Transformers #2: Transformers and Vision

9 March 2021

Federico Baldassarre

Announcements 📅

- 23 March
  - Transformers for NLP
  - Youssef
- 6 April
  - Transformers application, other domains and alternatives
  - Yonk and Sofia

Outline 📋

- Main paper
  - An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale (Vision Transformer)
  - Discussion
- Secondary papers
  - End-to-End Object Detection with Transformers (DETR)
  - Discussion
  - Generative Pretraining from Pixels (iGPT)
  - Discussion
- General comments and discussion

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

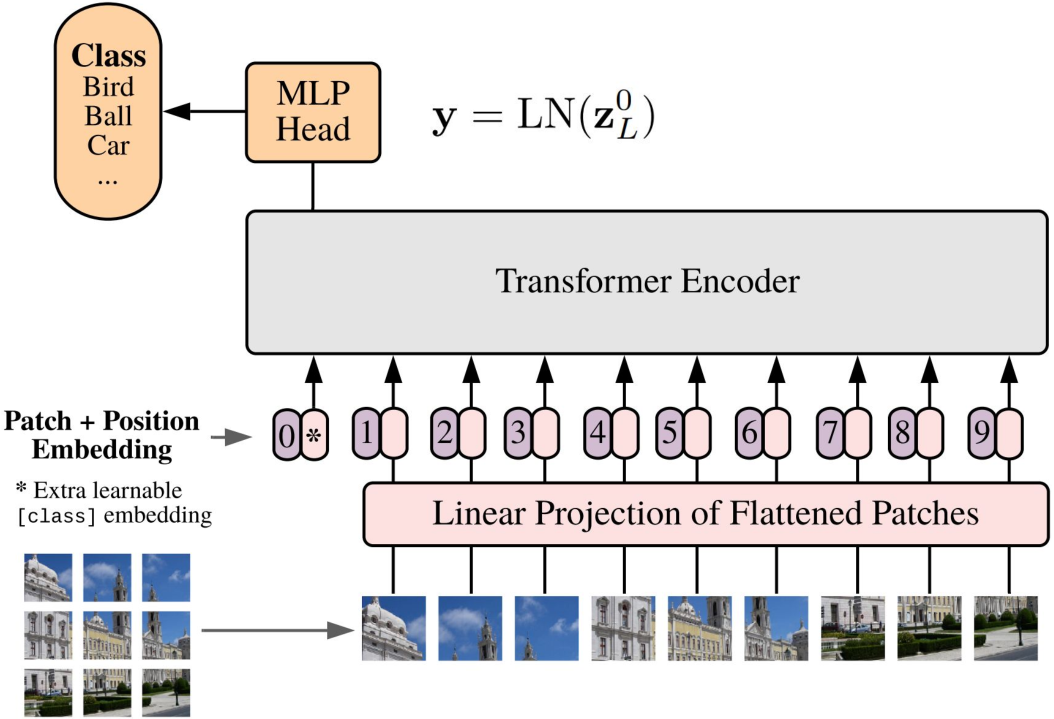
Dosovitskiy, Beyer, Kolesnikov, Weissenborn, Zhai, Unterthiner, Dehghani, Minderer, Heigold, Gelly, Uszkoreit, Houlsby (Google Research)

Published at ICLR 2021 (Oral)

Motivation

- From NLP: large-scale pre-training of Transformers
- Convolution is an established inductive bias. Can attention replace it?
- Related works using attention require specialized architectures

Approach: the Vision Transformer



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

$$\mathbf{z'}_\ell = \mathrm{MSA}(\mathrm{LN}(\mathbf{z}_{\ell-1})) + \mathbf{z}_{\ell-1}$$

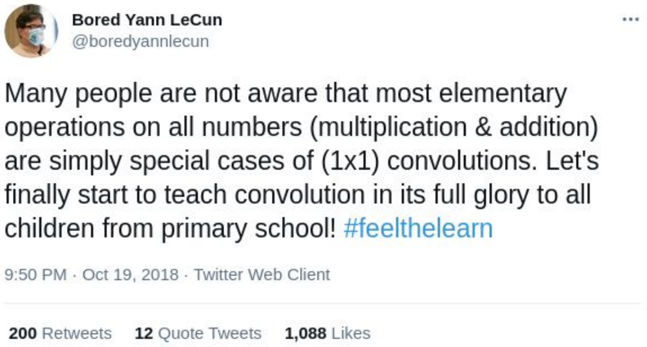
$$\mathbf{z}_\ell = \mathrm{MLP}(\mathrm{LN}(\mathbf{z'}_\ell)) + \mathbf{z'}_\ell,$$

$$\mathbf{z}_0 = [\mathbf{x}_{\text{class}};\, \mathbf{x}_p^1\mathbf{E};\, \mathbf{x}_p^2\mathbf{E};\, \cdots;\, \mathbf{x}_p^N\mathbf{E}] + \mathbf{E}_{pos}$$

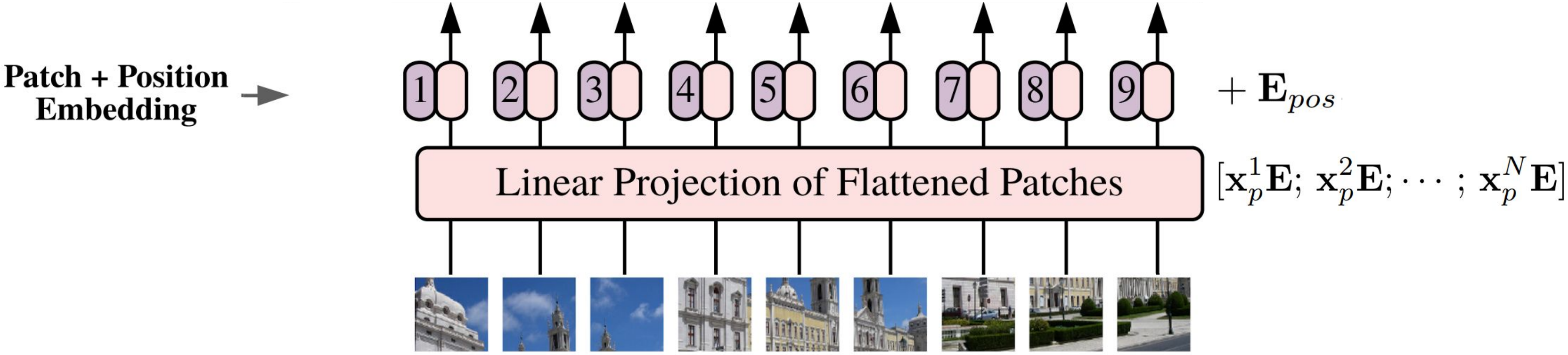
Image as 16x16 patches

$$[\mathbf{x}_p^1\mathbf{E};\, \mathbf{x}_p^2\mathbf{E};\, \cdots;\, \mathbf{x}_p^N\mathbf{E}]$$

- 16x16 patches ➔ sequence of tokens
- Weak 2D locality prior (bias #1)
- Convolution hiding in plain sight?

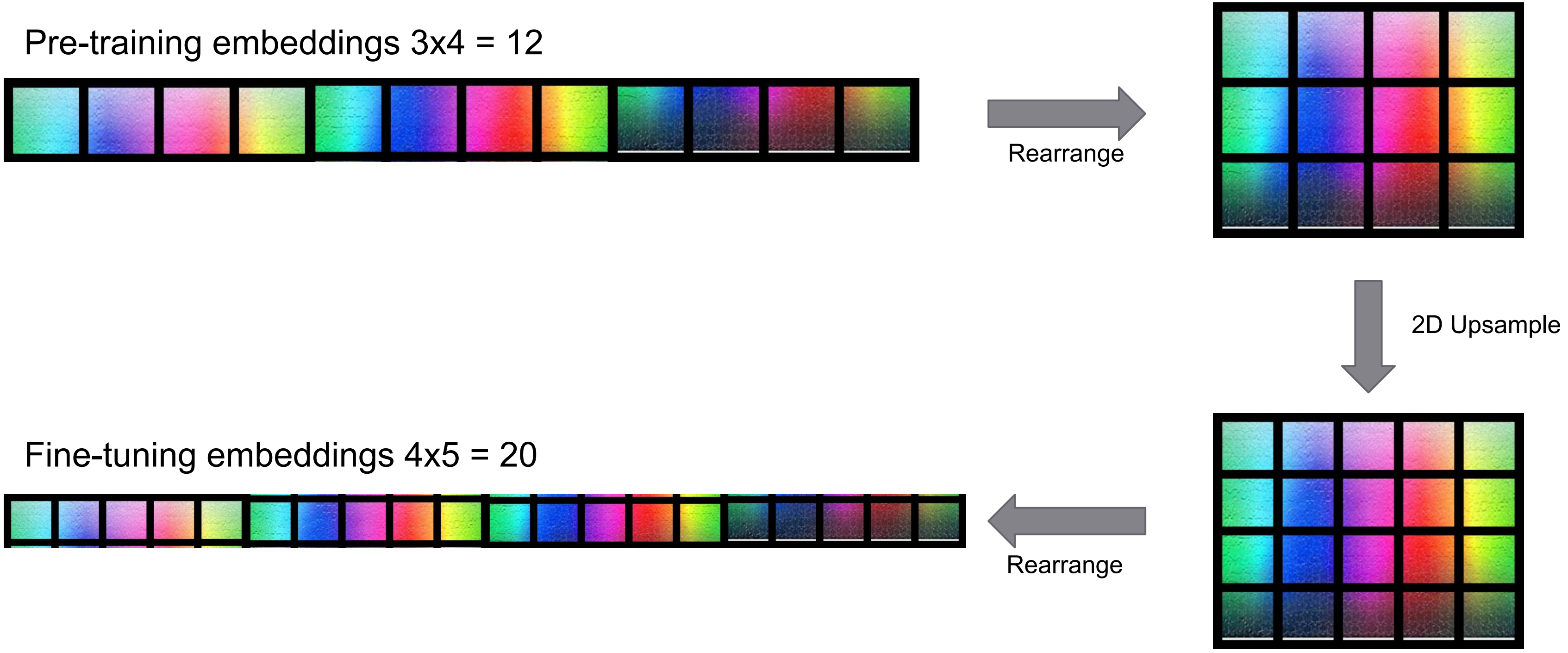


Reintroducing spatial information

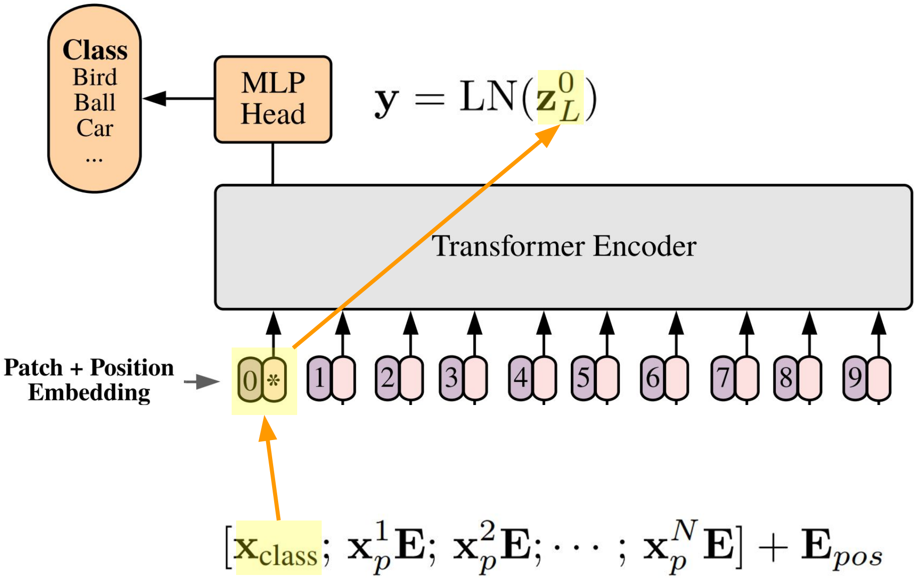


- Learned embedding for each patch position
- For inference at higher resolutions, embeddings are interpolated in a 2D grid (bias #2)

Embedding interpolation



Almost a standard Transformer



- Sequence:
  image patches + classification token
- Output class is predicted from the classification token
- During fine-tuning, the MLP head is replaced

Experiments

General setting:

- Pre-train on a large-scale supervised classification task (low resolution)
- Fine-tune on a specific classification task (high resolution)

Research questions:

- Comparison with CNNs (training cost, transfer accuracy)
- Interplay between convolutional bias and compute budget
- Inspection of learned weights and typical attention maps

Datasets

Pre-training

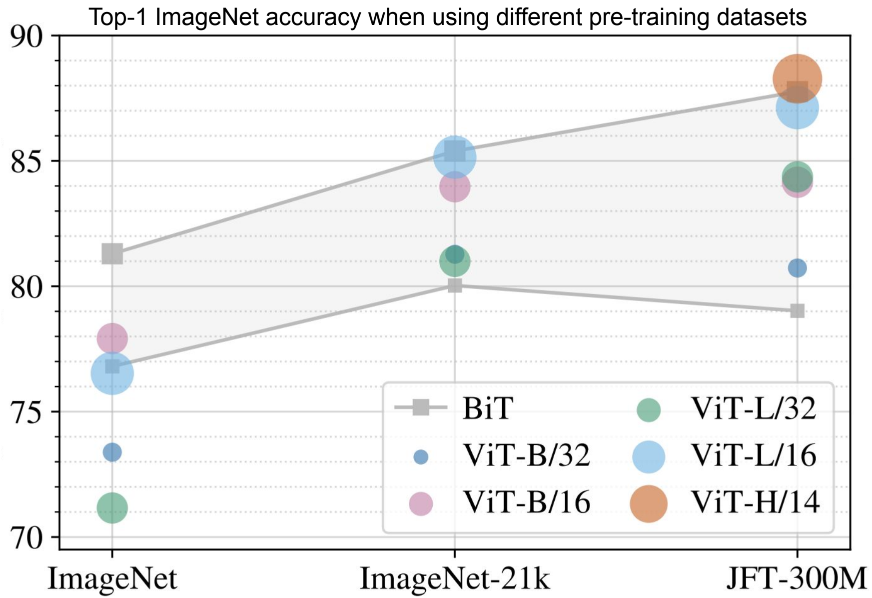
- ImageNet
  - 1k classes
  - 1.3M images
- ImageNet-21k
  - 21k classes
  - 14M images
- JFT
  - 18k classes
  - 303M images
  - *Private Google dataset*

Fine-tuning

- CIFAR 10/100
- ImageNet
- Oxford Pets/Flowers
- VTAB 19-task suite

Pre-training uses a lower resolution than fine-tuning. Both are supervised.

Comparison with CNNs



“Small” pre-train datasets

ResNet > Transformer

Larger pre-train datasets

Transformer ≈ ResNet

Saturation for bigger Vision Transformers not yet observed

Training time ⏱️

	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** $\pm$ 0.04	87.76 $\pm$ 0.03	85.30 $\pm$ 0.02	87.54 $\pm$ 0.02	88.4/88.5*
TPUv3-core-days	2.5k	0.68k	0.23k	9.9k	12.3k
Parameters	632M	307M	307M	928M	

^ BiT-L is trained on JFT: Big Transfer (BiT): General Visual Representation Learning, Kolesnikov et al.

Training cost $

Finally, the ViT-L/16 model pre-trained on the public ImageNet-21k dataset performs well on most datasets too, while taking fewer resources to pre-train: it could be trained using a standard cloud TPUv3 with 8 cores in approximately 30 days.

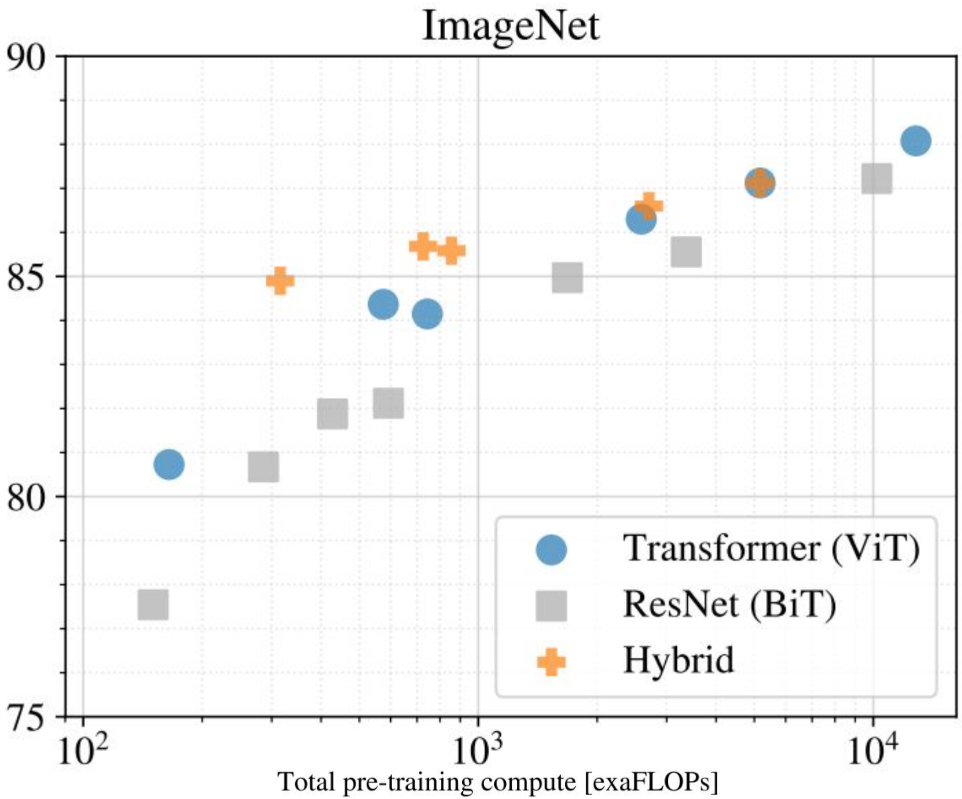
Iowa (us-central1)

Monthly Hourly

TPU type (v2)	v2 cores	Total memory	On-demand price (USD)	Preemptible price (USD)
v2-8	8	64 GiB	$3,285 / month	$986 / month
TPU type (v3)	v3 cores	Total memory	On-demand price (USD)	Preemptible price (USD)
v3-8	8	128 GiB	$5,840 / month	$1,752 / month

TPU prices only, virtual machine billed separately: Cloud TPU pricing

Convolutional bias vs. compute budget



Vision Transformer vs. Big Transfer

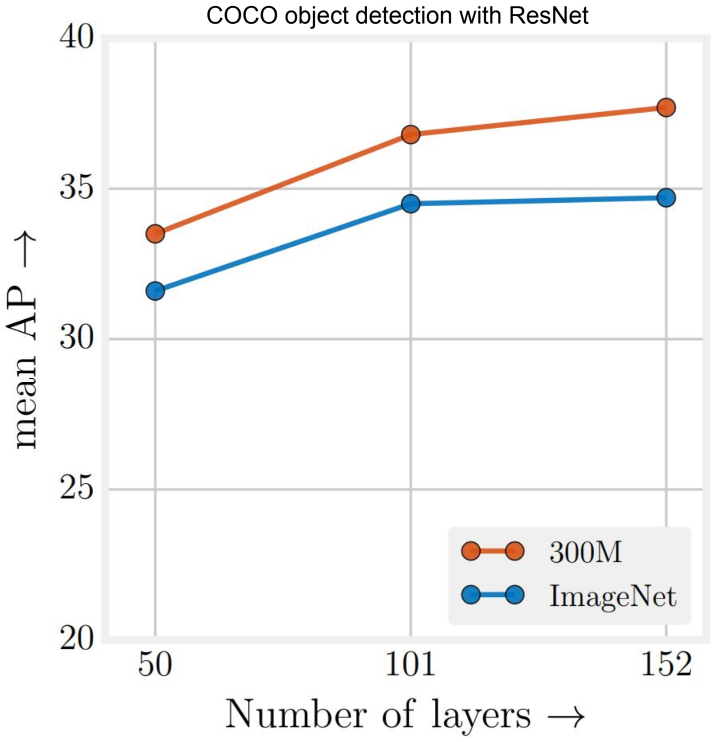
- Given similar budget, Transformers perform generally better than CNNs

Hybrid vs. pure transformer

- In low-compute regime, the convolutional bias helps
- With enough budget, the bias becomes unnecessary.

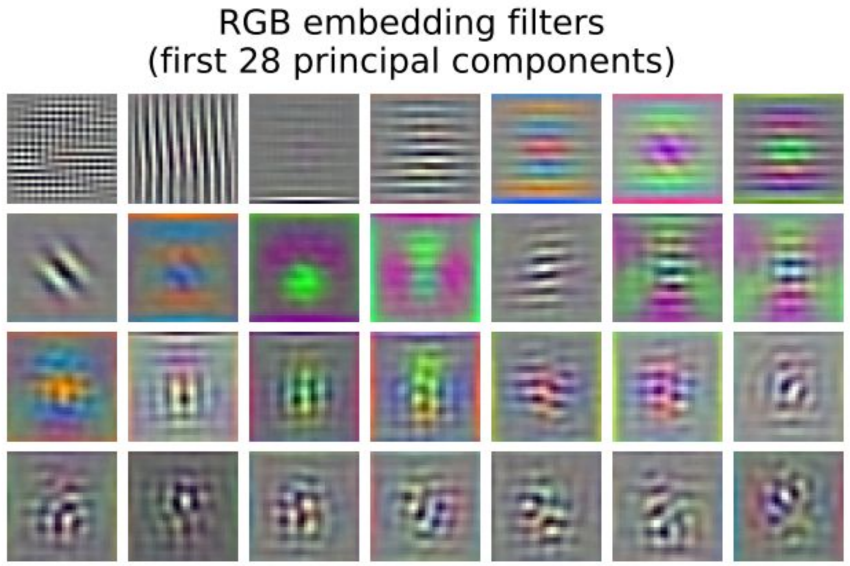
All models are pre-trained on JFT

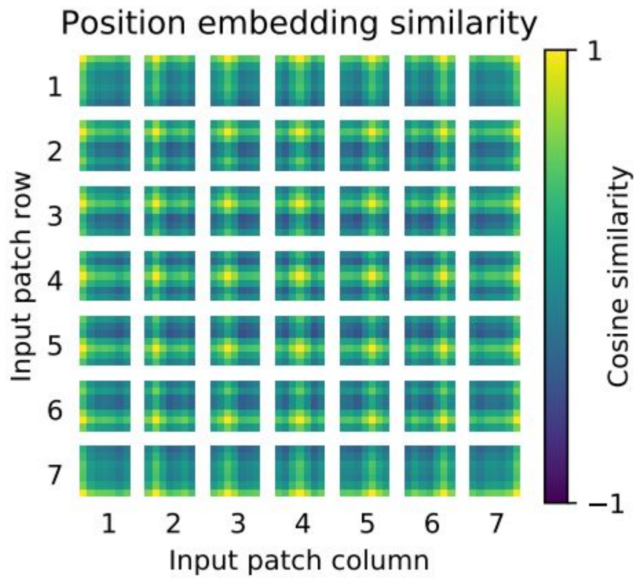
Discussion: just add data?



Our paper is an attempt to put the focus back on the data. The models seem to be plateauing but when it comes to the performance with respect to data – but modest performance improvements are still possible for exponential increases of the data. Another major finding of our paper is that having better models is not leading to substantial gains because ImageNet is no more sufficient to use all the parameters or their representational power.

Model inspection

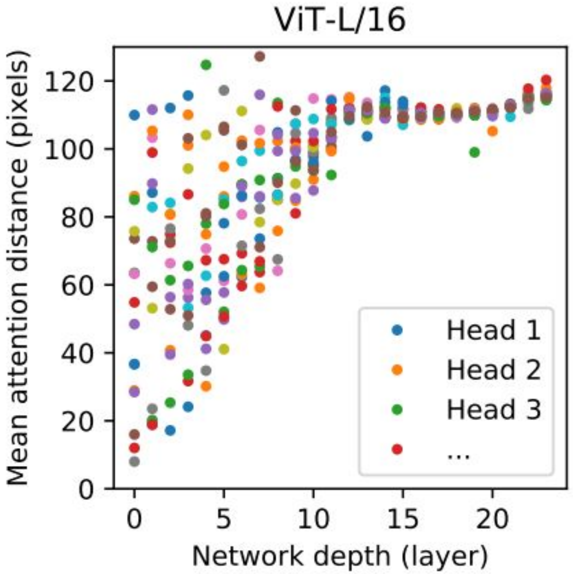


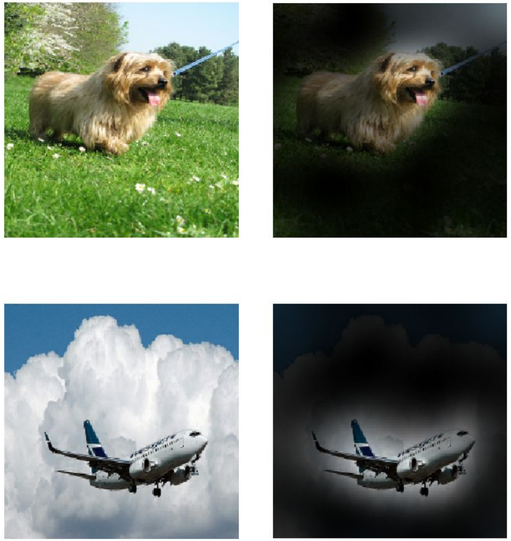


Learned linear projection filters
resemble low-level filters in CNNs

Learned position embeddings
encode 2D information

Model inspection





Learned attention patterns match the typical CNN receptive fields

Attention maps of the classification head have (seemingly) learned to localize objects

Conclusions

With large amounts of data:

- The convolutional bias can be dropped
- Weak 2D biases remain necessary
- Transformers are more efficient than CNNs

Future work:

- Other computer vision tasks, e.g. DETR
- ViT remains fully-supervised, see iGPT

Vision Transformer

Discussion

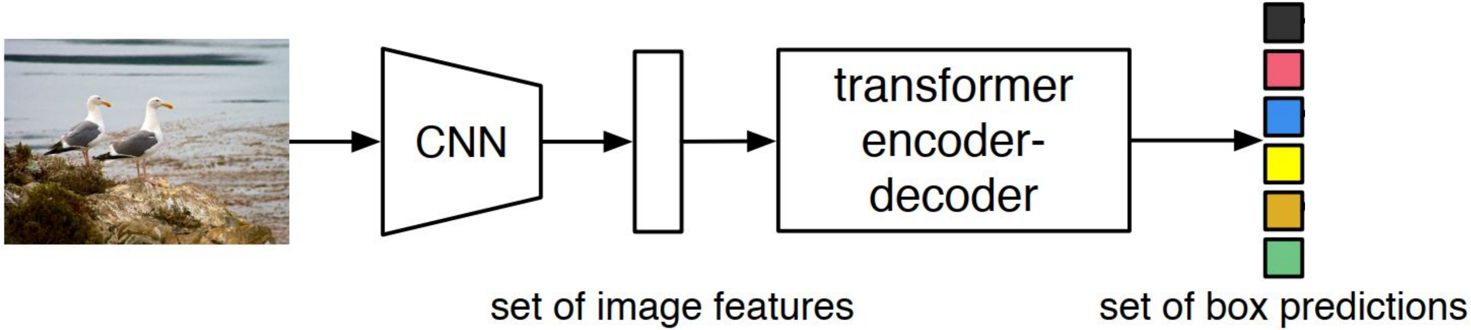
End-to-End Object Detection with Transformers

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, Sergey Zagoruyko (Facebook AI)

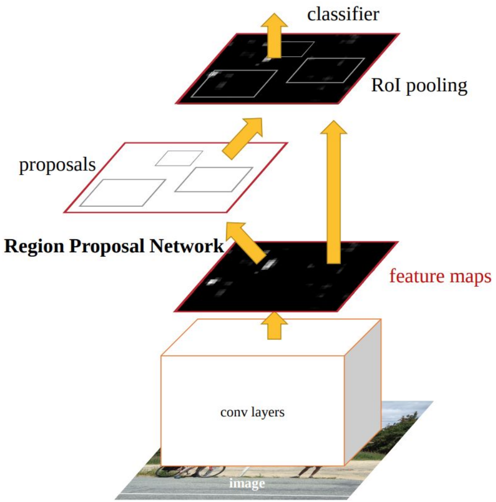
Published at ECCV 2020

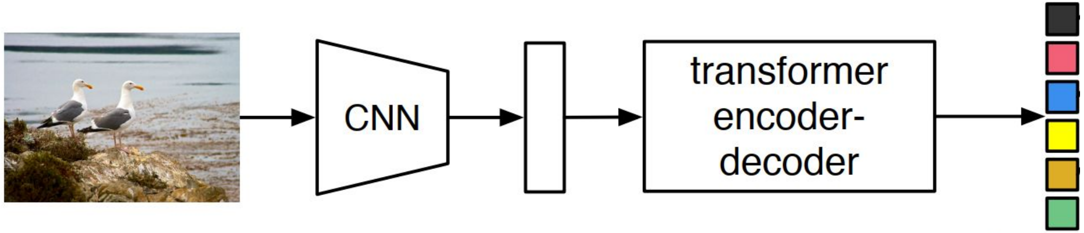
Ideas

- Object detection can be seen as a set-to-set problem
- Transformers are efficient set-to-set architectures



Goal: simplicity





Faster R-CNN:

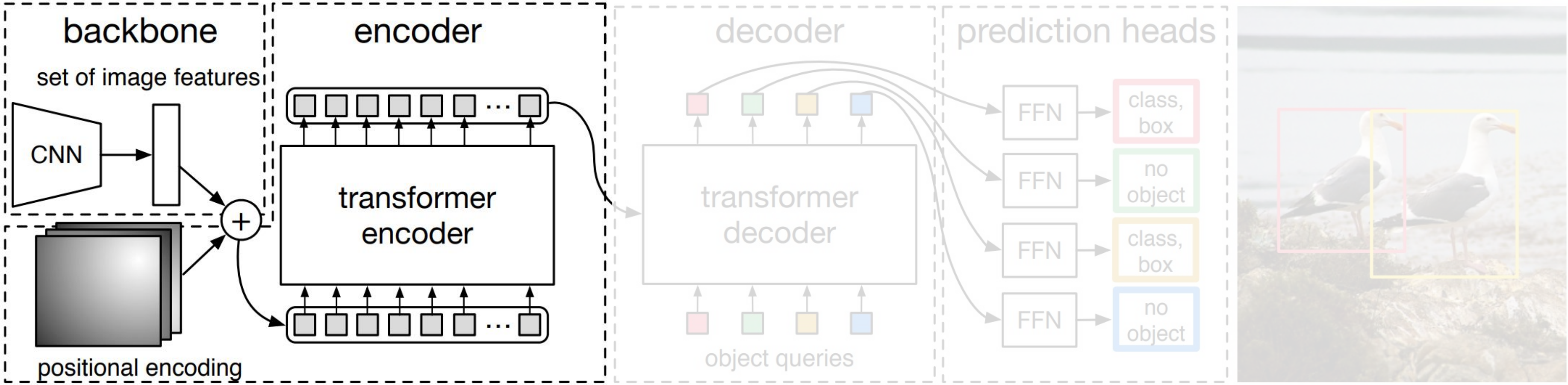
- Several components
- Detectron2 codebase is well-written but complex

DETR:

- Straightforward set-to-set prediction
- Off-the-shelf Transformer layers

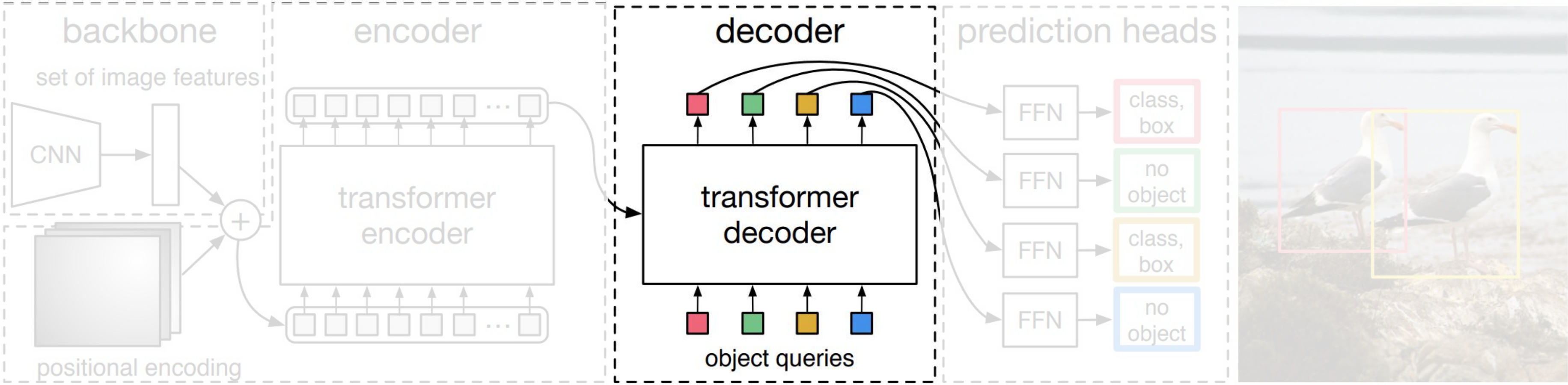
Source: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, Ren et al.

Feature extraction



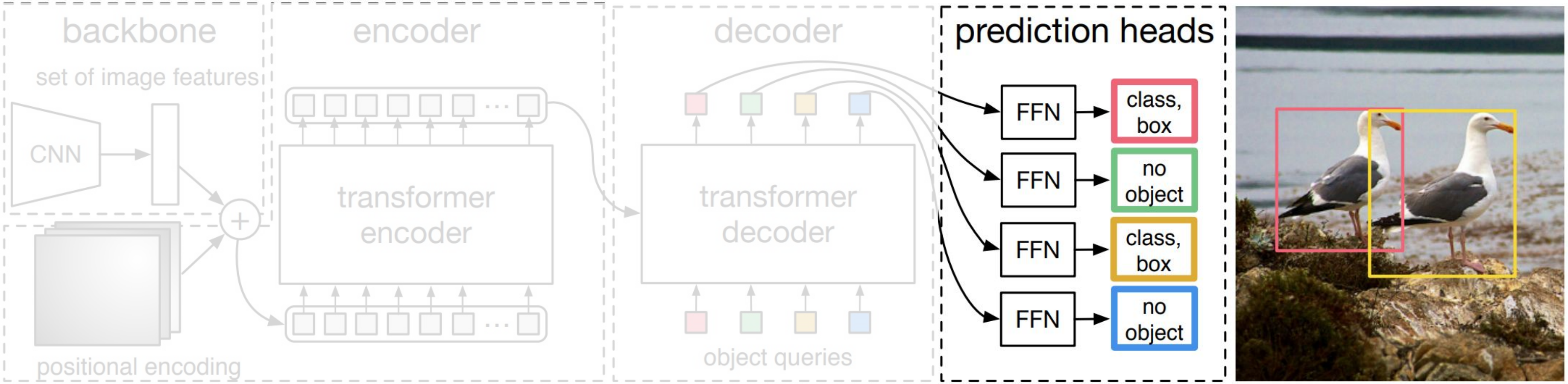
- Transformer encoder: a stack of self-attention layers
- Features from a CNN backbone
- 2D positional encoding

Set-to-set prediction



- Transformer decoder: object queries attend to image patches
- Parallel decoding (all at once, not autoregressive)

Matching loss



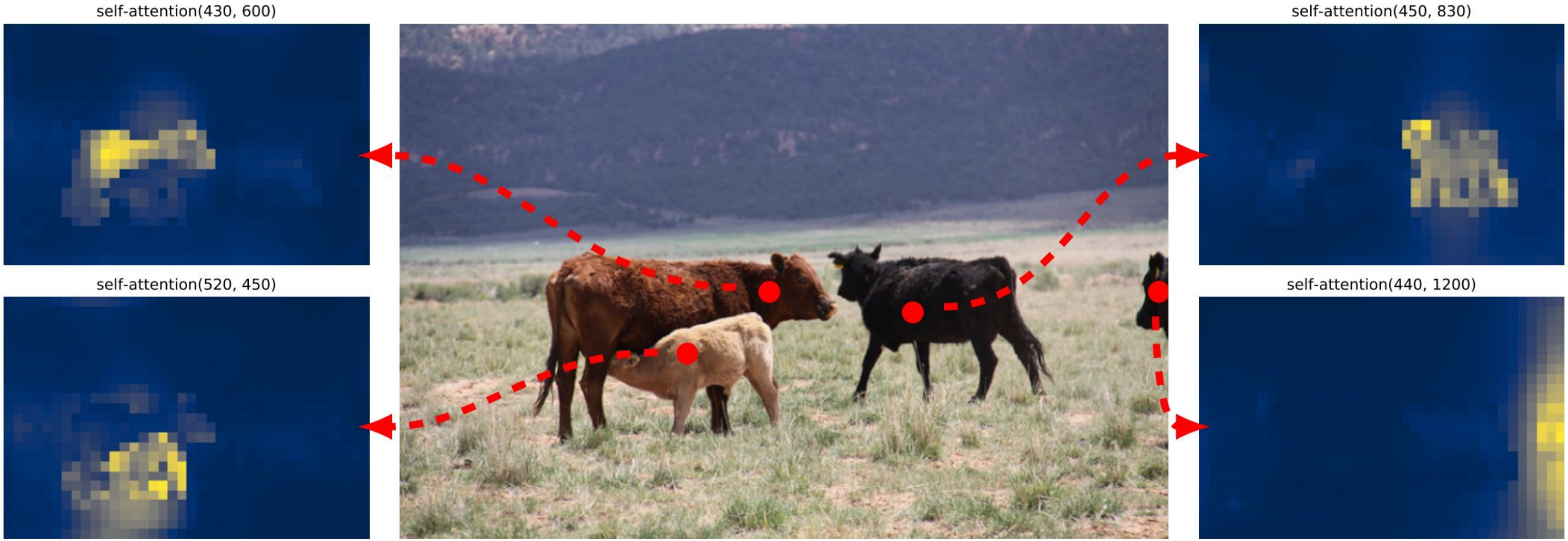
- Bipartite matching between predictions and ground-truth boxes
- Loss is a combination of object classification and box regression

Results

Model	GFLOPS/FPS	#params	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Faster RCNN-DC5	320/16	166M	39.0	60.5	42.3	21.4	43.5	52.5
Faster RCNN-FPN	180/26	42M	40.2	61.0	43.8	24.2	43.5	52.0
Faster RCNN-R101-FPN	246/20	60M	42.0	62.5	45.9	25.2	45.6	54.6
Faster RCNN-DC5+	320/16	166M	41.1	61.4	44.3	22.9	45.9	55.0
Faster RCNN-FPN+	180/26	42M	42.0	62.1	45.5	26.6	45.4	53.4
Faster RCNN-R101-FPN+	246/20	60M	44.0	63.9	**47.8**	**27.2**	48.1	56.0
DETR	86/28	41M	42.0	62.4	44.2	20.5	45.8	61.1
DETR-DC5	187/12	41M	43.3	63.1	45.9	22.5	47.3	61.1
DETR-R101	152/20	60M	43.5	63.8	46.4	21.9	48.0	61.8
DETR-DC5-R101	253/10	60M	**44.9**	**64.7**	47.7	23.7	**49.5**	**62.3**

- Competitive with highly-optimized Faster R-CNN models
- Detection of small objects needs to be improved (likely with a FPN)

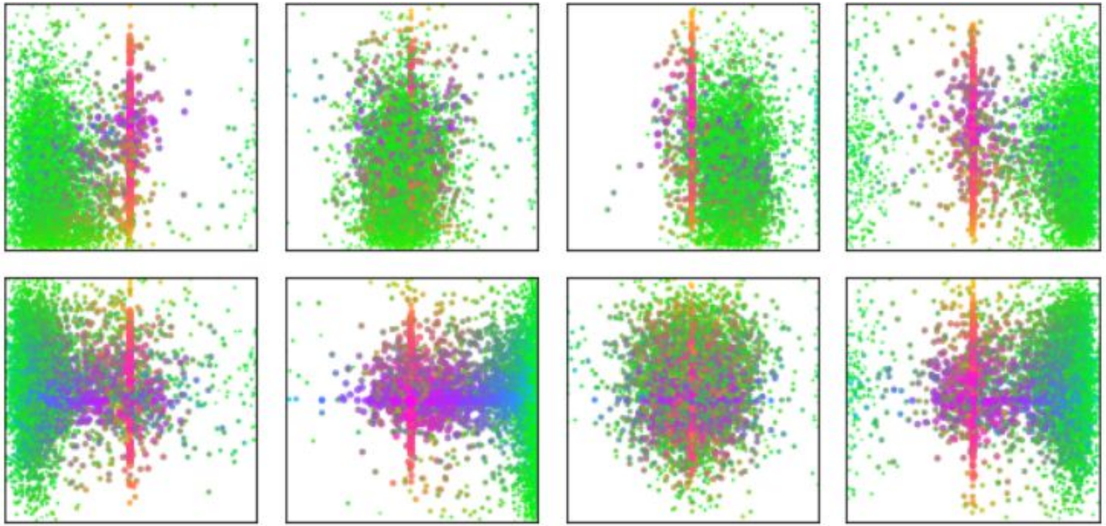
Encoder self-attention inspection



- Self-attention weights resemble object masks
- Encoder already builds a representation for object detection

Decoder object queries inspection

Where are the object boxes that are predicted by each object query?



Green: small boxes
Red: large horizontal boxes
Blue: large vertical boxes

- Specialised for certain sizes at certain locations
- “Is there an object here?”

Conclusions

Novel approach for object detection

- Set-to-set transformers with matching loss
- Drop hand-engineered components
- Attention maps might be useful for inspecting the model and panoptic segmentation

DETR

Discussion

Generative Pretraining from Pixels

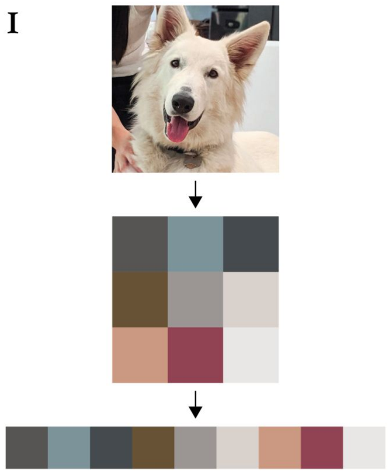
Mark Chen, Alec Radford, Rewon Child, Jeff Wu, Heewoo Jun, Prafulla Dhariwal, David Luan, Ilya Sutskever (OpenAI)

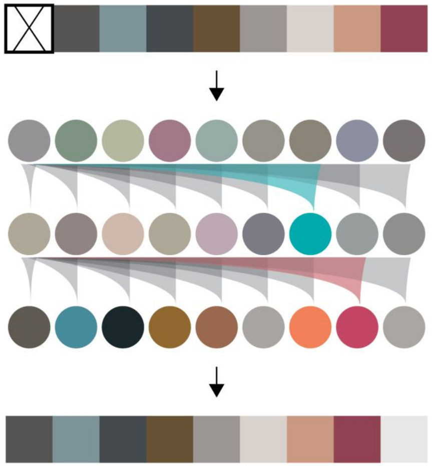
Published at ICML 2020

Ideas

- Pixel-by-pixel image generation using an autoregressive transformer
- Self-supervised pre-training, no labels needed
- Evaluate learned representation using linear probes

Next-pixel prediction





$$p(x) = \prod_{i=1}^{n} p(x_{\pi_i} | x_{\pi_1}, ..., x_{\pi_{i-1}}, \theta)$$

$$L_{AR} = \mathop{\mathbb{E}}_{x \sim X} [-\log p(x)]$$

- Each pixel is a token
- Resolution limitations
- Reduced color palette

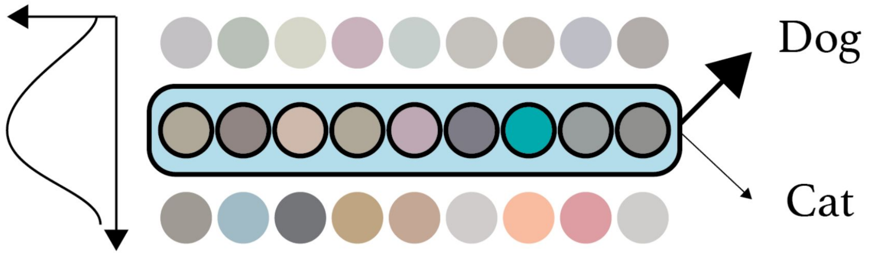
- Autoregressive architecture
- Standard sequence modeling objective

Linear probe

- Average over the latent codes of each pixel in one layer

$$f^l = \langle n_i^l \rangle_i$$

- Train a linear classifier over the layer representation



Datasets

Self-supervised pre-training

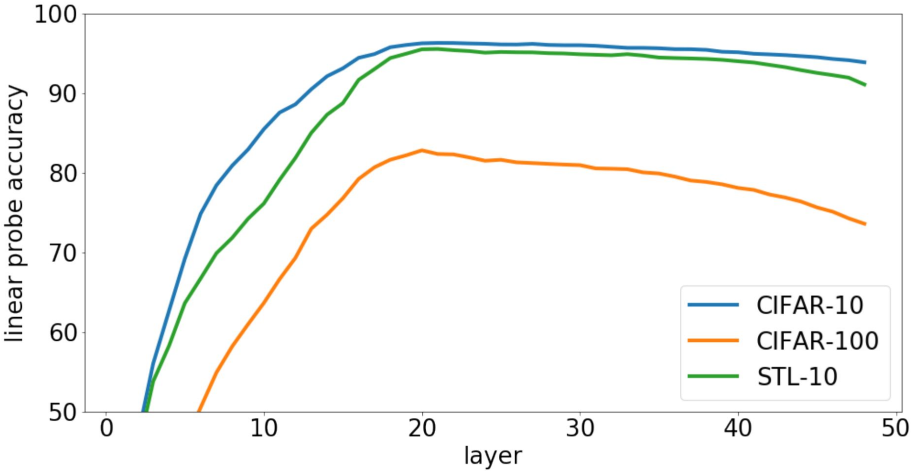
- ImageNet
  - 1.2M images
  - Low resolution
  - 9-bit color palette
  - No labels
- Web images
  - 100M images
  - Low resolution
  - 9-bit color palette
  - No labels

Supervised linear probing

- CIFAR-10
- CIFAR-100
- STL-10
- ImageNet

Pre-training and fine-tuning use the same resolution. For most experiments 32x32, otherwise 48x48, or 64x64.

Representation quality by layer



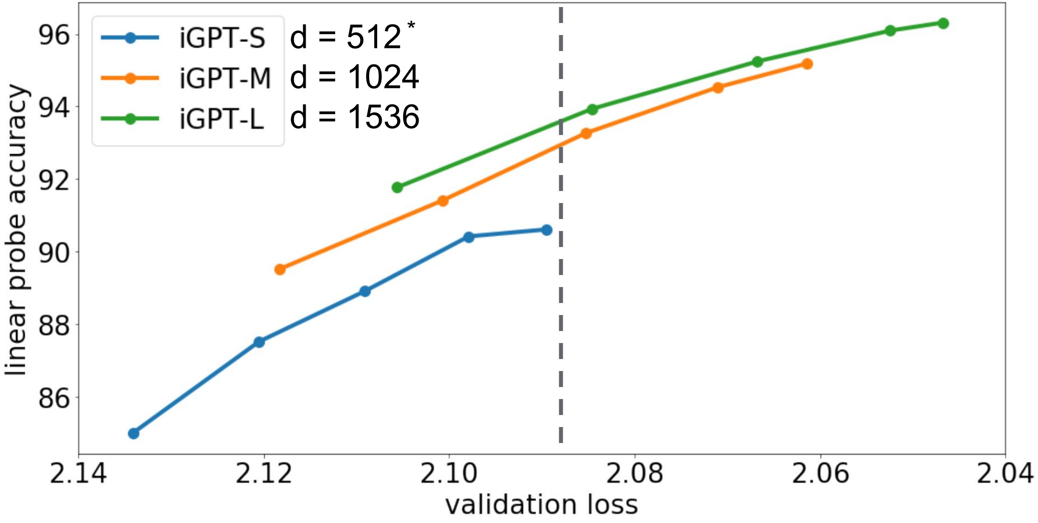
Low-level input representation

Semantic representation

Low-level output representation

- Best representations in the middle
- Authors' hypothesis:
  iGPT behaves similarly to an autoencoder, but without the bottleneck

Representation quality by model size



- Horizontal axis: larger models are better generators
- Correlation: generation performance and probe accuracy
- Generation performance being equal, larger models learn more discriminative features

* Note the different sizes of *dim model*. In iGPT-L, the linear probe has 3x more values to work with.

State-of-the-art accuracies

Model	Acc	Unsup Transfer	Sup Transfer
CIFAR-10			
ResNet-152	94		√
SimCLR	95.3	√	
iGPT-L	96.3	√	
CIFAR-100			
ResNet-152	78.0		√
SimCLR	80.2	√	
iGPT-L	82.8	√	
STL-10			
AMDIM-L	94.2	√	
iGPT-L	95.5	√	

iGPT linear probe accuracy

Model	Acc	Unsup Transfer	Sup Transfer
CIFAR-10			
AutoAugment	98.5		
SimCLR	98.6	√	
GPipe	99.0		√
iGPT-L	99.0	√	
CIFAR-100			
iGPT-L	88.5	√	
SimCLR	89.0	√	
AutoAugment	89.3		
EfficientNet	91.7		√

iGPT fine-tuning accuracy

iGPT is pre-trained on ImageNet (unlabeled and downsampled). Sup transfer means pre-trained on ImageNet with labels. iGPT-L is bigger and more expensive than the other models.

Conclusions

Confirm that

- Self-supervised pixel-wise pre-training can learn good representations
- 2D inductive biases can be abandoned (no patches, no conv layer at the input)

However:

- Abandoning 2D priors causes scaling issues at higher resolutions
- Humans do not “see” the world one pixel at the time, row-by-row.
  Is this the best kind of self-supervision to learn meaningful representations?

iGPT

Discussion

General discussion

General discussion

Adapting Transformers for vision, approaches:

- Features from a CNN become tokens (DETR)
- Pixel patches become tokens (ViT)
- Quantized pixels become tokens (iGPT)

Fewer biases,
More compute

General discussion

Comparison with CNNs:

- Can attention layers generalize convolutional layers?
    - Translational equivariance
    - Sparsity of connections
    - Locality
    - Positional embedding
- Are Transformers more computationally efficient than ConvNets? Why? Is it related to how GPUs kernels work?

General discussion

Research trends and future directions:

- Making architectures more general
- Removing inductive biases
- Training on huge datasets
- Is this the way to go?

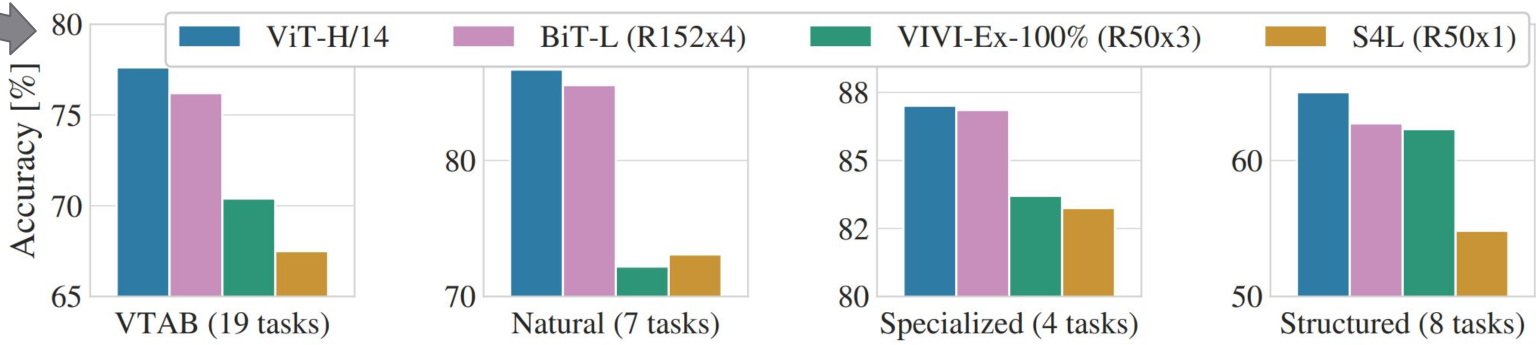
Related read: The Bitter Lesson, Sutton, 2019

Thanks for the *attention*!

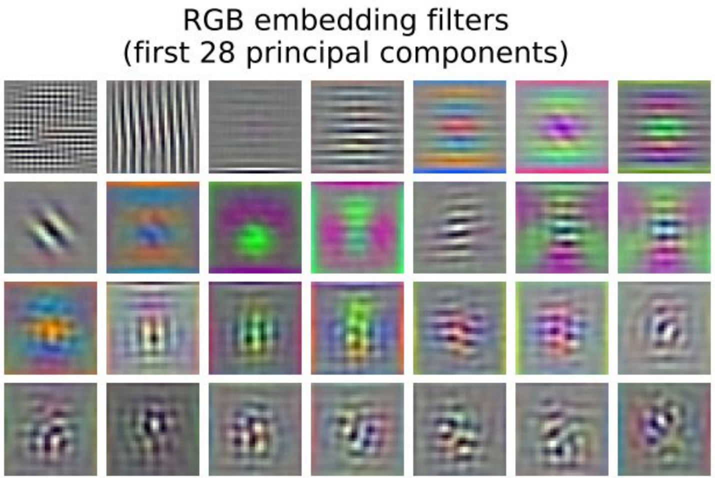
Extra slides

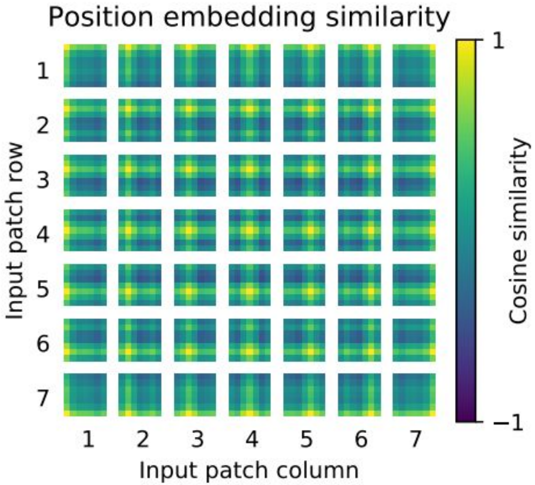
ViT state-of-the-art comparison

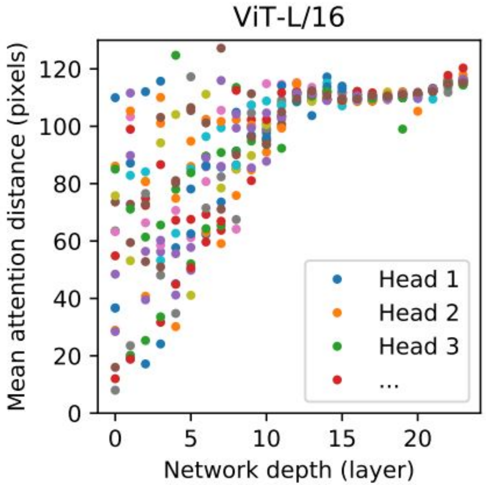
	Ours-JFT (ViT-H/14)	Ours-JFT (ViT-L/16)	Ours-I21K (ViT-L/16)	BiT-L (ResNet152x4)	Noisy Student (EfficientNet-L2)
ImageNet	**88.55** ± 0.04	87.76 ± 0.03	85.30 ± 0.02	87.54 ± 0.02	88.4/88.5*
ImageNet ReaL	**90.72** ± 0.05	90.54 ± 0.03	88.62 ± 0.05	90.54	90.55
CIFAR-10	**99.50** ± 0.06	99.42 ± 0.03	99.15 ± 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



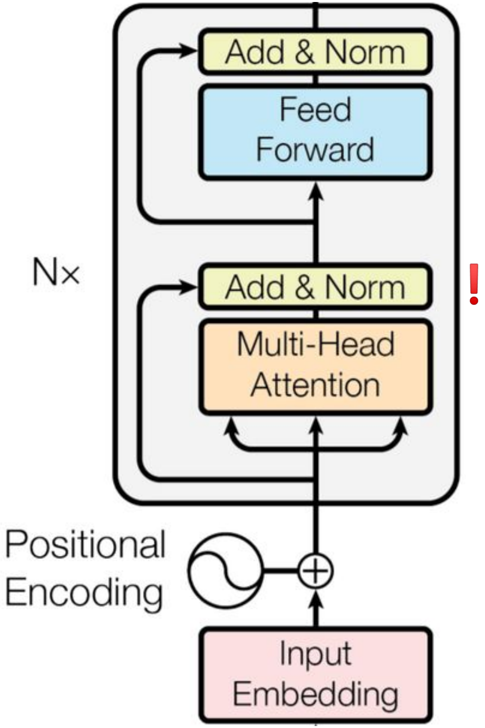
ViT: model inspection



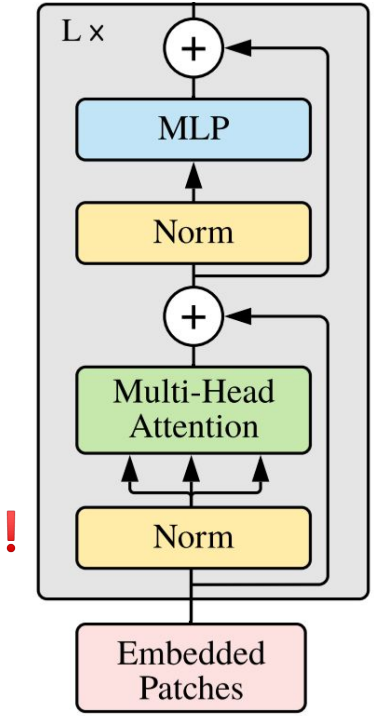




ViT: small architectural difference

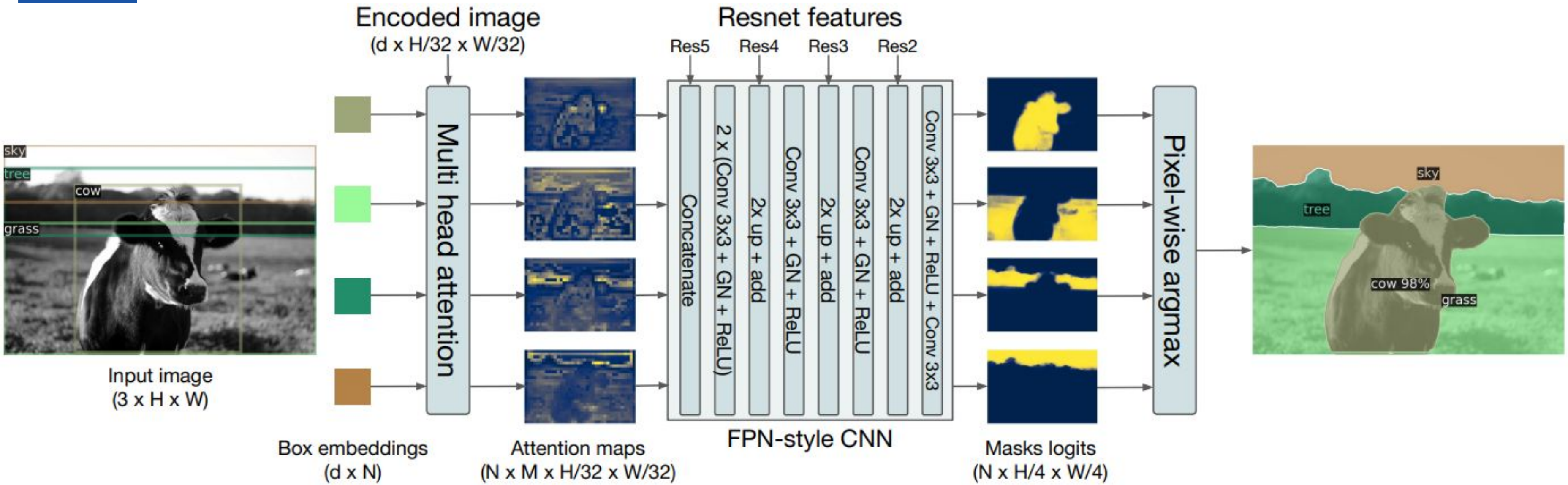


"Attention Is All You Need", Vaswani et al.



"Vision Transformer", Dosovitskiy et al.

DETR: panoptic segmentation



- For each detected object, the attention map of the corresponding object query can be used as the input to a segmentation model
- Can be trained jointly with the object detector or later
- Performs well on COCO (53 stuff classes, 80 object classes)

iGPT: autoregressive (GPT) vs masking (BERT)