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

Kishore Kumar Alajangi, James Dwyer, Michael Fielder, Joseph Rivera
CAP 6412
Spring 2022

Outline

1. Introduction
2. Objectives
3. Related Work
4. Method
5. Experiments
6. Conclusion

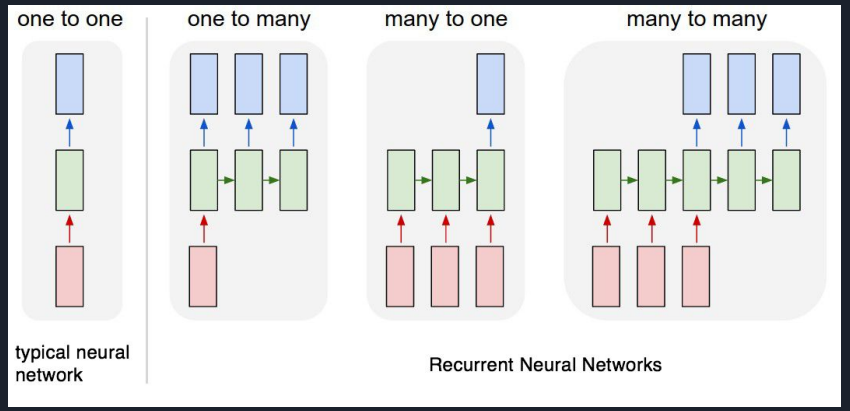
Introduction

- Using transformers for Natural Language Processing (NLP) involves pre-training and fine-tuning
- Previous use of transformers for computer vision lacked scalability and generalization
- Pure transformer applied to image patches – similar to NLP, but with patches instead of words
- Transformer pre-trained on huge (14 million to 300 million images) datasets

Objectives

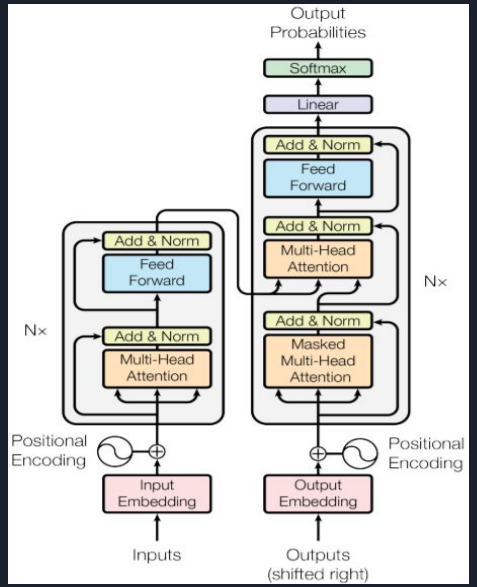
- Apply transformers used for Natural Language Processing to computer vision problems - Vision Transformer (ViT)
- Utilize transformers without relying on CNNs
- Demonstrate results with a number of models and datasets
- Achieve comparable or improved results with transformers compared to CNNs
- Lower the computational complexity

Related Work



Andre Karpathy (http://karpathy.github.io/2015/05/21/rnn-effectiveness/)

- Recurrent Neural Networks first handled sequence data, then LSTMs
  - Slow training
  - limited memory i.e. small window size
- Transformers in NLP Tasks
  - *Attention Is All You Need* introduced the transformer in 2017
  - Transformers allow parallel training of longer sequences
  - BERT and GPT serve as the inspiration from NLP tasks



Attention Is All You Need (NuerIPS (NIPS), 2017

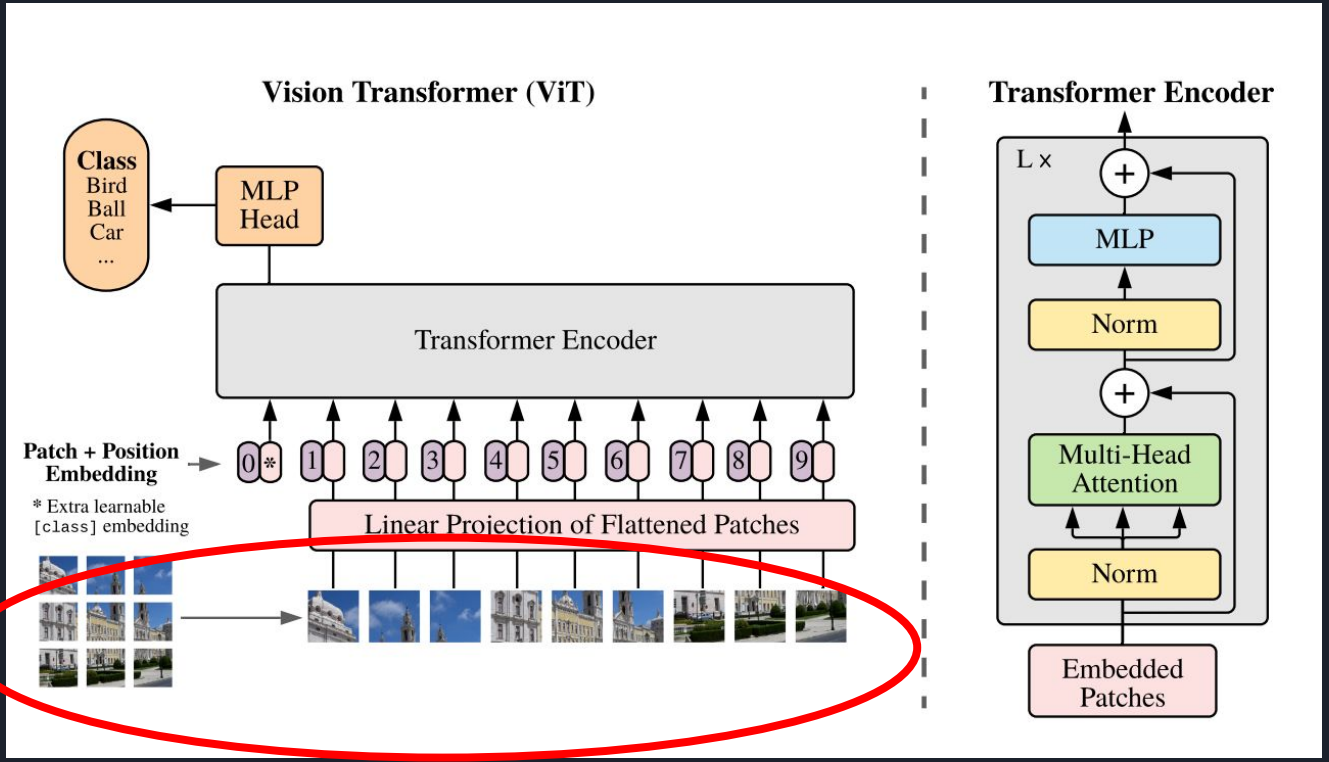
Comparison to State of the Art on ImageNet

- Convolutional Neural Networks dominated classification tasks until recently
- Transformers have recently overtaken state of the art with caveats like extra data or hybrid models

Model Category	Model Name	ImageNet Top-1 Accuracy	Reference
Transformer	ViT-H/14	88.55%*	"AN IMAGE IS WORTH 16X16 WORDS…" (ICLR, 2021)
CNN	EfficientNet-L2	90.2%*	"Meta Psuedo Labels" (EfficientNet-L2) (CVPR, 2021)
Transformer	ViT-G/14	90.45%*	"Scaling Vision Transformers (ArXiv, 2021)
Hybrid CNN-Transformer	CoAtNet-7	90.88%*	"CoAtNet: Marrying Convolution and Attention for All Data Sizes" (NuerIPS, 2021)

*trained with additional data from Google's Internal Datasets (JFT-300M & JFT-3B)

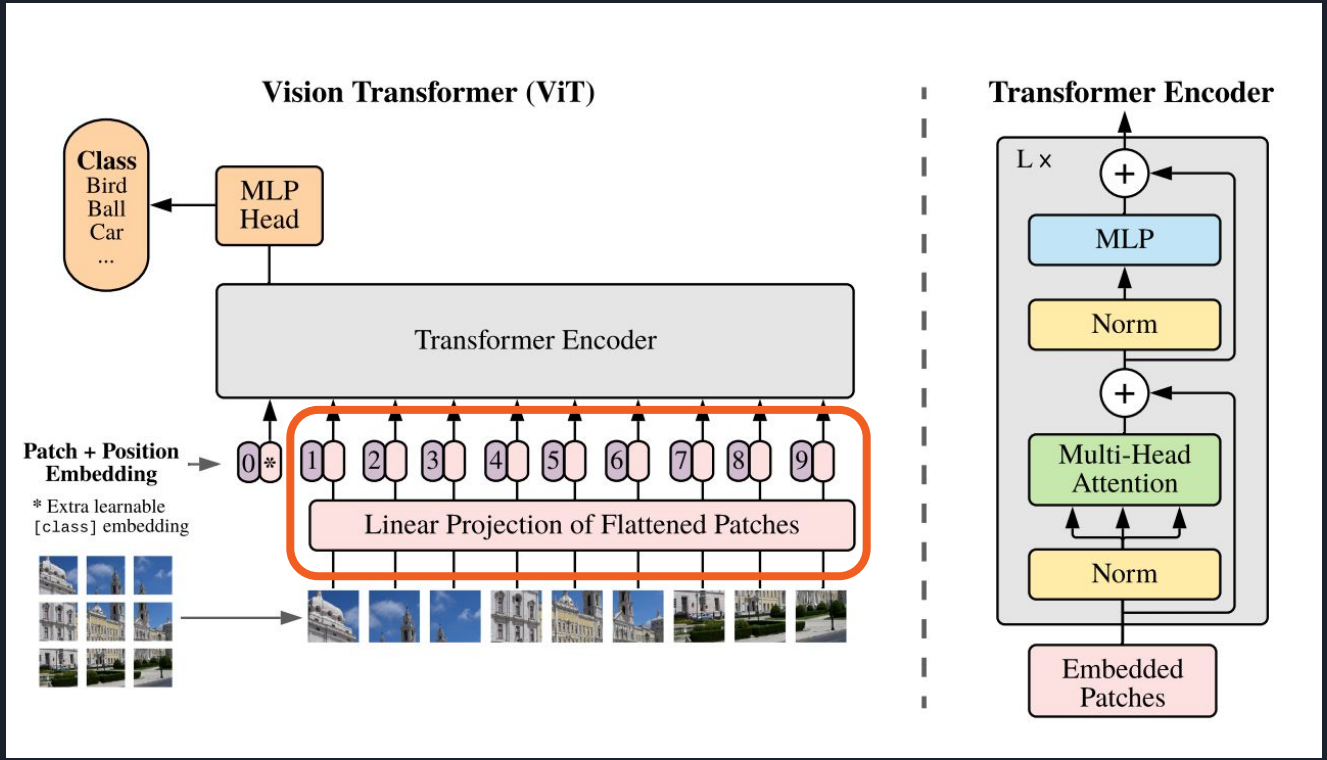
Method

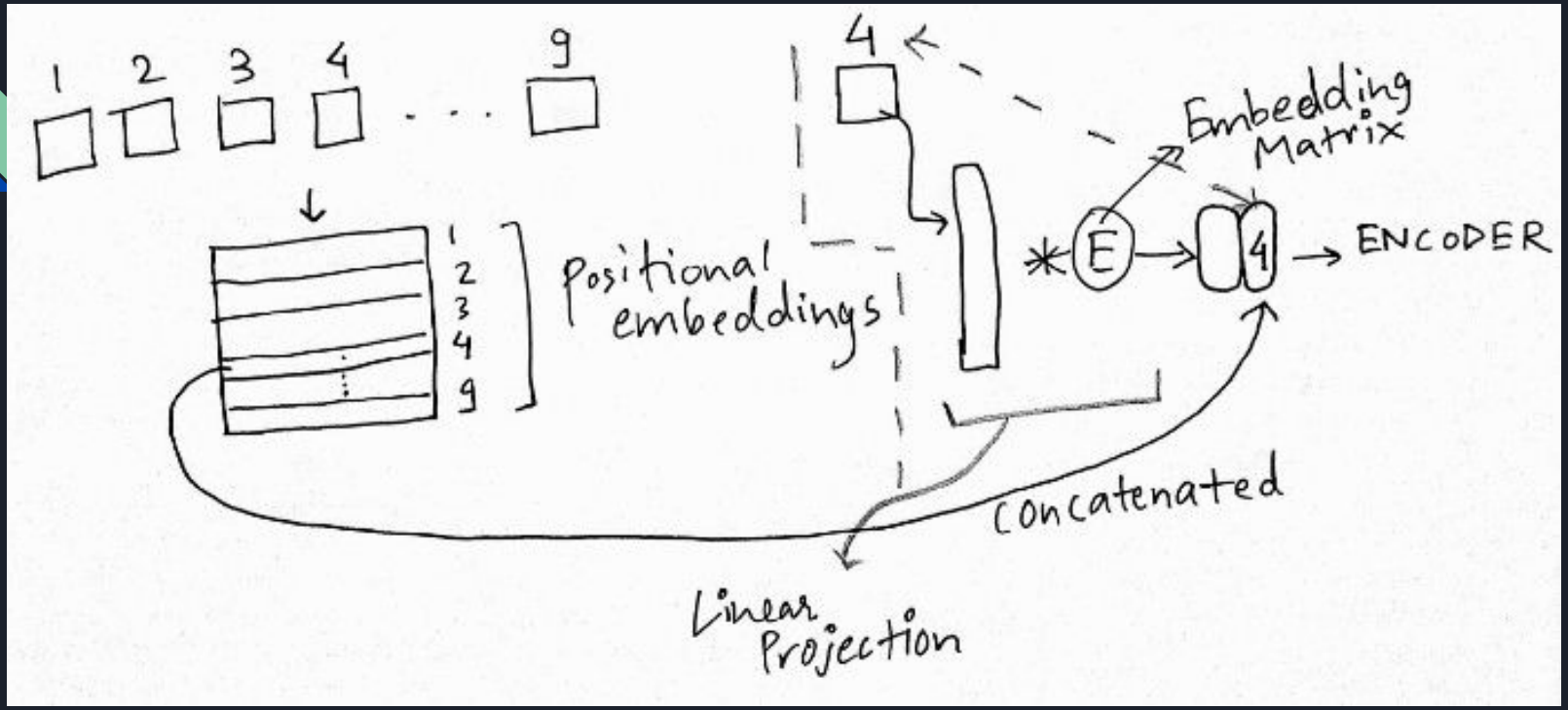

Vision Transformer (ViT)
Class
Bird
Ball
Car
...
MLP Head
Transformer Encoder
Patch + Position Embedding
* Extra learnable [class] embedding
0 * 1 2 3 4 5 6 7 8 9
Linear Projection of Flattened Patches
Transformer Encoder
L x
MLP
Norm
Multi-Head Attention
Norm
Embedded Patches


1 2 3
4 5 6
7 8 9


1 2 3 4 5 6 7 8 9

Method


Vision Transformer (ViT)
Class
Bird
Ball
Car
...
MLP Head
Transformer Encoder
Patch + Position Embedding
* Extra learnable [class] embedding
0 * 1 2 3 4 5 6 7 8 9
Linear Projection of Flattened Patches
Transformer Encoder
L x
MLP
Norm
Multi-Head Attention
Norm
Embedded Patches

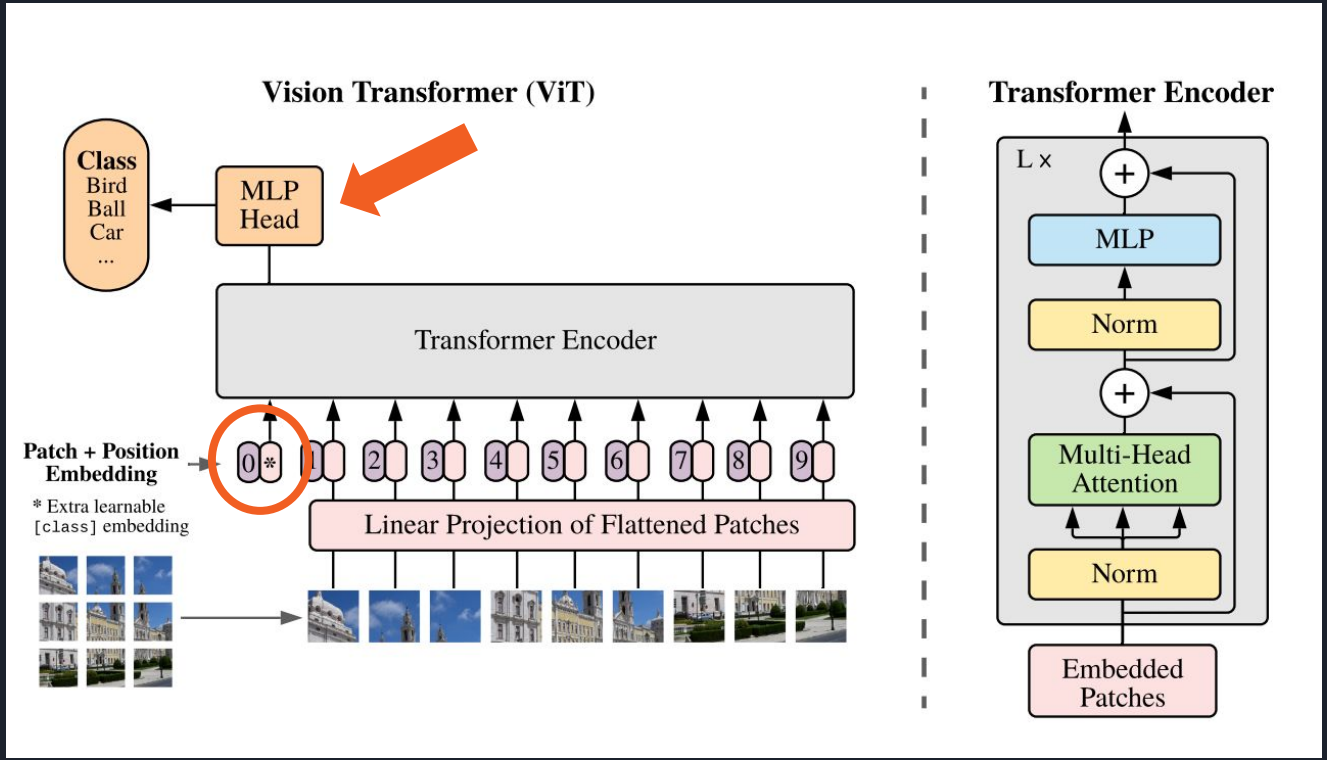


$$\mathbf{x}_p \in \mathbb{R}^{N \times (P^2 \cdot C)}$$

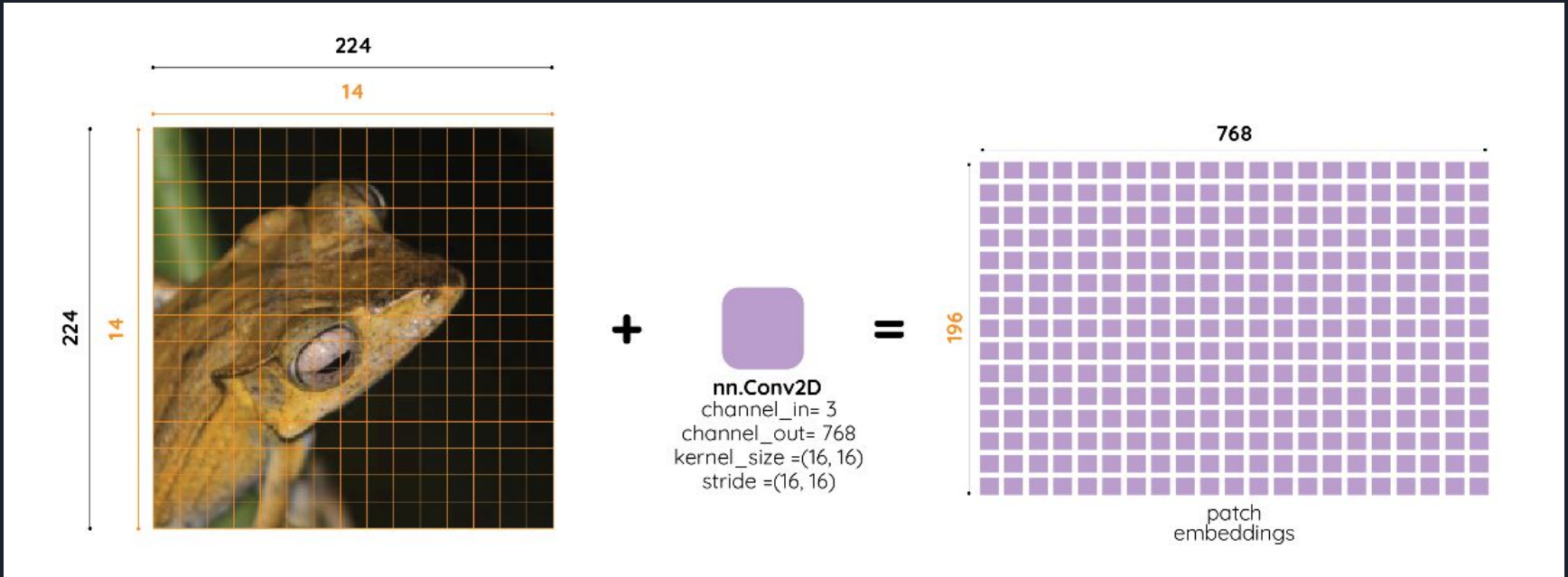
Source: https://medium.com/analytics-vidhya/vision-transformers-bye-bye-convolutions-e929d022e4ab

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

Method

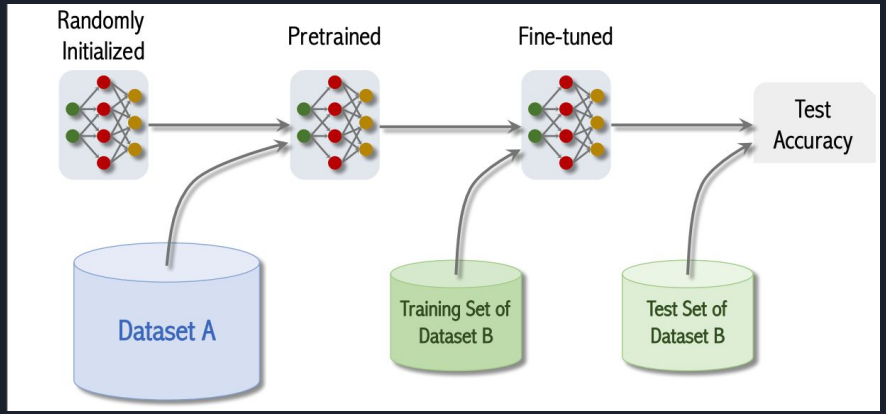

Vision Transformer (ViT)
Class
Bird
Ball
Car
...
MLP Head
Transformer Encoder
Patch + Position Embedding
* Extra learnable [class] embedding
0 * 1 2 3 4 5 6 7 8 9
Linear Projection of Flattened Patches
Transformer Encoder
L x
MLP
Norm
Multi-Head Attention
Norm
Embedded Patches

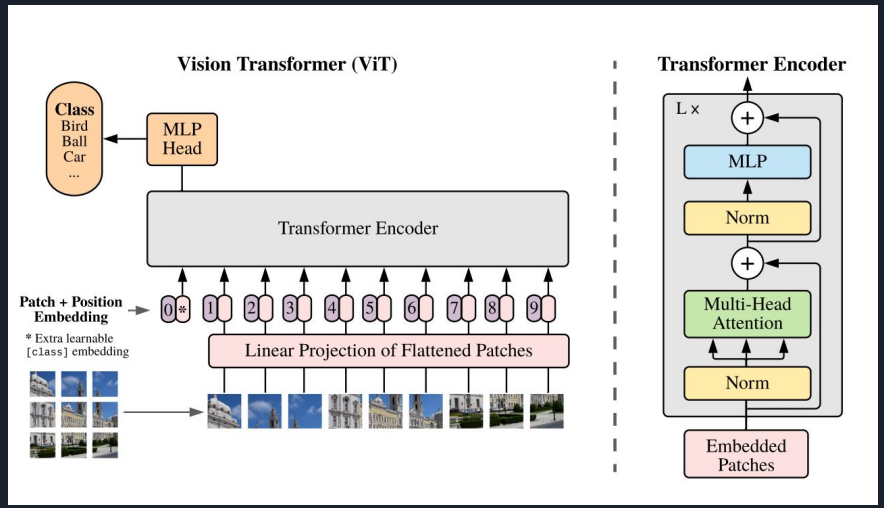
Hybrid Architecture



Source: https://amaarora.github.io/2021/01/18/ViT.html

Method: Fine-Tuning and Higher Resolution





Source: https://www.youtube.com/watch?v=HZ4j_U3FC94

Experiments: Setup: Explanation of Datasets and Model Variants - Datasets

- Pre-training Datasets
  - ILSVRC-2012 ImageNet (i.e., ImageNet), ~1000 classes, ~1.3 million images
  - ImageNet-21k, ~21000 classes, ~14 million images
  - JFT, ~18000 classes, ~303 million images

- Fine-tuning Datasets
  - ImageNet - both original validation labels and Reassessed Labels (ReaL)
  - CIFAR-10/100, ~60000 images each
  - Oxford-IIIT Pets, ~7400 images
  - Oxford Flowers-102, ~7100 images
  - VTAB: natural, specialized, structured
- Preprocessing
  - Pre-training: image cropped, random horizontal mirroring, resize to 224x224
  - Fine-tuning: Images resized to 448x448, random crop of 384x384
  - Random horizontal flips

Experiments: Setup: Explanation of Datasets and Model Variants - Model Variants

- ViT Model Variants
  - Base - 32x32 and 16x16
  - Large - 32x32 and 16x16
  - Huge - 14x14

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

Table 1: Details of Vision Transformer model variants.

- CNNs (ResNet (BiT))
  - ResNet(50, 101, 152, 200)
  - Batch Normalization replaced with Group Normalization
- Hybrids
  - Use ResNet50 output from stage 4
  - Output (feature maps) fed to ViT with patch size of 1 pixel

Experiments: Setup: Explanation of Datasets and Model Variants - Training and Fine Tuning

- Training
  - Uses Adam optimizer with β_1 = 0.9 and β_2 = 0.999
  - Batch size 4096
  - Weight decay 0.1
  - Linear learning rate warmup and decay

- Fine-tuning
  - SGD with momentum
  - Batch size 512
  - Cosine learning rate decay
  - No weight decay

	ResNet50		ResNet152x2	
Dataset	Adam	SGD	Adam	SGD
ImageNet	77.54	78.24	84.97	84.37
CIFAR10	97.67	97.46	99.06	99.07
CIFAR100	86.07	85.17	92.05	91.06
Oxford-IIIT Pets	91.11	91.00	95.37	94.79
Oxford Flowers-102	94.26	92.06	98.62	99.32
Average	89.33	88.79	94.01	93.72

Experiments: Setup: Explanation of Datasets and Model Variants - Metrics

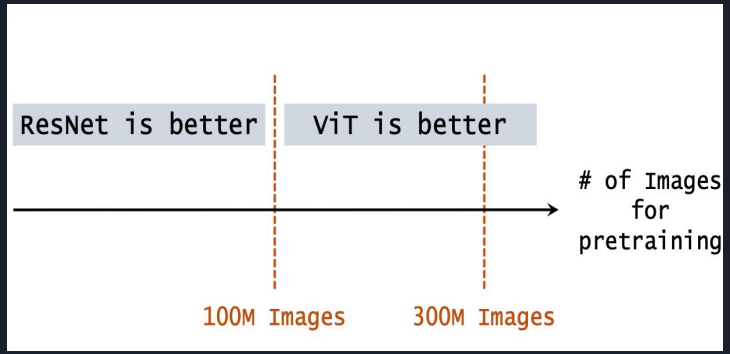
- Metrics
  - Fine-tuning accuracy: results after fine-tuning
  - Few-shot accuracy: results using a small subset of data (used for quick evaluations)

Experiments: Pre-training Data Requirements

	# of Images	# of Classes
ImageNet (Small)	1.3 Million	1 Thousand
ImageNet-21K (Medium)	14 Million	21 Thousand
JFT (Big)	300 Million	18 Thousand

Experiments: Scaling Study

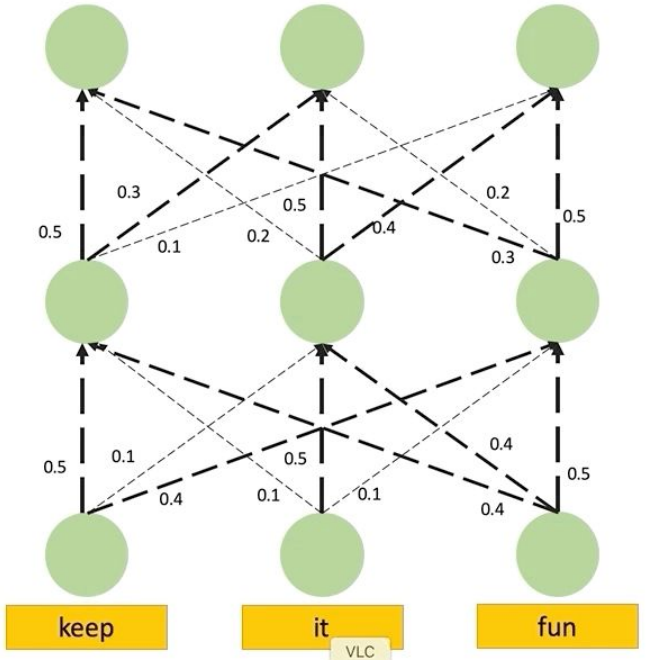
	Ours-JFT (ViT-H/14)	Ours-JFT (ViT-L/16)	Ours-I21k (ViT-L/16)	BiT-L (ResNet152x4)	Noisy Student (EfficientNet-L2)
ImageNet	**88.55** ± 0.04	87.76 ± 0.03	85.30 ± 0.02	87.54 ± 0.02	88.4/88.5*
ImageNet ReaL	**90.72** ± 0.05	90.54 ± 0.03	88.62 ± 0.05	90.54	90.55
CIFAR-10	**99.50** ± 0.06	99.42 ± 0.03	99.15 ± 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



Source: https://www.youtube.com/watch?v=HZ4j_U3FC94

Inspecting Vision Transformers

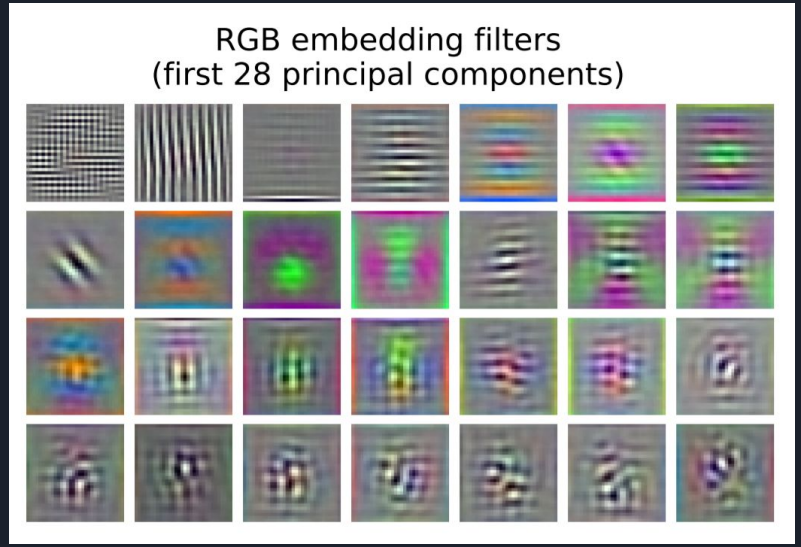
Attention Rollout (Samira Abnar, 2020)



Visualization of Embedding Filters

What does the model learn?

- Calculate PCA on the embedding E i.e. create new features that summarize information from all feature vectors
- The learned filter look similar to low level CNN feature maps (left)
- The transformer is learning visual information in a familiar way to CNNs e.g. lines for edge detection in various orientations



What Does The Position Embedding Learn?

- Recall the Position Embedding is a matrix with N+1 Rows by D features

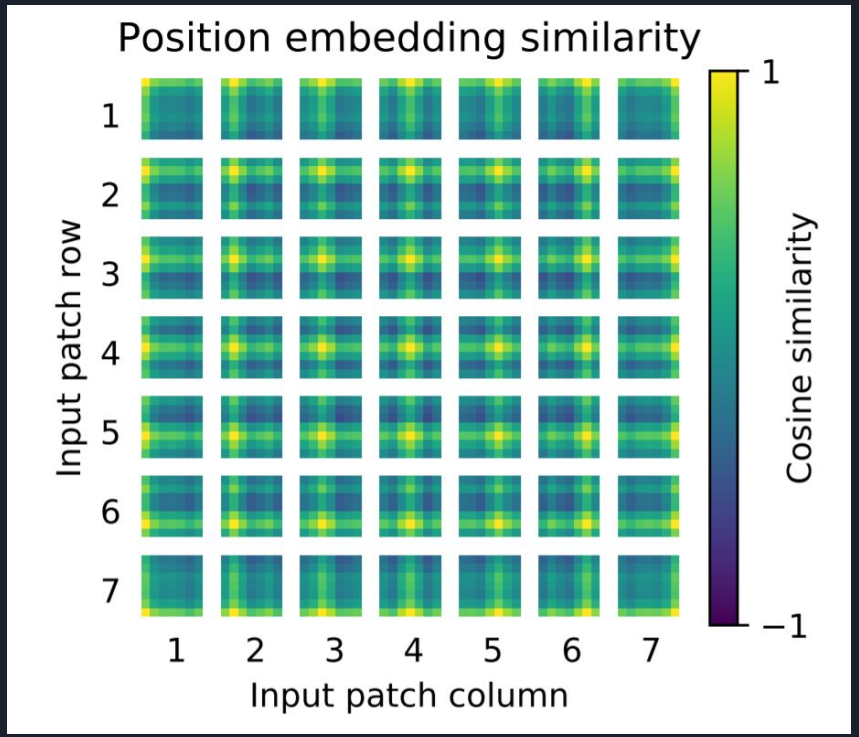
$$\mathbf{E}_{pos} \in \mathbb{R}^{(N+1)\times D}$$

- For all patches N
  - Compute similarity between each row vector and itself

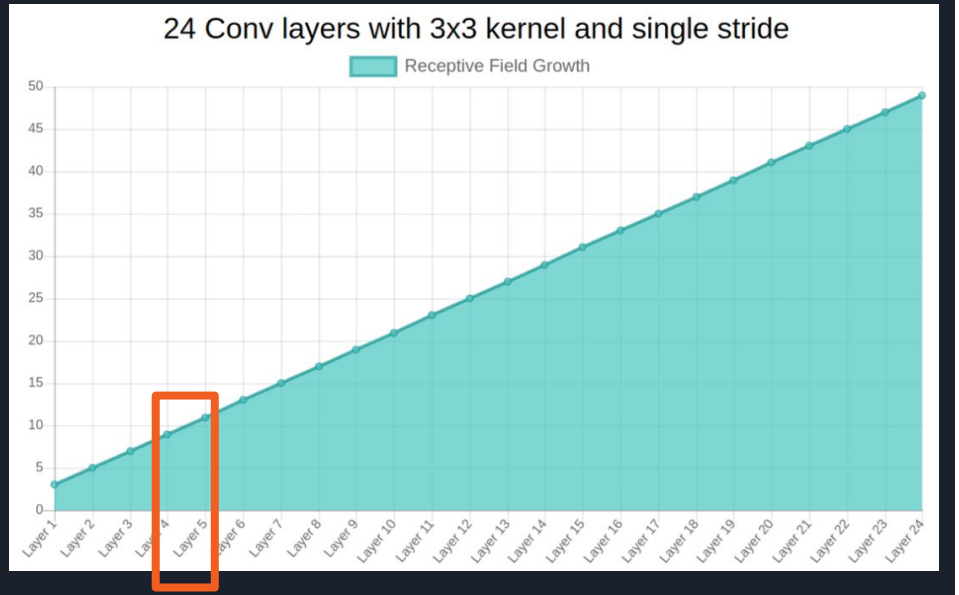
$$similarity(A,B) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}$$

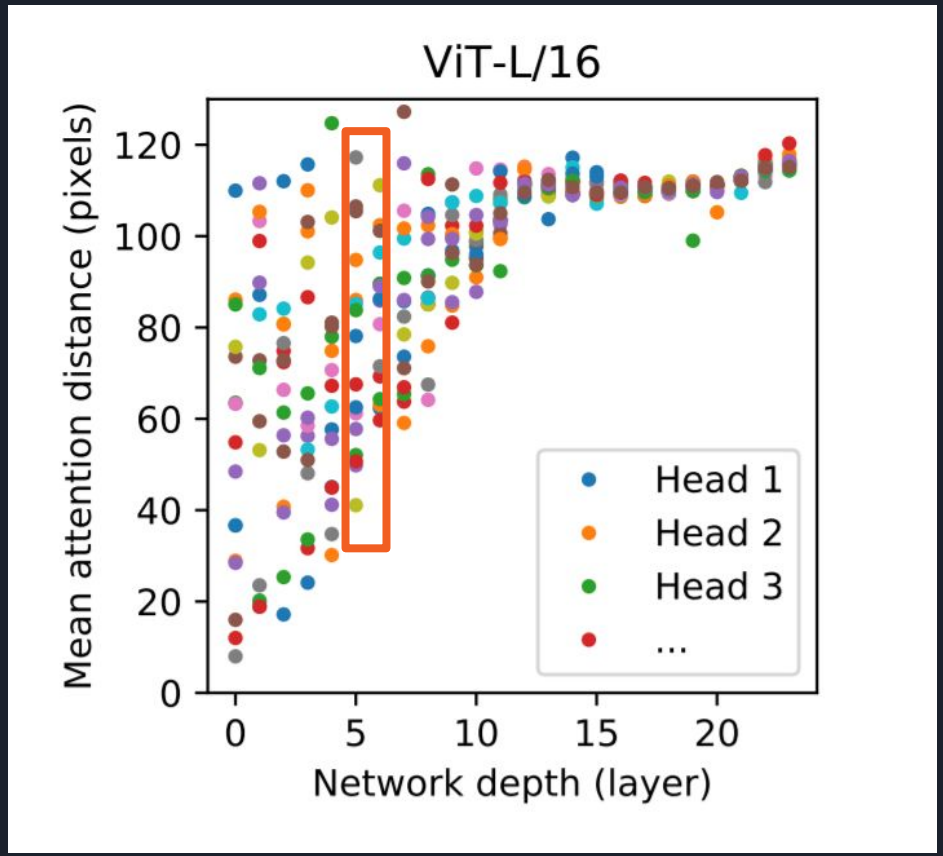
- Plotting the similarities show that the embedding is highly correlated between the rows and columns of its original 2-D position.

ViT-L/32

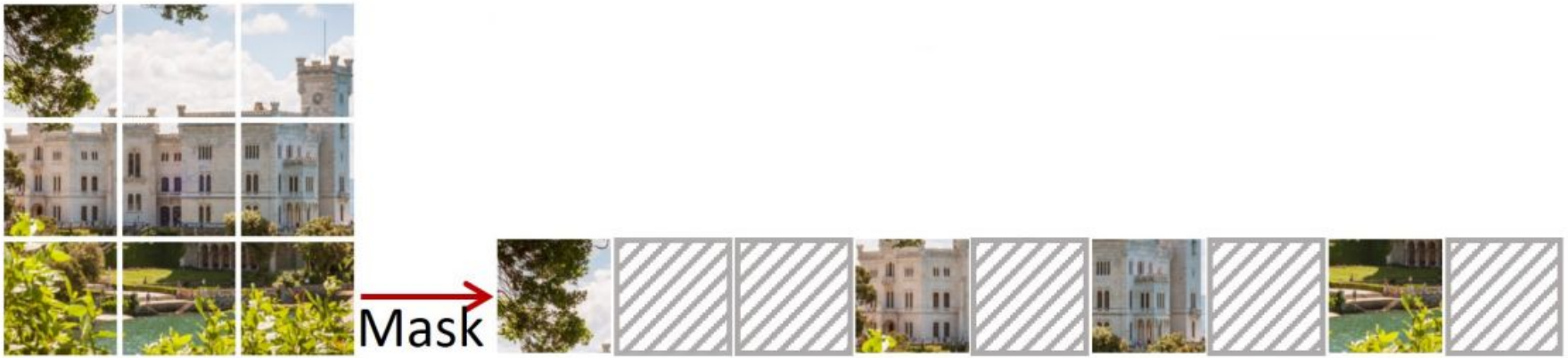


Attention Distance Over Layer Depth





Self-Supervision



Source: https://arxiv.org/pdf/2111.09886.pdf

Conclusion - Advantages

- Self-Attention
- Parallelized training over other sequence models
- Can learn spatial mapping of where patches were pulled
- Attention is widely spread across pixels even at early layers, revealing global understanding
- Diagrams are clear and well-made

Conclusion - Disadvantages

- Trained on additional data that is internal to Google
- Extremely large networks, not suited for deployment on edge computing

Computing Z_0 - *Output of the Embedding*