AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE

1. ABSTRACT
   a. ViT는 CNN과 함께 적용되거나 convnet의 전체적인 구조를 기반으로 특정 구성 요소를 대처하는데 사용됩니다.
   b. 훈련하는데 적은 리소스가 필요하며 대용량 데이터에 대해 사전 학습 한 뒤 다른 이미지 인식 벤치마크(ImageNet, CIFAR-100, VTAB 등)으로 전송 하였을때 우수한 결과를 얻을 수 있습니다. = 리소스 적게 들고 성능이 좋다.

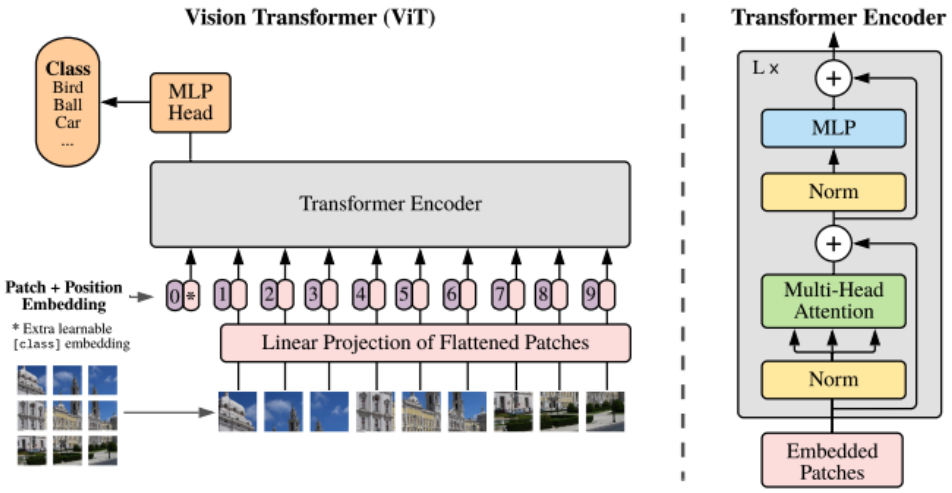
2. 모델 개요

   i. 이미지 패치화: Vision Transformer는 입력 이미지를 고정 크기의 작은 정사각형 패치로 나눕니다. 일반적으로, 이미지는 (224 x 224)의 크기로 전처리되며, 그렇게 되면 16 x 16의 크기의 패치로 나뉩니다. 이렇게 자른 패치들을 신경망에 입력합니다.

   ii. 패치 벡터화 및 위치 정보: 각 패치는 Flat한(1차원) 벡터 형태로 변환되며, 동시에 각 패치의 위치에 대한 정보가 추가됩니다. 이를 통해 Transformer는 이미지의 각 부분에 대한 위치 정보를 이해할 수 있게 됩니다.

   iii. Transformer 인코더: 벡터화하고 위치 정보를 적용한 패치들을, Transformer 인코더의 입력으로 사용합니다. 인코더는 Self-Attention 및 Feed-Forward 신경망을 사용하여 이미지에서 복잡한 패턴과 문맥 정보를 추출합니다.

   iv. `분류 토큰 및 MLP 헤드: 패치 벡터화 과정에서 특별한 분류 토큰(Classification Token)을 입력에 추가합니다. 이 토큰은 인코더를 거치면서 각 패치의 정보가 포함된 벡터로 변형되며, Transformer 인코더의 출력에서 이 토큰 부분을 사용하여 분류 작업을 수행하는 데 쓰입니다. 이를 위해 MLP 헤드가 사용되며, 벡터를 소프트맥스 활성화 함수가 적용된 최종 분류 결과로 변환합니다.



3. METHOD
   a. VISION TRANSFORMER (VIT

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

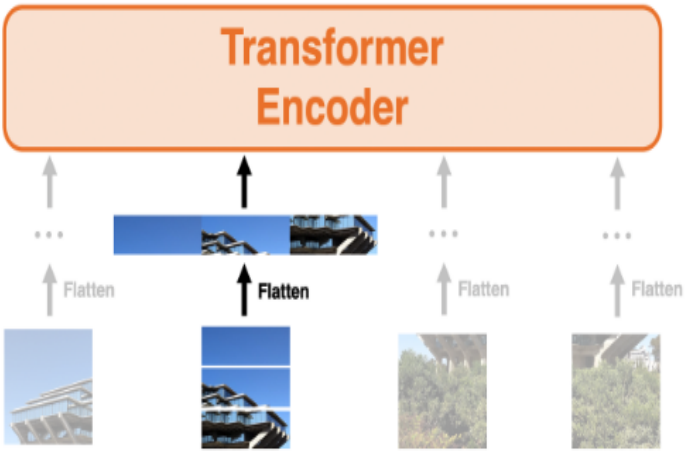
$$\mathbf{z}'_\ell = \text{MSA}(\text{LN}(\mathbf{z}_{\ell-1})) + \mathbf{z}_{\ell-1}, \qquad \ell = 1 \ldots L \tag{2}$$

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

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

- 수식 설명

각 패치를 단어처럼 다루기 때문에 실제는 패치를 벡터에서 Flatten한다. 아래의 그림에서 왼쪽에서 부터 2번째의 패치에 대해 Flatten를 실행했을 때의 모습을 보여주는 것이다. 이처럼 모든 패치에 대해 동등하게 벡터와 Flatten처리하여 엔코더에 입력한다.



마지막에는 이것을 기호를 사용하여 나타낸다. 원래 이미지가 아래와 같을 때,

$$\mathbf{x} \in \mathbb{R}^{H\times W\times C}$$

Flatten한 뒤부터 ViT에 입력하는 것은 아래와 같이 벡터로 처리하여 입력하는 것이다.

$$\mathbf{x}_p \in \mathbb{R}^{N\times(P^2\cdot C)}$$

여기서 N은 패치 수이고, P는 패치의 크기를 의미한다. 여기서 패치는 정사각형이므로, 즉 N은 다음과 같이 표현할 수 있다.

$$N = HW/P^2$$

위 이미지의 원본 이미지의 사이즈가 256라고 한다면, N은 4, P는 128이 된다. 다음은 아키텍처에 대해 살펴보자.

$$\mathbf{E} \in \mathbb{R}^{(P^2 \cdot C) \times D}$$

-> P^2*c차원의 이미지 패치를 학습가능한 *D*차원으로 매핑시키는 linear projection과정을 추가

$$\mathbf{E}_{pos} \in \mathbb{R}^{(N+1) \times D}$$

-> 각 패치에 위치 엔코딩 Epos를 계산

-> 2차원 정보를 유지하는 위치임베딩도 활용해 보았으나, 유의미한 성능향상은 없어서 1차원의 임베딩을 사용

그리고 BERT의 [class]토큰과 유사하게, 임베딩된 패치 시퀀스 앞에 학습 가능한 Xclass 임베딩을 추가
이 임베딩은 트랜스포머 인코더의 output 에서 이미지 label을 반환 하는 역할을 수행

1. Inductive bias
    a. ViT는 2차원 구조를 제한적으로 사용하기 때문에 CNN에 비해 이미지 특정 인과성(inductive bias)이 훨씬 적다.
    b. 패치 간의 모든 공간관계는 처음부터 학습이 되어야햐함.
2. Hybrid Architecture
    a. raw image patches를 대신해서 CNN의 결과 feature map을 input sequence를 형성할 수 있다.
    b. 이러한 모델에서 Patch embedding projection E는 CNN의 feature map으로 부터 추출된 패치에 적용됨
    c. 분류입력 임베딩과, 위치 임베딩은 위의 방법을 그대로 사용

b. FINE-TUNING AND HIGHER RESOLUTION
    i. ViT모델을 대규모 데이터셋에서 사전훈련하고, 작은규모의 데이터셋으로 파인튜닝을 진행함
    ii. pre-trained prediction head는 제거하고 0으로 초기화된 DxK의 feed forward 레이어 부착
    iii. 고해상도 이미지 사용시에는 더 긴 sequence length를 가지게되므로 길이에 맞춰 2D interpolation을 진행함

4. EXPERIMENTS
    a. INSPECTING VISION TRANSFORMER

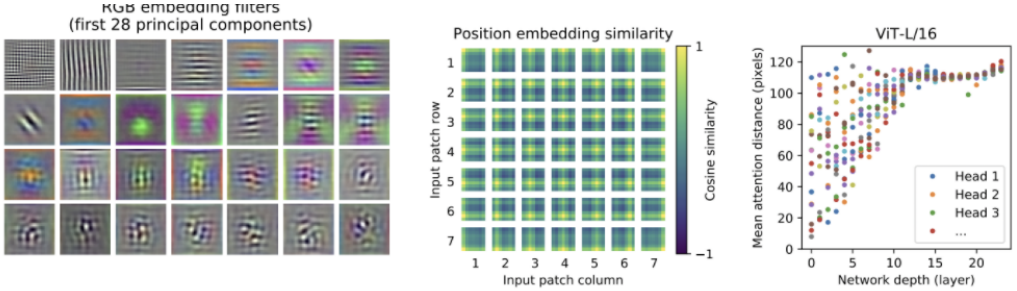


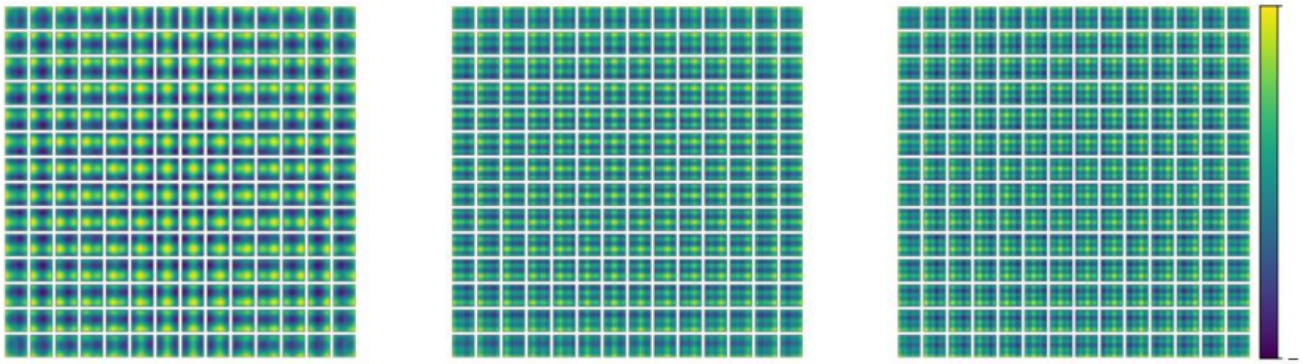
Figure 7: **Left:** Filters of the initial linear embedding of RGB values of ViT-L/32. **Center:** Similarity of position embeddings of ViT-L/32. Tiles show the cosine similarity between the position embedding of the patch with the indicated row and column and the position embeddings of all other patches. **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.

    i. ViT가 이미지를 어떻게 처리하는지 이해하기 위해 내부의 representation들을 분석

ii. ViT의 첫 번째 층은 flatten된 patch들을 저차원 공간으로 linearly project한다. 위 그림의 왼쪽 사진이 해당 층에서 학습된 임베딩 필터들의 상위 주 성분들을 보여주고있다.(이미지 patch를 flatten시키고 2차원으로 projection을 한 후에 그 행렬에 대한 주 성분을 분석한 것)
iii. 이러한 성분들은 각 패치 내의 미세 구조를 저 차원으로 표현하기 위한 그럴듯한 기저함수를 닮았다고 볼 수 있다.
iv. projection이후에는 학습된 position embedding이 patch representation에 추가된다.
v. 1d pos embedding과 2d pos embedding의 차이는 거의 없다.
vi. attention weights에 의해 통합되는 image space내의 평균거리를 계산(오른쪽 그림)
vii. 층이 깊어질수록 어텐션거리가 증가한다. 가장 아래 layer에서도 몇몇 attention head가 이미지 대부분에 attend하고있는 것을 확인할 수 있는데, 이는 정보를 global하게 integrate할 수 있는 능력을 모델이 실제로 사용하고 있음을 보여준다.
viii. local한 어텐션 능력은 hybrid 모델에서는 덜 나타난다.(local한 어텐션이 CNN의 초기 convolutional layer들과 유사한 기능을 수행함으로써)
ix. attention 길이는 depth가 깊어질 수록 증가
x. ViT가 실제로 분류에 의미적으로 관련있는 영역에 attend한다는 것을 발견
xi. 모델이 image 내부의 요소에 따라 그 부분을 분리해서 파악

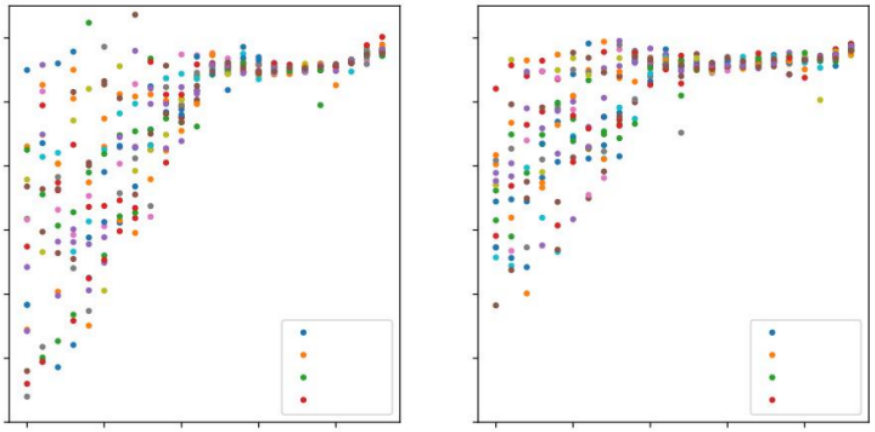
b. SELF-SUPERVISION
i. 트랜스포머기반 모델들은 대규모의 자기지도 사전훈련(self-supervised pre-training)으로부터 인상깊은 퍼포먼스를 보여주었다.
ii. 연구진들은 BERT에서 사용된 masked language modeling task를 모방하여 masked patch prediction for self-supervision를 실험해보았고, 그 결과는 scratch로부터 학습시키는 것보다 유의미한 성능향상을 가져다주었다.
iii. 그러나 여전히 지도방식의 사전훈련(supervised pre-training)에는 많이 못미치는 성능이었다.

5. POSITIONAL EMBEDDING (D.4)



a. 다양한 종류의 Pos-embedding에 대해서 실험을 진행

6. ATTENTION DISTANCE (D.7)
a. ViT가 self-attention을 사용하여 이미지 전체에 정보를 통합하는 방법을 이해하기 위해 다양한 레이어에서 attention 가중치에 따른 평균 거리를 분석했습니다.



b. 그림은 헤드 및 네트워크 깊이에 따른 영역의 크기. ATTENTION DISTANCE는 픽셀과 다른 모든 픽셀 사이의 거리를 가중치를 적용하여 평균화. (224픽셀 128개의 예시 이미지)
c. ATTENTION DISTANCE는 CNN에서 수용장 크기와 유사한 개념이며, 하위 레이어에서는 어텐션 거리가 다양하지만 네트워크의 두 번째 절반에서는 대부분의 헤드가 토근 전체에 걸쳐 넓게 어텐션을 기울임.