**AN IMAGE IS WORTH 16x16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE 논문리뷰**

국민대학교 BIT대학원 DELab. 석사과정 이승건

1. **Introduction**

Transformer 아키텍처가 제안된 이후, NLP에서는 표준이 되었다. CV(Computer Vision) 분야에서는 CNN이 여전히 지배적이었다. Self-Attention을 적용하려는 시도는 있었으나, 현대 Hardware 가속기에서 효과적으로 확장되지는 못했다. 논문 저자들은 Transformer를 NLP에 scaling 성공에 영감을 받아 이미지 패치를 직접 입력으로 받아들이는 ViT(Vision Transformer)를 제안한다. 논문 제목에서 볼 수 있듯이, Transformer를 이용하므로 이미지를 Patch로 분할 후 Sequence로 입력한다.

ImageNet과 같은 중간 규모의 데이터셋에서 강한 정규화 없이 훈련할 때, ViT는 ResNet보다 낮은 정확도를 보인다. Transformer는 Inductive Bias(귀납적 편향)을 결여하고 있어 불충분한 양의 데이터로 훈련할 때 잘 일반화되지 않는다.

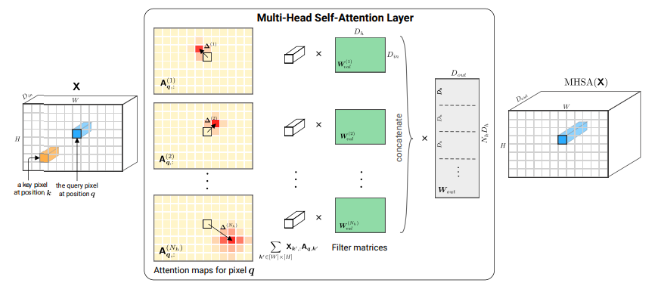
하지만 큰 데이터셋(1400만~3억개의 이미지)으로 훈련할 경우 대규모 훈련이 Inductive Bias보다 우월하다는 점을 발견했다.

ViT는 여러 Benchmark에서 최첨단 CNN보다 우수한 성과를 내며(SOTA 달성), 적은 컴퓨팅 자원으로도 훈련이 가능하다.

1. **Related Work**

* Transformer: Self-Supervised pre-training task(BERT)
* Naive(단순한) 방식: 이미지 처리할 때 Self-Attention을 적용하면 각 pixel이 모든 pixel에 주목해야 하고, 이는 비효율적이기에 그에 따른 여러 시도가 있었다.

1. Parmer: 각 query pixel에 대해 지역 이웃에서만 Self-Attention을 적용했다.
2. Weissenborn: 다양한 크기의 block을 Scale Attention한다. 세분화된 Attention 구조가 CV task에서 좋은 결과로 이어지지만, Hardware 가속기에서 효율적인 사용을 위해 복잡한 엔지니어링이 필요하다.
3. Cordonnier: ViT와 가장 관련 있는 모델로, 2x2pixel의 patch를 추출하여 full Self-Attention을 적용한다. 작은 patch 크기를 사용하여 낮은 해상도의 이미지만 처리할 수 있다. **ViT의 경우 중간 해상도의 이미지도 처리할 수 있다**.

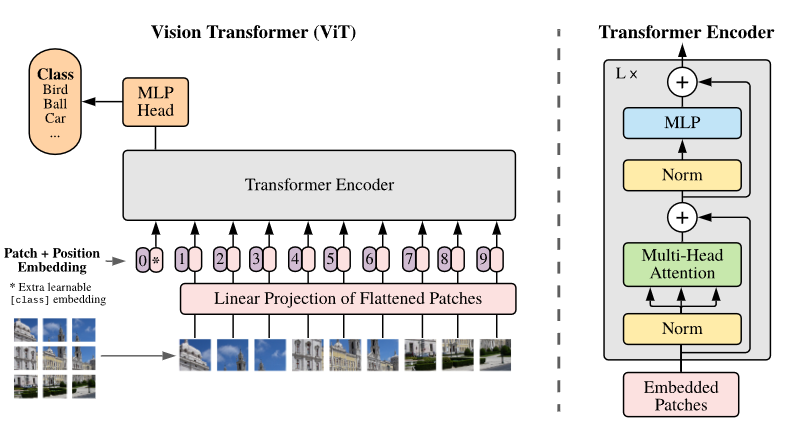


1. CNN과 Self-Attention의 결합: CNN의 feature map을 사용해 Self-Attention을 적용하고, 여러 Vision task에서 성과를 보이고 있다.
2. iGPT(이미지 GPT): 이미지 해상도와 Color 공간을 줄인 후 이미지 pixel에 Transformer를 적용한다. 생성 모델로서 Unsupervised 방식으로 training된다.

ViT의 목적은 이전 연구에서 사용된 ResNet기반 Model 대신 Transformer를 통해 ImageNet보다 더 큰 대규모 데이터셋에서 훈련한 모델이 Benchmark에서 더 우수함을 증명하는 것이라 생각한다.

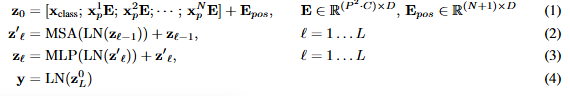
1. **Method**

Model 설계에서 ViT는 원래 Transformer를 가능한 한 가깝게 따른다. 이를 통해 NLP에서 사용하던 Transformer를 즉시 사용할 수 있다는 장점이 있다.



ViT model의 전체적인 구조이다. Encoder, Multi-Head Attention은 기본적으로 Transformer 형식을 따른다.

**Step1. Embedding for Transformer**

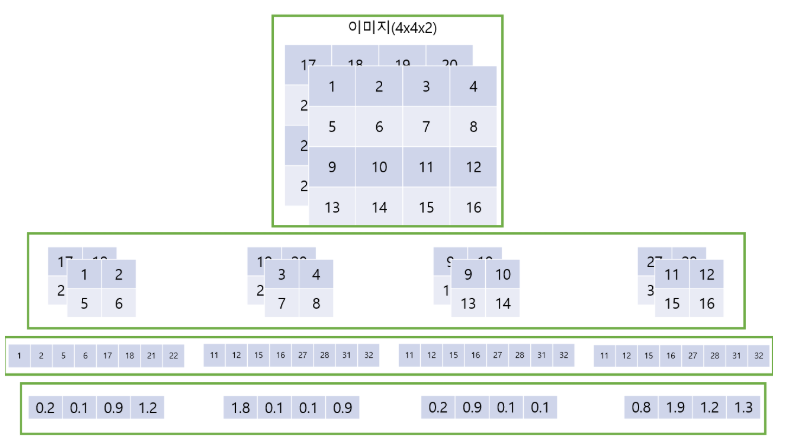
****

식 1은 embedding, 식2, 3은 Encoder 계층에서 MSA(Multi-Head Self-Attention), MLP(다층 퍼셉트론)를 나타내며 교차 Layer로 구성된다. LN(Layer Norm)은 각 블록 이전에 적용되며, Residual Connections(잔차 연결)은 각 블록 이후에 적용된다. 식 4의 y는 최종 출력으로, transformer의 최종 layer L에서의 출력이다. Layer 정규화를 통해 최종적인 이미지 표현을 계산한다.

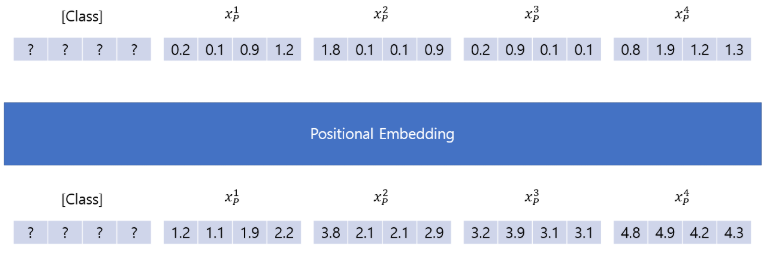
기존의 Transformer는 1차원의 Token Embedding들의 sequence를 입력으로 받는다. 이미지는 2D이기 때문에 1D로 변환하기 위한 작업(Patch Embedding)을 한다.

이미지 를 flattening하여 로 재구성한다. (H, W)는 원래 이미지 해상도, C는 채널의 수, (P, P)는 patch 해상도, 는 생성된 patch 수를 나타내는 동시에 Transformer의 유효 입력 sequence 길이 역할도 한다. 이후 D차원의 vector로 mapping하는 과정을 **patch embedding**이라고 한다. 각 출력을 (n=1, 2, ..., N)이라고 정의한다.

추가적으로 BERT의 [Class] token과 비슷하게, 입력 Embedding 맨 앞에 [Class] patch를 넣어준다. [Class] patch는 출력()중 맨 앞()에 대응되며 MLP Head에 입력으로 들어가 Classification 작업에 사용된다.



위 예시는 4x4x2 이미지를 2x2 patch로 나눈 뒤 4크기의 vector로 patch embedding한 것이다.



이후 Position Embedding을 더하여 각 Patch Embedding의 위치 정보를 추가해준다.

**Step2. Inductive bias(유도 편향)**

ViT는 CNN보다 이미지에 특화된 Inductive bias가 훨씬 적다. Self-Attention은 전역적이고, Position Embedding은 Patch의 2D위치에 대한 정보가 담겨있지 않으며, Patch간 모든 공간적 관계는 처음부터 배워야 한다. Inductive bias는 Appendix.a에서 자세히 설명한다.

**Step3. Hybrid Architecture**

이미지 Patch 대신, 입력 sequence가 CNN의 Feature map에서 형성될 수 있다. 이 경우 CNN에서 추출된 Feature map에 Patch embedding vector E를 적용시킨다.

Patch는 1x1크기의 D(차원)을 가질 수 있으며, Feature map을 flattening하여 이후 Tranformer의 입력에 맞게끔 D차원 크기의 vector로 projection하는 것이 가능하다는 것을 의미한다.

**Step4. Fine-tuning and Higher resolution**

ViT는 대규모 데이터셋에서 pre-training 후 downstream task에 대해 Fine-tuning 작업이 이루어진다. Fine-tuning시 pre-trained된 예측 head를 제거하고, downstream class K에 대한 zero-initialized된 D x K Feed Forward Layer를 연결한다.

Pre-trained때보다 높은 해상도의 이미지를 Fine-tuning에 적용할수록 성능 향상에 도움이 되는데, 이때 patch 크기는 동일하게 유지한다.

높은 해상도의 이미지를 입력할 경우 sequence길이가 증가하므로 pre-trained PE가 의미가 없을 수 있다. 따라서 2D 보간(interpolation)을 수행하여 PE를 조정한다. 이 과정이 ViT의 2D 구조에 대한 Inductive bias를 수동으로 주입하는 유일한 과정이다.

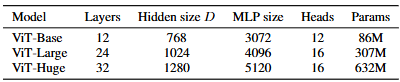
1. **Experiments**

ResNet, ViT, Hybrid model의 학습 능력을 평가한다. 본 논문은 SOTA CNN model과 비교하여, ViT가 기존의 model보다 대규모 데이터셋을 이용하여 학습 시 성능이 뛰어남을 보이고자 한다.

**Datasets:** ImageNet-1k(1천개 class, 130만개 이미지), ImageNet-21k(2만천개 class, 1400만개 이미지), JFT-18k(1만8천개 class, 3억3백만개 고해상도 이미지)

-> Fine-tuning (ImageNet-1k)

**Model Variants:**



* ViT: BERT에서 제안된 model과 동일하게 Base, Large model을 구성하고 추가로 Huge model을 구성한다. ViT-L/16은 Large model에 patch 16 x 16을 의미한다.
* ResNet: Batch Normalization을 Group Normalization & Weight Standardization으로 대체한다.
* Hybrid: Feature map을 1x1 patch로 ViT에 입력한다. 다양한 sequence길이를 실험하기 위해 i)ResNet50 4단계 출력 사용. ii)ResNet50 4단계 제거 후, Layer 수를 유지하며 확장된 3단계 출력을 사용한다.

**Training & Fine-tuning**

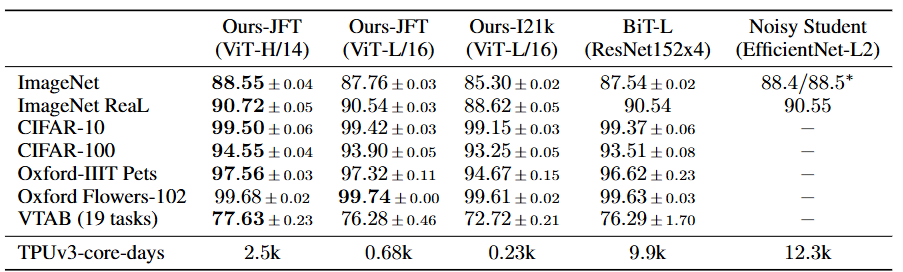
* Training: Optimizer-Adam, Batch-size-4096, Weight Decay-
* Fine-tuning: Optimizer-SGD, Batch-size-512, 해상도 변경- ViT-L/16=512, ViT-L/14 = 518, Weight Decay = 0.9999

**Metrics**

Few-shot 또는 Fine-tuning accuracy를 통해 정확도를 측정한다. 주로 Fine-tuning에 초점을 두지만, 비용이 너무 높아질 경우 linear few-shot을 이용한다.

**Compared to SOTA**

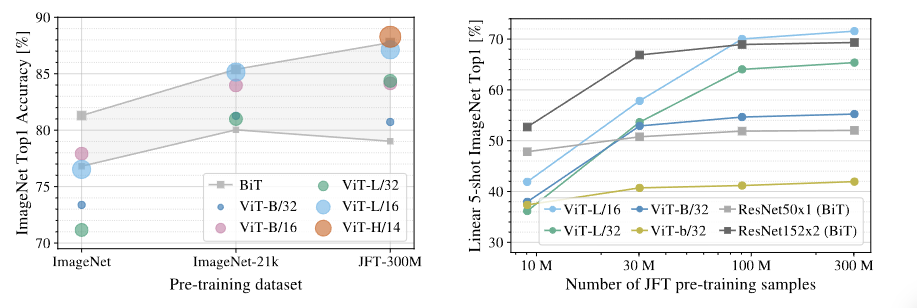
ViT-H/14와 ViT-L/16, CNN SOTA model인 BiT(Big Transfer, ResNet)와 Noisy Student(EfficientNet)를 비교했다. 모든 모델은 TPUv3 Hardware에서 훈련되었다.



ViT가 성능, 자원 소모 부분에서 SOTA를 달성했음을 알 수 있다.

**Pre-training data requirements**

ViT는 JFT-18k dataset으로 pre-training했을 때 성능이 가장 좋았다. Dataset이 많을수록 성능이 좋다는 것인데, 그럼 얼마나 많은 dataset이 필요할지에 대한 실험이다.



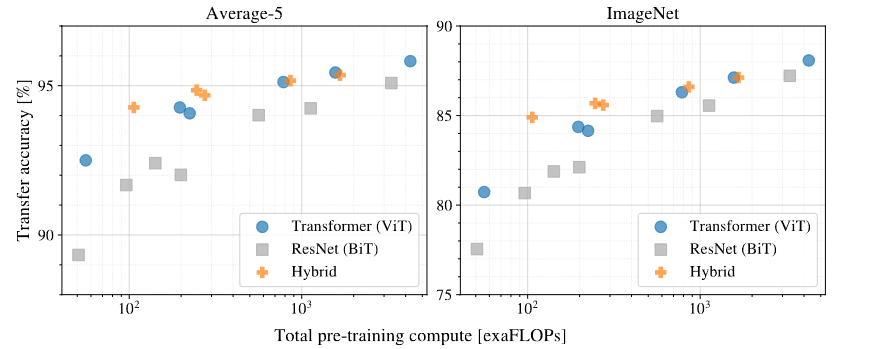
왼쪽 그래프는 데이터셋 크기(x축)에 따라 pre-training 후 같은 데이터셋(ImageNet)으로 fine-tuning한 결과이다.

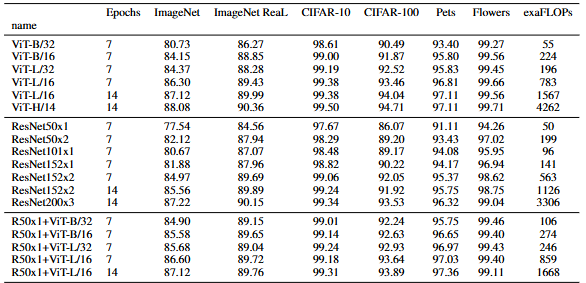
오른쪽 그래프는 데이터셋 크기에 따른 pre-training 후 전이 학습 성능 비교이다.

데이터의 크기가 커지면 커질수록 ViT의 성능 향상이 급격하게 증가됨을 알 수 있다.

**Scaling study**

model별로 pre-training을 하는데 필요한 cost에 대한 비교 실험을 진행한다.

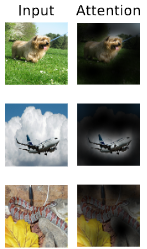




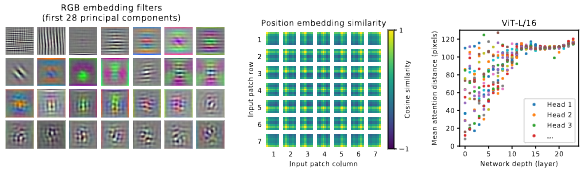
x축이 pre-training시 필요한 cost(FLOPs, FLoating Operations Per Sec)이다. 두 그래프에서 비슷한 양상을 보이며 같은 성능을 내기 위해 ViT가 BiT에 비해 약 2~4배정도의 적은 FLOPs를소비한다. Hybrid의 경우 적은 cost 계산에서는 우위에 있지만, 이후 차이는 점점 줄어든다. 결론은 작은 크기의 model일수록 ViT보다 Hybrid가 우세하고, 기본적으로 ViT가 BiT보다 우수하다. ViT의 성능이 포화되지 않은 것을 고려한다면, 성능이 더 좋아질 여지가 충분히 있다고 볼 수 있다.

**Inspecting Vision Transformer**

ViT가 이미지 데이터를 어떻게 처리하는지에 대한 분석이다.



위 사진은 학습한 ViT를 기반으로 model이 어느 곳을 집중(Attention)하는지에 대해 시각화 한 부분으로, Class 정보와 관련된 이미지에 대해 적절한 영역에 집중하여 결과를 출력하는 것을 보여준다.



* 왼쪽 그림: embedding filter를 시각화한 것이다. 많은 데이터를 pre-training하면, embedding filter가 저차원의 CNN 기능과 비슷함을 보인다.
* 가운데 그림: 학습된 PE에 대한 코사인 유사도의 Heatmap을 보여준다. 각 patch별 실제 위치와 비슷하게 유사도가 높은 것을 확인할 수 있다.
* 오른쪽 그림: Self-Attention이 얼마나 전체적인 정보를 잘 통합하는가에 대해 확인하기 위해 이미지 공간에서 각 Head가 Attention 가중치에 의해 어떤 정보와 통합이 되었는지 평균 거리를 계산하여 시각화한 것이다.

Attention distance = Receptive Field(CNN), 즉 Attention의 거리가 CNN의 수용영역과 같은 역할을 한다. 낮은 Layer의 Head는 CNN처럼 Localization 효과를 보인다. Layer가 깊어질수록 Attention distance는 증가하고 이는 model이 classification에 의미론적으로 유의미한 지역에 Attend함을 의미한다.

**Self-Supervision**

Transformer model이 NLP에서 혁신을 불러왔지만, model만으로 성능 향상이 이루어진 것은 아니다. 특히 BERT에서는 Self-Supervised를 통해 성능이 더 향상되었다.

ViT model에서도 BERT와 동일하게 Patch의 일부를 masking하는 방식, 즉 Self-Supervision을 통해 성능을 향상시키려 시도했다. ViT-B/16은 ImageNet에서 scratch(처음부터) 훈련할 때보다 2% 개선을 보여주었으나, 여전히 Supervised pre-training에 비해 4% 낮다.

개인적인 생각으로는 언젠가는 NLP의 영역처럼 CV분야에서도 Self-Supervision을 통해 성능개선이 될 날이 올 것이라 생각한다.

1. **Conclusion**

간단하면서 높은 확장성을 가진 Transformer구조를 성공적으로 CV분야에 적용하였다. 이미지에 특정된 inductive bias를 architecture에 주입하지 않는다(Patch 추출 제외). 대규모 데이터셋에서 pre-training이 이루어지면, 기존의 CNN SOTA model보다 좋은 성능을 기록한 점, 그리고 상대적으로 적은 cost로 pre-training을 진행 수 있는 점이 CV분야에 큰 기여를 했다.

앞으로의 도전 과제는,

* 본 논문에서는 classification(분류) task에만 적용하였기에 segmentation, detection 등 다른 CV task에 대한 검증
* Pre-training시 self-supervised learning을 성공적으로 적용시킬 방안
* Model 구조개선을 통한 성능 향상의 여지
* 데이터셋이 커질수록 성능이 향상되는데, 포화점이 어디일지

정도가 있다.

1. **Appendix**

**a. Inductive Bias(유도 편향,직역하면 귀납적 편향) – Relational, non-relational로 나뉜다.**

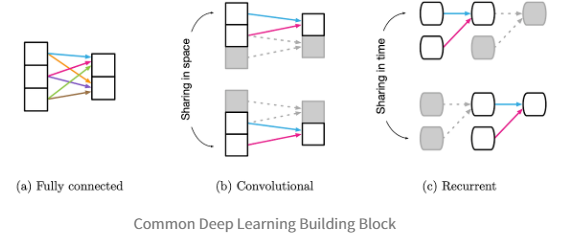
* Bias: 타겟과 예측값이 얼마나 멀리 떨어져 있는가, Bias가 높다면 데이터로부터 타겟과의 연관성을 잘 찾아내지 못하는 underfitting 발생
* Variance: 예측값이 얼마나 퍼져 있는가, Variance가 높다면 데이터의 사소한 noise나 random한 부분까지 고려하는 Overfitting 발생

따라서 Bias를 학습 알고리즘의 잘못된 가정에 의해 발생하는 오차라고도 하는데, 잘못된 가정에 의해 데이터에서 중요한 부분을 놓치고 있다는 의미로 볼 수도 있다.

Machine, Deep Learning은 특정 상황(task)에 맞춰 개발되는 경우가 대부분이다.

**주어지지 않은 입력의 출력을 예측하는 것, 즉 일반화의 성능을 높이기 위해 만약의 상황에 대비한 추가적인 가정이 Inductive Bias이다.**

Relational이란, 입력 요소와 출력 요소의 관계에 초점을 맞춘 것이다.



* FCNN(Fully Connected): 가중치와 편향으로 각 층의 요소들이 서로 모두 연결되어 있으므로, 모든 입력의 요소가 어떤 출력의 요소던지 영향을 미칠 수 있기 때문에 Inductive Bias가 매우 약하다.
* CNN: Filter가 입력을 Sliding Window하게 된다. 요소간의 관계가 약하므로, Locality & Translation Invariance의 Relational Inductive Biases를 갖는다.
* RNN: CNN이 공간적 개념이라면, RNN은 시간적 개념이다. Sequential & Temporal Invariance의 Relational Inductive Biases를 갖는다.

CNN은 이미지가 지역적으로 얻을 정보가 많다는 것을 가정하고 만들어진 모델이다. 반면, Transformer는 PE(Positional Embedding)과 Self-Attention을 통해 모든 정보를 활용한다. 즉, Transformer는 CNN보다 Inductive Biases가 부족하다고 볼 수 있다.

**b. Few-shot – 적은 데이터셋으로 모델을 학습하는 Meta Learning의 한 종류**

Few-shot은 Few한 데이터도 잘 분류를 할 수 있다. Few한 데이터로 학습을 한다는 의미는 아니다. **구분하는 방법을 배우고자 하는 방법론**이고, 이를 위해 많은 데이터가 필요한 것은 마찬가지이다. 기존의 DL과 다른 점은, 구분하고자 하는 대상이 반드시 학습 데이터셋에 없어도 된다는 점이다.

**Few-shot Learning process** - Training Set, Support Set, Query Image가 필요하다.

Training Set을 이용하여 구분하는 법을 배우고, Query Image가 들어왔을 때, Query Image가 Support Set중 어떤 것과 같은 종류인지 맞춘다.

결과적으로 “어떤 클래스에 속하는가”를 맞추는 것이 아닌, “특정 클래스와 같은 클래스인가”를 푸는 문제라고 생각하면 된다.

k-way는 Support Set이 k개의 클래스로 이루어진 것이고, k의 값이 클수록 model의 정확도는 낮아진다. N-shot은 각 클래스가 가진 sample 수이고, n이 클수록 model의 정확도는 높아진다.