DETR Fruit Detector

심하은

목차

- 개요
- Dataset
- DETR
- Loss
- Hyper Parameters
- Training Process
- Result
- Reference & Source code

개요

- Object detection을 위해 DETR Model 개발
- Fruit dataset을 사용해 이미지에서 사과, 오렌지, 바나나를 탐지
- 사과, 오렌지, 바나나 탐지 시 Bounding box 생성

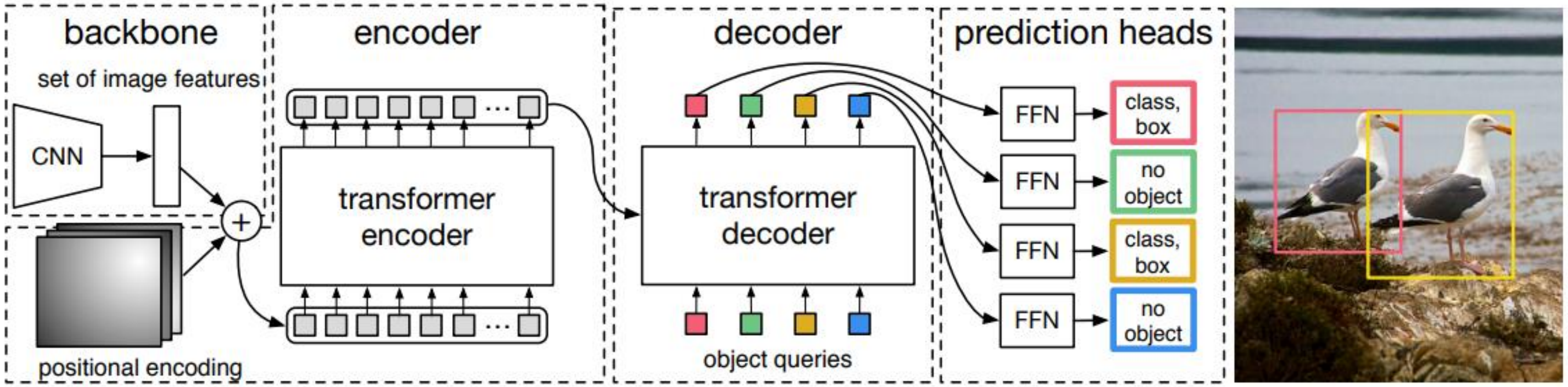
Dataset

Fruit Images for Object Detection

content: Apple, Banana, Orange

- [출처] https://www.kaggle.com/datasets/mbkinaci/fruit-images-for-object-detection

DETR(DEtection TRansformer) 구조



- Backbone
- Transformer encoder - window attention 기능 추가
- Transformer decoder
- Prediction feed-forward network

DETR - Backbone

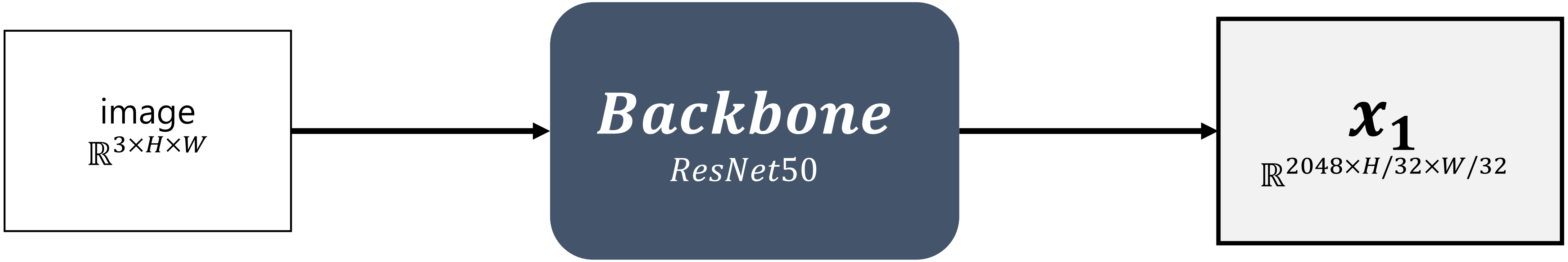
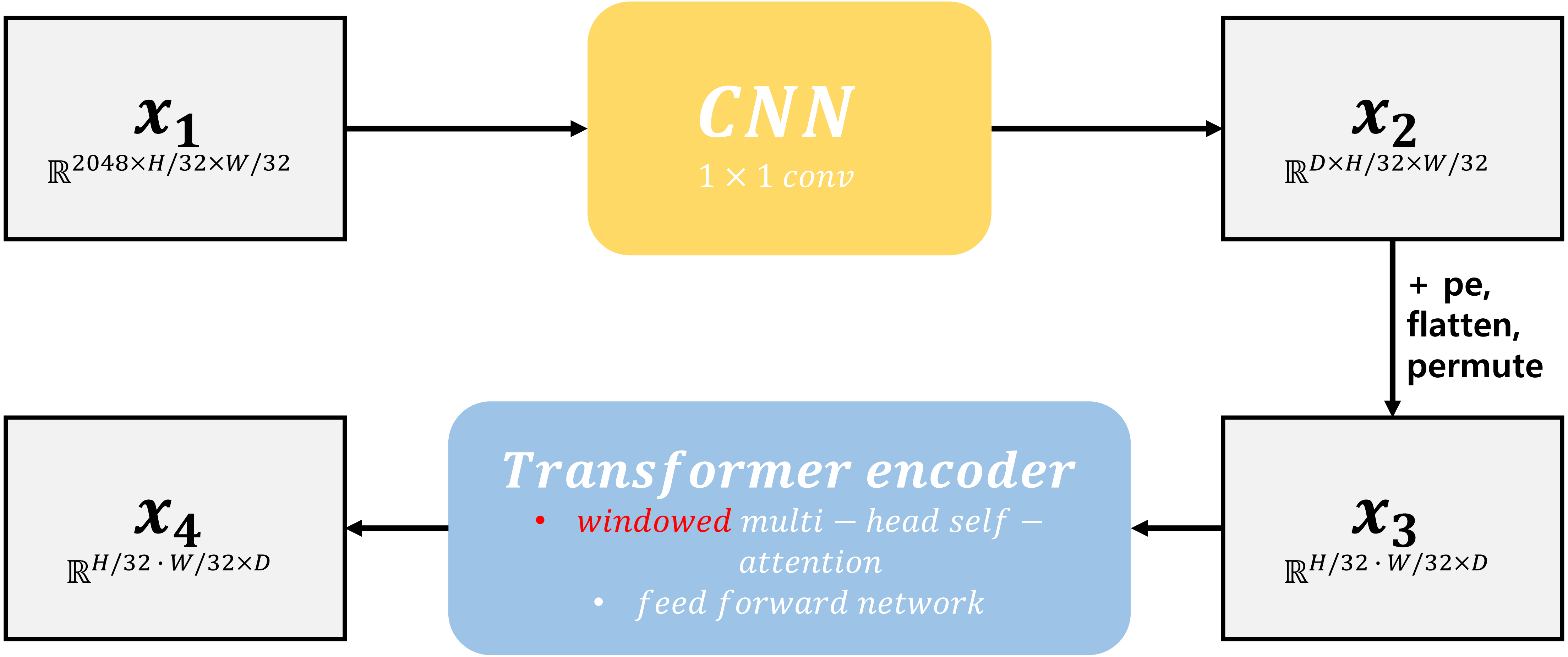


image
$\mathbb{R}^{3 \times H \times W}$
$\boldsymbol{Backbone}$
$ResNet50$
$\boldsymbol{x_1}$
$\mathbb{R}^{2048 \times H/32 \times W/32}$

DETR – Transformer encoder



x_1

$\mathbb{R}^{2048\times H/32\times W/32}$

CNN

$1\times 1\ conv$

x_2

$\mathbb{R}^{D\times H/32\times W/32}$

+ pe,
flatten,
permute

x_3

$\mathbb{R}^{H/32\cdot W/32\times D}$

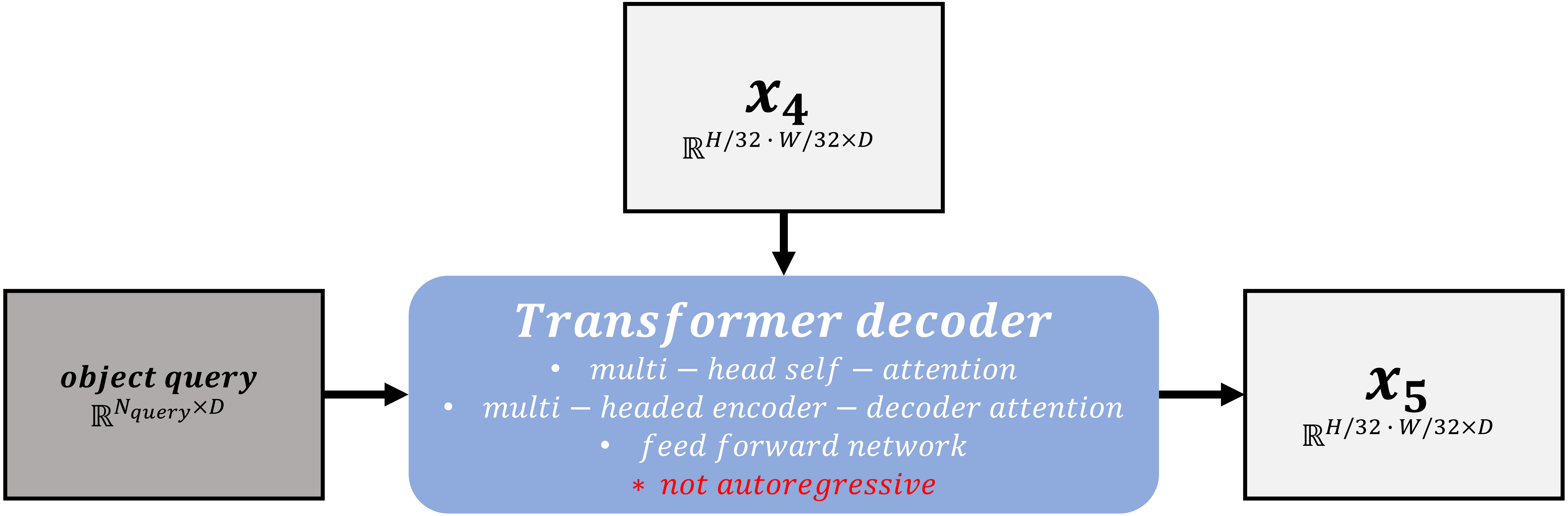
Transformer encoder

- *windowed multi − head self − attention*
- *feed forward network*

x_4

$\mathbb{R}^{H/32\cdot W/32\times D}$

DETR – Transformer decoder



x_4

$\mathbb{R}^{H/32 \cdot W/32 \times D}$

object query

$\mathbb{R}^{N_{query} \times D}$

Transformer decoder

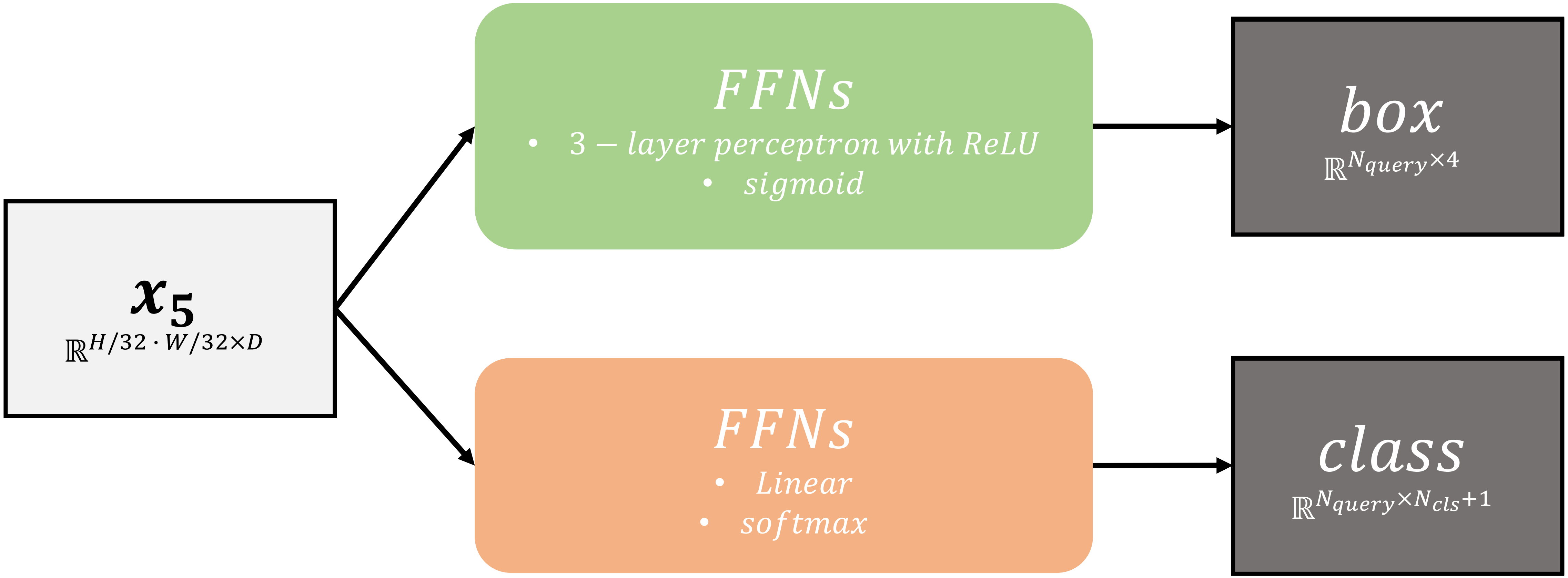
- *multi − head self − attention*
- *multi − headed encoder − decoder attention*
- *feed forward network*

∗ *not autoregressive*

x_5

$\mathbb{R}^{H/32 \cdot W/32 \times D}$

DETR – Prediction feed-forward networks(FFNs)



x_5

$\mathbb{R}^{H/32\cdot W/32\times D}$

$FFNs$

- $3 - layer\ perceptron\ with\ ReLU$
- $sigmoid$

box

$\mathbb{R}^{N_{query}\times 4}$

$FFNs$

- $Linear$
- $softmax$

$class$

$\mathbb{R}^{N_{query}\times N_{cls}+1}$

Loss

$$\boldsymbol{index\ \sigma(i), probability\ of\ class\ c_i\ as\ \hat{p}_{\sigma(i)}\ (c_i\ ), predicted\ box\ as\ \hat{b}_{\sigma(i)}}$$

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

$$\mathcal{L}_{match}\left(y_i, \hat{y}_{\sigma(i)}\right) = -\mathbb{1}_{\{c_i \neq \emptyset\}}\hat{p}_{\sigma(i)}(c_i) + \mathbb{1}_{\{c_i \neq \emptyset\}}\mathcal{L}_{box}(b_i, \hat{b}_{\sigma(i)})$$

$$\hat{\sigma} = argmin_{\sigma \in \mathfrak{S}_N} \sum_{i}^{N} \mathcal{L}_{match}\left(y_i, \hat{y}_{\sigma(i)}\right)$$

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

Hyper Parameter

- image size = 224
- batch size = 32

- dim = 256
- Transformer nheads = 8
- attention window size = 7
- global attention iter = 2

- Transformer encoder depth = 6
- Transformer decoder depth = 6

- n_query = 100

- learning rate = 1e-4
- Training epoch = 1000

Training Process

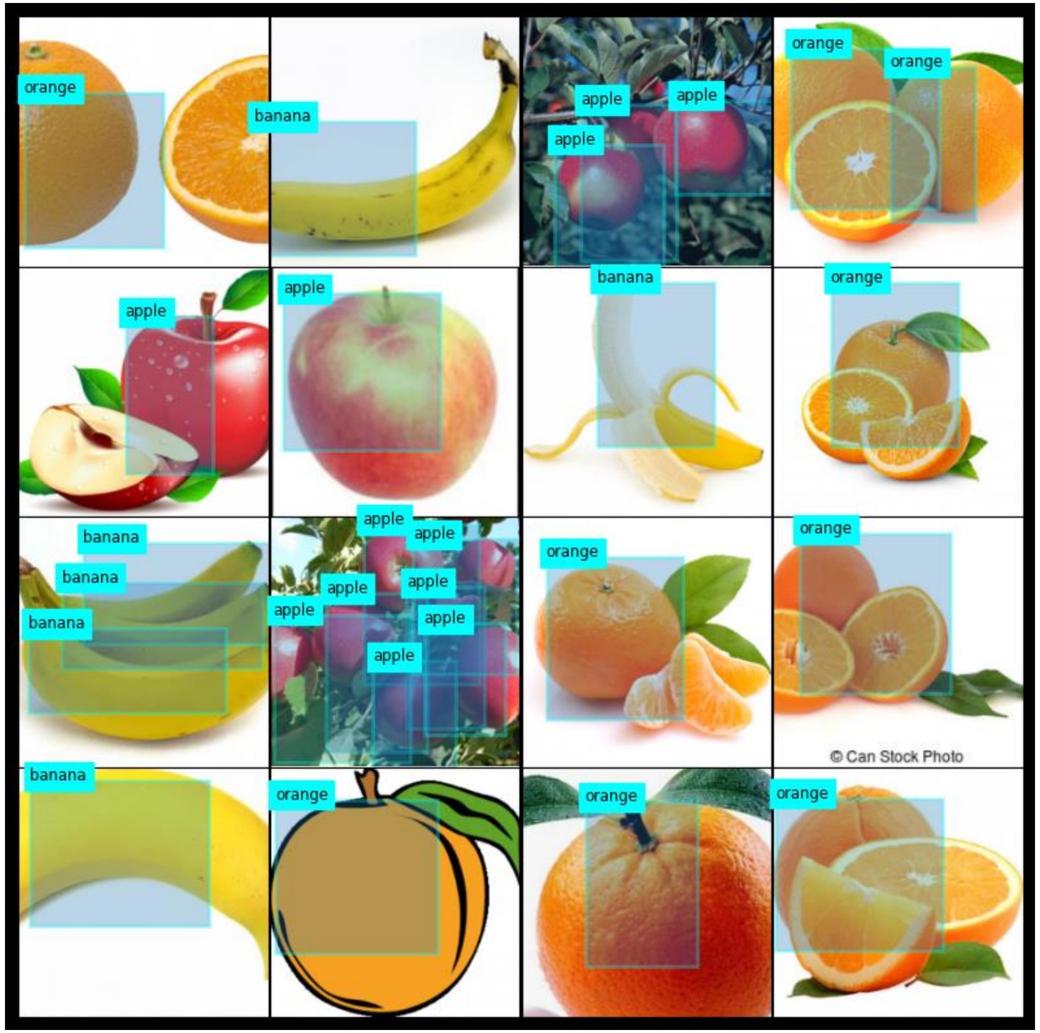
- Fruit dataset의 이미지 preprocessing 진행
  - Image Resizing: C X H X W ->3 X 224 X 224
  - Padding

- Model의 input으로 preprocessing된 이미지와 100개의 object query 사용

- Transformer encoder에서 window attention 수행, 3번에 한번 씩 global attention 수행
- Model의 prediction과 ground truth간의 $\mathcal{L}_{match}$을 구해 총 loss의 합계가 작은 pair간 $\mathcal{L}_{Hungarian}$ 계산

- Adam optimizer(lr=1e-4)을 사용 및 gradient clipping(max norm=0.1) 적용하여 학습

Result – 1000 epoch



Reference

- [DETR] https://arxiv.org/pdf/2005.12872.pdf
- [window attention] https://arxiv.org/pdf/2203.16527.pdf

Source code

- [Notebook] https://github.com/5121eun/dl/blob/main/detr_example.ipynb
- [Model] https://github.com/5121eun/dl/blob/main/models/detr.py