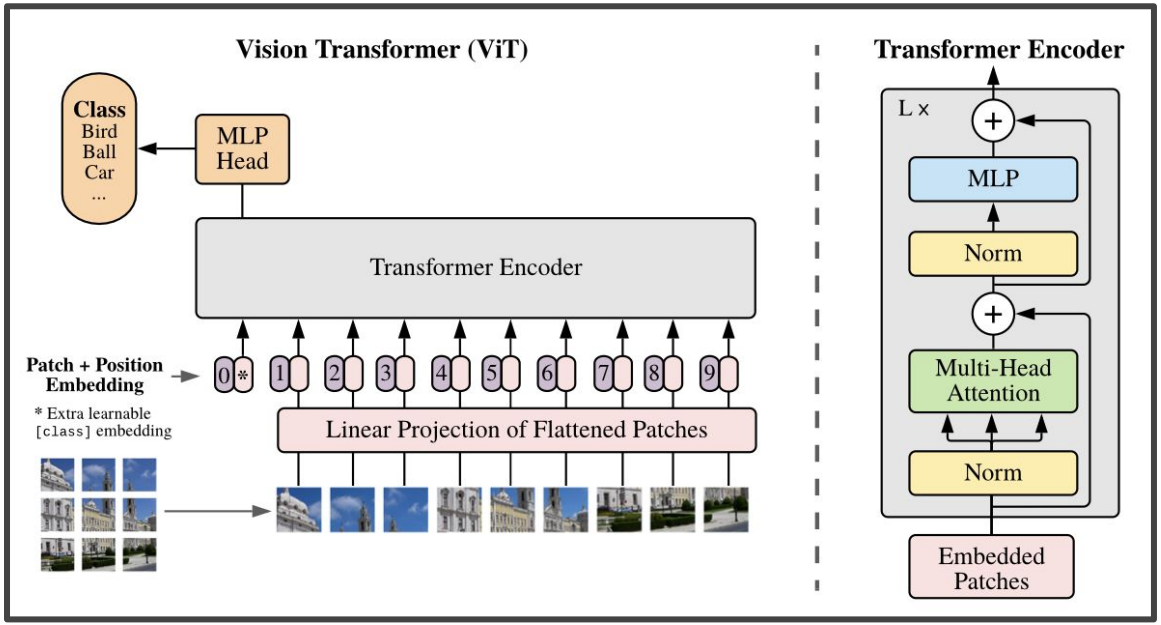
Vision Transformers

An image is worth 16x16 words: Transformers for image recognition at scale, Dosovitskiy et al.,



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]

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**[*,†]

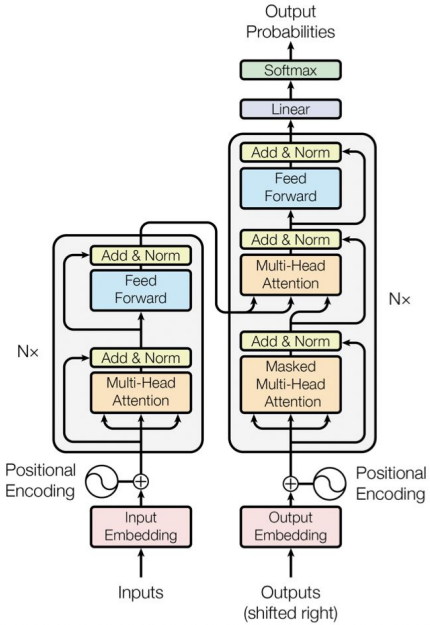
[*]equal technical contribution, [†]equal advising

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]



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]

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]

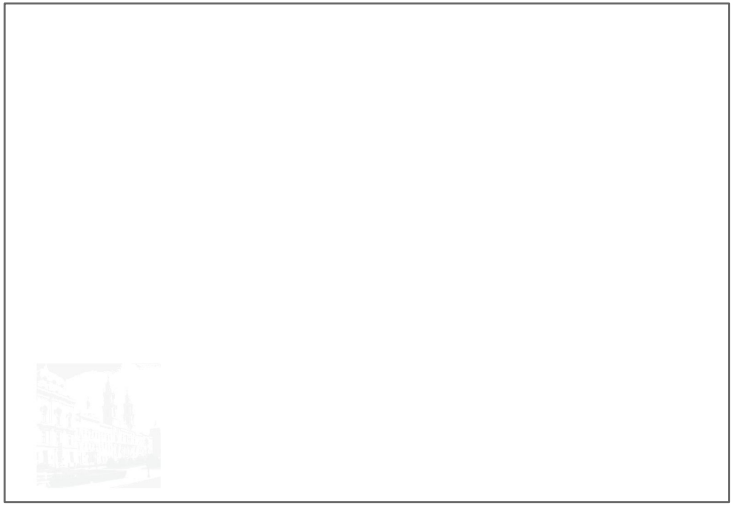


Image: Google AI blog

Outline of the video

In this video, we will see:

- Motivation for ViT
- Vision Transformer architecture
- Fine-tuning ViT
- Training ViT
- ViT vs ResNet
- Inspecting ViT representations
- Self-supervision
- Vision Transformers vs CNNs
- ViT Implementations
- ViTs follow-up works: Scaling ViT, Better plain ViT, How to train your ViT, Scaling ViT to 22B Parameters

Outline of the video

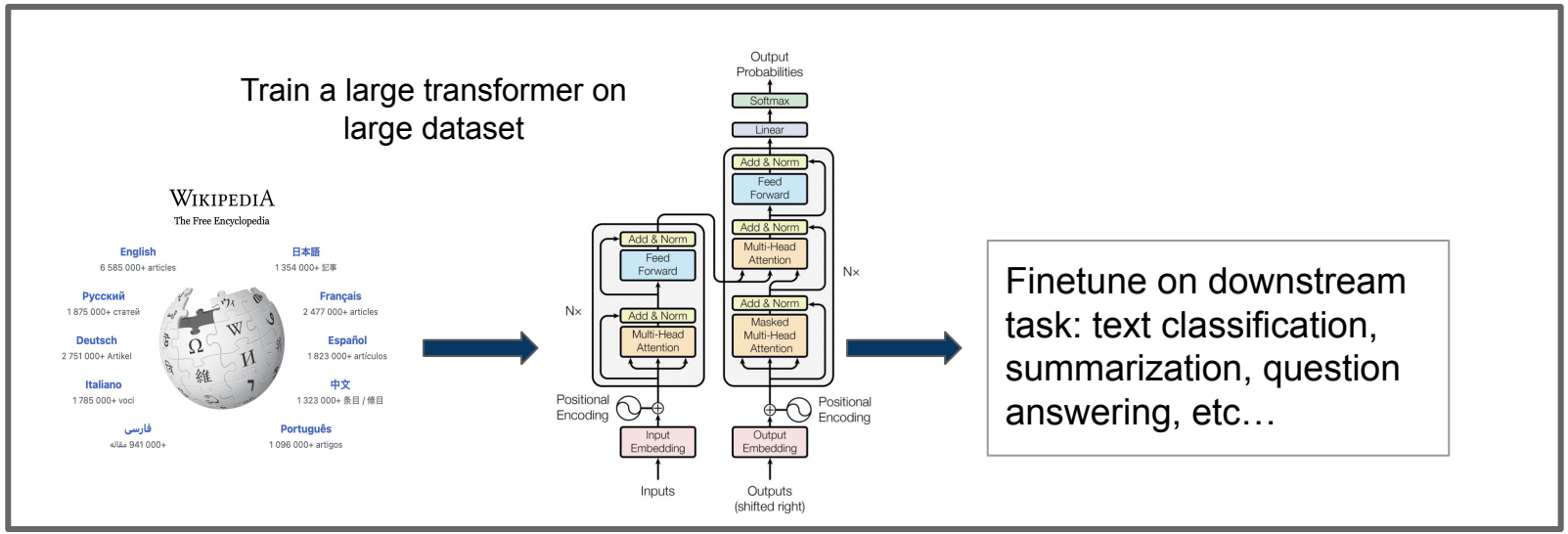
In this video, we will see:

- Motivation for ViT
- Vision Transformer architecture
- Fine-tuning ViT
- Training ViT
- ViT vs ResNet
- Inspecting ViT representations
- Self-supervision
- Vision Transformers vs CNNs
- ViT Implementations
- ViTs follow-up works: Scaling ViT, Better plain ViT, How to train your ViT, Scaling ViT to 22B Parameters

Unless mentioned, all images and graphs used in this video are taken from the paper ***An image is worth 16x16 words: Transformers for image recognition at scale, Dosovitskiy et al.***.

Motivation for Vision Transformer

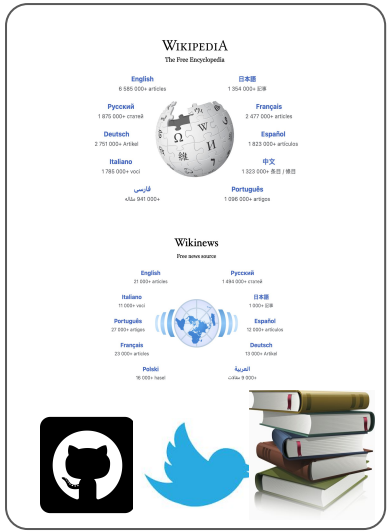
Due to Transformer architecture that has extreme scalability and efficiency, pre-training in NLP has been successful. The theme is always pre-training a large Transformer model on large dataset(wikipedia, news, books, codes) and fine-tuning on downstream tasks.



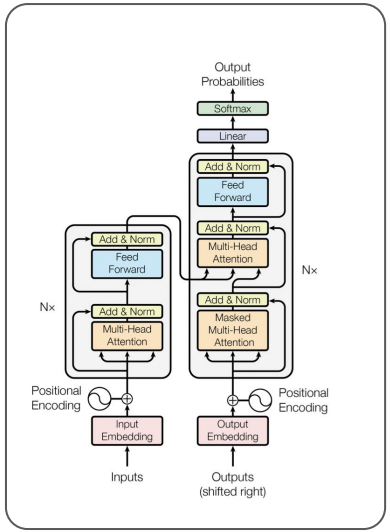
Motivation for Vision Transformer

NLP has been successful due to 3 main ingredients that became happened to be available at the right time: large datasets, powerful and efficient models(Transformer), compute.

Large datasets



Efficient models

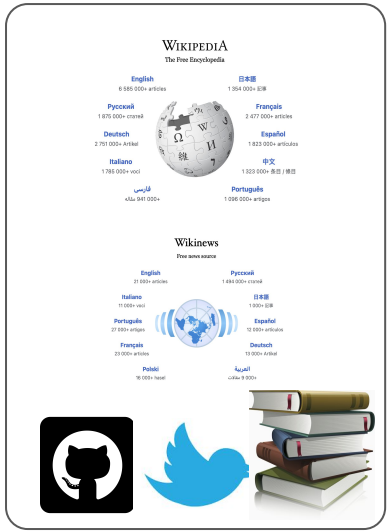


Compute

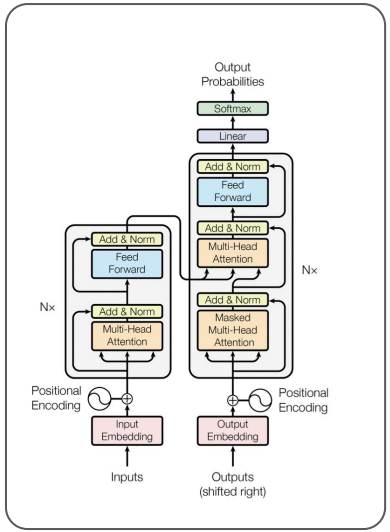
Motivation for Vision Transformer

NLP has been successful due to 3 main ingredients that became happened to be available at the right time: large datasets, powerful and efficient models(Transformer), compute.

Large datasets



Efficient models



Compute

Transformer seems to be the engine of all. Can we apply the same Transformer to image recognition task?

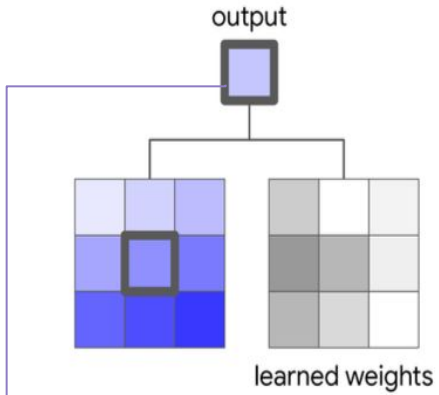
Motivation for Vision Transformer

Before the emergence of Vision Transformer architecture, most works attempted to apply Transformer to computer vision by either:

- Replacing all convolution operations with local attention
- Combining convolution and attention
- Applying Transformer to image pixels directly

Transformers in Vision: Replacing all Convolution Operations with Local Self-attention Layer

Local convolution

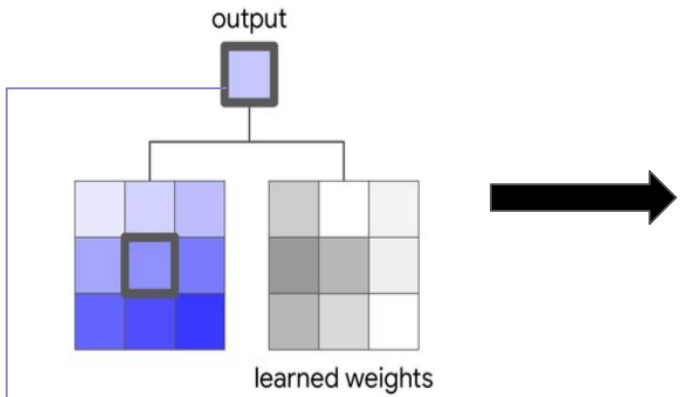


Output: Inner product of convolutional kernel and local window/receptive field.

Ramachandran et al., "Stand-Alone Self-Attention in Vision Models", 2018

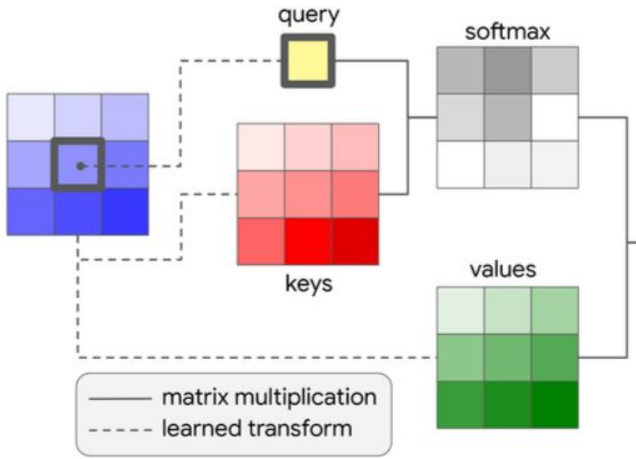
Transformers in Vision: Replacing all Convolution Operations with Local Self-attention Layer

Local convolution



Output: Inner product of convolutional kernel and local window/receptive field.

Local self-attention layer



Query: center pixel of local window.

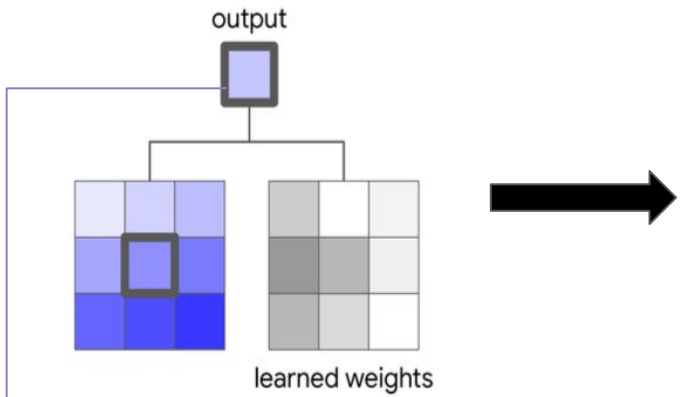
Keys & **values**: All other pixels of local window.

Output: Standard self-attention applied to receptive field/local window.

Ramachandran et al., "Stand-Alone Self-Attention in Vision Models", 2018

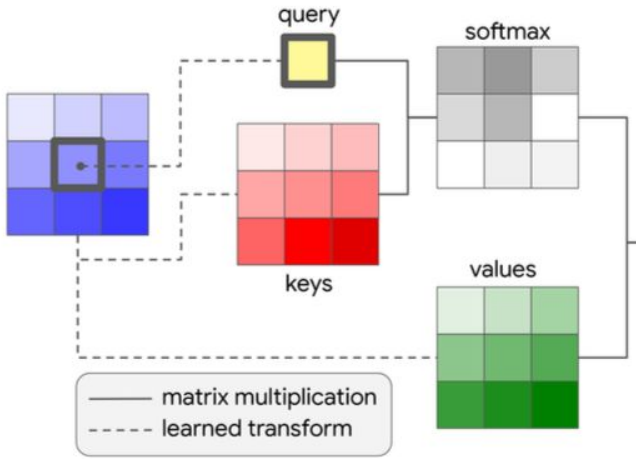
Transformers in Vision: Replacing all Convolution Operations with Local Self-attention Layer

Local convolution



Output: Inner product of convolutional kernel and local window/receptive field.

Local self-attention layer



Query: center pixel of local window.

Keys & **values**: All other pixels of local window.

Output: Standard self-attention applied to receptive field/local window.

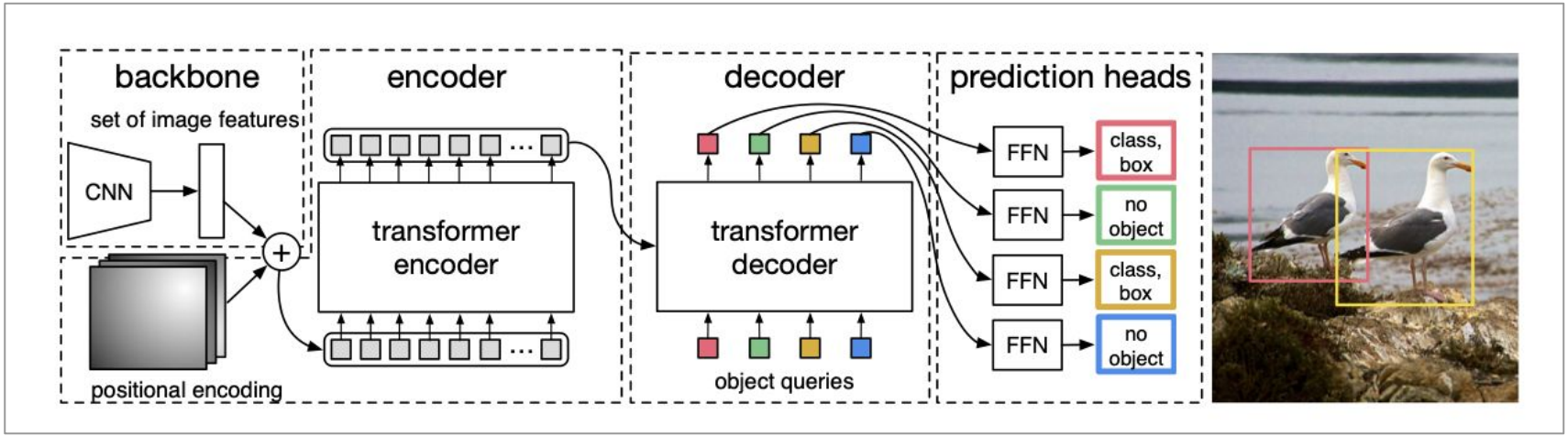
!!

Local self-attention is not scalable and hard to implement. Also not compute efficient although attention is computed in local windows.

Ramachandran et al., "Stand-Alone Self-Attention in Vision Models", 2018

Transformers in Vision: Combining Convolution with Attention

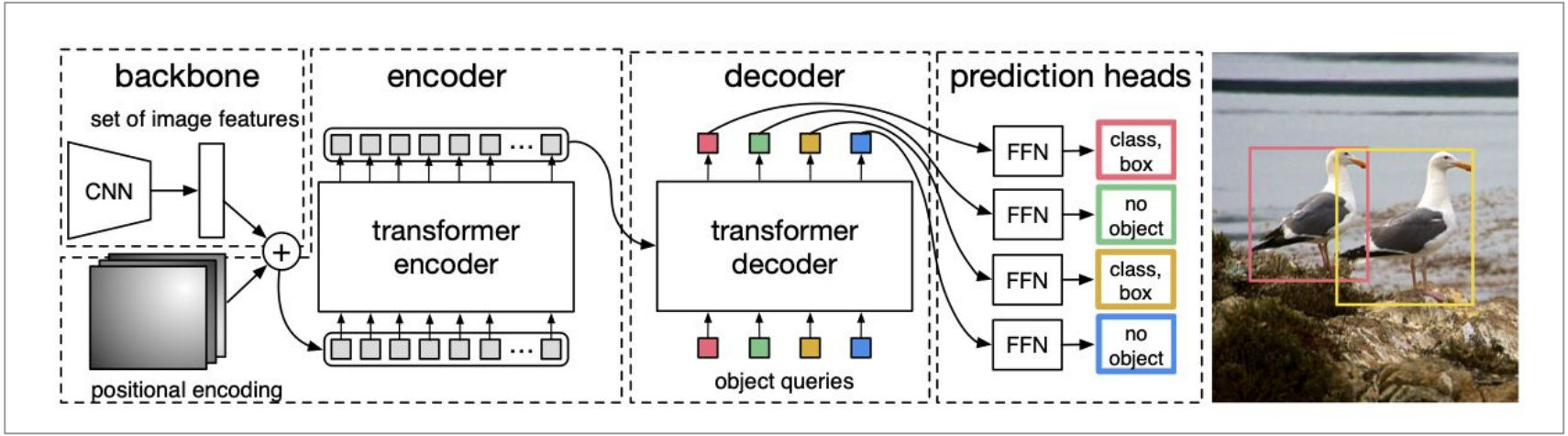
As demonstrated by DETR, combining convolution and attention can result in SOTA in object detection. DETR learns 2D visual features of image with convolutional backbone(ex: ResNet), feed the features to Transformer and predict the box and object class.



Carion et al., "End-to-end object detection with Transformers", 2020

Transformers in Vision: Combining Convolution with Attention

As demonstrated by DETR, combining convolution and attention can result in SOTA in object detection. DETR learns 2D visual features of image with convolutional backbone(ex: ResNet), feed the features to Transformer and predict the box and object class.

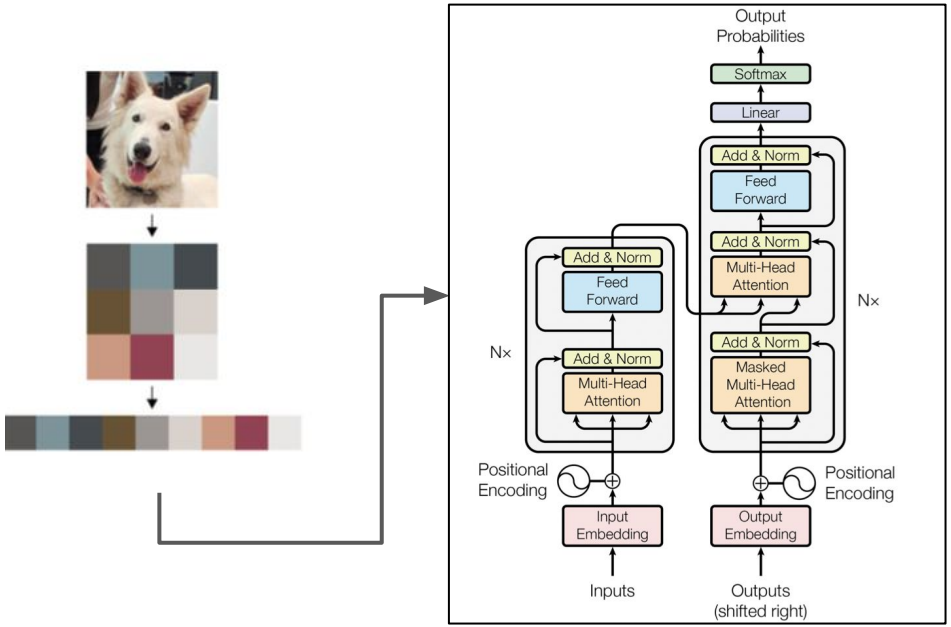


CNNs + Attention seems to work great when done right! DETR is indeed one of the remarkable papers in vision.

Carion et al., "End-to-end object detection with Transformers", 2020

Transformers in Vision: Applying Transformer to Pixels Directly

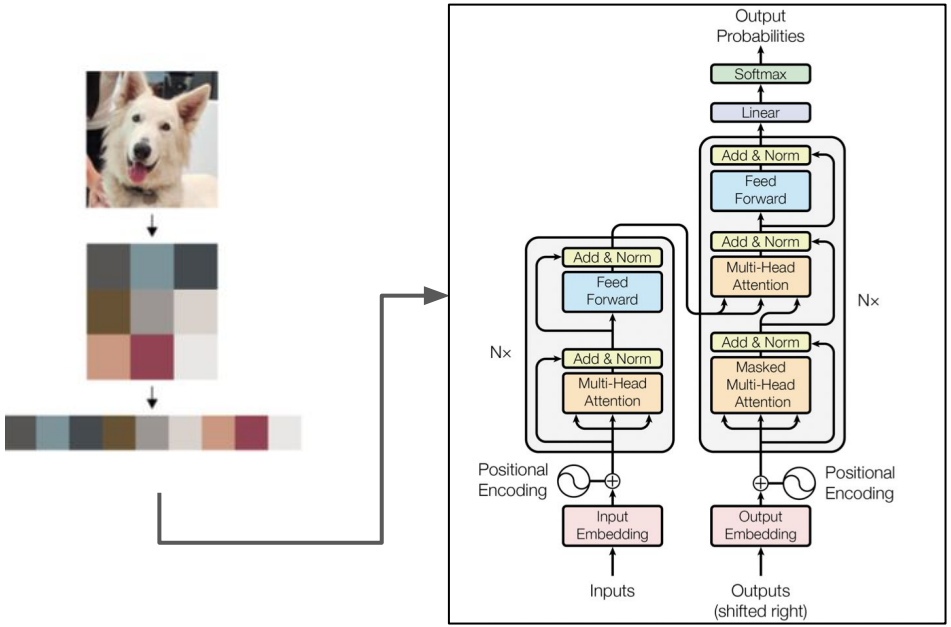
Treat a raw image as a sequence of pixels just as sentence is a sequence of words in NLP. Resize to low resolution since attention is computational expensive when applied to the whole image and feed the flattened pixels to a normal Transformer to predict the next pixel autoregressively or with masked pixel prediction(like in BERT).



Chen et al., "Generative Pretraining from Pixels", 2020

Transformers in Vision: Applying Transformer to Pixels Directly

Treat a raw image as a sequence of pixels just as sentence is a sequence of words in NLP. Resize to low resolution since attention is computational expensive when applied to the whole image and feed the flattened pixels to a normal Transformer to predict the next pixel autoregressively or with masked pixel prediction(like in BERT).



!!

Very computationally expensive since each pixel in input image attends to each other!!

Chen et al., "Generative Pretraining from Pixels", 2020

The main goal of ViT authors were to transfer NLP pretraining success to computer vision and to remove the need of convolution. Given the previous works, the biggest breakthrough in ViT was not applying Transformer to images. It was dividing images into sequence of patches and feeding those patches to standard Transformer encoder.

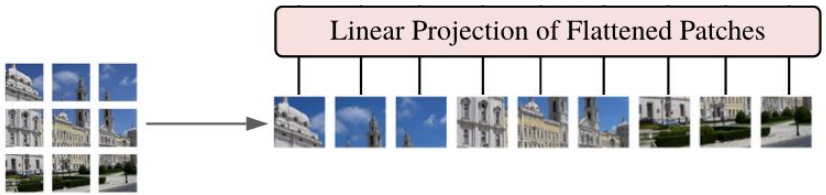
Vision Transformer Architecture

Divide an input image into non-overlapping patches (16x16x3 each) and flatten them into a sequence of 1D patches

Number of patches N = Effective input sequence length for the Transformer, N = HW/PP, H: width, W: width, P: patch size

Vision Transformer Architecture

Linearly project patches to D(dimension of the model)



Divide an input image into non-overlapping patches (16x16x3 each) and flatten them into a sequence of 1D patches

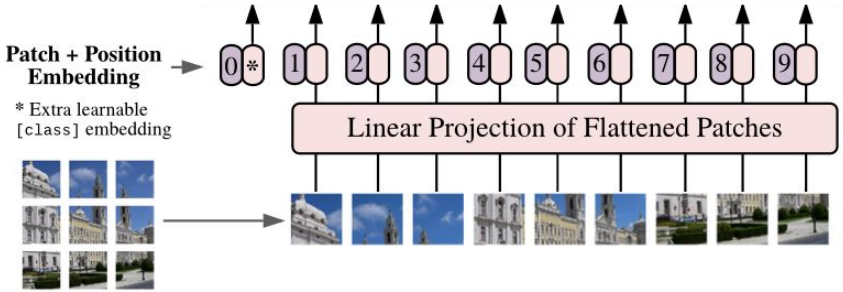
Number of patches N = Effective input sequence length for the Transformer, N = HW/PP, H: width, W: width, P: patch size

Vision Transformer Architecture

Apply positional embeddings to patches

Linearly project patches to D(dimension of the model)

Divide an input image into non-overlapping patches (16x16x3 each) and flatten them into a sequence of 1D patches



Number of patches N = Effective input sequence length for the Transformer, N = HW/PP, H: width, W: width, P: patch size

Vision Transformer Architecture

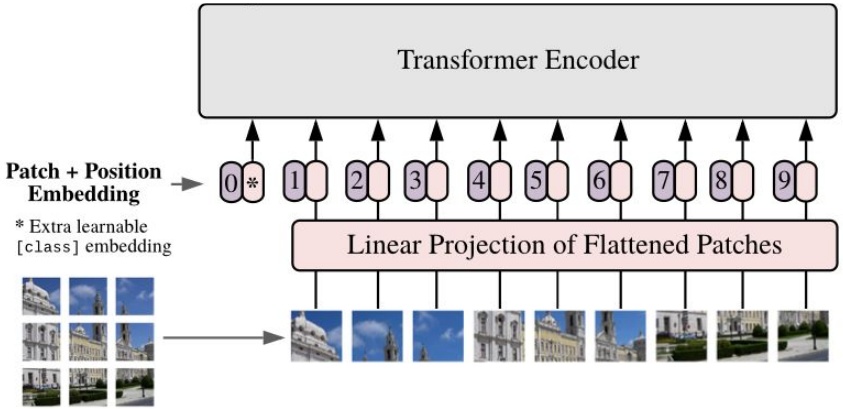
Transformer encoder in ViT looks like NLP except layernorm that comes before attention & MLPs

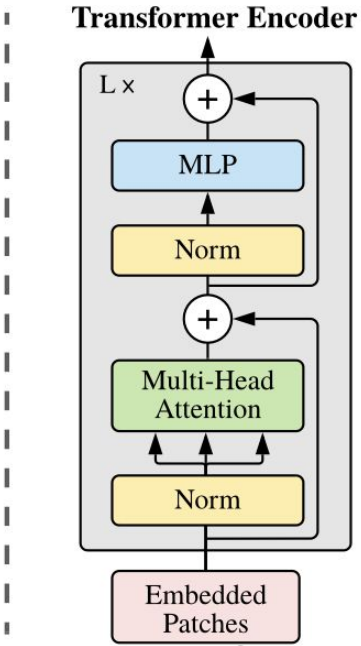
Feed the patches to a normal Transformer encoder.

Apply positional embeddings to patches

Linearly project patches to D(dimension of the model)

Divide an input image into non-overlapping patches (16x16x3 each) and flatten them into a sequence of 1D patches





Number of patches N = Effective input sequence length for the Transformer, N = HW/PP, H: width, W: width, P: patch size

Vision Transformer Architecture

Transformer encoder in ViT looks like NLP except layernorm that comes before attention & MLPs

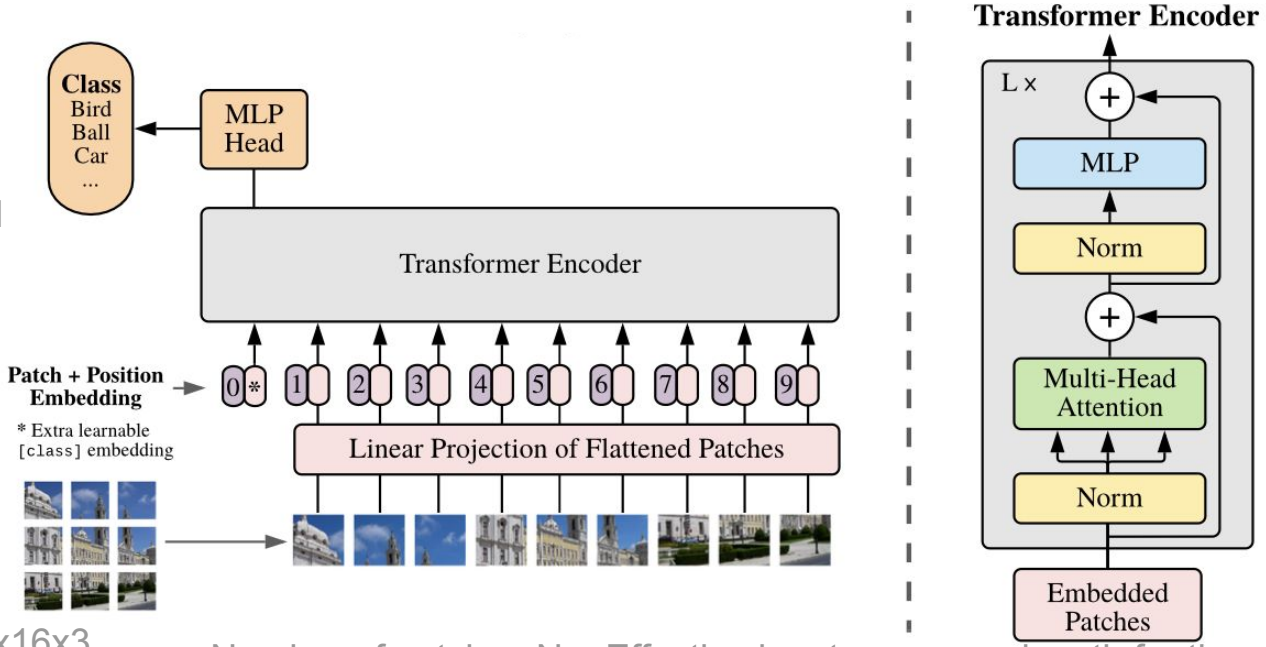
Add a classification head to perform image classification.

Feed the patches to a normal Transformer encoder.

Apply positional embeddings to patches

Linearly project patches to D(dimension of the model)

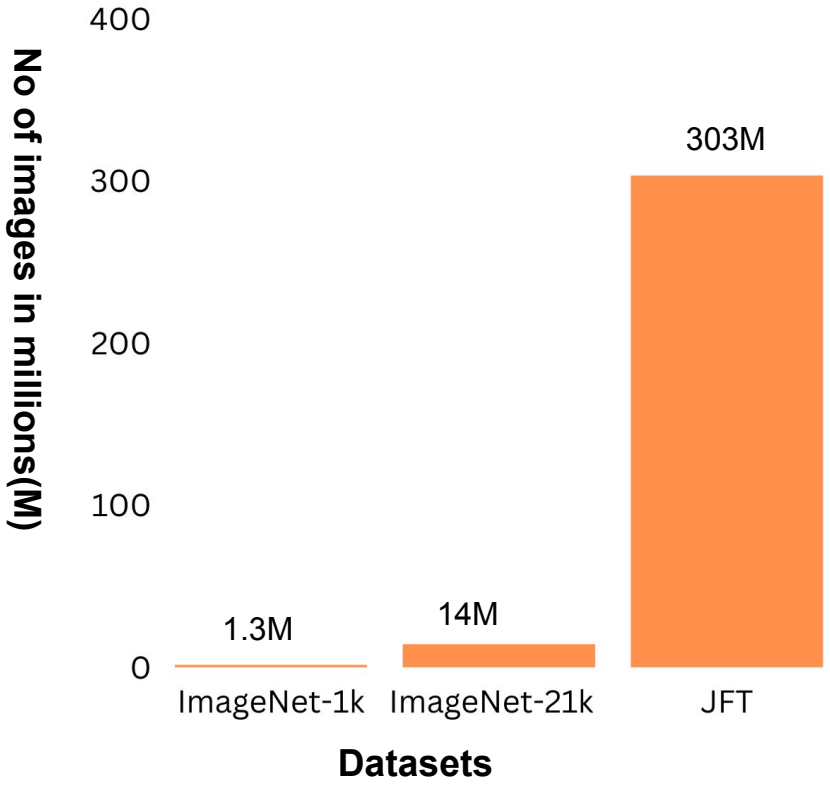
Divide an input image into non-overlapping patches (16x16x3 each) and flatten them into a sequence of 1D patches



Number of patches N = Effective input sequence length for the Transformer, N = HW/PP, H: width, W: width, P: patch size

Training Vision Transformer

Vision Transformer(ViT) was pre-trained on 3 datasets of varying size and scale.



Pre-training datasets

- ImageNet-1K: 1.3M images, 1K classes
- ImageNet-21k: 14M images, 21K classes
- JFT: 303M images, 18K classes

ImageNet-1k and ImageNet-21K are also used for fine-tuning!

Training Vision Transformer

Vision Transformer(ViT) was pre-trained with same configurations of as BERT.

Model	Layers	Hidden size	MLPs size	Heads	Params
ViT-Base	12	768	3072	12	86M
ViT-Large	24	1024	4096	16	307M
ViT-Huge	32	1280	5120	16	632M

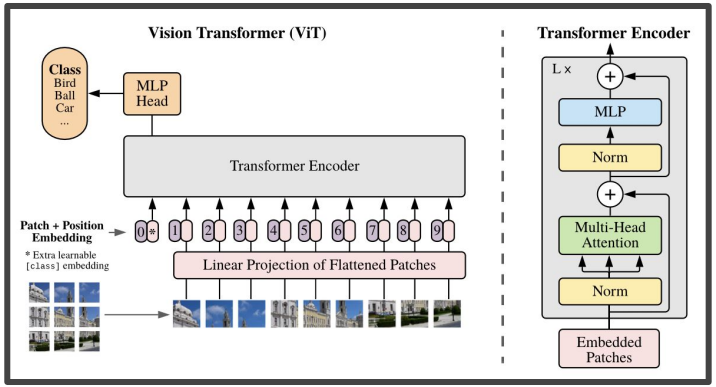
Layers: number of Transformer encoder layers

Hidden size: the dimension of the model

MLPs size: the number of neurons in MLP layers

Heads: number of attention layers in Multi-Head Attention(MHA)

Params: Total parameters



Training Vision Transformer

Vision Transformer(ViT) was pre-trained with same configurations of as BERT.

Model	Layers	Hidden size	MLPs size	Heads	Params
ViT-Base	12	768	3072	12	86M
ViT-Large	24	1024	4096	16	307M
ViT-Huge	32	1280	5120	16	632M

ViT is isotropic architecture - same resolution is maintained across the whole network.

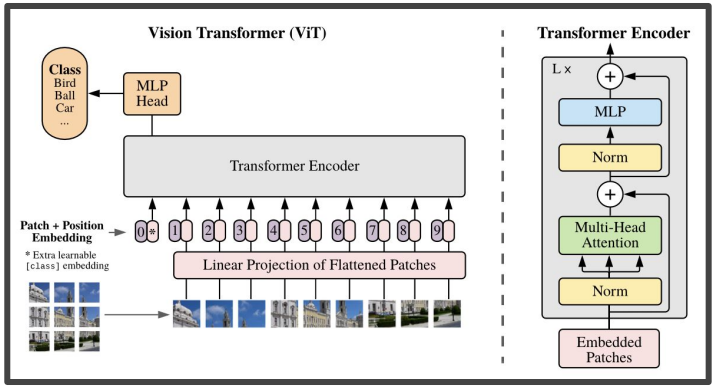
Layers: number of Transformer encoder layers

Hidden size: the dimension of the model

MLPs size: the number of neurons in MLP layers

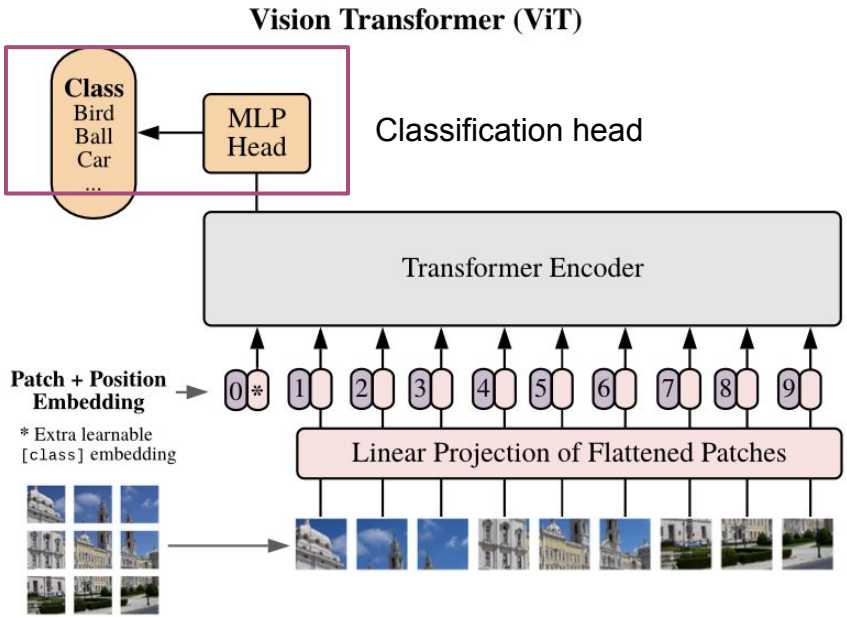
Heads: number of attention layers in Multi-Head Attention(MHA)

Params: Total parameters



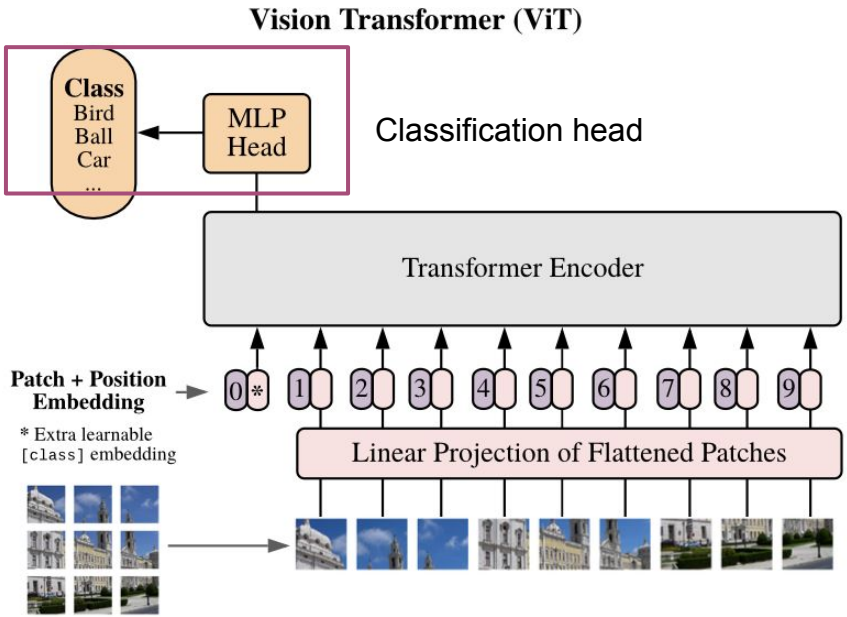
Fine-tuning Vision Transformer

ViT is pre-trained on large datasets and fine-tuned on small downstream datasets. Same as BERT, during fine-tuning, the pre-trained classification head is removed and replaced with zero-initialized **D**x**K**(**D** is hidden size, **K** is number of downstream classes) classification head.



Fine-tuning Vision Transformer

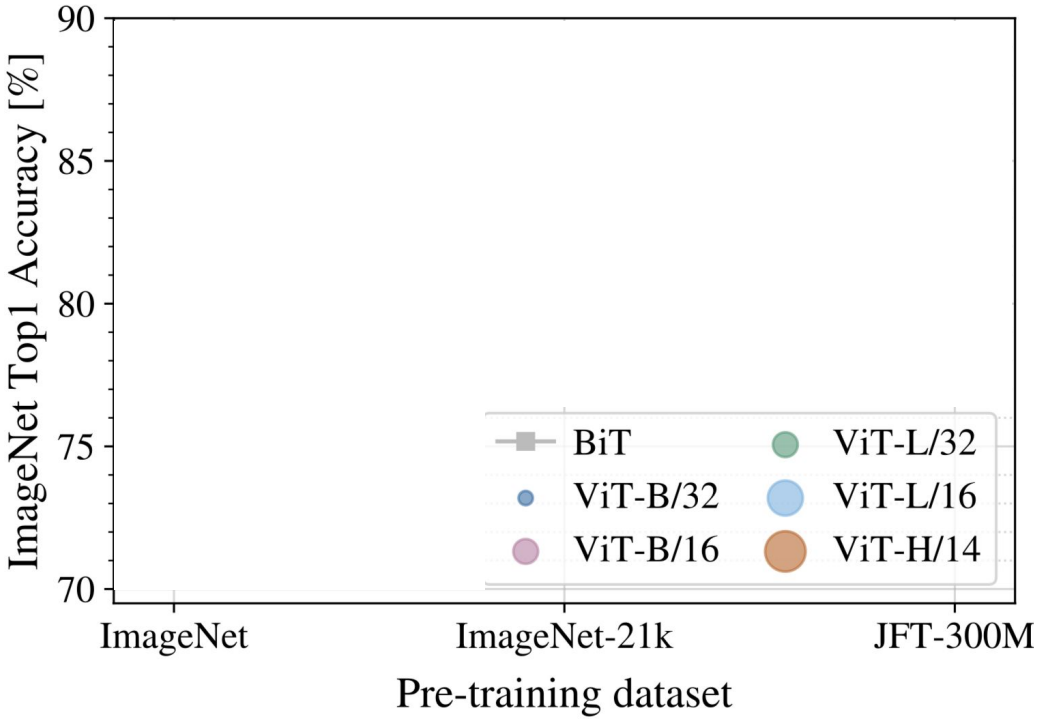
ViT is pre-trained on large datasets and fine-tuned on small downstream datasets. Same as BERT, during fine-tuning, the pre-trained classification head is removed and replaced with zero-initialized **D**x**K**(**D** is hidden size, **K** is number of downstream classes) classification head.



Fine-tuning details

- **Downstream datasets:** ImageNet(1k & 21k), Cifar-10, Cifar-100, Oxford-III pets, Oxford flowers-102, VTAB(combines natural, specialized, and structured tasks).
- Fine-tuning at higher resolutions(**384x384**) yields better results - ViT is trained at **224x224** resolution. At fixed patch size, higher resolutions results in larger input sequence length. >> N = HW/PP.
- You can finetune ViT at any higher resolution since it can handle arbitrary patches length but it may affects pre-trained embeddings.

Vision Transformer vs SOTA CNN(ResNet)



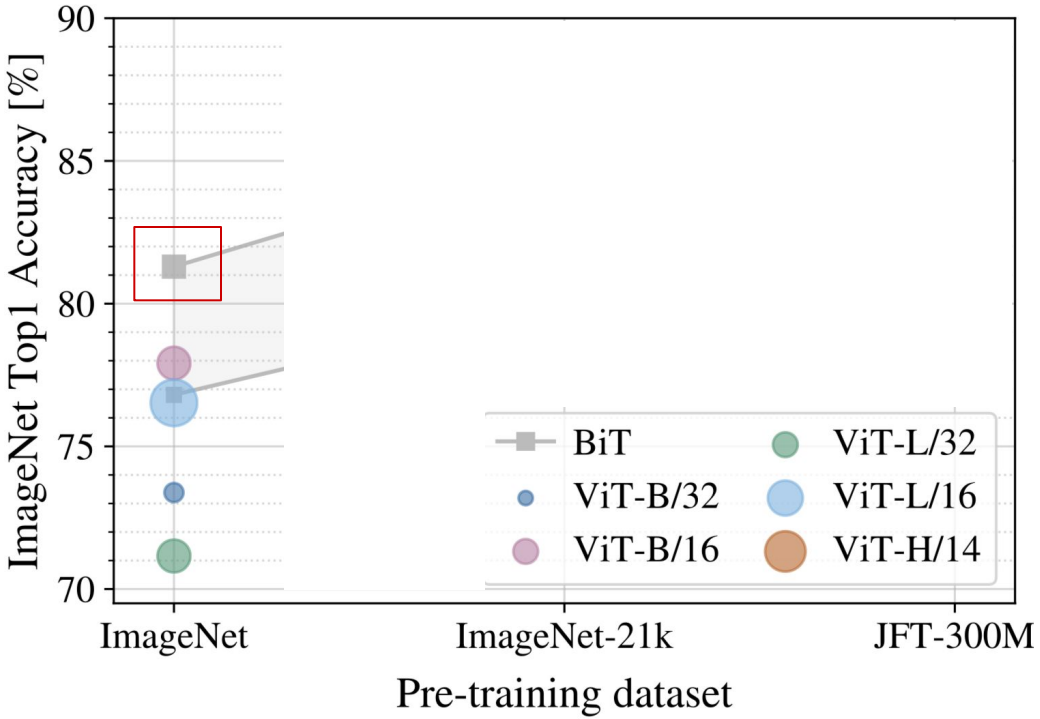
B: Base, L: Large, H: Huge

14, 16, 32: Patch size(the smaller patch size, the more the patches, and the bigger the model) >> **N = HW/PP**

Ex: ViT-B/16: Base ViT with **16x16** patch size

BiT(ResNet): Kolesnikov, "Big Transfer: General Visual Representation Learning", 2020

Vision Transformer vs SOTA CNN(ResNet)



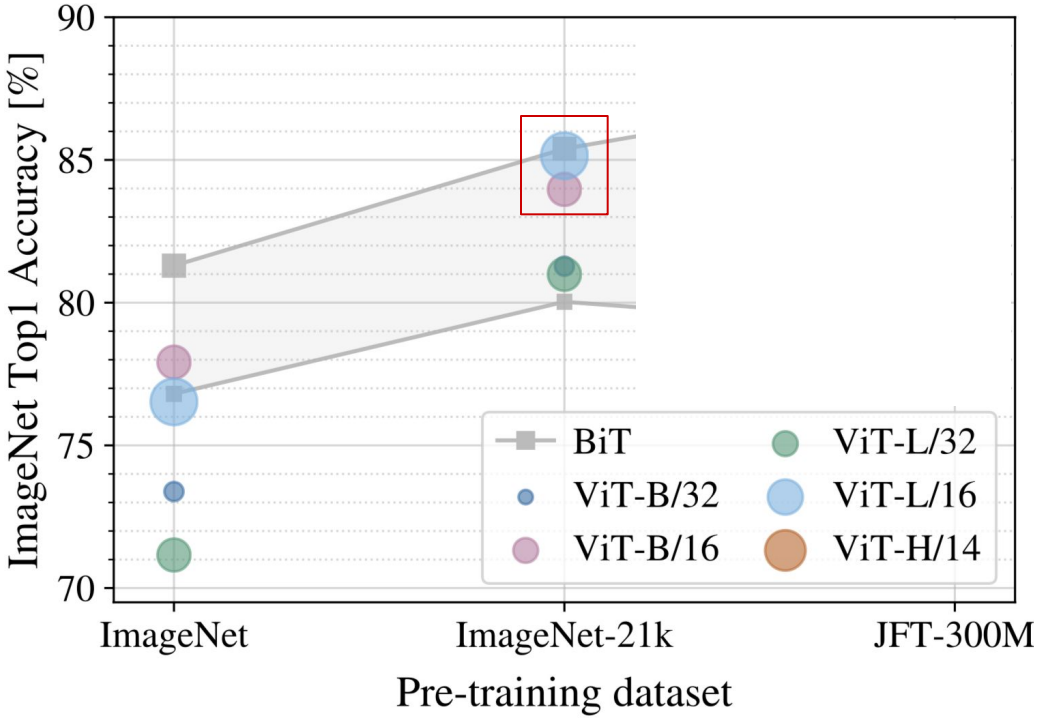
B: Base, L: Large, H: Huge
14, 16, 32: Patch size(the smaller patch size, the more the patches, and the bigger the model) >> **N = HW/PP**
Ex: ViT-B/16: Base ViT with **16x16** patch size

Results

- On small pre-training dataset(ImageNet-1k, 1.3M images), ResNet performs better than ViT due to CNN spatial inductive biases that compensate for small dataset.

BiT(ResNet): Kolesnikov, "Big Transfer: General Visual Representation Learning", 2020

Vision Transformer vs SOTA CNN(ResNet)

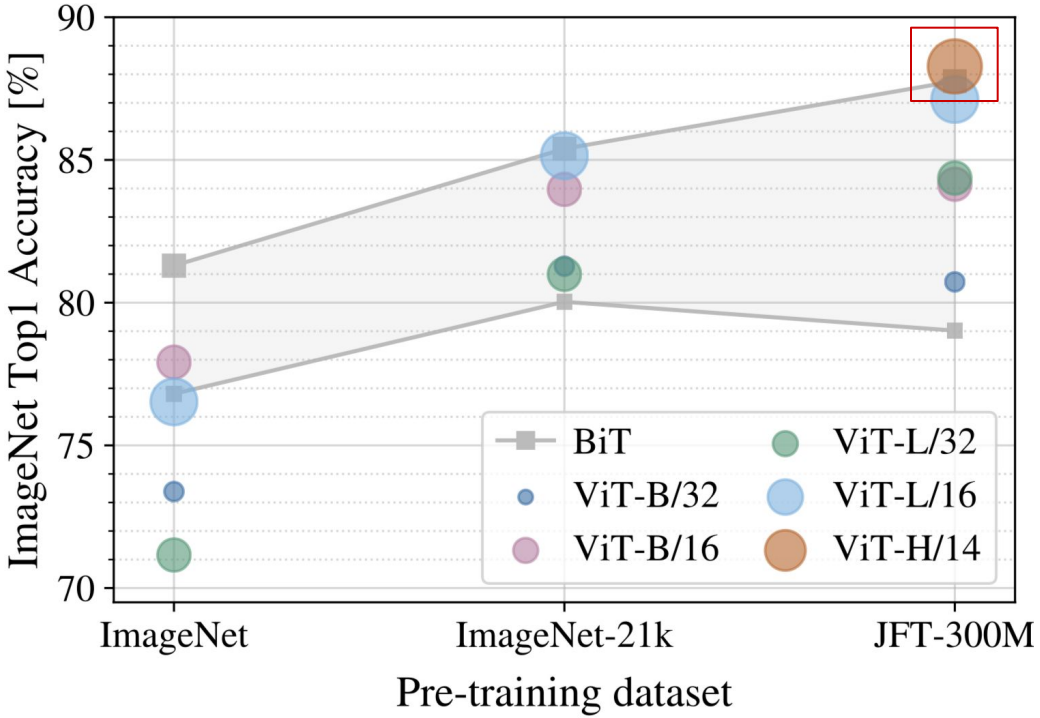


B: Base, L: Large, H: Huge
14, 16, 32: Patch size(the smaller patch size, the more the patches, and the bigger the model) >> **N = HW/PP**
Ex: ViT-B/16: Base ViT with **16x16** patch size

Results

- On small pre-training dataset(ImageNet-1k, 1.3M images), ResNet performs better than ViT due to CNN spatial inductive biases that compensate for small dataset.
- On medium pre-training dataset(Imagenet-21k, 14M images), ViTs and ResNet performance are almost similar although ViTs perform slightly better.

Vision Transformer vs SOTA CNN(ResNet)

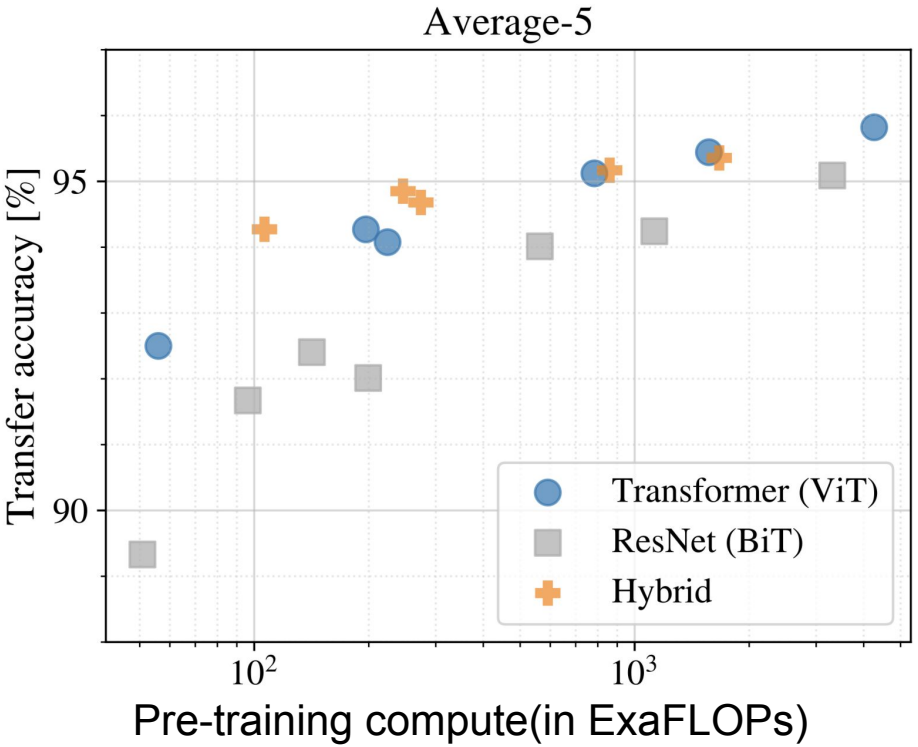


B: Base, L: Large, H: Huge
14, 16, 32: Patch size(the smaller patch size, the more the patches, and the bigger the model) >> **N = HW/PP**
Ex: ViT-B/16: Base ViT with **16x16** patch size

Results

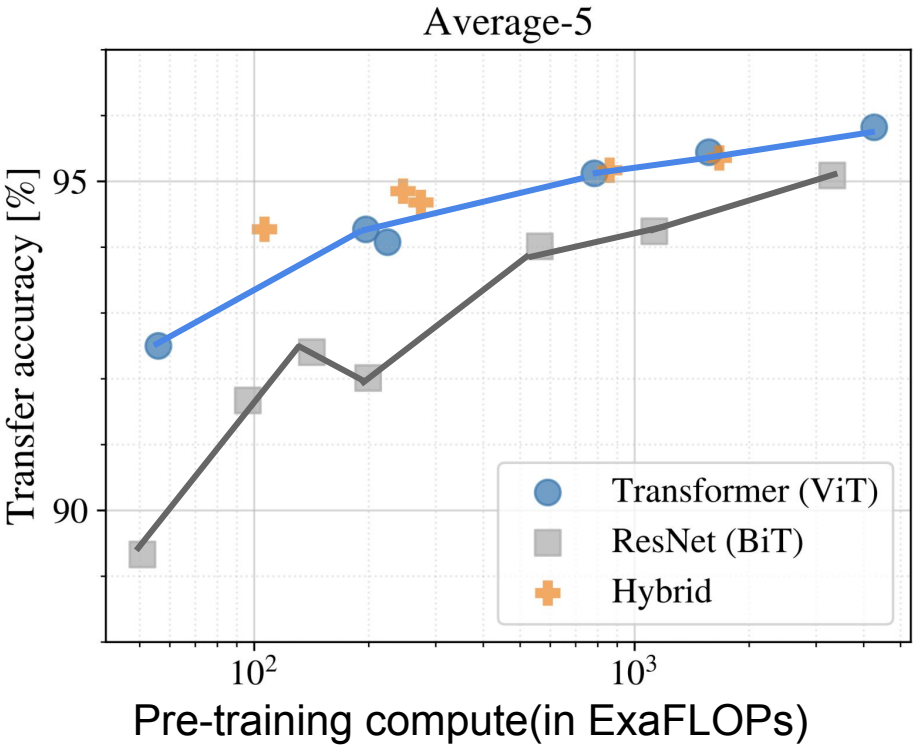
- On small pre-training dataset(ImageNet-1k, 1.3M images), ResNet performs better than ViT due to CNN spatial inductive biases that compensate for small dataset.
- On medium pre-training dataset(Imagenet-21k, 14M images), ViTs and ResNet performance are almost similar although ViTs perform slightly better.
- On large pre-training dataset(JFT, 303M images), large ViT outperforms ResNet and show no sign of plateau.

Vision Transformer vs SOTA CNN(ResNet)



Exa: 10^{18}, 1 FLOP = 1 multiply-add(wx+b) per second
FLOPs: floating point operations per second

Vision Transformer vs SOTA CNN(ResNet)

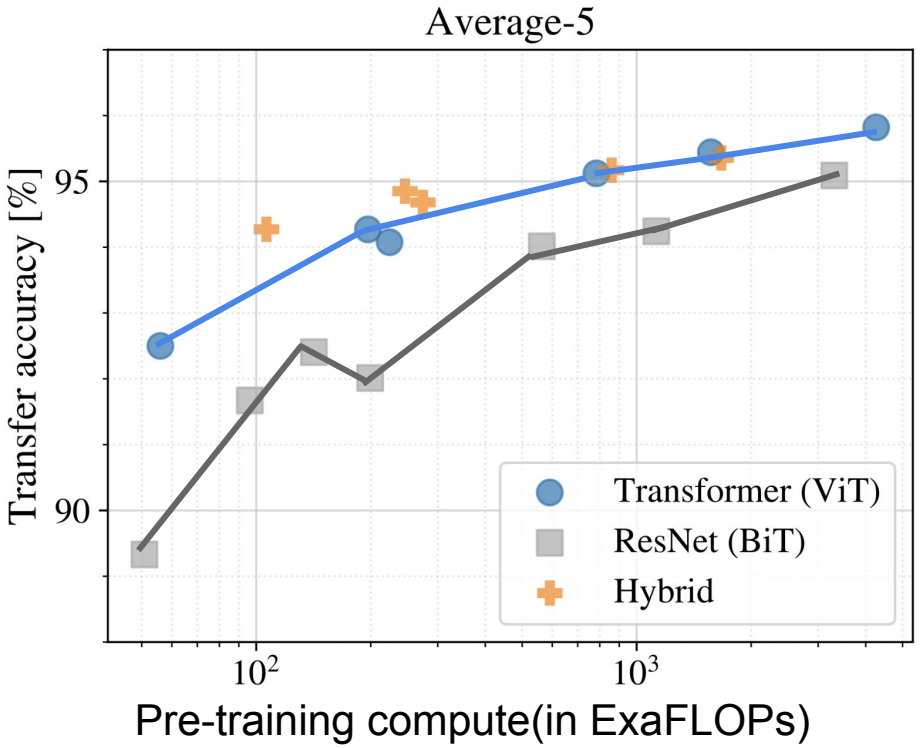


Exa: 10^{18}, 1 FLOP = 1 multiply-add(wx+b) per second
FLOPs: floating point operations per second

Pre-training compute

- ViT clearly outperforms ResNet on performance/compute trade-off.

Vision Transformer vs SOTA CNN(ResNet)

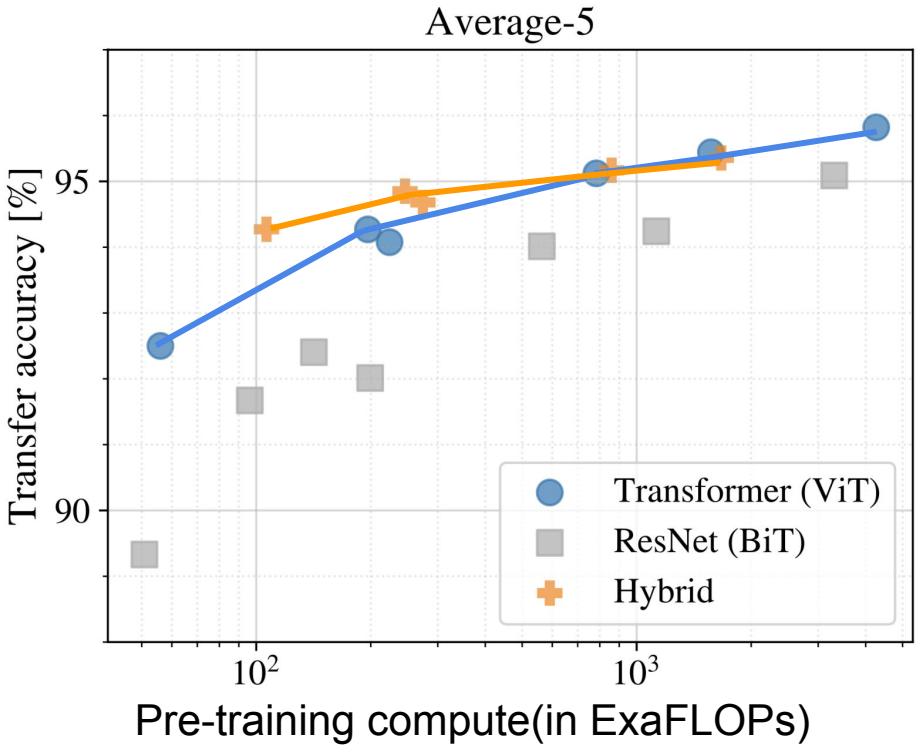


Exa: 10^{18}, 1 FLOP = 1 multiply-add(wx+b) per second
FLOPs: floating point operations per second

Pre-training compute

- ViT clearly outperforms ResNet on performance/compute trade-off.
- ViT uses approximately 2-4x less compute to achieve the same transfer accuracy(average of all downstream datasets).

Vision Transformer vs SOTA CNN(ResNet)

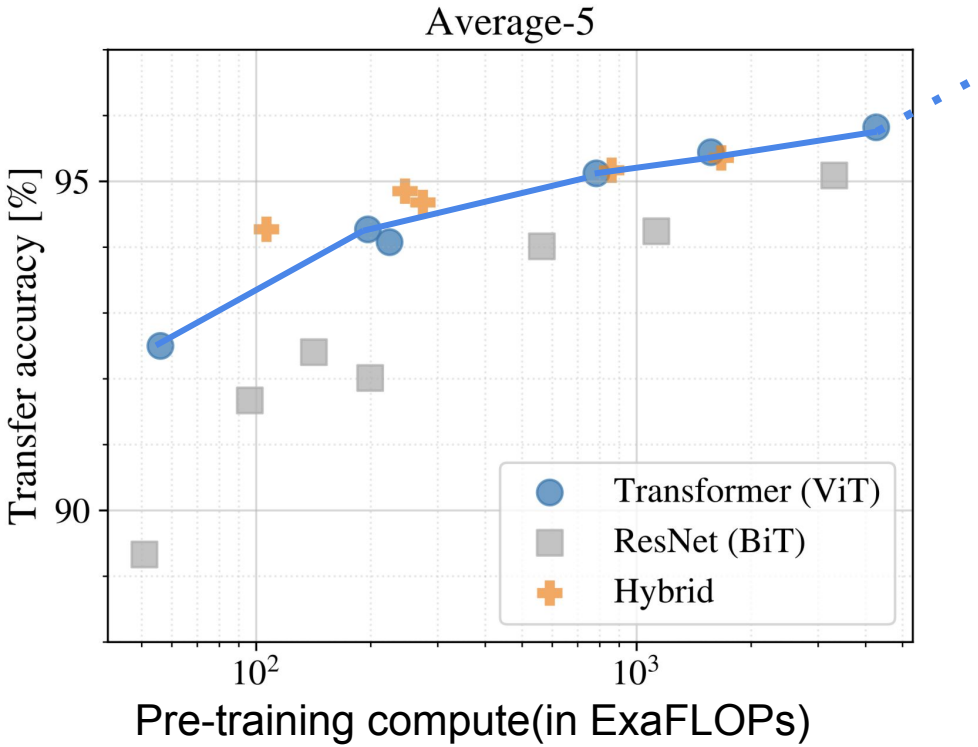


Exa: 10^{18}, 1 FLOP = 1 multiply-add(wx+b) per second
FLOPs: floating point operations per second

Pre-training compute

- ViT clearly outperforms ResNet on performance/compute trade-off.
- ViT uses approximately 2-4x less compute to achieve the same transfer accuracy(average of all downstream datasets).
- Hybrid(CNN+ViT) slightly outperforms ViT on relatively small compute, but vanishes on large compute budget.

Vision Transformer vs SOTA CNN(ResNet)



Exa: 10^{18}, 1 FLOP = 1 multiply-add(wx+b) per second
FLOPs: floating point operations per second

Pre-training compute

- ViT clearly outperforms ResNet on performance/compute trade-off.
- ViT uses approximately 2-4x less compute to achieve the same transfer accuracy(average of all downstream datasets).
- Hybrid(CNN+ViT) slightly outperforms ViT on relatively small compute, but vanishes on large compute budget.
- ViT shows extreme scaling behavior. Its performance doesn't seem to saturate for increased compute.

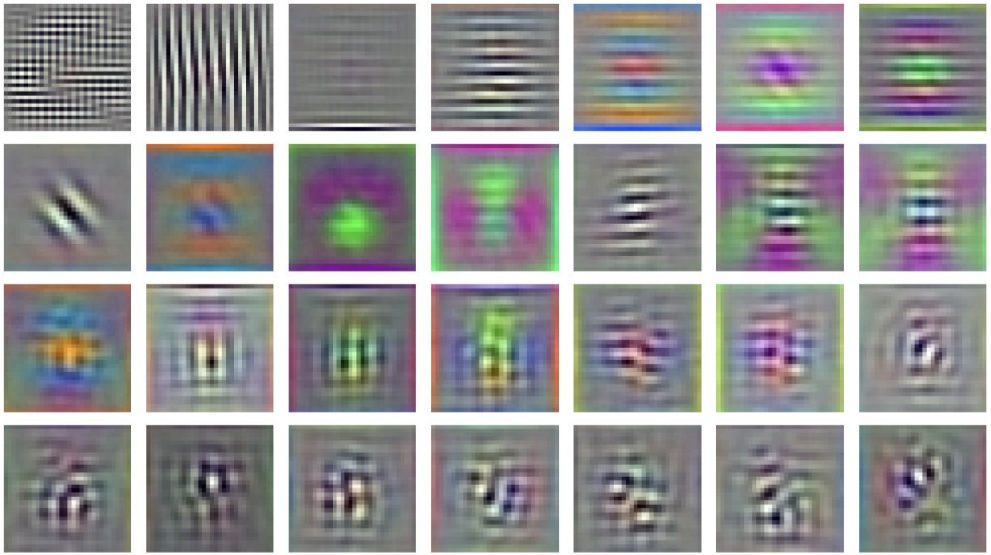
Inspecting ViT Representation

Vision Transformer shows remarkable performance when trained on massive datasets. How does it processes images internally?

Inspecting ViT Representation

Vision Transformer shows remarkable performance when trained on massive datasets. How does it processes images internally?

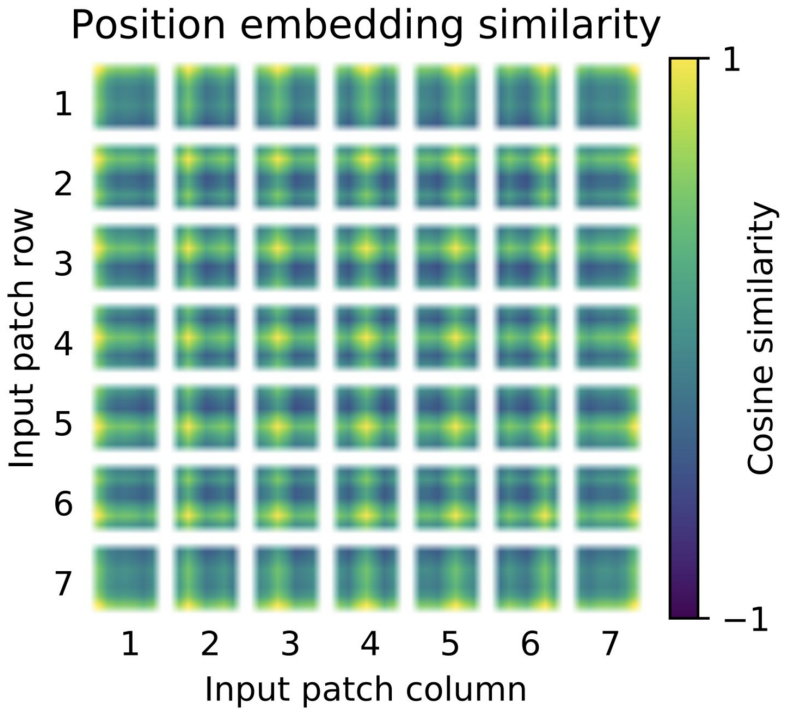
RGB embedding filters
(first 28 principal components)



The visualized linear embedding of flattened patches shows that the first layer of ViT(linear projection) learns the low level features(such as edges, blobs) of the input image much like ConvNets do!

Inspecting ViT Representation

Vision Transformer shows remarkable performance when trained on massive datasets. How does it processes images internally?



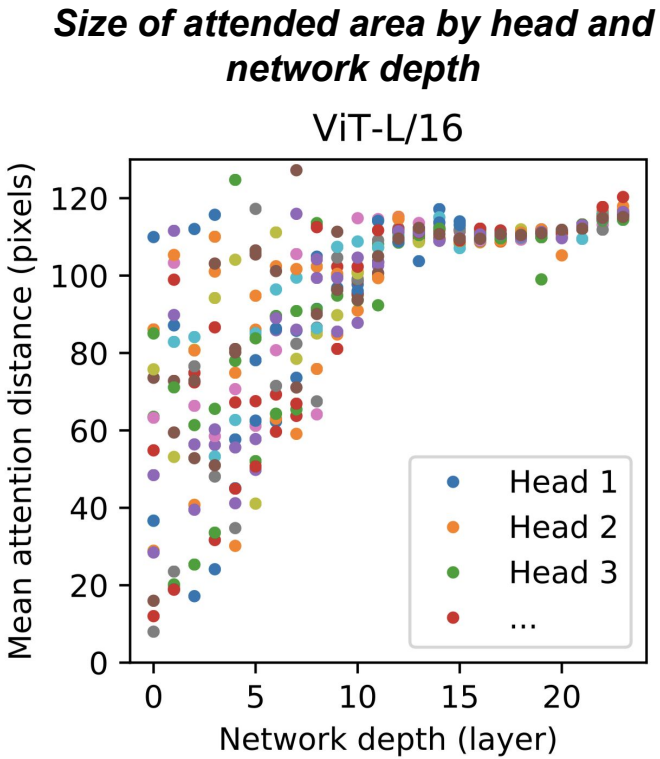
Vision Transformer(ViT) learns to encode the distance within the image in the similarity of position embeddings.

Closer patches are likely to have similar position embeddings.

The row-column structure is also maintained for all patches. The patches in the same row and column have similar embeddings.

Inspecting ViT Representation

Vision Transformer shows remarkable performance when trained on massive datasets. How does it processes images internally?



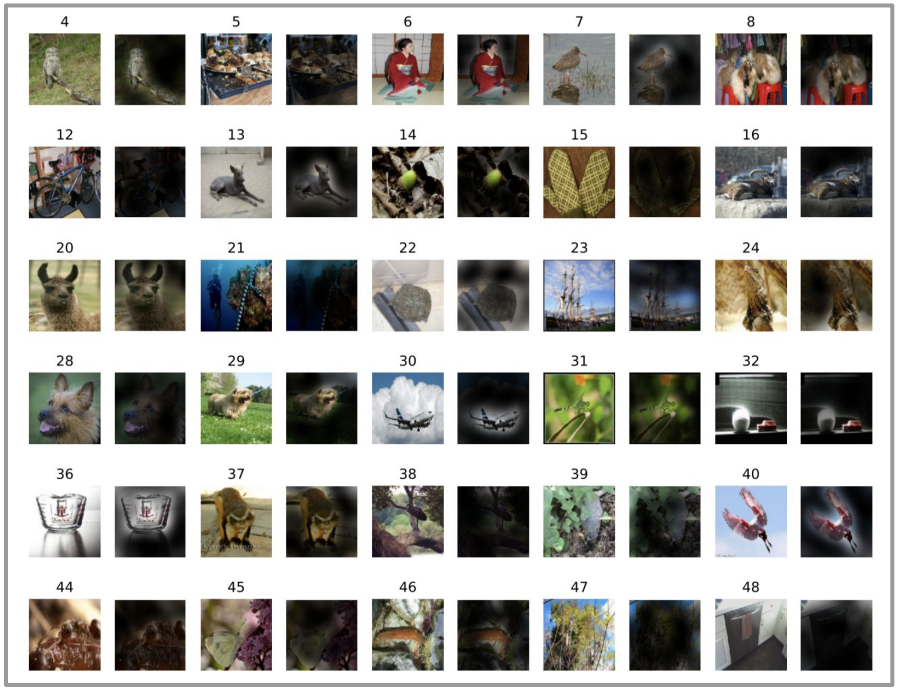
Each dot represents the mean attention distance(analogous to receptive field size in CNN) across images for every head(of 16 heads) and layer.

Mean attention distance increases with layers but some heads attends to most of image even in early layers.

Attention can attend to the whole image with fewer layers than CNNs.

Inspecting ViT Representation

Vision Transformer shows remarkable performance when trained on massive datasets. How does it processes images internally?



On global level, ViT attends to the meaningful part of the image and ignore the rest.

Inspecting ViT Representation

The representations learned by ViT were more explored in recent works. A few of them:

How Do Vision Transformers Work?

Namuk Park[1,2], **Songkuk Kim**[1]
[1]Yonsei University, [2]NAVER AI Lab
{namuk.park,songkuk}@yonsei.ac.kr

Abstract

The success of multi-head self-attentions (MSAs) for computer vision is now indisputable. However, little is known about how MSAs work. We present fundamental explanations to help better understand the nature of MSAs. In particular, we demonstrate the following properties of MSAs and Vision Transformers (ViTs): ❶ MSAs improve not only accuracy but also generalization by flattening the loss landscapes. Such improvement is primarily attributable to their data specificity, not long-range dependency. On the other hand, ViTs suffer from non-convex losses. Large datasets and loss landscape smoothing methods alleviate this problem; ❷ MSAs and Convs exhibit opposite behaviors. For example, MSAs are low-pass filters, but Convs are high-pass filters. Therefore, MSAs and Convs are complementary; ❸ Multi-stage neural networks behave like a series connection of small individual models. In addition, MSAs at the end of a stage play a key role in prediction. Based on these insights, we propose AlterNet, a model in which Conv blocks at the end of a stage are replaced with MSA blocks. AlterNet outperforms CNNs not only in large data regimes but also in small data regimes.

Do Vision Transformers See Like Convolutional Neural Networks?

Maithra Raghu
Google Research, Brain Team
maithrar@gmail.com

Thomas Unterthiner
Google Research, Brain Team
unterthiner@google.com

Simon Kornblith
Google Research, Brain Team
kornblith@google.com

Chiyuan Zhang
Google Research, Brain Team
chiyuan@google.com

Alexey Dosovitskiy
Google Research, Brain Team
adosovitskiy@google.com

Abstract

Convolutional neural networks (CNNs) have so far been the de-facto model for visual data. Recent work has shown that (Vision) Transformer models (ViT) can achieve comparable or even superior performance on image classification tasks. This raises a central question: *how are Vision Transformers solving these tasks*? Are they acting like convolutional networks, or learning entirely different visual representations? Analyzing the internal representation structure of ViTs and CNNs on image classification benchmarks, we find striking differences between the two architectures, such as ViT having more uniform representations across all layers. We explore how these differences arise, finding crucial roles played by self-attention, which enables early aggregation of global information, and ViT residual connections, which strongly propagate features from lower to higher layers. We study the ramifications for spatial localization, demonstrating ViTs successfully preserve input spatial information, with noticeable effects from different classification methods. Finally, we study the effect of (pretraining) dataset scale on intermediate features and transfer learning, and conclude with a discussion on connections to new architectures such as the MLP-Mixer.

Inspecting ViT Representation

The representations learned by ViT were more explored in recent works. A few of them:

WHAT DO VISION TRANSFORMERS LEARN? A VISUAL EXPLORATION

Amin Ghiasi[*1] **Hamid Kazemi**[*1] **Eitan Borgnia**[1] **Steven Reich**[1] **Manli Shu**[1]

Micah Goldblum[2] **Andrew Gordon Wilson**[2] **Tom Goldstein**[1]

[1] University of Maryland - College Park [2] New York University [*] Equal contribution

ABSTRACT

Vision transformers (ViTs) are quickly becoming the de-facto architecture for computer vision, yet we understand very little about why they work and what they learn. While existing studies visually analyze the mechanisms of convolutional neural networks, an analogous exploration of ViTs remains challenging. In this paper, we first address the obstacles to performing visualizations on ViTs. Assisted by these solutions, we observe that neurons in ViTs trained with language model supervision (e.g., CLIP) are activated by semantic concepts rather than visual features. We also explore the underlying differences between ViTs and CNNs, and we find that transformers detect image background features, just like their convolutional counterparts, but their predictions depend far less on high-frequency information. On the other hand, both architecture types behave similarly in the way features progress from abstract patterns in early layers to concrete objects in late layers. In addition, we show that ViTs maintain spatial information in all layers except the final layer. In contrast to previous works, we show that the last layer most likely discards the spatial information and behaves as a learned global pooling operation. Finally, we conduct large-scale visualizations on a wide range of ViT variants, including DeiT, CoaT, ConViT, PiT, Swin, and Twin, to validate the effectiveness of our method.

Intriguing Properties of Vision Transformers

Muzammal Naseer[†⋆] **Kanchana Ranasinghe**[+⋆] **Salman Khan**[⋆†]
Munawar Hayat[¶] **Fahad Shahbaz Khan**[⋆§] **Ming-Hsuan Yang**[‡○▽]
[†]Australian National University, [⋆]Mohamed bin Zayed University of AI, [+]Stony Brook University,
[¶]Monash University, [§]Linköping University, [‡]University of California, Merced,
[○]Yonsei University, [▽]Google Research
muzammal.naseer@anu.edu.au

Abstract

Vision transformers (ViT) have demonstrated impressive performance across numerous machine vision tasks. These models are based on multi-head self-attention mechanisms that can flexibly attend to a sequence of image patches to encode contextual cues. An important question is how such flexibility (in attending image-wide context conditioned on a given patch) can facilitate handling nuisances in natural images e.g., severe occlusions, domain shifts, spatial permutations, adversarial and natural perturbations. We systematically study this question via an extensive set of experiments encompassing three ViT families and provide comparisons with a high-performing convolutional neural network (CNN). We show and analyze the following intriguing properties of ViT: (a) Transformers are highly robust to severe occlusions, perturbations and domain shifts, *e.g.,* retain as high as 60% top-1 accuracy on ImageNet even after randomly occluding 80% of the image content. (b) The robustness towards occlusions is not due to texture bias, instead we show that ViTs are significantly less biased towards local textures, compared to CNNs. When properly trained to encode shape-based features, ViTs demonstrate shape recognition capability comparable to that of human visual system, previously unmatched in the literature. (c) Using ViTs to encode shape representation leads to an interesting consequence of accurate semantic segmentation without pixel-level supervision. (d) Off-the-shelf features from a single ViT model can be combined to create a feature ensemble, leading to high accuracy rates across a range of classification datasets in both traditional and few-shot learning paradigms. We show effective features of ViTs are due to flexible and dynamic receptive fields possible via self-attention mechanisms. Code: https://git.io/Js15X.

Self-supervision

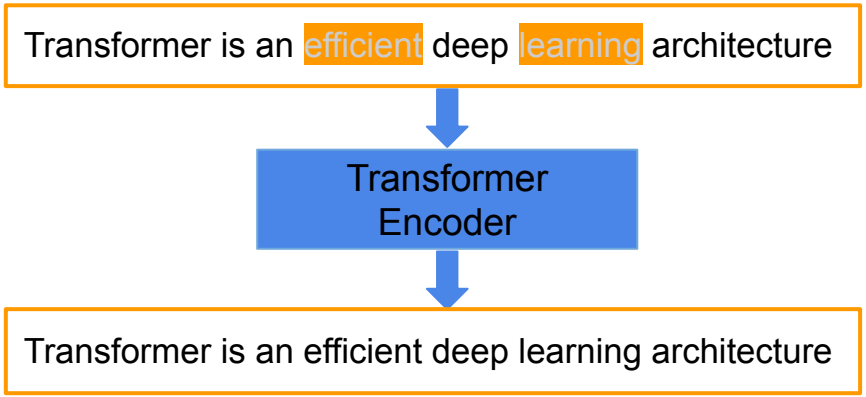
Most of Transformers success in NLP is the result of large-scale self-supervised pre-training where Transformer is trained on massive unlabelled data from the web.

Self-supervision

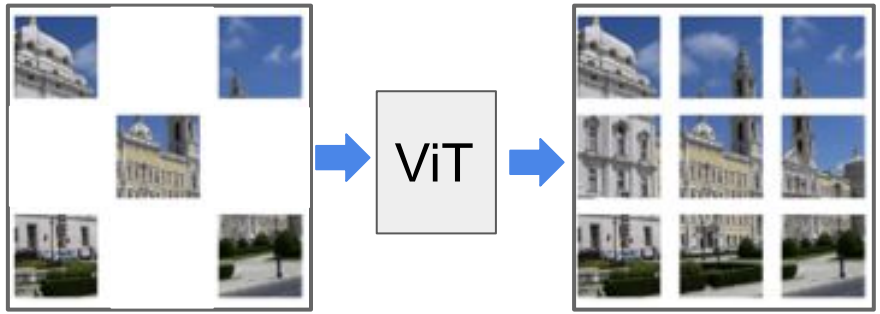
Most of Transformers success in NLP is the result of large-scale self-supervised pre-training where Transformer is trained on massive unlabelled data from the web.

Using masked work prediction technique that were used in BERT(randomly masking words in input sentence), ViT designers also tried the same technique where they masked 50% of patches(masked patch prediction) but achieved less performance than supervised pre-training(79.9% ACC on ImageNet while supervised pre-training is ~85%).

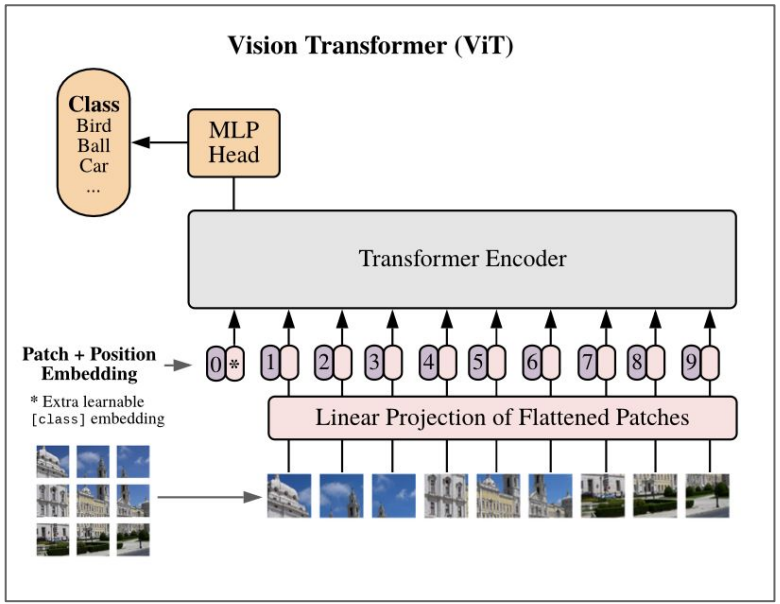
Masked-word prediction in BERT

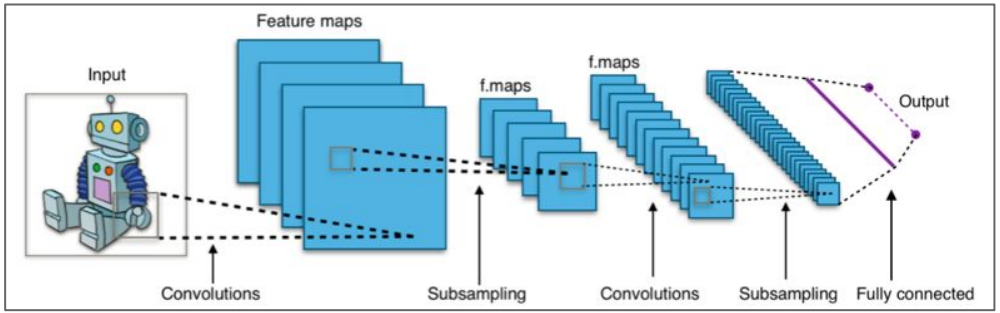


Masked patch prediction ViT



ViTs Vs ConvNets

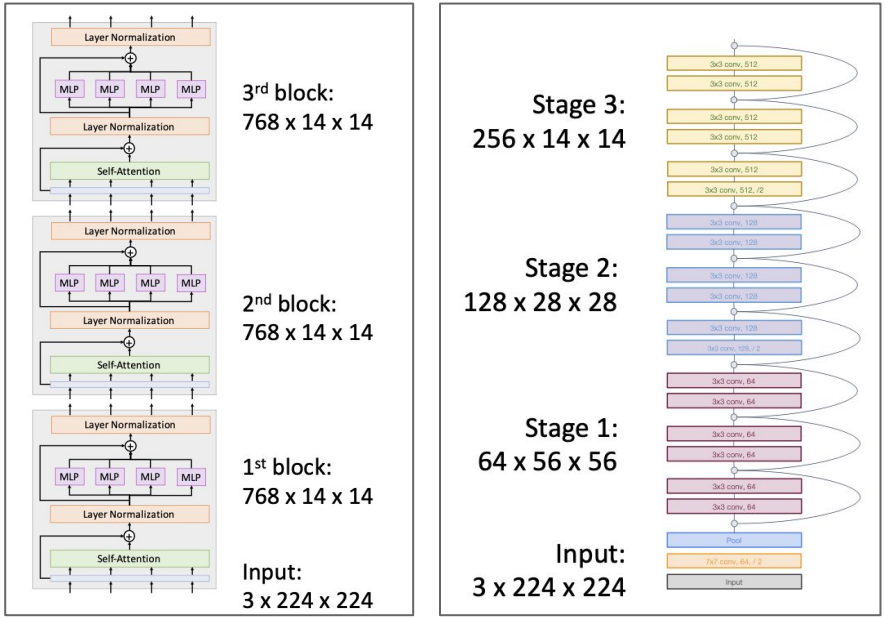




ViTs Vs ConvNets

ViTs and ConvNets can both learn useful representations but they are fundamentally different. ViTs are isotropic architectures(maintain same resolutions and number of channels across the whole network) where as ConvNets are hierarchical(resolution decrease and channel increase with depth).

ViT is isotropic architecture - maintain same resolution and channels across in all blocks



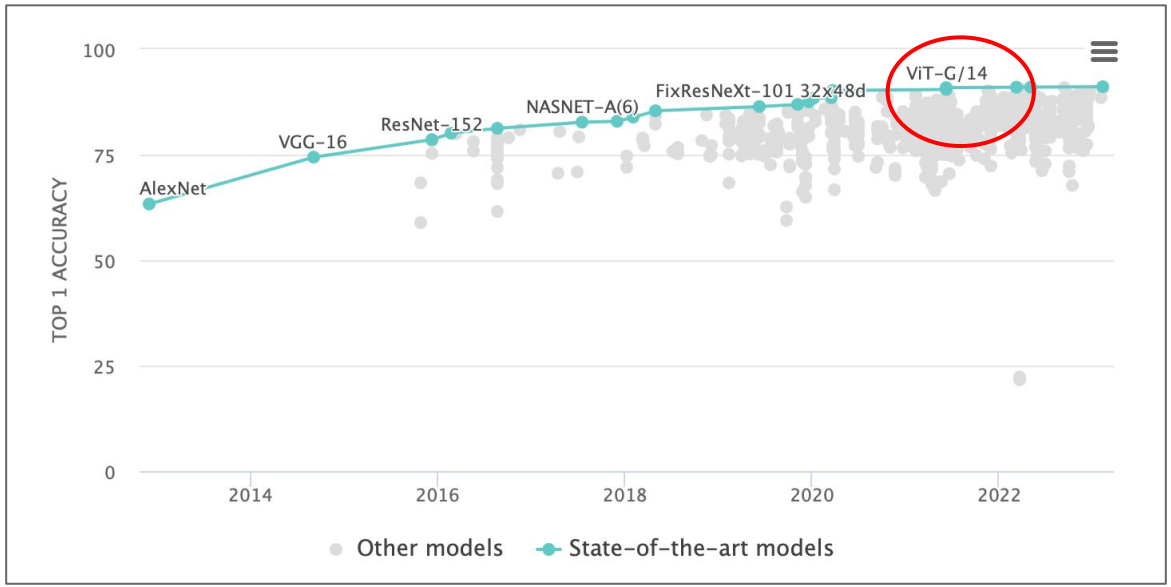
ConvNets are hierarchical. Resolution decrease with depth. Channel increase with depth.

Image: Justin Johnson

ViTs Vs ConvNets - Which is better?

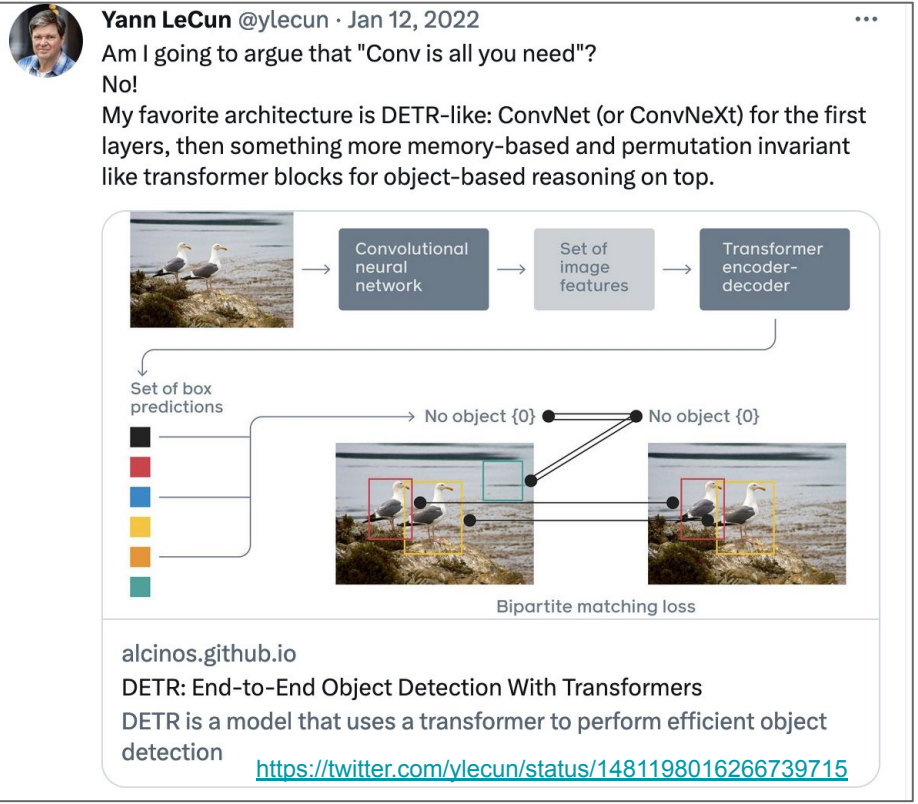
Vision Transformers are one of the greatest and recent remarkable papers in visual representation learning. They have outperformed ConvNets on various benchmarks, but it is still an open research question whether they will replace ConvNets or not.

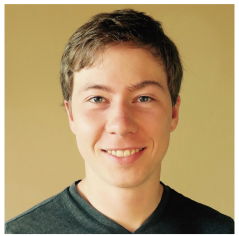
ViT was once SOTA on popular computer vision benchmarks such as ImageNet and Cifar100.



SOTA on ImageNet classification. Source: Paper with Code

ViTs Vs ConvNets - Which is better?





Vision transformers are an evolution, not a revolution. We can still fundamentally solve the same problems as with CNNs.

Main benefit is probably speed: Matrix multiply is more hardware- friendly than convolution, so ViTs with same FLOPs as CNNs can train and run much faster.

Justin Johnson, University of Michigan

ViTs Implementations

Vision Transformer and MLP-Mixer Architectures

In this repository we release models from the papers

- An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale
- MLP-Mixer: An all-MLP Architecture for Vision
- How to train your ViT? Data, Augmentation, and Regularization in Vision Transformers
- When Vision Transformers Outperform ResNets without Pretraining or Strong Data Augmentations
- LiT: Zero-Shot Transfer with Locked-image text Tuning
- Surrogate Gap Minimization Improves Sharpness-Aware Training

The models were pre-trained on the ImageNet and ImageNet-21k datasets. We provide the code for fine-tuning the released models in JAX/Flax.

Table of contents:

Vision Transformer repository is the official implementation of Vision Transformers in JAX/Flax. It also contain other related models such as MLP-Mixer.

https://github.com/google-research/vision_transformer

ViTs Implementations

Big Vision

This codebase is designed for training large-scale vision models using Cloud TPU VMs or GPU machines. It is based on Jax/Flax libraries, and uses tf.data and TensorFlow Datasets for scalable and reproducible input pipelines.

The open-sourcing of this codebase has two main purposes:

1. Publishing the code of research projects developed in this codebase (see a list below).
2. Providing a strong starting point for running large-scale vision experiments on GPU machines and Google Cloud TPUs, which should scale seamlessly and out-of-the box from a single TPU core to a distributed setup with up to 2048 TPU cores.

`big_vision` aims to support research projects at Google. We are unlikely to work on feature requests or accept external contributions, unless they were pre-approved (ask in an issue first). For a well-supported transfer-only codebase, see also vision_transformer.

The following research projects were originally conducted in the `big_vision` codebase:

Architecture research

- An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, by Alexey Dosovitskiy*, Lucas Beyer*, Alexander Kolesnikov*, Dirk Weissenborn*, Xiaohua Zhai*, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby*

Big Vision is another codebase of ViT and other related models/papers. It contains checkpoints and instructions for training/fine-tuning ViT models.

https://github.com/google-research/big_vision

ViTs Implementations

Vision Transformer - Pytorch

Implementation of Vision Transformer, a simple way to achieve SOTA in vision classification with only a single transformer encoder, in Pytorch. Significance is further explained in Yannic Kilcher's video. There's really not much to code here, but may as well lay it out for everyone so we expedite the attention revolution.

For a Pytorch implementation with pretrained models, please see Ross Wightman's repository here.

The official Jax repository is here.

A tensorflow2 translation also exists here, created by research scientist Junho Kim! 🙏

Flax translation by Enrico Shippole!

Install

```
$ pip install vit-pytorch
```

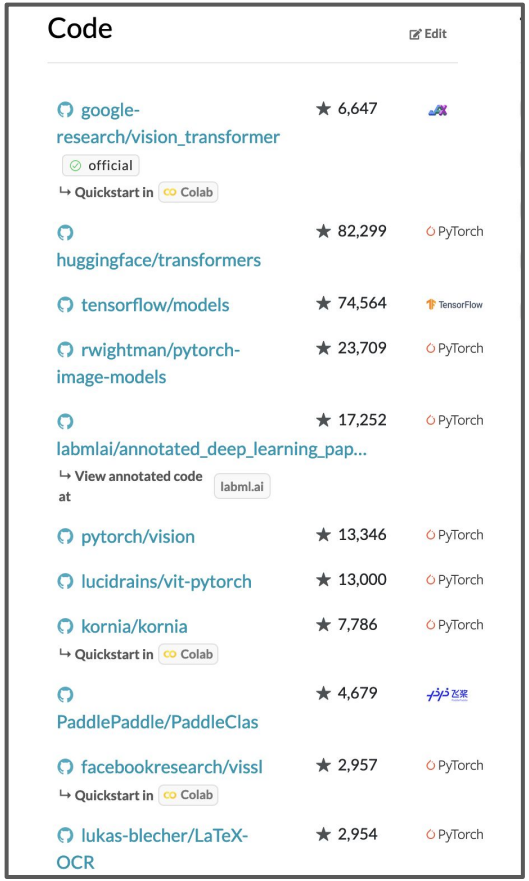
Usage

```
import torch
from vit_pytorch import ViT

v = ViT(
    image_size = 256,
    patch_size = 32,
    num_classes = 1000,
    dim = 1024,
    depth = 6,
    heads = 16,
    mlp_dim = 2048,
    dropout = 0.1,
    emb_dropout = 0.1
)

img = torch.randn(1, 3, 256, 256)

preds = v(img) # (1, 1000)
```

Vision Transformer Pytorch contains implementation of ViT and other popular Vision Transformer models.

https://github.com/lucidrains/vit-pytorch

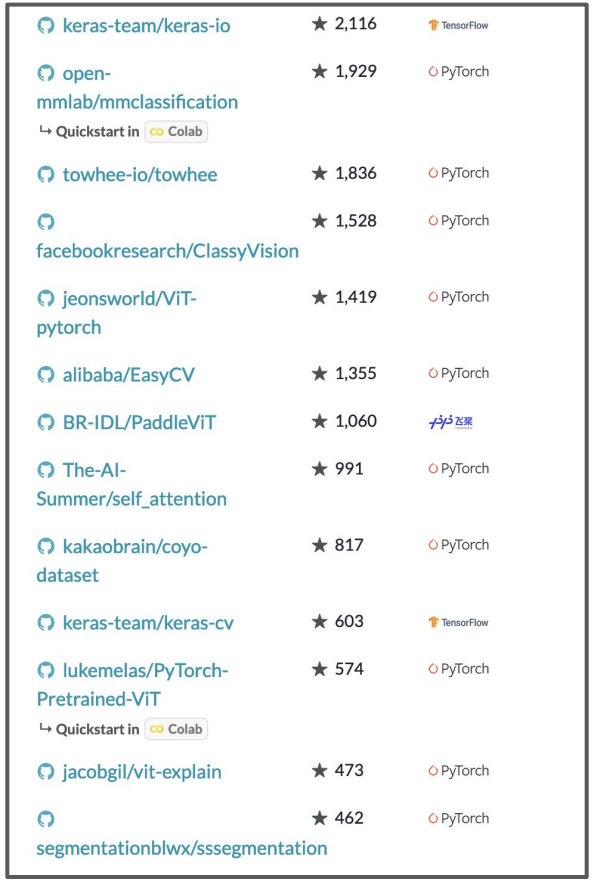
Other ViT implementations on Paper With Code



Code

Edit

google-research/vision_transformer official ↳ Quickstart in Colab	★ 6,647	
huggingface/transformers	★ 82,299	PyTorch
tensorflow/models	★ 74,564	TensorFlow
rwightman/pytorch-image-models	★ 23,709	PyTorch
labmlai/annotated_deep_learning_pap... ↳ View annotated code at labml.ai	★ 17,252	PyTorch
pytorch/vision	★ 13,346	PyTorch
lucidrains/vit-pytorch	★ 13,000	PyTorch
kornia/kornia ↳ Quickstart in Colab	★ 7,786	PyTorch
PaddlePaddle/PaddleClas	★ 4,679	
facebookresearch/vissl ↳ Quickstart in Colab	★ 2,957	PyTorch
lukas-blecher/LaTeX-OCR	★ 2,954	PyTorch



keras-team/keras-io	★ 2,116	TensorFlow
open-mmlab/mmclassification ↳ Quickstart in Colab	★ 1,929	PyTorch
towhee-io/towhee	★ 1,836	PyTorch
facebookresearch/ClassyVision	★ 1,528	PyTorch
jeonsworld/ViT-pytorch	★ 1,419	PyTorch
alibaba/EasyCV	★ 1,355	PyTorch
BR-IDL/PaddleViT	★ 1,060	
The-AI-Summer/self_attention	★ 991	PyTorch
kakaobrain/coyo-dataset	★ 817	PyTorch
keras-team/keras-cv	★ 603	TensorFlow
lukemelas/PyTorch-Pretrained-ViT ↳ Quickstart in Colab	★ 574	PyTorch
jacobgil/vit-explain	★ 473	PyTorch
segmentationblwx/sssegmentation	★ 462	PyTorch

https://paperswithcode.com/paper/an-image-is-worth-16x16-words-transformers-1

ViT - Follow-up Works

Scaling Vision Transformers

Xiaohua Zhai*, Alexander Kolesnikov*, Neil Houlsby, Lucas Beyer*

Google Research, Brain Team, Zürich

{xzhai, akolesnikov, neilhoulsby, lbeyer}@google.com

Abstract

Attention-based neural networks such as the Vision Transformer (ViT) have recently attained state-of-the-art results on many computer vision benchmarks. Scale is a primary ingredient in attaining excellent results, therefore, understanding a model's scaling properties is a key to designing future generations effectively. While the laws for scaling Transformer language models have been studied, it is unknown how Vision Transformers scale. To address this, we scale ViT models and data, both up and down, and characterize the relationships between error rate, data, and compute. Along the way, we refine the architecture and training of ViT, reducing memory consumption and increasing accuracy of the resulting models. As a result, we successfully train a ViT model with two billion parameters, which attains a new state-of-the-art on ImageNet of 90.45% *top-1 accuracy. The model also performs well for few-shot transfer, for example, reaching* 84.86% *top-1 accuracy on ImageNet with only 10 examples per class.*

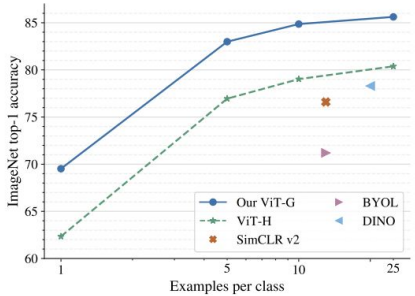


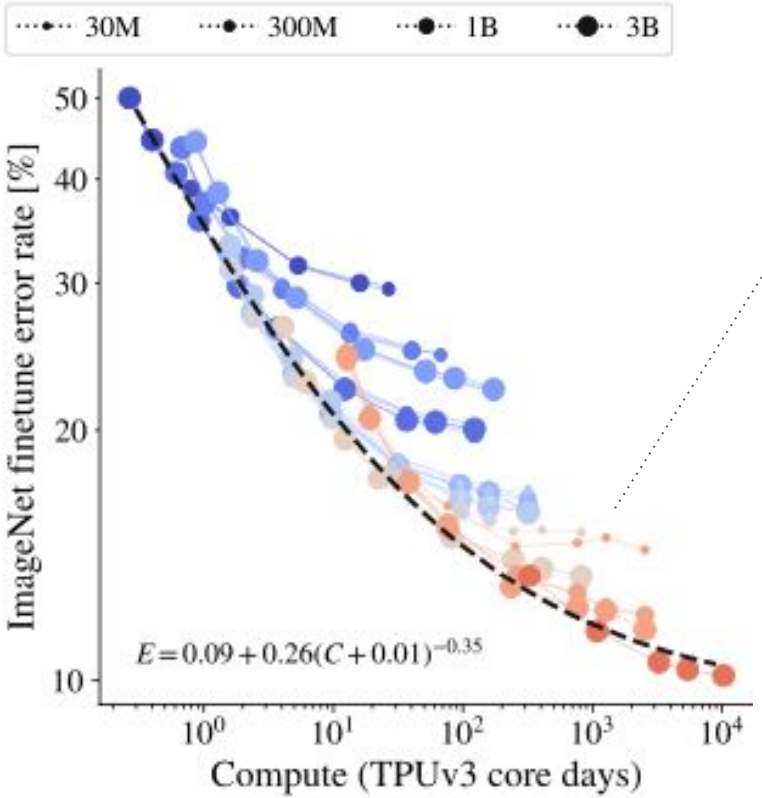
Figure 1. Few-shot transfer results. Our ViT-G model reaches 84.86% top-1 accuracy on ImageNet with 10-shot linear evaluation.

tion tasks. In particular, we experiment with models ranging from five million to two billion parameters, datasets ranging from one million to three billion training images and compute budgets ranging from below one TPUv3 core-day to beyond 10 000 core-days. Our main contribution is a characterization of the performance-compute frontier for ViT models, on two datasets.

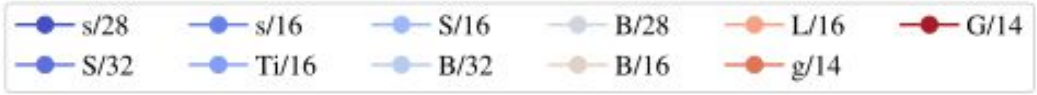
Successfully train ViT of 2B parameters and achieve 90.45% top-1 accuracy on ImageNet.

ViT - Follow-up Works

Size of marker: size of training data
Models trained on fewer images: small markers



ViT of different sizes
Smaller models: blue shading, large models: red shading

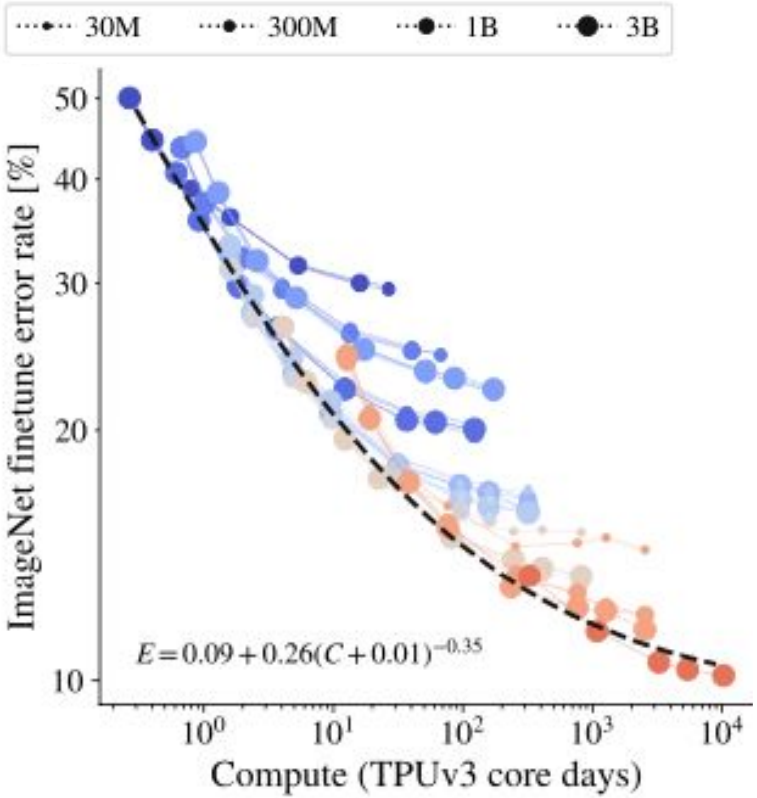


Models trained for fewer images(small markers) saturates and stop improving when trained for longer.

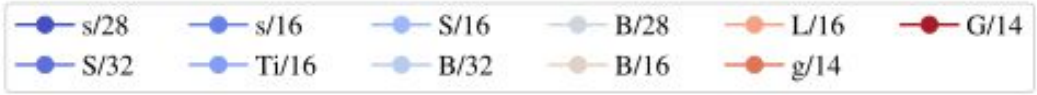
Zhai et al., "Scaling Vision Transformers", 2022

ViT - Follow-up Works

Size of marker: size of training data
Models trained on fewer images: small markers



ViT of different sizes
Smaller models: blue shading, large models: red shading



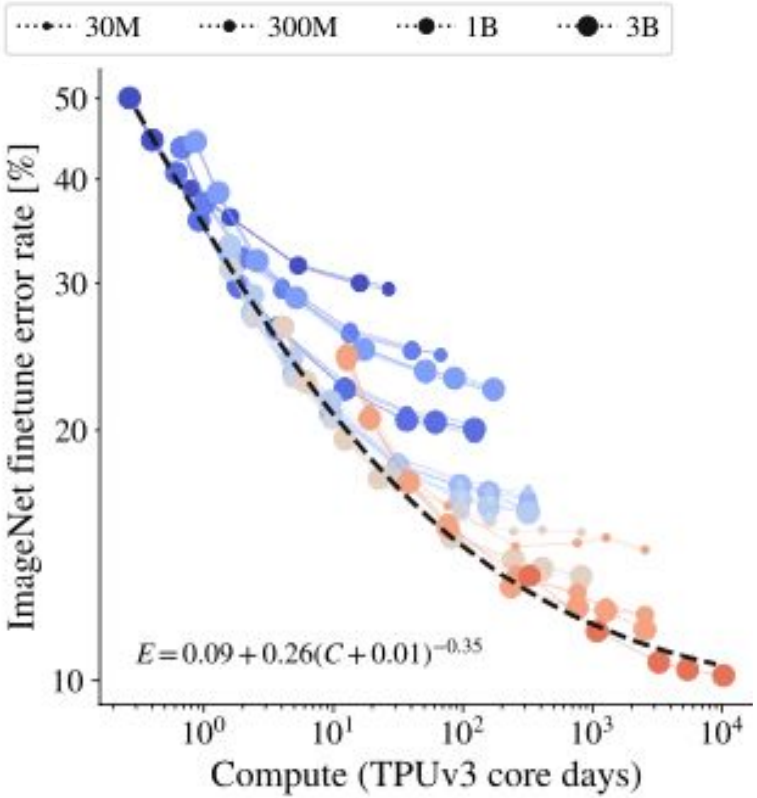
Models trained for fewer images(small markers) saturates and stop improving when trained for longer.

Scaling up compute, model and data together improves representation quality.

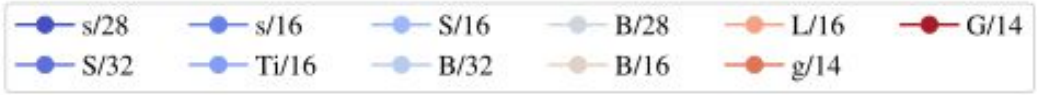
Zhai et al., "Scaling Vision Transformers", 2022

ViT - Follow-up Works

Size of marker: size of training data
Models trained on fewer images: small markers



ViT of different sizes
Smaller models: blue shading, large models: red shading



Models trained for fewer images(small markers) saturates and stop improving when trained for longer.

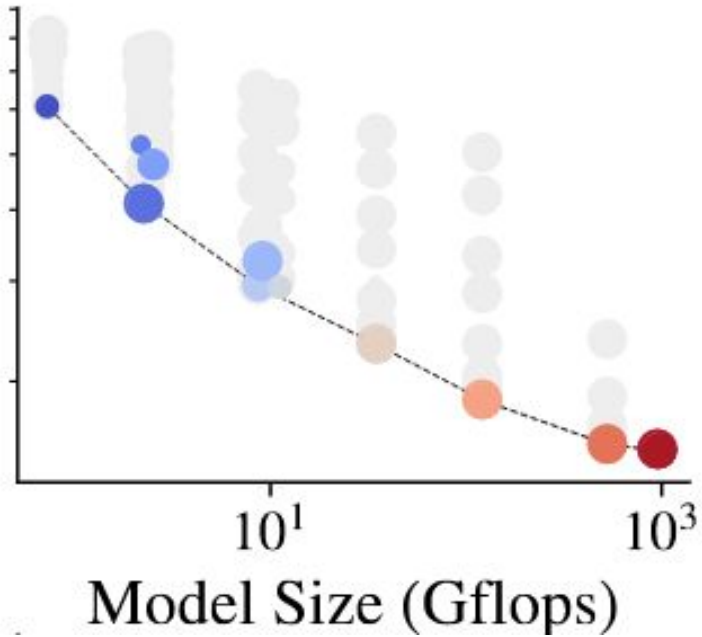
Scaling up compute, model and data together improves representation quality.

A model (one with connected points) with largest size, dataset size, and compute achieves the lowest error rate. This is consistent across fine-tuning and few-shot learning.

Zhai et al., "Scaling Vision Transformers", 2022

ViT - Follow-up Works

Size of marker: size of training data
Models trained on fewer images: small markers



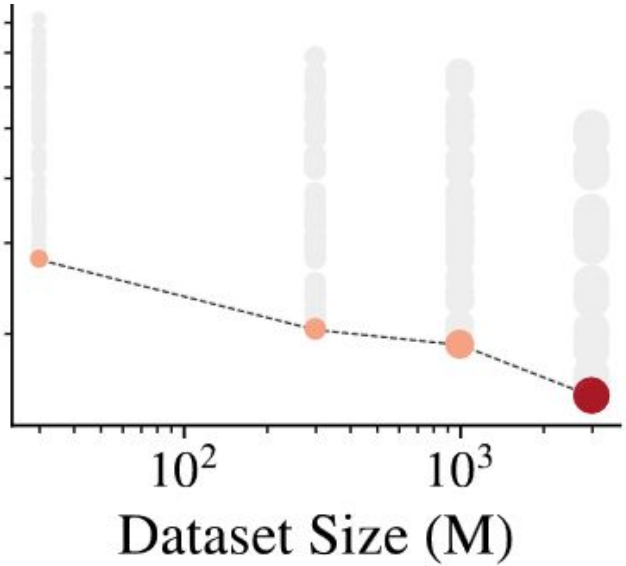
ViT of different sizes
Smaller models: blue shading, large models: red shading

Representation quality is bottlenecked by model size. Small models are not able to benefit from either the largest data or compute resources while large models benefit from both.

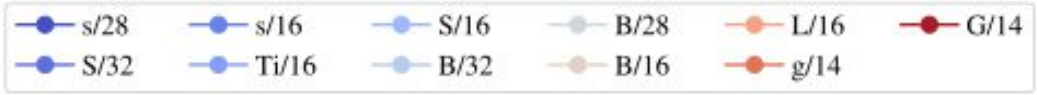
Zhai et al., "Scaling Vision Transformers", 2022

ViT - Follow-up Works

Size of marker: size of training data
Models trained on fewer images: small markers



ViT of different sizes
Smaller models: blue shading, large models: red shading



When scaling up the model size, the representation quality is limited by small datasets. Large models benefit from increased dataset but smaller models are hurted by large datasets.

Zhai et al., "Scaling Vision Transformers", 2022

ViT - Follow-up Works

Better plain ViT baselines for ImageNet-1k

Lucas Beyer Xiaohua Zhai Alexander Kolesnikov
Google Research, Brain Team Zürich

https://github.com/google-research/big_vision

Abstract

It is commonly accepted that the Vision Transformer model requires sophisticated regularization techniques to excel at ImageNet-1k scale data. Surprisingly, we find this is not the case and standard data augmentation is sufficient. This note presents a few minor modifications to the original Vision Transformer (ViT) vanilla training setting that dramatically improve the performance of plain ViT models. Notably, 90 epochs of training surpass 76% top-1 accuracy in under seven hours on a TPUv3-8, similar to the classic ResNet50 baseline, and 300 epochs of training reach 80% in less than one day.

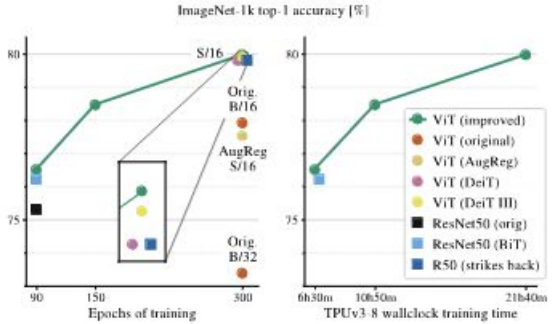


Figure 1. Comparison of ViT model for this note to state-of-the-art

Simple ViT can achieve comparable performance to ResNet with few modifications: standard data augmentation, using fixed positional embeddings instead of learnable embeddings, batchsize of 1024 instead of 4096, global average pool instead of extra class token, MLP head instead of linear head.

	90ep	150ep	300ep
Our improvements	76.5	78.5	80.0
no RandAug+MixUp	73.6	73.7	73.7
Posemb: sincos2d → learned	75.0	78.0	79.6
Batch-size: 1024 → 4096	74.7	77.3	78.6
Global Avgpool → [cls] token	75.0	76.9	78.2
Head: MLP → linear	76.7	78.6	79.8
Original + RandAug + MixUp	71.6	74.8	76.1
Original	66.8	67.2	67.1

ViT - Follow-up Works

How to train your ViT? Data, Augmentation, and Regularization in Vision Transformers

Andreas Steiner* *andstein@google.com*

Alexander Kolesnikov* *akolesnikov@google.com*

Xiaohua Zhai* *xzhai@google.com*

Ross Wightman† *rwightman@gmail.com*

Jakob Uszkoreit *usz@google.com*

Lucas Beyer* *lbeyer@google.com*

*Google Research, Brain Team, Zürich * Equal technical contribution, † independent researcher*

Reviewed on OpenReview: *https://openreview.net/forum?id=4nPswr1KcP*

Abstract

Vision Transformers (ViT) have been shown to attain highly competitive performance for a wide range of vision applications, such as image classification, object detection and semantic image segmentation. In comparison to convolutional neural networks, the Vision Transformer's weaker inductive bias is generally found to cause an increased reliance on model regularization or data augmentation ("AugReg" for short) when training on smaller training datasets. We conduct a systematic empirical study in order to better understand the interplay between the amount of training data, AugReg, model size and compute budget.[1] As one result of this study we find that the combination of increased compute and AugReg can yield models with the same performance as models trained on an order of magnitude more training data: we train ViT models of various sizes on the public ImageNet-21k dataset which either match or outperform their counterparts trained on the larger, but not publicly available JFT-300M dataset.

Problem with ViTs: Training ViTs require an insane amount of training data. What's the interplay between amount of training data, augmentation/regularization, model size, and compute budget?

ViT - Follow-up Works

How to train your ViT? Data, Augmentation, and Regularization in Vision Transformers

Andreas Steiner* *andstein@google.com*

Alexander Kolesnikov* *akolesnikov@google.com*

Xiaohua Zhai* *xzhai@google.com*

Ross Wightman† *rwightman@gmail.com*

Jakob Uszkoreit *usz@google.com*

Lucas Beyer* *lbeyer@google.com*

*Google Research, Brain Team, Zürich * Equal technical contribution, † independent researcher*

Reviewed on OpenReview: *https://openreview.net/forum?id=4nPswr1KcP*

Abstract

Vision Transformers (ViT) have been shown to attain highly competitive performance for a wide range of vision applications, such as image classification, object detection and semantic image segmentation. In comparison to convolutional neural networks, the Vision Transformer's weaker inductive bias is generally found to cause an increased reliance on model regularization or data augmentation ("AugReg" for short) when training on smaller training datasets. We conduct a systematic empirical study in order to better understand the interplay between the amount of training data, AugReg, model size and compute budget.[1] As one result of this study we find that the combination of increased compute and AugReg can yield models with the same performance as models trained on an order of magnitude more training data: we train ViT models of various sizes on the public ImageNet-21k dataset which either match or outperform their counterparts trained on the larger, but not publicly available JFT-300M dataset.

Problem with ViTs: Training ViTs require an insane amount of training data. What's the interplay between amount of training data, augmentation/regularization, model size, and compute budget?

The paper showed that carefully selected amount of regularization and data augmentation can compensate for ViT's need of more training data. In particular, a good mix of regularization and augmentations roughly corresponds to a 10x increase in training data size.

ViT - Follow-up Works

How to train your ViT? Data, Augmentation, and Regularization in Vision Transformers

Andreas Steiner* *andstein@google.com*

Alexander Kolesnikov* *akolesnikov@google.com*

Xiaohua Zhai* *xzhai@google.com*

Ross Wightman† *rwightman@gmail.com*

Jakob Uszkoreit *usz@google.com*

Lucas Beyer* *lbeyer@google.com*

*Google Research, Brain Team, Zürich * Equal technical contribution, † independent researcher*

Reviewed on OpenReview: *https://openreview.net/forum?id=4nPswr1KcP*

Abstract

Vision Transformers (ViT) have been shown to attain highly competitive performance for a wide range of vision applications, such as image classification, object detection and semantic image segmentation. In comparison to convolutional neural networks, the Vision Transformer's weaker inductive bias is generally found to cause an increased reliance on model regularization or data augmentation ("AugReg" for short) when training on smaller training datasets. We conduct a systematic empirical study in order to better understand the interplay between the amount of training data, AugReg, model size and compute budget.[1] As one result of this study we find that the combination of increased compute and AugReg can yield models with the same performance as models trained on an order of magnitude more training data: we train ViT models of various sizes on the public ImageNet-21k dataset which either match or outperform their counterparts trained on the larger, but not publicly available JFT-300M dataset.

Problem with ViTs: Training ViTs require an insane amount of training data. What's the interplay between amount of training data, augmentation/regularization, model size, and compute budget?

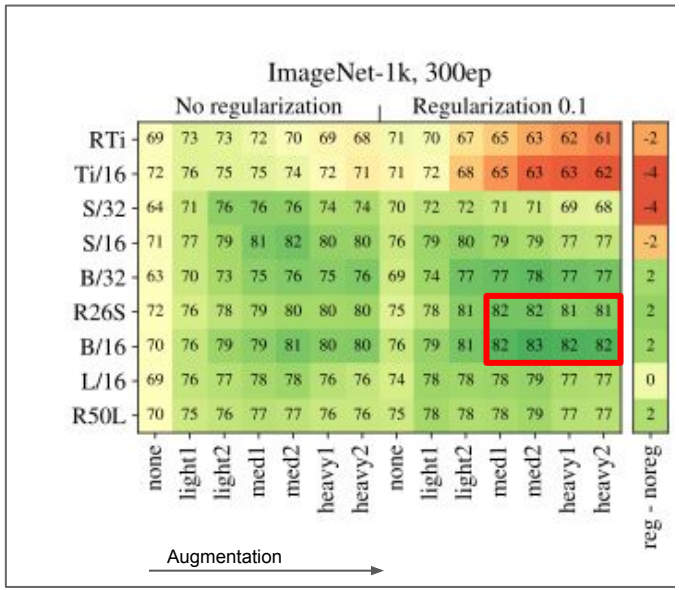
The paper showed that carefully selected amount of regularization and data augmentation can compensate for ViT's need of more training data. In particular, a good mix of regularization and augmentations roughly corresponds to a 10x increase in training data size.

Regularization: weight decay, stochastic depth, dropout(in FC)

Data augmentation: mixup, randaugment

ViT - Follow-up Works

For small datasets, both regularization and data augmentation help.

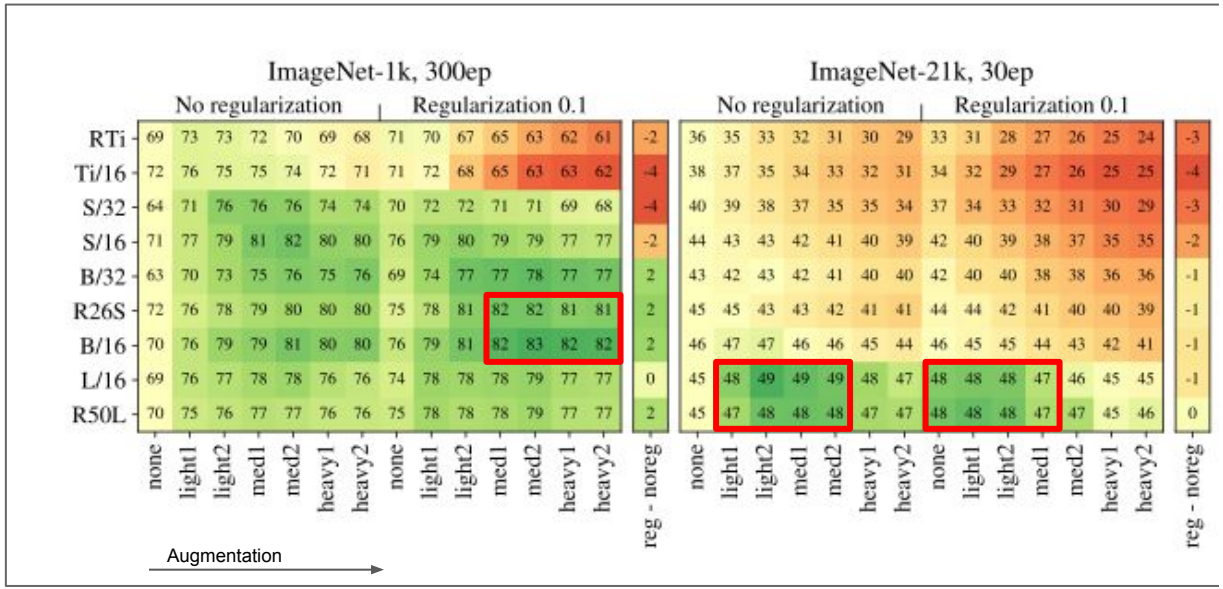


Steiner et al, "How to train your ViT? Data, Augmentation, and Regularization in Vision Transformers", 2021

ViT - Follow-up Works

For small datasets, both regularization and data augmentation help.

For medium-large datasets, both regularization and data augmentation hurt when training compute is fixed.



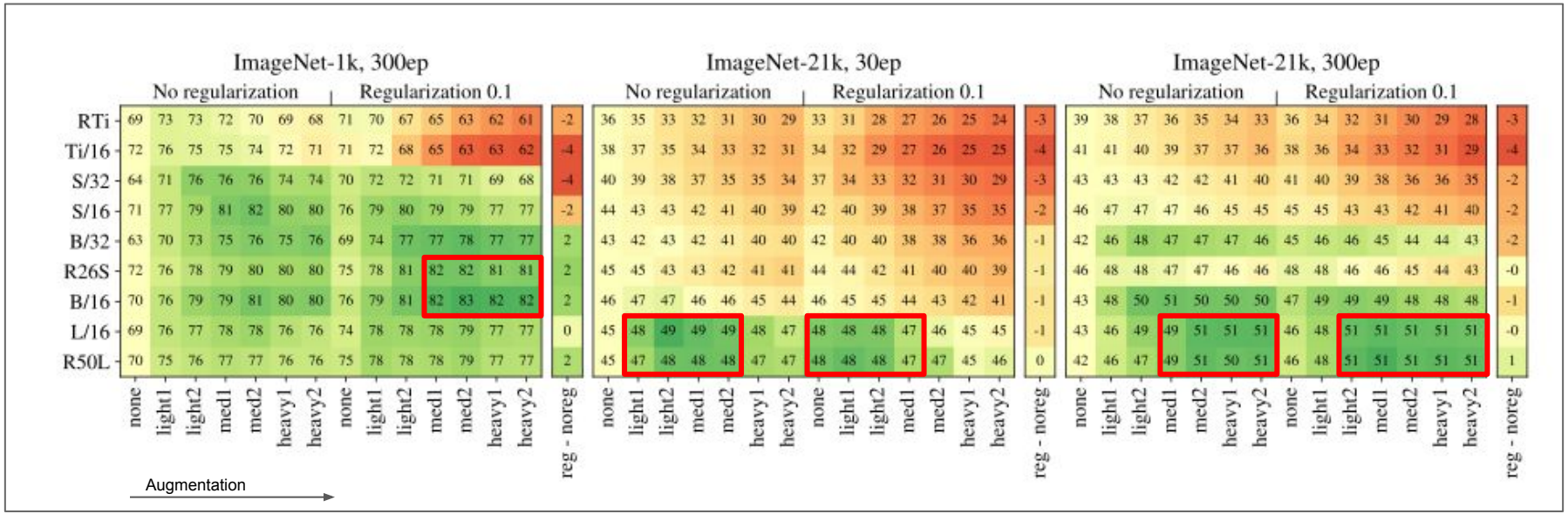
Steiner et al, "How to train your ViT? Data, Augmentation, and Regularization in Vision Transformers", 2021

ViT - Follow-up Works

For small datasets, both regularization and data augmentation help.

For medium-large datasets, both regularization and data augmentation hurt when training compute is fixed.

For medium-large datasets, both regularization and data augmentation yield better accuracy for increased training compute(ie, 30ep → 300ep).



Steiner et al, "How to train your ViT? Data, Augmentation, and Regularization in Vision Transformers", 2021

ViT - Follow-up Works

Scaling Vision Transformers to 22 Billion Parameters

Mostafa Dehghani* Josip Djolonga* Basil Mustafa* Piotr Padlewski* Jonathan Heek*
Justin Gilmer Andreas Steiner Mathilde Caron Robert Geirhos Ibrahim Alabdulmohsin
Rodolphe Jenatton Lucas Beyer Michael Tschannen Anurag Arnab Xiao Wang
Carlos Riquelme Matthias Minderer Joan Puigcerver Utku Evci Manoj Kumar
Sjoerd van Steenkiste Gamaleldin F. Elsayed Aravindh Mahendran Fisher Yu
Avital Oliver Fantine Huot Jasmijn Bastings Mark Patrick Collier Alexey A. Gritsenko
Vighnesh Birodkar Cristina Vasconcelos Yi Tay Thomas Mensink Alexander Kolesnikov
Filip Pavetić Dustin Tran Thomas Kipf Mario Lučić Xiaohua Zhai Daniel Keysers
Jeremiah Harmsen Neil Houlsby*
Google Research

Abstract

The scaling of Transformers has driven breakthrough capabilities for language models. At present, the largest large language models (LLMs) contain upwards of 100B parameters. Vision Transformers (ViT) have introduced the same architecture to image and video modelling, but these have not yet been successfully scaled to nearly the same degree; the largest dense ViT contains 4B parameters (Chen et al., 2022). We present a recipe for highly efficient and stable training of a 22B-parameter ViT (ViT-22B) and perform a wide variety of experiments on the resulting model. When evaluated on downstream tasks (often with a lightweight linear model on frozen features), ViT-22B demonstrates increasing performance with scale. We further observe other interesting benefits of scale, including an improved tradeoff between fairness and performance, state-of-the-art alignment to human visual perception in terms of shape/texture bias, and improved robustness. ViT-22B demonstrates the potential for "LLM-like" scaling in vision, and provides key steps towards getting there.

Language models(LM) can be scaled to over 100B parameters. Largest ViT model contains 4B parameters by far.

Both use almost same transformer. Why can't ViT models be scaled to more billions parameters?

The paper presents the largest dense 22B ViT model(to date) along with its training recipes.

ViT - Follow-up Works

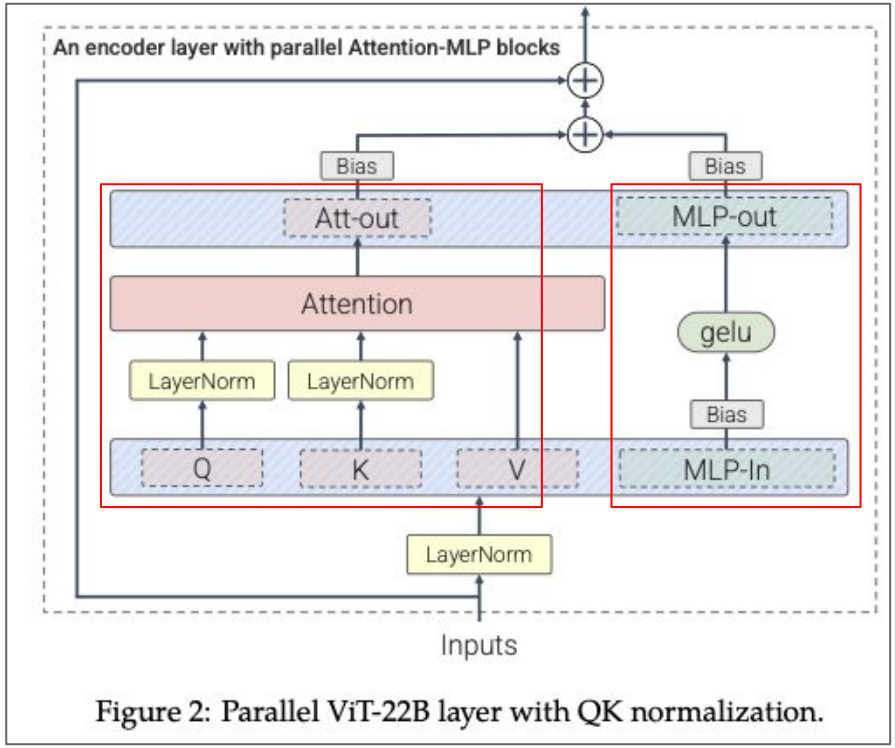


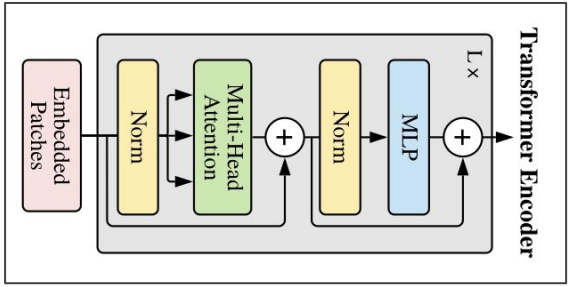
Figure 2: Parallel ViT-22B layer with QK normalization.

$$y' = \text{LayerNorm}(x),$$
$$y = x + \text{MLP}(y') + \text{Attention}(y').$$

ViT-22B is roughly an original ViT with three main architectural changes.

- **Parallel layers:** Instead of arranging Attention and MLPs blocks in series, ViT-22B applies them in parallel.

Original ViT



Dehghani et al., "Scaling Vision Transformers to 22 Billion Parameters, 2023

ViT - Follow-up Works

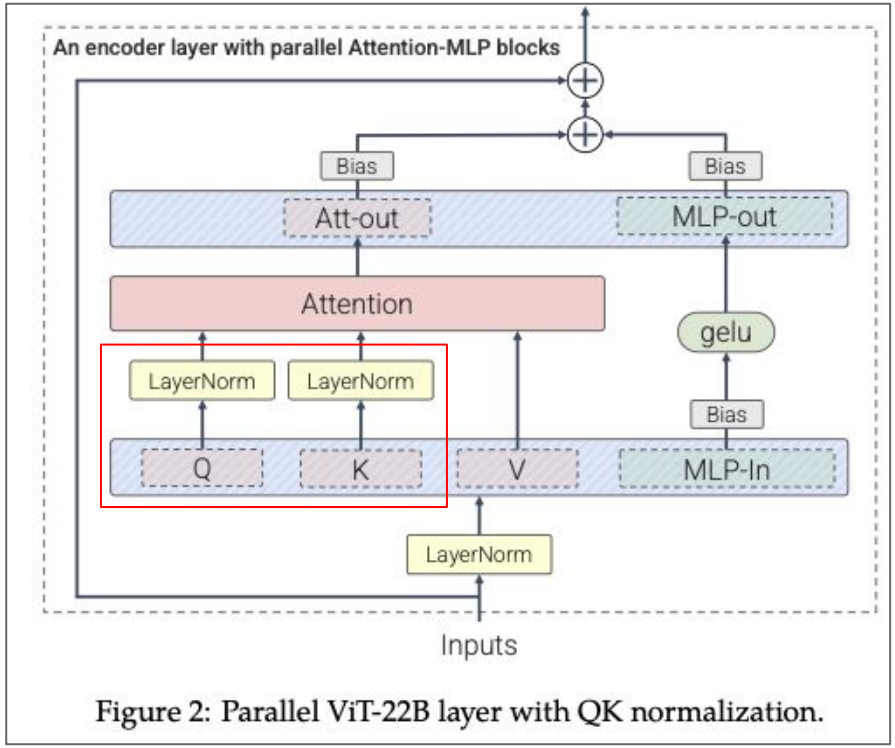


Figure 2: Parallel ViT-22B layer with QK normalization.

$$y' = \text{LayerNorm}(x),$$
$$y = x + \text{MLP}(y') + \text{Attention}(y').$$

ViT-22B is roughly an original ViT with three main architectural changes.

- **Parallel layers:** Instead of arranging Attention and MLPs blocks in series, ViT-22B applies them in parallel.
- **Applying normalization(LayerNorm) to queries Q and keys K**: when attention logits are extremely large, attention weights can vanish. Q and K are fed through LayerNorm before dot product attention to stabilize the training.

$$\text{softmax}\left[\frac{1}{\sqrt{d}}\text{LN}(XW^Q)(\text{LN}(XW^K))^T\right]$$

Dehghani et al., "Scaling Vision Transformers to 22 Billion Parameters, 2023

ViT - Follow-up Works

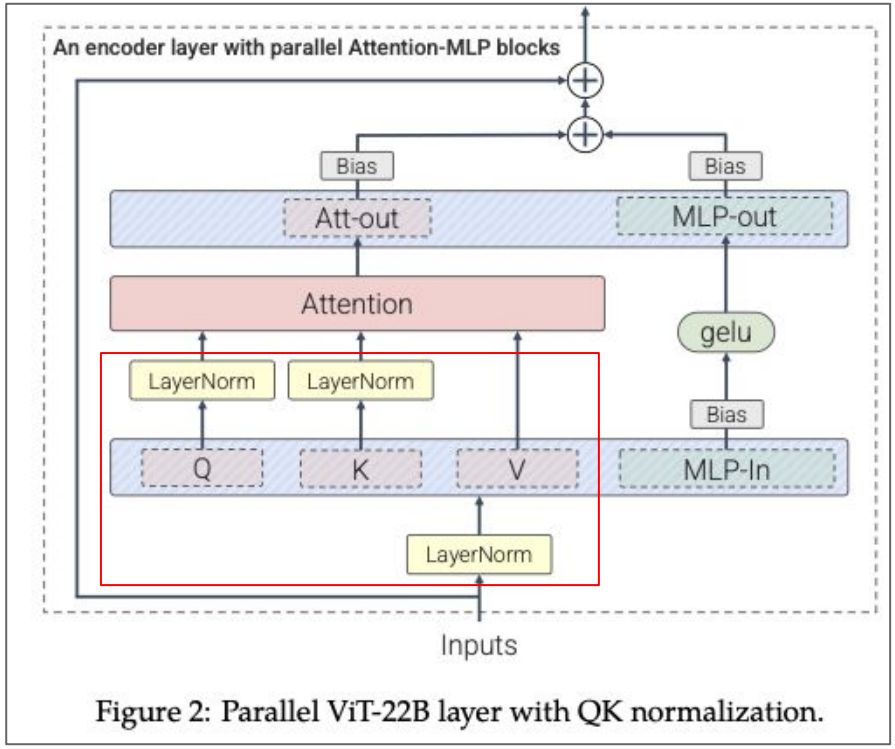


Figure 2: Parallel ViT-22B layer with QK normalization.

$$y' = \text{LayerNorm}(x),$$
$$y = x + \text{MLP}(y') + \text{Attention}(y').$$

ViT-22B is roughly an original ViT with three main architectural changes.

- **Parallel layers:** Instead of arranging Attention and MLPs blocks in series, ViT-22B applies them in parallel.
- **Applying normalization(LayerNorm) to queries Q and keys K**: when attention logits are extremely large, attention weights can vanish. Q and K are fed through LayerNorm before dot product attention to stabilize the training.
- **Omitting bias terms in QKV projections and LayerNorms**: Following previous novel works like PaLM, ViT-22B suppresses biases terms in QKV linear projections layers and LayerNorms but keeps them in MLP in & out and Att-out.

Dehghani et al., "Scaling Vision Transformers to 22 Billion Parameters, 2023

ViT - Follow-up Works

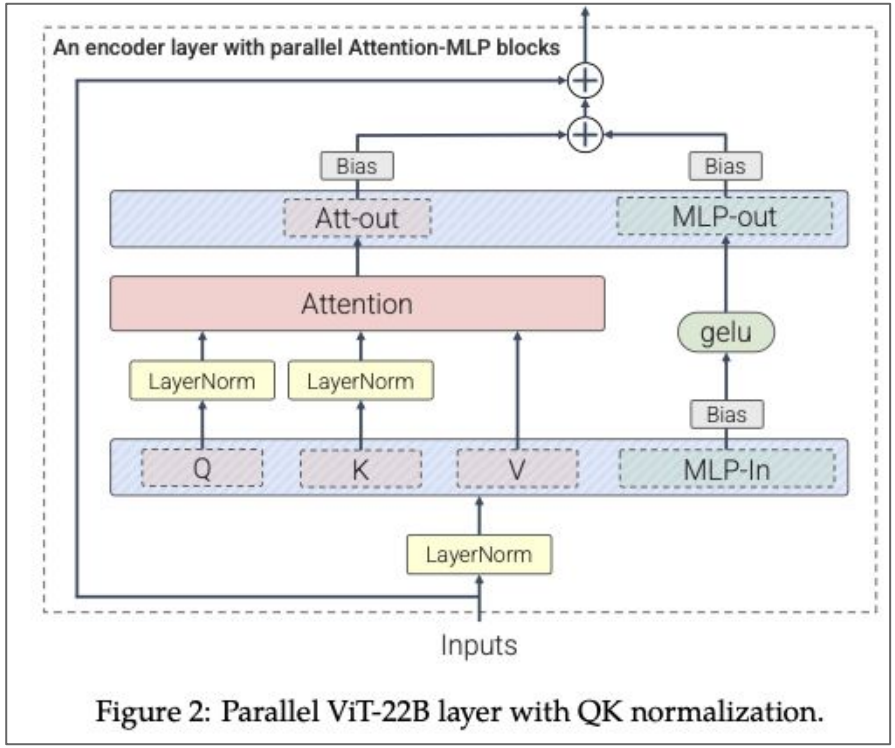


Figure 2: Parallel ViT-22B layer with QK normalization.

$$y' = \text{LayerNorm}(x),$$
$$y = x + \text{MLP}(y') + \text{Attention}(y').$$

ViT-22B improves training and hardware stability, achieves SOTA on several vision benchmarks, and claims to be more aligned with humans for shape and texture bias and more robust compared to existing models based on ViT.

Dehghani et al., "Scaling Vision Transformers to 22 Billion Parameters, 2023

End of the video

Image: Google AI blog

Thank you for your visual attention!