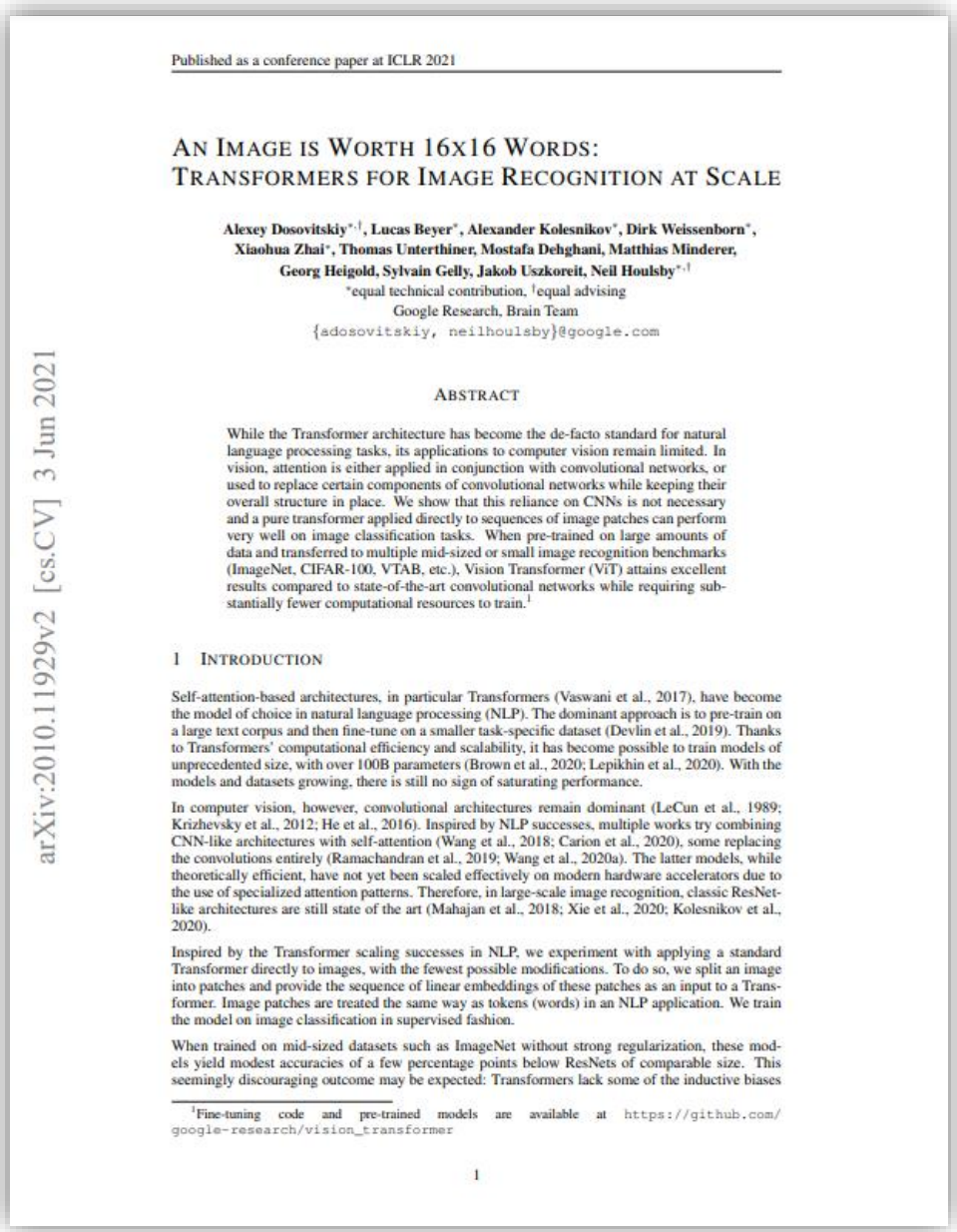
VISION TRANSFORMER REVIEW

2023.08.09 김채원



Published as a conference paper at ICLR 2021

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

Alexey Dosovitskiy*,†, Lucas Beyer*, Alexander Kolesnikov*, Dirk Weissenborn*, Xiaohua Zhai*, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby*,†

*equal technical contribution, †equal advising

Google Research, Brain Team

{adosovitskiy, neilhoulsby}@google.com

ABSTRACT

While the Transformer architecture has become the de-facto standard for natural language processing tasks, its applications to computer vision remain limited. In vision, attention is either applied in conjunction with convolutional networks, or used to replace certain components of convolutional networks while keeping their overall structure in place. We show that this reliance on CNNs is not necessary and a pure transformer applied directly to sequences of image patches can perform very well on image classification tasks. When pre-trained on large amounts of data and transferred to multiple mid-sized or small image recognition benchmarks (ImageNet, CIFAR-100, VTAB, etc.), Vision Transformer (ViT) attains excellent results compared to state-of-the-art convolutional networks while requiring substantially fewer computational resources to train.[1]

1 INTRODUCTION

Self-attention-based architectures, in particular Transformers (Vaswani et al., 2017), have become the model of choice in natural language processing (NLP). The dominant approach is to pre-train on a large text corpus and then fine-tune on a smaller task-specific dataset (Devlin et al., 2019). Thanks to Transformers' computational efficiency and scalability, it has become possible to train models of unprecedented size, with over 100B parameters (Brown et al., 2020; Lepikhin et al., 2020). With the models and datasets growing, there is still no sign of saturating performance.

In computer vision, however, convolutional architectures remain dominant (LeCun et al., 1989; Krizhevsky et al., 2012; He et al., 2016). Inspired by NLP successes, multiple works try combining CNN-like architectures with self-attention (Wang et al., 2018; Carion et al., 2020), some replacing the convolutions entirely (Ramachandran et al., 2019; Wang et al., 2020a). The latter models, while theoretically efficient, have not yet been scaled effectively on modern hardware accelerators due to the use of specialized attention patterns. Therefore, in large-scale image recognition, classic ResNet-like architectures are still state of the art (Mahajan et al., 2018; Xie et al., 2020; Kolesnikov et al., 2020).

Inspired by the Transformer scaling successes in NLP, we experiment with applying a standard Transformer directly to images, with the fewest possible modifications. To do so, we split an image into patches and provide the sequence of linear embeddings of these patches as an input to a Transformer. Image patches are treated the same way as tokens (words) in an NLP application. We train the model on image classification in supervised fashion.

When trained on mid-sized datasets such as ImageNet without strong regularization, these models yield modest accuracies of a few percentage points below ResNets of comparable size. This seemingly discouraging outcome may be expected: Transformers lack some of the inductive biases

[1]Fine-tuning code and pre-trained models are available at https://github.com/google-research/vision_transformer

Conference : ICLR (2021)

Published date: 2020.10.22

Author : Alexey Dosovitskly, Lucas Beyer, Alexander Kolensnikov et al. (Google Research)

1.Abstract

- In NLP , the transformer structure has established standard model

- However, in computer vision, convolution architectures remain dominant

- Inspired by NLP successes, multiple works try combining CNN with self-attention or uses to replace only a few elements based on the CNN structure.

(Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In CVPR, 2018.)

(Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jon Shlens. Stand-alone self- attention in vision models. In NeurIPS, 2019.)

1.Abstract

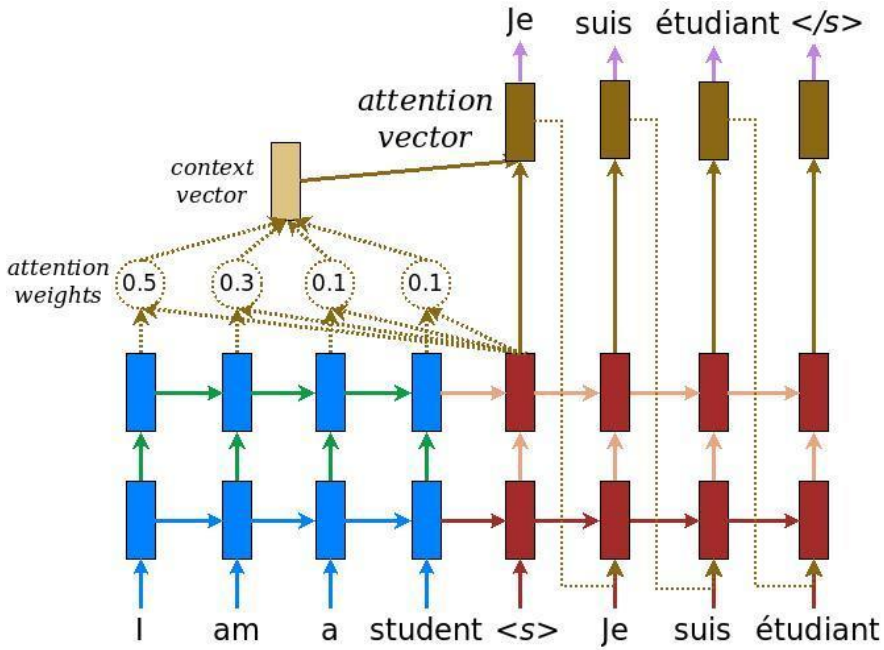
- Idea : Let's apply transformer directly to the image

- They show that CNNs is not necessary in computer vison and pure transformer applied directly to image patches can perform very well

- When pre-trained on large amounts of data and transferred to multiple mid-sized or small image recognition benchmarks(ImageNet,CIFAR-100, etc ..), Vision Transformer(ViT) attains excellent result compared to CNN

2.Introduction

- In NLP, self-attention-based architectures (especially Transformer) have become standard model



2.Introduction

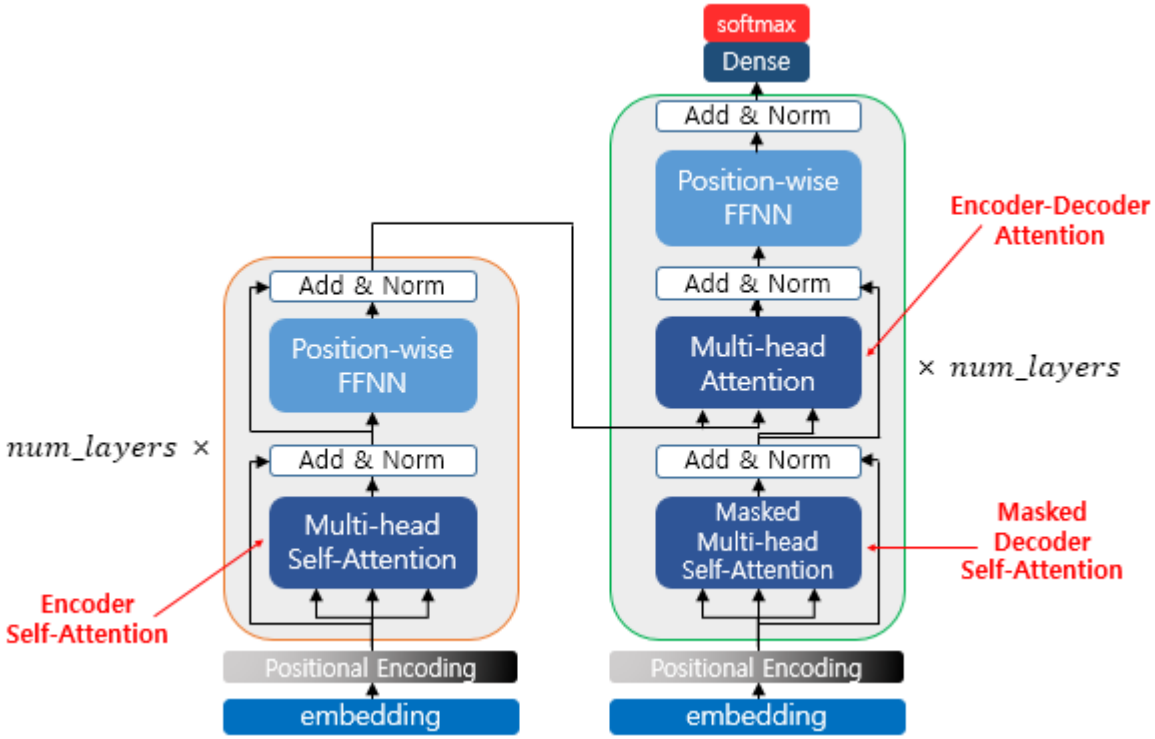
- Thanks to the efficiency and scalability of transformer, it's possible to train larger size model than in the past

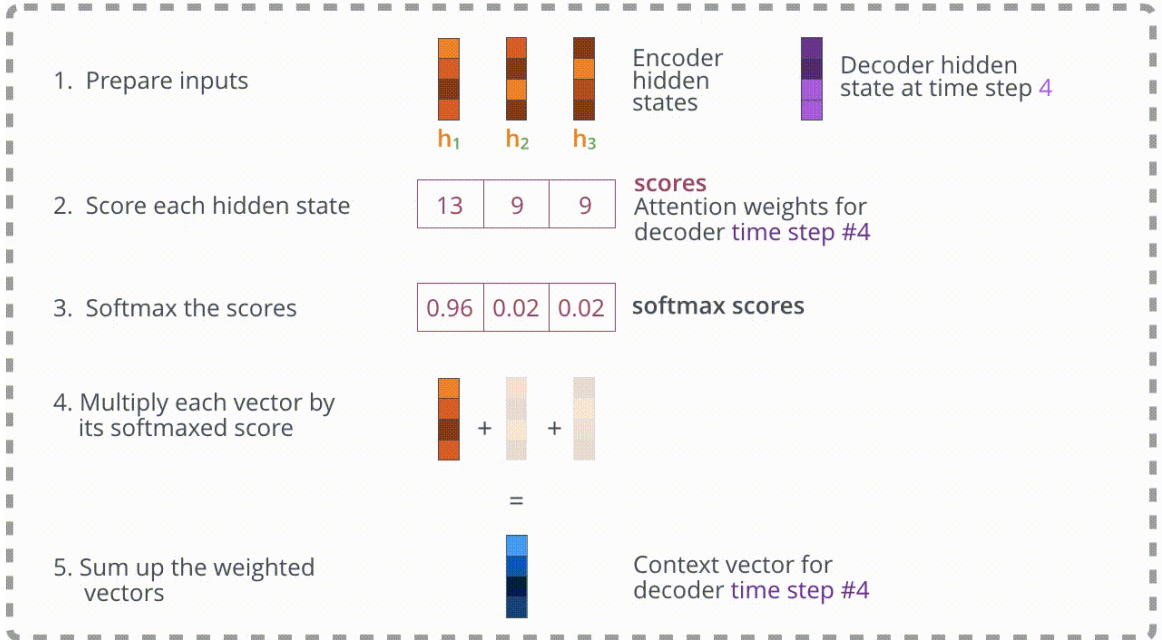
- In addition, despite the increase in model and data size, performance was not saturated

- Why?

2.Introduction

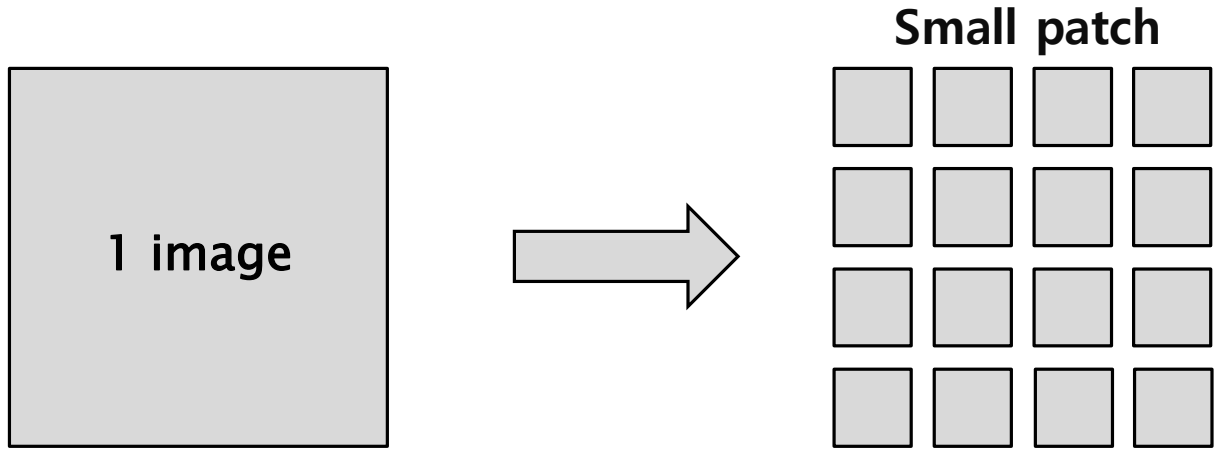
- Transformer





2.Introduction

- Nevertheless, the reason why CNN was dominant in computer vision was that attention's calculation method did not work efficiently in gpu to apply it to images



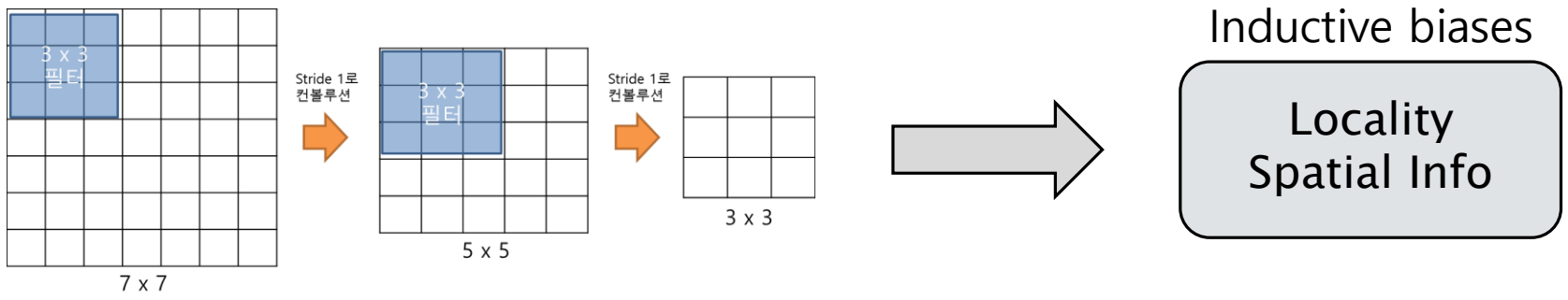
기존방식

새로운 방식

2.Introduction

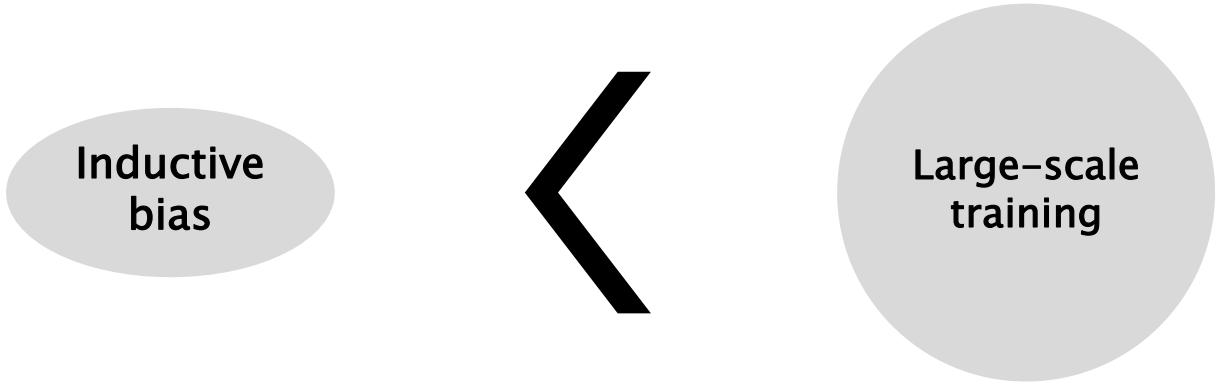
- However, when training for medium-sized datasets (data amount) such as ImageNet, It shows a few % lower accuracy than ResNet

- Because Transformers lack inductive biases such as translation equivariance and locality inherent in CNN.



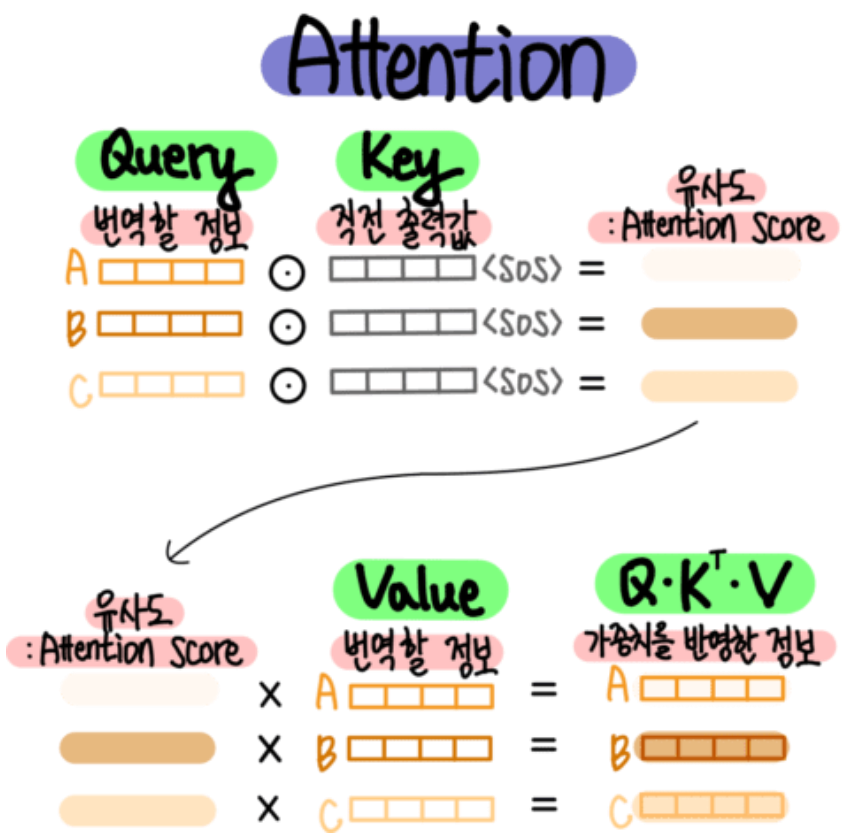
2.Introduction

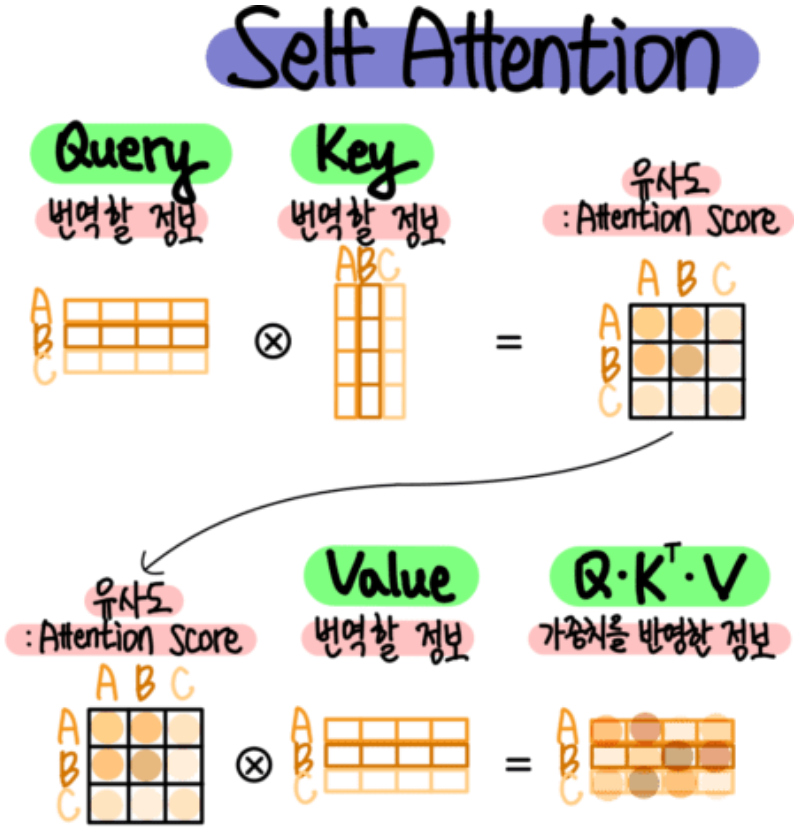
- When training for big-sized datasets (data amount), It shows great results that match or exceed the performance of existing SOTA models



3.Related work

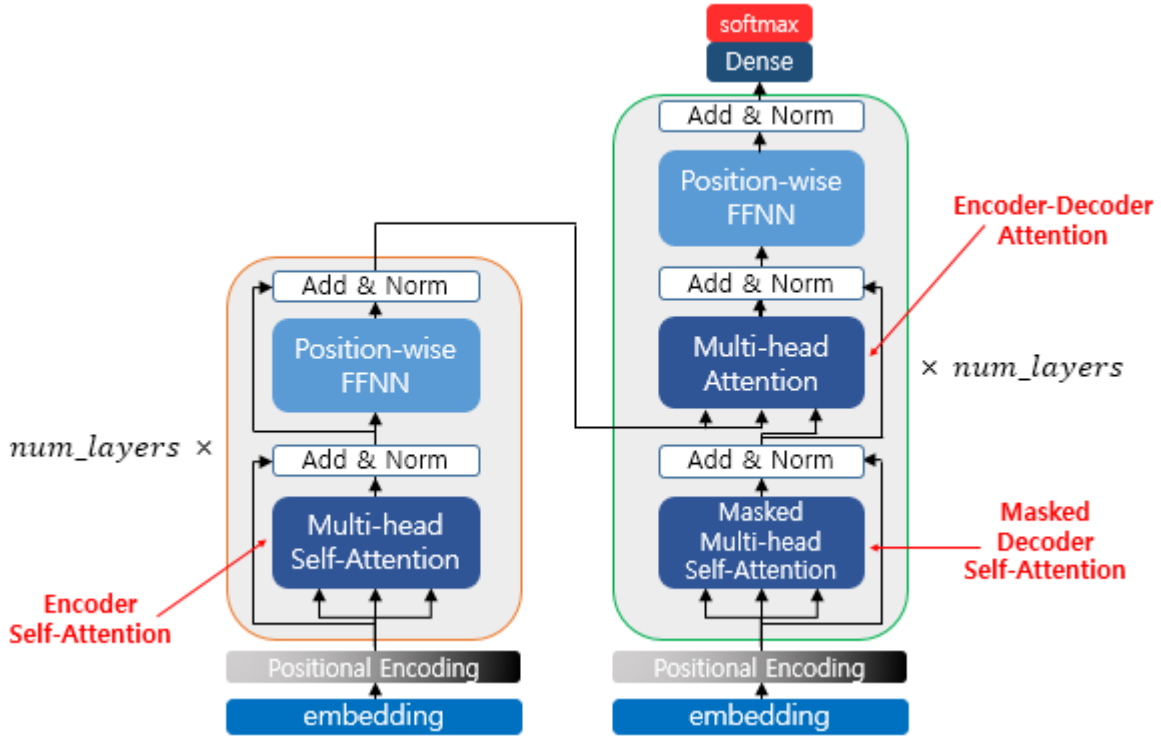
- self-attention





3.Related work

- Transformer



3.Related work

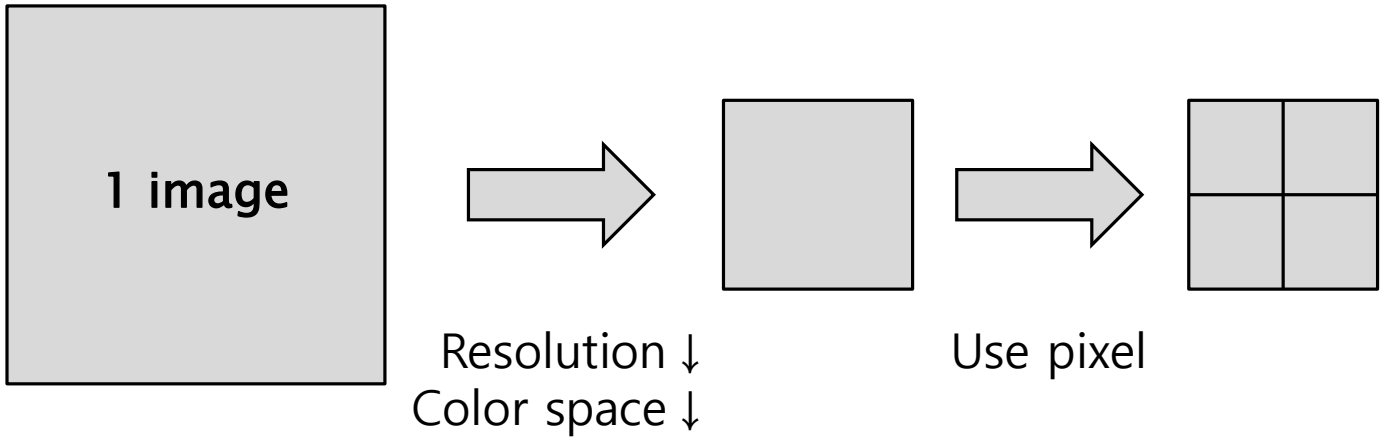
- Prior research to use transformer in computer vision

1. local self-attention
2. sparse attention
3. applying it in blocks of varying size
4. feature map augmentation using self-attention
5. add attention to CNN's output

global self-attention

3.Related work

- iGPT



4.Method

- ViT

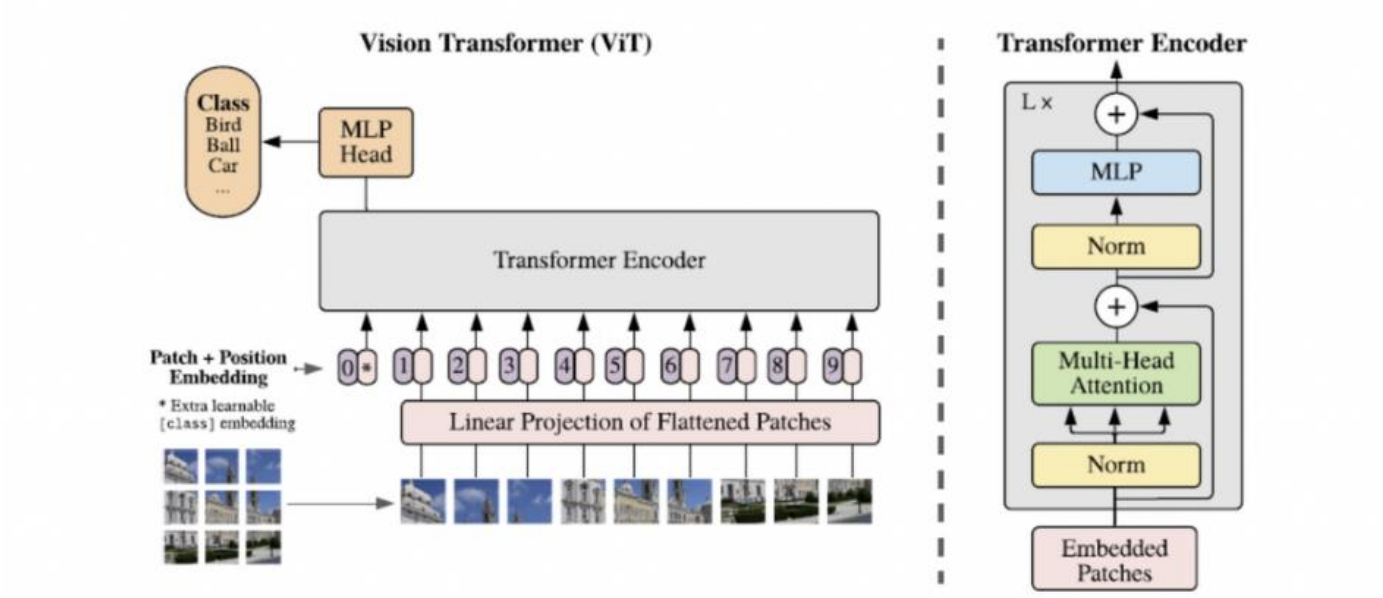
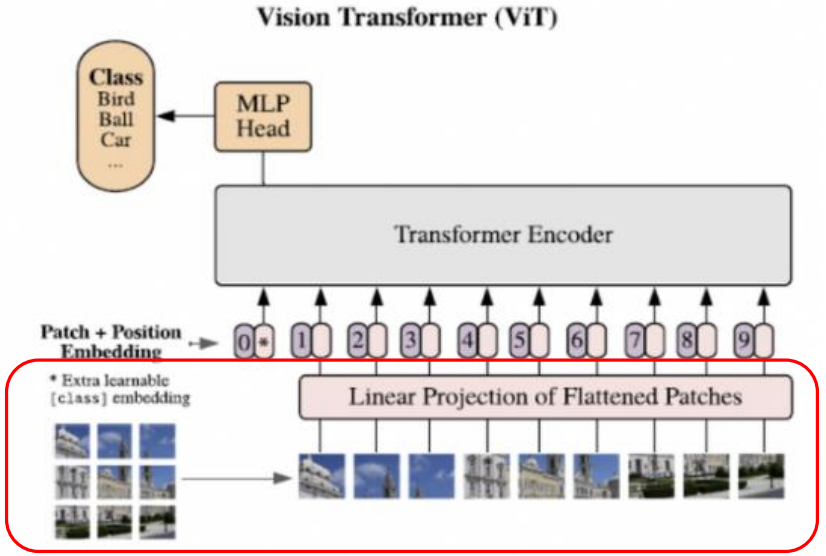


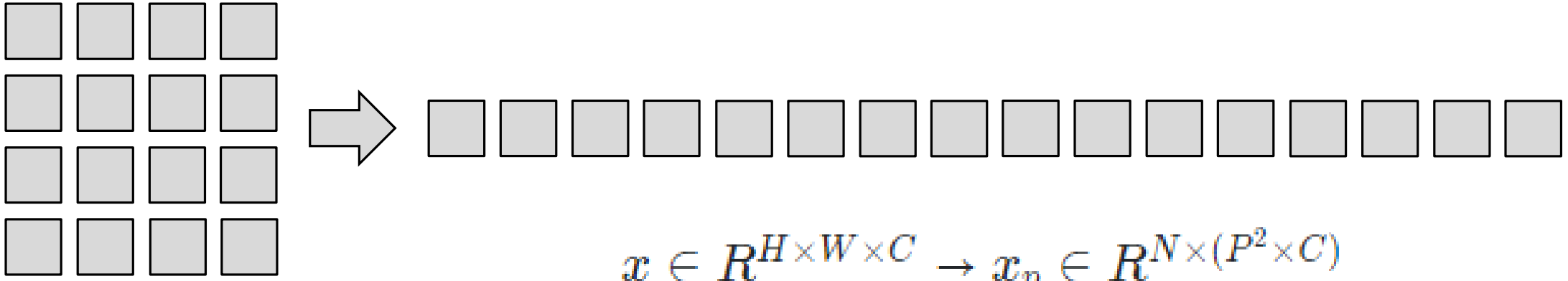
Figure 1: Model overview. We split an image into fixed-size patches, linearly embed each of them, add position embeddings, and feed the resulting sequence of vectors to a standard Transformer encoder. In order to perform classification, we use the standard approach of adding an extra learnable "classification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

4.Method

- Squeeze



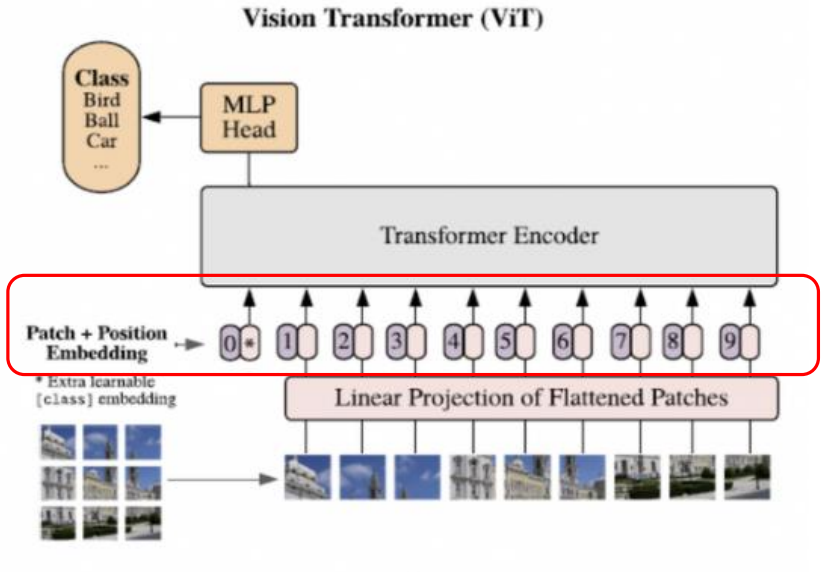
Small patch



$$x \in R^{H \times W \times C} \rightarrow x_p \in R^{N \times (P^2 \times C)}$$

(H,W) : resolution of image (P,P) : resolution of patch
N : num of patch (HW/p^2) C : num of channels

4.Method



- Classification token & Positional Embedding

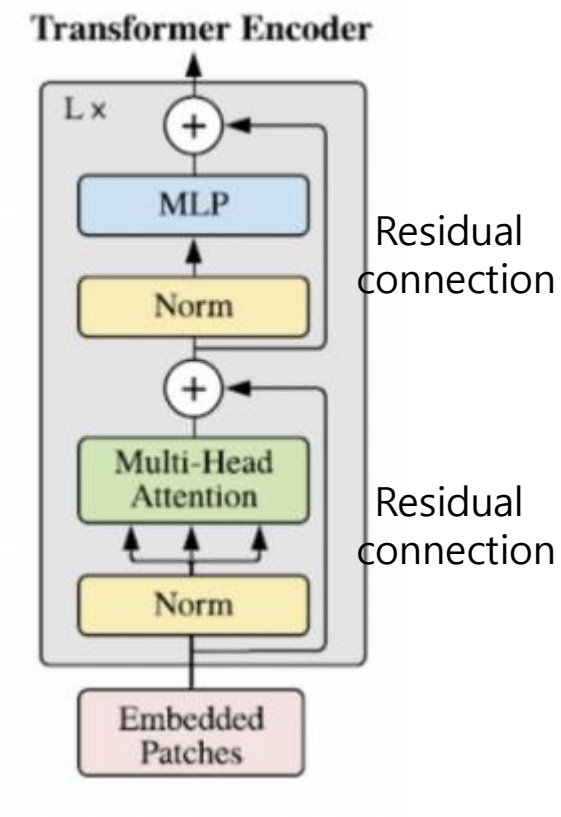
- Add one learnable class token embedding vector to the front of the patches

- At the final output, it serves as a one-dimensional representation vector for the image

- Also, add positional embedding to maintain image location information

4.Method

- Transformer Encoder



(1) $z_0 = [x_{class}; x_p^1 E; x_p^2 E; \cdots; x_p^N E] + E_{pos},\ E \in R^{(P^2 \cdot C) \times D},\ E_{pos} \in R^{(N+1) \times D}$

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

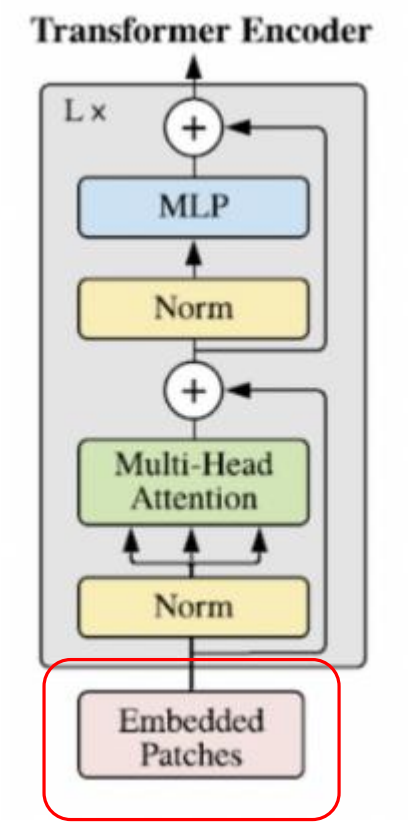
(3) $z_\ell = MLP(LN(z'_\ell)) + z'_\ell,\ \ell = 1 \ldots L$

(4) $y = LN(z_L^0)$

MLP(Multi-Layer Perceptron)
↑
Layer Normalization
↑
MSA(Multi-head Self-Attention)
↑
Layer Normalization

4.Method

- Transformer Encoder



patch

$$(1)\ z_0 = [x_{class}; x_p^1 E; x_p^2 E; \cdots; x_p^N E] + E_{pos},\ E \in R^{(P^2 \cdot C) \times D}, E_{pos} \in R^{(N+1) \times D}$$

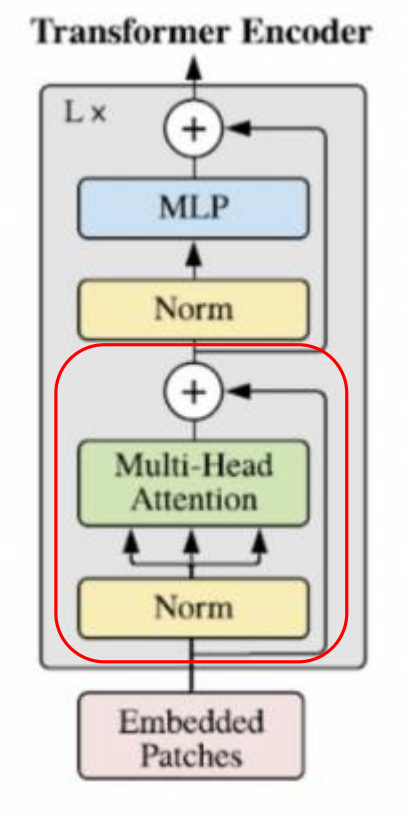
Classification token

Positional Encoding

- Input

4.Method

- Transformer Encoder



Residual connection

Multi-head self Attention

$$z'_\ell = MSA(LN(z_{\ell-1})) + z_{\ell-1}, \ell = 1 \dots L \quad (2)$$

Layer Normalization

- Use residual connection to take the previous input Z_{l-1} and add it to the final output

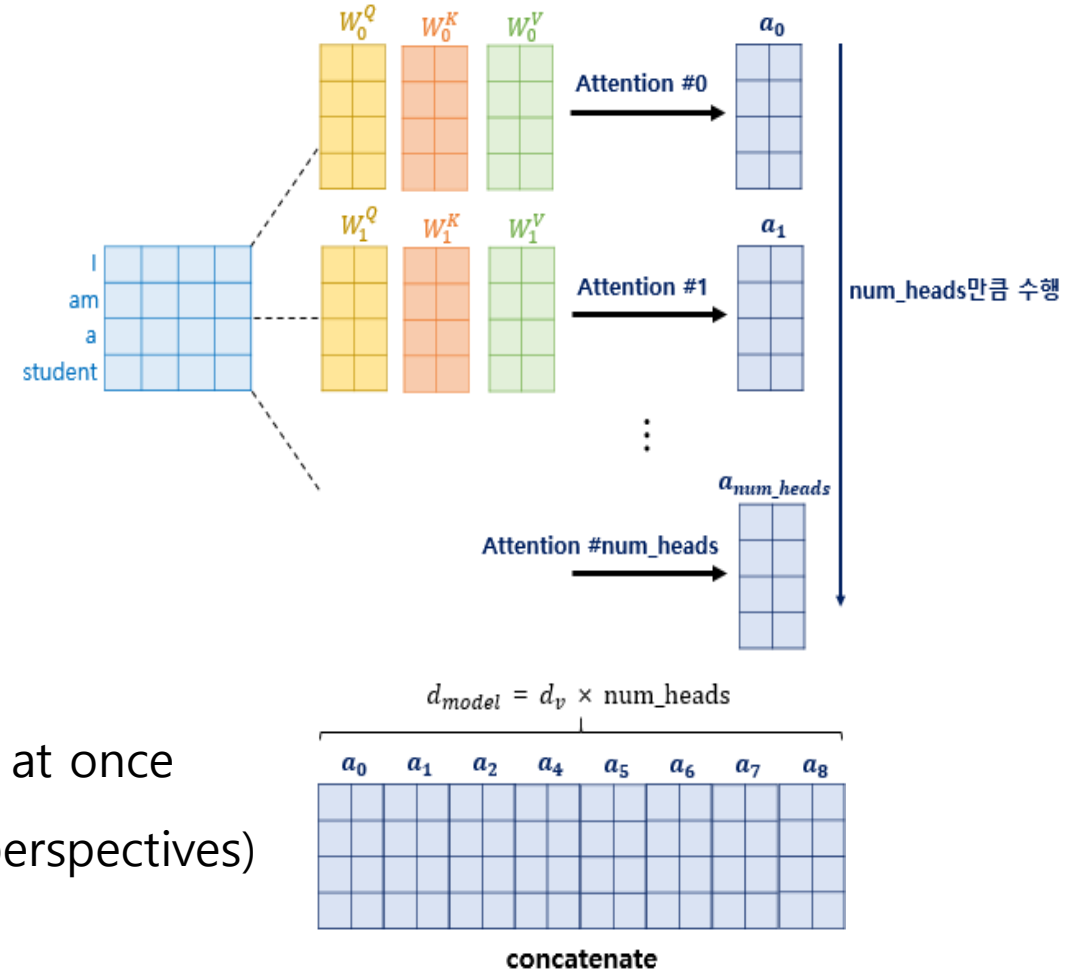
4.Method

▶ Multi-Head Attention

- Calculated each Attention by the number of heads in parallel

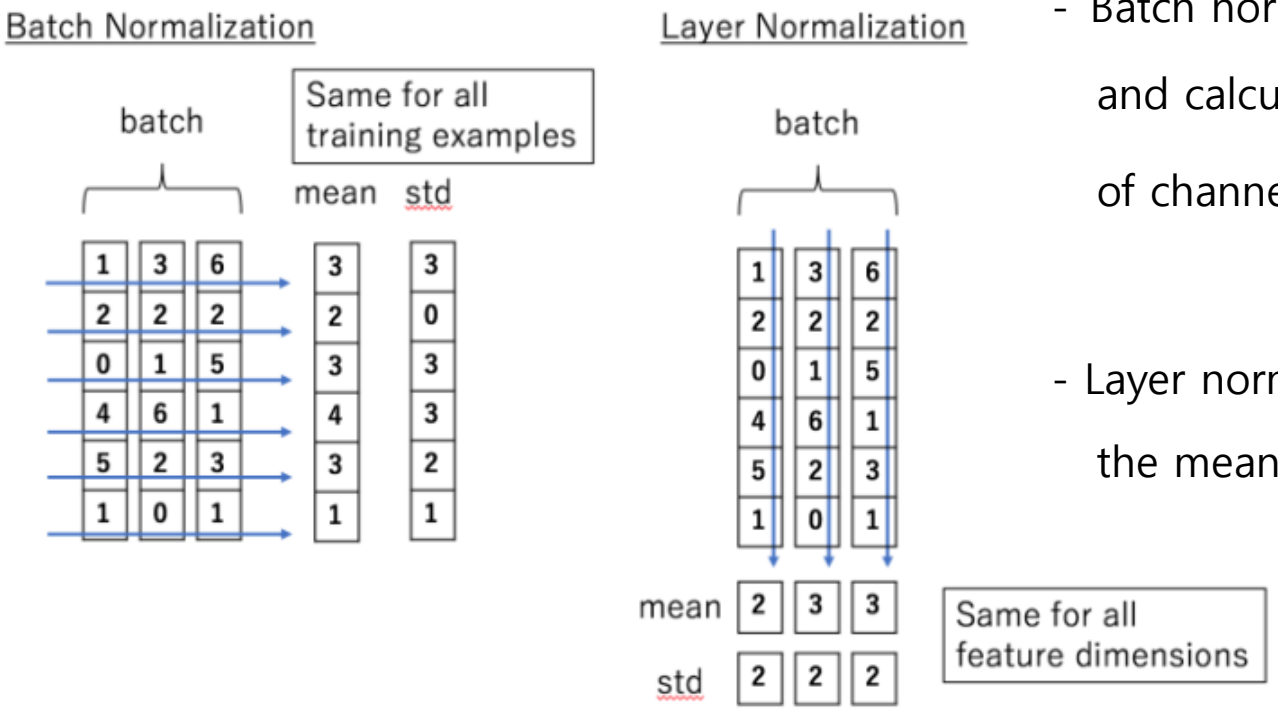
- The derived Attention Values are merged into one through concatenate

- Mechanically better performance than calculation at once (the effect of observing an image from multiple perspectives)



4.Method

▶ Layer Normalization

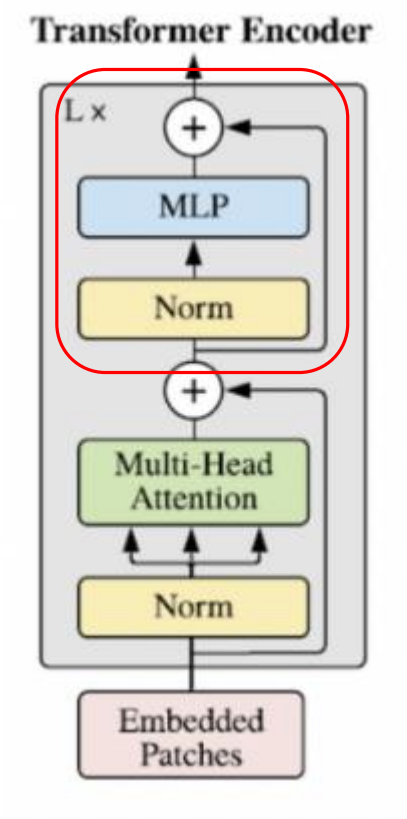


- Batch normalization is normalized for each channel and calculates the mean and variance by the number of channels.

- Layer normalization is normalized by data and calculates the mean and variance by the number of minibatch

4.Method

- Transformer Encoder



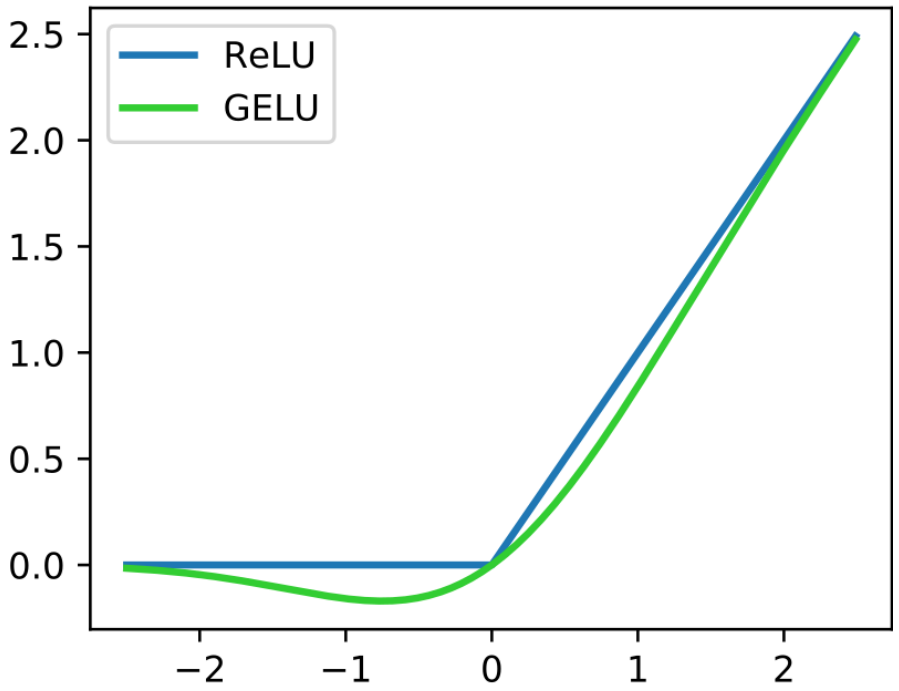
(3) $z_\ell = MLP(LN(z'_\ell)) + z'_\ell, \ell = 1 \ldots L$

Multi layer Perceptron

- In MLP, use 2 FC layer and GELU activation

4.Method

- GELU (Gaussian Error Linear Unit)



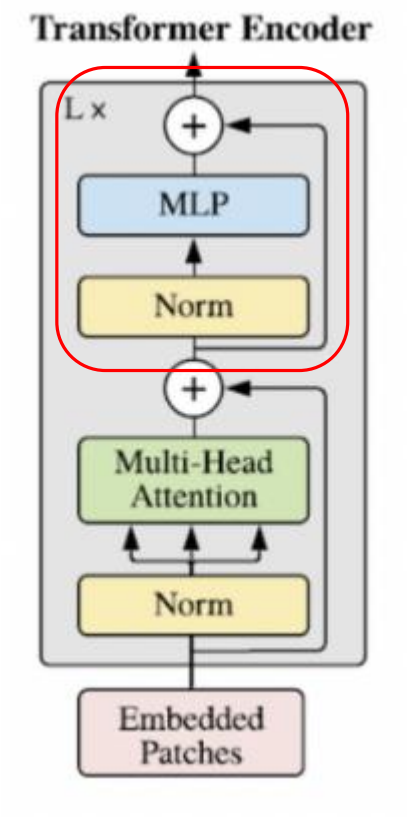
$$0.5x(1 + tanh[\sqrt{2/\pi}(x + 0.044715x^3)])$$

- The most powerful function used in the most recent paper at the time

- The advantage is that probabilistic interpretation is possible because the value is adjusted as a ratio of how large the input value x is compared to other inputs.

4.Method

- Transformer Encoder



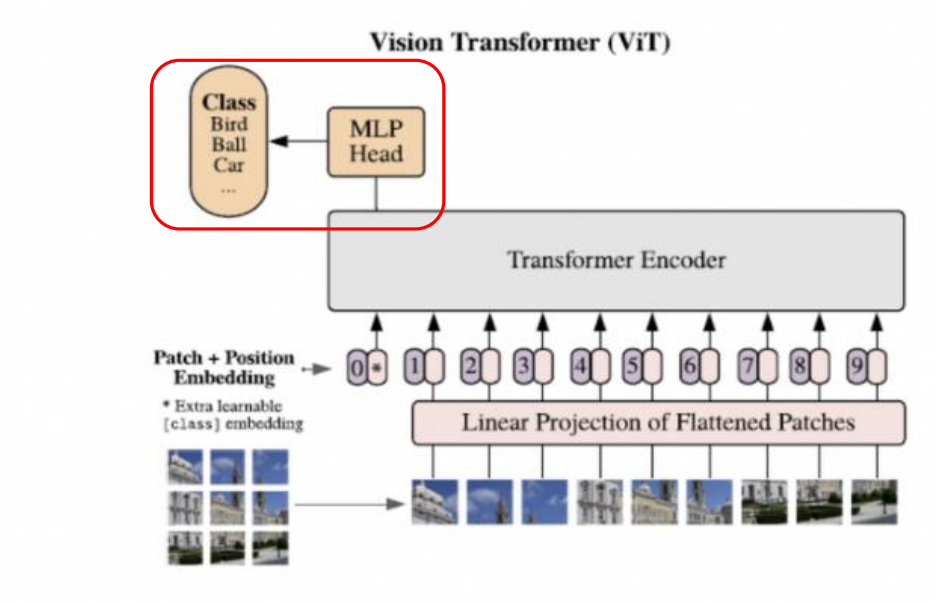
(4) $y = LN(z_L^0)$

Layer Normalization

- After repeating Transformer L times, the output is transmitted to the next step..

4.Method

- Classification

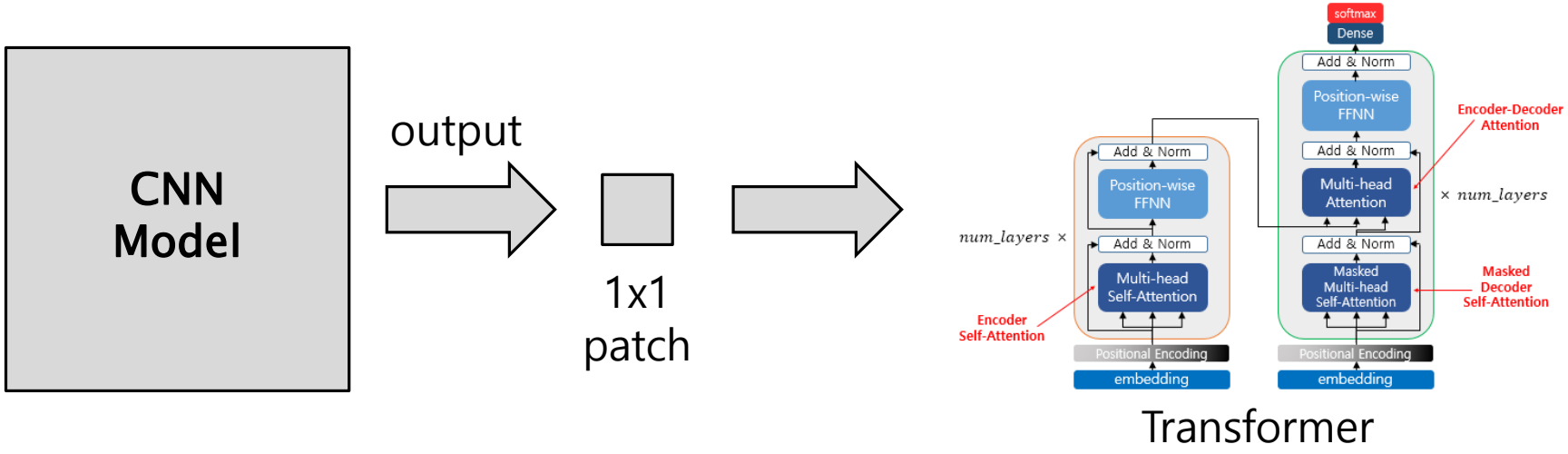


- In the output of the Transformer, only the class token is used for the classification problem and classifies the class using an additional MLP at the end

4.Method

▶ Hybrid Architecture

- Not only raw image patch but also feature map on CNN is available as input



4.Method

- Fine-tuning and higher-resolution

- When fine tuning at the classification layer, it is effective to use a higher resolution image than pre-train image

- However, in this case, the pre-trained position embedding will be lost, so the value will be filled through 2d interpolation

- That is, it is to manually add an image-specific inductive bias.

5.Results

- Vit Model & Classification

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

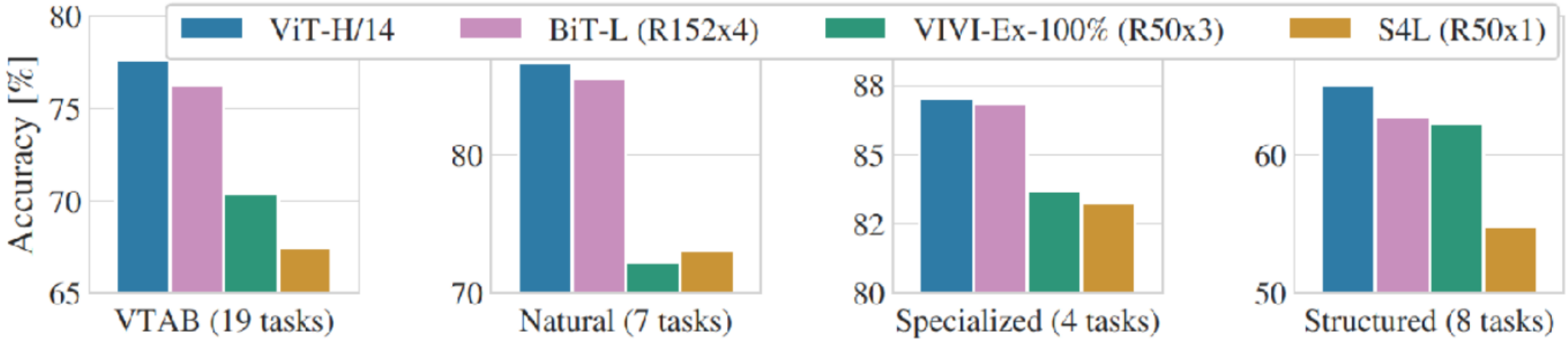
Base, large's backbone : BERT
Huge's backbone : Ours

	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

Vit-*/ 16 = use 16*16 patch size

5.Results

- Comparison based on data characteristics



5.Results

- Pre-Training Data Requirements

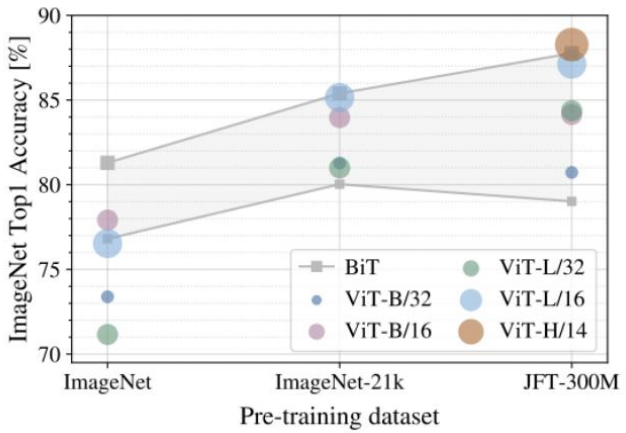


Figure 3: Transfer to ImageNet. While large ViT models perform worse than BiT ResNets (shaded area) when pre-trained on small datasets, they shine when pre-trained on larger datasets. Similarly, larger ViT variants overtake smaller ones as the dataset grows.

- Bit's accuracy is ahead when pre-trained with a small dataset, but as the pre-training dataset bigger, ViT wins BiT performance

5.Results

- Pre-Training Data Requirements

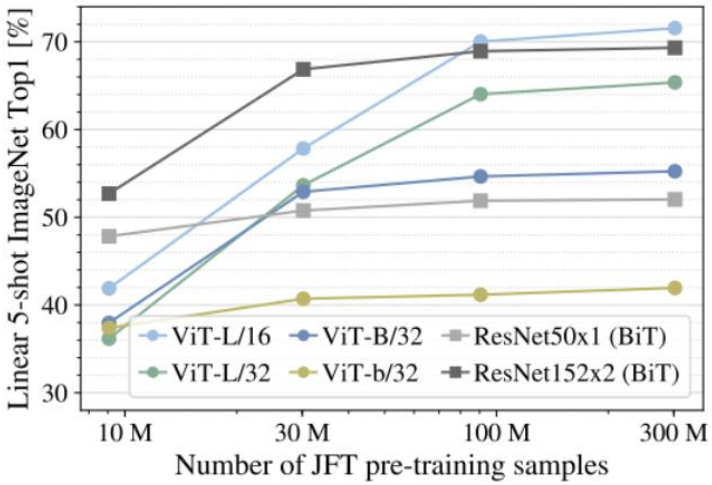


Figure 4: Linear few-shot evaluation on ImageNet versus pre-training size. ResNets perform better with smaller pre-training datasets but plateau sooner than ViT, which performs better with larger pre-training. ViT-b is ViT-B with all hidden dimensions halved.

- ViT has been overfitted faster than BiT for these small datasets

- Inductive biases are useful for relatively small datasets

- But for a large dataset, just learning the right pattern is enough or even more beneficial than having an inductive bias

5.Results

- Scaling Study (evaluation cost)

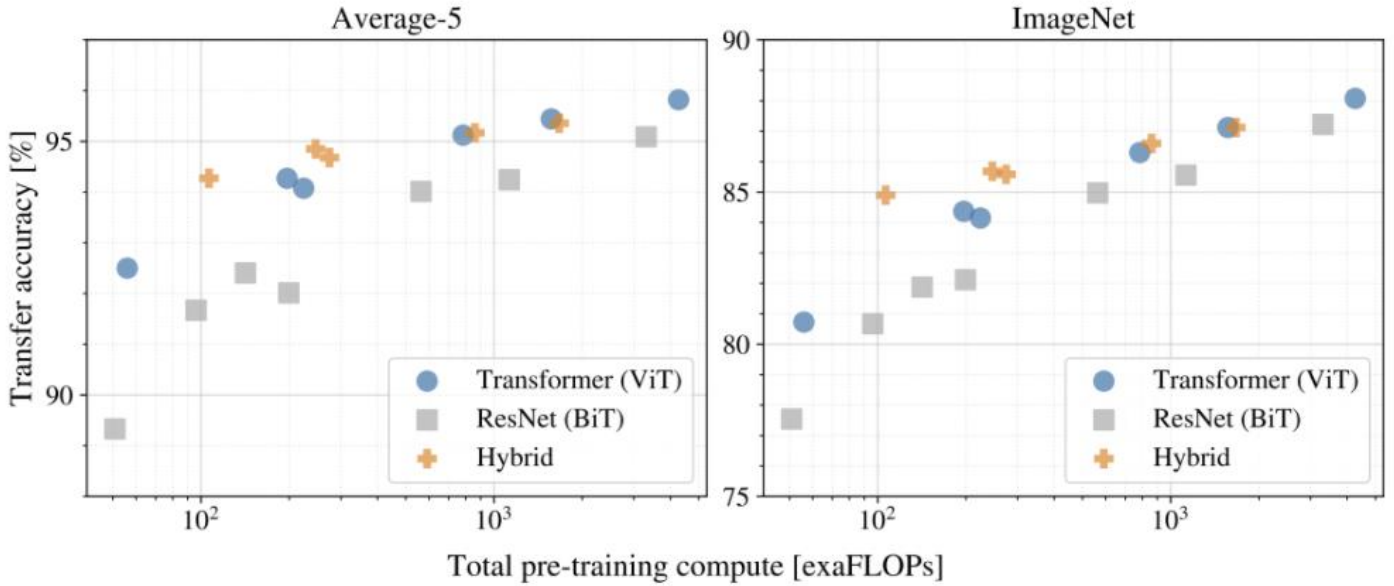


Figure 5: Performance versus pre-training compute for different architectures: Vision Transformers, ResNets, and hybrids. Vision Transformers generally outperform ResNets with the same computational budget. Hybrids improve upon pure Transformers for smaller model sizes, but the gap vanishes for larger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

5.Results

- Scaling Study (evaluation cost)

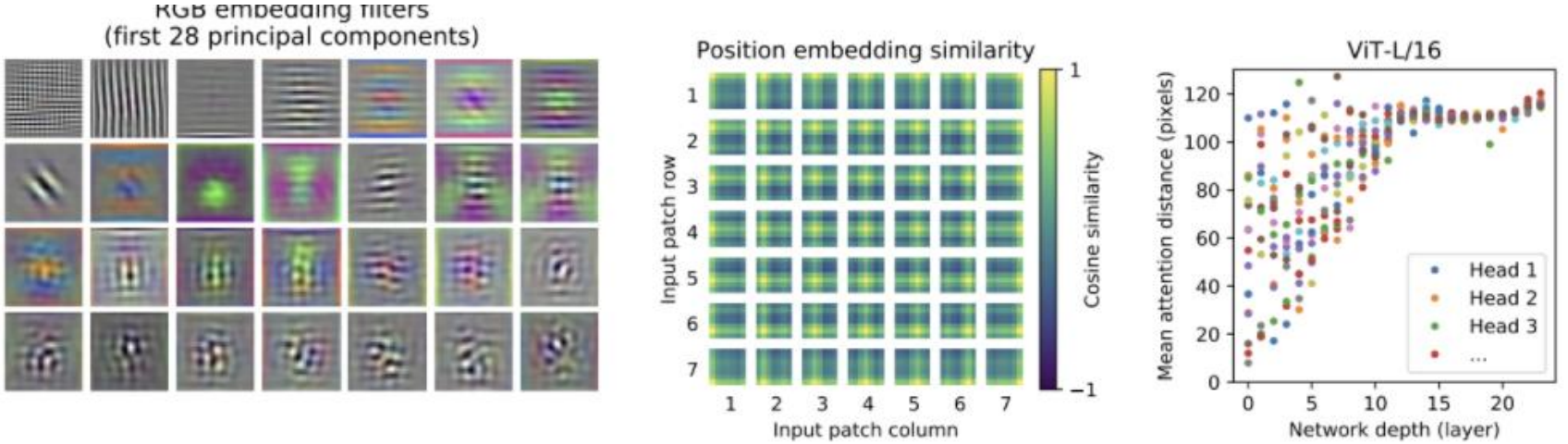


Figure 7: **Left:** Filters of the initial linear embedding of RGB values of ViT-L/32. **Center:** Similarity of position embeddings of ViT-L/32. Tiles show the cosine similarity between the position embedding of the patch with the indicated row and column and the position embeddings of all other patches.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

6.Conclusion

1. Applying Transformer Directly to computer vision

2. Interpret the image as a patch sequence and process it through a standard transformer encoder, as in NLP

3. Vision Transformer matches or exceeds the state of the art on many image classification datasets, whilst being relatively cheap to pre-train