Transformers in Computer Vision

Sergey Zagoruyko

October 7, 2024

Skoltech
Skolkovo Institute of Science and Technology

Course contents:

1. Transformers in Computer Vision
2. Transformers in Object Detection
3. Transformers in Autonomous Driving

What is convolution?

Convolution Example

Input:

1	4	-1	0	2	-2	1	3	3	1

Filter:

1	2	0	-1

Output:

Source: François Fleuret's deep learning course

Convolution Example

Input:

1	4	-1	0	2	-2	1	3	3	1

Filter:

1	2	0	-1

Output:

9						

Source: François Fleuret's deep learning course

Convolution Example

Input:

1	4	-1	0	2	-2	1	3	3	1

Filter:

1	2	0	-1

Output:

9	0					

Source: François Fleuret's deep learning course

Convolution Example

Input:

1	4	-1	0	2	-2	1	3	3	1

Filter:

1	2	0	-1

Output:

9	0	1				

Source: François Fleuret's deep learning course

Convolution Example

Input:

1	4	−1	0	2	−2	1	3	3	1

Filter:

1	2	0	−1

Output:

9	0	1	3			

Source: François Fleuret's deep learning course

Convolution Example

Input:

1	4	-1	0	2	-2	1	3	3	1

Filter:

1	2	0	-1

Output:

9	0	1	3	-5		

Source: François Fleuret's deep learning course

Convolution Example

Input:

1	4	-1	0	2	-2	1	3	3	1

Filter:

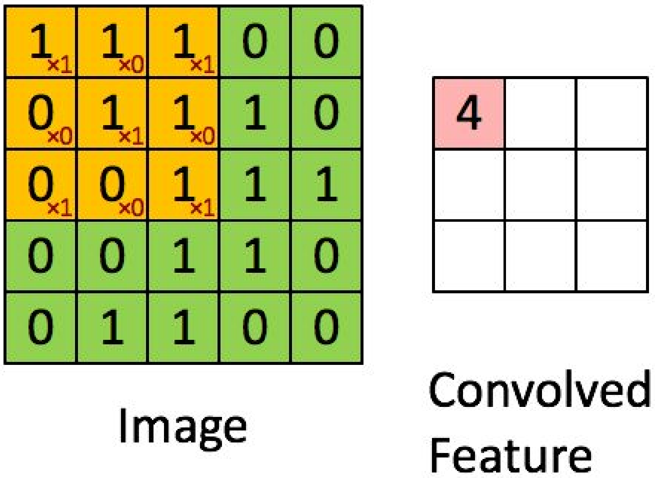
1	2	0	-1

Output:

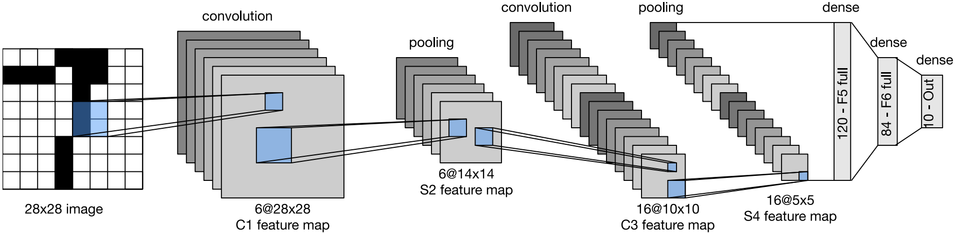
9	0	1	3	-5	-3	

Source: François Fleuret's deep learning course

Convolutions in 2D



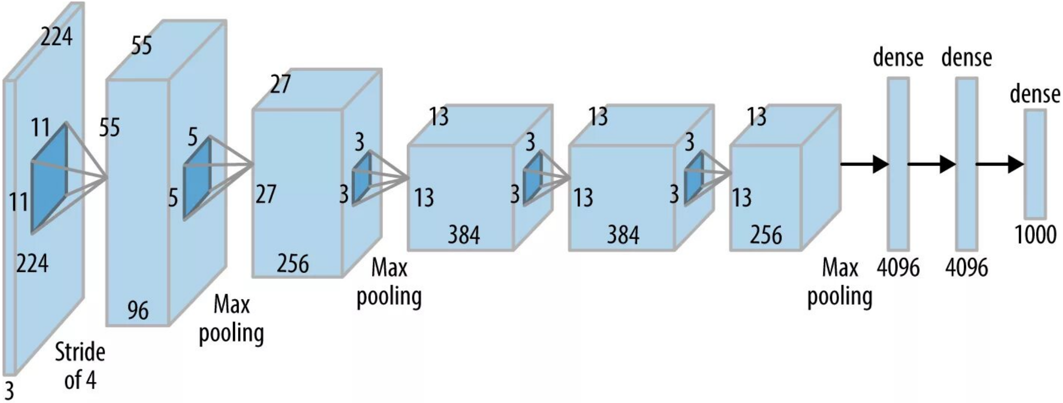
LeNet



LeNet-5 in 1998

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

AlexNet



AlexNet in 2012

https://machinelearningmastery.ru/the-w3h-of-alexnet-vggnet-resnet-and-inception-7baaaecccc96/

Inductive Bias in Neural Networks

- **Definition:**
  - Inductive bias refers to the set of assumptions a learning algorithm makes to generalize from the training data to unseen data.

Inductive Bias in Neural Networks

- **Definition:**
  - Inductive bias refers to the set of assumptions a learning algorithm makes to generalize from the training data to unseen data.
- **Importance:**
  - Helps in guiding the learning process and making predictions.
  - Determines the types of patterns a model can learn.

Inductive Bias in Neural Networks

- **Definition:**
  - Inductive bias refers to the set of assumptions a learning algorithm makes to generalize from the training data to unseen data.
- **Importance:**
  - Helps in guiding the learning process and making predictions.
  - Determines the types of patterns a model can learn.
- **Examples in Neural Networks:**
  - *Convolutional Neural Networks (CNNs):* Assumption of spatial hierarchies in images.
  - *Recurrent Neural Networks (RNNs):* Assumption of sequential dependencies in time-series data.

Inductive Bias in Neural Networks

- **Definition:**
  - Inductive bias refers to the set of assumptions a learning algorithm makes to generalize from the training data to unseen data.
- **Importance:**
  - Helps in guiding the learning process and making predictions.
  - Determines the types of patterns a model can learn.
- **Examples in Neural Networks:**
  - *Convolutional Neural Networks (CNNs):* Assumption of spatial hierarchies in images.
  - *Recurrent Neural Networks (RNNs):* Assumption of sequential dependencies in time-series data.
- **Trade-offs:**
  - Stronger biases can lead to faster learning but may reduce flexibility.
  - Weaker biases increase flexibility but may require more data.

Two pillars of deep learning:

Two pillars of deep learning:

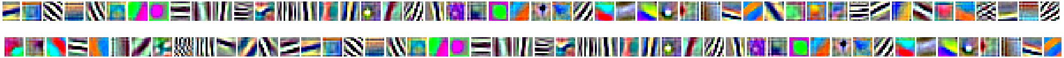
- Large amounts of rich diverse data

Two pillars of deep learning:

- Large amounts of rich diverse data
- Large amounts of compute

AlexNet

Visualizing parameters



Francois Fleuret's Deep Learning Course

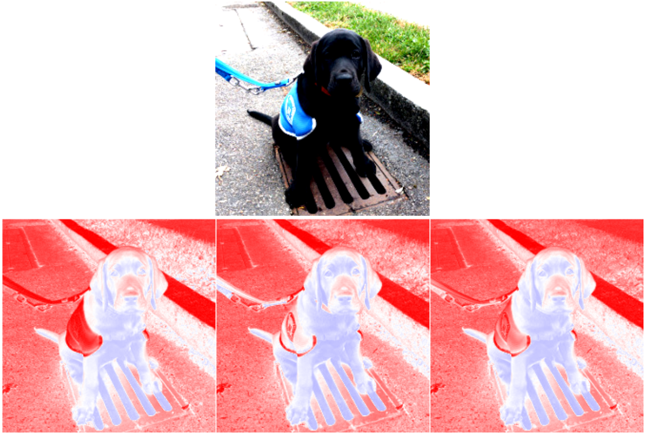
AlexNet

Visualizing activations

Francois Fleuret's Deep Learning Course

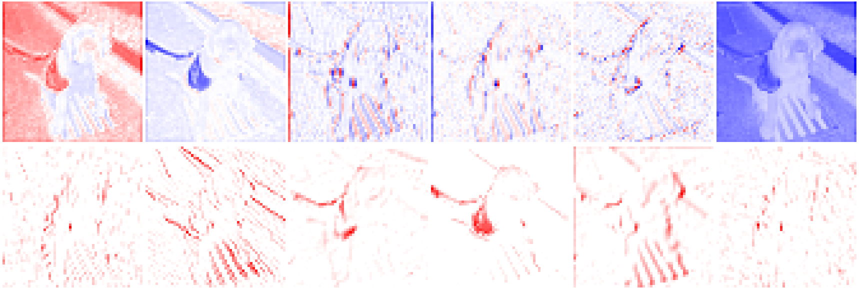
AlexNet

Visualizing activations



AlexNet

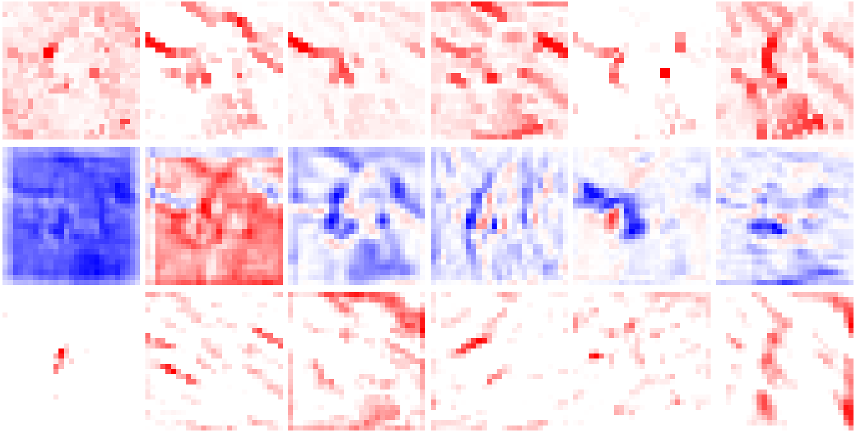
Visualizing activations



Francois Fleuret's Deep Learning Course

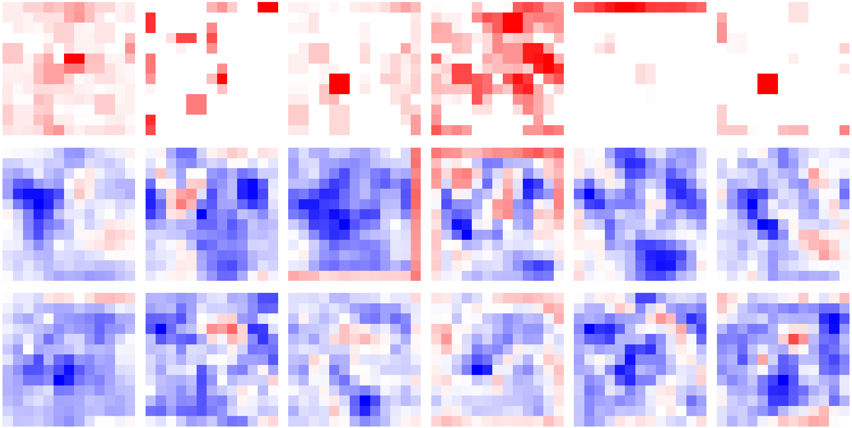
AlexNet

Visualizing activations



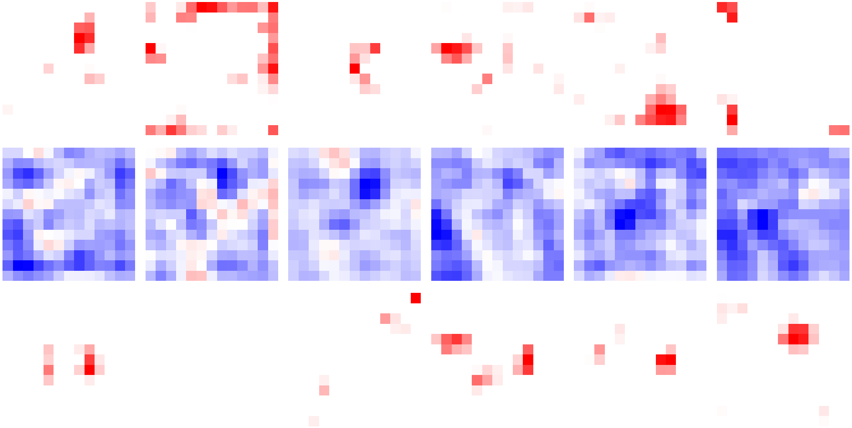
AlexNet

Visualizing activations

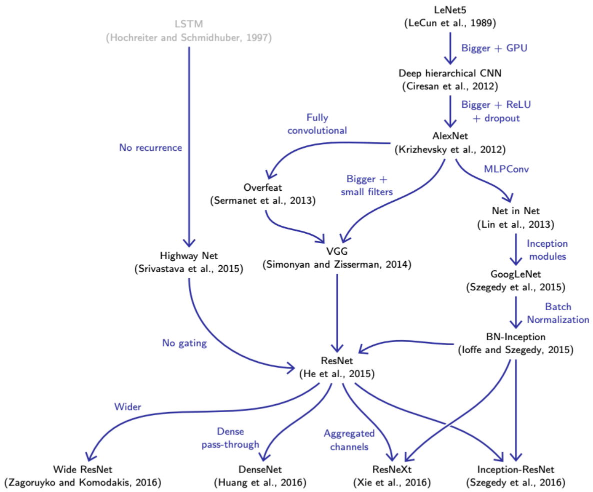


AlexNet

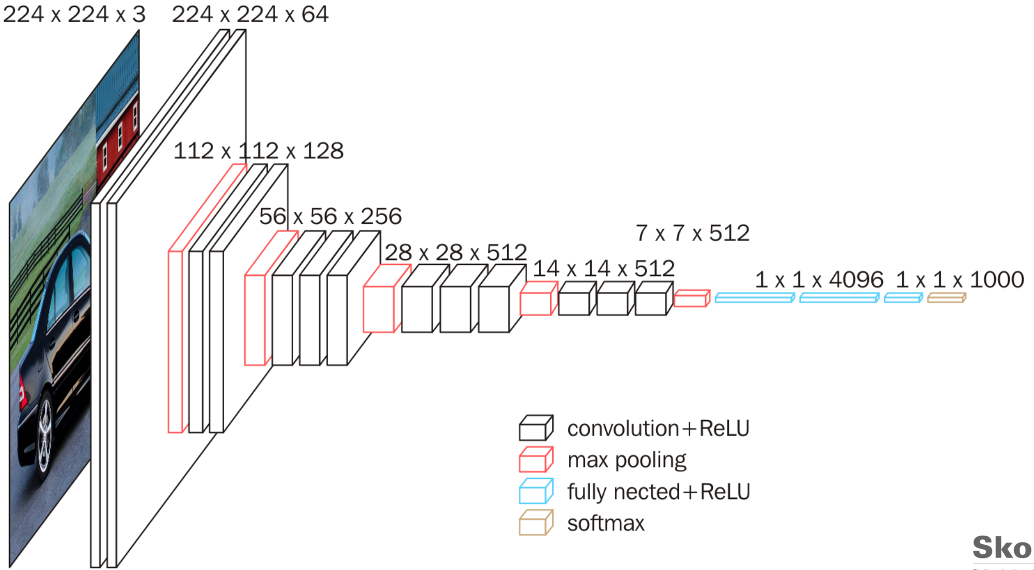
Visualizing activations



AlexNet

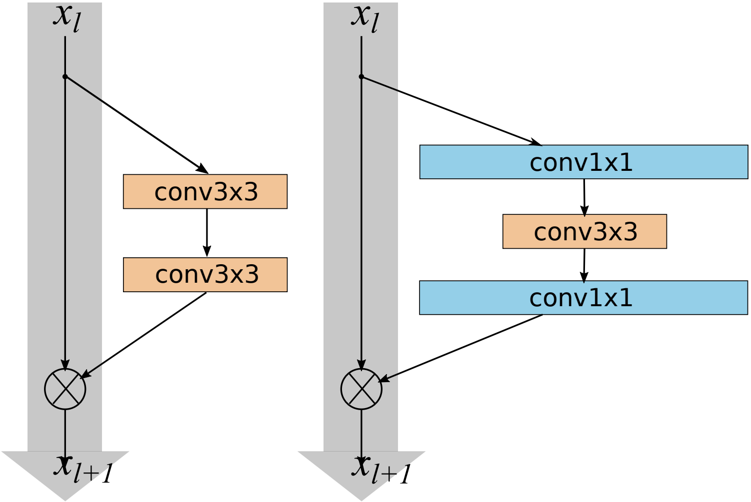

LSTM
(Hochreiter and Schmidhuber, 1997)
LeNet5
(LeCun et al., 1989)
Bigger + GPU
Deep hierarchical CNN
(Ciresan et al., 2012)
Bigger + ReLU
+ dropout
Fully
convolutional
No recurrence
AlexNet
(Krizhevsky et al., 2012)
MLPConv
Bigger +
small filters
Overfeat
(Sermanet et al., 2013)
Net in Net
(Lin et al., 2013)
Inception
modules
Highway Net
(Srivastava et al., 2015)
VGG
(Simonyan and Zisserman, 2014)
GoogLeNet
(Szegedy et al., 2015)
Batch
Normalization
BN-Inception
(Ioffe and Szegedy, 2015)
No gating
ResNet
(He et al., 2015)
Wider
Dense
pass-through
Aggregated
channels
Wide ResNet
(Zagoruyko and Komodakis, 2016)
DenseNet
(Huang et al., 2016)
ResNeXt
(Xie et al., 2016)
Inception-ResNet
(Szegedy et al., 2016)

VGG



Skoltech
Skolkovo Institute of Science and Technology

ResNet



(Zagoruyko and Komodakis 2016)

What is a Transformer?

Self-attention

Core operation in the Transformer:

$$
\begin{aligned}
Q &= W_q X \\
K &= W_k X \\
V &= W_v X
\end{aligned}
$$

$$Z = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right) V$$

Self-attention

Core operation in the Transformer:

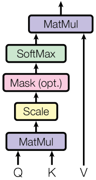
$$
\begin{aligned}
Q &= W_q X \\
K &= W_k X \\
V &= W_v X
\end{aligned}
$$

$$Z = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d}}\right) V$$

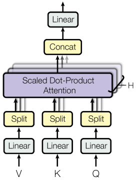
- Quadratic cost
- Input order equivariant

Multi-head attention

Scaled Dot-Product Attention



Multi-head Attention



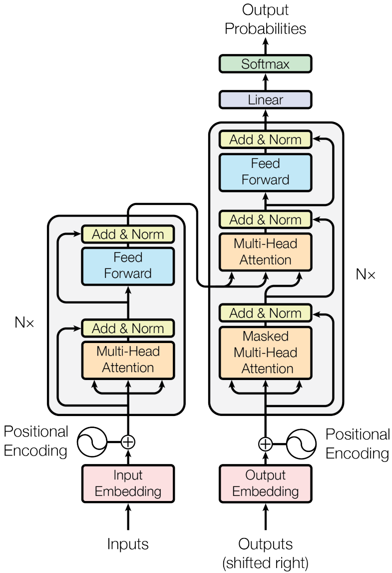
$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

$$MultiHead(Q, K, V) = Concat(H_1, ..., H_h)W^O$$

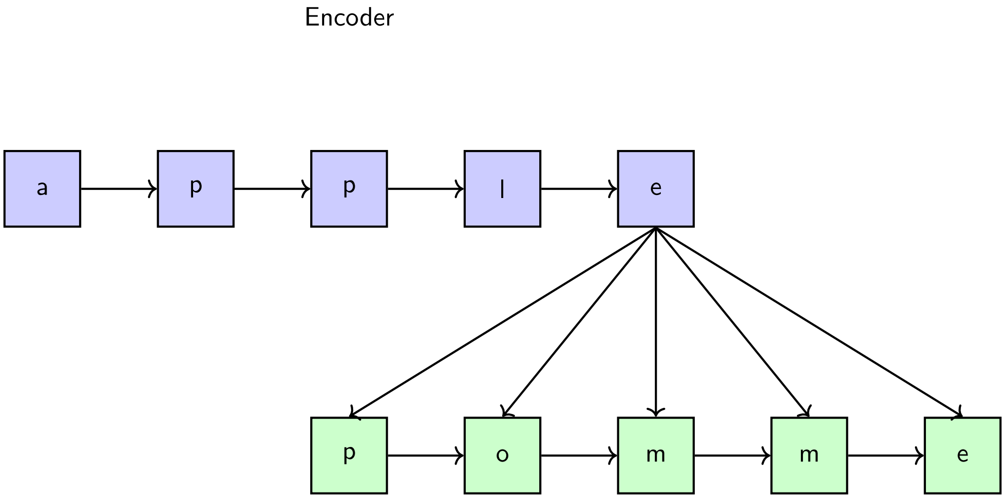
$$H_i = Attention(QW_i^Q, KW_i^K, VW_i^V), i = 1, ..., h$$

$$W_i^Q \in \mathbb{R}^{d_{model} \times d_k}, W_i^K \in \mathbb{R}^{d_{model} \times d_k}, W_i^V \in \mathbb{R}^{d_{model} \times d_v}, W_i^O \in \mathbb{R}^{hd_v \times d_{model}}$$

Original Transformer (Vaswani et al. 2017)



Seq2Seq Translation: Apple to Pomme



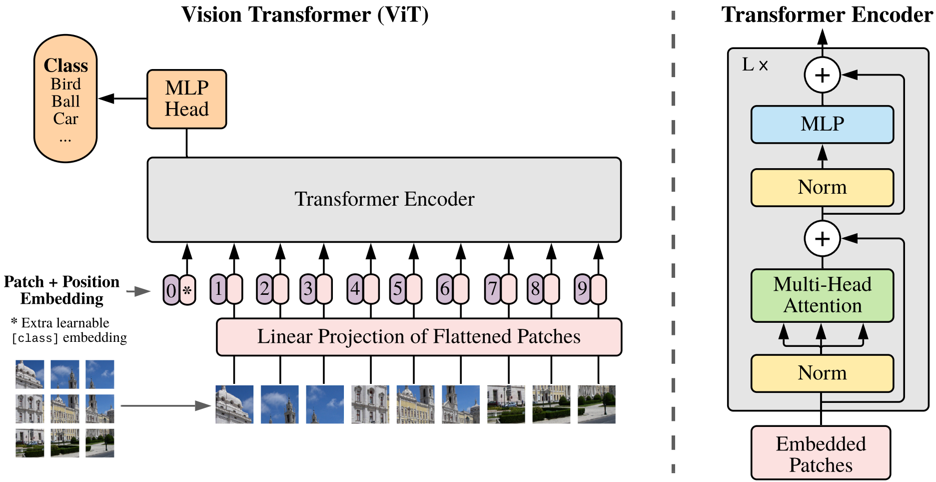
ViT: Vision Transformers

Vision Transformer, **ViT** (Dosovitskiy et al. 2020) -

.

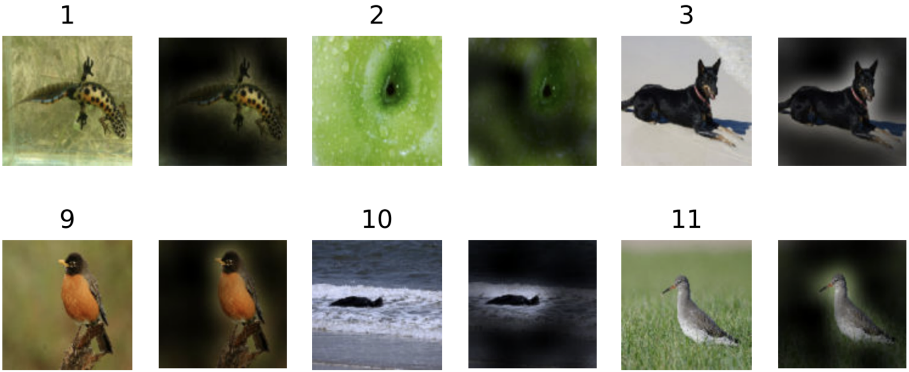
> *Inspired by the Transformer scaling successes in NLP, we experiment with applying a standard Transformer directly to images, with the fewest possible modifications. To do so, we split an image into patches and provide the sequence of linear embeddings of these patches as an input to a Transformer. Image patches are treated the same way as tokens (words) in an NLP application. We train the model on image classification in supervised fashion. (**Dosovitskiy et al. 2020**)*

ViT: Vision Transformers

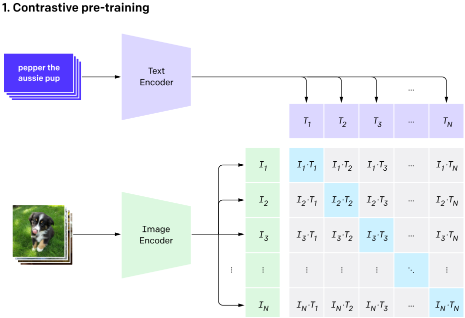


(Dosovitskiy et al. 2020)

ViT attention



What is CLIP?

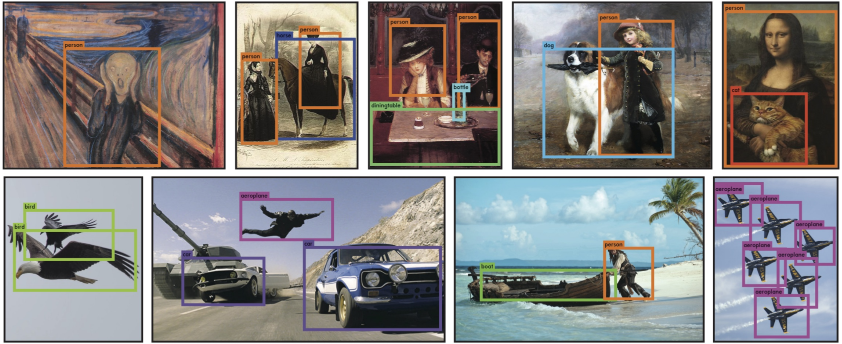


- **CLIP** stands for **Contrastive Language–Image Pretraining**.
- Developed by OpenAI, it combines text and image understanding.
- Utilizes a large dataset of text-image pairs for training.

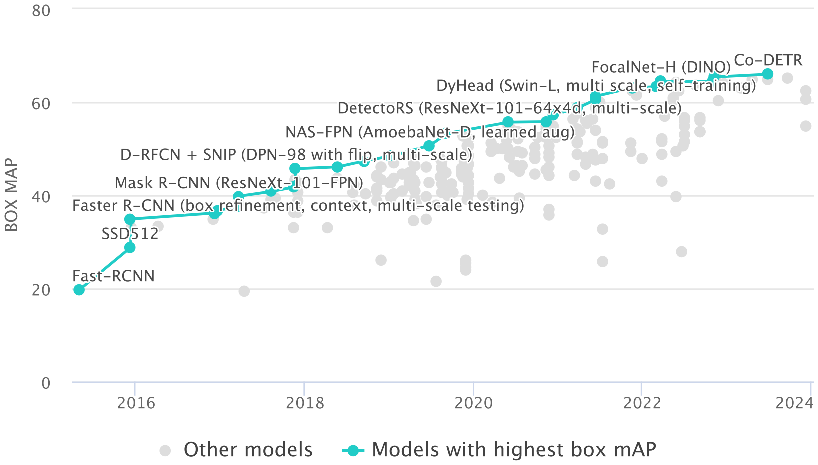
https://openai.com/index/clip

Object Detection

Object Detection

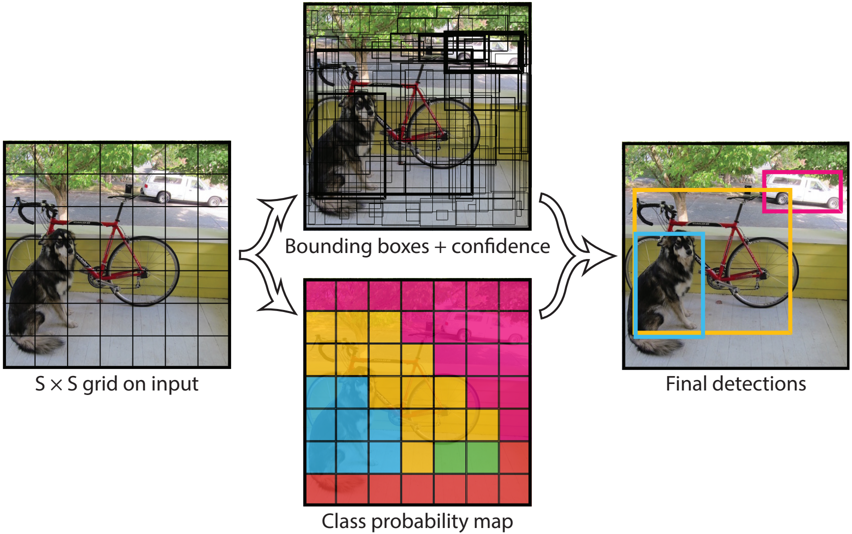


Object Detection



https://paperswithcode.com/sota/object-detection-on-coco

YOLO


S × S grid on input
Bounding boxes + confidence
Class probability map
Final detections

Classical approach to detection

- Popular approach: detection $:=$ classification of boxes

Classical approach to detection

- Popular approach: detection := classification of boxes
- Requires selecting a subset of candidate boxes

Classical approach to detection

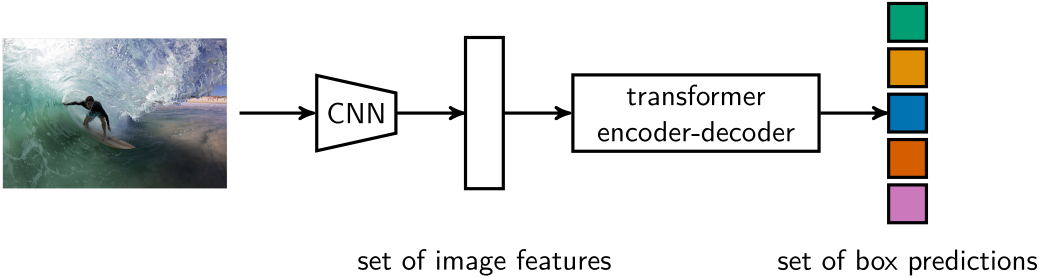
- Popular approach: detection := classification of boxes
- Requires selecting a subset of candidate boxes
- Regression step to refine the predictions

Classical approach to detection

- Popular approach: detection $:=$ classification of boxes
- Requires selecting a subset of candidate boxes
- Regression step to refine the predictions
- Typically non-differentiable

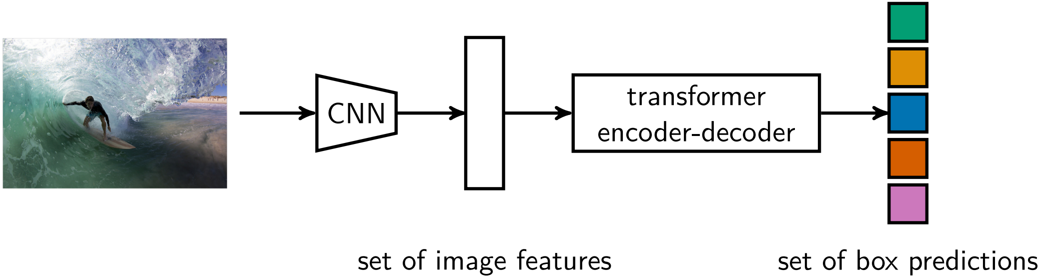
DETR: Rethinking object detection

- *Neural machine translation* from features to boxes



DETR: Rethinking object detection

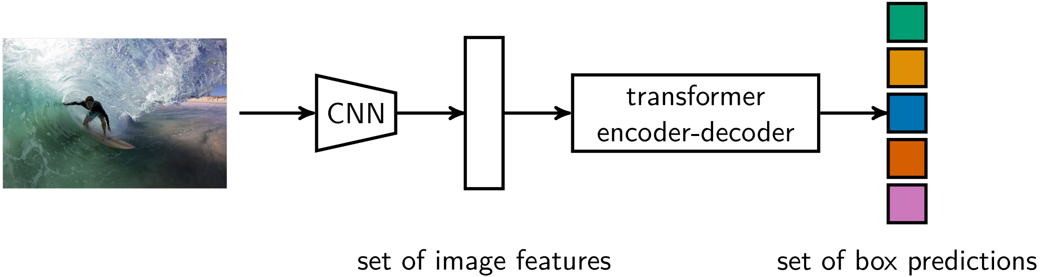
- *Neural machine translation* from features to boxes



- End-to-end parallel set prediction

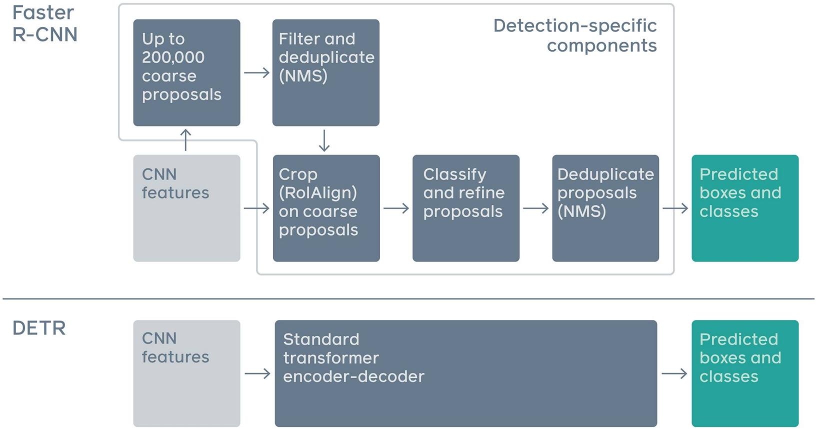
DETR: Rethinking object detection

- *Neural machine translation* from features to boxes

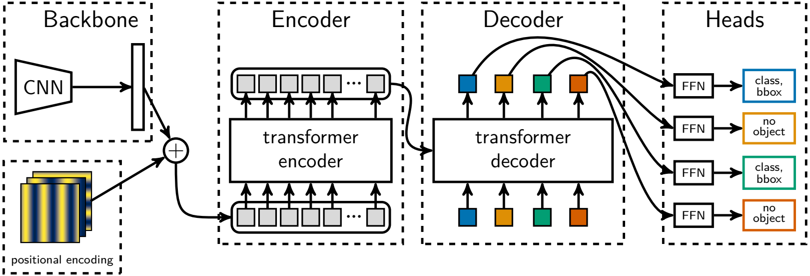


- End-to-end parallel set prediction
- Global scene reasoning

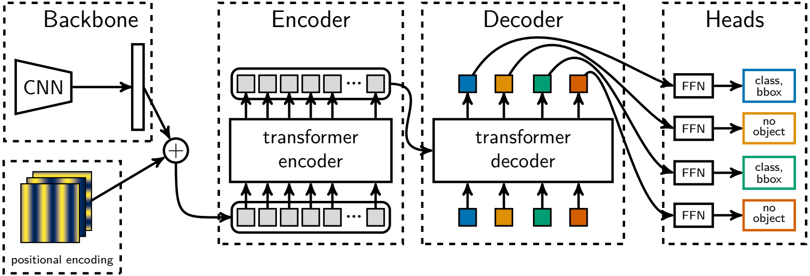
Streamlined detection pipeline



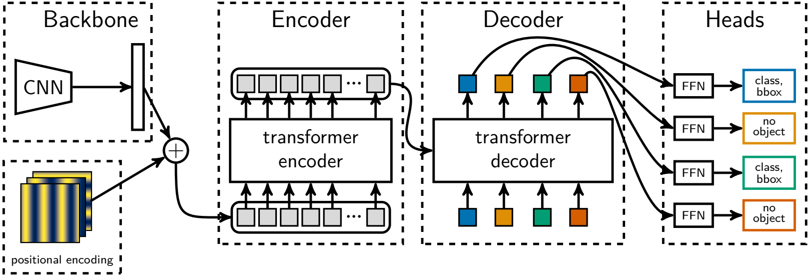
- We use standard ResNet from torchvision



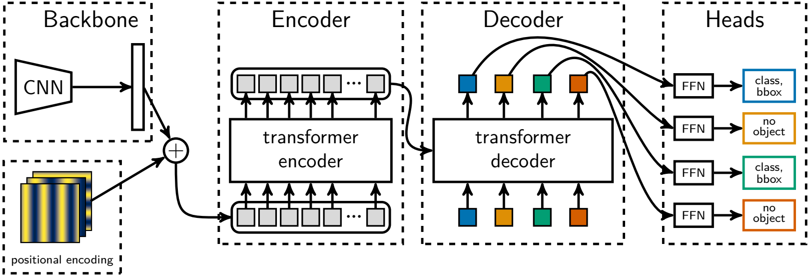
- We use standard ResNet from torchvision
- Pretraining on Imagenet is key (labels or SSL)



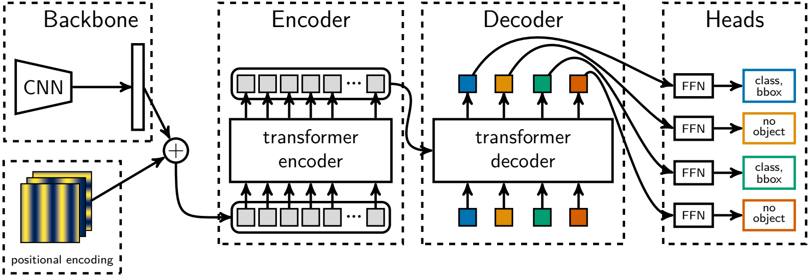
- We use 2D sine/cosine embeddings



- We use 2D sine/cosine embeddings
- Embeddings are added to features

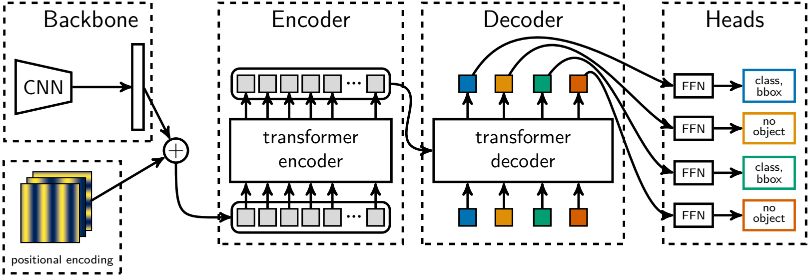


- We use 6 layers of transformer encoder



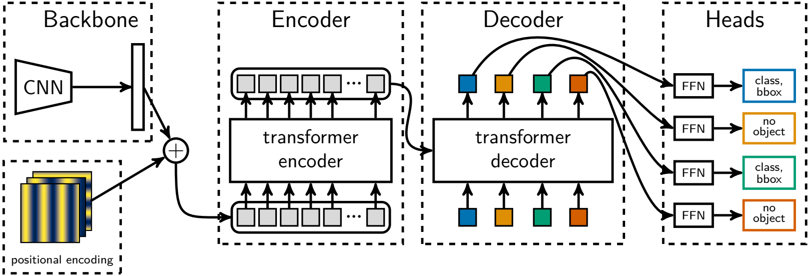
Skoltech
Skolkovo Institute of Science and Technology

- We use 6 layers of transformer encoder
- Global reasoning through attention



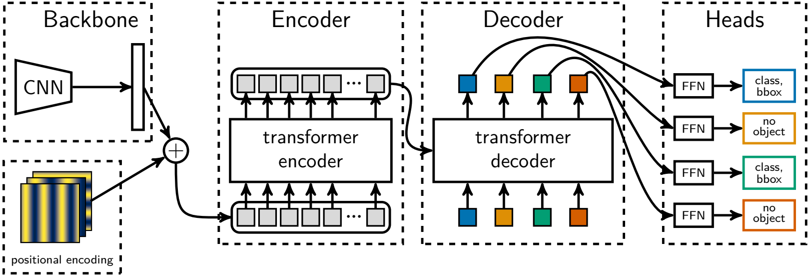
- We use 6 layers of transformer encoder
- Global reasoning through attention
- Starts separating instances

;

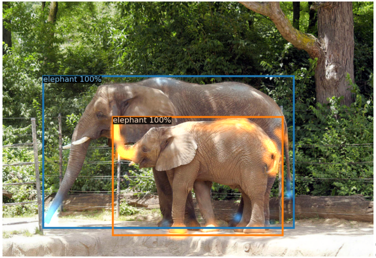


Skoltech
Skolkovo Institute of Science and Technology

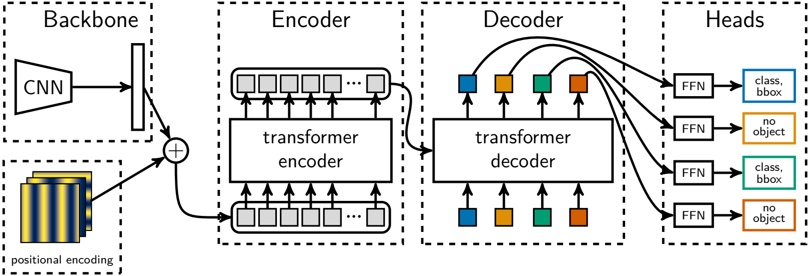
- We use 6 layers of transformer decoder;



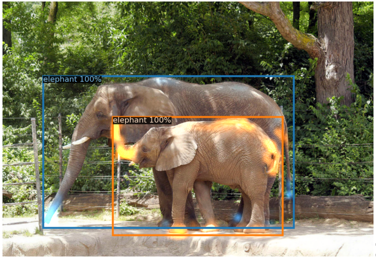
- We use 6 layers of transformer decoder;
- Attention focuses on extremities



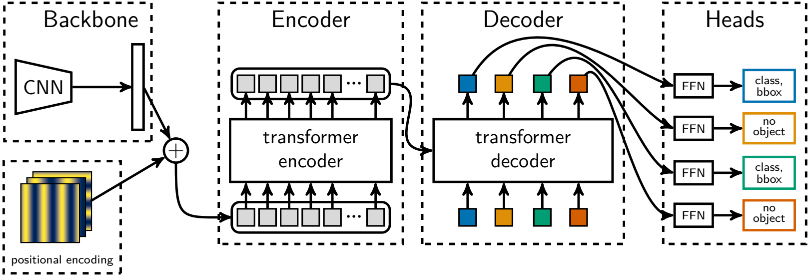
;



- We use 6 layers of transformer decoder;
- Attention focuses on extremities
- Predictions are refined at each layer **in parallel**



;



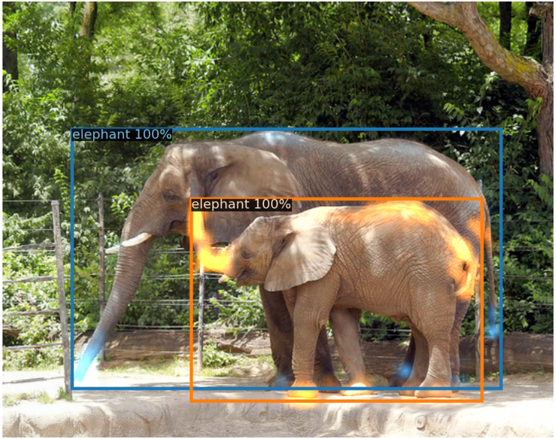
Decoder attention weights

6 decoder layers of:

- self-attention
- enc-dec attention
- FFN
- LayerNorm

All outputs are decoded *in parallel*

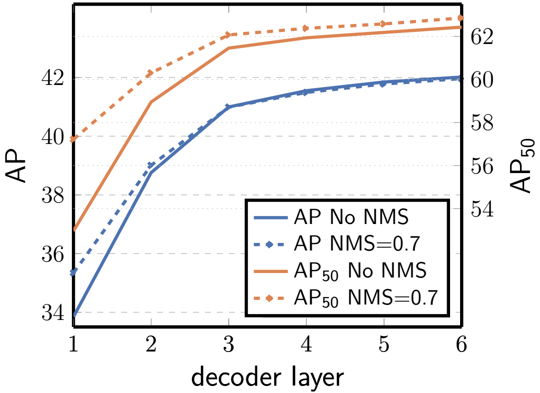
Attention focuses on extremities



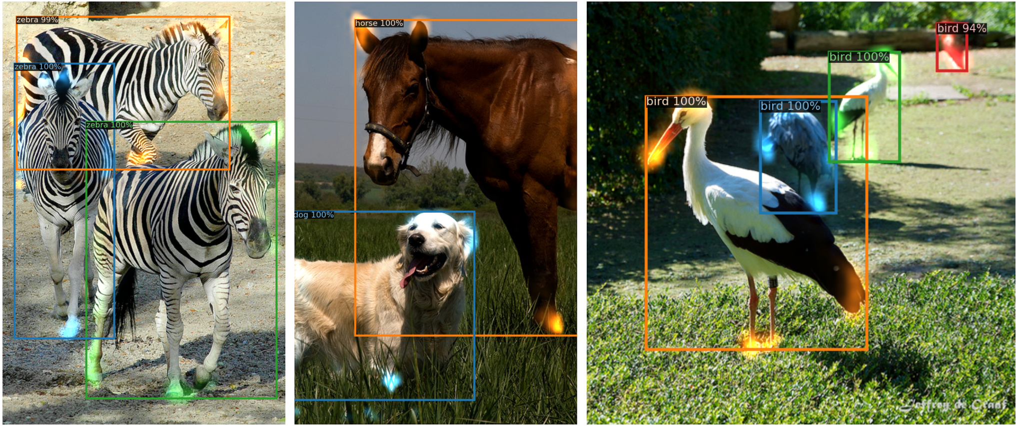
Decoder: no NMS needed

- AP/AP_{50} goes up in lower layers (no communication)
- AP goes down in the last layers
- AP_{50} goes up slightly

There is no need for NMS in DETR

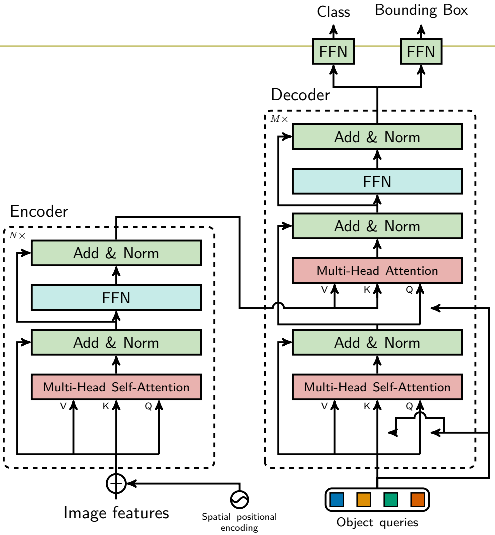


Decoder attention weights



Transformer architecture variations

- Base transformer
  39.2 AP[a]

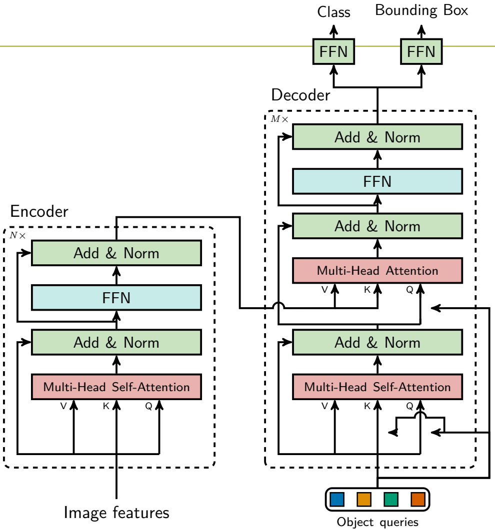


[a] Ashish Vaswani et al. (2017). "Attention is All you Need". In: *NeurIPS*.

Skoltech
Skolkovo Institute of Science and Technology

Transformer architecture variations

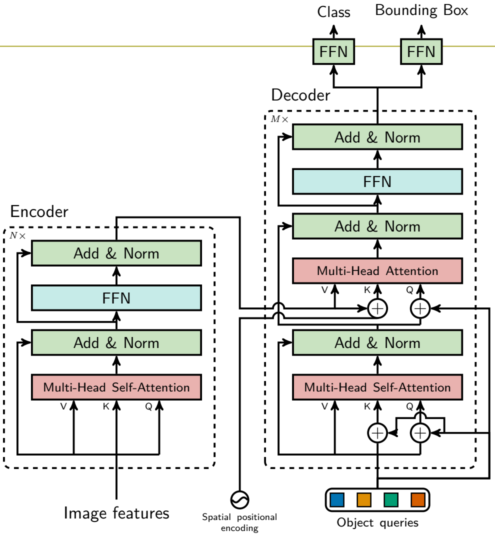
- Base transformer
  39.2 AP[a]
- No input positional encoding
  32.8 AP



[a] Ashish Vaswani et al. (2017). "Attention is All you Need". In: *NeurIPS*.

Transformer architecture variations

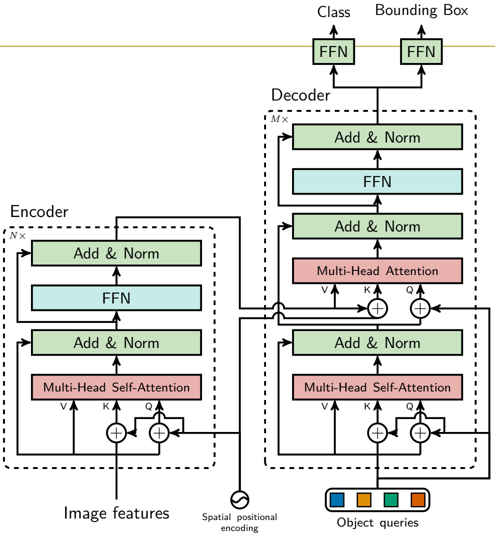
- Base transformer
  39.2 AP[a]
- No input positional encoding
  32.8 AP
- Encodings in decoder attentions only
  39.3 AP



[a] Ashish Vaswani et al. (2017). "Attention is All you Need". In: *NeurIPS*.

Transformer architecture variations

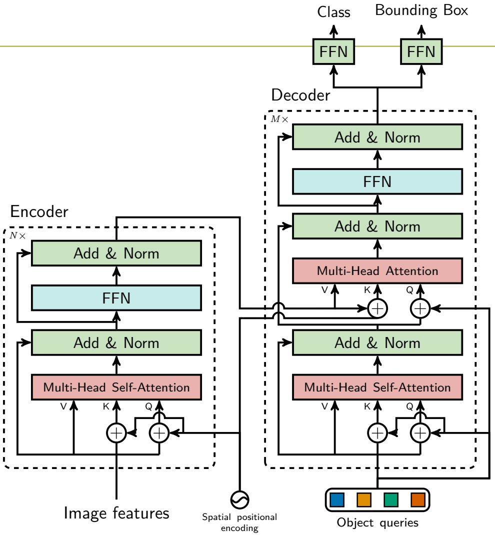
- Base transformer
  39.2 AP[a]
- No input positional encoding
  32.8 AP
- Encodings in decoder attentions only
  39.3 AP
- Encodings in all attentions
  40.6 AP



[a] Ashish Vaswani et al. (2017). "Attention is All you Need". In: *NeurIPS*.

Transformer architecture variations

- Base transformer
  39.2 AP[a]
- No input positional encoding
  32.8 AP
- Encodings in decoder attentions only
  39.3 AP
- Encodings in all attentions
  40.6 AP



All transformer parts are contributing!

[a] Ashish Vaswani et al. (2017). "Attention is All you Need". In: *NeurIPS*.