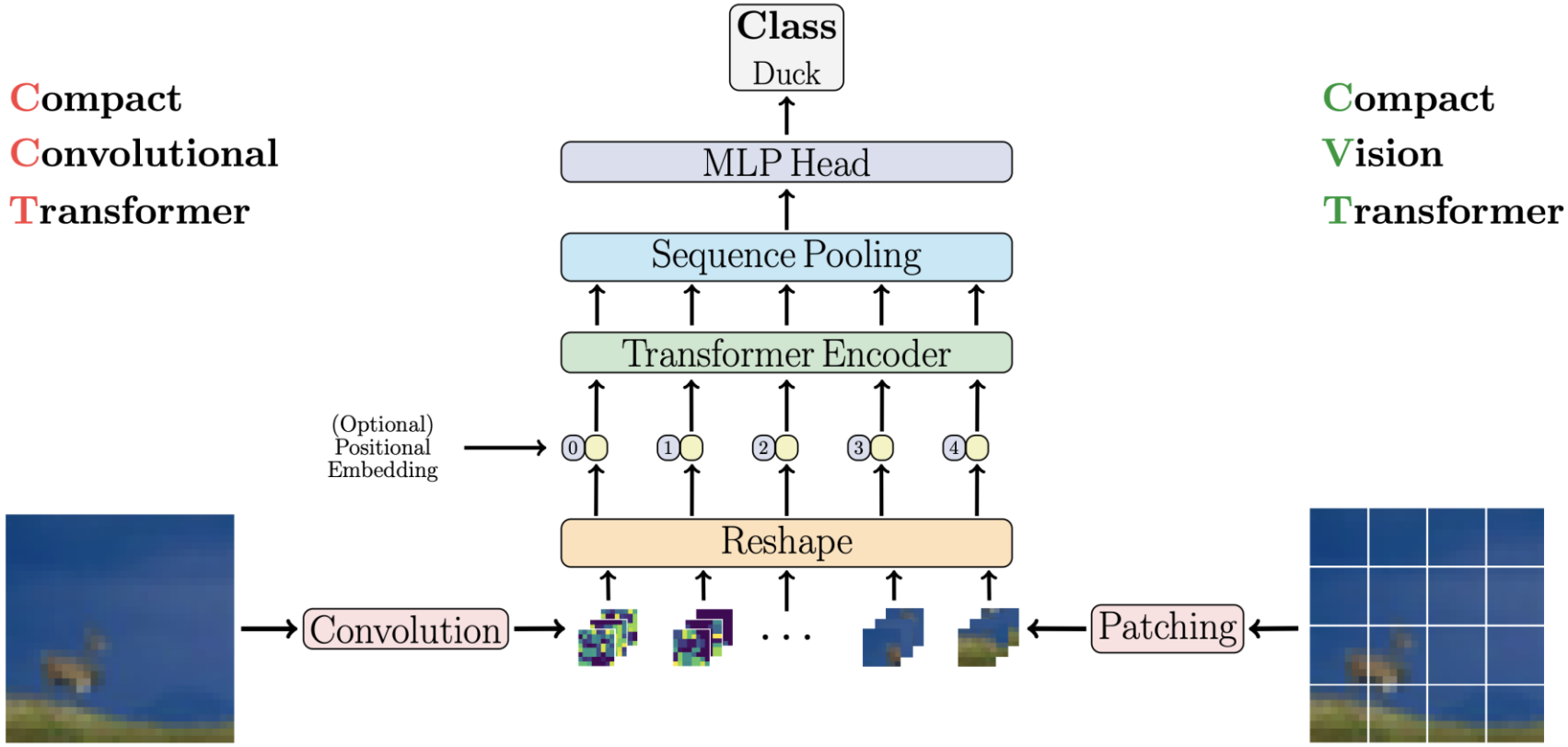
쉽게 이해 : 이미지를 Patch 단위로 나누는데 이는 NLP 분야에서 단어단위로 나눈 tokens과 같은 역할

2. ViT Model – Forward 전체 과정

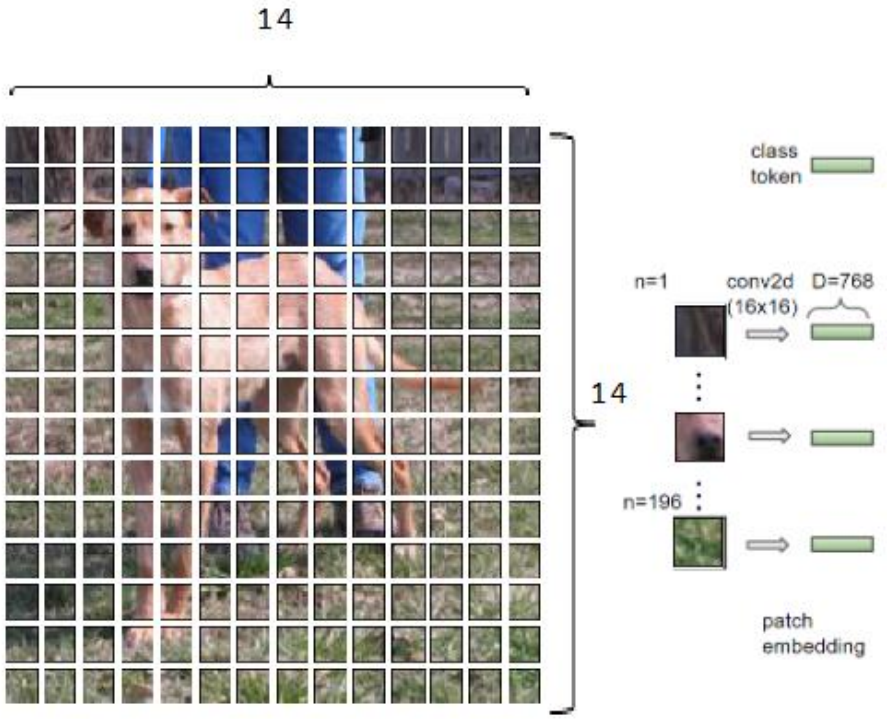
2. ViT Model – Patch &Patch Embedding

Compact
Convolutional
Transformer

Compact
Vision
Transformer



2. ViT Model – Patch &Patch Embedding

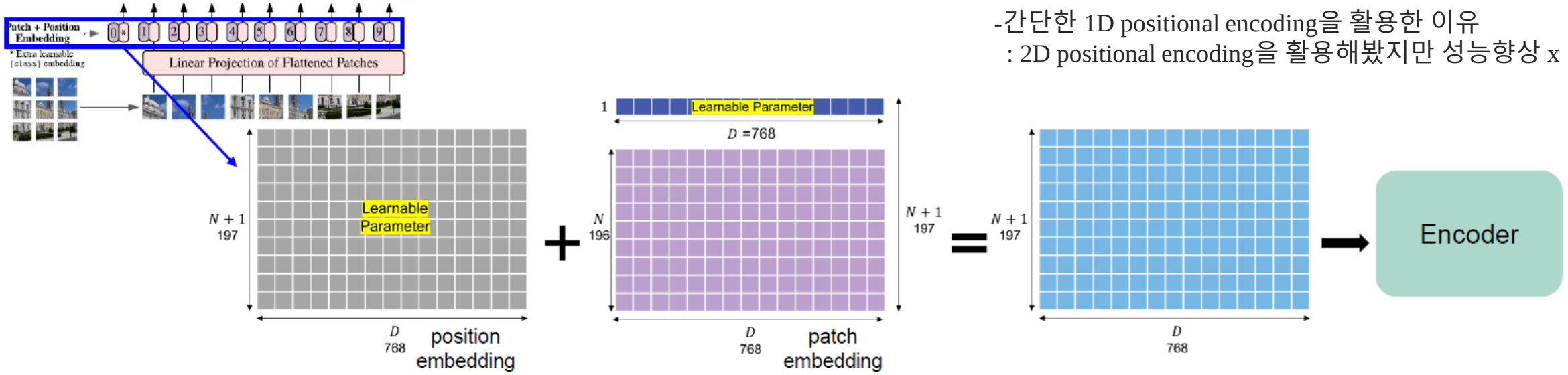


D차원으로
Linear Projection
: 쉽게 말해서 Linear Layer 통과

Input Image
(C, H, W)

(Ph, Pw) 사이즈로 split – N개로 Split (N = HW/P^2)
: 논문에서는 base로 16x16이지만, 14, 32등 다양한 사이즈 이용가능

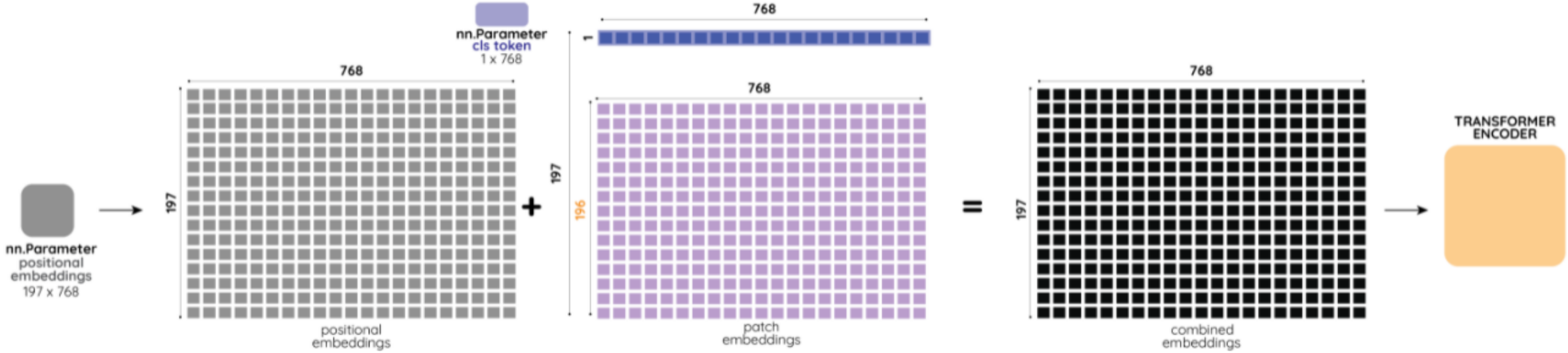
2. ViT Model – Positional Encoding



-간단한 1D positional encoding을 활용한 이유
: 2D positional encoding을 활용해봤지만 성능향상 x

- 이미지를 flatten 했기 때문에 공간정보가 사라진 상태
- 따라서 patch에 순서 정보를 주는 positional encoding을 해준다
- 방법은 간단 : 해당 (N+1)(D) 만큼의 학습가능한 파라미터를 만들어서 더해주면 된다
- 이때 왜 N+1 인가 ? : **- Class token 때문**

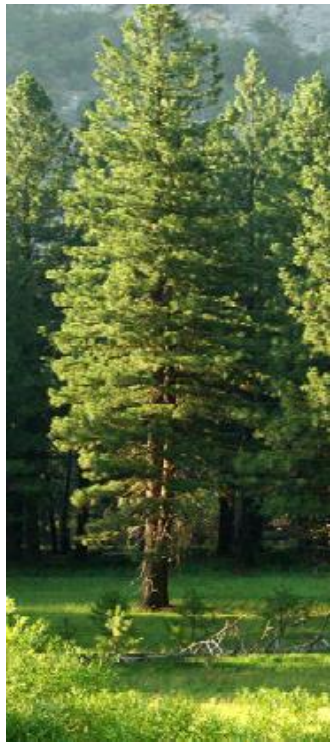
2. ViT Model – Class Token



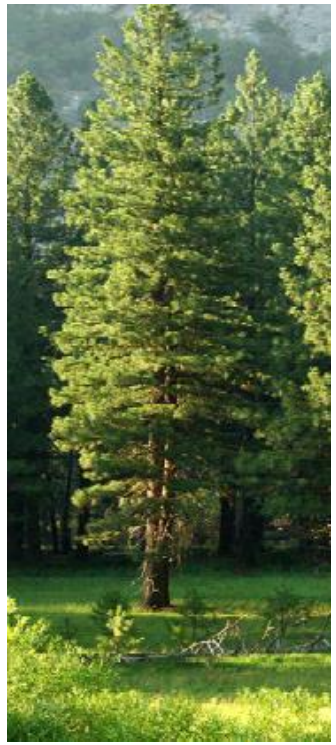
- Bert의 class token과 같은 역할
- 이미지 전체의 representation을 나타내는 token의 역할 – 이후 나올 inductive bias와 연관

2. ViT Model – Class Token

0 (CLS)



1

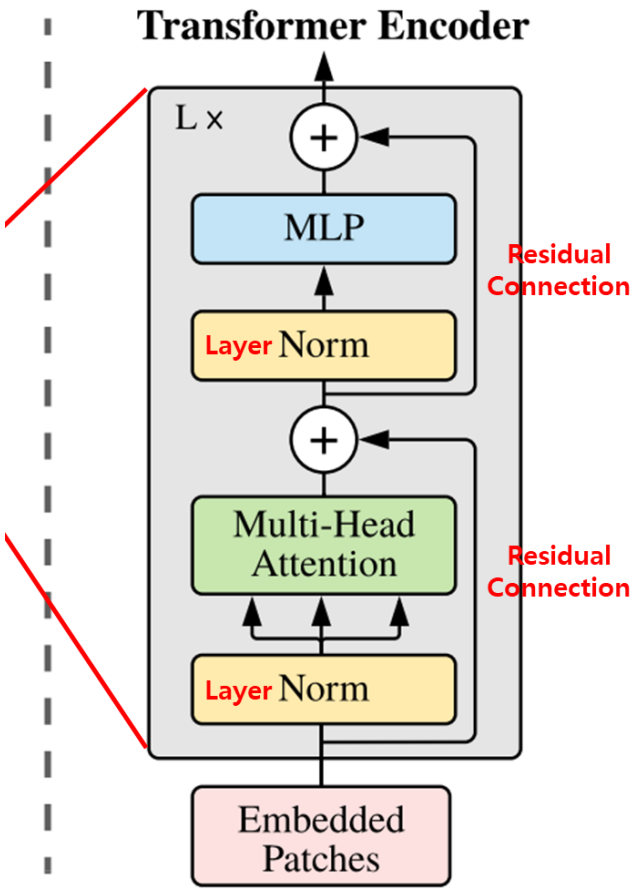


2

...

-Attention 수행 시 – 각 나무들의 관계는 잘 파악하지만 전체 숲과의 관계는 파악 x (만약 cls가 없다면)
-이를 해결하기 위해서 전체 이미지를 표현하는 CLS를 삽입

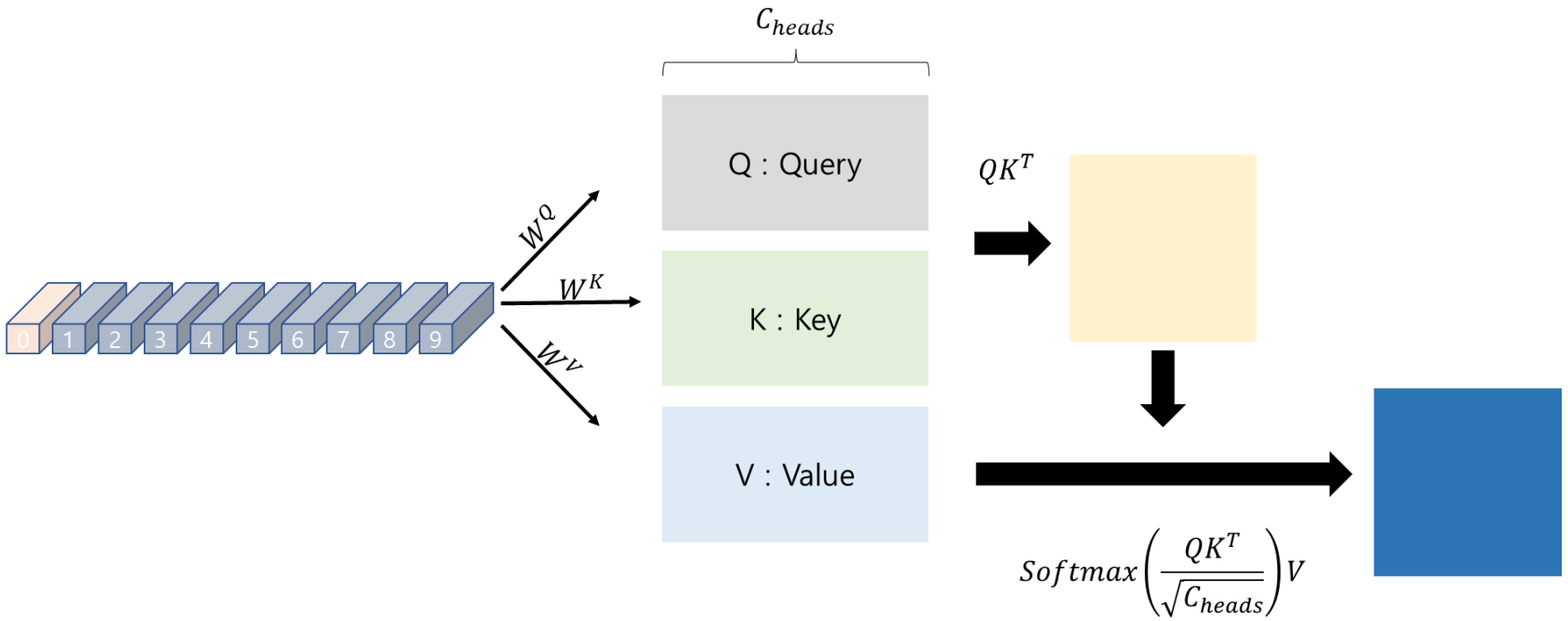
2. ViT Model – Transformer Encoder



- 앞에서 Encoding 한 input을 Encoder를 통과시키면서 attention 연산을 수행
- 이때 구조는 Transformer의 Encoder와 동일한 구조
- 따라서 이는 self-attention
- 똑같이 Residual connection, Layer Norm 적용

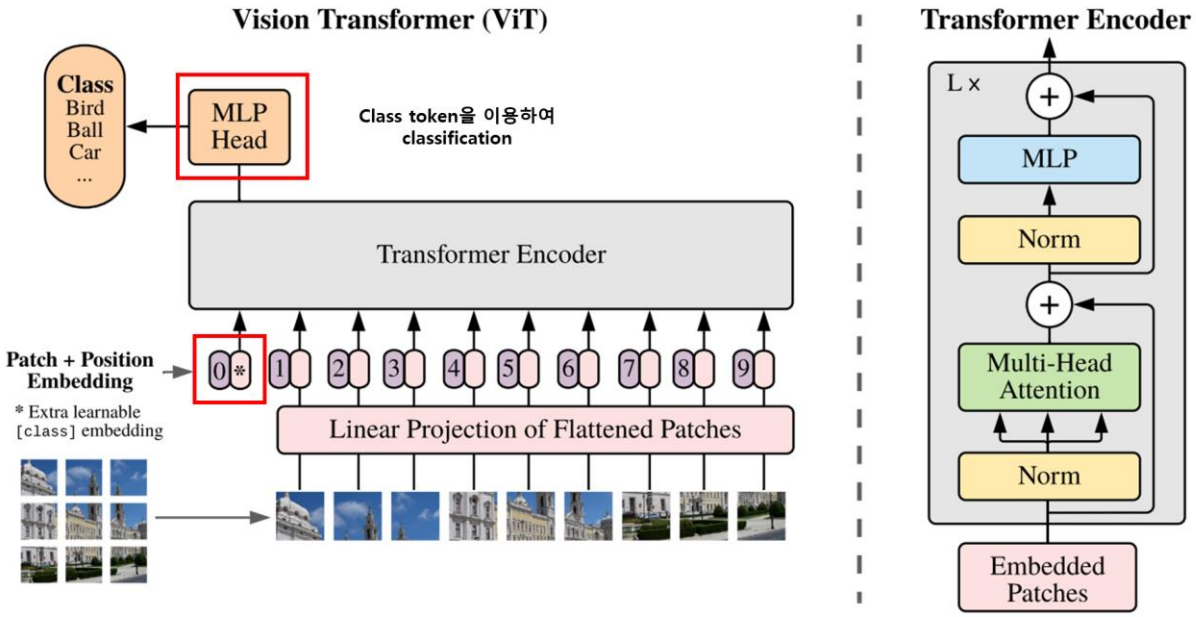
2. ViT Model – Transformer Encoder

- 혹시 Transformer의 attention 연산 과정을 잊어버린 사람을 위해서…



https://nlpinkorean.github.io/illustrated-transformer/

2. ViT Model – MLP Head



- 그리고 CLS를 활용해서 Classification인 경우 MLP를 통과해서 Class를 분류

2. ViT Model – Code

https://hongl.tistory.com/235

3. Paper +

- 이런 ViT 에는 단점이 존재

- 기존 CNN 보다 낮은 **Inductive bias**을 갖는다 – 여기서 inductive bias가 낮다는 것은 편견이 적은 모델로 이해

- 즉, 일반화될 가능성이 높아지는 것이다

- 하지만 이는 적은 데이터만을 사용할 때 문제가 발생하는데, 적은 데이터로는 문제를 풀 수 없게 된다
- Trad off 관계에 있다 (일반화 – 특정 task 성능)

- **따라서 많은 양의 데이터가 필요하다**

-이러한 낮은 inductive bias의 경우 patch 단위로 이미지를 나눌 때 이미지의 공간 정보가 손실
-이를 해결하기 위해서 positional encoding을 해주만 그래도 절대적 공간 정보 상실에서 기인한다

- 따라서 이러한 문제 때문에 데이터가 적은 경우 Backbone으로 사용하기가 어려움

3. Paper +

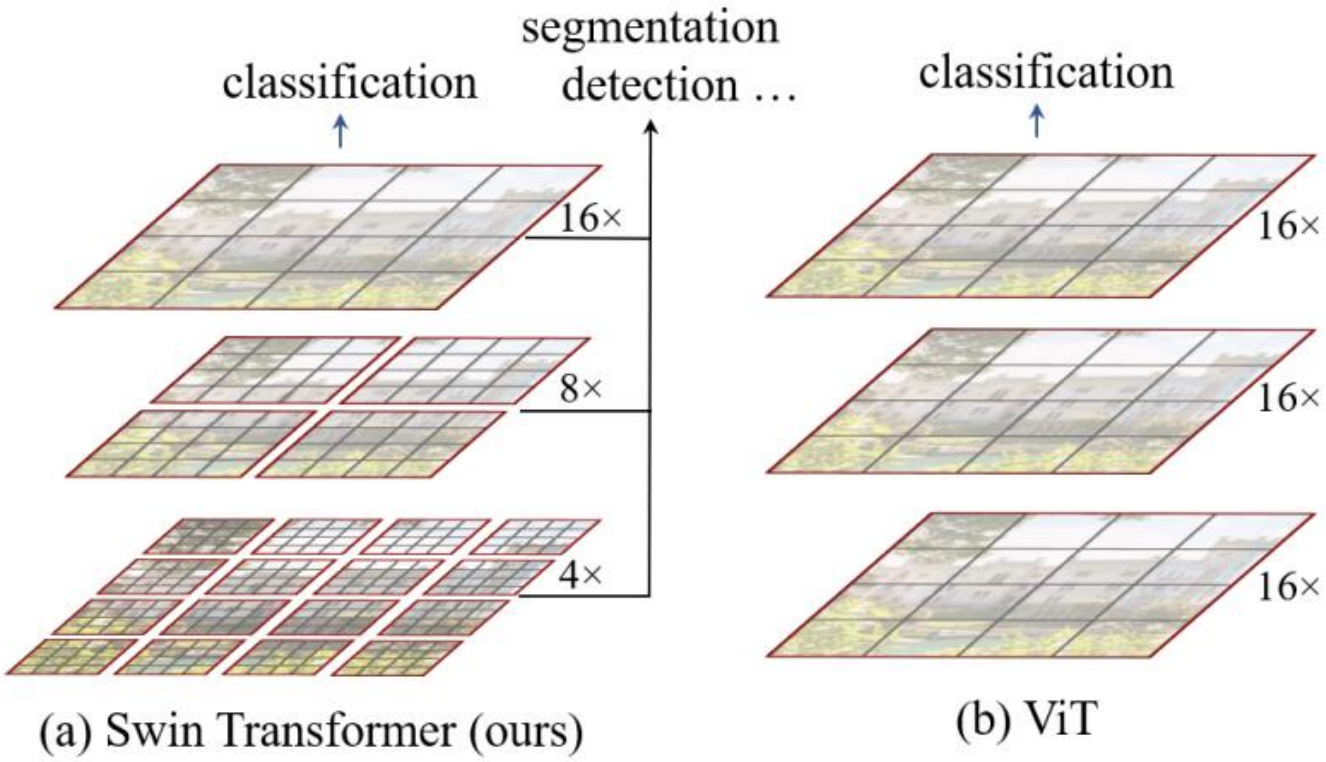
- 또한 patch 단위로 이미지를 요약? 하기때문에 공간 정보가 상실되는 문제 때문에

- Object detection과 같은 task에서 작은 객체를 탐지 못한다는 문제가 발생
- 또한 Dense map을 예측해야하는 Segmentation 같은 task에서도 문제가 발생

- 다양한 task에 대한 backborn의 성능이 떨어지게 된다 … - 물론 Data를 추가해줌으로 (엄청큰 데이터셋으로) 해당 문제를 해결 해볼 수 있지만 – 효율적이지 못하다

- 이를 해결하기 위해서 등장한 모델이 바로 - **`Swin Transformer`**

3. Paper +



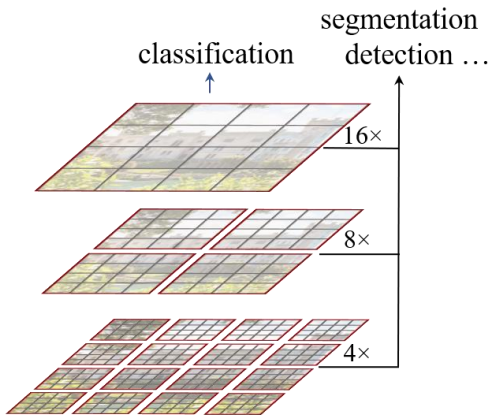
- 계층적인 구조를 활용

- 이를 통해 공간적인 정보를 잘 보존

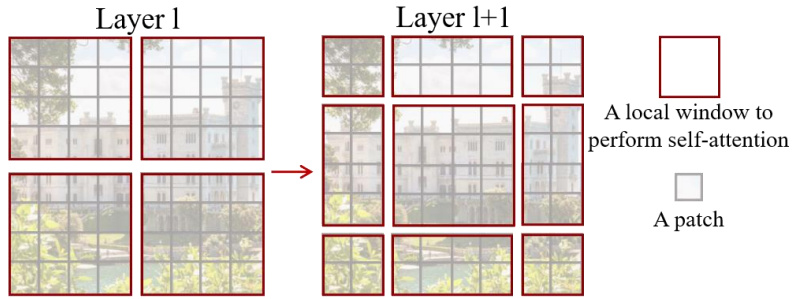
- 적은 데이터에서도 잘 작동
(다른 CNN 모델 보다 좋은 성능)

- 따라서 효율적인 Backbone 모델로 활용가능

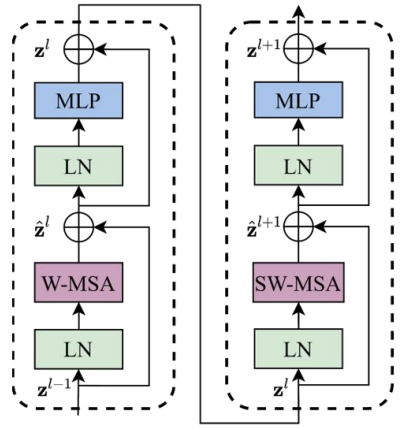
그래서 다음 논문 리뷰는 …



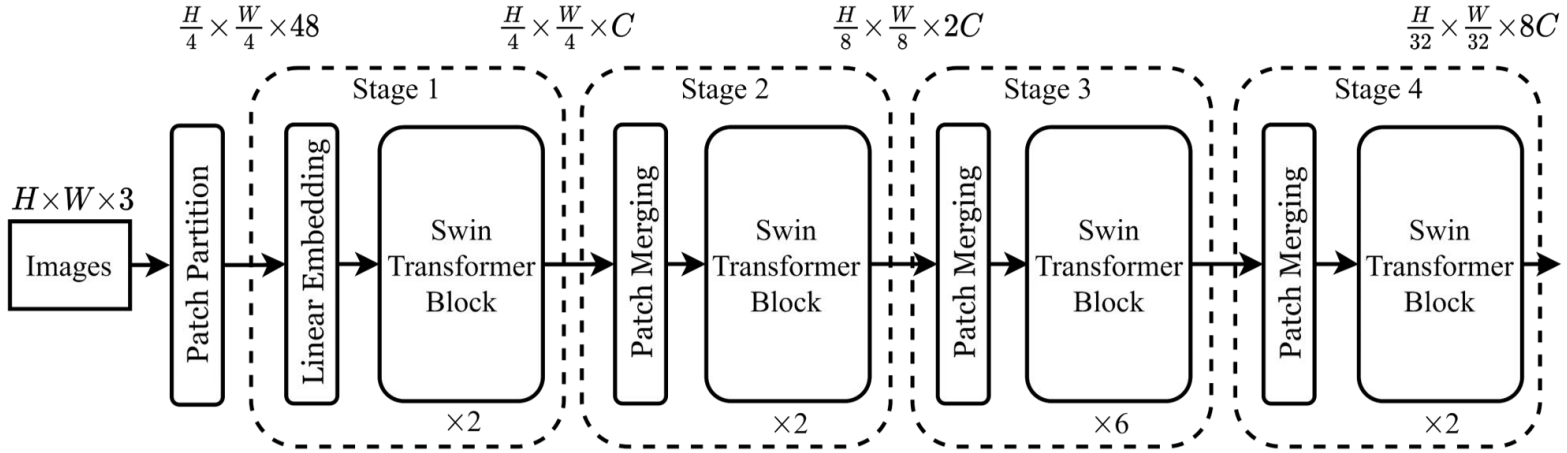
(a) Swin Transformer



(b) Shifted Window



(c) Two Successive Swin Transformer Blocks



(d) Architecture