# Vision Transformers

Resources

📄 Article
Dosovitskiy, An Image is Worth 16x16 Words

GitHub
Implementation

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 sub- stantially fewer computational resources to train.

# Introduction

## Rise of Self-Attention Based Models

Self-Attention based architectures, **Transformers** in particular, have become the model of choice for natural language processing tasks.

### Approach

1. Pre-train first on *large corpus* of text.
2. Fine-tune on smaller, *task-specific* text.

### Effectiveness

- Due to computational *efficiency and scalability*, possible to train models upto 100B params! 😱
- Even with growing size of datasets and models, *no sign of saturation in performance*. 💪

## Combination with Convolutional Architectures

In computer vision, *convolution-based architectures* are still dominating. Inspired by the effectiveness of transformers, multiple works have attempted to combine CNNs with self-attention.

### Current Drawbacks

- Models theoretically efficient, but neither tested nor scaled effectively on modern hardware accelertors due to use of specialized attention-patterns.
- So ResNet architectures still "state-of-the-art" in image classification landscape.

## Patches as Tokens

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

1. Applying standard transformer directly to images, with fewest possible mods.
2. Split image into patches.
3. Linear embedding on each patch.
4. Sequence of embeddings → Transformer, *just like usual tokens*

5. Train model on image classification data in supervised fashion.

# Size Matters

## Mid-Sized Datasets 🐤

- When trained on **mid-sized** datasets like ImageNet *without strong regularization*, accuracy is a few percentage points below ResNets of comparable size.
- This is expected, as Transformers lack some of the inductive biases inherent to CNNs, like *translational equivariance* and *locality*. So they cannot generalize well on insufficient amounts of data.

Properties of CNNs

### Translational Equivariance

1. Translation equivariance is a fundamental property of Convolutional Neural Networks (CNNs) that allows them to *maintain the spatial relationships of features* when the input image is translated.
2. Translation equivariance refers to the ability of a function or system to produce outputs that shift in the same way as its inputs when those inputs are translated. In the context of CNNs, if an input image is translated (shifted), the output feature maps generated by the convolutional layers will also be translated by the same amount.
   blog.papersapce
   datascience.stackexchange
   chriswolfvision.medium.com

### Locality

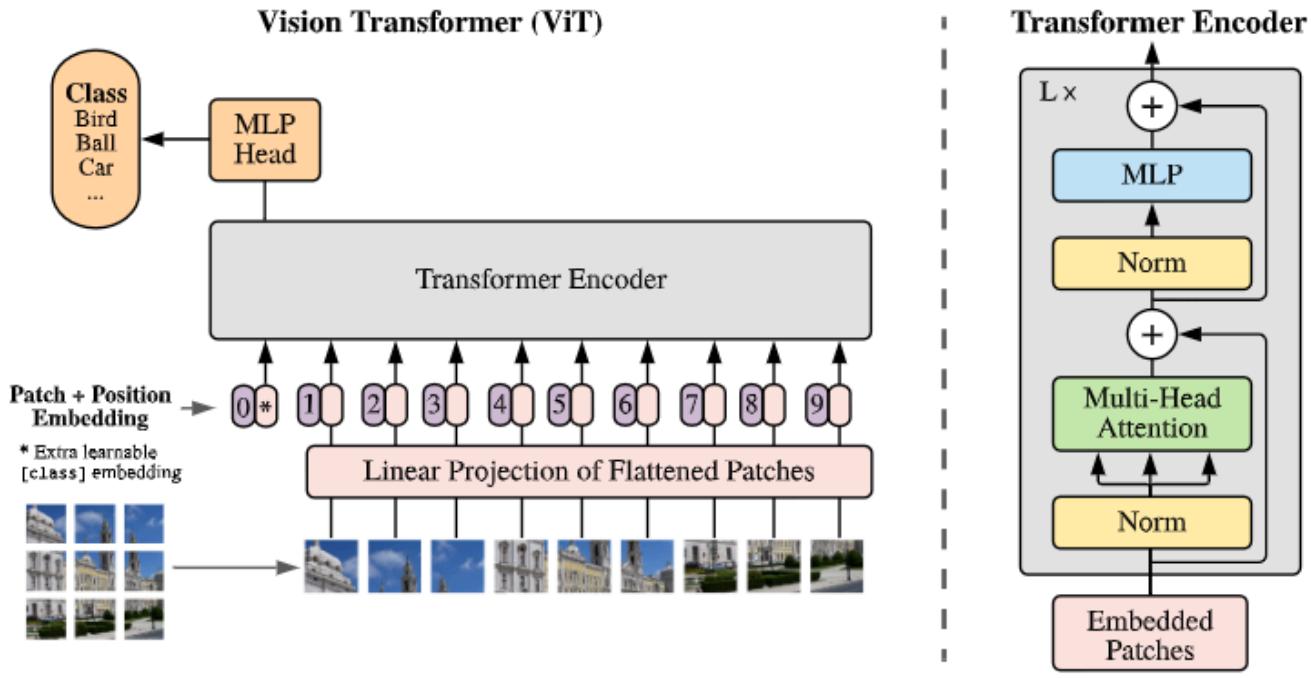
Locality in Convolutional Neural Networks (CNNs) refers to the principle that nearby pixels in an image are more likely to be correlated than distant pixels. This concept is crucial for effectively processing visual data, as it allows CNNs to *focus on local patterns and features within an image*, rather than treating all pixels as independent.

## Large Datasets 🐔

When trained on enormous datasets of 14M-300M examples, large scale training trumps inductive bias. Pre-trained on ImageNet 21k, JFT-300M, ViT beats state of the art models on image recognition benchmarks.

| Dataset | Accuracy |
| --- | --- |
| ImageNet | 88.55% |
| ImageNet-ReaL | 90.72% |
| CIFAR-100 | 94.55% |
| VTAB (19 tasks) | 77.63% |

# Methodology




> In model design we follow the original Transformer (Vaswani et al., 2017) as closely as possible. An advantage of this intentionally simple setup is that scalable NLP Transformer architectures - and their efficient implementations - can be used almost out of the box.

Vision-Transformers, page 3

## Vision Transformer (ViT)

## Handling 2D Images

1. Original image $x \in \mathbb{R}^{H \times W \times C}$ is reshaped to sequence of flattened 2D patches.
2. Flattened 2D images $x_p \in \mathbb{R}^{N \times (P^2 \cdot C)}$ where $N = HW/P^2$
3. Transformer uses $D = d_{model}$ as the constant latent vector size through all layers.
4. The image patches are embedded to $D$ dimensions by flattening and projecting to $D$ dimensions using a trainable linear projection ($\mathbf{E}$).

$$e^i_{\text{patch}} = x^i_{\text{patch}} \cdot \mathbf{E}$$

| Symbol | Meaning |
|---|---|
| $(H, W)$ | Original image resolution |
| $C$ | Number of channels, also serves as effective input sequence length for the transformer |
| $(P, P)$ | Resolution of each image patch |
| $N$ | Number of patches |
| $D = d_{\text{model}}$ | The latent vector size of the transformer |
| $\mathbf{E}$ | The trainable weights for linear projection of patches to $D$ dimensions |

## [class] Token as a Learnable Embedding

- A learnable embedding $\mathbf{z}_0^0 = \mathbf{x}_{\text{class}}$ is *appended to the sequence* of embedded patches.
- The **state of this embedding** $\mathbf{z}_\mathbf{L}^0$ at the *output of the transformer* encoder serves as the image representation $\mathbf{y}$.
- During pre-training as well as fine-tuning, as **classification head** is attached to $\mathbf{z}_\mathbf{L}^0$. It is implemented using an MLP with:
  - One hidden layer during pre-training
  - Single linear layer during fine-tuning

> **Notes**
>
> - **Positional encodings** are applied ONLY to the *patch embeddings*.
> - (Below) the MLP contains two layers with a GELU activation.

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

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

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

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

# Properties of ViT

## Inductive Bias

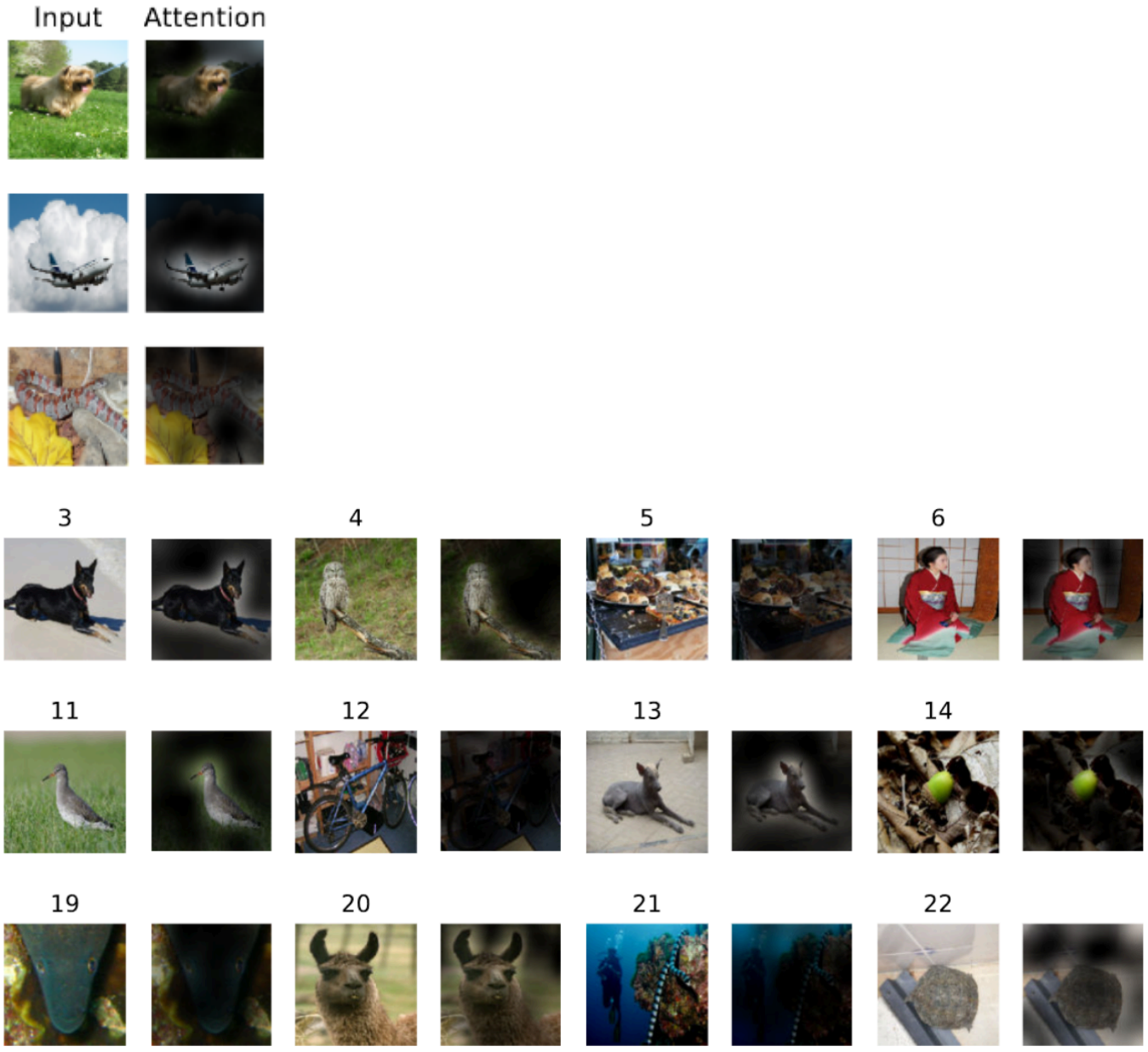
- Vision Transformer has much less image-specific inductive bias than CNNs.
  Learning Deep Learning: ViT (patrick-llgc.github.io)
  Difference Between CNN and ViT (viso.ai)
- CNNs:
  - Leverage locality, 2D structure, and translation equivariance throughout layers.
    - Preserve spatial relationships within images inherently.
- ViTs:
  - Primarily use global self-attention.
  - Only MLPs are local and translationally equivariant.
  - Learn spatial relationships between image patches via position embedding, not encoded initially.

## Hybrid Architecture

- **Feature Maps from CNNs:** Use CNNs to generate hierarchical feature maps from input images.
- **Patch Embedding Projection:** Apply a projection to extract patches from these feature maps, preparing them as input for the Transformer.
- **1x1 Patches:** Optionally treat each spatial location in the feature map as an individual patch, flattening spatial dimensions into a sequence.
- **Integration with Transformer:** Feed patch embeddings into a Transformer to leverage its ability to capture global dependencies and relationships across the input sequence.

---

# Results




> We find that the model attends to image regions that are semantically relevant for classification

Vision-Transformers, page 8