Vision Transformer(vit)

by beomgon.yu

Contents

Introduction

1. NLP에서 transformer가 큰 성공을 거두었다.
   a. 가볍고 단순하며 scalable
   b. 자연어처리를 넘어서 음성인식 이나 비전, video등 다른 도메인으로 확장되는 중

2. Vision에서도 attention을 이용한 시도
   a. CNN backbone에 일부 모듈 attention 삽입
      SEnet, non local network
   b. fully attentional model
      Stand-alone self-attention in vision models
      scalable하지 않음, 구현이 어려움(specialized attention patterns)
   c. sparse transformer - global attention(pixel단위 attention시 seq len is too long -> global attention)
   d. Axial-deeplab: Stand-alone axial-attention for panoptic segmentation
   e. video등에서 transformer/bert 구조를 사용함, VideoBert
      image/video + text
      image tokenization 필요, clustering
   f. Segmentation - DETR(transformer), deformable DETR
   g. iGPT - 유사

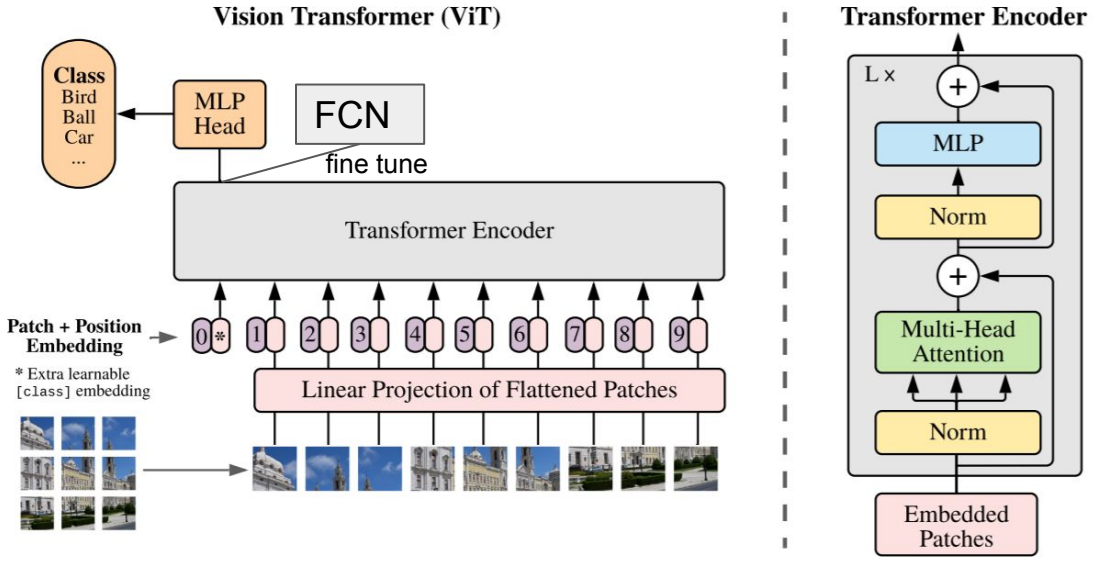
3. 자연어 처리에서 쓰는 것과 거의 유사하게 transformer model을 이용한 것은 vit가 처음
   a. CNN에 비해 가벼우면서 성능은 sota를 찍음
   b. CNN is not necessary
   c. Big Transfer와 같이 large dataset을 사용(천만에서 수억장의 images)

Introduction

1. Large scale dataset
    a. because of inductive bias, cnn is better than transformer,
    (translation invariance/equivariance, locality)
    big size of data can solve this problem.
    (ImageNet-21k and JFT-300M.)

2. tokenization
    a. 16*16 patches - linear projection (similar to filter in low layer in CNN)

Model



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

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

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

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

1vs 2d positional embeding
hybird model
ex)
256*256*3 -> 768*256

Downstream task
remove MLP, use zero initialized linear layer
high resolution is better(than pretrain)
positional embeding is give by interpolation of org image

how about relative positional embeding?

Experiment

Datasets.

- Pre-train dataset
  - ImageNet, 1K classes, 1.3M images
  - ImageNet-21k, 21K classes, 14M images
  - JFT, 18K classes, 303M images

Transfer Learning dataset

- ImageNet
- ImageNet-Real
- CIFAR 10/100
- Oxford-IIIT Pets
- Oxford Flowers-102
- 19-task VTAB classification suite
  - VTAB evaluates low-data transfer to diverse tasks, using 1 000 training examples per task

Experiment

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

lots of memory, for pretraining, batch size should be small and training time would be long because of lots of data size
only fine tuning, use pretrain model

Training & Fine-tuning

- Pre-train all models, including ResNets, using Adam with $\beta_1 = 0.9$, $\beta_2 = 0.999$, a batch size of 4096 and apply a high weight decay of 0.1.
- For fine-tuning they use SGD with momentum, batch size 512.

Metrics

- Report results on downstream datasets either through few-shot or fine-tuning accuracy

resnet152 : 60M, but computational overhead is worse that transformer

result

	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

result

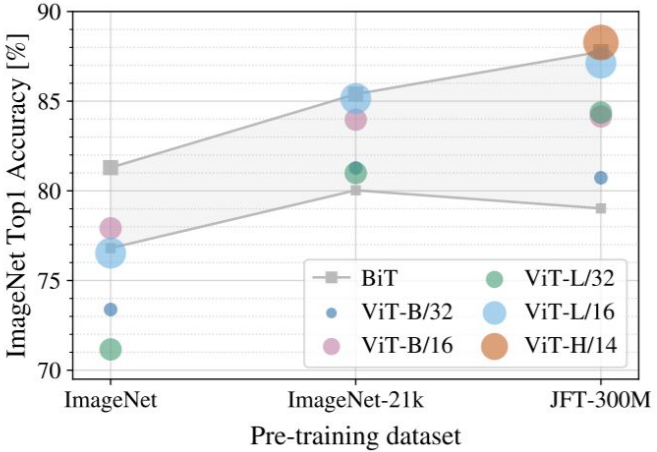


Figure 3: Transfer to ImageNet. While large ViT models perform worse than BiT ResNets (shaded area) when pre-trained on small datasets, they shine when pre-trained on larger datasets. Similarly, larger ViT variants overtake smaller ones as the dataset grows.

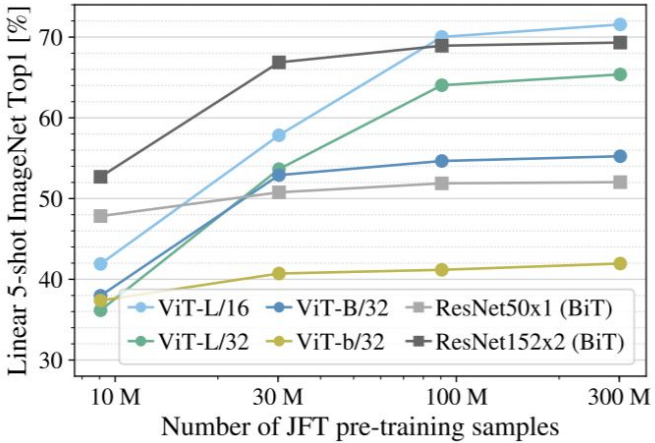
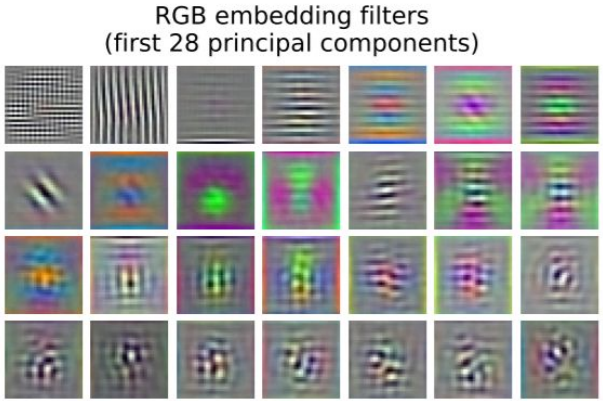


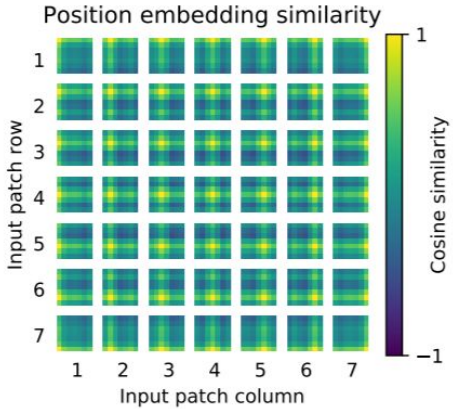
Figure 4: Linear few-shot evaluation on ImageNet versus pre-training size. ResNets perform better with smaller pre-training datasets but plateau sooner than ViT, which performs better with larger pre-training. ViT-b is ViT-B with all hidden dimensions halved.

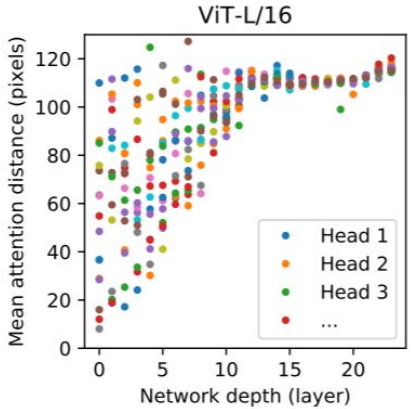
INSPECTING VISION TRANSFORMER

for Linear embeding, convolution is used

```
self.embedding = nn.Conv2d(3, emb_dim, kernel_size=(fh, fw), stride=(fh, fw))
```







attention distance is like receptive field
how got get attention distance, use attention weights,
Sigma(w*d)/seq_len

Self supervision

1. masking is introduced(like bert), denoising auto encoder.
    a. 2% more than scratch but less than pretrain(jft-300)

        50% of patch embeddings by either replacing their embeddings with a learnable [mask] embedding (80%), a random other patch embedding (10%) or just keeping them as is (10%).
        we predict the 3-bit, mean color (i.e., 512 colors in total) of every corrupted patch using their respective patch representations, maybe MSB?

    b. future work for contrastive learning
        i. pre training
        ii. fine tuning
2. More work
    a. object detection or segmentation(DETR)
    b. better self supervised learning, (need to check iGPT)

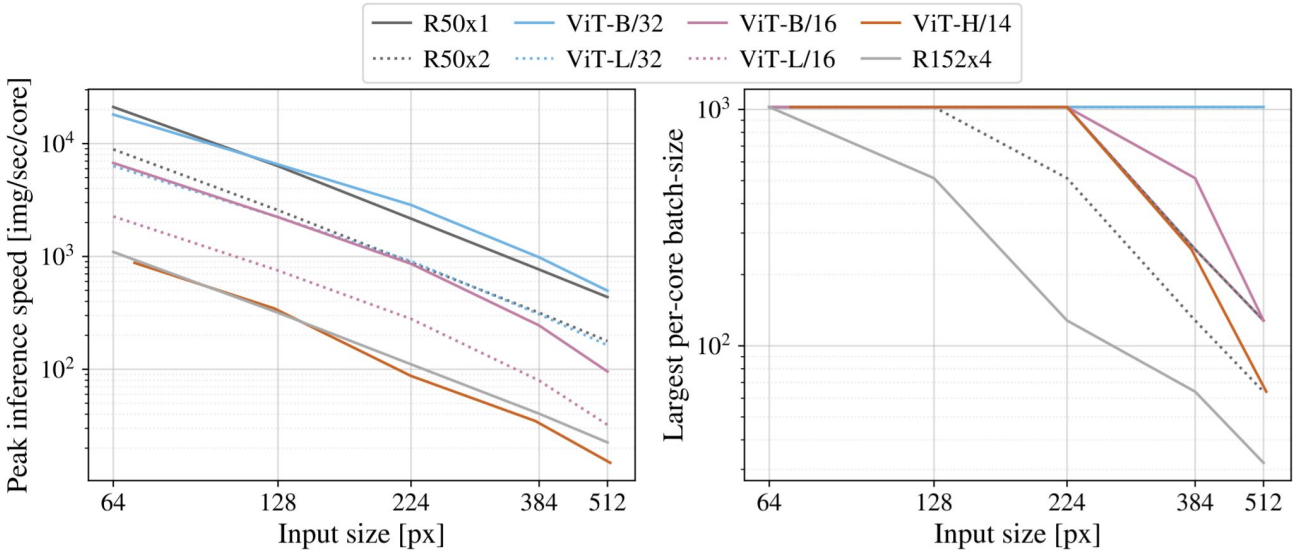
appendix

1. positional embedding
   a. 1d, 2d, relative
   b. where to feed
   c. when fine tuning, higher resolution is used, use linear interpolation for positional embeding of high resolution

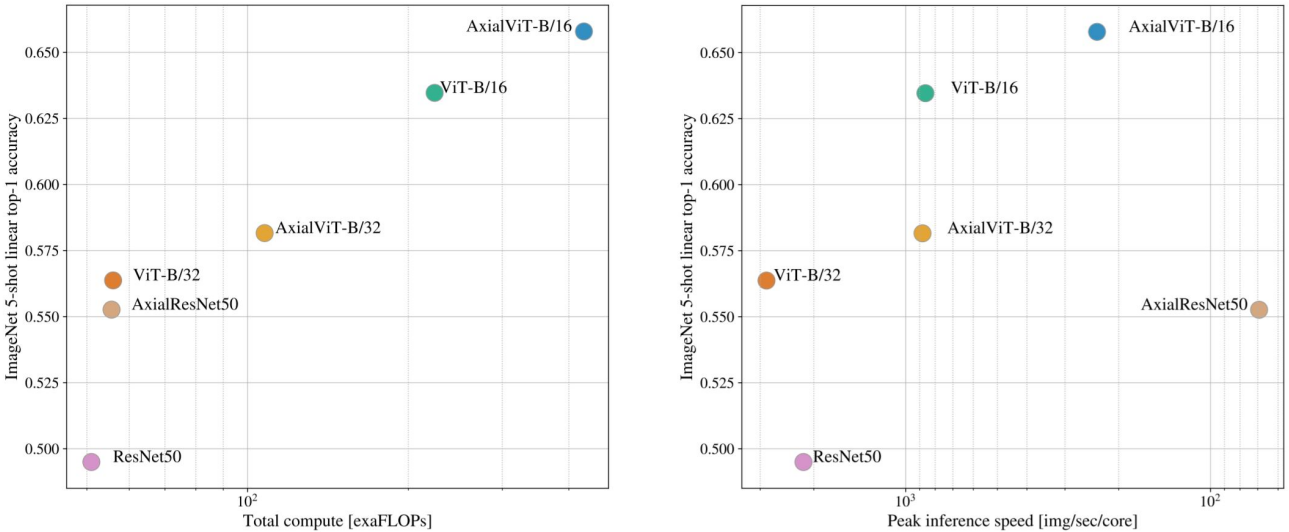
Pos. Emb.	Default/Stem	Every Layer	Every Layer-Shared
No Pos. Emb.	0.61382	N/A	N/A
1-D Pos. Emb.	0.64206	0.63964	0.64292
2-D Pos. Emb.	0.64001	0.64046	0.64022
Rel. Pos. Emb.	0.64032	N/A	N/A

14 × 14 as opposed to 224 × 224, and learning to represent the spatial relations in this resolution is equally easy for these different positional encoding strategies

COMPUTATIONAL COSTS



Axial attention

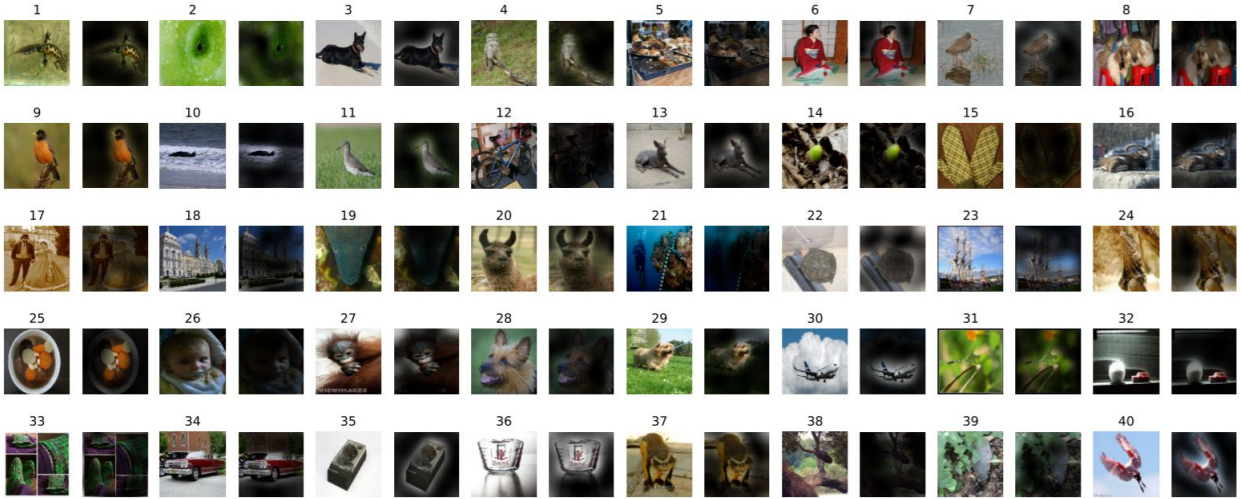


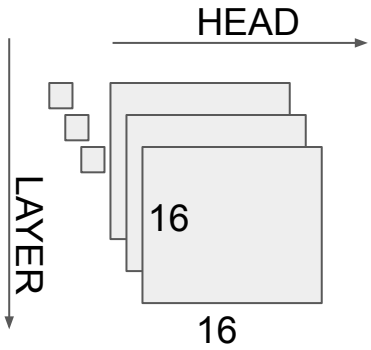
accuracy/compute trade-off, implementation is extremely slow on TPUs

ATTENTION MAPS

To compute maps of the attention from the output token to the input space, we used Attention Rollout (Abnar & Zuidema, 2020). Briefly, we averaged attention weights of ViTL/16 across all heads and then recursively multiplied the weight matrices of all layers. This accounts for the mixing of attention across tokens through all layers.

a kind of explanable AI, visualization is easier then cnn



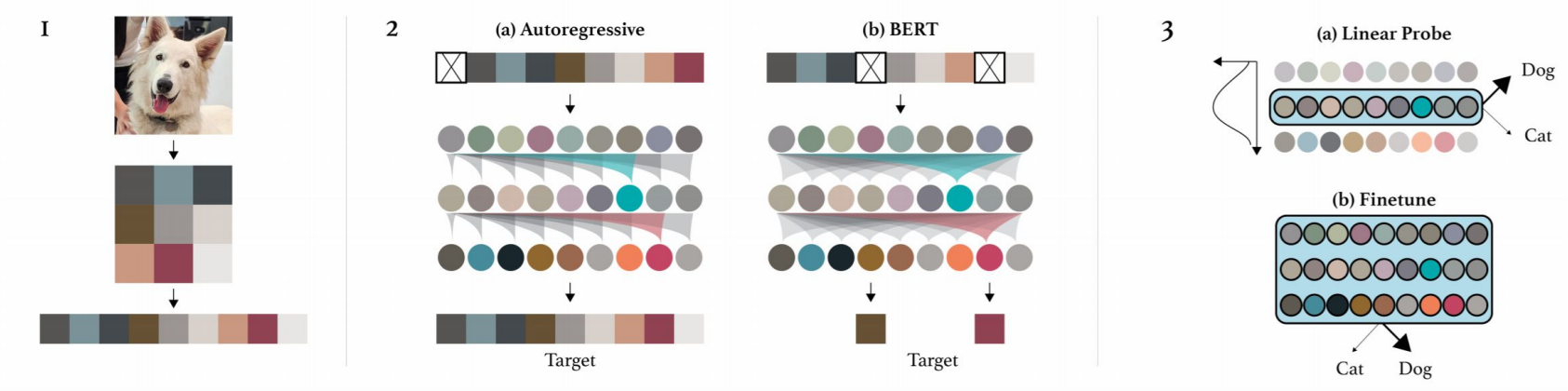


Question

1. about patch size?
   a. overlapping window is used? -> NO
   b. if overlapping window is used, close distance attention is better,
      effect of cnn
      image is continuous, not discrete
   c. Tokens-to-Token ViT

2. self supervised learning
   a. 3-bit, mean color means MSB of 512 colors??
      RGB 3bit, 8*8*8 = 512

iGPT

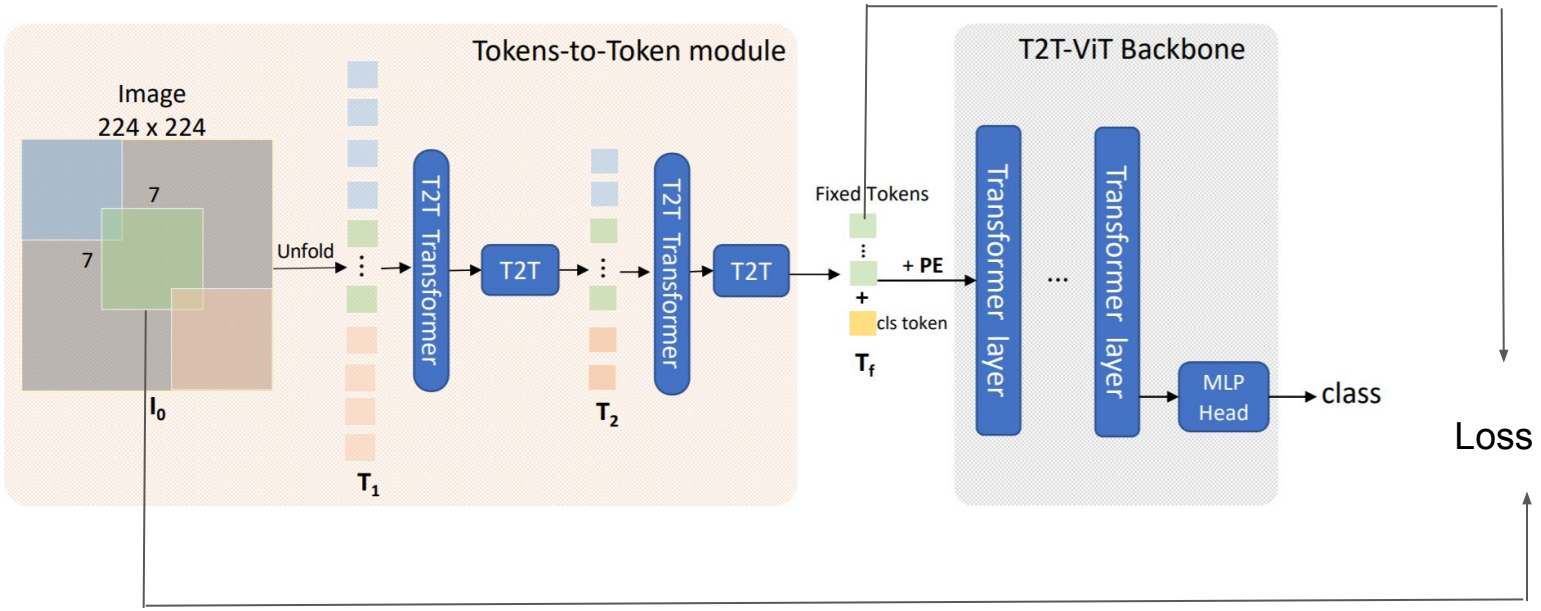


1. there are heuristic in tokenization
   Use linear projection in vit, for 1. use pretrained value
   is it possible, label is not fixed, varing or should be calculated during training, possible but complex

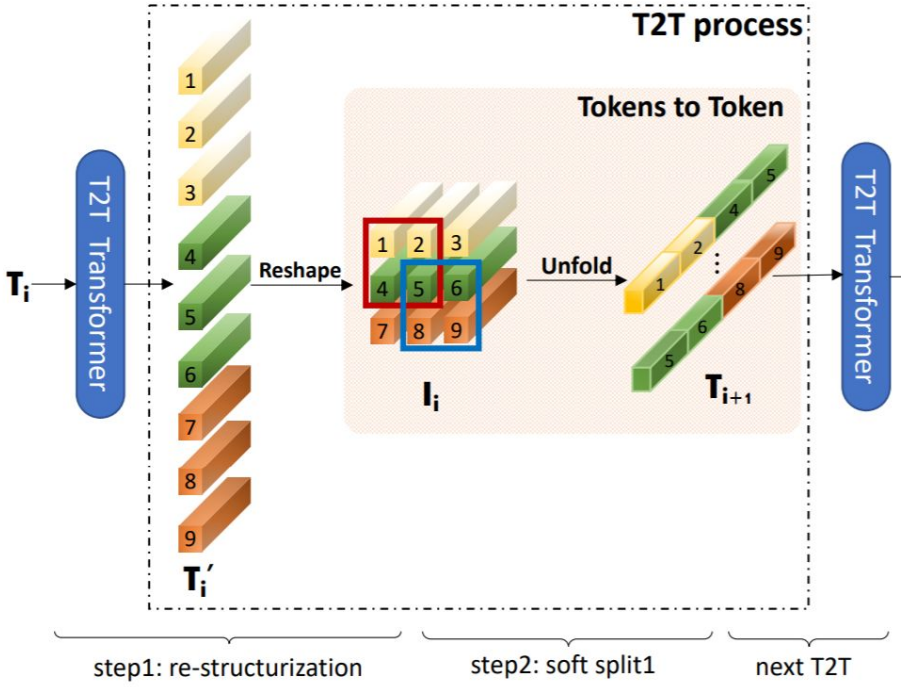
Tokens-to-Token ViT

1. vit is not good at low datasets or from scratch
    a. the simple tokenization of input images fails to model the important local structure (e.g., edges, lines)
    b. the redundant attention backbone design of ViT leads to limited feature richness in fixed computation budgets and limited training samples

Tokens-to-Token ViT



Tokens-to-Token ViT



Tokens-to-Token ViT

1. self supervised learning에서 Lable이 고정되어 있지 않고 변한다.
   구현은 가능할 것 같다. 그러나 computational overhead,
   fixed token을 low dimension으로 linear projection을 해주고, mlp block에서 class개수가 아닌 fixed token의 low dimension으로 Linear projection한다. 전체 fixed token과 내적 후 select max target.
   그러나 학습의 의미가 없을 수도 있다.
   mask를 씌워도 raw token에서 하는 것보다 찾기가 쉬울 듯, 즉 학습이 쉬워지는 문제 발생..

2. And cnn으로 feature를 줄이면 어떨까??
   vit의 hybrid model과 유사하지만 여기서는 seq len을 줄인다.

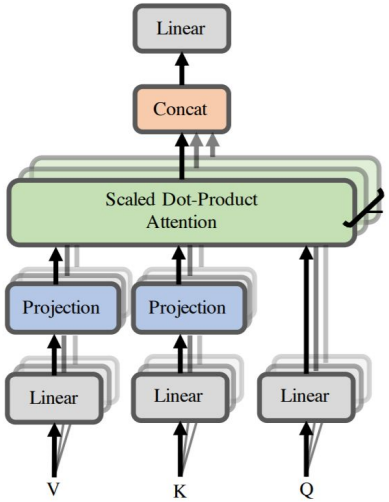
3. just use Linear projection like linformer
   in 자연어처리, total seq len should be keeped, but image, no need to keep.
   easy to implement, but 정보 손실 발생, cnn과 같이 feature map의 size는 줄이면서 dim을 늘리는 방식으로.
   but there are many redundancy in image itself, rank of input embeding matrix maybe very row.
   linear projection is simple
   there are trade off linear projection vs CNN(tokens to token VIT)
   position information maybe disapper in linear projection
4. in Linformer, only key and value has linear projection layer, query has still position information
5.

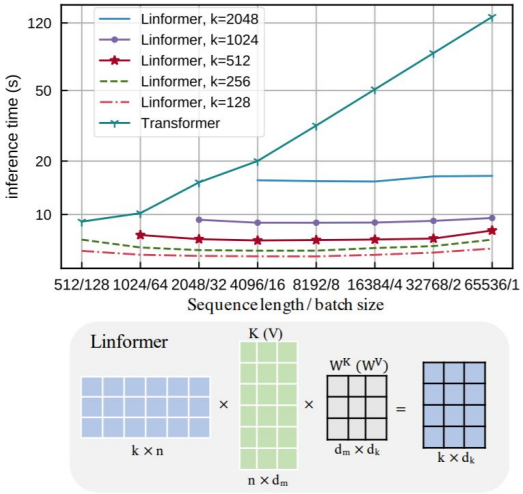
Tokens-to-Token ViT

Table 3. Comparison between our T2T-ViT with ResNets on ImageNet. T2T-ViT$_t$-14: using Transformer in T2T module. T2T-ViT-14: using Performer in T2T module. * means we train the model with our training scheme for fair comparisons.

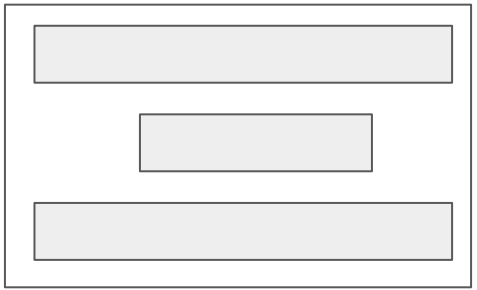
Models	Top1-Acc (%)	Params (M)	MACs (G)
ResNet50 [15]	76.2	25.5	4.3
ResNet50*	79.1	25.5	4.3
T2T-ViT-14	**80.6**	21.4	4.8
T2T-ViT$_t$-14	**80.7**	21.5	5.2
ResNet101 [15]	77.4	44.6	7.9
ResNet101*	79.9	44.6	7.9
T2T-ViT-19	**81.2**	39.0	8.0
T2T-ViT$_t$-19	**81.4**	39.0	8.4
ResNet152 [15]	78.3	60.2	11.6
ResNet152*	80.8	60.2	11.6
T2T-ViT-24	**81.8**	63.9	12.6
T2T-ViT$_t$-24	**82.2**	64.1	13.2

Linformer



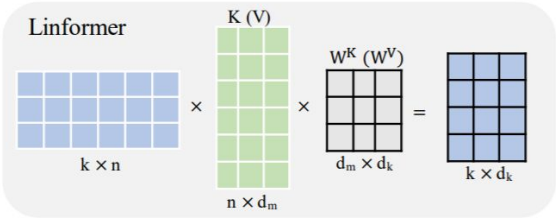


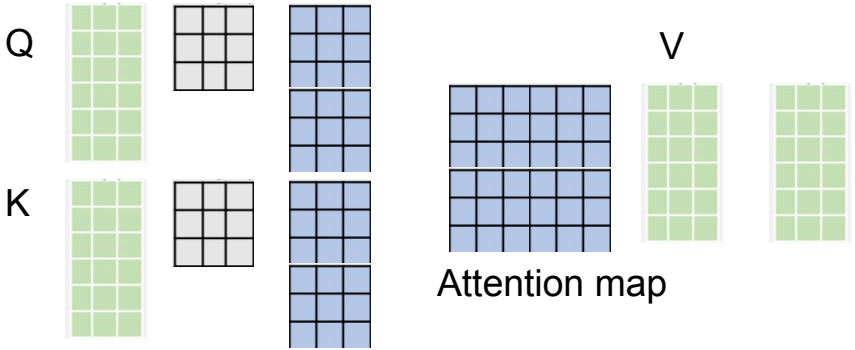
X layer

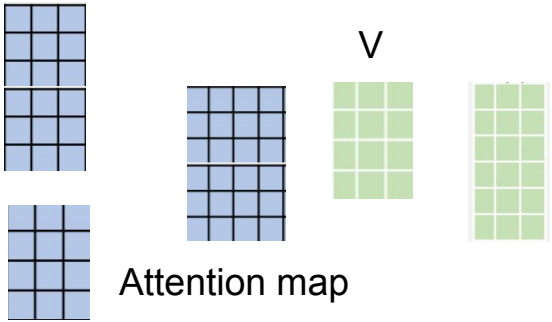


Linformer

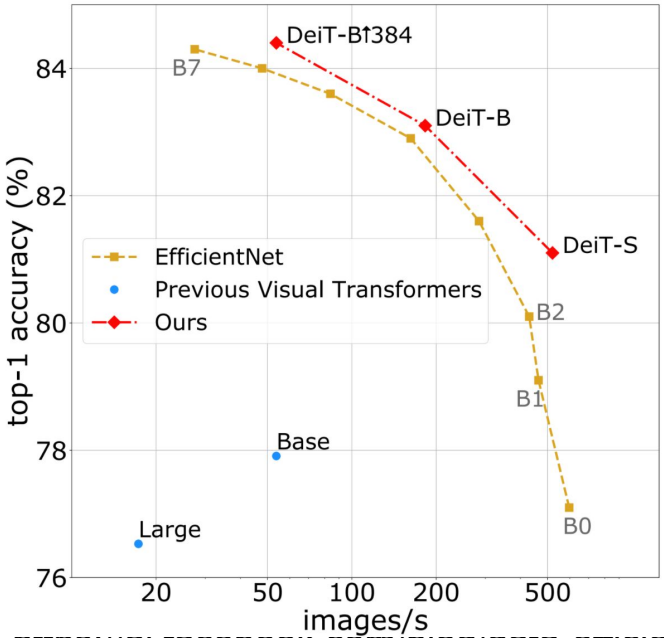
1. fixed size input
2. query has position information
   (only V and K has linear projection)







Training data-efficient image transformers and distillation through attention

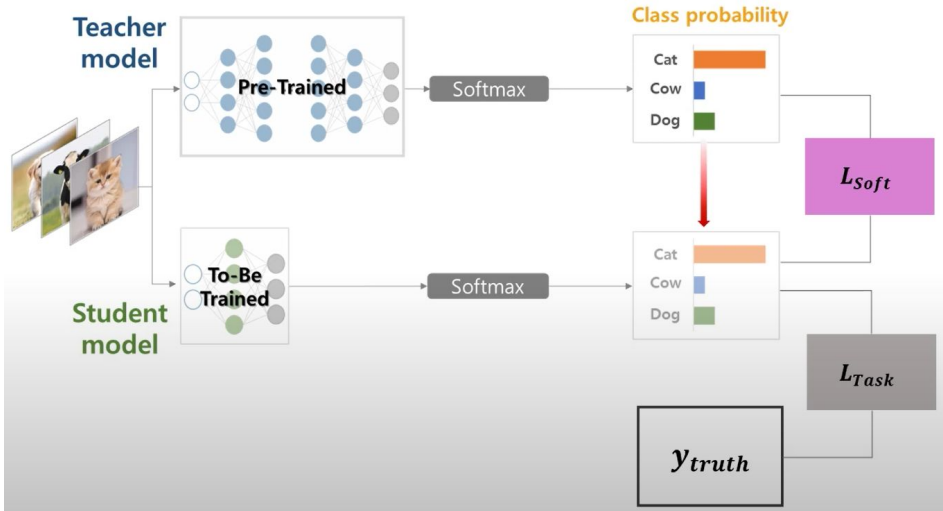


vit + distillation
small data size + data augmentation

https://ai.facebook.com/blog/data-efficient-image-transformers-a-promising-new-technique-for-image-classification/

Training data-efficient image transformers and distillation through attention

Distillation 방법 : Offline - distillation



- $f_T(x_i)$: Teacher 모델의 logit 값
- $f_T(x_i)$: Student 모델의 logit 값
- τ :Scaling 역할의 하이퍼 파라미터

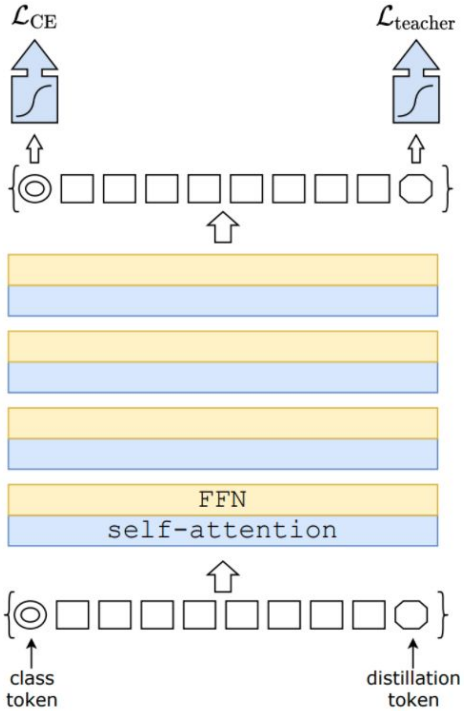
$$L_{Soft} = \sum_{x_i \in X} KL(softmax(\frac{f_T(x_i)}{\tau}), softmax(\frac{f_s(x_i)}{\tau}))$$

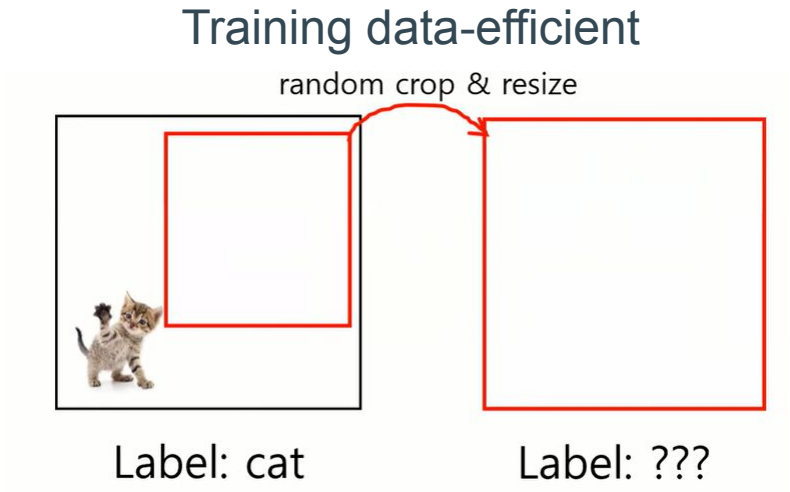
$$L_{Task} = CrossEntropy\ (softmax(f_s(x_i)), y_{truth})$$

$$Student\ L_{Total} = L_{Task} + \lambda \cdot L_{Soft}$$

https://www.youtube.com/watch?v=pgfsxe8sROQ

Training data-efficient image transformers and distillation through attention





self supervised learning

BYOL, simclr

pretrained model + self supervised learning + fine tuning

https://github.com/lucidrains/vit-pytorch
https://reposhub.com/python/deep-learning/lukemelas-PyTorch-Pretrained-ViT.html
https://github.com/asyml/vision-transformer-pytorch

further list

1. transformer in vision의 목표는 ??

1. imagenet의 data만 가지고도 CNN만큼의 performance(top1 accuracy)를 내는 것.
spatial domain 줄이고 depth(channel) 늘리고, like cnn
tokens to tokens
linear projection이 cnn model의 1st level cnn이라고 할때 filter size, depth가 너무 많다.
depth를 유지하기 위함이지만, depth를 variable하게 한다면 처음 depth를 줄일 수 있다.
16*16 patches : 16*16 *768(suppose image is 256*256*3)
8*8 patches : 32*32 *64 (resnet low 부분 참고, 7*7*64 filter is used)
: 4*4 * 1024, after global average pooling + FCN(no need to use global token)

2. parameter size down

1번을 하면 자연스럽게 parameter size도 줄어들 것