DERT

Paper Details

- Paper Title: End-to-End Object Detection with Transformers
- Publication Date: 28 May 2020
- Publisher: Nicolas Carion , Francisco Massa , …
- Affiliation: Fackbook AI

Objectives

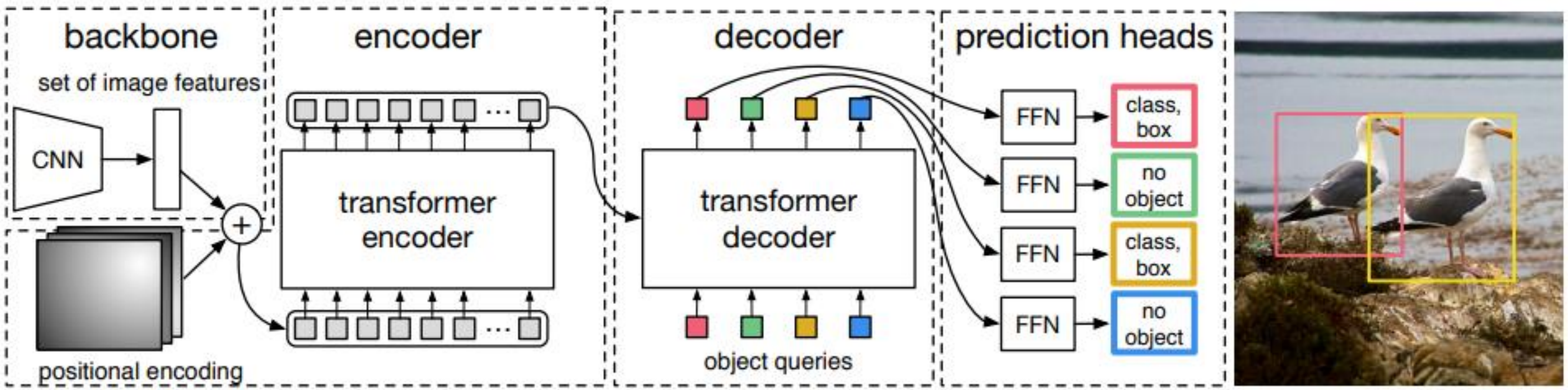
- end-to-end object detection

Motivation

- detectors do set prediction task in an indirect way
  - simplify the pipeline
- Transformer in other fields
  - Bridge the gap

Overall Architecture

- Backbone
- Transformer Encoder
- Transformer Decoder
- Prediction feed-forward network



Overall Architecture

- Backbone

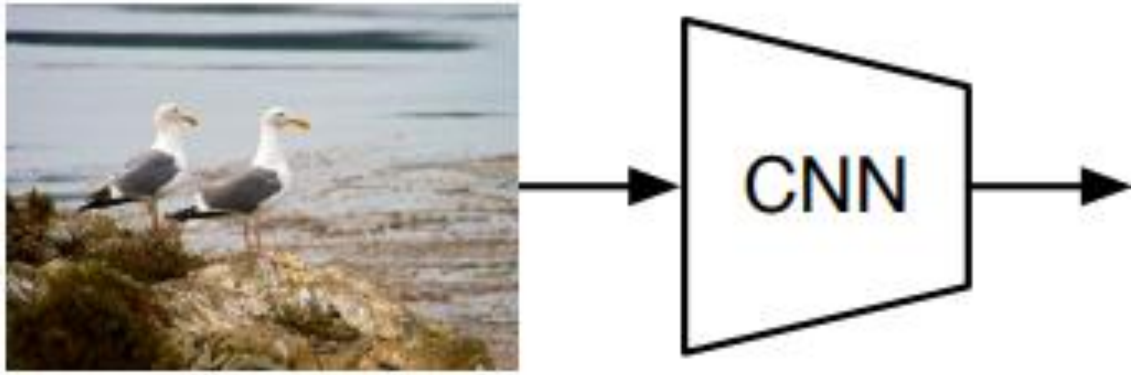
$x_{\text{img}} \in \mathbb{R}^{3 \times H_0 \times W_0}$

$\mathbb{R}^{C \times H \times W}$

$C = 2048 \text{ and } H, W = \frac{H_0}{32}, \frac{W_0}{32}.$

Overall Architecture

- Backbone



$x_{\text{img}} \in \mathbb{R}^{3\times H_0\times W_0}$ $\mathbb{R}^{C\times H\times W}$

ImageNet-pretrained ResNet model

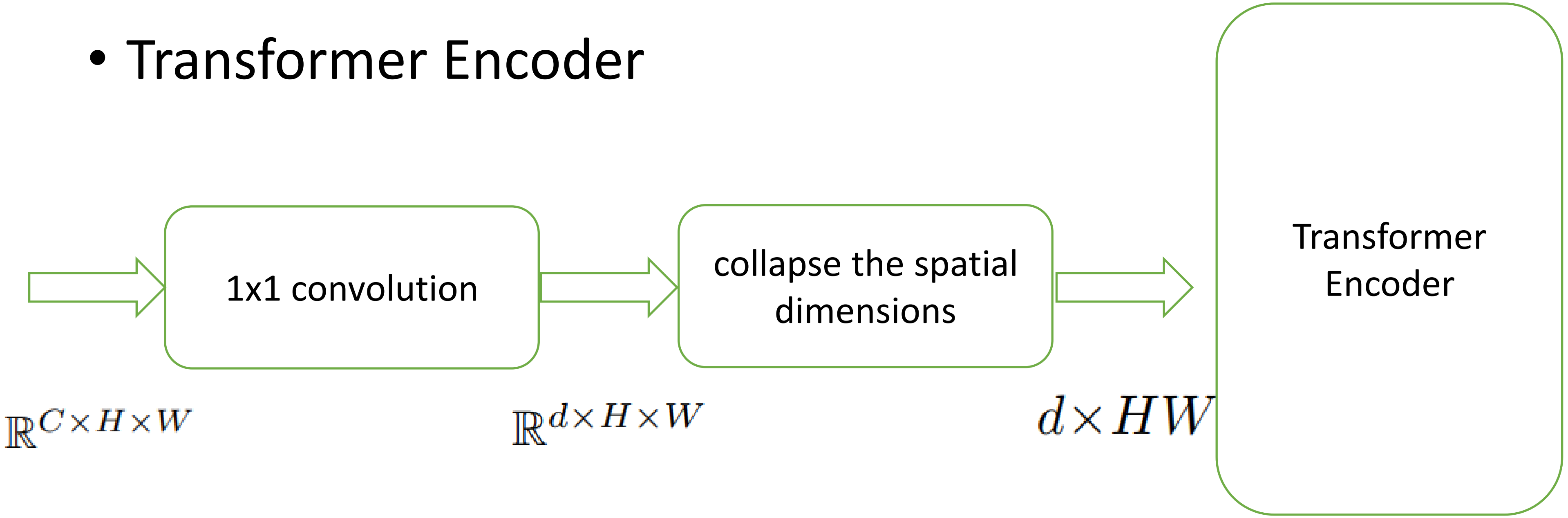
ResNet-50 → DETR-DC5

two different backbones dilated C5 stage

ResNet-101 → DETR-DC5-R101

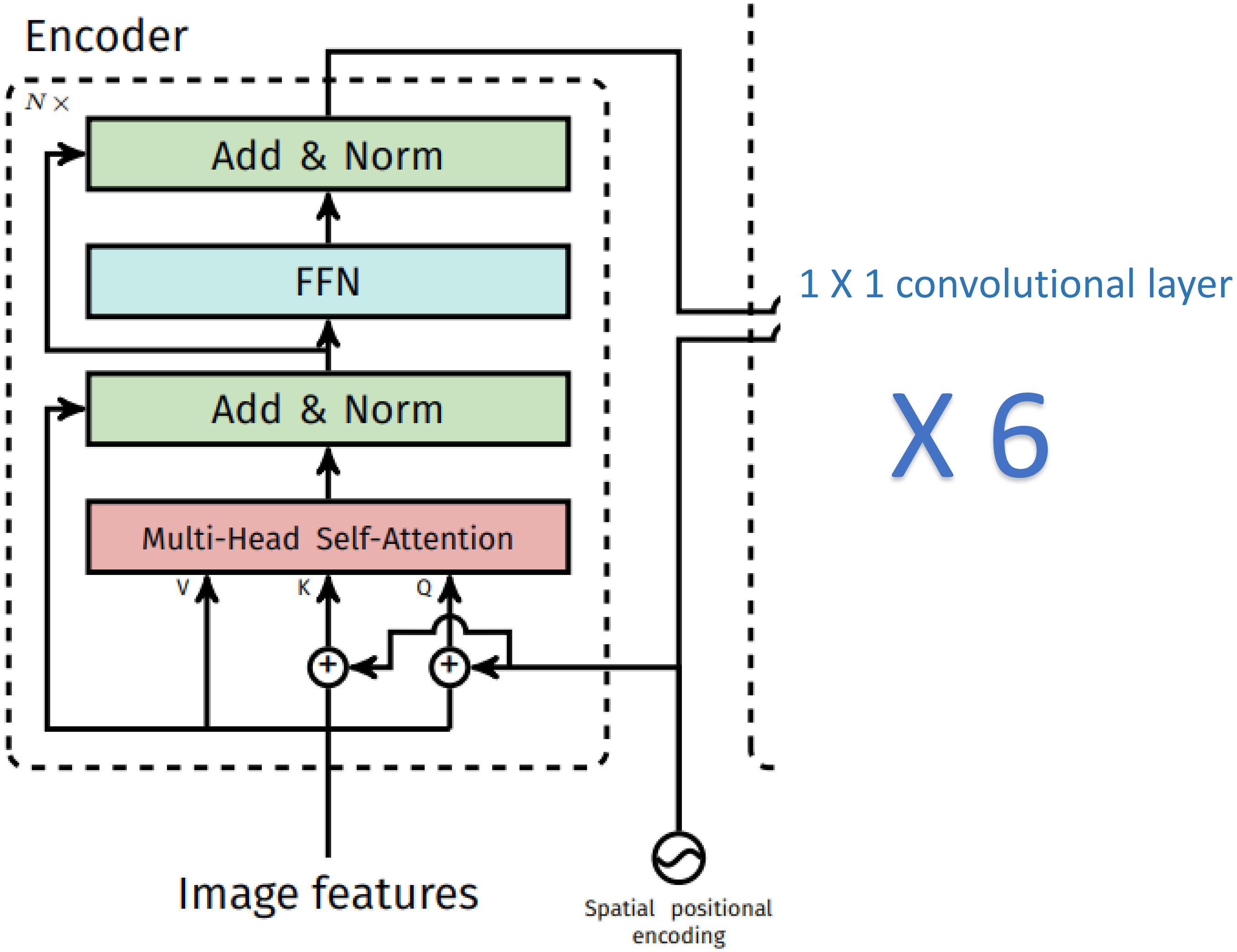
Overall Architecture

- Backbone
- Transformer Encoder



Overall Architecture

- Backbone
- Transformer Encoder



Overall Architecture

- Backbone
- Transformer Encoder
  - positional encodings

$$\text{Self-Attention}(\boldsymbol{X})_{t,:} := \text{softmax}\left(\boldsymbol{A}_{t,:}\right) \boldsymbol{X} \boldsymbol{W}_{val},$$

$$\boldsymbol{A} := \boldsymbol{X} \boldsymbol{W}_{qry} \boldsymbol{W}_{key}^{\top} \boldsymbol{X}^{\top}$$

$$\boldsymbol{A} := (\boldsymbol{X} + \boldsymbol{P}) \boldsymbol{W}_{qry} \boldsymbol{W}_{key}^{\top} (\boldsymbol{X} + \boldsymbol{P})^{\top}$$

Overall Architecture

- Backbone
- Transformer Encoder
  - positional encodings

$$\text{Self-Attention}(\boldsymbol{X})_{t,:} := \text{softmax}\left(\boldsymbol{A}_{t,:}\right)\boldsymbol{X}\boldsymbol{W}_{val},$$

$$\boldsymbol{A} := \boldsymbol{X}\boldsymbol{W}_{qry}\boldsymbol{W}_{key}^{\top}\boldsymbol{X}^{\top}$$

$$\boldsymbol{A} := (\boldsymbol{X}+\boldsymbol{P})\boldsymbol{W}_{qry}\boldsymbol{W}_{key}^{\top}(\boldsymbol{X}+\boldsymbol{P})^{\top}$$

Absolute Encoding

$$\mathbf{A}^{\mathrm{abs}}_{\boldsymbol{q},\boldsymbol{k}} = (\mathbf{X}_{\boldsymbol{q},:}+\mathbf{P}_{\boldsymbol{q},:})\boldsymbol{W}_{qry}\boldsymbol{W}_{key}^{\top}(\mathbf{X}_{\boldsymbol{k},:}+\mathbf{P}_{\boldsymbol{k},:})^{\top}$$

Relative Encoding

$$\mathbf{A}^{\mathrm{rel}}_{\boldsymbol{q},\boldsymbol{k}} := \mathbf{X}_{\boldsymbol{q},:}^{\top}\boldsymbol{W}_{qry}^{\top}\boldsymbol{W}_{key}\mathbf{X}_{\boldsymbol{k},:} + \mathbf{X}_{\boldsymbol{q},:}^{\top}\boldsymbol{W}_{qry}^{\top}\widehat{\boldsymbol{W}}_{key}\boldsymbol{r}_{\boldsymbol{\delta}} + \boldsymbol{u}^{\top}\boldsymbol{W}_{key}\mathbf{X}_{\boldsymbol{k},:} + \boldsymbol{v}^{\top}\widehat{\boldsymbol{W}}_{key}\boldsymbol{r}_{\boldsymbol{\delta}}$$

Overall Architecture

- Backbone
- Transformer Encoder
  - positional encodings

$$\text{Self-Attention}(\boldsymbol{X})_{t,:} := \text{softmax}\left(\boldsymbol{A}_{t,:}\right) \boldsymbol{X} \boldsymbol{W}_{val},$$

$$\boldsymbol{A} := \boldsymbol{X} \boldsymbol{W}_{qry} \boldsymbol{W}_{key}^{\top} \boldsymbol{X}^{\top}$$

$$\boldsymbol{A} := (\boldsymbol{X} + \boldsymbol{P}) \boldsymbol{W}_{qry} \boldsymbol{W}_{key}^{\top} (\boldsymbol{X} + \boldsymbol{P})^{\top}$$

Absolute Encoding

spatial pos. enc. encoder	spatial pos. enc. decoder	output pos. enc. decoder	AP	Δ	AP_{50}	Δ
none	none	learned at input	32.8	-7.8	55.2	-6.5
sine at input	sine at input	learned at input	39.2	-1.4	60.0	-1.6
learned at attn.	learned at attn.	learned at attn.	39.6	-1.0	60.7	-0.9
none	sine at attn.	learned at attn.	39.3	-1.3	60.3	-1.4
sine at attn.	sine at attn.	learned at attn.	**40.6**	-	**61.6**	-

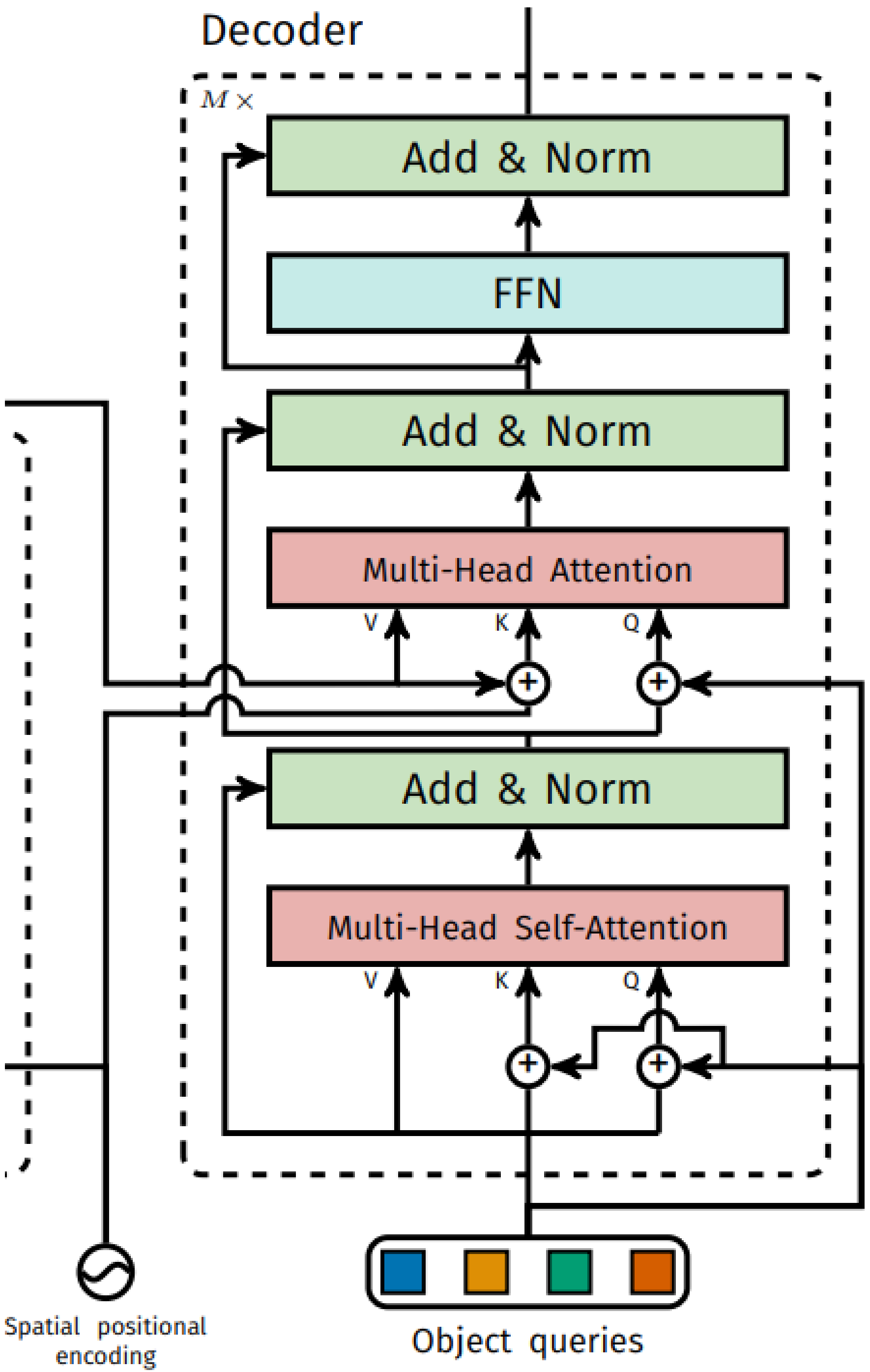
Attention Augmented Convolutional Networks

Relative Encoding

Position Encodings	mAP_{COCO}	mAP_{50}	mAP_{75}
None	37.7	56.0	40.2
CoordConv [29]	37.4	55.5	40.1
Relative (ours)	38.2	56.5	40.7

Overall Architecture

- Backbone
- Transformer Encoder
- Transformer Decoder

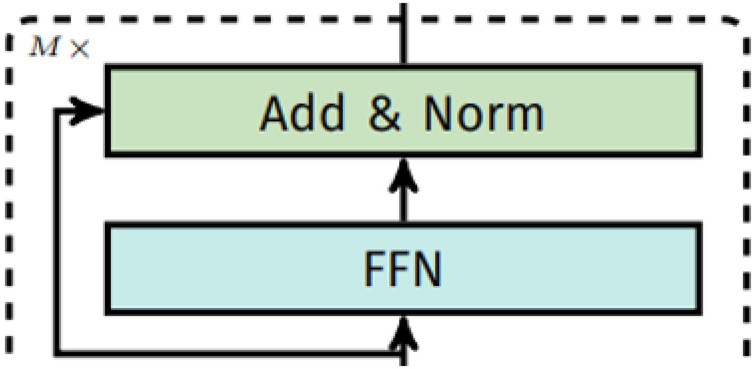


width 256.

X 6

Overall Architecture

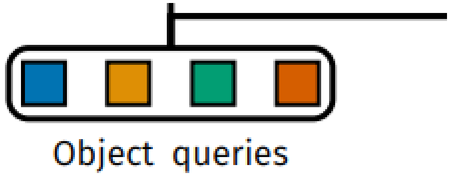
- Backbone
- Transformer Encoder
- Transformer Decoder



N input embeddings

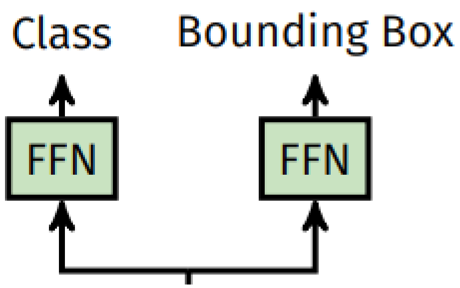
different

learnt positional encodings

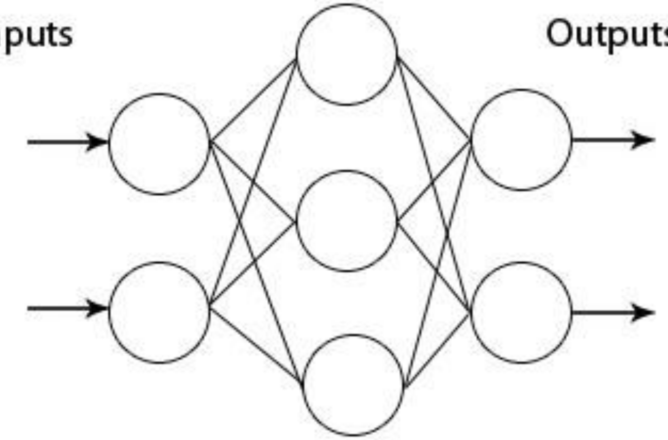


Overall Architecture

- Backbone
- Transformer Encoder
- Transformer Decoder
- Prediction feed-forward network



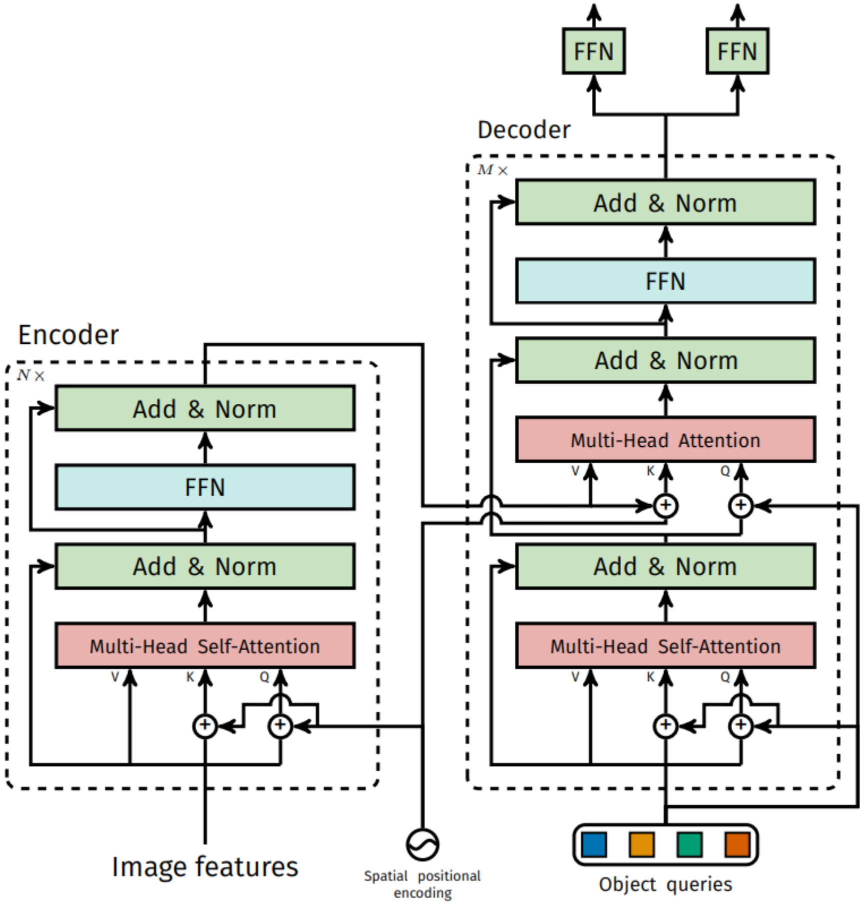
ReLU activation function

Prediction = 

\+ Linear projection layer

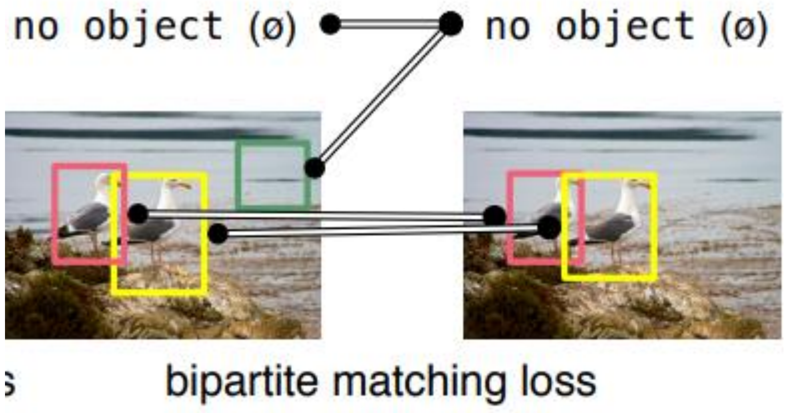
Hidden dimension d

Overall Architecture



Prediction Loss

- Bipartite matching



$$\hat{\sigma} = \underset{\sigma \in \mathfrak{S}_N}{\arg\min} \sum_{i}^{N} \mathcal{L}_{\mathrm{match}}(y_i, \hat{y}_{\sigma(i)})$$

Prediction Loss

- Bipartite matching
  - Hungarian Algorithm

$$\hat{\sigma} = \underset{\sigma \in \mathfrak{S}_N}{\arg\min} \sum_{i}^{N} \mathcal{L}_{\text{match}}(y_i, \hat{y}_{\sigma(i)})$$

$$\mathcal{L}_{\text{Hungarian}}(y, \hat{y}) = \sum_{i=1}^{N} \left[-\log \hat{p}_{\hat{\sigma}(i)}(c_i) + \mathbb{1}_{\{c_i \neq \varnothing\}} \mathcal{L}_{\text{box}}(b_i, \hat{b}_{\hat{\sigma}}(i)) \right]$$

Trick : we down-weight the log-probability term when ci = Ø by a factor 10 to account for class imbalance

$$\mathcal{L}_{\text{box}}(b_i, \hat{b}_{\sigma(i)}) \qquad \lambda_{\text{iou}} \mathcal{L}_{\text{iou}}(b_i, \hat{b}_{\sigma(i)}) + \lambda_{\text{L1}} ||b_i - \hat{b}_{\sigma(i)}||_1$$

Classification loss
l1 bounding box distance loss
GIoU loss

Experiments

- Dataset
- Training

Experiments

- Dataset
  - COCO 2017
    - Average 7 instance in an image (maximal 63 instances)
    - AP refers to bbox AP

Experiments

- Dataset
- Training
  - Whole Architecture
  - Data Process
  - Backbone
  - Transformer
  - Losses

Experiments

- Dataset
- Training
  - Whole Architecture
    - AdamW with improved weight decay 10^(-4), maximum gradient norm

Experiments

- Dataset
- Training
  - Whole Architecture
  - Data Process
    - Scale augementation resize images

Experiments

- Dataset
- Training
    - Whole Architecture
    - Data Process
    - Backbone
        - Backbone batch normalization weights and statistics are frozen during training
        - Learning rate 10^(-5)

Experiments

- Dataset
- Training
  - Whole Architecture
  - Data Process
  - Backbone
  - Transformer
    - Learning rate 10^(-4)
    - Dropout 0.1

Experiments

- Dataset
- Training
  - Whole Architecture
  - Data Process
  - Backbone
  - Transformer
  - Losses

$$\lambda_{\mathrm{L1}} = 5 \text{ and } \lambda_{\mathrm{iou}} = 2$$

Panoptic segmentations

- Predicting Box
- Mask Head

Panoptic segmentations

- Predicting Box
- Mask Head