# Attention und Transformer Netzwerke

1. Attention Konzept
2. **Transformer Architektur**

# Vision Transformer

braucht viele Daten, weil 2-D Bild erst 1-D reduziert wird und dann Info verloren geht: Ergebnisse trotzdem gut

Multi-Layer-Perceptron

Positional Embedding

D: Dimension des Patch-Embeddings (ist beliebig definiert) = Hyperparameter

Class
Bird
Ball
Car
...

MLP Head

Multi-Head Attention

Transformer Encoder

Patch + Position Embedding

* Extra learnable [class] embedding

0* 1 2 3 4 5 6 7 8 9

Vektor der Dimension D

1 Vektor pro Patch

H. x W. x C

Linear Projection of Flattened Patches

Lernbare Embedding der Klasseninformation

=> y

Input in „Sequenz" aufteilen wie Satz bei Sprach-Tokens

↳ Zerlegung des Bildes in Patches

=> Sequenzsimulation

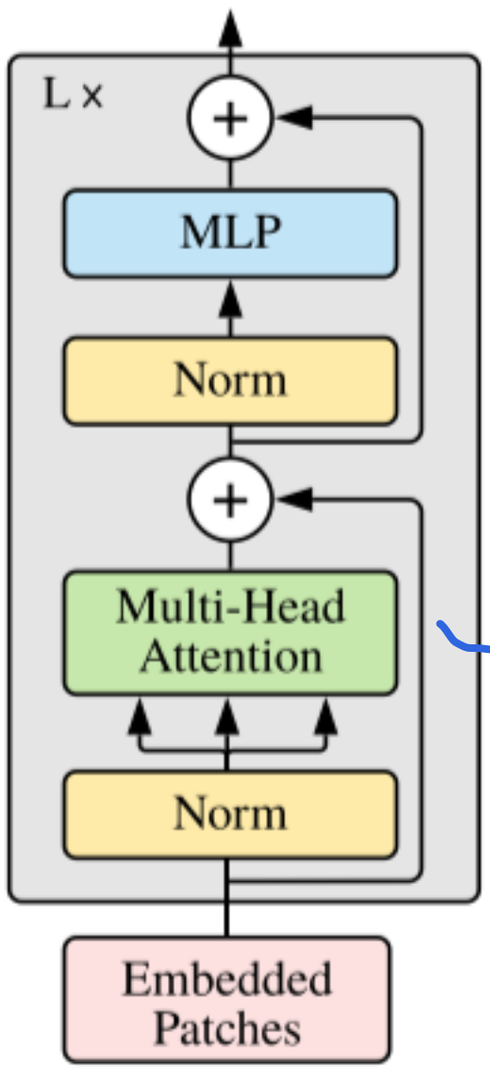
↳ Flattening => 1 Vektor

# Vision Transformer

MLP => 1 hidden Layer bei Pre-Training => 1 Layer bei Fine-Tuning

Linear

Transformer Encoder




→ wiederholt sich L mal

→ Residual Connection eingebaut

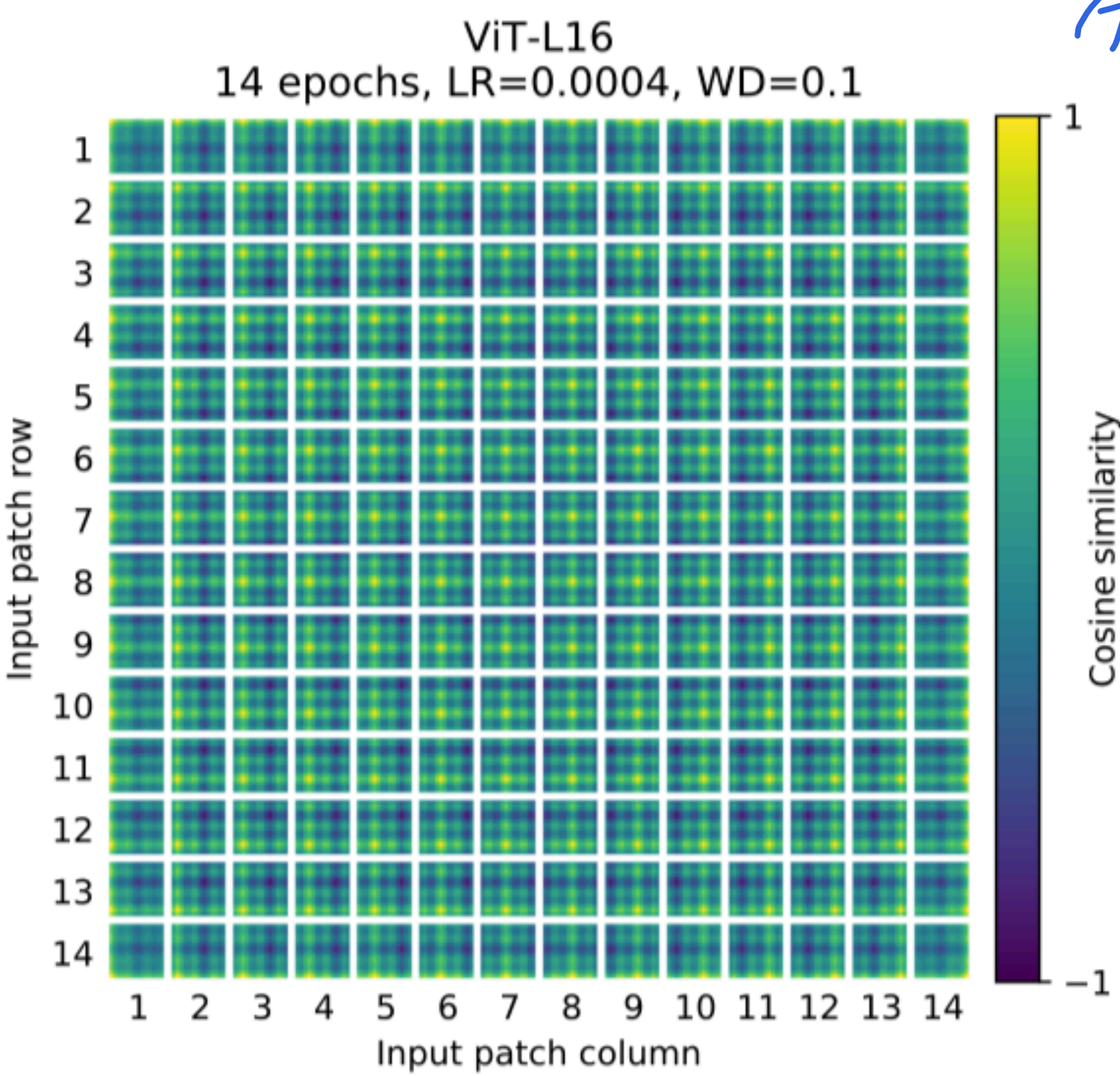
2 Schritte des Trainings

1) Pre-Training – nicht hochaufgelöste Bilder

2) Fine-Tuning – hochaufgelöste Bilder

© Dosovitskiy et al., 2021

# Vision Transformer




Abstände der gelernten Positional Embeddings

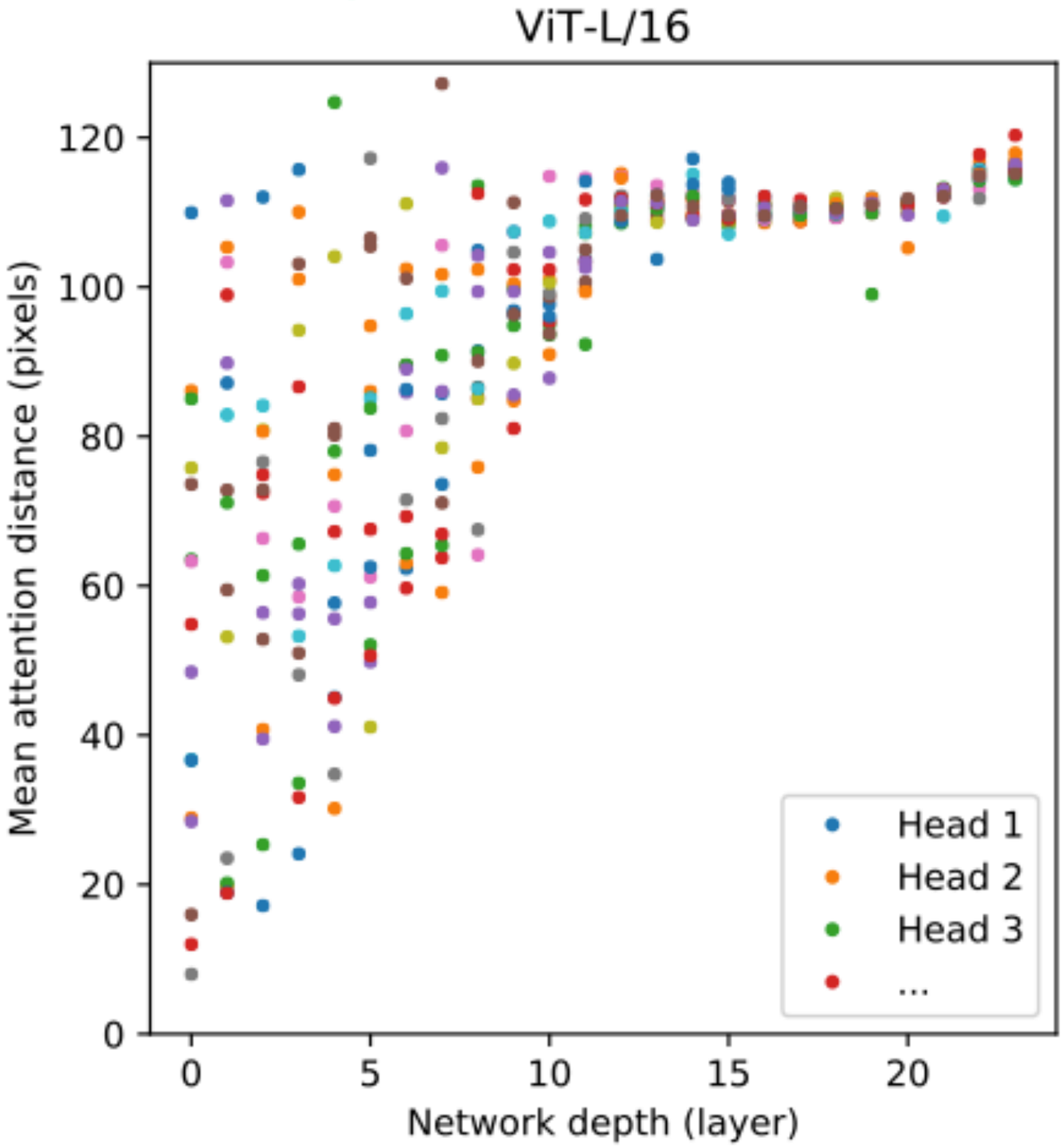
⇒ räumlich nahe Patches haben ähnliche Positional Embeddings

Zusammenhang kaptselle zwischen räumlich weit entfernten Dingen

© Dosovitskiy et al., 2021

# Vision Transformer

Neuronal ungesehen




-> analog to receptive fields bei CNNs

-> große Abstände schon bei den tiefen Layern relevant

# Vision Transformer






-> semantisch relevante Bildregionen
werden als wichtig erachtet

© Dosovitskiy et al., 2021