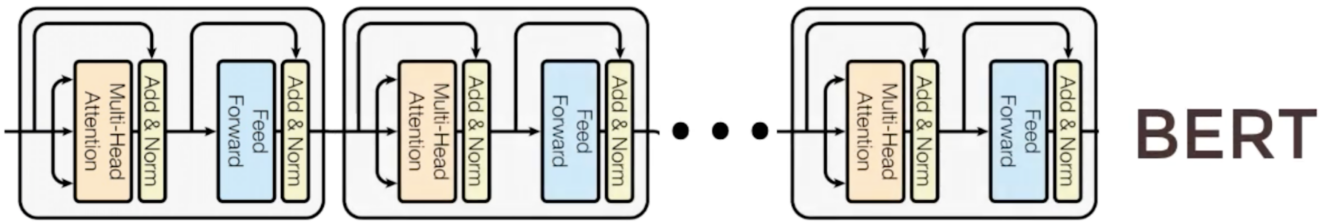
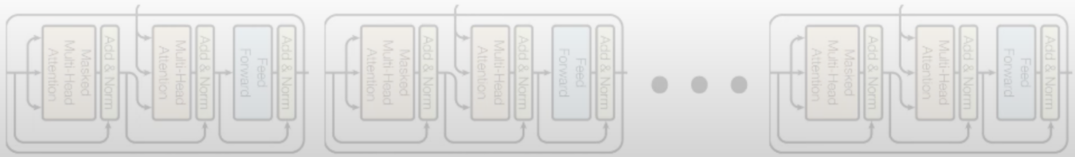
Transformer Flow

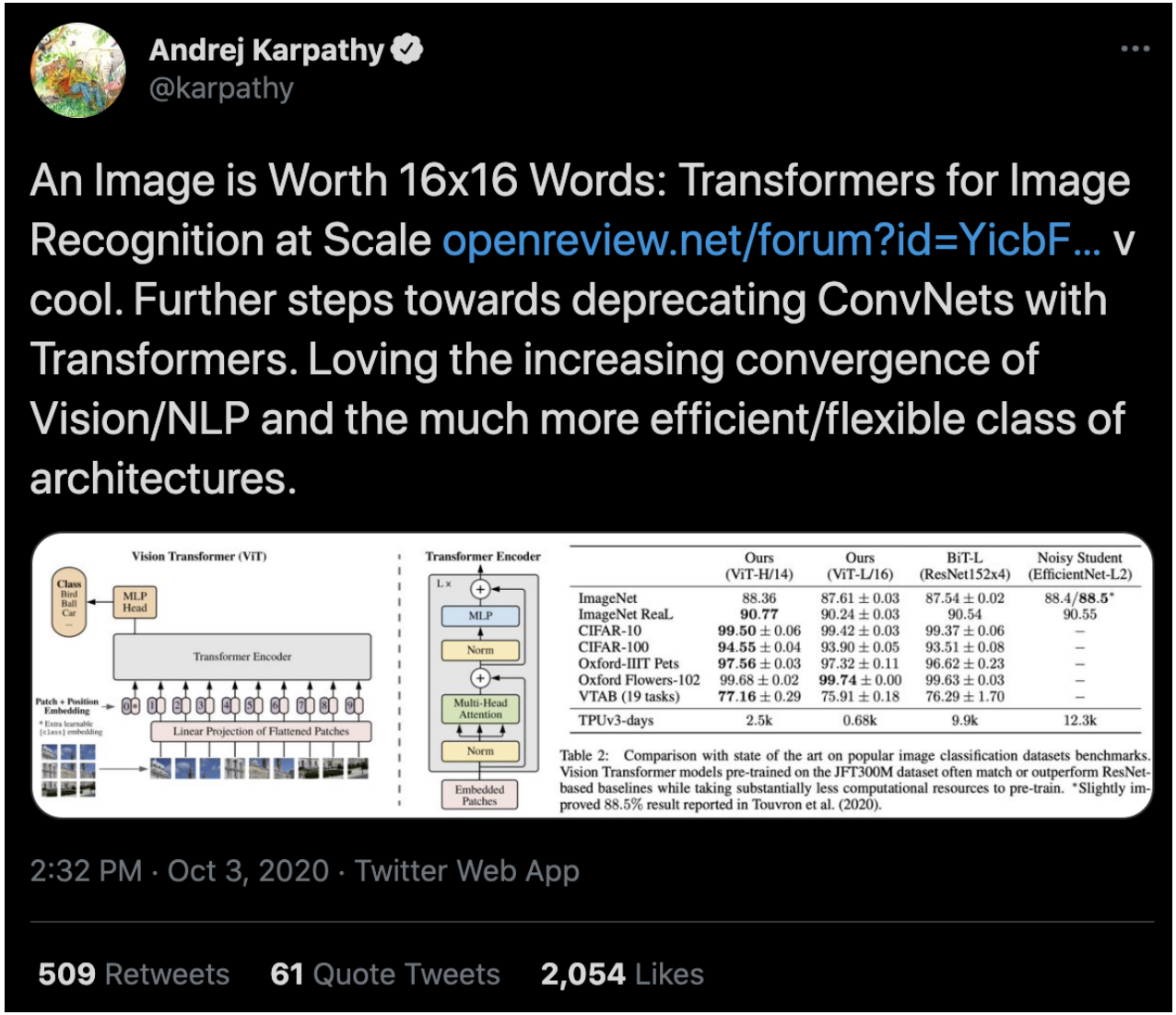


Bidirectional Encoder Representation from Transformers



- We stack the encoders and we get BERT

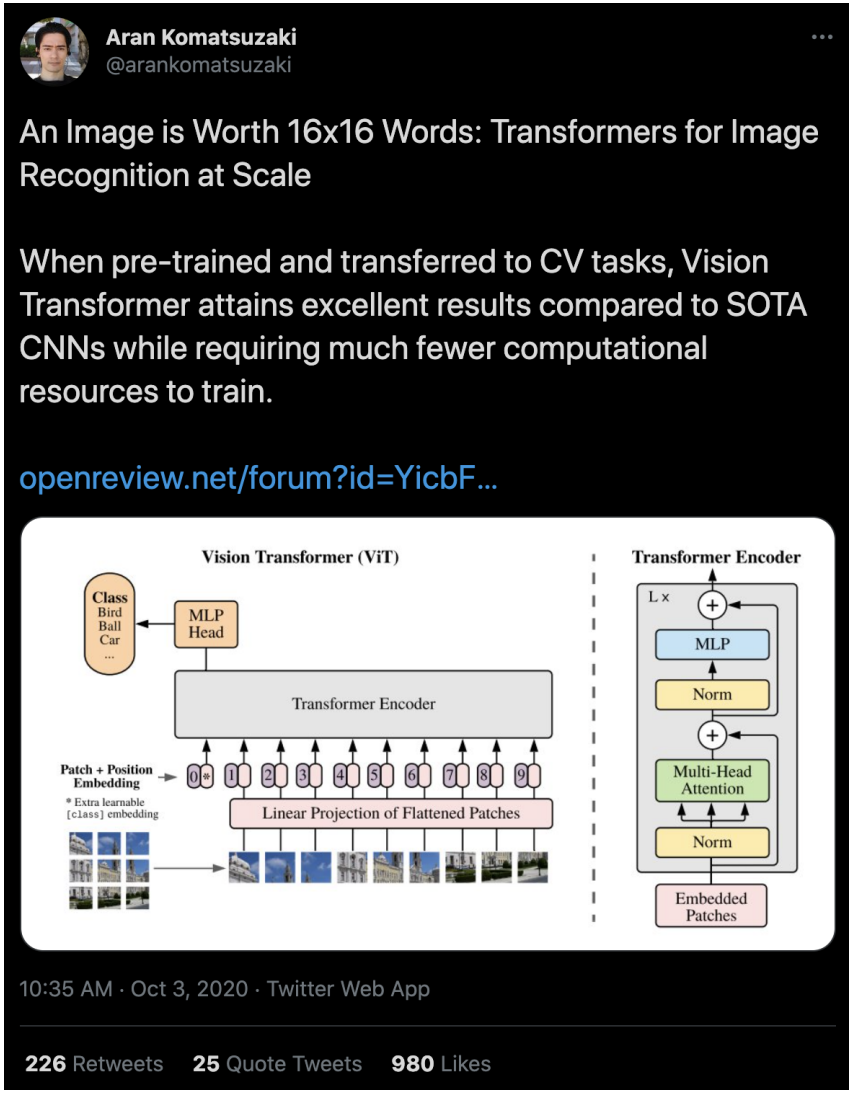
Transformers for Computer Vision



	Ours (ViT-H/14)	Ours (ViT-L/16)	BiT-L (ResNet152x4)	Noisy Student (EfficientNet-L2)
ImageNet	88.36	87.61 ± 0.03	87.54 ± 0.02	88.4/**88.5***
ImageNet ReaL	**90.77**	90.24 ± 0.03	90.54	90.55
CIFAR-10	**99.50** ± 0.06	99.42 ± 0.03	99.37 ± 0.06	–
CIFAR-100	**94.55** ± 0.04	93.90 ± 0.05	93.51 ± 0.08	–
Oxford-IIIT Pets	**97.56** ± 0.03	97.32 ± 0.11	96.62 ± 0.23	–
Oxford Flowers-102	99.68 ± 0.02	**99.74** ± 0.00	99.63 ± 0.03	–
VTAB (19 tasks)	**77.16** ± 0.29	75.91 ± 0.18	76.29 ± 1.70	–
TPUv3-days	2.5k	0.68k	9.9k	12.3k

Table 2: Comparison with state of the art on popular image classification datasets benchmarks. Vision Transformer models pre-trained on the JFT300M dataset often match or outperform ResNet-based baselines while taking substantially less computational resources to pre-train. *Slightly improved 88.5% result reported in Touvron et al. (2020).

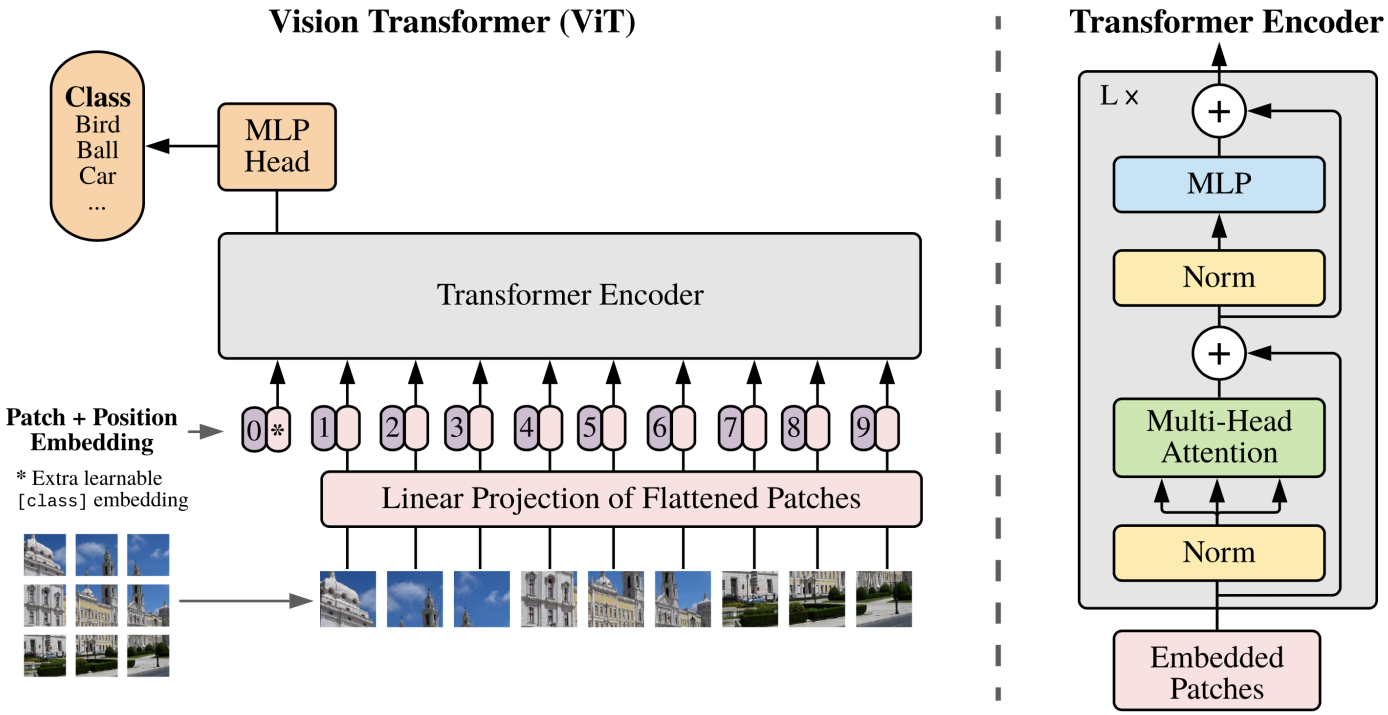
Transformers for Computer Vision



Early Attempts on Transformers for Computer Vision

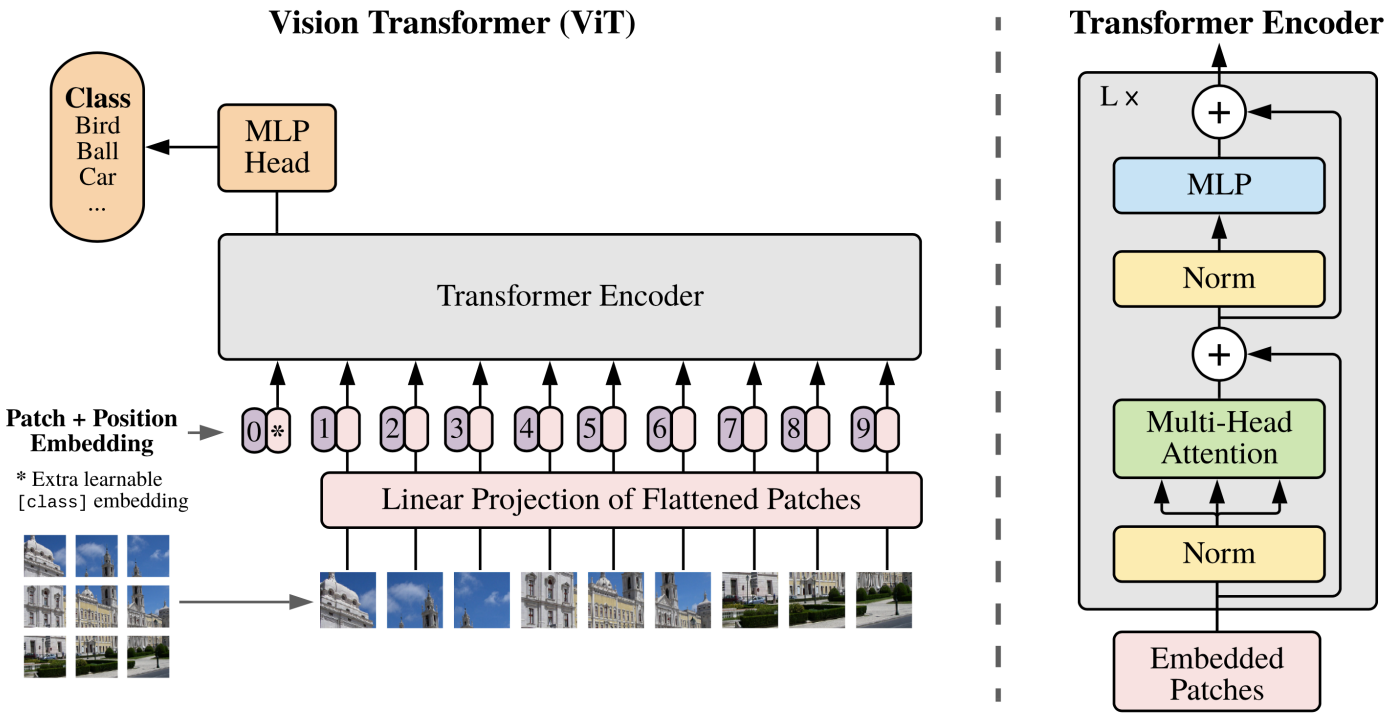
- Until recently, convolutional architectures remain dominant
  - Mahajan et al., 2018; Xie et al., 2020; Kolesnikov et al., 2020
- Combining CNN-like architectures with self-attention
  - Wang et al., 2018; Carion et al., 2020
- Replacing the convolutions entirely
  - Ramachandran et al., 2019; Wang et al., 2020
  - They did not scale effectively on modern hardware accelerators
    - Because they used specialized attention patterns

Overview of ViT (Vision Transformer)



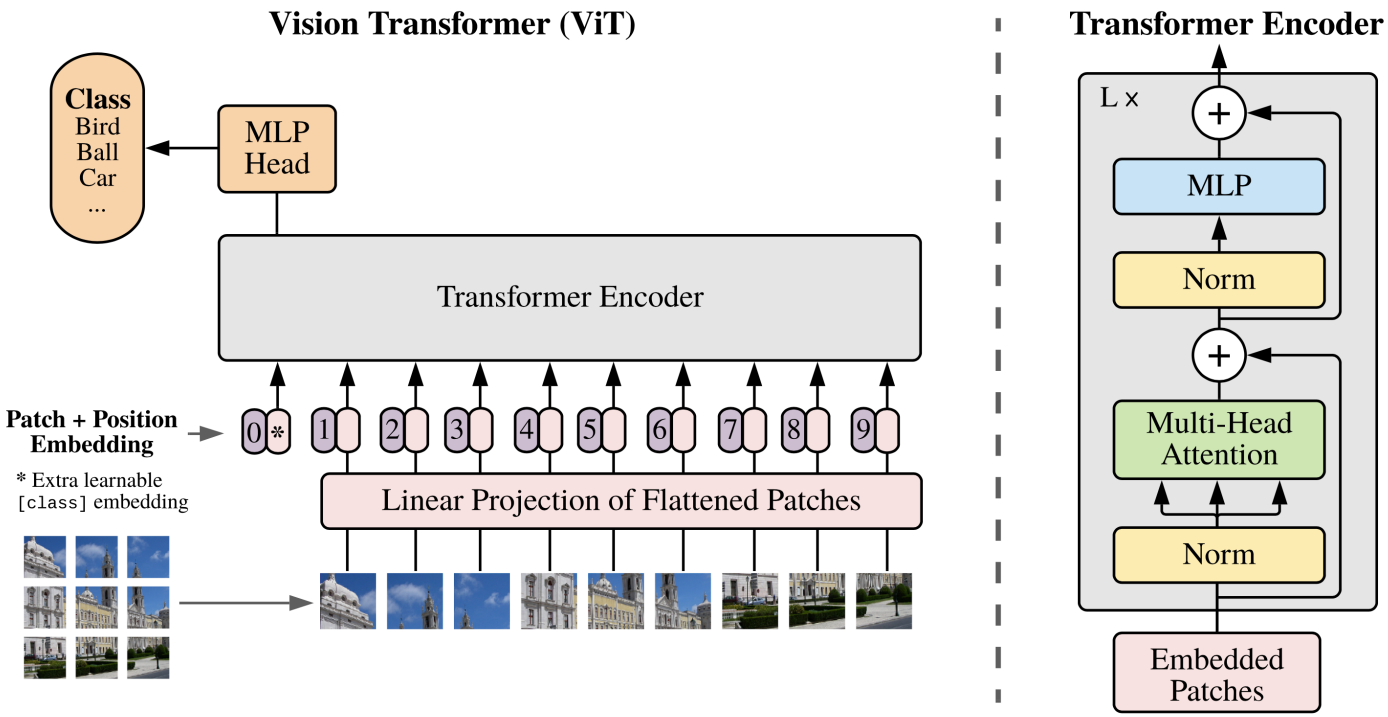
- Partition an image into fixed-size patches
- Linearly embed each of them
- Add position embeddings
- Feed the resulting sequence of vectors to a Transformer encoder

Overview of ViT (Vision Transformer)



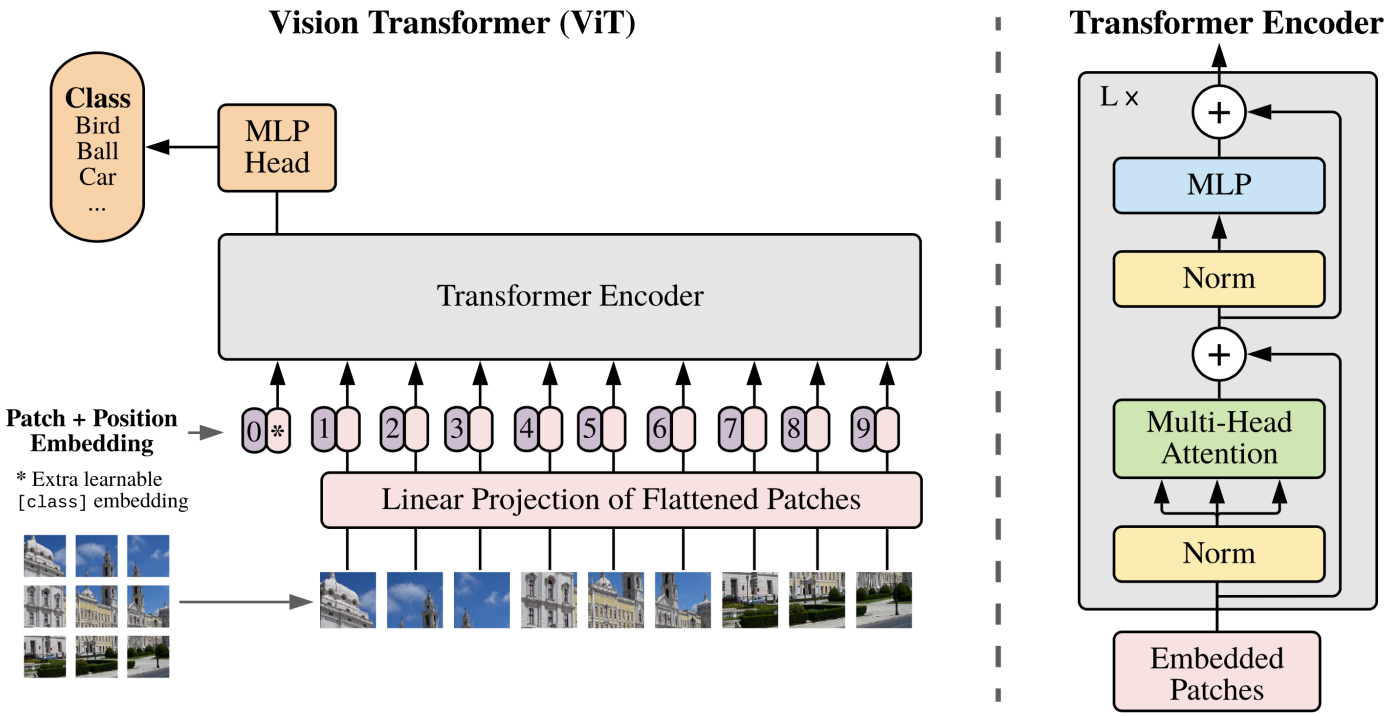
- To perform classification, they add an extra learnable "classification token" to the sequence
  - It is a standard approach

ViT (Vision Transformer) Input Processing



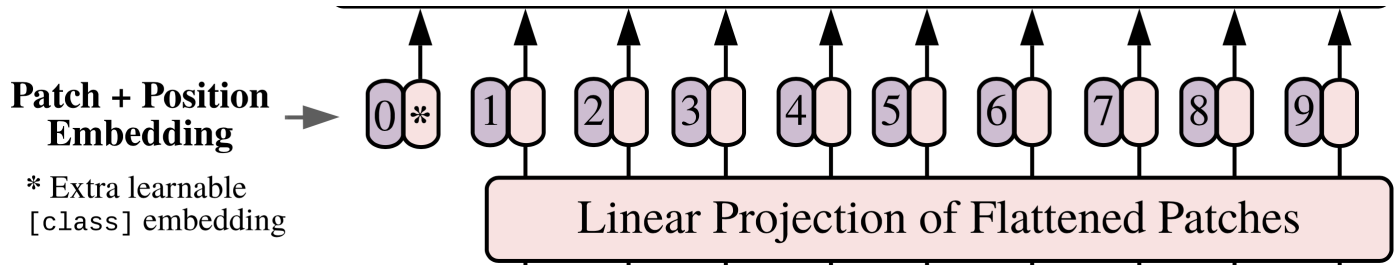
- The input of a standard NLP Transformer is a 1D sequence of token embeddings
  - To handle 2D images, we reshape the image $x \in R^{H \times W \times C}$ into a sequence of flattened 2D patches $x_p \in R^{N \times (P^2 C)}$
  - (H, W): resolution of the original image & C: number of channels
  - (P, P): resolution of an image patch & $N = HW/P^2$: number of patches
    - N is the effective input sequence length for the Transformer

ViT (Vision Transformer) Input Processing



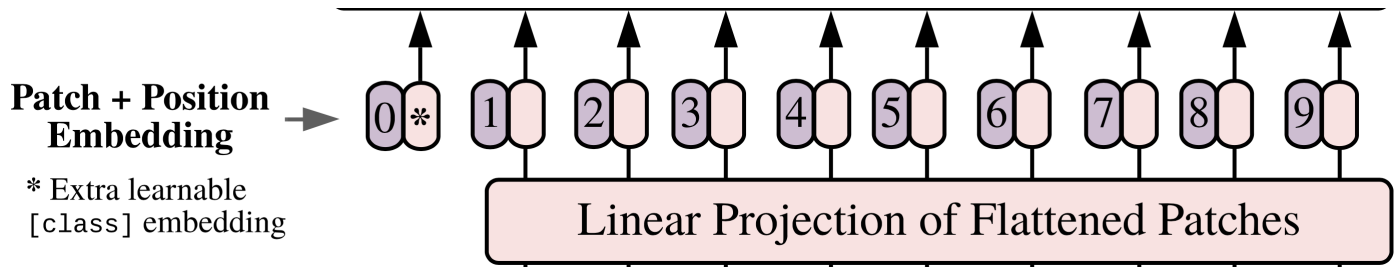
- ViT uses constant latent vector size D through all of its layers
  - They flatten the patches and map to D dimensions with a trainable linear projection
    - The output of this projection is called as the patch embeddings

Position Embedding



- Transformer does not have the position information
  - We need to use position embedding to add position information to the model
- The purple box is the position embedding
- The pink box is the patch embedding
  - The flattened patch after linear projection
- ViT puts position embedding and patch embedding together to give the model position information

Learnable Embedding



- The pink box with a star is not an image patch
  - It is a learnable embedding (denoted as x^*)
  - It is similar to the [class] token t^* in BERT
    - After encoder, [class] token t^* represents the whole sentence
  - Similarly, after encoder, x^* represents of the whole image
- Why BERT or ViT has this additional token t^* or patch x^*?
  - If we pick an existing image patch (x') or token (t') to represent the whole image or whole sentence, it will likely only represent x' or t'
  - t^* or x^* has nothing to do with a specific token or patch, can represent the whole sentence or image

Inductive Bias

- Inductive Bias: the assumption on a problem by an algorithm
  - CNN assumes the image has the property of "locality", so CNN puts neighboring features together (i.e. the idea of CNN kernel)
  - ViT has a weaker Inductive Bias than CNN
    - Different patches are equal, no matter they are far away or close
- In ViT, only MLP layer is local, while self-attention layer is global
  - ViT can minimize the usage of 2D neighborhood structure
    - In the beginning of the model: cutting the image into patches
    - At fine-tuning time: adjusting the position embeddings for images of different resolution
  - The position embeddings at initialization time carry no information about the 2D positions of the patches
    - It is just a 1D sequence

Hybrid Architecture

- As an alternative to raw image patches, the input sequence can be formed from feature maps of a CNN
- In this hybrid model, the patch embedding projection E is applied to patches extracted from a CNN feature map
- The classification input embedding and position embeddings are added using the same way as a regular ViT

Fine-Tuning and Higher-Resolution

- Typically, we pre-train ViT on large datasets, and fine-tune to (smaller) downstream tasks
  - For this, we remove the pre-trained prediction head and attach a zero-initialized $D \times K$ feedforward layer
    - K is the number of downstream classes
- It is beneficial to fine-tune at higher resolution than pre-training
  - Using a lower resolution in pre-training to reduce the training cost
- When feeding images of higher resolution, we keep the patch size the same, which results in a longer effective sequence length
  - The Vision Transformer can handle arbitrary sequence lengths
  - But the pre-trained position embeddings may no longer work
    - How can we solve this problem?
    - 2D interpolation of the pre-trained position embeddings[1]

[1] The resolution adjustment and patch extraction are the only points at which an inductive bias about the 2D structure of images is manually injected into the Vision Transformer

Dataflow of a ViT encoder layer

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

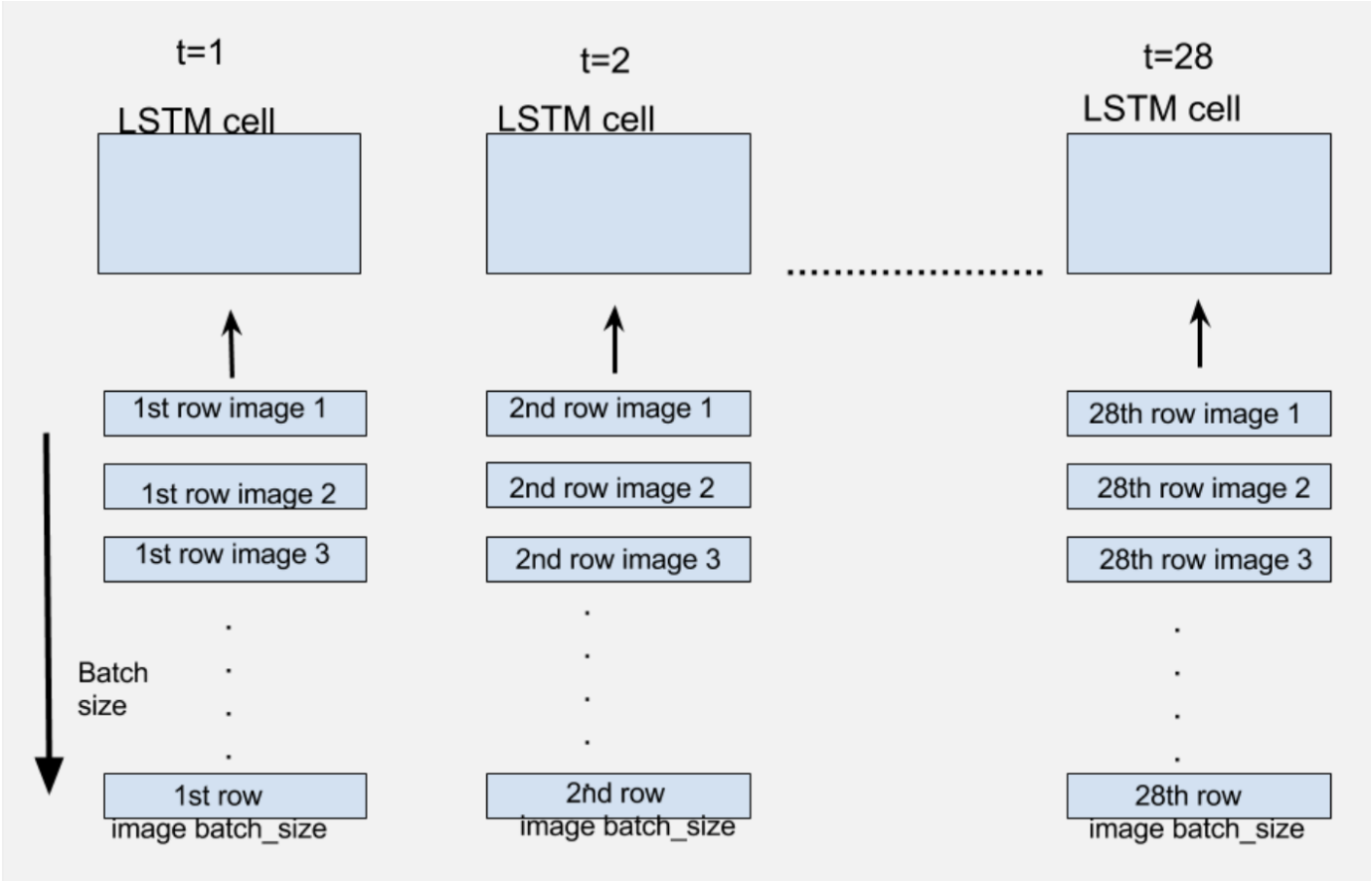
$$\mathbf{z}'_\ell = \text{MSA}(\text{LN}(\mathbf{z}_{\ell-1})) + \mathbf{z}_{\ell-1}, \qquad \ell = 1 \ldots L \tag{2}$$

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

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

- For l-th layer of the encoder
  - z_{l-1} is the input
  - z_l is the output
- In each layer, we have two main parts
  - MSA (Multi-head Self-Attention)
  - MLP (Multi-Layer Perceptron)

A similar idea before ViT



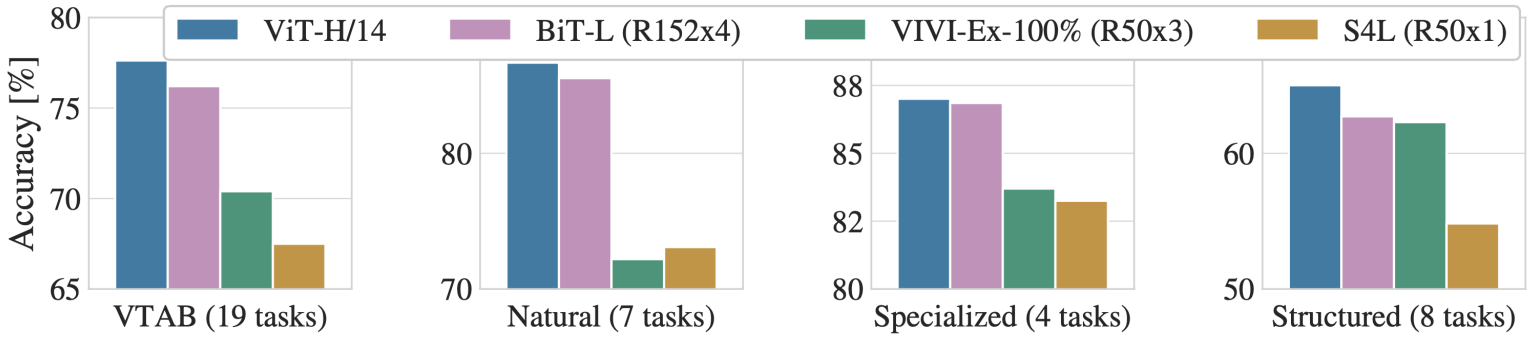
- LSTM for MNIST (image classification)
- https://jasdeep06.github.io/posts/Understanding-LSTM-in-Tensorflow-MNIST/

Different versions of ViT models

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

- ResNet-50: roughly 24M parameters

VTAB results



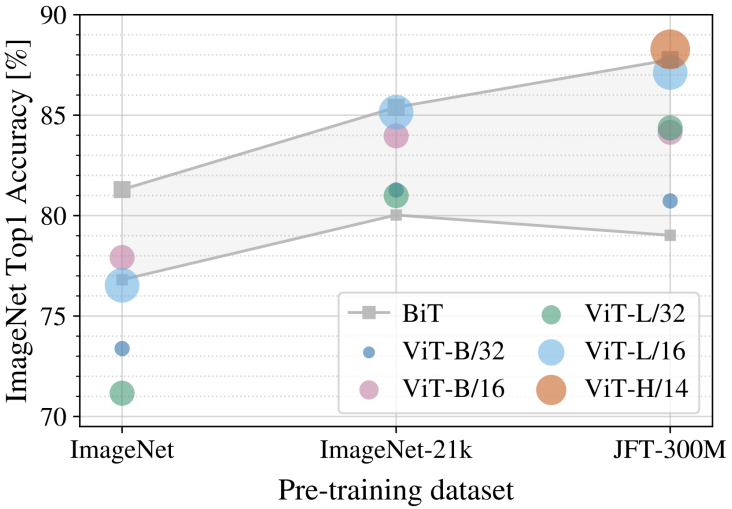
- VTAB (Visual Task Adaptation Benchmark) is an evaluation protocol designed to measure progress towards general and useful visual representations
- VTAB consists of a suite of evaluation vision tasks that a learning algorithm must solve

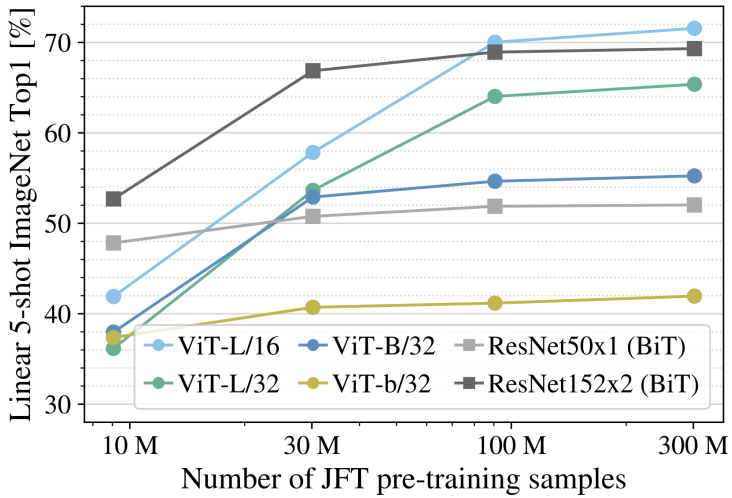
Compared to state-of-the-art

	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** $\pm$ 0.04	87.76 $\pm$ 0.03	85.30 $\pm$ 0.02	87.54 $\pm$ 0.02	88.4/88.5*
ImageNet ReaL	**90.72** $\pm$ 0.05	90.54 $\pm$ 0.03	88.62 $\pm$ 0.05	90.54	90.55
CIFAR-10	**99.50** $\pm$ 0.06	99.42 $\pm$ 0.03	99.15 $\pm$ 0.03	99.37 $\pm$ 0.06	–
CIFAR-100	**94.55** $\pm$ 0.04	93.90 $\pm$ 0.05	93.25 $\pm$ 0.05	93.51 $\pm$ 0.08	–
Oxford-IIIT Pets	**97.56** $\pm$ 0.03	97.32 $\pm$ 0.11	94.67 $\pm$ 0.15	96.62 $\pm$ 0.23	–
Oxford Flowers-102	99.68 $\pm$ 0.02	**99.74** $\pm$ 0.00	99.61 $\pm$ 0.02	99.63 $\pm$ 0.03	–
VTAB (19 tasks)	**77.63** $\pm$ 0.23	76.28 $\pm$ 0.46	72.72 $\pm$ 0.21	76.29 $\pm$ 1.70	–
TPUv3-core-days	2.5k	0.68k	0.23k	9.9k	12.3k

- Vision Transformer models pre-trained on the JFT-300M dataset outperform ResNet-based baselines on all datasets, while taking substantially less computational resources to pre-train.
- ViT pre-trained on the smaller public ImageNet-21k dataset performs well too.

Larger Models and Datasets





- Figure 1: while ViT models perform worse than BiT ResNets (shaded area) when pre-trained on small datasets, they become better when pre-trained on larger datasets.
- Figure 2: ResNets perform better with smaller pre-training datasets but plateau sooner than ViT.