An Image is Worth 16x16 words 논문 리뷰

Problem to solve

1. 기존의 트랜스포머 구조는 NLP에서는 거의 표준으로 활용되었으나, CV task에서의 적용에는 한계가 있었다. CV에서의 attention은 convolution과 그 변형 구조로 적용되고 있다.

Key Architecture

1. Model is designed by original Transformer as closely as possible
    - To achieve advantages of scalable NLP Transformer architectures, and efficient implementations

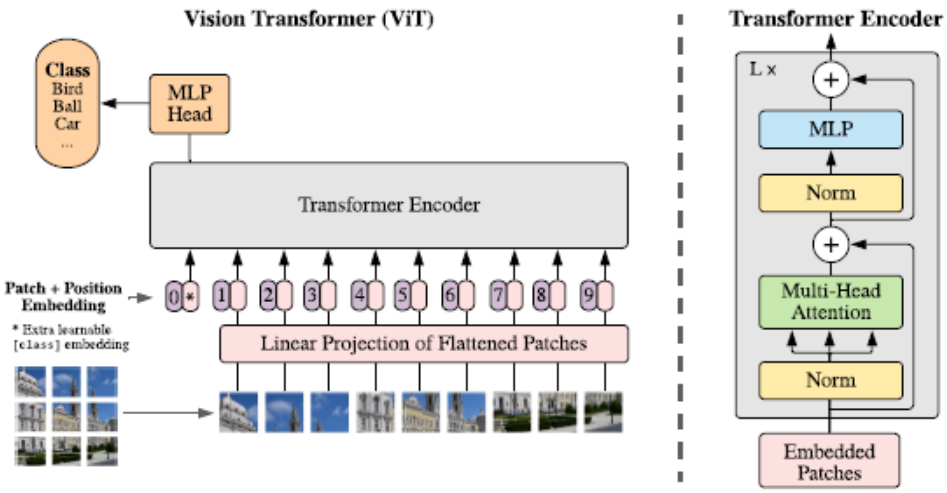


Figure 1: Model overview. We split an image into fixed-size patches, linearly embed each of them, add position embeddings, and feed the resulting sequence of vectors to a standard Transformer encoder. In order to perform classification, we use the standard approach of adding an extra learnable "classification token" to the sequence. The illustration of the Transformer encoder was inspired by Vaswani et al. (2017).

- 그냥 성능이 더 좋아서 pre-train Optimizer 는 Adam (Appendix D.1), fine-tunning 에는 SGD

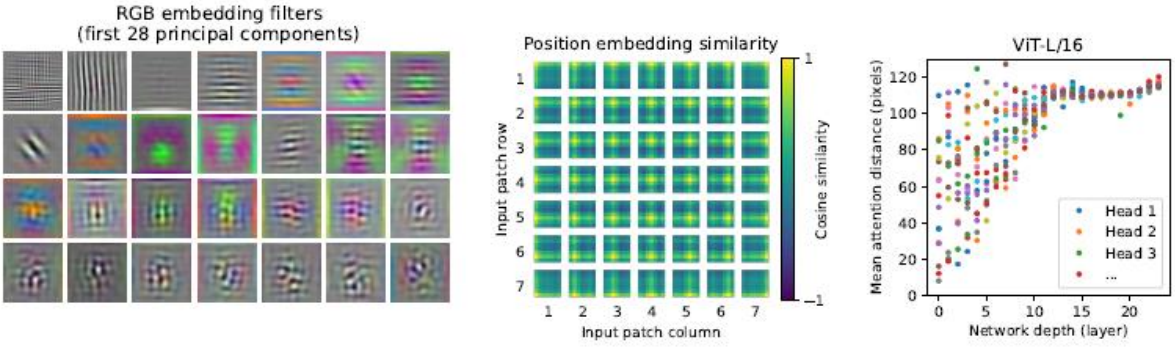


Figure 7: **Left:** Filters of the initial linear embedding of RGB values of ViT-L/32. **Center:** Similarity of position embeddings of ViT-L/32. Tiles show the cosine similarity between the position embedding of the patch with the indicated row and column and the position embeddings of all other patches. **Right:** Size of attended area by head and network depth. Each dot shows the mean attention distance across images for one of 16 heads at one layer. See Appendix D.7 for details.

2. Transformer Encoder architecture

   A. Multiheaded self-attention (MSA, appendix A)

$$[\mathbf{q}, \mathbf{k}, \mathbf{v}] = \mathbf{z}\mathbf{U}_{qkv} \qquad \mathbf{U}_{qkv} \in \mathbb{R}^{D \times 3D_h}, \tag{5}$$

$$A = \text{softmax}\left(\mathbf{q}\mathbf{k}^\top / \sqrt{D_h}\right) \qquad A \in \mathbb{R}^{N \times N}, \tag{6}$$

$$\text{SA}(\mathbf{z}) = A\mathbf{v}\,. \tag{7}$$

$$\text{MSA}(\mathbf{z}) = [\text{SA}_1(z); \text{SA}_2(z); \cdots; \text{SA}_k(z)]\,\mathbf{U}_{msa} \qquad \mathbf{U}_{msa} \in \mathbb{R}^{k \cdot D_h \times D} \tag{8}$$

   B. MLP blocks contains two layers with a GELU non-linearity

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

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

   C. LayerNorm(LN)

   D. Residual connection

3. Image Patches (224x224 Image -> 14x14 개의 16x16 image patch 로)

   - Image 를 작은 크기의 정사각형 patch 로 나누어 linearly embed 하고 positional embedding 을 더해 input vector 를 만들고, 순차적으로 transformer encoder 에 입력한다. (patch 가 NLP 에서의 word 에 해당)

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

   - Patch 를 Trainable linear projection 으로 linearly embed : Trainable linear projection)

- Hybrid Architecture 로 Input 을 patch 가 아닌 CNN 을 통과한 feature map 을 사용할 수도 있다.

- Smaller patch size -> long sequence -> computationally expensive

4. Positional Encoding (Appendix D.4)

   - patch 들간의 위치정보, distance 를 positional encoding 을 통해서 반영한다.

   - Learnable 1D positional embeddings (P^2xC->D, P: patch width, height)의 raster order(행 순서로 concatenate) vector 로 구현, Realative positional embeddings(patch 들간의 상대적 거리 전부 반영), 2D positional embedding (P^2xC->D/2+D/2)로도 구현해 봤으나 성능 차이 없음.

   - Transfomer encoder 에 feed 직전의 flatten patch 에 positional vector 를 더함. 다른 방식으로 더하는 경우도 있으나 성능 차의 거의 없음. 아마 pixel level 이 아닌, patch-level 이기에 spatial dimension 이 훨씬 작기 때문에 방법에 상관없이 positional 정보 학습이 쉬울 것.

Pos. Emb.	Default/Stem	Every Layer	Every Layer-Shared
No Pos. Emb.	0.61382	N/A	N/A
1-D Pos. Emb.	0.64206	0.63964	0.64292
2-D Pos. Emb.	0.64001	0.64046	0.64022
Rel. Pos. Emb.	0.64032	N/A	N/A

Table 8: Results of the ablation study on positional embeddings with ViT-B/16 model evaluated on ImageNet 5-shot linear.

   - 그러나 hyperparameter 에는 영향을 받음 Figure 10

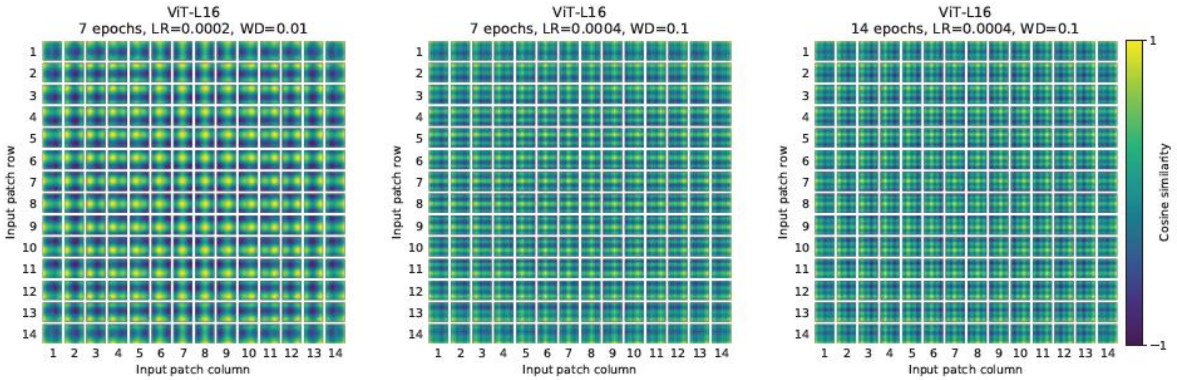


Figure 10: Position embeddings of models trained with different hyperparameters.

5. Transformer 앞 단에 Extra learnable "classification token", z_0^0: similar to BERT "class token" (Appendix D.3)

   - Transofmer 의 Output z_L^0 should be image representation y

$$y = \mathrm{LN}(z_L^0) \quad (4)$$

   - Classification head is implemented by single hidden layer MLP(with tanh as non-linearity) . token 이 이 네트워크를 통과하여 class prediction, z_L^0으로 predict

   - Image patch-embeddings 와 Global average-pooling 으로 classification 하는 방법도 learning rate 만 잘 맞추면 token 을 사용하지 않아도 충분한 성능 획득 가능함.

6. Fine Tunning

   - Pre-trained prediction head(MLP) 대신 DxK(D: input feature size, K: class 수) feedforward layer 를 붙인다.

   - 보통 pre-train 보다 높은 화질에서 이득을 취할 수 있다. Patch 사이즈는 그대로, sequence 길이가 더 길어지는 방식. Positional embedding 은 interpolation 한다.

7. 성능 측정방법

   - Fine-tunning accuracy 의 cost 가 너무 커지는 경우 on-the-fly evaluation 의 속도 향상과 closed form 을 위하여 few-shot accuracy 로 least-squares regression 으로 vector 값을 추정함.

Achievement

1. Image classification task에서 CNN의 사용 없이 pure transformer architecture만으로 기존의 SOTA CNN 기반 네트워크보다 pre-training cost는 더 작게, performance는 비슷하거나 더 뛰어남을 확인 (Figure 3, 4, 5 : 같은 performance일 경우, 2-4배 computing이 작음). (CNN, RNN이 특정 크기 이상의 큰 데이터에 대한 학습이 되면 사실 필요가 없고, Transformer가 훨씬 일반화를 잘하는 특이점에 도달하게 되었다. Fig 5에서 확인할 수 있듯이 convolution local feature의 정보는 데이터가 커지면 무의미해진다. 그러나 어떤 크기의 경우에도 Convolution 정보가 성능 향상에 도움이 되었다.)

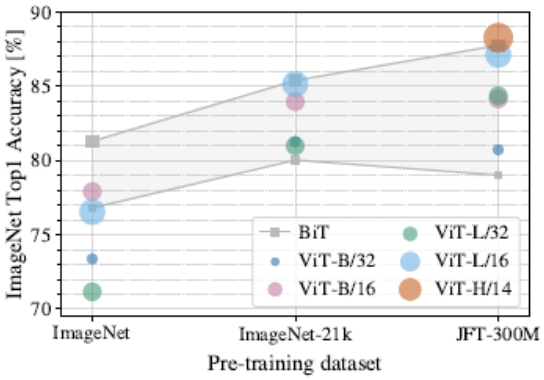


Figure 3: Transfer to ImageNet. While large ViT models perform worse than BiT ResNets (shaded area) when pre-trained on small datasets, they shine when pre-trained on larger datasets. Similarly, larger ViT variants overtake smaller ones as the dataset grows.

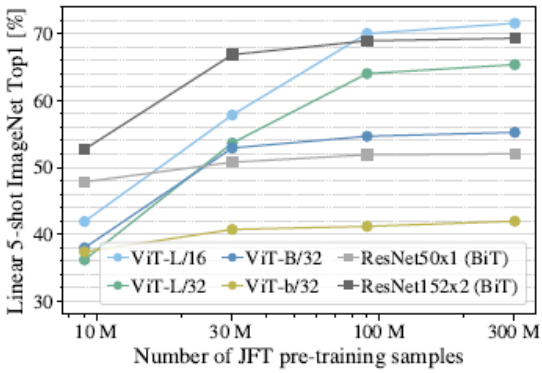


Figure 4: Linear few-shot evaluation on ImageNet versus pre-training size. ResNets perform better with smaller pre-training datasets but plateau sooner than ViT, which performs better with larger pre-training. ViT-b is ViT-B with all hidden dimensions halved.

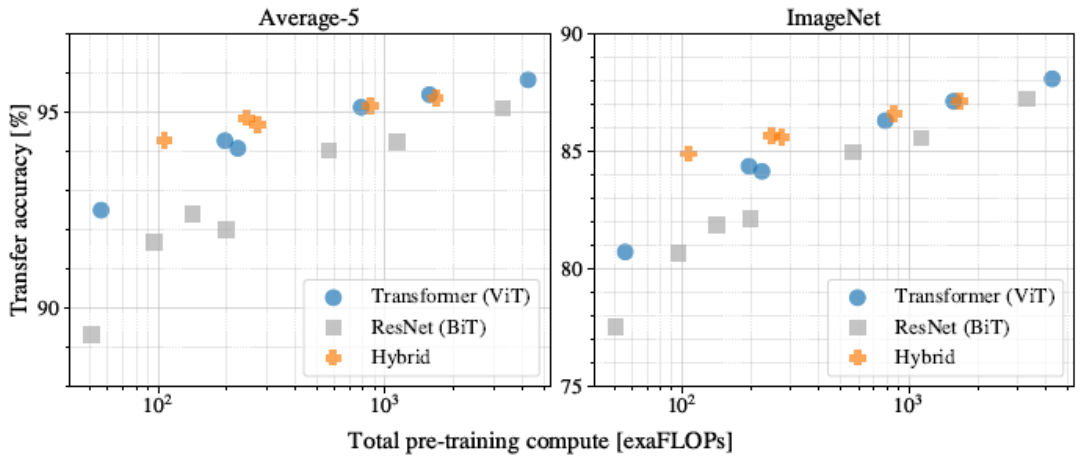


Figure 5: Performance versus pre-training compute for different architectures: Vision Transformers, ResNets, and hybrids. Vision Transformers generally outperform ResNets with the same computational budget. Hybrids improve upon pure Transformers for smaller model sizes, but the gap vanishes for larger models.

2. Inductive bias를 2d positional embedding을 사용할 때를 제외하고는 input으로 넣지 않고도 image classification에서 SOTA 성능을 보였다.

Limit

1. 큰 데이터셋에서의 supervised pre-train이 필요.
   - Mid-size dataset에 대해서는 같은 크기의 ResNet 기반 모델보다 성능이 조금 떨어졌다. (Transformer에는 CNN 기반 모델과 같은 귀납적인 편향, inductive biases가 부족하

다. Such as translation equivariance, locality(지역정보)) -> 충분히 큰 데이터셋 필요. (데이터셋이 커지면 inductive bias를 학습한다.)

- 따라서 충분히 큰 데이터셋에서의 pre-train이 필요하고, 작은 데이터셋으로 transfer 시켜야 한다.

2. 이미지 화질이 아직은 Low resolution이다.

3. 활용.

- Image detection, segmentation으로의 활용.

- 아직 Saturation이 되지 않았기에 더 큰 network로의 scailing 가능할 것.

- Self-supervised pre-training 방법으로의 도전. (지금은 Large-scale supervised learning 이기에.) Appendix B.1.2

	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

Table 2: Comparison with state of the art on popular image classification benchmarks. We report mean and standard deviation of the accuracies, averaged over three fine-tuning runs. Vision Transformer models pre-trained on the JFT-300M dataset outperform ResNet-based baselines on all datasets, while taking substantially less computational resources to pre-train. ViT pre-trained on the smaller public ImageNet-21k dataset performs well too. *Slightly improved 88.5% result reported in Touvron et al. (2020).

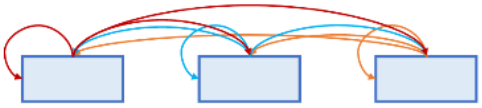
Prior Knowledge

1. Transformer (Attention is all you need) : 유투브 동빈나 참고 (https://www.youtube.com/watch?v=AA621UofTUA&t=12s&ab_channel=%EB%8F%99%EB%B9%88%EB%82%98)

- NLP에서 lstm, seq2seq의 바로 이전 단어들 과의 연관성만을 반영(Context vector 하나로 인풋 문장의 모든 정보를 압축 – 다양한 길이의 인풋 문장이 고정된 크기의 context vector로 모든 정보를 담아야 한다. 병목현상으로 성능저하)하는 것의 문제점이 존재했었고, 해결방안으로 매번 입력 문장의 출력(인코더 출력) 전부를 매 디코더 단어마다 입력으로 받도록 설계하는 방법(seq2seq with Attention)으로 해결.

Transformer는 RNN, CNN을 전혀 사용하지 않고 Attention만을 사용. (RNN, CNN을 사용하지 않으므로 positional encoding 필요)세가지 Multi head attention(Query, Key, Value 이해)를 도입하였다. Encoder, Decoder의 입력과 출력의 크기가 같기 때문에 여러 번 중첩하여 deep layer를 만들 수 있다.

■ Encoder Self-Attention

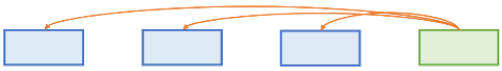


입력 문장의 각각의 단어들끼리의 모든 조합의 Attention을 구함

■ Masked Decoder Self-Attention

출력 문장의 각각의 단어들, 그러나 앞쪽 query 앞의 단어와의 Attention

■ Encoder-Decoder Attention



출력 단어의 입력 문장의 각각의 단어와의 Attention

- Positional encoding : sin, cos의 주기 함수를 이용. (각 단어에 대한 위치정보를 알수만 있다면 사용할 수 있음.) / 주기 함수 이용보다 별도의 embedding layer를 활용해서 위치 정보를 학습시키는 것이 더 효율적임

- Transformer는 computational efficiency와 scalability로 전례 없는 1000억 parameters의 모델 훈련을 가능하게 하였으며, 데이터셋과 모델이 점점 커짐에도 성능의 포화가 여전히 나타나지 않고 있다.

2. Convoultional Approaches

   - 이론적으로는 더 효율적인 Attention 기반의 CNN 모델들이 제안되나, 특이한 패턴의 사용으로 일반적인 하드웨어에서 효과적으로 큰 모델로 확장 될 수 없는 한계에 봉착해있다. 여전히 Classic ResNet-like architecture model이 SOTA 성능을 보임.(2018,2020)

3. BERT's [class] token

4. Translation equivalence : CNN이 image에 적합한 이유 (https://seoilgun.medium.com/cnn%EC%9D%98-stationarity%EC%99%80-locality-610166700979)

   - 입력의 위치가 평행이동되면 -> 출력의 위치도 똑같이 평행이동된 채로 출력된다는 것. 이미지의 location stationary 특성과 매우 잘 맞음.

아이디어 및 논문에 관한 생각

1. CNN기반 네트워크의 학습을 Transformer로 완전히 대체할 수 있다. 반대로 얘기하면 기존의 NLP에서 Transformer로 학습하던 task를 CNN의 self-attention기반으로 대체할 수 있지 않을까?

   ➔ CNN이 구조적으로 지역적인 attention을 획득하는 것에 반해, Transformer는 더 generalized 된 전역적인 attention을 구하는 architecture로 on-the-fly로 각 attention weights를 학습한다. 그러나, 누구나가 월등한 computing power를 가진 것은 아니므로, 작은 데이터 셋에서 학습 성능이 CNN의 모델이 더 뛰어난 것과 같이 NLP에서 비슷한 낱말을 새로운 dimension에 넣는다 라던지 다른 언어로의 번역을 병렬적으로 concatenate하면 CNN이 특정 task에 대해 더 효율적인 모델이 만들어질 수 있을 것 같다.

   ➔ 한계가 있을 듯. BERT, GPT와 같이 큰 데이터셋에서 학습 시켜서 fine-tune하는 게 더 나을지도.

2. Sparse convolution과의 관계는?

   - Sparse Transformers가 global self-attention을 image에 적용가능하게 하였다.