An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale

Round 1

What problem is the paper tackling?

The paper aims to use Transformer models for image recognition tasks, hoping to reduce reliance on traditional convolutional neural networks (CNNs). It argues that Transformers can achieve similar or better results with potentially less computational cost.

What is the relevant background for this problem?

Transformers have been used extensively for NLP tasks, such as the current state-of-the-art models BERT, GPT, and their variations. There's been some other work done on using transformers on image tasks, but they are generally very cost-heavy.

Round 2

What is the conclusion?

Ultimately, these results point towards a possibility of transformers becoming a universal model, capable of learning across a wide domain of human tasks and enjoying the ability to scale with data at an extraordinary scale.

What is the limitation?

The main focus is on image classification, leaving out tasks like object detection and segmentation. Also, the performance improvement with Transformers is less significant compared to CNNs when the dataset is not very large.

Round 3

What methods did the paper propose to address the problem?

To adjust the image input to fit the input for the transformer, the paper reshapes the 2D images into a sequence of flattened 2D patches. A learnable embedding was prepended to the sequence of embedded patches. This token serves a similar purpose as BERT's [class] token. Position embeddings were then added to the patch embeddings to retain positional information.

The transformer encoder consists of alternating layers of multi-headed self-attention and MLP blocks. The state of the output of the Transformer encoder serves as the image representation. During pre-training and fine-tuning, a classification head, MLP, is attached to the output of the Transformer encoder. During pre-training the MLP has one hidden layer, and during fine-tuning it is implemented with a single layer.

The Vision Transformer (ViT) was pre-trained on large datasets and then fine-tuned to smaller downstream tasks. Fine-tuning was done by removing the pre-trained prediction head and replacing it with a zero-initialized feedforward layer.

In a quick conclusion, the process can be divided by (generated by ChatGPT 4o):

1. **Patch Creation**: Divide the image into small patches (e.g., 16x16 pixels).
2. **Flattening and Embedding**: Flatten each patch into a vector and then embed these vectors into a higher-dimensional space.
3. **Positional Encoding**: Add positional information to these vectors to keep track of their position in the original image.
4. **Transformer Encoder**: Process these embedded patches through a Transformer encoder.
5. **Classification Head**: Use the output from the Transformer encoder to classify the image.

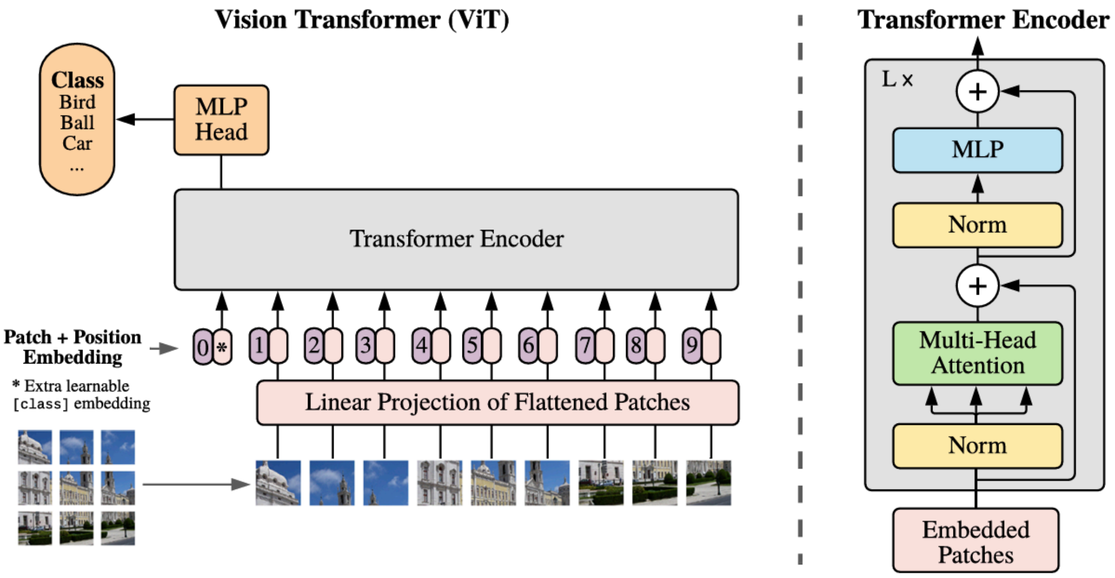
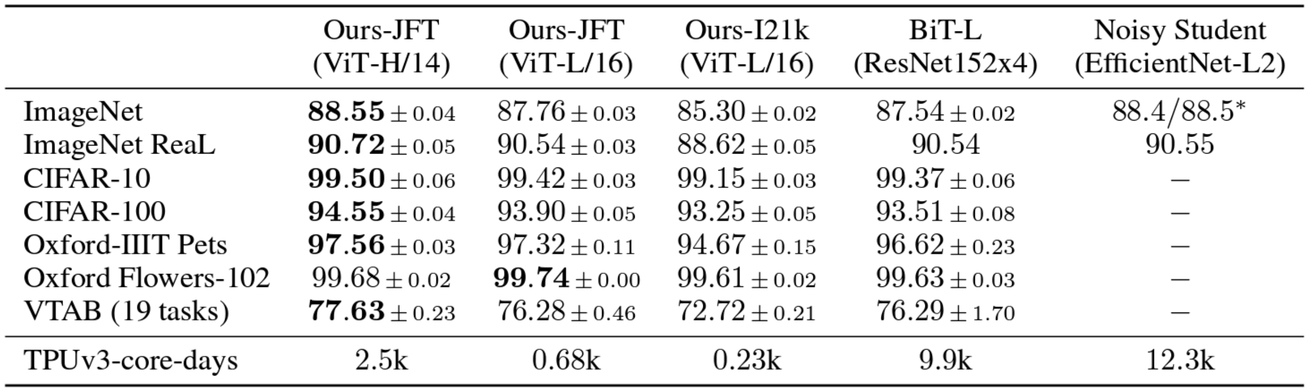


Figure 1: Model overview. We split an image into fixed-size patches, linearly embed each of them, add position embeddings, and feed the resulting sequence of vectors to a standard Transformer encoder. In order to perform classification, we use the standard approach of adding an extra learnable "classification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

What results did it get?

This paper distinguishes itself as a successful application of standalone transformers for CV. For each of the main contributions, it differs as follows:

- **Accuracy with less compute time**: ViT has decreased the training time by 5 times (20% of training time) against Noisy Student but achieving similar accuracy.



	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

Table 2: Comparison with state of the art on popular image classification benchmarks. We report mean and standard deviation of the accuracies, averaged over three fine-tuning runs. 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. *Slightly improved 88.5% result reported in Touvron et al. (2020).

- **No Convolutions**: In theory, a MLP performs better than a CNN model. However, data has been a large barrier with respect to the performance of MLP models. The inductive bias imposed by CNNs has greatly advanced the field of CV, and with the large dataset used by the authors, they are able to overcome the need for an inductive bias. A transformer is slightly different from a traditional MLP, with its core mechanism being **self-attention**. This gives transformers the ability to understand the relationship between inputs. When used in NLP, it computes the relation between words in a bidirectional manner, which means the order is less strict, unlike a unidirectional RNN.
- **Efficacy of Transformer**: ViTs can capture global image information from the lower layers and use positional embeddings to understand spatial relationships between patches.

How did the paper assess its results?

This paper uses three major datasets: ImageNet, JFT, and VTAB. They measure performance through fine-tuning and few-shot accuracy, which looks at how well the model performs after training on large and small datasets, respectively.

They compare ViTs against benchmarks like Big Transfer and Noisy Student. They also test self-supervised training, which improves accuracy by 2%.