HTWG Konstanz

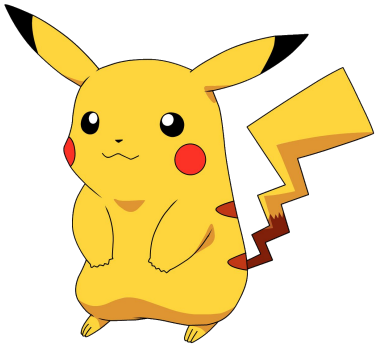
MSI Seminar Advanced Topics in Data Analysis and Deep Learning

Vision Transformer

Alexander Haab

Sommersemester 2025

Contents



- Paper Introduction
- Inspecting Vision Transformer
- Model Variants
- Training & Fine-tuning
- Comparison to State of the Art
- Fine-tuning a Vision Transformer on a Pokemon dataset
- Applications
- Conclusion / Outlook

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

(Dosovitskiy et al., 2021)

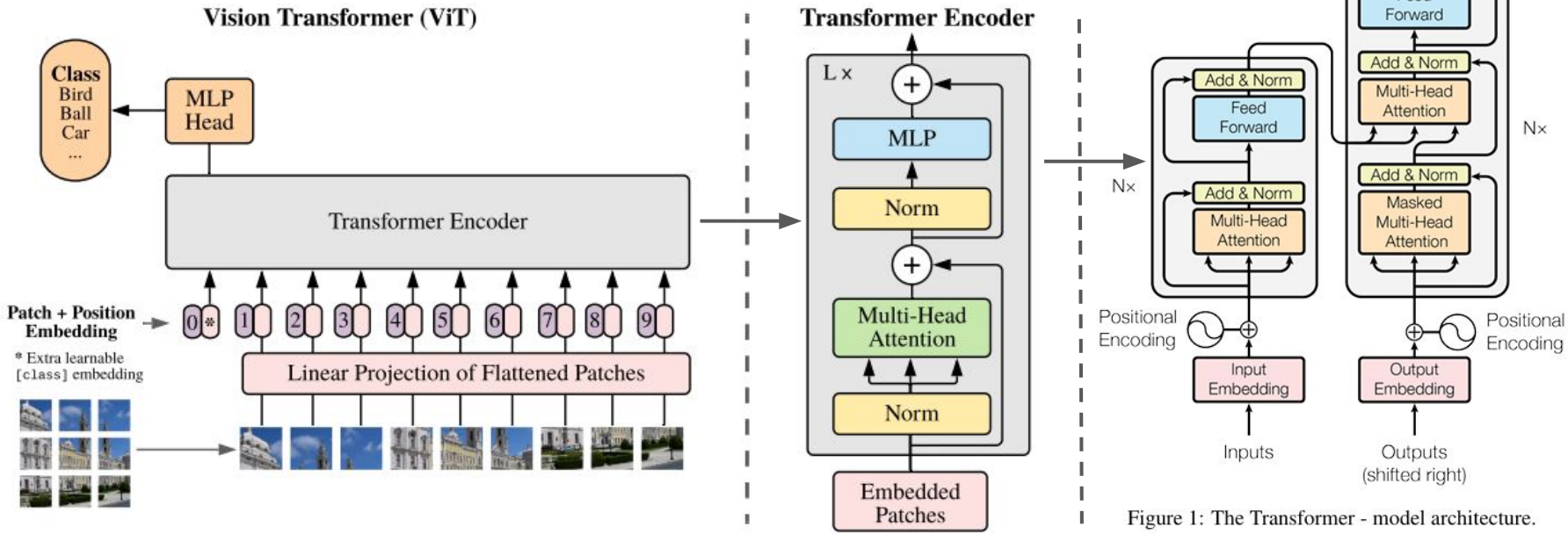
- NLP: Transformer
- Vision: CNN

- Introducing ViT architecture
- Google Research, 2020

- Pure transformer
- Image classification tasks

Source: https://www.kaggle.com/code/vigneshwar472/google-vision-transformer-results

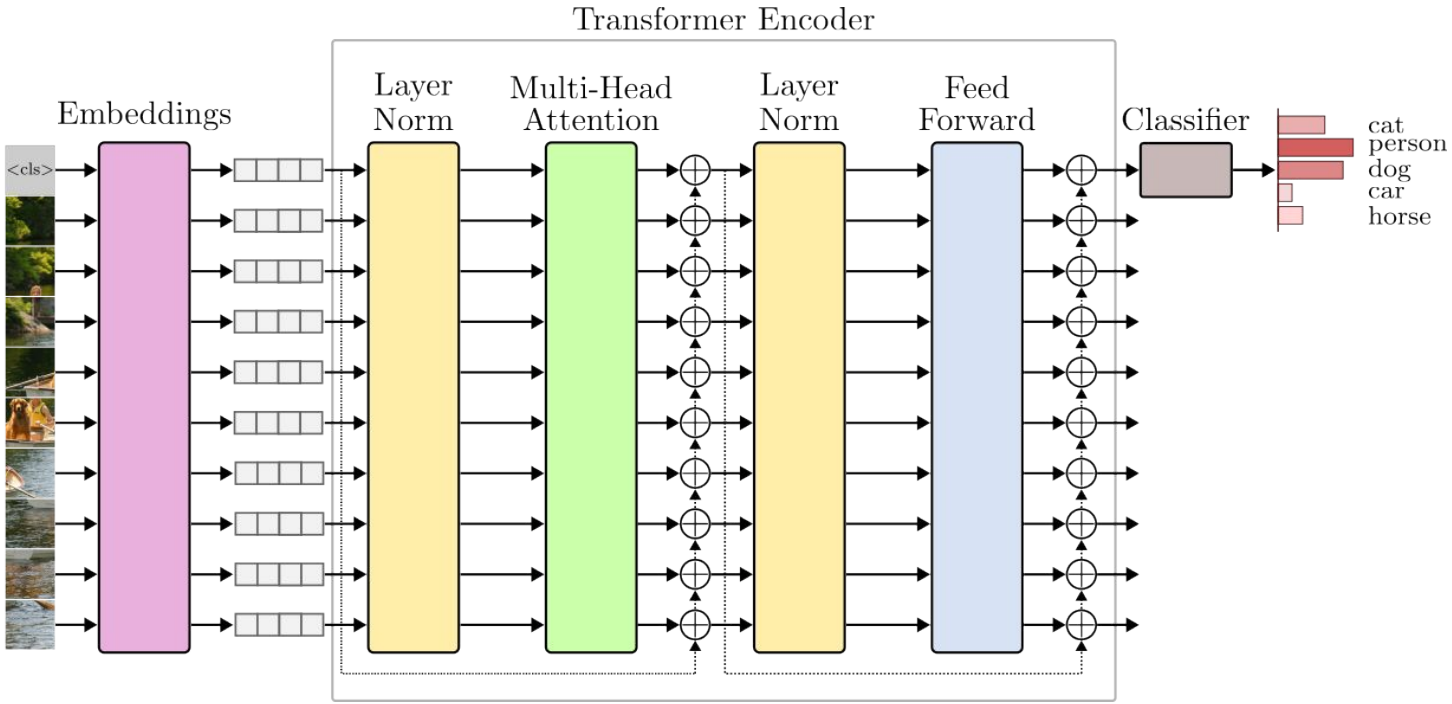
An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale



(Dosovitskiy et al., 2021)

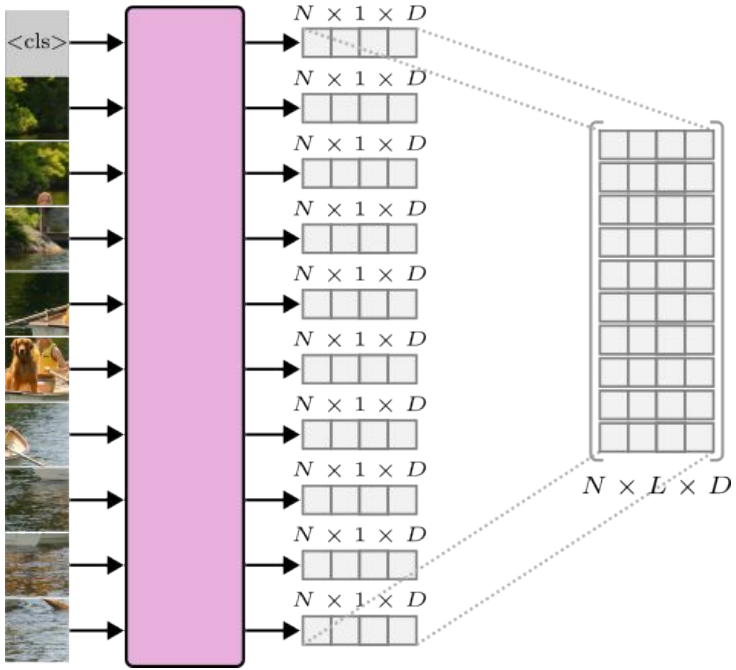
Figure 1: The Transformer - model architecture.
(Vaswani et al., 2017)

Image -> Embeddings -> Transformer Encoder -> Classifier -> Classification



(Silva, 2023)

Embeddings



(Silva, 2023)

The standard **Transformer** receives a **1D sequence of token embeddings** as **input.**

(Dosovitskiy et al., 2021)

The Embedding layer **maps 2D image patches into 1D vectors.**

Token Embeddings

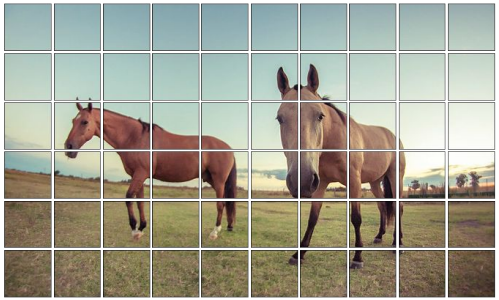
The quick brown fox jumps over the lazy dog.

The quick brown fox jumps over the lazy dog.

T h e q u i c k b r o w n f o x j u m p s o v e r t h e l a z y d o g .

(Silva, 2023)

Split image into patches



Reshape the **image x** into **sequence** of

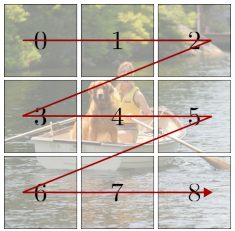
flattened 2D patches x_p

$x \in R^{H \times W \times C} \rightarrow x_p \in R^{N \times (P^2 * C)}$

(H, W): Input image resolution, (C): Channels, (P, P): Resolution of each image patch

$N = HW / P^2$: number of patches

(Dosovitskiy et al., 2021)



Example:

image resolution: **224** x **224** pixel x **3** RGB

patch resolution: **16** x **16** pixel (“An Image is Worth **16 x 16 Words**”)

N = **196** Patches

$x \in R^{224 \times 224 \times 3} \rightarrow x_p \in R^{196 \times 768}$

(Silva, 2023)

Linear Projection

Transformer uses vector size D (=768 Base Model) through all of its layers.

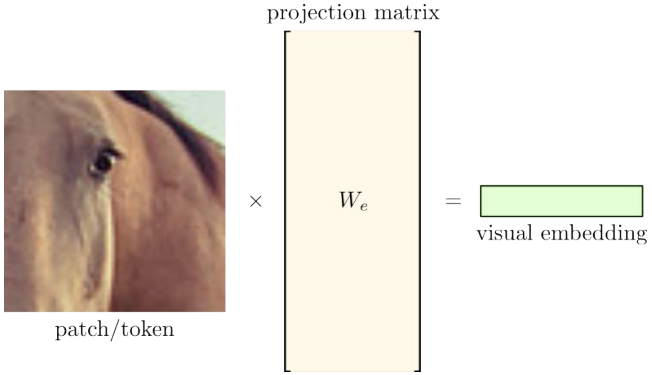
Map each patch $x_p \in R^{196 \times 768}$ to D dimensions.

Trainable linear projection matrix $E \in R^{768 \times 768}$

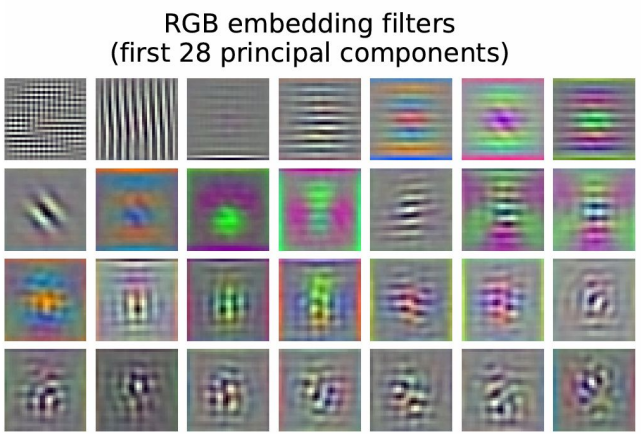
Output: patch embeddings $\in R^{196 \times 768}$

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

(Dosovitskiy et al., 2021)

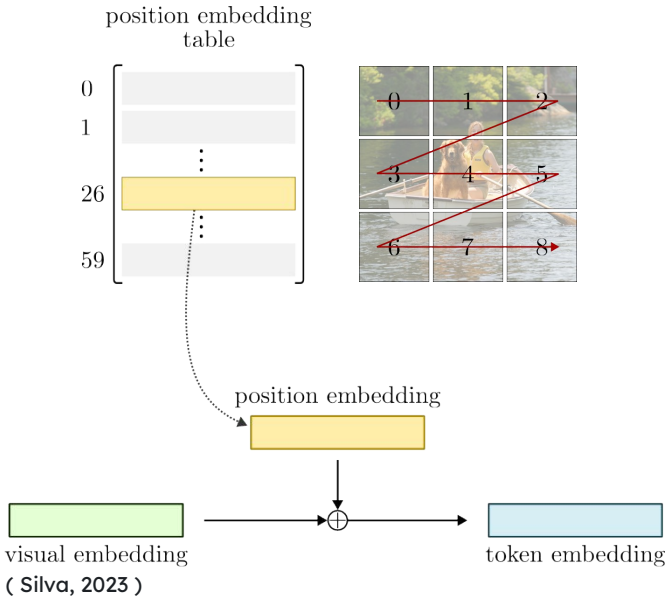


(Silva, 2023)



(Dosovitskiy et al., 2021)

Position embedding



(Silva, 2023)

Position embeddings E_{pos} are **added** to the **patch embeddings** to retain **positional information**.

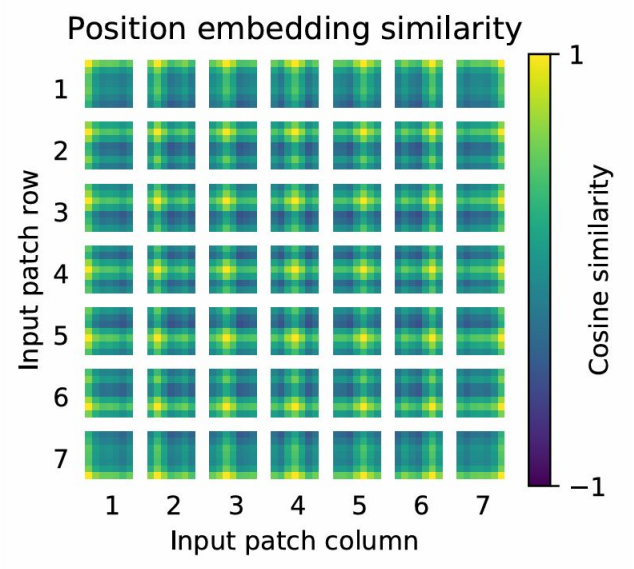
Patch embeddings $\in R^{196 \times 768}$

Learnable 1D position embeddings $\in R^{196 \times 768}$

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

(Dosovitskiy et al., 2021)

Position embedding



The model learns to **encode distance** within the image in the **similarity of position embeddings**.

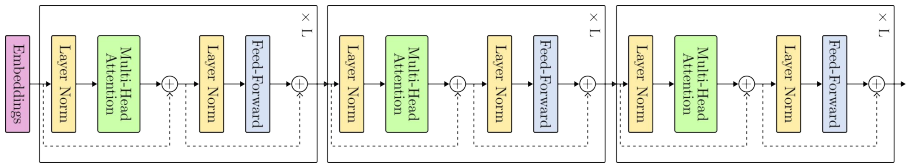
Closer patches tend to have more **similar position embeddings**.

Row-column structure appears.

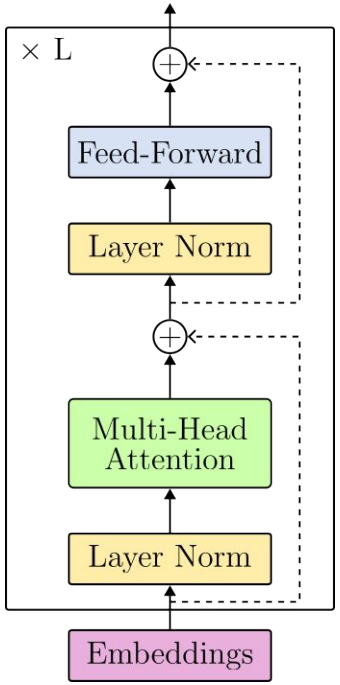
(Dosovitskiy et al., 2021)

Transformer Encoder

- **Multi-Head Attention Layer**
- **Feed-Forward (MLP)**
  - two layers
  - GELU non-linearity
- **Layer Norm & Residual Connections**
  - ensures robust and efficient training



x12 (Base)



(Silva, 2023)

Multi-Head Attention Layer

Self-attention allows to **integrate information across the entire image**.

Attends to **image regions** that are **relevant for classification**.

$$[\mathbf{q}, \mathbf{k}, \mathbf{v}] = \mathbf{z}\mathbf{U}_{qkv} \qquad \mathbf{U}_{qkv} \in \mathbb{R}^{D \times 3D_h}, \tag{5}$$

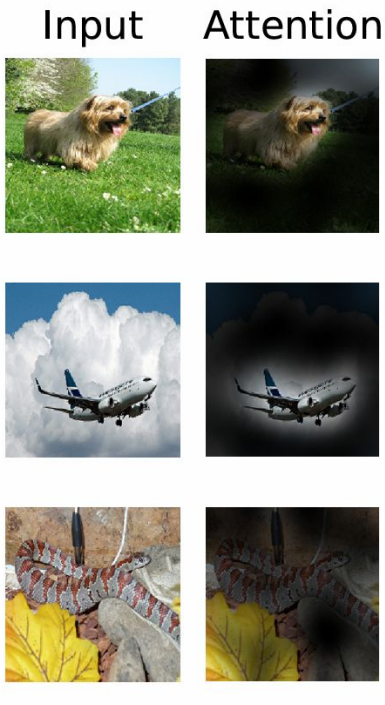
$$A = \text{softmax}\left(\mathbf{q}\mathbf{k}^\top / \sqrt{D_h}\right) \qquad A \in \mathbb{R}^{N \times N}, \tag{6}$$

$$\text{SA}(\mathbf{z}) = A\mathbf{v}\,. \tag{7}$$

Multi-head self-attention (MSA) runs **k self-attention** operations, called “**heads**”, in **parallel**, and project their concatenated outputs.

$$\text{MSA}(\mathbf{z}) = [\text{SA}_1(z); \text{SA}_2(z); \cdots; \text{SA}_k(z)]\,\mathbf{U}_{msa} \qquad \mathbf{U}_{msa} \in \mathbb{R}^{k \cdot D_h \times D} \tag{8}$$

(Dosovitskiy et al., 2021)

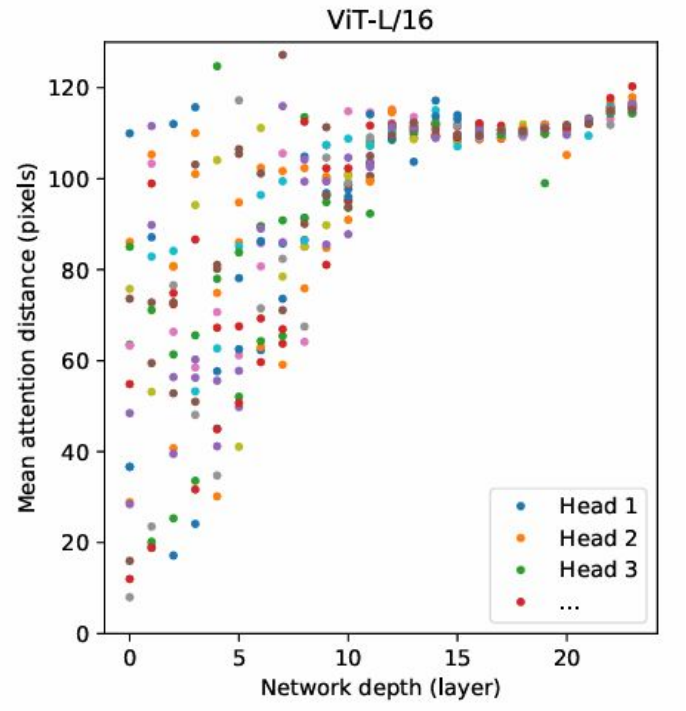


Multi-Head Attention Layer

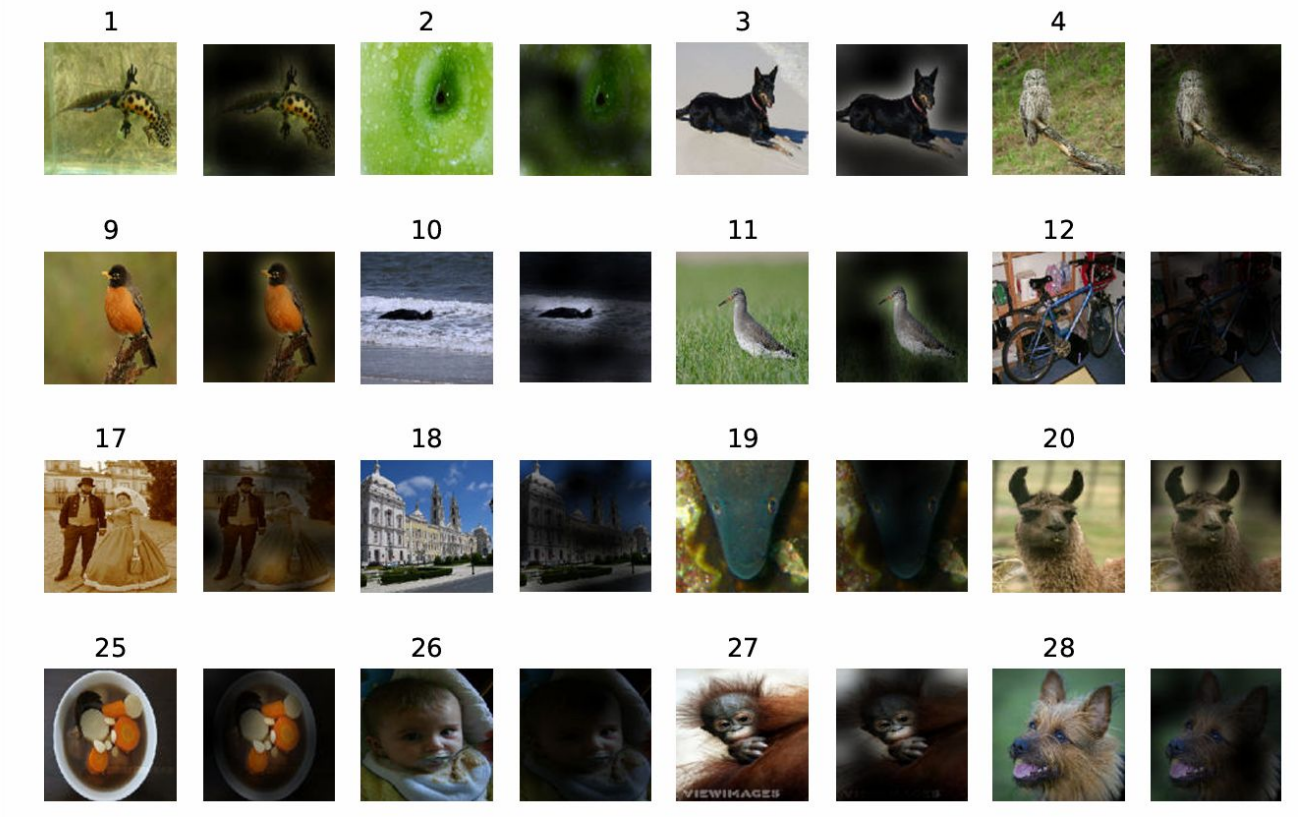
Some **heads attend most** of the image,
showing the ability to **integrate information globally.**

Other attention heads have **consistently small attention distances**.

Further, the **attention distance increases** with **network depth**.



(Dosovitskiy et al., 2021)

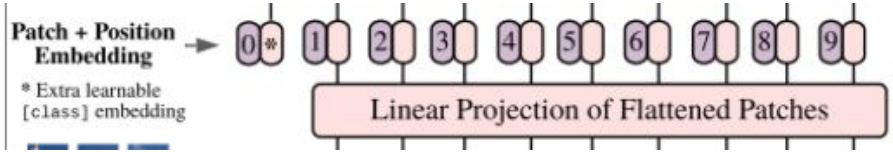


(Dosovitskiy et al., 2021)

Classifier

Prepend a **learnable embedding [class] token** to the sequence of embedded patches.

Serves as the **image representation** y, after Transformer encoder output.



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

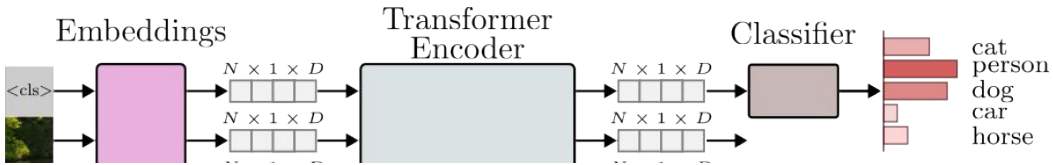
$$\mathbf{z}'_\ell = \text{MSA}(\text{LN}(\mathbf{z}_{\ell-1})) + \mathbf{z}_{\ell-1}, \quad \ell = 1 \ldots L \tag{2}$$

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

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

Classification head is implemented by a **MLP**:

- pre-training time -> **one hidden layer**
- fine-tuning time -> **single linear layer**



Model Variants

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.

Models	Dataset	Epochs	Base LR	LR decay	Weight decay	Dropout
ViT-B/{16,32}	JFT-300M	7	$8 \cdot 10^{-4}$	linear	0.1	0.0
ViT-L/32	JFT-300M	7	$6 \cdot 10^{-4}$	linear	0.1	0.0
ViT-L/16	JFT-300M	7/14	$4 \cdot 10^{-4}$	linear	0.1	0.0
ViT-H/14	JFT-300M	14	$3 \cdot 10^{-4}$	linear	0.1	0.0
R50x{1,2}	JFT-300M	7	10^{-3}	linear	0.1	0.0
R101x1	JFT-300M	7	$8 \cdot 10^{-4}$	linear	0.1	0.0
R152x{1,2}	JFT-300M	7	$6 \cdot 10^{-4}$	linear	0.1	0.0
R50+ViT-B/{16,32}	JFT-300M	7	$8 \cdot 10^{-4}$	linear	0.1	0.0
R50+ViT-L/32	JFT-300M	7	$2 \cdot 10^{-4}$	linear	0.1	0.0
R50+ViT-L/16	JFT-300M	7/14	$4 \cdot 10^{-4}$	linear	0.1	0.0
ViT-B/{16,32}	ImageNet-21k	90	10^{-3}	linear	0.03	0.1
ViT-L/{16,32}	ImageNet-21k	30/90	10^{-3}	linear	0.03	0.1
ViT-$*$	ImageNet	300	$3 \cdot 10^{-3}$	cosine	0.3	0.1

Table 3: Hyperparameters for training. All models are trained with a batch size of 4096 and learning rate warmup of 10k steps. For ImageNet we found it beneficial to additionally apply gradient clipping at global norm 1. Training resolution is 224.

(Dosovitskiy et al., 2021)

Training & Fine-tuning

- Pre-trained on large amounts
- Fine-tuned for task
- Trained in supervised fashion

- ImageNet
  - 1.3M images
  - 1k classes
- ImageNet-21k
  - 14M images
  - 21k classes
- JFT-300M
  - 300M images
  - 18k classes

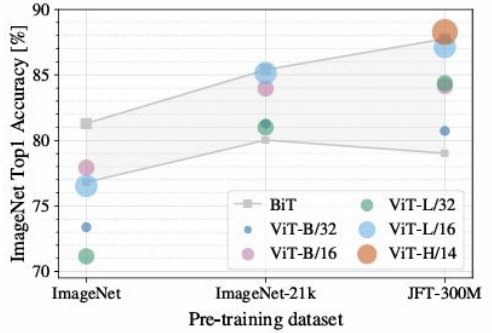


Figure 3: Transfer to ImageNet. While large ViT models perform worse than BiT ResNets (shaded area) when pre-trained on small datasets, they shine when pre-trained on larger datasets. Similarly, larger ViT variants overtake smaller ones as the dataset grows.

(Dosovitskiy et al., 2021)

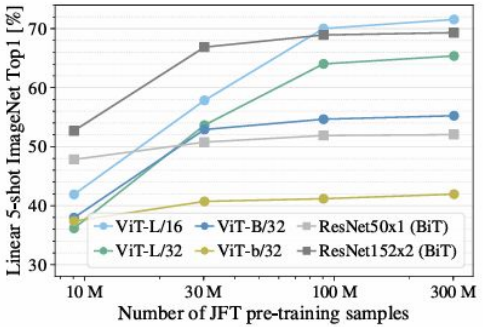


Figure 4: Linear few-shot evaluation on ImageNet versus pre-training size. ResNets perform better with smaller pre-training datasets but plateau sooner than ViT, which performs better with larger pre-training. ViT-b is ViT-B with all hidden dimensions halved.

Training & Fine-tuning

CIFAR-10:

- 60000 images
- 10 classes

CIFAR-100:

- 60000 images
- 100 classes

Oxford Flowers-102

- 102 classes
- 8000 images

		ViT-B/16	ViT-B/32	ViT-L/16	ViT-L/32	ViT-H/14
ImageNet	CIFAR-10	98.13	97.77	97.86	97.94	-
	CIFAR-100	87.13	86.31	86.35	87.07	-
	ImageNet	77.91	73.38	76.53	71.16	-
	ImageNet ReaL	83.57	79.56	82.19	77.83	-
	Oxford Flowers-102	89.49	85.43	89.66	86.36	-
	Oxford-IIIT-Pets	93.81	92.04	93.64	91.35	-
ImageNet-21k	CIFAR-10	98.95	98.79	99.16	99.13	99.27
	CIFAR-100	91.67	91.97	93.44	93.04	93.82
	ImageNet	83.97	81.28	85.15	80.99	85.13
	ImageNet ReaL	88.35	86.63	88.40	85.65	88.70
	Oxford Flowers-102	99.38	99.11	99.61	99.19	99.51
	Oxford-IIIT-Pets	94.43	93.02	94.73	93.09	94.82
JFT-300M	CIFAR-10	99.00	98.61	99.38	99.19	99.50
	CIFAR-100	91.87	90.49	94.04	92.52	94.55
	ImageNet	84.15	80.73	87.12	84.37	88.04
	ImageNet ReaL	88.85	86.27	89.99	88.28	90.33
	Oxford Flowers-102	99.56	99.27	99.56	99.45	99.68
	Oxford-IIIT-Pets	95.80	93.40	97.11	95.83	97.56

Table 5: Top1 accuracy (in %) of Vision Transformer on various datasets when pre-trained on ImageNet, ImageNet-21k or JFT300M. These values correspond to Figure 3 in the main text. Models are fine-tuned at 384 resolution. Note that the ImageNet results are computed without additional techniques (Polyak averaging and 512 resolution images) used to achieve results in Table 2.

(Dosovitskiy et al., 2021)

Comparison to State of the Art

- Beats state-of-the-art convolutional networks,
- on multiple image recognition benchmarks,
- while requiring substantially fewer computational resource to train

	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

(Dosovitskiy et al., 2021)

Comparison to State of the Art

Image Classification on ImageNet

Rank	Modell	Accuracy	Num. Parameters	Year
#1	Coco	91%	2100M	2022
#15	ViT-L/16	89.6%	307M	2023

Image Classification in CIFAR-10

Rank	Model	Accuracy	Num. Parameters	Year
#1	ViT-H/14	99.5%	632M	2020

Source: https://paperswithcode.com/task/image-classification

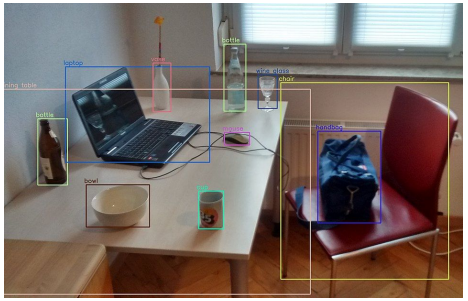
Fine-tuning a Vision Transformer on a Pokemon dataset

Github: https://github.com/alexanderhaab/msi_seminar_vision_transformer

Copyright Nintendo

Applications

- Image Classification
- Image captioning
- Object Detection
- Video Deepfake Detection
- Image segmentation
- Anomaly detection
- Autonomous driving



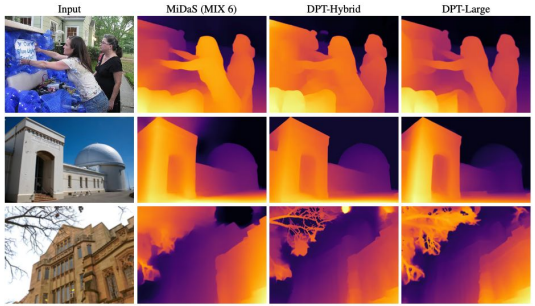
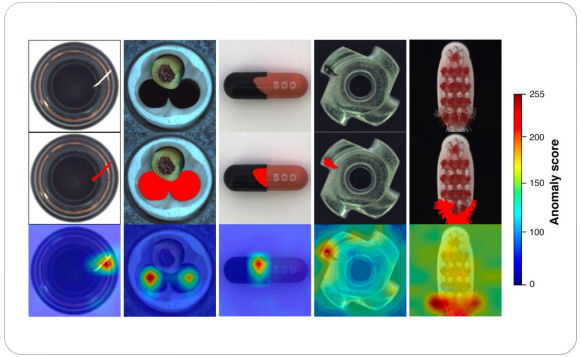


Figure 2. Sample results for monocular depth estimation. Compared to the fully-convolutional network used by MiDaS, DPT shows better global coherence (e.g., sky, second row) and finer-grained details (e.g., tree branches, last row).



Source:

https://www.v7labs.com/blog/vision-transformer-guide

https://en.wikipedia.org/wiki/Vision_transformer

Conclusion

- Benefits of pure transformers to computer vision:
  - computational efficiency
  - scalability (possibility to train large models)
- Matches or exceeds the state of the art

Outlook

- Apply ViT to other computer vision tasks
- continue exploring self-supervised pre-training methods.
- Further scaling would likely lead to improve performance.

References

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021).
An image is worth 16x16 words: Transformers for image recognition at scale.
arXiv. https://arxiv.org/abs/2010.11929
Github. https://github.com/google-research/vision_transformer

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017).
Attention Is All You Need.
arXiv. https://arxiv.org/abs/1706.03762

Silva, Thalles Santos (2023).
An Intuitive Introduction to the Vision Transformer
Github. https://sthalles.github.io/an-intuitive-introduction-to-the-vision-transformer/