Vision Transformer

An Image is Worth 16x16 Words: Transformers For Image Recognition At Scale

이찬호
국방인공지능융합연구소

Vision Transformer – Introduction

- **구글**이 개발하여 **2021년** ICLR에서 발표
- **Image Classification**에 **Transformer 구조**를 성공적으로 **적용**
- Transformer 구조의 **NLP task**에서 **성공**
- **높은 연산 효율성**과 **확장성**을 보임
- 데이터 셋과 모델 크기가 계속 커져도 모델 성능이 포화되지 않고 **지속적으로 개선 및 발전**
- **Computer Vision**(이하 'CV') Task에서는 **제대로 적용되지 않음**
- **Self-Attention을 추가한 CNN 계열 모델 구조**와 아예 **Convolution을 attention으로 대체한 구조**들이 제안됨.
- 이론상으로 효율성이 보장되었지만, 실 적용시 호환성이 떨어짐

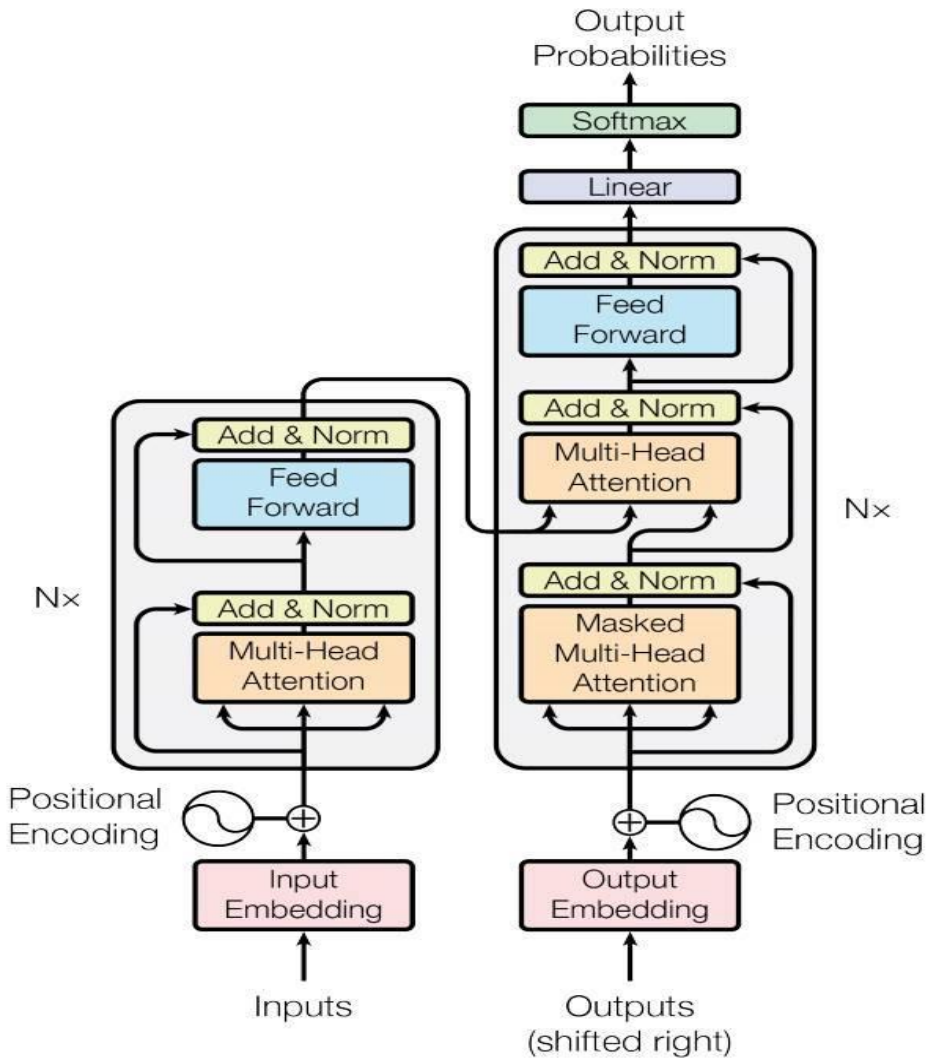


Fig 1. Transformer

Vision Transformer – Introduction

- 이미지 자체를 **Standard Transformer**에 직접적으로 넣어주는 형태로 모델 구성
- 이를 위해 이미지를 **패치 단위**로 쪼개 **'토큰화'**
- **패치**에 **Linear embedding 적용** 후, **Sequence로 만들어둔 형태**로 Transformer 입력
- **중간 사이즈**의 데이터 셋에 학습 시, **별도의 강력한 Regularization 없이 기존 ResNet 대비 나은 성능을 보이지 못함**
- 이는 Transformer 구조 자체가 CNN 구조에 비해 **Inductive bias가 부족해 많은 양의 데이터 없이 일반화가 제대로 이뤄지지 않음**
- **큰 사이즈** 데이터 셋(1,400만 ~ 3억장)에서 학습 시 위 구조적 한계(Inductive bias) **극복 가능**
- **ViT**는 충분한 크기의 데이터 셋(ImageNet-21K or JFT-300M)에서 **사전 학습 후**, 적은 데이터셋(ImageNet, ImageNet-Real, CIFAR-100, VTAB)을 가진 Task에 **전이 학습을 시킬 때 좋은 성능**을 보임

Vision Transformer – Introduction

- 이미지 자체를 **Standard Transformer**에 직접적으로 넣어주는 형태로 모델 구성
- 이를 위해 이미지를 **패치 단위**로 쪼개 **'토큰화'**
- **패치**에 **Linear embedding 적용** 후, **Sequence로 만들어둔 형태**로 Transformer 입력
- **중간 사이즈**의 데이터 셋에 학습 시, **별도의 강력한 Regularization 없이 기존 ResNet 대비 나은 성능을 보이지 못함**
- 이는 Transformer 구조 자체가 CNN 구조에 비해 **Inductive bias가 부족해 많은 양의 데이터 없이 일반화가 제대로 이뤄지지 않음**
- **큰 사이즈** 데이터 셋(1,400만 ~ 3억장)에서 학습 시 위 구조적 한계(Inductive bias) **극복 가능**
- **ViT**는 충분한 크기의 데이터 셋(ImageNet-21K or JFT-300M)에서 **사전 학습 후**, 적은 데이터셋(ImageNet, ImageNet-Real, CIFAR-100, VTAB)을 가진 Task에 **전이 학습을 시킬 때 좋은 성능**을 보임

Vision Transformer – Related Work

- Self-Attention을 이미지에 단순 적용하는 방식은 해당 픽셀과 모든 픽셀 간의 Attention Weight를 구해야하기 때문에 계산 비용이 픽셀 개수 n에 대해 $O(n^2)$ 복잡도를 가짐
- Parmar et al. (2018): Local Neighborhood에만 Self-Attention을 적용
- Sparse Transformers (2019), Weissenborn et al. (2019): Attention의 범위를 Scaling하는 방식으로 Self-Attention을 적용

→ 이런 방식의 Attention의 경우 하드웨어 가속기에서 연산을 효율적으로 수행하기에는 다소 번거로운 작업 포함

- Cordonnier et al. (2020): 이미지를 2x2의 패치로 나눈 뒤, Self-Attention을 적용
- Image GPT (Chen et al., 2020): 이미지 해상도와 Color Space를 줄인 후, 이미지 픽셀 단위로 Transformer를 적용한 생성 모델
- 기존 연구 대비 ViT의 차별점
  - 표준 ImageNEt 데이터 셋보다 더 큰 크기의 데이터 셋에서 Image Recognition 실험을 진행
  - 더 큰 크기의 데이터 셋에서 학습시켜 기존 ResNet 기반 CNN보다 더 좋은 성능을 냄

Vision Transformer – Method

- Before Input
  - (C, H, W)크기의 이미지를 크기가 (P, P)인 패치 n개로 분할 후, n개의 1D 벡터($P^2 \cdot C$차원)로 Flatten
  - $N = \frac{HW}{P^2}$로 계산되며, P는 하이퍼 파라미터
  - 실험에서는 모델 크기에 따라 $P = 14, 16, 32$ 등 다양함
  - 이후, Linear Projection을 수행하여 크기가 D인 벡터의 시퀀스로 차원 변경
    - 이 때, D는 모든 레이어 전체 고정 값.

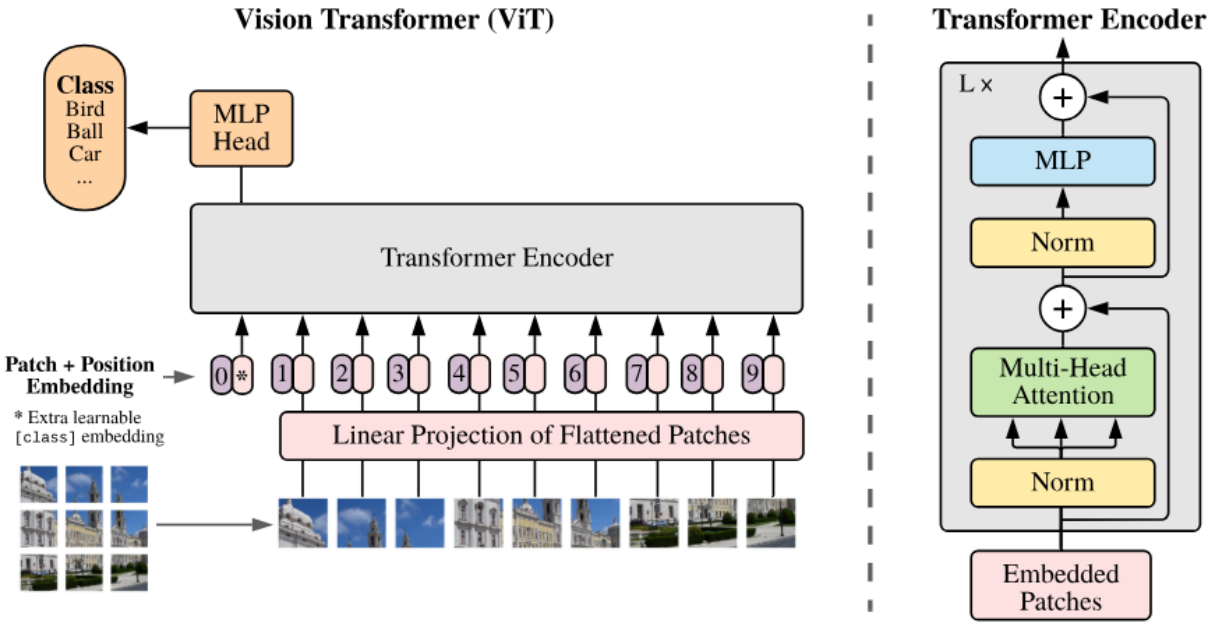


Fig 2. ViT Architecture

Vision Transformer – Method

- Embedding (P=16, H,W=224, N=196, D=768)
  - 가장 첫 패치 Embedding 앞에 학습 가능한 Embedding 벡터를 붙임
    (추후 이미지 전체에 대한 표현을 나타내게 됨)
  - $N+1$개의 학습 가능한 1D 포지션 Embedding 벡터 (이미지 패치의 위치를 특정)를 만들고, 이를 각 이미지 패치 벡터와 합침 (두 Matrix 합)
  - 만들어진 Embedding 패치를 Transformer Encoder에 입력으로 넣어줌

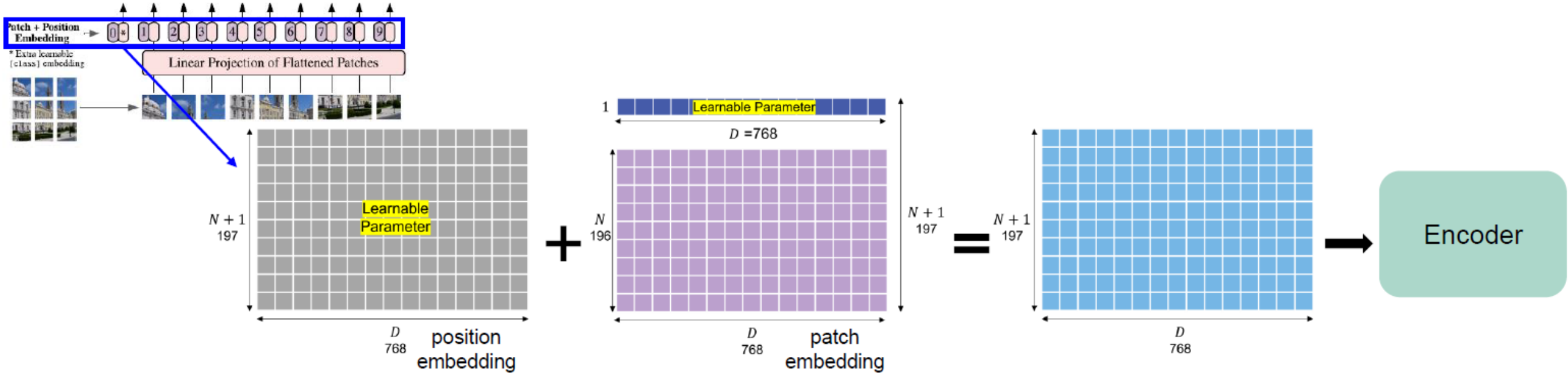


Fig 3. Embedding

Vision Transformer – Method

- Attention
  - Single Head Self-Attention (SSA)
    - Linear Layer (Q, K, V Matrix를 만들어주기 위한 차원 변경 수행)

$$[\mathbf{q}, \mathbf{k}, \mathbf{v}] = \mathbf{z}\mathbf{U}_{qkv}, \mathbf{U}_{qkv} \in \mathbb{R}^{D \times 3D_h}$$

    - D_h는 보통 D/k로 설정 (k는 Attention Head 개수)하며, 이는 파라미터 개수를 Head 개수에 무관하게 동일하도록 해주고 이후 Q, K, V로 분할

$$\mathbf{A} = softmax(\frac{\mathbf{q}\mathbf{k}^T}{\sqrt{D_h}}), \mathbf{A} \in \mathbb{R}^{N \times N} \quad SA(\mathbf{z}) = \mathbf{A}\mathbf{v}$$

    - Attention 가중치 A 계산을 시행한 뒤 가중치 A로 V의 가중합 계산
  - Multi Head Self-Attention
    - SSA를 Multi Head로 확장시키기 위해서는 차원 변경을 해줄 때 Head dimension 반영

$$MSA(\mathbf{z}) = [SA_1(\mathbf{z}); SA_1(\mathbf{z}); \ldots; SA_k(\mathbf{z})]\mathbf{U}_{msa}, \quad \mathbf{U}_{msa} \in \mathbb{R}^{k \cdot D_h \times D}$$

    - 이 때 $D_h = D/k$이므로 위 식의 $D_h \; X \; D$를 $D \; X \; D$로 쓸 수 있음
    - 처음의 U_{qkv}의 차원을 $D \; X \; 3D_h$가 아닌 Head 개수를 곱해준 $D \; X \; 3kD_h$로 변경
    - 이때 $D_h = D/k$이므로, U_{qkv}의 차원은 $D \; X \; 3D$로 정해짐

Vision Transformer – Method

- MLP
  - 2개의 Hidden Layer와 GELU (Gaussian Error Linear Unit) 활성화 함수로 구성
  - Hidden Layer의 차원의 경우 하이퍼 파라미터로 3072, 4096, 5120 옵션 존재
- Classification Head
  - 앞의 과정을 반복 후, 마지막으로 Classification을 수행하기 위해 Encoder 최종 아웃풋의 가장 첫 번째 벡터 y를 하나의 Hidden Layer (D X C)로 구성된 MLP Head 통과
  - Fine-Tuning 시에는 위의 MLP Head가 아닌, Single Linear Layer를 통과

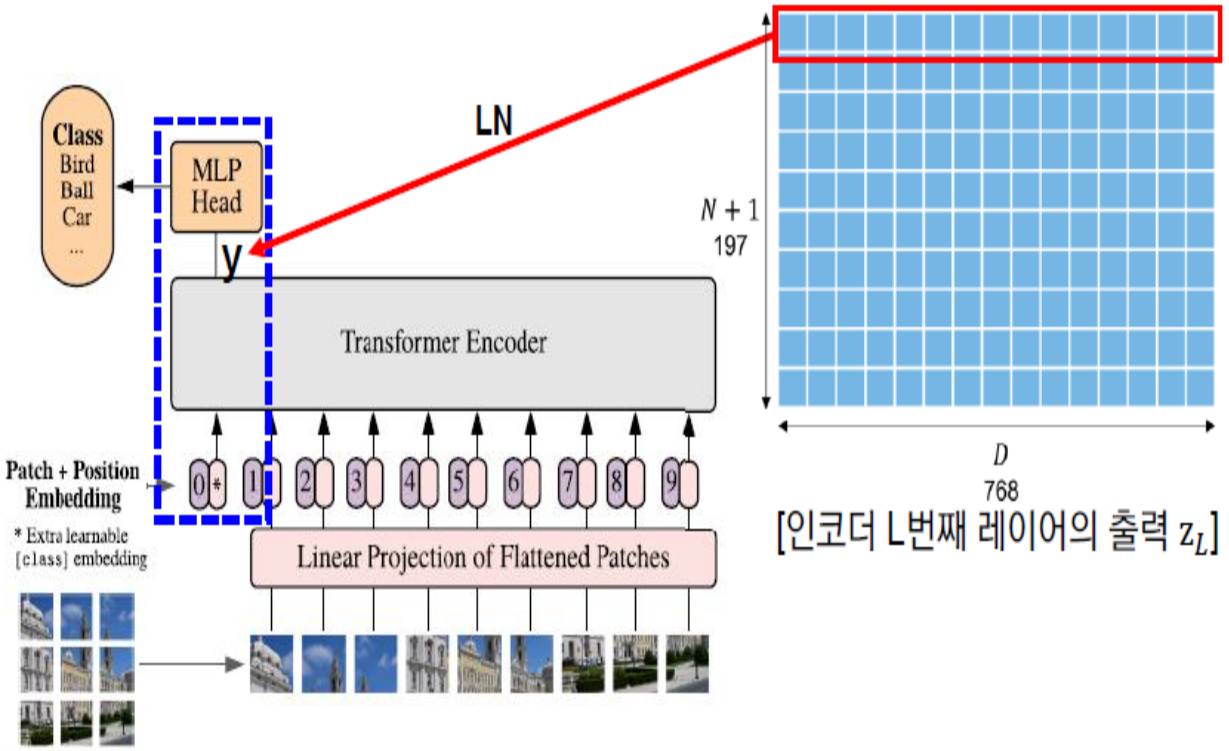


Fig 4. Classification Head

Vision Transformer – Method

- Inductive Bias
  - Locality, 2D Neighborhood structure (2차원적으로 이웃하는 구조), Translation Equivariance
- Inductive Bias Operates on ViT
  - MLP layers: Locality, Translation Equivariance
  - 2D Neighborhood structure: 입력 패치로 자르는 과정 (학습), Position Embedding 조정 (Fine-Tuning)

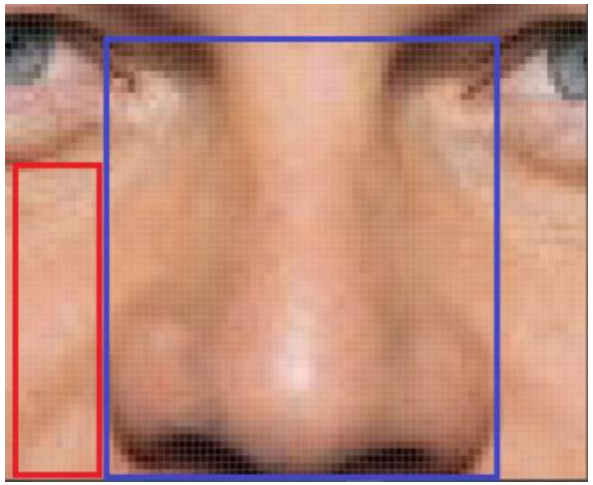


Fig 5. Locality

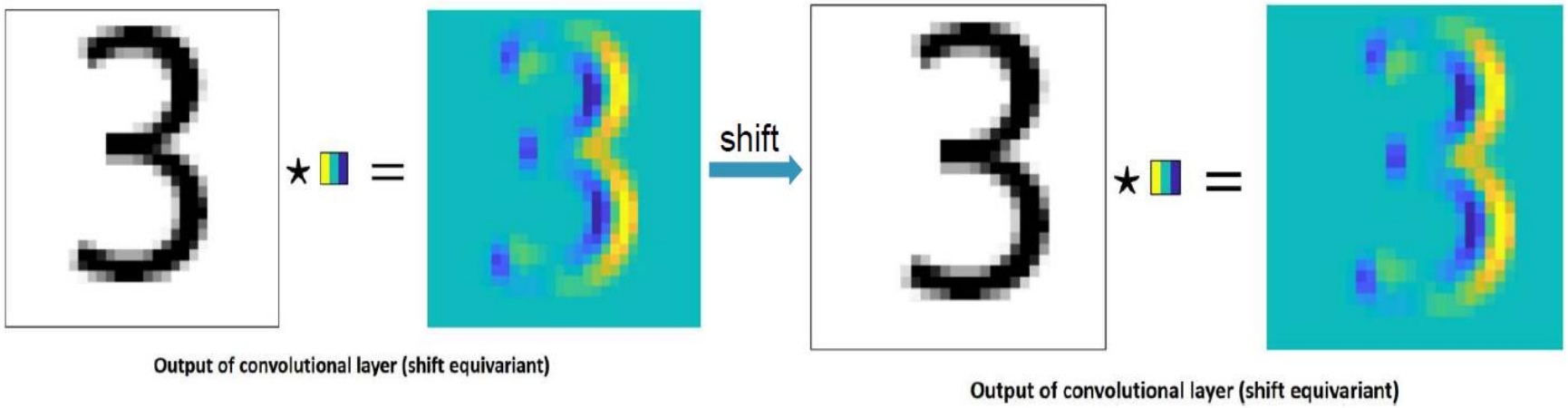


Fig 6. Translation Equivariance

Vision Transformer – Method

- Hybrid Architecture
  - Raw Image Patches가 아닌, CNN을 통과한 Feature Map을 입력 시퀀스로 넣음
- Fine-Tuning & Higher-Resolution
  - 사전 학습된 Prediction Head를 제거하고 0으로 초기화 된 Feed-Forward Layer 붙임
  - 사전 학습 시 보다 높은 해상도의 이미지로 Fine-Tuning하는 것이 더 좋은 결과를 보이지만, 이 때 기존의 Positional Embedding이 더 이상 의미 없음
  - 입력 이미지 크기 내 패치 위치에 맞게 Positional Embedding도 2차원 보간을 적용해 값을 채워 해상도 조정과 패치 추출이 ViT 내에서 수동적으로 Image-Specific Inductive Bias를 추가

Vision Transformer – Datasets

- Dataset
  - ImageNet-1k
  - ImageNet-21k
  - JFT
- Benchmark tasks
  - ReaL labels
  - CIFAR-10/100
  - Oxford-IIIT Pets
  - Oxford Flowers-102
  - VTAB Datasets 19-task

Vision Transformer – Model Variants

- 3개의 모델에 대해 실험, 각 모델에서도 다양한 패치 크기에 대해 실험
- Base, Large 모델은 BERT 모델에서 직접적으로 채택, Huge는 저자들이 추가, Hybrid는 ViT에 ResNet의 Stage 4 Feature Map을 입력으로 넣어주고 Patch Size = 1로 설정

Model	Layers	Hidden size D	MLP size	Heads	Params
ViT-Base	12	768	3072	12	86M
ViT-Large	24	1024	4096	16	307M
ViT-Huge	32	1280	5120	16	632M

Fig 7. Details of Vision Transformer model variants

Vision Transformer – Training Details

- Adam은 Pre-Train에 사용
- SGD는 fine-Tuning에 사용
- Batch Size: 4096

Models	Dataset	Epochs	Base LR	LR decay	Weight decay	Dropout
ViT-B/{16,32}	JFT-300M	7	$8 \cdot 10^{-4}$	linear	0.1	0.0
ViT-L/32	JFT-300M	7	$6 \cdot 10^{-4}$	linear	0.1	0.0
ViT-L/16	JFT-300M	7/14	$4 \cdot 10^{-4}$	linear	0.1	0.0
ViT-H/14	JFT-300M	14	$3 \cdot 10^{-4}$	linear	0.1	0.0
R50x{1,2}	JFT-300M	7	10^{-3}	linear	0.1	0.0
R101x1	JFT-300M	7	$8 \cdot 10^{-4}$	linear	0.1	0.0
R152x{1,2}	JFT-300M	7	$6 \cdot 10^{-4}$	linear	0.1	0.0
R50+ViT-B/{16,32}	JFT-300M	7	$8 \cdot 10^{-4}$	linear	0.1	0.0
R50+ViT-L/32	JFT-300M	7	$2 \cdot 10^{-4}$	linear	0.1	0.0
R50+ViT-L/16	JFT-300M	7/14	$4 \cdot 10^{-4}$	linear	0.1	0.0
ViT-B/{16,32}	ImageNet-21k	90	10^{-3}	linear	0.03	0.1
ViT-L/{16,32}	ImageNet-21k	30/90	10^{-3}	linear	0.03	0.1
ViT-∗	ImageNet	300	$3 \cdot 10^{-3}$	cosine	0.3	0.1

Fig 8. Training Details

Vision Transformer – Comparison to SOTA

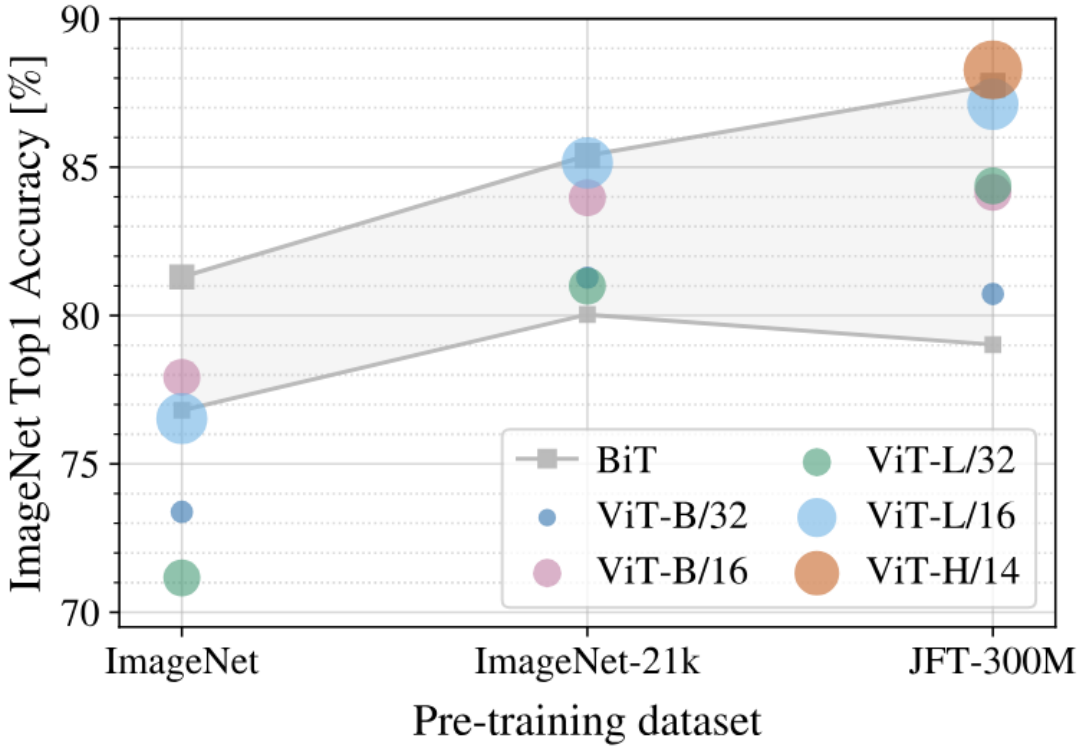
- Metrics
  - Few-shot ACC: Training Set에 없는 Class를 맞추는 문제에 대한 정확도
  - Fine-Tuning ACC: Fine-Tuning 후 정확도

	Ours-JFT (ViT-H/14)	Ours-JFT (ViT-L/16)	Ours-I21k (ViT-L/16)	BiT-L (ResNet152x4)	Noisy Student (EfficientNet-L2)
ImageNet	**88.55** ± 0.04	87.76 ± 0.03	85.30 ± 0.02	87.54 ± 0.02	88.4/88.5*
ImageNet ReaL	**90.72** ± 0.05	90.54 ± 0.03	88.62 ± 0.05	90.54	90.55
CIFAR-10	**99.50** ± 0.06	99.42 ± 0.03	99.15 ± 0.03	99.37 ± 0.06	–
CIFAR-100	**94.55** ± 0.04	93.90 ± 0.05	93.25 ± 0.05	93.51 ± 0.08	–
Oxford-IIIT Pets	**97.56** ± 0.03	97.32 ± 0.11	94.67 ± 0.15	96.62 ± 0.23	–
Oxford Flowers-102	99.68 ± 0.02	**99.74** ± 0.00	99.61 ± 0.02	99.63 ± 0.03	–
VTAB (19 tasks)	**77.63** ± 0.23	76.28 ± 0.46	72.72 ± 0.21	76.29 ± 1.70	–
TPUv3-core-days	2.5k	0.68k	0.23k	9.9k	12.3k

Fig 9. Comparison to SOTA

Vision Transformer – Pre-Training Data Requirements

- ViT는 Inductive Bias가 기존 CNN 계열보다 부족하기 때문에 사전 학습 데이터 셋 크기가 클 때 좋은 성능을 보임
- 사전 학습 시 사용되는 데이터 셋 크기가 ViT에 어떤 영향을 미치는지에 대한 실험 진행
- ImageNet-1k, ImageNet-21k, JFT에 사전 학습하고 ImageNet 데이터 셋에서 분류 성능 평가 진행



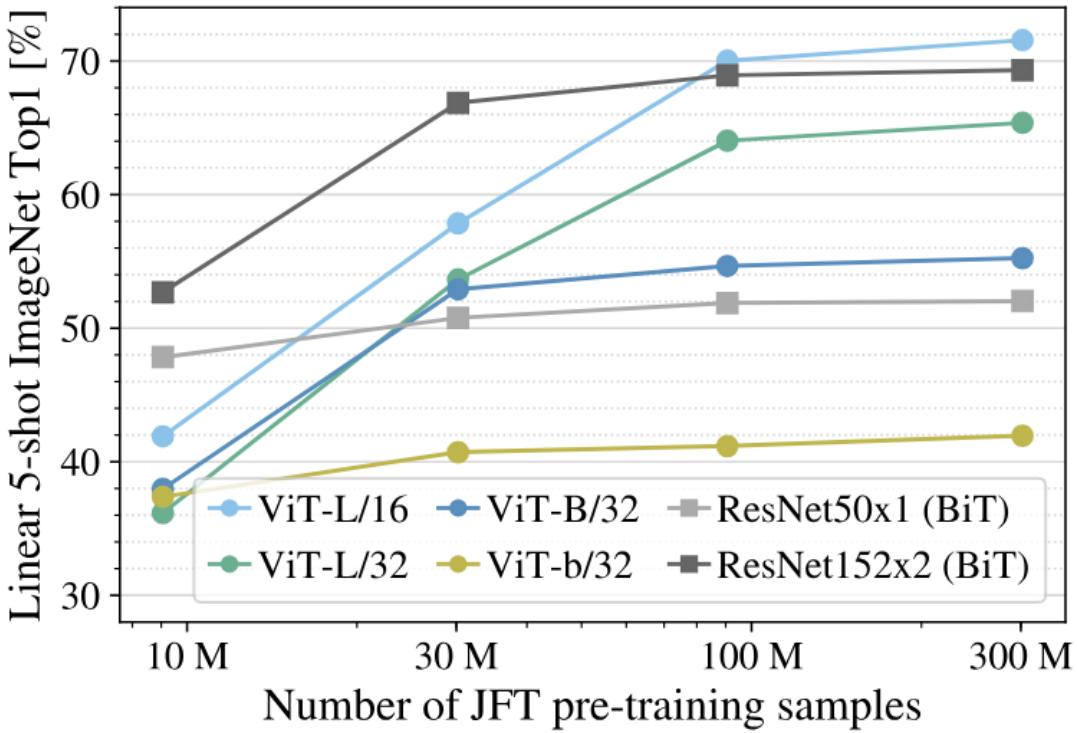


Fig 10. Training Details

Vision Transformer – Scaling Study

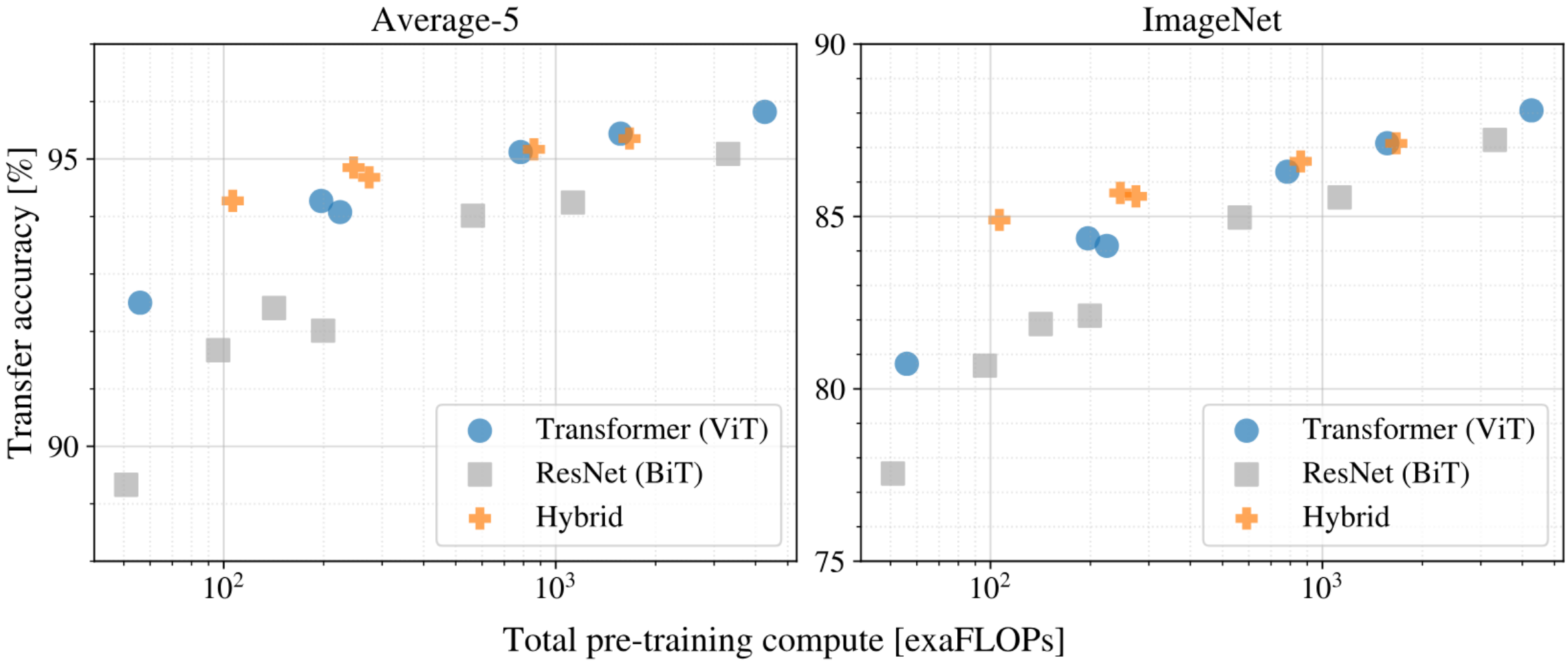


Fig 11. Scaling Study

Vision Transformer – Inspecting ViT

- Filter Visualization
  - 첫 번째 Linear Projection 부분에서 주요 요소 28개 선정 및 시각화
  - 잘 학습된 CNN 앞쪽 레이어를 시각화 했을 때와 유사한 결과가 나타남
  - 즉, 잘 학습된 CNN과 같이 이미지 인식에 필요한 Edge, Color 등 Low-Level 특징들을 잘 포착하고 있음을 파악
- Attention Distance
  - Attention을 사용한 만큼 Task 수행 시 이미지의 어느 부분에 집중하는지 파악할 수 있음

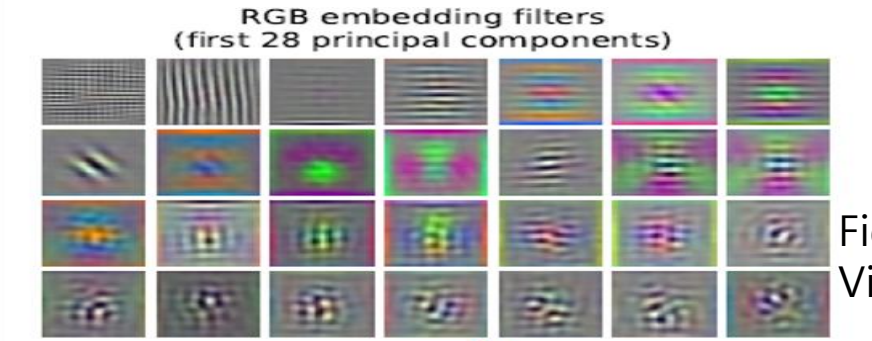
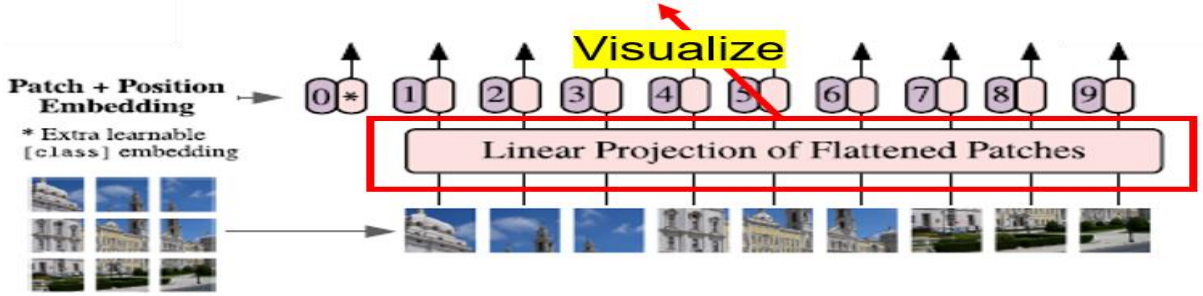


Fig 12. Filter Visualization



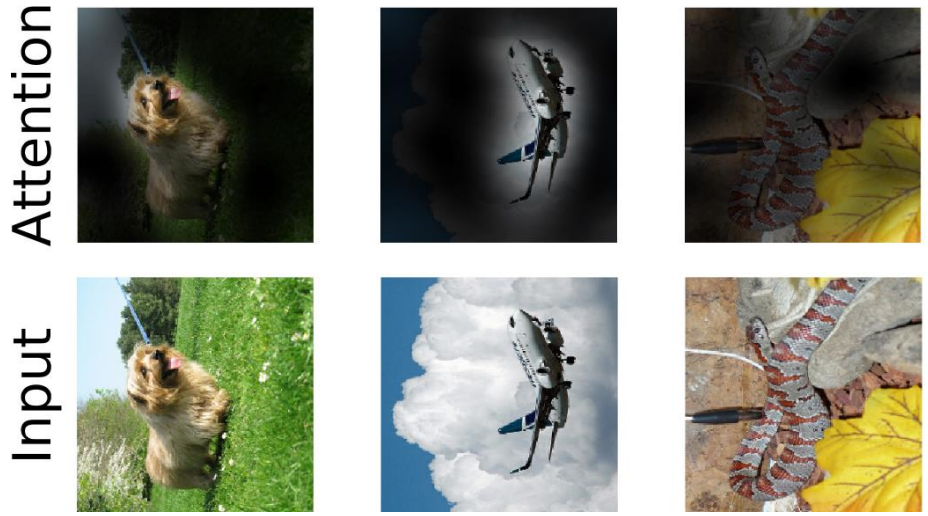


Fig 13. Attention Distance

Vision Transformer – Conclusion

- Contribution
  - Image Recognition 분야에 Transformer를 직접적으로 적용한 첫 사례
  - 간단하면서 높은 확장성을 가진 Transformer 구조를 성공정으로 Image Recognition 적용
  - 이미지에 특정된 Inductive Bias를 Architecture에 주입하지 않음
    (이미지 패치 추출 부분 제외)
  - 큰 사이즈의 데이터 셋에서 사진 학습한 후에 기존 CNN 기반 Baseline보다 더 좋은 성능
  - 사전학습이 크게 어렵지 않음
- Challenges
  - 본 연구에서 Classification Task에만 적용했기에 Segmentation, Detection 등 다른 CV에서도 적용이 성공적으로 되는지 아직 정확히 알 수 없음
  - 사전 학습 시 Self-Supervised Learning 방법론을 성공적으로 적용시킬 방안을 탐색해야 함
  - Self-Supervised Learning 방식으로 사전 학습 시 나쁘지 않은 결과를 얻었지만 Large-Scale Supervised Pre-Training과 큰 차이가 있음
  - 모델 구조 개선을 통한 모델 성능 향상의 여지가 존재

THANKS FOR YOUR ATTENTION

이찬호
국방인공지능응용학과