Hochschule Konstanz
Technik, Wirtschaft und Gestaltung

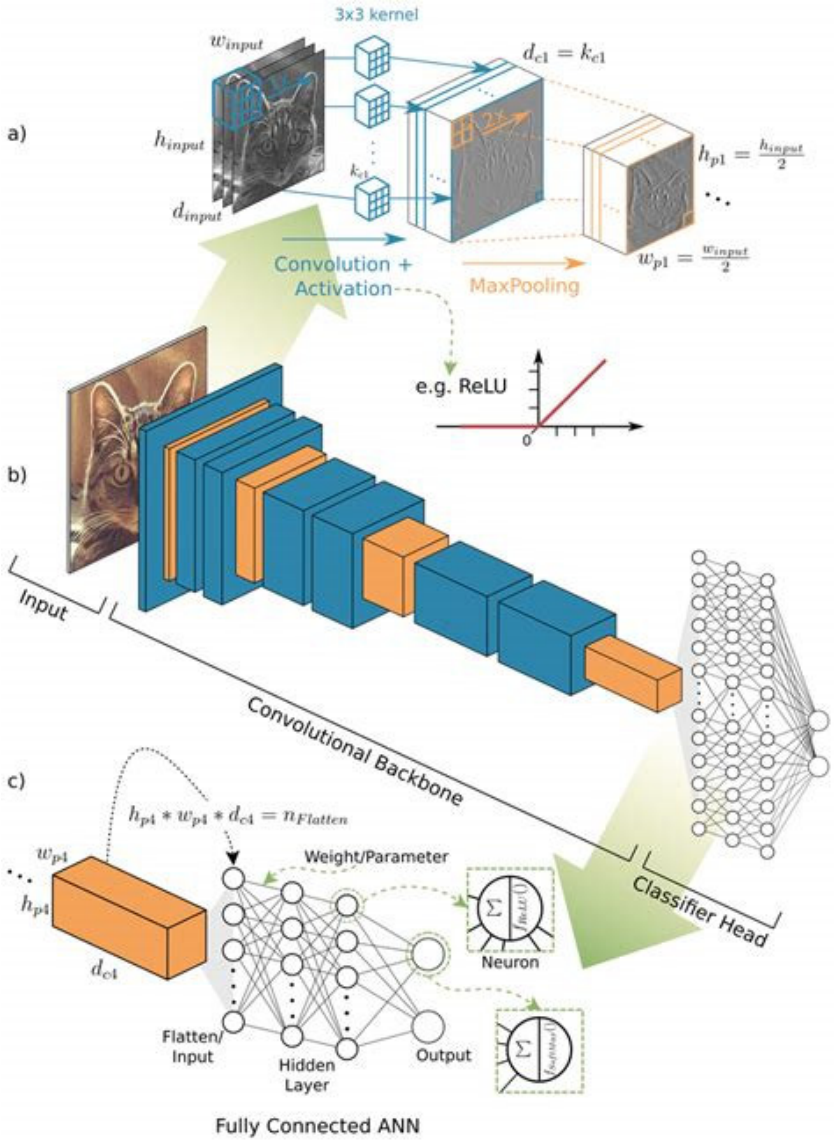
IOS
INSTITUT FÜR OPTISCHE SYSTEME

Universität Konstanz

Brownbag on Vision Image Transformer (ViT) Networks

Matthias Hermann, Institute for Optical Systems

Convolutional neural networks (CNNs)



Hoeser, T. et. Al. (2020). Object Detection and Image Segmentation with Deep Learning on Earth Observation Data: A Review-Part I: Evolution and Recent Trends.

Vision Transformers (ViT) outperform CNN

	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	$88.4/\mathbf{88.5}^*$
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).

Dosovitskiy, A., et. Al (2020). An image is worth 16x16 words: Transformers for image recognition at scale.

Vision image transformer (ViT) architecture

Dosovitskiy, A., et. Al (2020). An image is worth 16x16 words: Transformers for image recognition at scale. / https://theaisummer.com/vision-transformer/

Content

- CNNs and ViT
- Images, patches and sequences
- Attention is all you need
- The multi-head self-attention layer
- Stacking multiple layers and classification
- Closing notes

Transforming images into patches

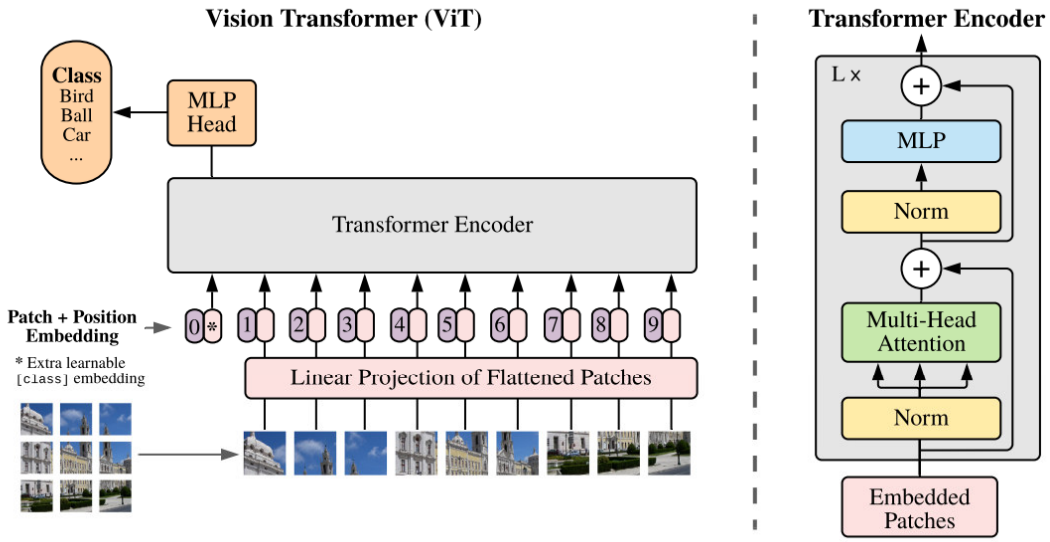
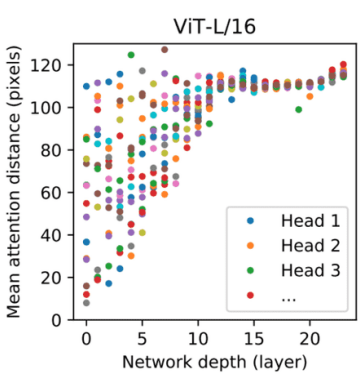


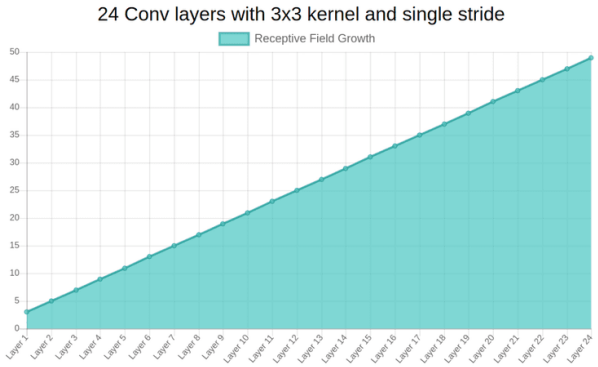
Figure 1: Model overview. We split an image into fixed-size patches, linearly embed each of them, add position embeddings, and feed the resulting sequence of vectors to a standard Transformer encoder. In order to perform classification, we use the standard approach of adding an extra learnable "classification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

Dosovitskiy, A., et. Al (2020). An image is worth 16x16 words: Transformers for image recognition at scale.

Why is attention all we need?
Attention distance and visualization.

The model attends to image regions that are semantically relevant for classification.



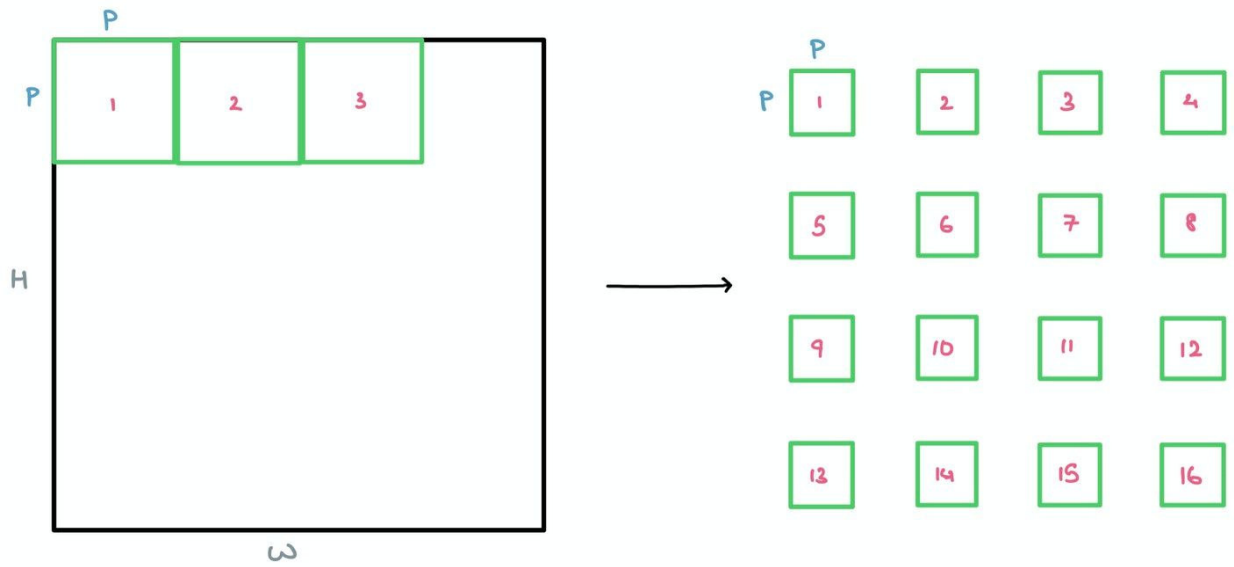


Attention distance was computed as the average distance between the query pixel and the rest of the patch, multiplied by the attention weight.

How the vision attention works

1. Split an image into patches
2. **Flatten the patches**
3. Produce lower-dimensional linear embeddings from the flattened patches
4. Add positional embeddings
5. Feed the sequence as an input to a standard transformer encoder
6. Pretrain the model with image labels (fully supervised on a huge dataset)
7. Fine-tune on the downstream dataset for image classification

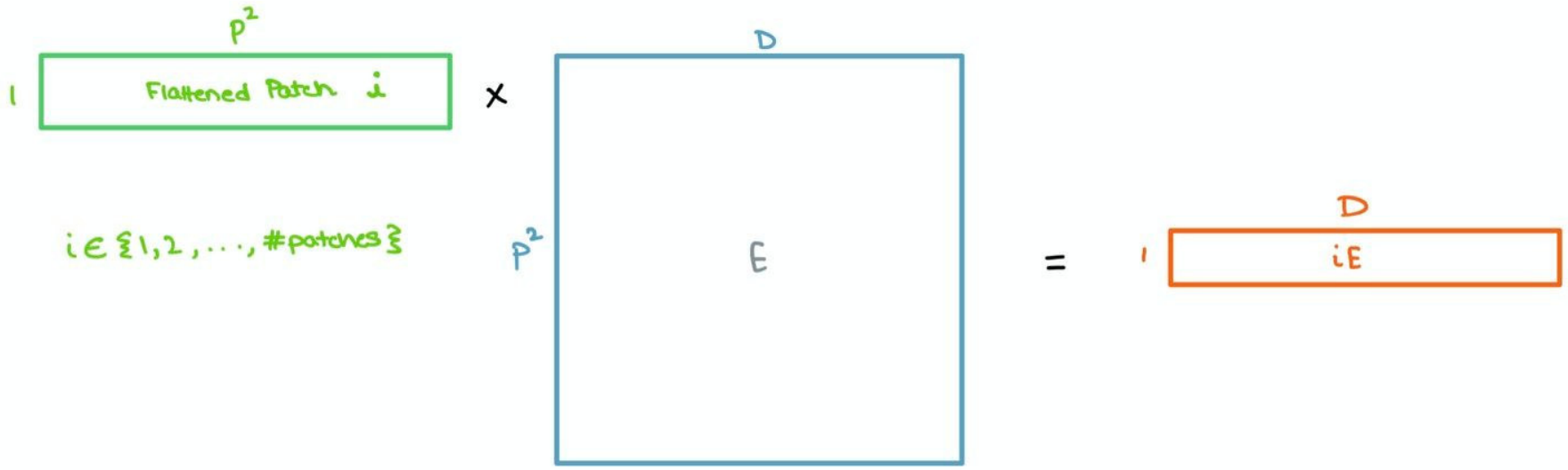
Transform image to patches



- These patches are then flattened and sent through a single Feed Forward layer to get a linear patch projection.
- This feed forward layer contains the embedding matrix **E**.
- Note that this is equivalent to a standard 2D-convolution with stride same as kernel size.

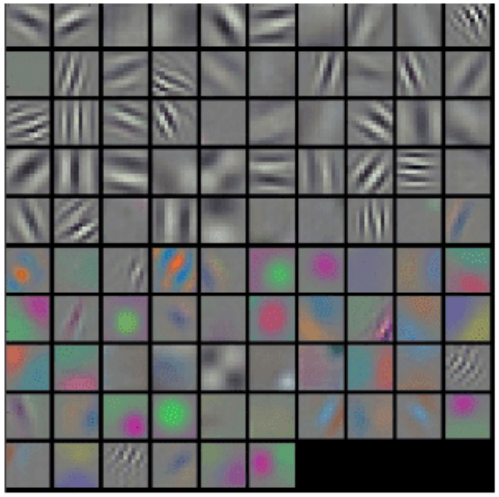
https://blog.paperspace.com/vision-transformers/

Single D-dimensional linear layer projection

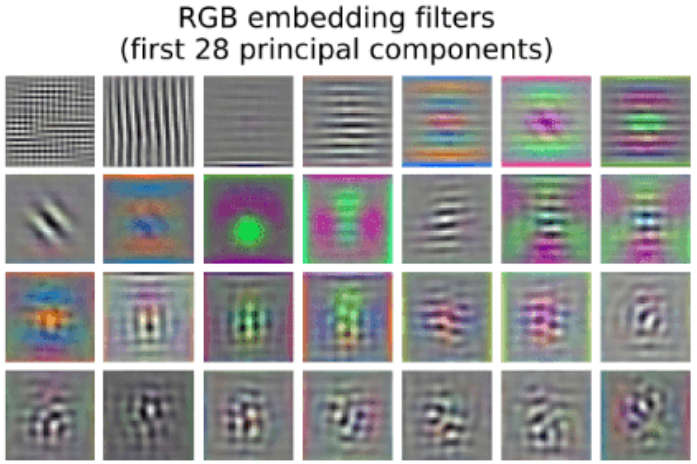


Linear embeddings look like CNNs

Alexnet 1st conv filters

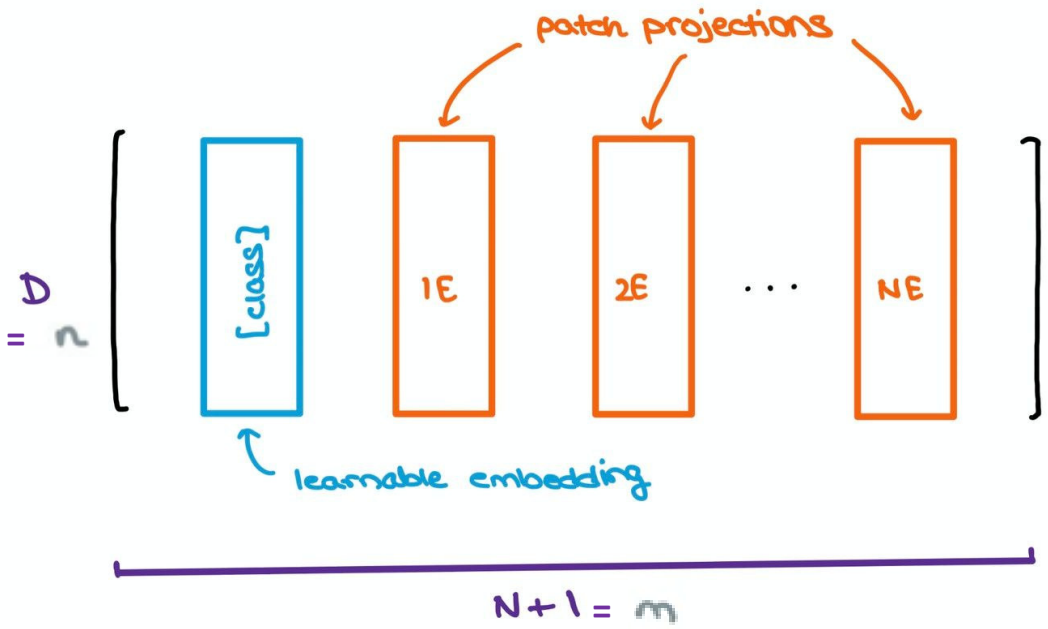


ViT 1st linear embedding filters



Dosovitskiy, A., et. Al (2020). An image is worth 16x16 words: Transformers for image recognition at scale.

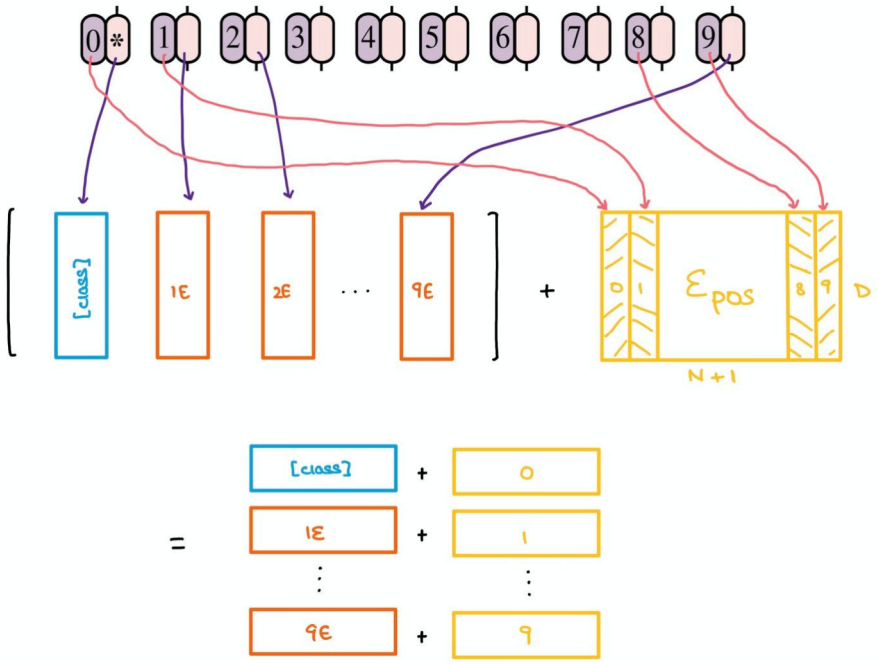
Add class token



- For classification, the authors took inspiration from the original BERT paper by concatenating a learnable (randomly initialized) [class] embedding with the other patch projections.
- The number of tokens m is then N+1.

https://blog.paperspace.com/vision-transformers/

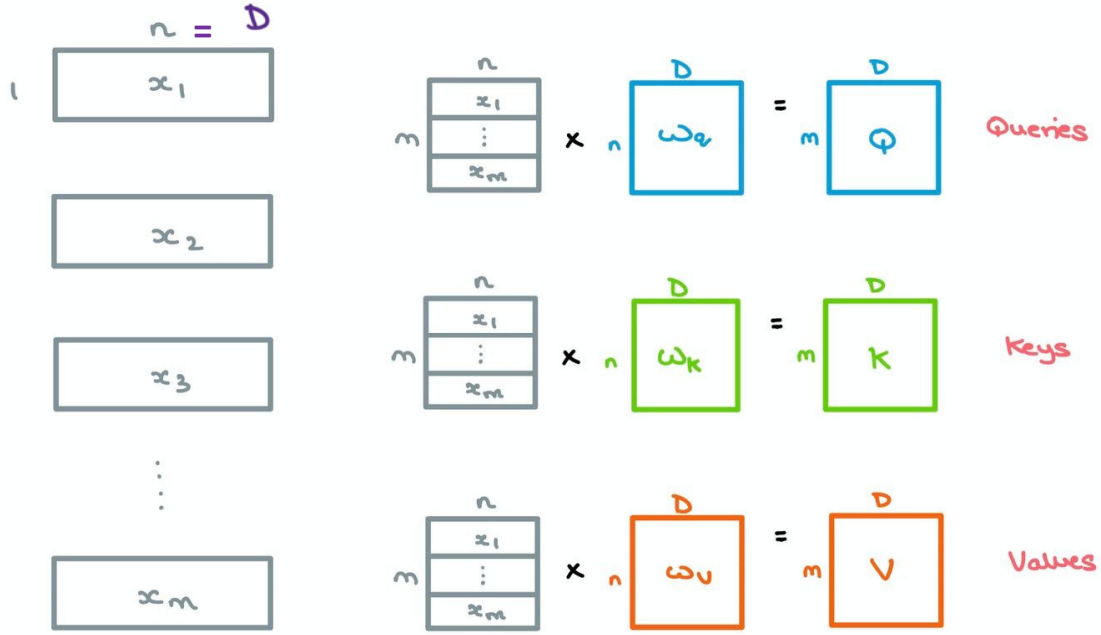
Add position token



- Another problem with Transformers is that the order of a sequence is not enforced naturally since data is passed in at a batch.
- The original Transformer paper suggests using Positional Encodings/Embeddings that establish an order in the inputs.

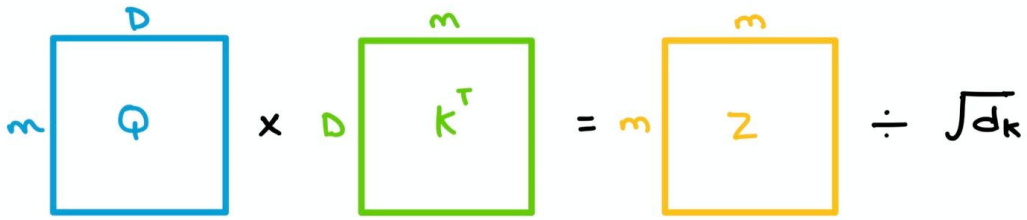
https://blog.paperspace.com/vision-transformers/

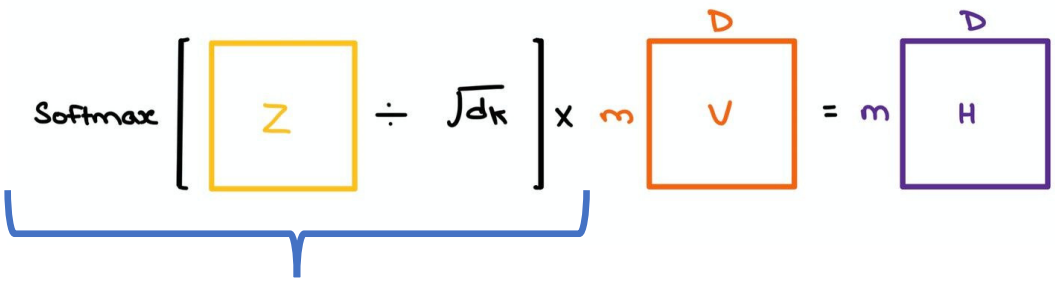
Image patches are now a sequence of tokens for a standard transformer



- The flattened patches (tokens) are transformed by another three different linear projection **Q**, **K**, and **V**.
- Think of three “split paths”.
- *n* and *D* typically identical.

The Scaled Dot Product Attention (SDPA) to prevent vanishing gradients

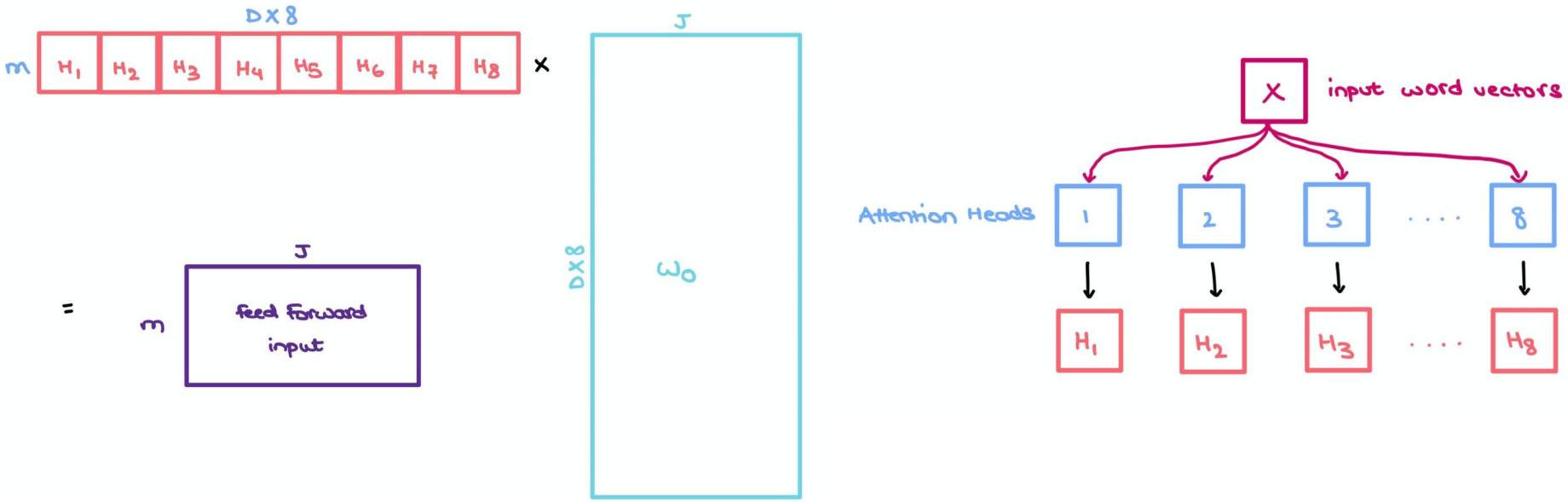




Attention weights **A**

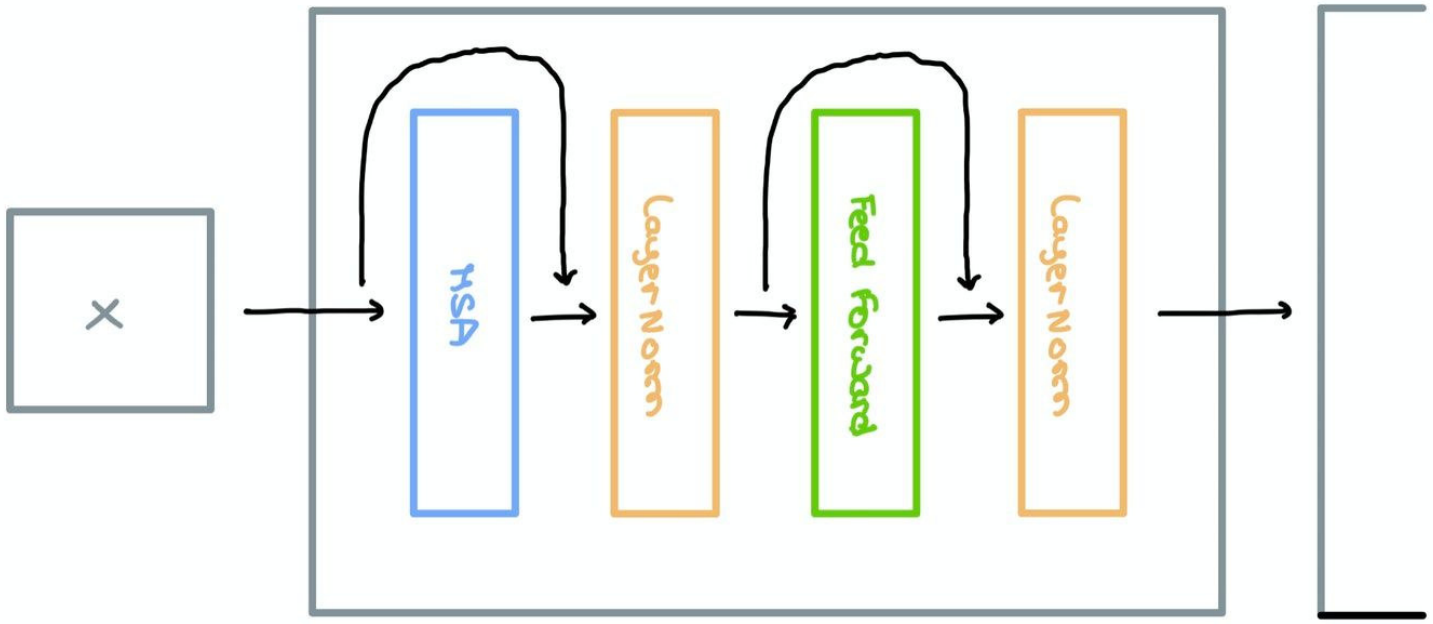
- The "Self" in Self Attention comes from comparing the tokens in a sequence with every other token in the same sequence.
- E.g., a sequence of 5 token vectors (i.e., 5 patches), the resulting attention matrix will have 5x5 entries learning the relationships between each word with the other words.
- Very similar to the Gram-matrix idea of the “Neural Style Transfer”.
- Has also “quadratic complexity”.

Multi-Head self-attention (MSA)

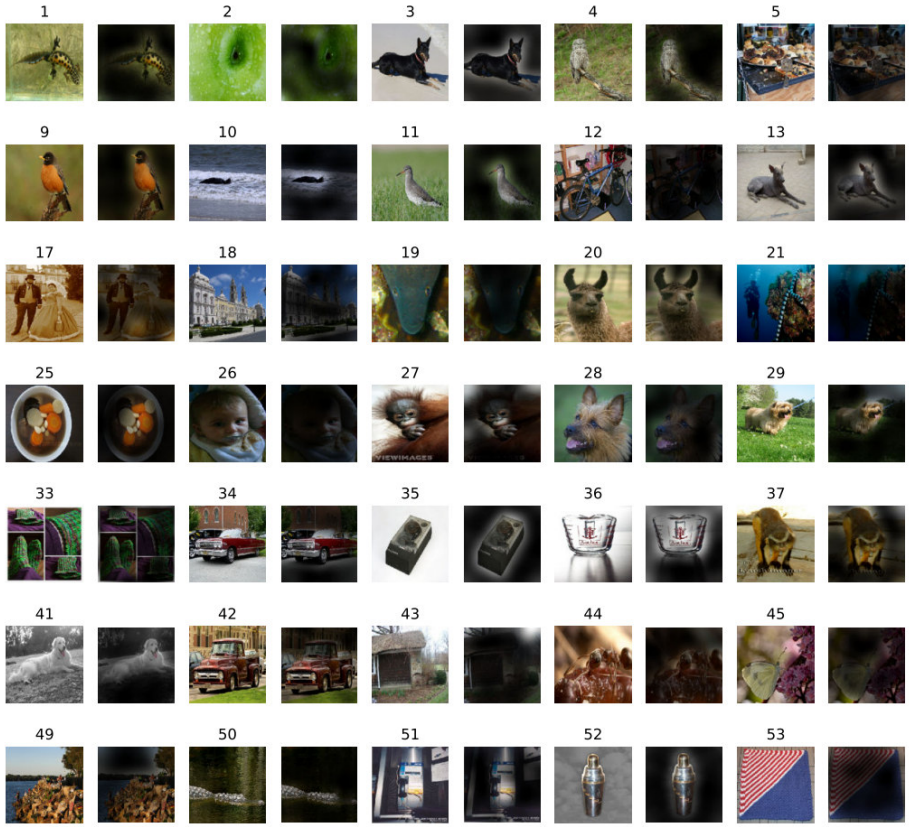


This resulting matrix captures the information of all the heads and can be sent to a linear feed forward layer with hidden layer dimension *J*.

Stack multiple “encoder blocks”



Attention map



- To compute maps of the attention from the output token to the input space Attention Rollout is used.
- Averaged attention weights of ViTL/16 across all heads and then recursively multiplied the weight matrices of all layers.

Abnar, S., & Zuidema, W. (2020). Quantifying attention flow in transformers. *arXiv preprint arXiv:2005.00928*.

How the vision attention works

1. Split an image into patches
2. Flatten the patches
3. Produce lower-dimensional linear embeddings from the flattened patches
4. Add positional embeddings
5. Feed the sequence as an input to a standard transformer encoder
6. Pretrain the model with image labels (fully supervised on a huge dataset)
7. **Finetune on the downstream dataset for image classification**

Classifier works on top of the “class token” only

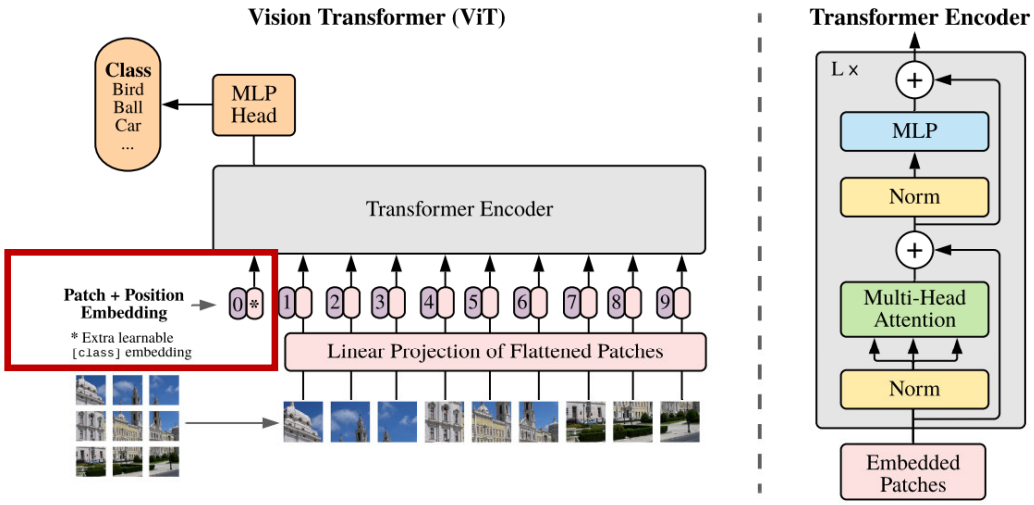
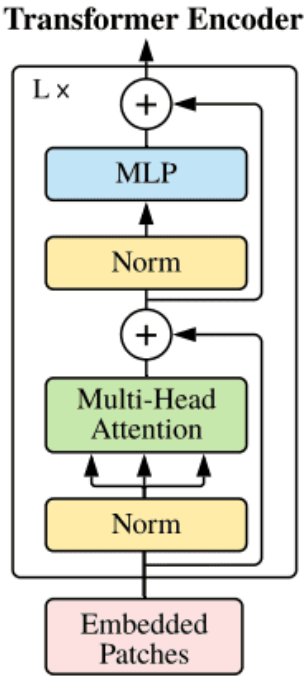


Figure 1: Model overview. We split an image into fixed-size patches, linearly embed each of them, add position embeddings, and feed the resulting sequence of vectors to a standard Transformer encoder. In order to perform classification, we use the standard approach of adding an extra learnable “classification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

How “big” are the models

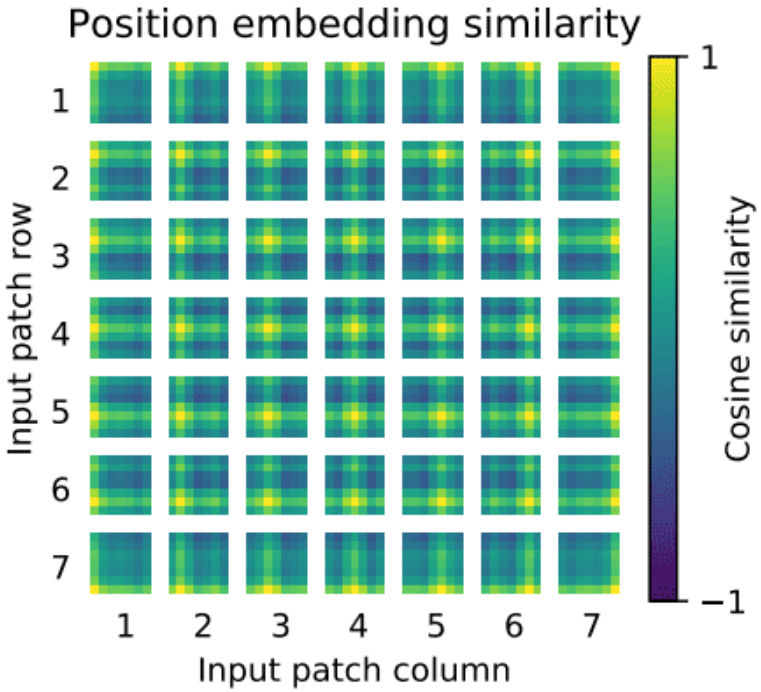


Model	Layers	Hidden size D	MLP size	Heads	Params
ViT-Base	12	768	3072	12	86M
ViT-Large	24	1024	4096	16	307M
ViT-Huge	32	1280	5120	16	632M

Model	2D-CNN	3D-CNN	Semi-CNN	
	Params	Params	Pre-Trained Params	Total Params
VGG-16	134.7 M	179.1 M	5.3 M	82.2 M
ResNet-18	11.4 M	33.3 M	0.4 M	31.7 M
ResNet-34	21.5 M	63.6 M	0.8 M	60.5 M
ResNet-50	23.9 M	46.4 M	0.9 M	45.8 M
ResNet-101	42.8 M	85.5 M	0.9 M	84.8 M
ResNet-152	58.5 M	117.6 M	1.4 M	115.6 M
DenseNet-121	7.2 M	11.4 M	0.8 M	10.4 M
DenseNet-169	12.8 M	18.8 M	0.8 M	17.9 M

Leong, M.C. et. al. (2020). Semi-CNN Architecture for Effective Spatio-Temporal Learning in Action Recognition.

Trainable position embeddings



- Similarity of position embeddings of ViT-L/32.
- Tiles show the cosine similarity between the position embedding of the patch with the indicated row and column and the position embeddings of all other patches.
- Even though many positional embedding schemes were applied, no significant difference was found.

Position embedding scheme is not relevant

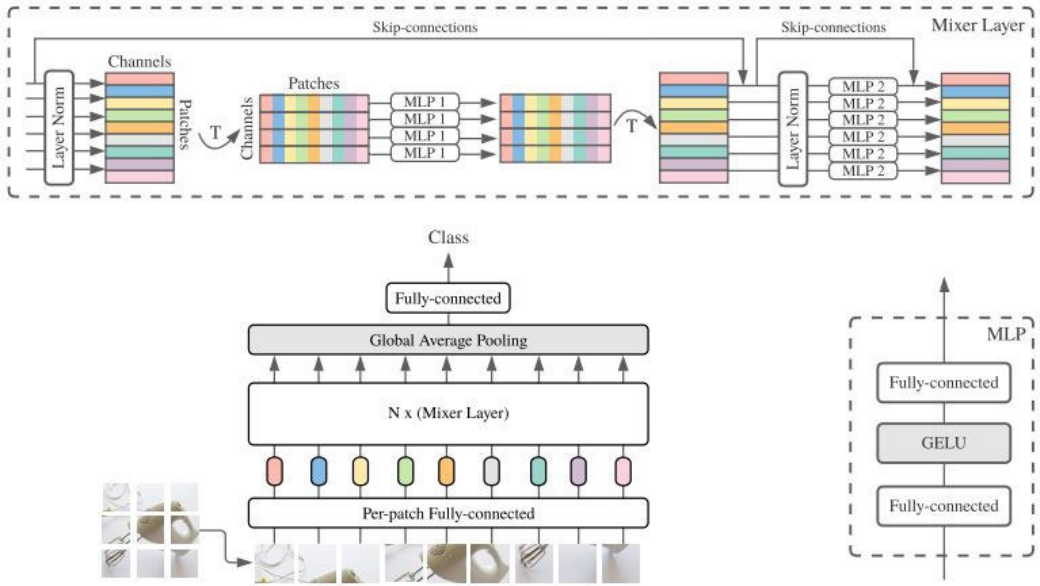
Pos. Emb.	Default/Stem	Every Layer	Every Layer-Shared
No Pos. Emb.	0.61382	N/A	N/A
1-D Pos. Emb.	0.64206	0.63964	0.64292
2-D Pos. Emb.	0.64001	0.64046	0.64022
Rel. Pos. Emb.	0.64032	N/A	N/A

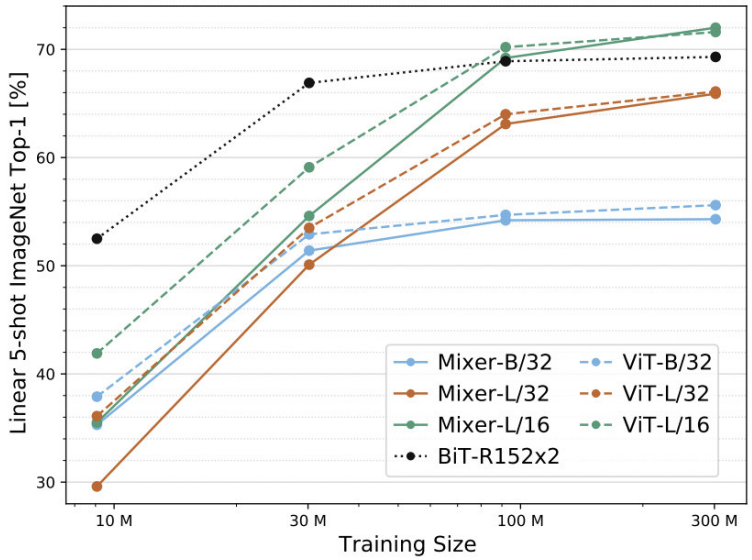
- **No positional** information:
  Considering the inputs as a bag of patches.
- (Default) **1-dimensional positional** embedding:
  Considering the inputs as a sequence of patches in the raster order.
- **2-dimensional positional** embedding:
  Considering the inputs as a grid of patches in two dimensions. In this case, two sets of embeddings are learned.
- **Relative positional** embeddings:
  Considering the relative distance between patches to encode the spatial information as instead of their absolute position.

Closing notes

- ViT is trained on datasets with more than **14M images** it can approach or beat state-of-the-art CNNs (ResNets or EfficientNets).
- Typically, the ViT is pretrained on the large dataset and then fine-tuned to small ones.
- Fine-tune may be done in **higher resolutions using 2D-interpolation** of the pre-trained position embeddings.
- Transformers aren’t mainstream yet.
- Main trick is intuitively, **solving a puzzle of 100 pieces** (patches) compared to 5000 pieces (pixels).

Updated version called: MLP-Mixer





Tolstikhin, I. O., et. al. (2021). Mlp-mixer: An all-mlp architecture for vision.

Thanks.

Matthias Hermann

Hochschule Konstanz
Institute for Optical Systems

matthias.hermann@htwg-konstanz.de

www.ios.htwg-konstanz.de

..novelty or surprise?

Berlyne D. E. (1960). Conflict, Arousal, and Curiosity.

References

https://blog.paperspace.com/vision-transformers/

https://theaisummer.com/vision-transformer/

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Hoeser, T. et. Al. (2020). Object Detection and Image Segmentation with Deep Learning on Earth Observation Data: A Review-Part I: Evolution and Recent Trends.

Leong, M.C. et. al. (2020). Semi-CNN Architecture for Effective Spatio-Temporal Learning in Action Recognition.

Abnar, S., & Zuidema, W. (2020). Quantifying attention flow in transformers. arXiv preprint arXiv:2005.00928.

Tolstikhin, I. O., Houlsby, N., Kolesnikov, A., Beyer, L., Zhai, X., Unterthiner, T., ... & Dosovitskiy, A. (2021). Mlp-mixer: An all-mlp architecture for vision. Advances in Neural Information Processing Systems, 34.