Vision Transformer ViT

AI Consultants Earth & Environment @DKRZ

Mosaku Adeniyi

Helmholtz AI

Vision Transformer Introduction

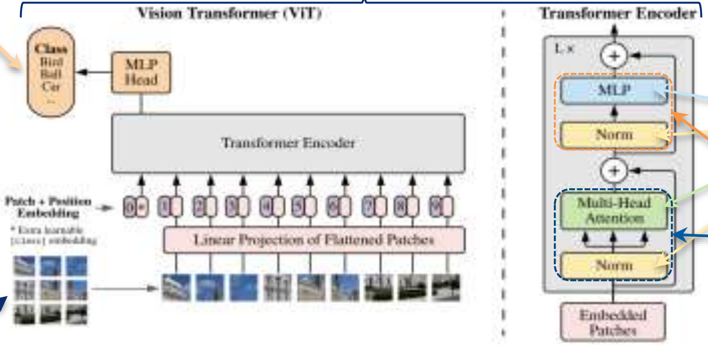


Figure 1: Model overview. We split an image into fixed-size patches, linearly embed each of the add position embeddings, and feed the resulting sequence of vectors to a standard Transform encoder. In order to perform classification, we use the standard approach of adding an extra learnat "classification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

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

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

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

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

- ViT : Another transformer flavour
- Vision Transformer Components are essential part of ViT

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.

- Four equations that explains the model architecture
- Model sizes based on different numbers of hyper-parameters

Helmholtz AI

Breaking the equations down - 1

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

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

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

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

- **Patch embeddings:**

➢ Embeddings are learnable representations, thus require grad

➢ Images are flattened to sequence of 2D patches in latent space

➢ Hidden size ***D*** = 768 is used as feature maps

➢ Each feature represent the encoded patched image

➢ Input of ***X***, the number of patches is ***N*** and expected output of ***Xp***

➢ Stride and Kernel size in Conv2D can be used to achieve this

$\mathbf{x} \in R^{H\times W\times C}$ $\quad$ $N = HW/P^2$ $\quad$ $\mathbf{x_p} \in R^{N\times(P^2\cdot C)}$

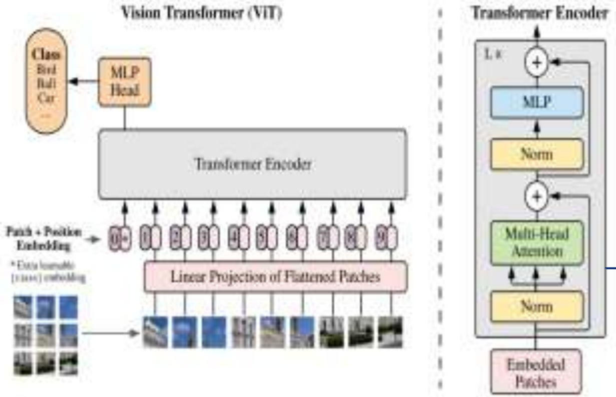


Figure 1: Model overview. We split an image into fixed-size patches, linearly embed each of them, add position embeddings, and feed the resulting sequence of vectors to a standard Transformer encoder. In order to perform classification, we use the standard approach of adding an extra learnable "classification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

- Transformer receives a 1D sequence of token embeddings

Helmholtz AI

Breaking the equations down - 1

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

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

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

$$\mathbf{y} = \mathrm{LN}(\mathbf{z}_L^0) \quad (4)$$

- **Class Tokens:**

➢ They are learnable parameters, serves as image representation

➢ We prepend it on the image patches

- **Position embeddings:**

➢ Position embedding preserve the position information

➢ It is also a learnable parameter used as an embedding

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

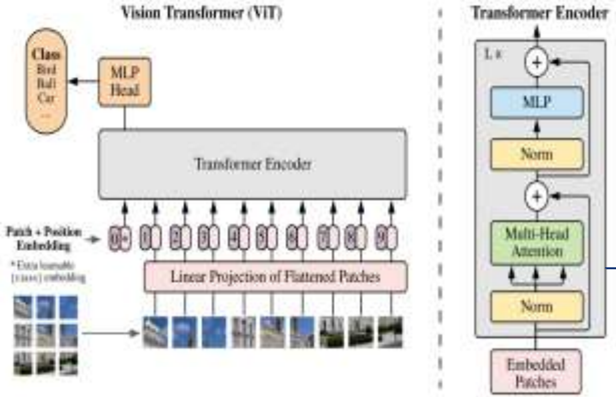


Figure 1: Model overview. We split an image into fixed-size patches, linearly embed each of them, add position embeddings, and feed the resulting sequence of vectors to a standard Transformer encoder. In order to perform classification, we use the standard approach of adding an extra learnable "classification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

- Transformer receives a 1D sequence of token embeddings

Helmholtz AI

Breaking the equations down - 2

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

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

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

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

- **Multi-Head Self Attention**

➢ Each flattened feature is ejected into the Encoder

➢ Each token ejects three entities - **Query Q**, **Key K**, **Value V**

➢ **QKV** serves as the weight to learn image feature maps

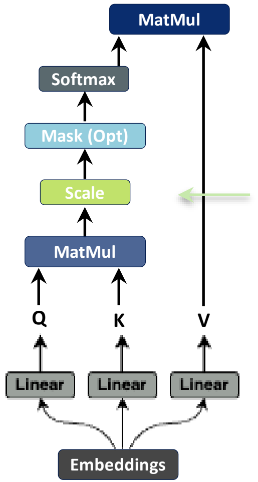
➢ Increase of affinity through matrix multiplication

➢ Scale along embedding dimension / latent space dimension

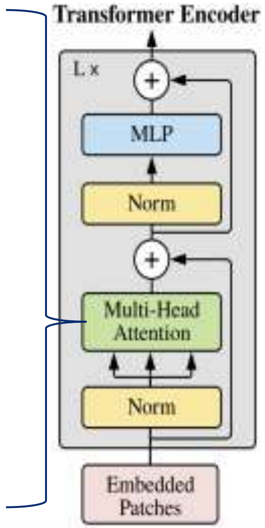
➢ The mask is optional, could be absent in sentiment analysis

➢ The dimesion results into (***T, T***)

$$Y = \underbrace{Softmax(QK^\top)}_{\text{Attention matrix}} V$$



$$\frac{1}{\sqrt{D_{\text{model}}}}$$



Helmholtz AI

Breaking the equations down - 2

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

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

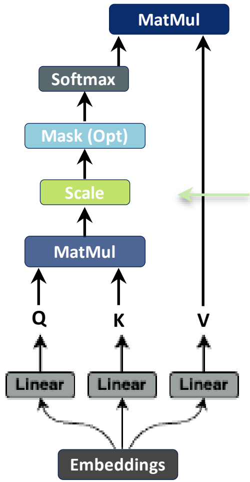
$$\mathbf{z}_\ell = \text{MLP}(\text{LN}(\mathbf{z}'_\ell)) + \mathbf{z}'_\ell, \qquad \ell = 1 \ldots L \tag{3}$$

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

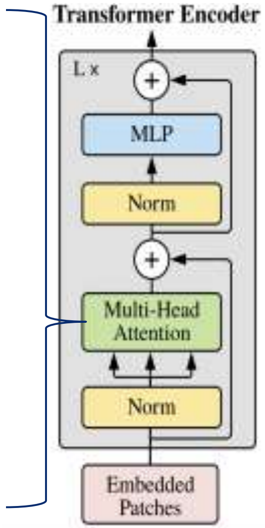
- **Residual Connection:**
  - Skip connection solves the problem of vanishing gradient effect

$$Y = \underbrace{Softmax(QK^\top)}_{\text{Attention matrix}} V$$

- **Layer Normalization:**
  - Normalization along embedding dimension
  - Standardization allows smoother gradient and better accuracy



$$\frac{1}{\sqrt{D_{\text{model}}}}$$



Helmholtz AI

Breaking the equations down – 3&4

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

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

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

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

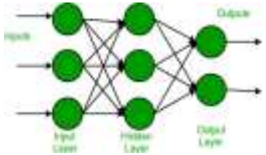
- **Multi-Layer Perceptron:**

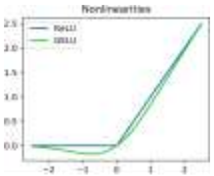
➢ MLP maps input to output – forward direction

➢ We could have multiple hidden layer

➢ GELU activation function is used in ViT

➢ Dropout is also utilised





- **Linear Layer:**

➢ Maps **input** to **class**

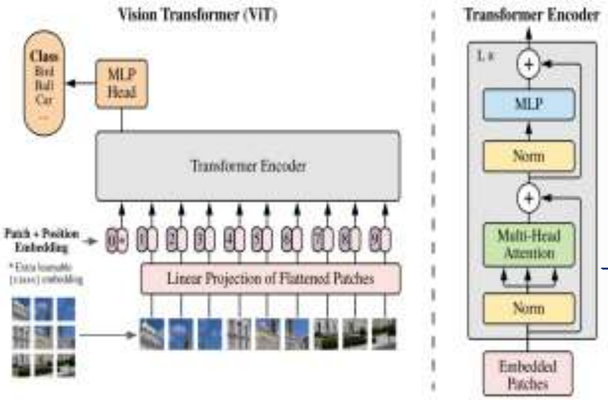


Figure 1: Model overview. We split an image into fixed-size patches, linearly embed each of them, add position embeddings, and feed the resulting sequence of vectors to a standard Transformer encoder. In order to perform classification, we use the standard approach of adding an extra learnable "classification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

- Transformer receives a 1D sequence of token embeddings

Helmholtz AI

QUESTIONS