An Image Is Worth 16X16 Words: Transformers for Image Recognition at Scale(ViT)

태그	CV-architecture
Review	☐

Abstract

Transformer → 자연어 처리 작업에서 주로 사용되었음

In Vision → Attention + 합성곱 네트워크 or 합성곱 네트워크의 일부 구성 요소를 대체

기존의 이러한 제한 사항에서 벗어나, 이미지 패치에 순수한 Transformer 적용

이를 Vision Transformer(ViT)라고 함

Introduction

NLP에서의 상황

Transformer를 활용 → 대용량의 corpus에서 pre-trained 한 다음, fine-tuning을 수행

이러한 작업이 주는 장점 → 계산 효율성 & 확장성 증가

CV에서의 상황

여전히 Convolution architecture가 주류를 이루고 있음

Transformer와 결합하려는 여러 시도가 존재하였지만, 효과적인 이득을 보진 못했음

이로 인해, ResNet과 같은 architecture가 여전히 최첨단 기술로써 활용되고 있음

Transformer와 CV를 결합하기 위한 노력

- 이미지를 패치로 분할
- 이러한 패치의 선형 임베딩 시퀀스를 Transformer의 입력으로 제공
- 이미지 패치 → NLP에서의 단어(Token)와 동일한 방식으로 처리

하지만, 이를 강력한 정규화 없이 훈련할 경우, 다른 모델보다 더 낮은 정확도를 보임

→ Transformer는 CNN의 특징인 translation equivariance와 locality를 일반화하기 어려움

> 💡 ChatGPT Explanation
>
> 1. Translation Equivariance: 이는 Convolutional Neural Networks (CNNs)에서 핵심적인 특징 중 하나로, **이미지의 위치가 변하더라도 동일한 객체는 동일한 방식으로 인식되어야 한다는 원칙**을 의미합니다. 예를 들어, 어떤 이미지가 오른쪽으로 이동하더라도 CNN이 동일한 객체를 인식하는 방식에는 변함이 없어야 합니다.
> 2. Locality: 이는 CNN의 또 다른 중요한 특성으로, **CNN이 입력 이미지의 작은, 국소적인 부분에만 집중하도록 하는 특성**입니다. CNN의 각 Convolutional Layer는 입력의 작은 영역, 일명 'receptive field'에만 응답하며, 이를 통해 이미지의 국소적 패턴을 학습할 수 있습니다. 이러한 접근법은 이미지가 수많은 픽셀로 구성되어 있더라도, CNN이 효과적으로 특징을 추출할 수 있게 합니다.

이를 개선하기 위한 방법 → 데이터셋의 크기를 매우 키운다

데이터셋 크기를 키우니, 이러한 단점을 모두 극복하였음

ViT → 대규모 데이터셋을 활용해 pre-train을 수행한 다음, fine-tuning 작업

Related Work

이미지에 self-attention을 단순 적용 시, 픽셀 수에 대해 제곱 비용 발생

이를 해결하기 위해 여러 가지 근사 방법이 시도되었음

ViT와 가장 유사한 기존의 연구

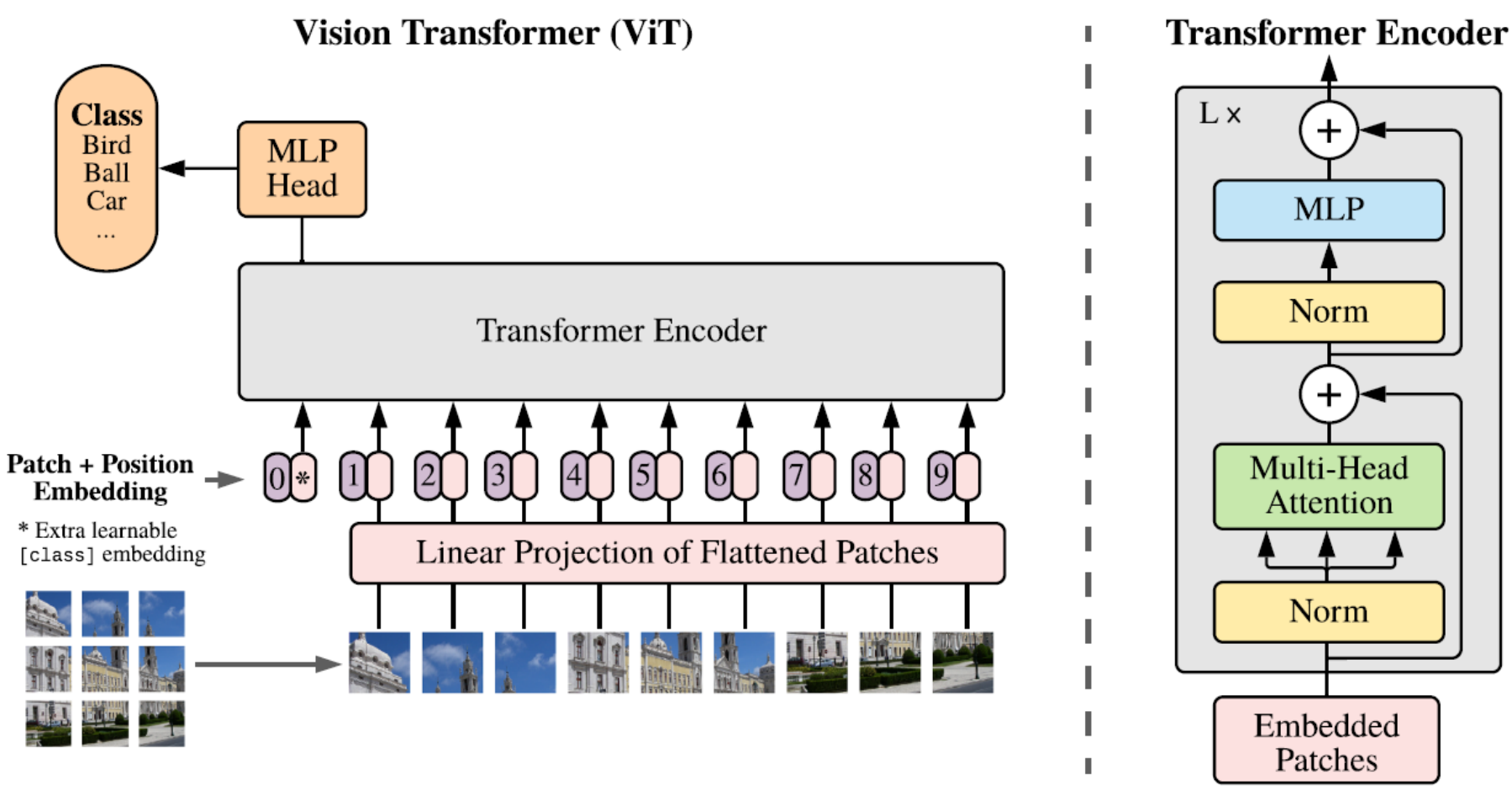
입력 이미지에서 2x2 크기의 패치를 추출 후, 위에 완전한 self-attention을 적용

해당 연구와 다른 점

- 대규모 데이터셋을 활용했다는 점
- 작은 해상도 이미지 뿐만 아니라, 중간 해상도 이미지도 처리할 수 있다는 점

Method

Model overview



모델 설계는 최대한 Transformer를 따르려고 했으며, 이는 NLP 아키텍쳐와 효율적인 구현을 거의 그대로 사용할 수 있다는 장점이 존재

Vision Transformer(ViT)

Transformer → 1차원 토큰 임베딩 시퀀스 입력

ViT → $x \in R^{H \times W \times C}$ 인 2차원 이미지를 $x_p \in R^{N \times (P^2 \cdot C)}$인 flatten된 패치 시퀀스 입력

(H, W) : 원본 이미지의 해상도, C : 채널의 수, (P, P) : 각 이미지 패치의 해상도

N = $\frac{HW}{P^2}$: 패치의 수 & Transformer의 입력 시퀀스 길이

Transformer → 모든 레이어에서 동일한 고정된 잠재 벡터 크기 D를 사용

ViT → 패치를 평탄화 후, 식 (1)을 통해 학습 가능한 선형으로 projection 하여 D 차원으로 매핑

이와 같은 projection의 출력을 패치 임베딩이라고 함

$$\mathbf{z}_0 = [\mathbf{x}_{\text{class}}; \mathbf{x}_p^1\mathbf{E}; \mathbf{x}_p^2\mathbf{E}; \cdots; \mathbf{x}_p^N\mathbf{E}] + \mathbf{E}_{pos}, \qquad \mathbf{E} \in \mathbb{R}^{(P^2 \cdot C) \times D}, \ \mathbf{E}_{pos} \in \mathbb{R}^{(N+1) \times D} \tag{1}$$

$$\mathbf{z'}_\ell = \text{MSA}(\text{LN}(\mathbf{z}_{\ell-1})) + \mathbf{z}_{\ell-1}, \qquad \ell = 1 \ldots L \tag{2}$$

$$\mathbf{z}_\ell = \text{MLP}(\text{LN}(\mathbf{z'}_\ell)) + \mathbf{z'}_\ell, \qquad \ell = 1 \ldots L \tag{3}$$

MSA : Multi-head Self-Attention, LN : Layer Normalization, MLP : Multi Layer Perceptron

$z_0^0 = x_{class}$: BERT 에서의 class 토큰과 유사하게, 임베딩된 패치 시퀀스 앞에 학습 가능한 임베딩을 추가

z_L^0 : Transformer 인코더의 출력, 이미지 표현인 y로 사용

$$\mathbf{y} = \text{LN}(\mathbf{z}_L^0) \tag{4}$$

pre-train 및 fine-tuning 중에는 z_L^0에 classification head를 추가

classification head → pre-train : 한 개의 은닉층을 가진 MLP, fine-tuning : 단일 선형 레이어

패치 임베딩 + 위치 임베딩 → 위치 정보를 유지

위치 임베딩으로는 1D를 활용 → 2D를 사용해도 성능 향상 x

Inductive bias(추론적 편향)

ViT는 CNN에 비해 이미지에 특화된 Inductive bias가 훨씬 적음

2차원 이웃 구조가 사용되는 곳

- 모델 시작 부분에서 이미지를 패치로 분할하는 용도
- fine-tuning 시 다른 해상도의 이미지에 대해 위치 임베딩을 조정하는 용도

이처럼 2차원 구조가 사용되는 곳이 적기 때문에, 편향이 매우 적음

→ 이를 데이터의 양을 늘려 해결하고자 함

Hybrid Architecture

Hybrid 모델에서는 패치 임베딩 projection이 CNN feature map에서 추출된 패치에 적용

Fine-Tuning and Higher Resolution

pre-train된 prediction head 제거, 0으로 초기화된 D x K feedforward layer를 연결(K 는 하위 작업의 클래스 수)

fine-tuning 시 사전 훈련 때보다 더 높은 해상도로 수행하는 것이 더 성능 좋음

이때 패치 크기를 유지하여 시퀀스 길이가 길어지지만 ViT는 가변 길이 처리 가능

이렇게 되면 pre-trained된 위치 임베딩은 의미가 없어지므로 이를 원래 이미지 위치에 따라 2D interpolation을 수행

Experiments

Setup

Dataset

ImageNet-21k : 21000개의 클래스와 14백만 개의 이미지

JFT : 18000개의 클래스와 3억 30만 개의 고해상도 이미지

사전 훈련 데이터셋을 하위 작업의 테스트 세트와 중복되지 않도록 처리

이러한 데이터셋에서 훈련된 모델을 여러 벤치마크 작업에 transfer

VTAB(Visual Task Adaptation Benchmark)를 통해 평가

Model Variants

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

ViT 구성은 BERT에 사용된 구성을 기반으로 제작

ViT-L/16 → 16×16 입력 패치 크기를 가진 Large 모델을 의미

패치 크기가 작은 모델일수록 계산적으로 더 비싸짐

기준 모델 : ResNet + Group Normalization + standardized convolutions

Training & Fine-tuning

pre-train → Adam($\beta_1 = 0.9,\ \beta_2 = 0.999$) + batch size 4096 + weight decay 0.1 로 학습

fine-tuning → SGD/모멘텀 사용, batch size 512

Metrics

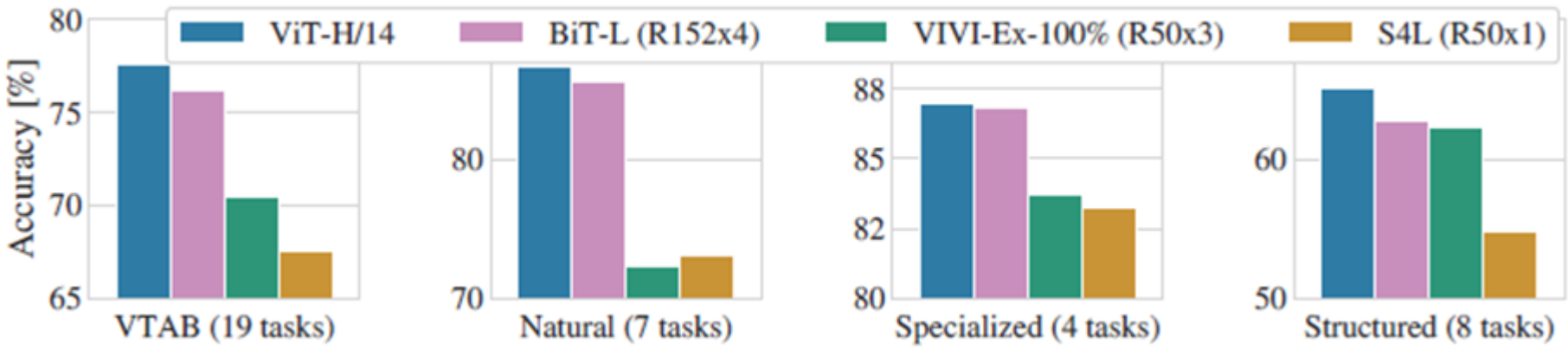
few-shot 또는 fine-tuning 정확도를 통해 평가

💡 few-shot : 제한된 예제를 통해 정확도를 평가하는 평가 지표

Comparison to State of the art

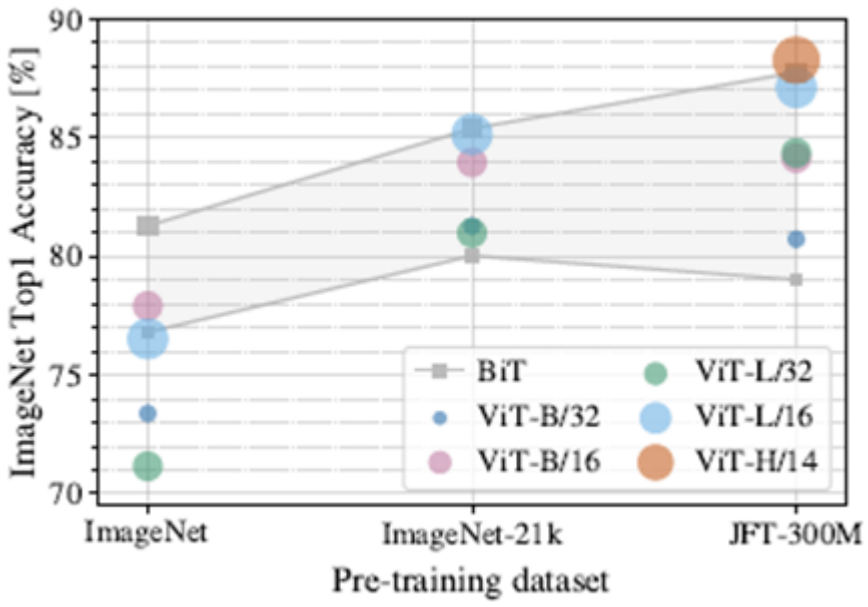
	Ours-JFT (ViT-H/14)	Ours-JFT (ViT-L/16)	Ours-I21k (ViT-L/16)	BiT-L (ResNet152x4)	Noisy Student (EfficientNet-L2)
ImageNet	**88.55** ± 0.04	87.76 ± 0.03	85.30 ± 0.02	87.54 ± 0.02	88.4/88.5*
ImageNet ReaL	**90.72** ± 0.05	90.54 ± 0.03	88.62 ± 0.05	90.54	90.55
CIFAR-10	**99.50** ± 0.06	99.42 ± 0.03	99.15 ± 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

fine-tuning 성능 비교

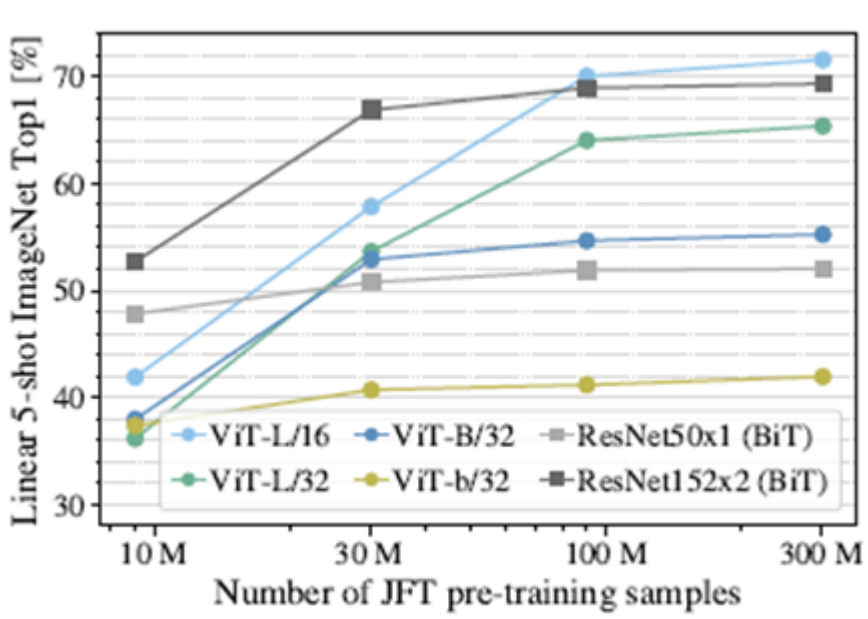


VTAB 성능 비교

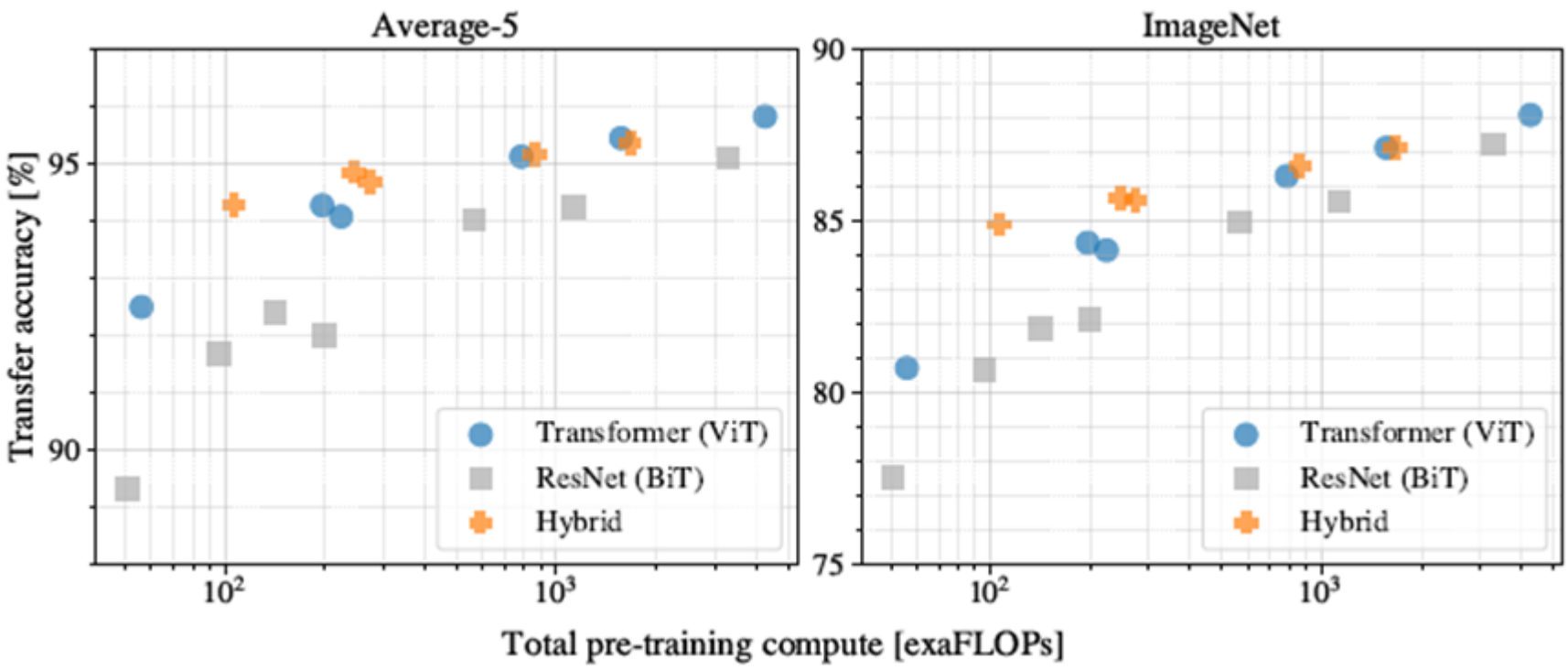
Pre-Training Data Requirements



데이터셋의 크기가 작으면 성능이 떨어지지만, 데이터셋의 크기가 커지면 성능을 추월

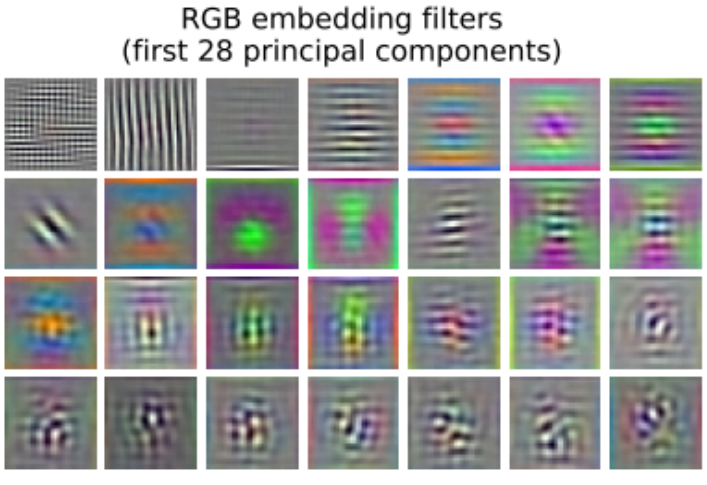


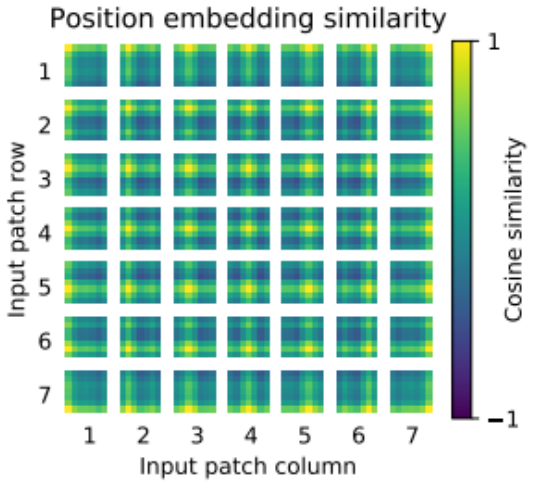
역시 데이터셋이 많을때 더 좋은 성능을 보인다는 것을 확인

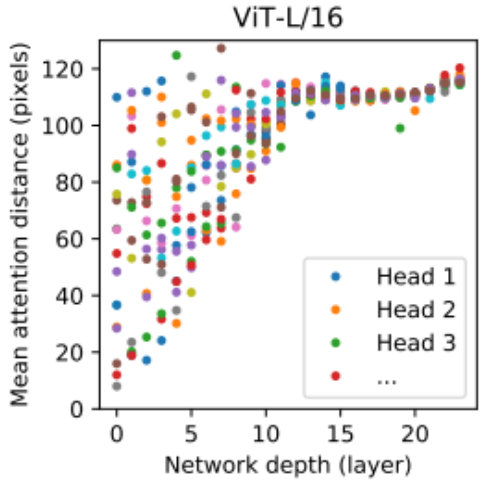


Hybrid 모델은 데이터셋이 작을때는 효과적이지만 커질수록 효과가 미미해짐

Inspecting Vision Transformer







ViT의 첫 번째 계층은 이미지 패치를 평탄화 후, 선형으로 저차원 공간으로 투영

그 후, 위치 임베딩이 패치 표현에 추가됨

유사한 위치에 있는 패치들은 비슷한 임베딩을 가진다는 것을 그림을 통해 확인

Self-attention을 통해 ViT는 이미지 전체에 걸친 정보를 가장 낮은 계층에서도 통합 가능

오른쪽 그림을 통해 낮은 계층에서 넓은 범위의 정보를 통합한다는 것을 확인

Conclusion

이미지 인식에 트랜스포머를 직접 적용하는 것을 탐구

이미지를 패치의 연속으로 해석, NLP에서 사용되는 표준 트랜스포머 인코더로 처리