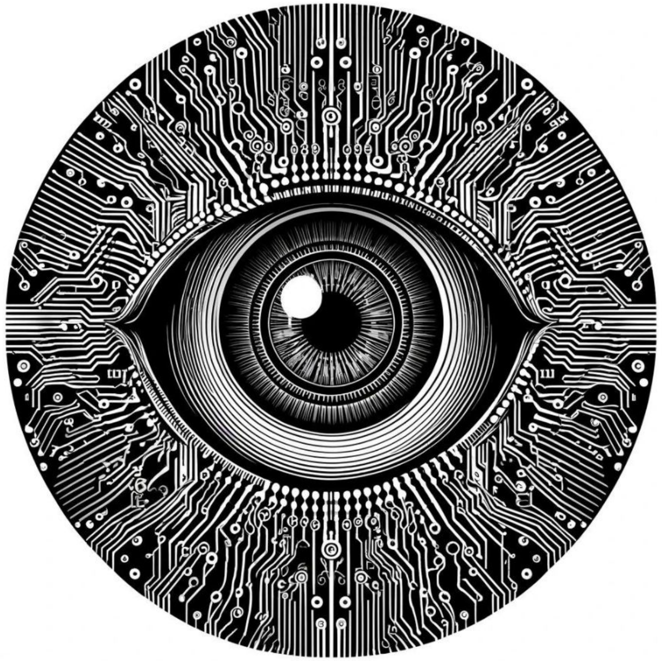
KI Service Zentrum by Hasso-Plattner-Institut HPI

Approaches to Object Detection



Antonio Rueda-Toicen

SPONSORED BY THE
Federal Ministry of Education and Research

Learning goals

- Recognize object detection as a regression and classification problem
- Describe the use of anchor boxes on single shot detectors (RetinaNet, YOLOv1-v5) and two-stage detectors (Faster R-CNN)
- Gain familiarity with anchor-box-free object detection approaches (YOLOv6+, DETR, Grounding DINO)

Object detection as bounding box localization

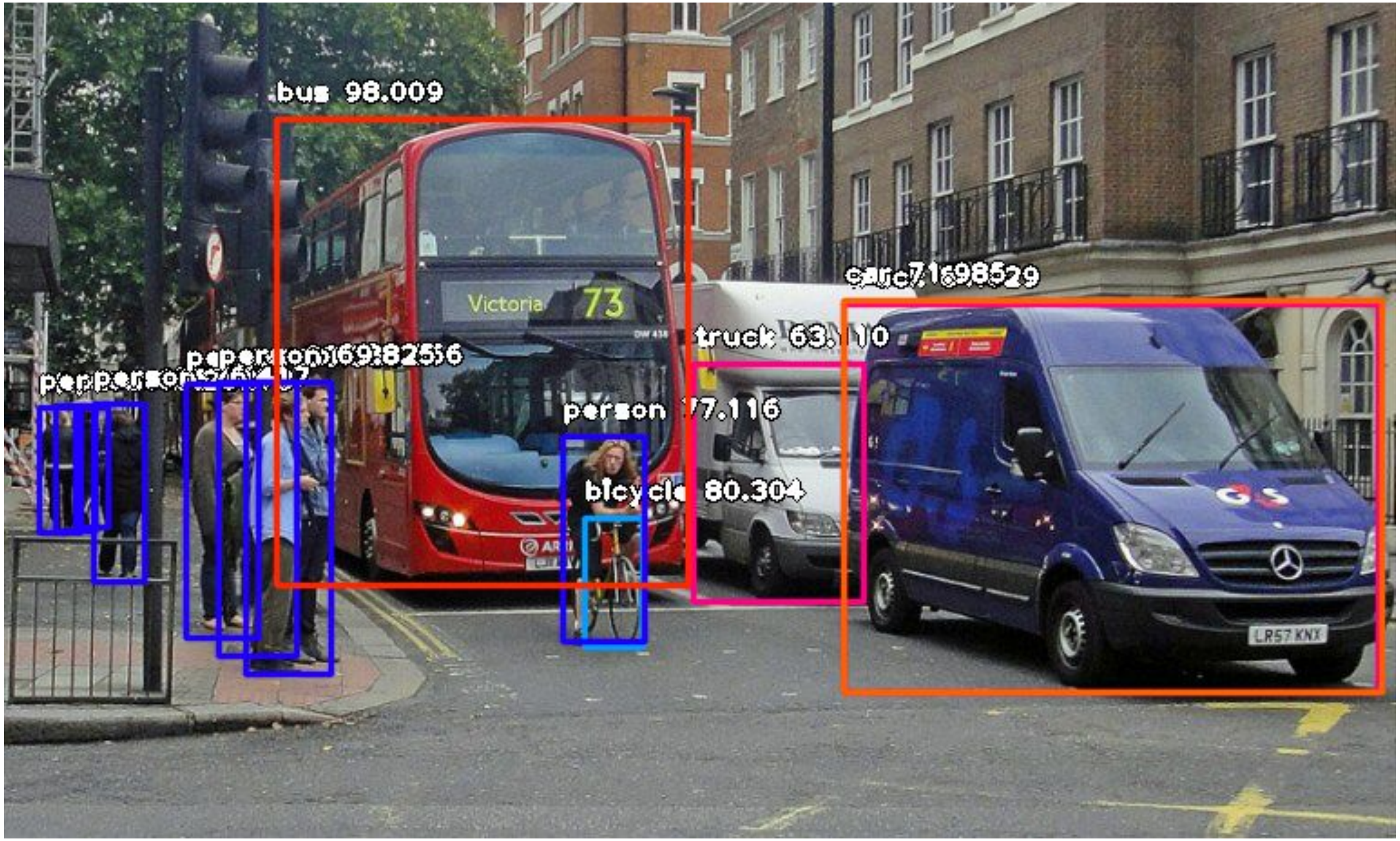
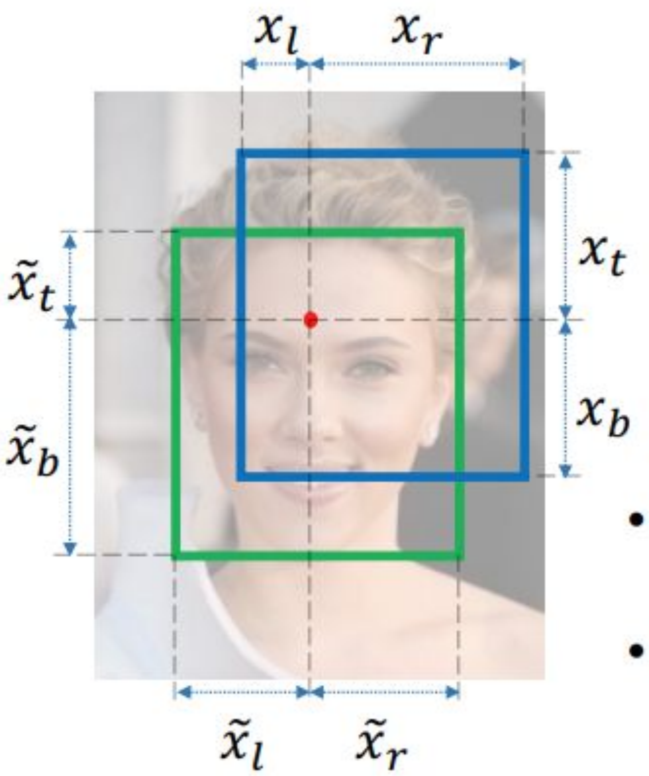


Image source

Object detection as bounding box regression and classification



Ground truth: $\tilde{x} = (\tilde{x}_t, \tilde{x}_b, \tilde{x}_l, \tilde{x}_r)$

Prediction: $x = (x_t, x_b, x_l, x_r)$

- $\ell_2 \;\; loss = ||\square - \square||_2^2$
- $IoU\ loss = -\ln \frac{Intersection(\square, \square)}{Union(\square, \square)}$

Image from source

‘Regression’ = predicting a continuous value (bounding box coordinates)

Loss functions combine classification and regression error

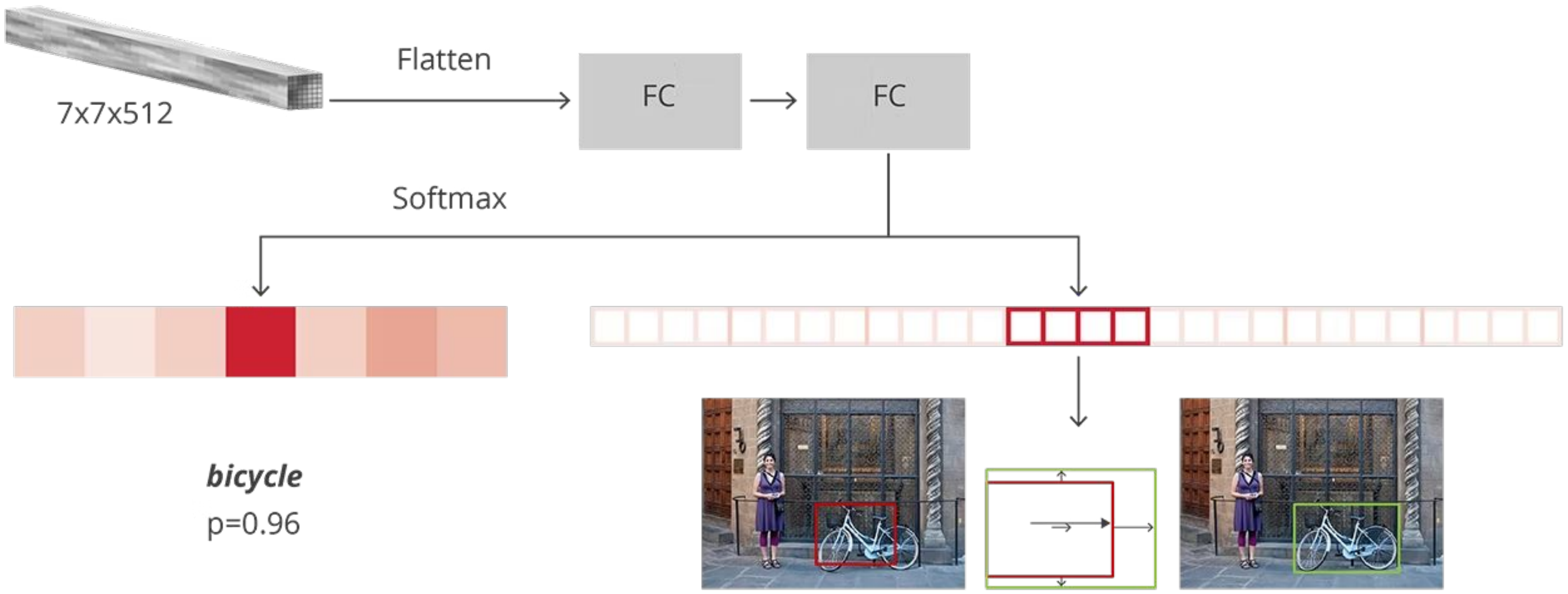
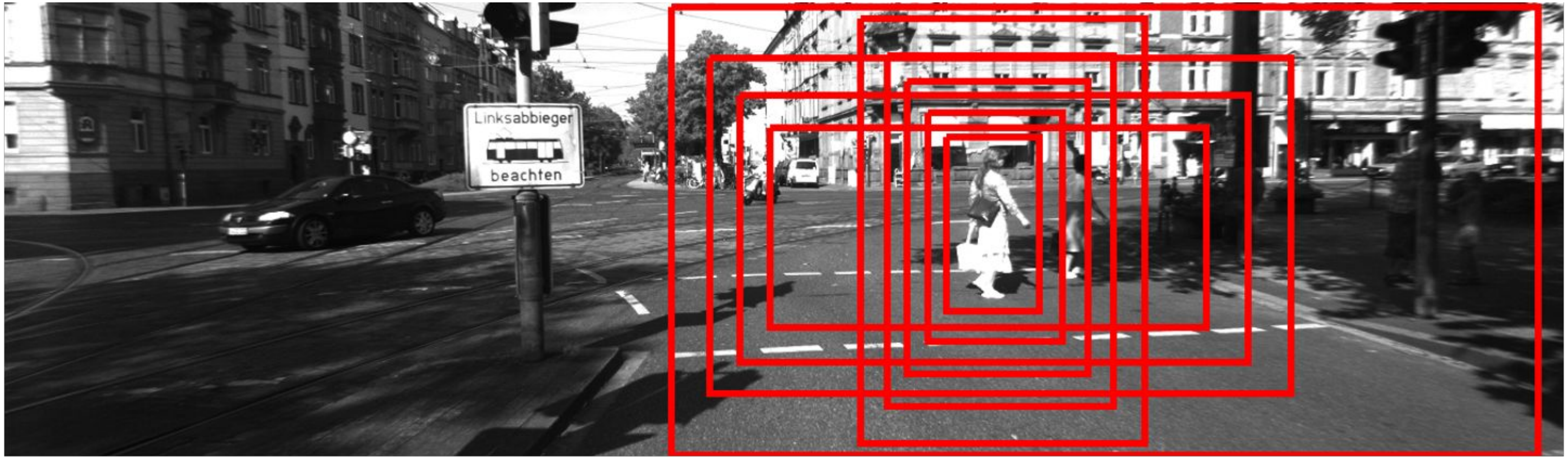


Image from Faster R-CNN: Down the rabbit hole of modern object detection

$$\mathcal{L} = -\sum_{x} P(x)\log(Q(x)) + \lambda\frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$$

Partitioning an image into regions



Possible approaches:

- Anchor-boxes
- Keypoint identification

Anchor boxes - defining potential detections

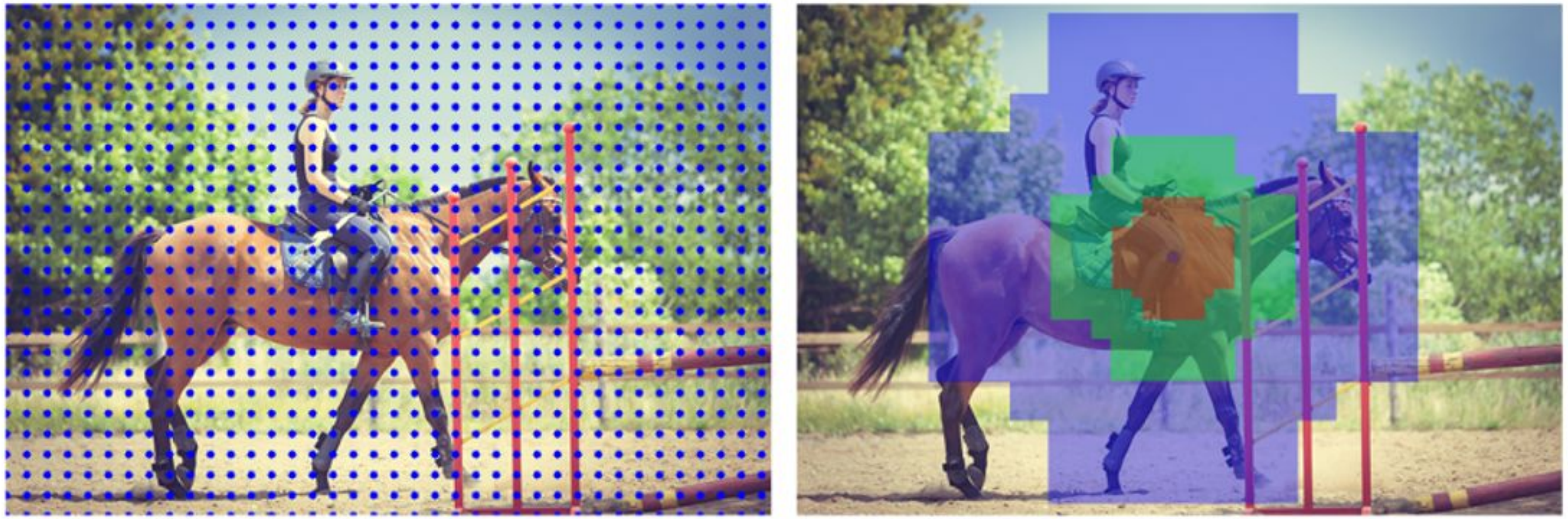


Figure 14.7: **Left:** Creating anchors starts with the process of sampling the coordinates of an image every r pixels ($r = 16$ in the original Faster R-CNN implementation). **Right:** We create a total of nine anchors centered around *each* sampled (x, y)-coordinate. In this visualization, $x = 300, y = 200$ (center blue coordinate). The nine total anchors come from every combination of scale: 64×64 (red), 128×128 (green), 256×256 (blue); and aspect ratio: $1:1$, $2:1$, $1:2$.

Image from [Faster R-CNNs - PyImageSearch]

Anchor boxes - defining sizes and aspect ratios

Image from source

Anchor boxes - the challenge of filtering candidates

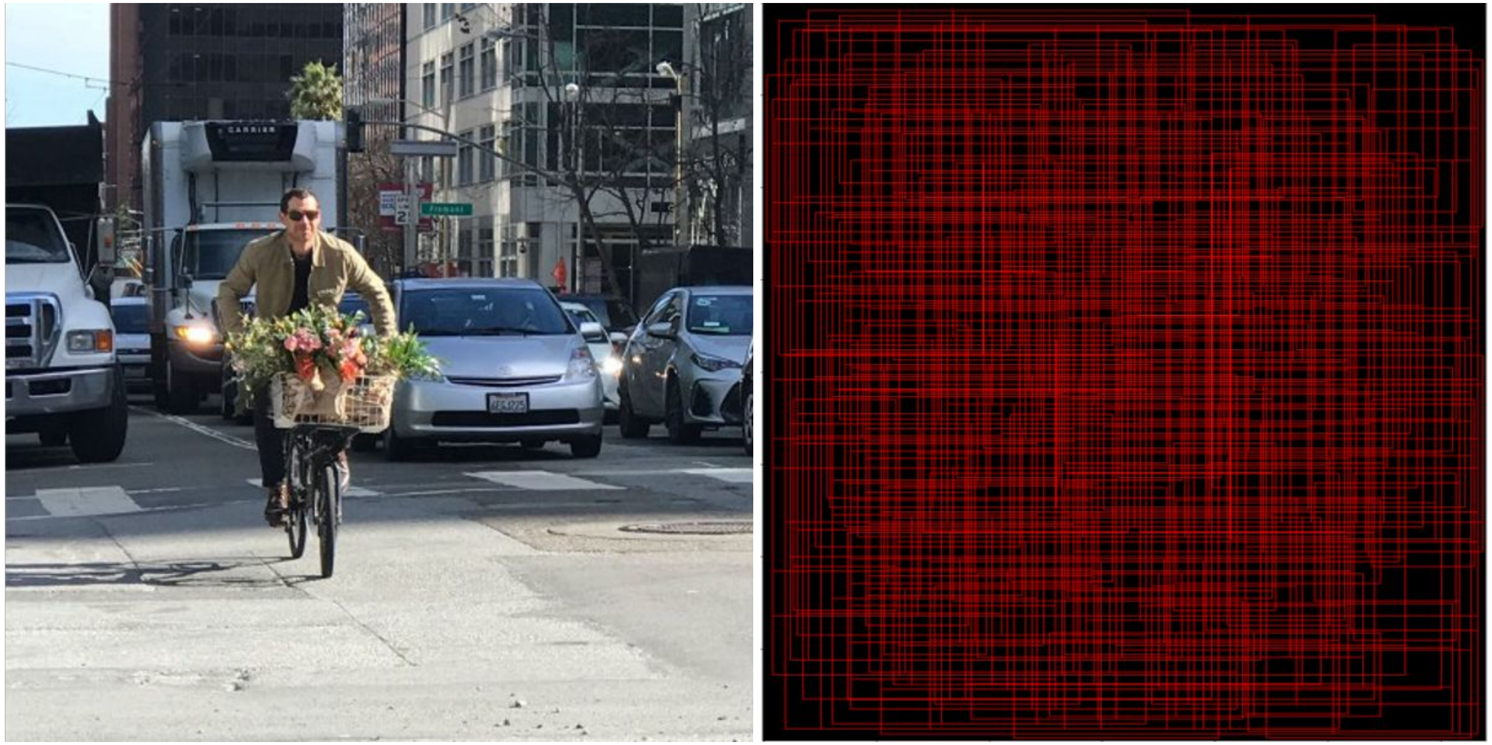
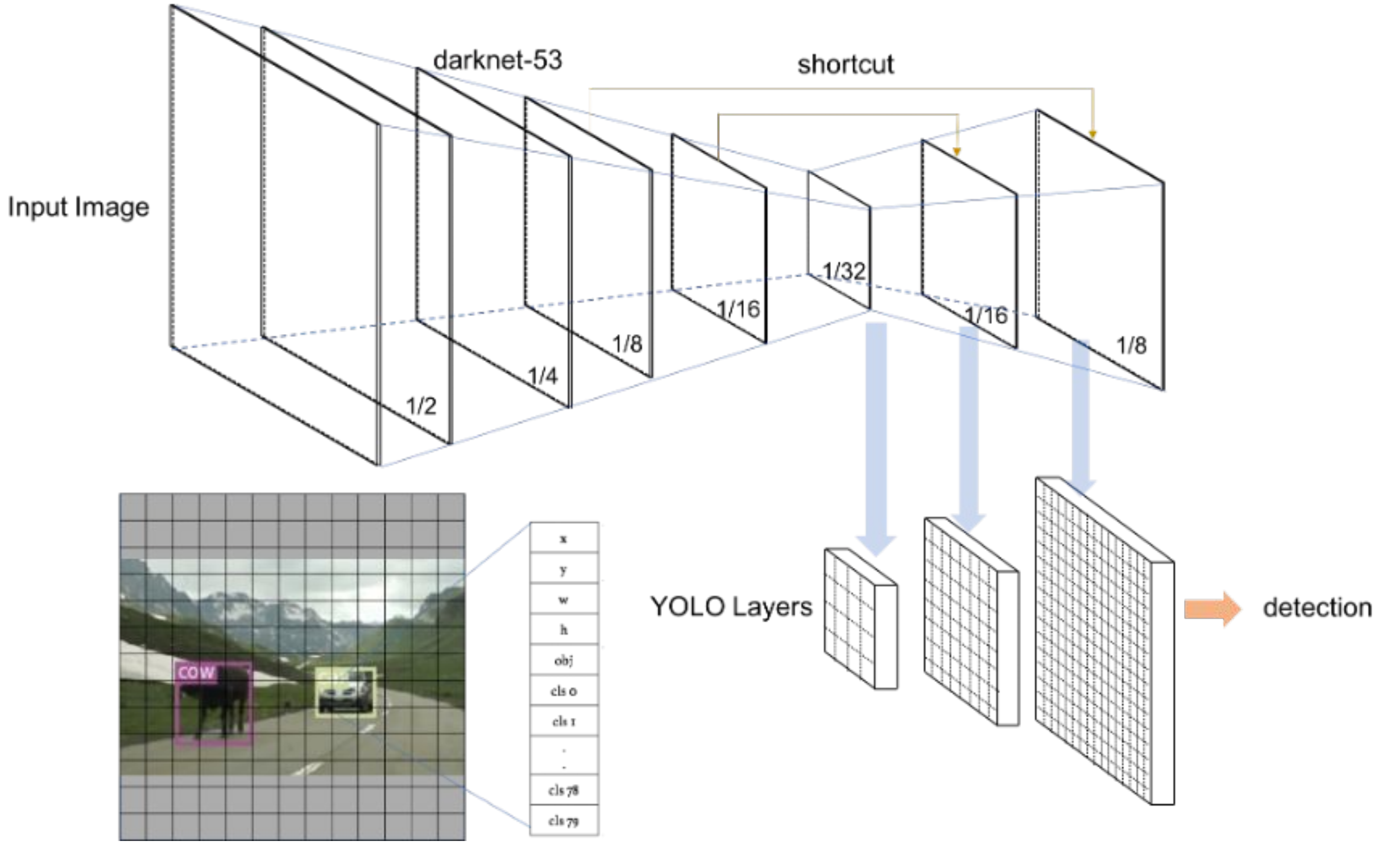


Image from [source]

Single shot detectors vs two stage detectors

YOLO (original)



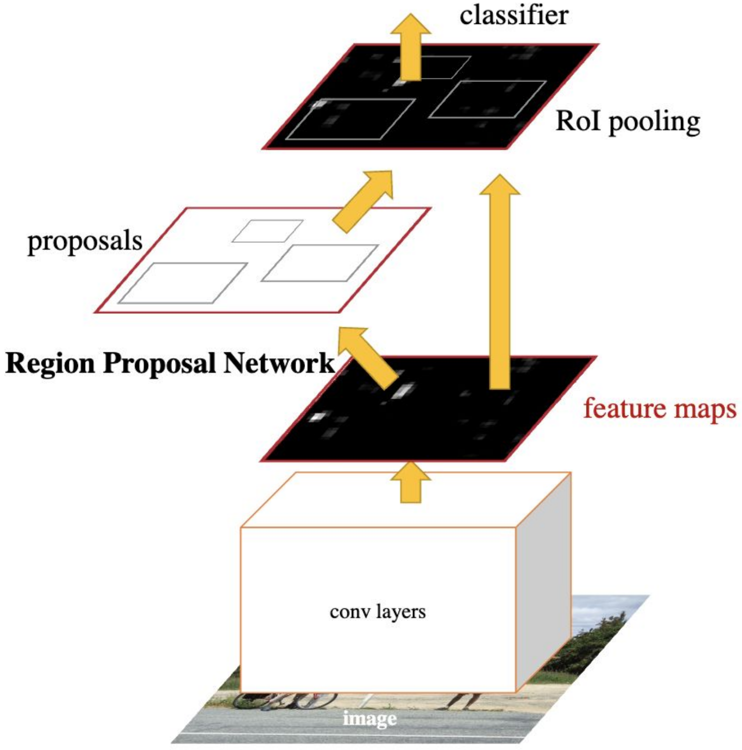
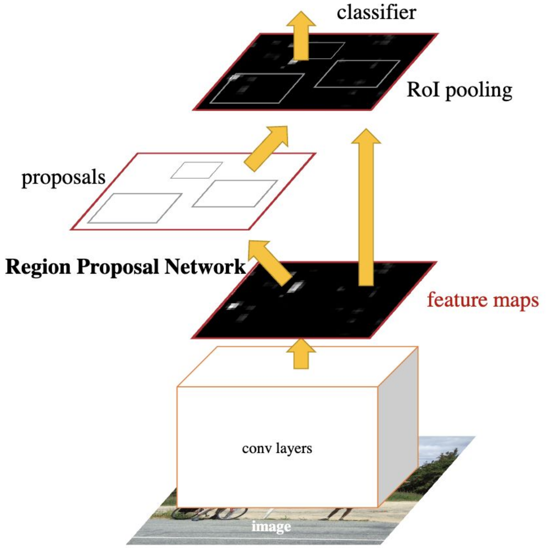


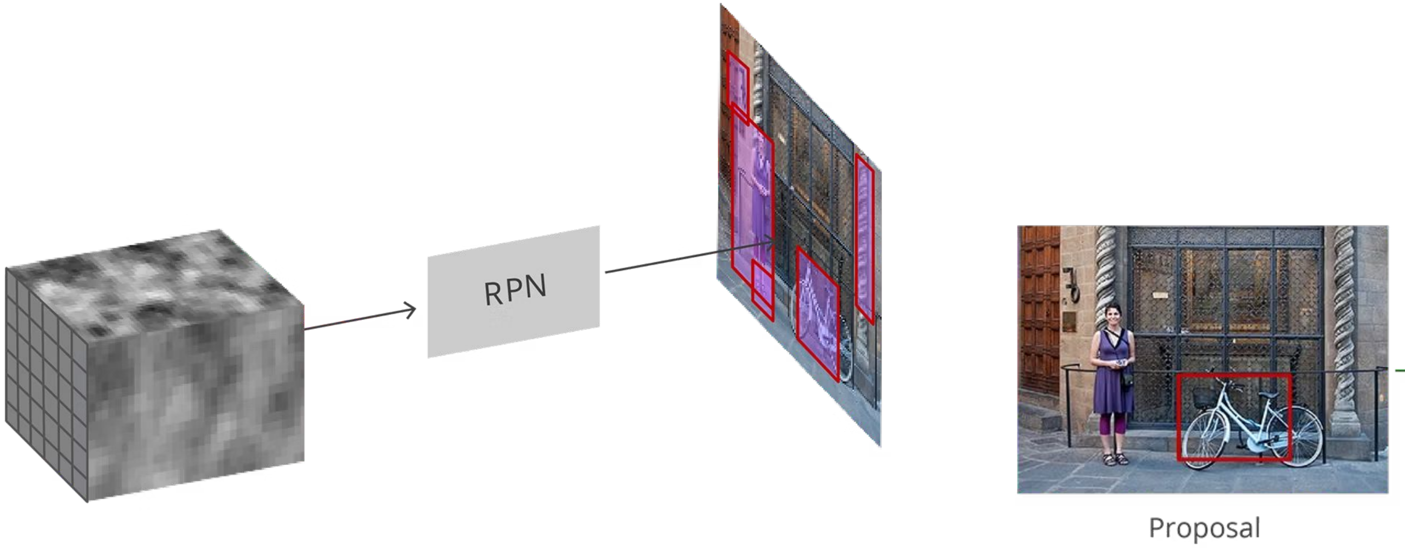
Image from YOLO: Real-Time Object Detection

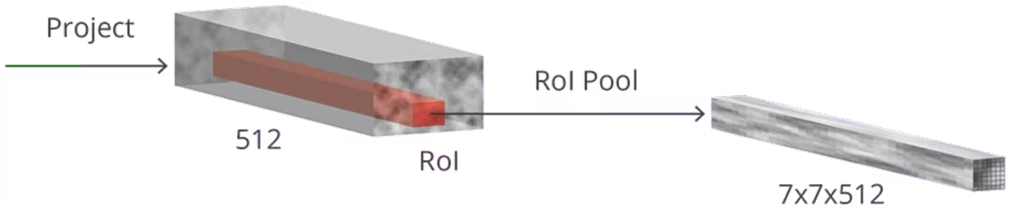
Image from Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks

Faster R-CNN’s Region Proposal Network (RPN)

The RPN takes the convolutional feature map and generates unlabeled proposals over the imag







Region of Interest (ROI) pooling uses the downsampled original features cropped on the proposal area to feed the classifier

Images from [Faster R-CNN: Down the rabbit hole of modern object detection]

Grid-based anchor-free detection (YOLOv11)

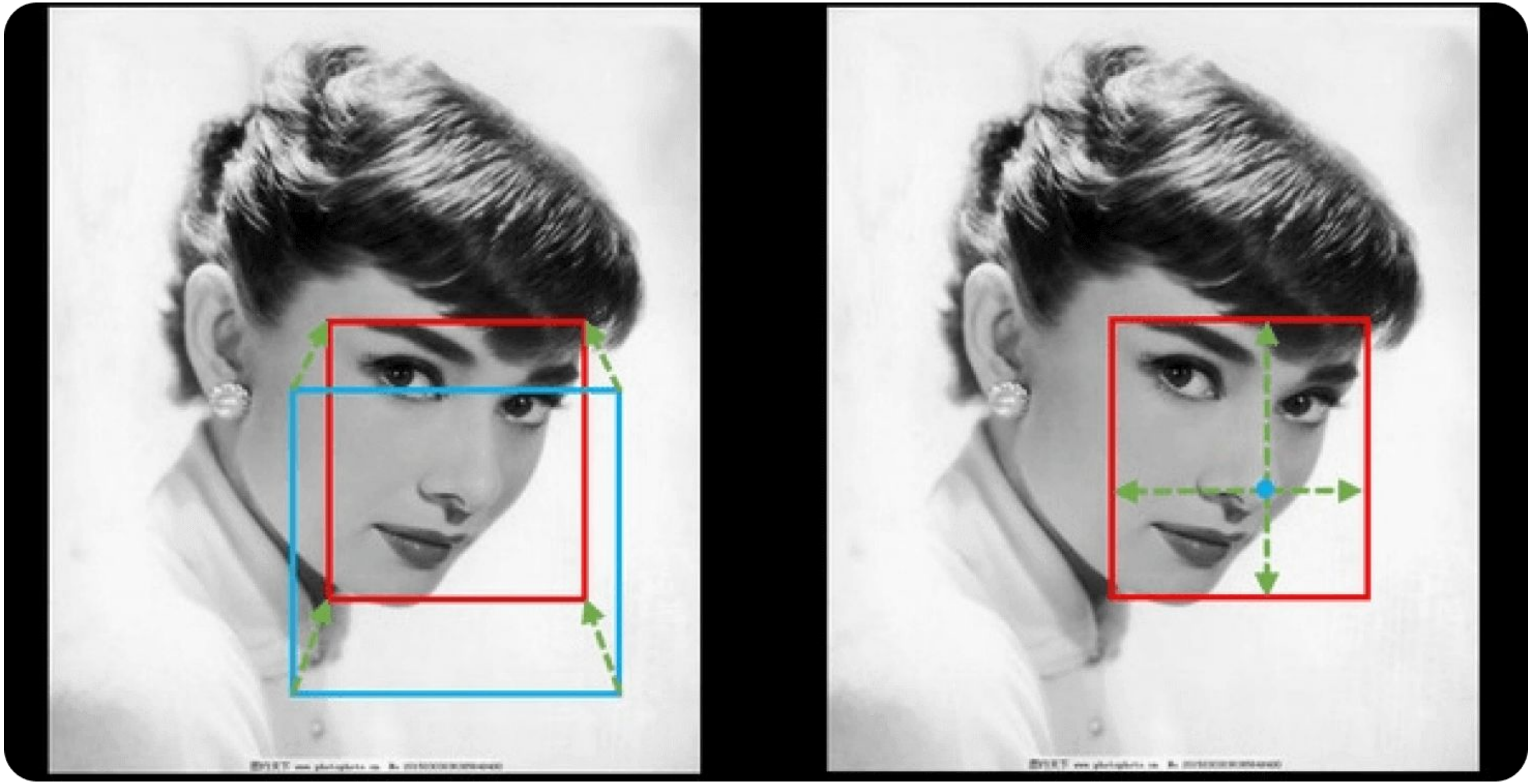


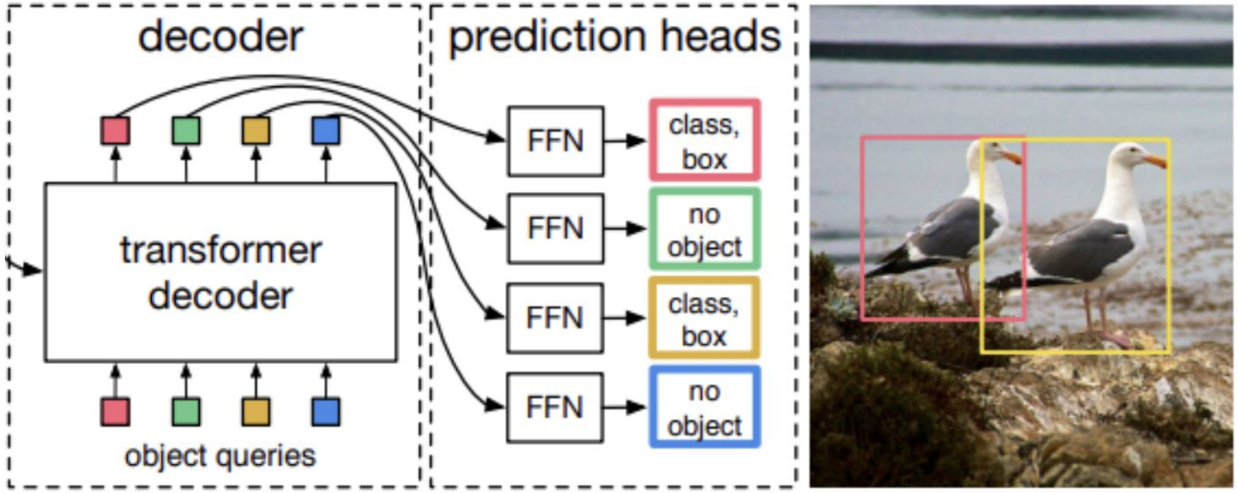
Image from the [Ultralytics blog]

Anchor-based vs anchor-free detection

- Require prior knowledge or aspect ratios and sizes of potential anchor boxes
- More settings to tune
- Remain competitive in different object scales and with partially occluded objects (Faster R-CNN excels at this)

- Fewer hyperparameters (less tuning)
- Difficulty handling partially occluded objects

DETR's object queries replace anchor boxes



- Unlike achor boxes: no geometric prior
- Learned embeddings with same dimension as all other embedded components of DETR
- Each object query embedding specializes on a region
- 100 by default, max number of detections
- Output class and bounding box after FFN

```
predictions = [ {"query": 1, "class": "bird", "box": [0.2, 0.3, 0.1, 0.1]},
                {"query": 2, "class": "bird", "box": [0.5, 0.6, 0.2, 0.2]} ]
```

Language-guided detection with Grounding DINO

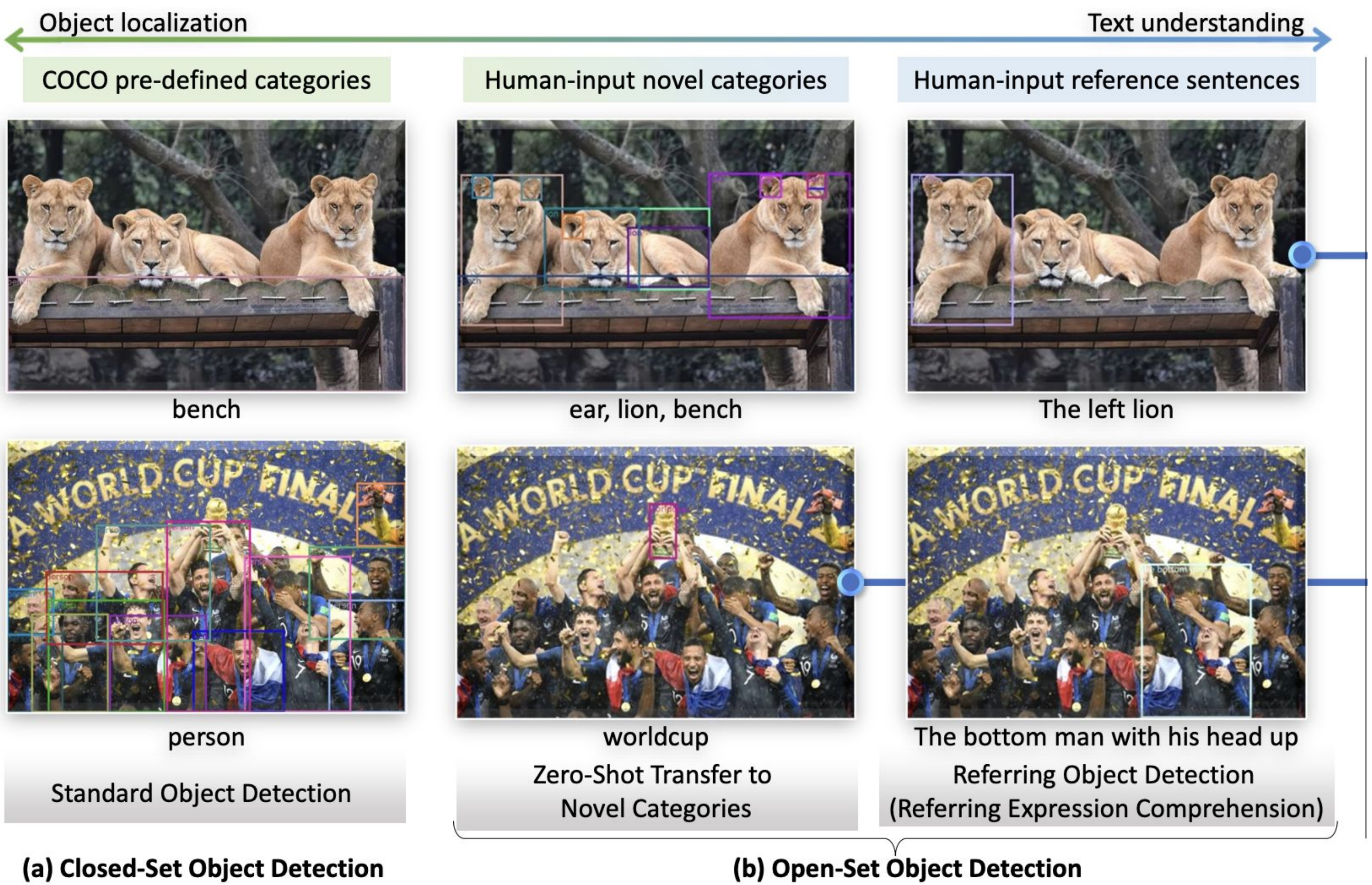


Image from [Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection]

Grounding DINO's architecture

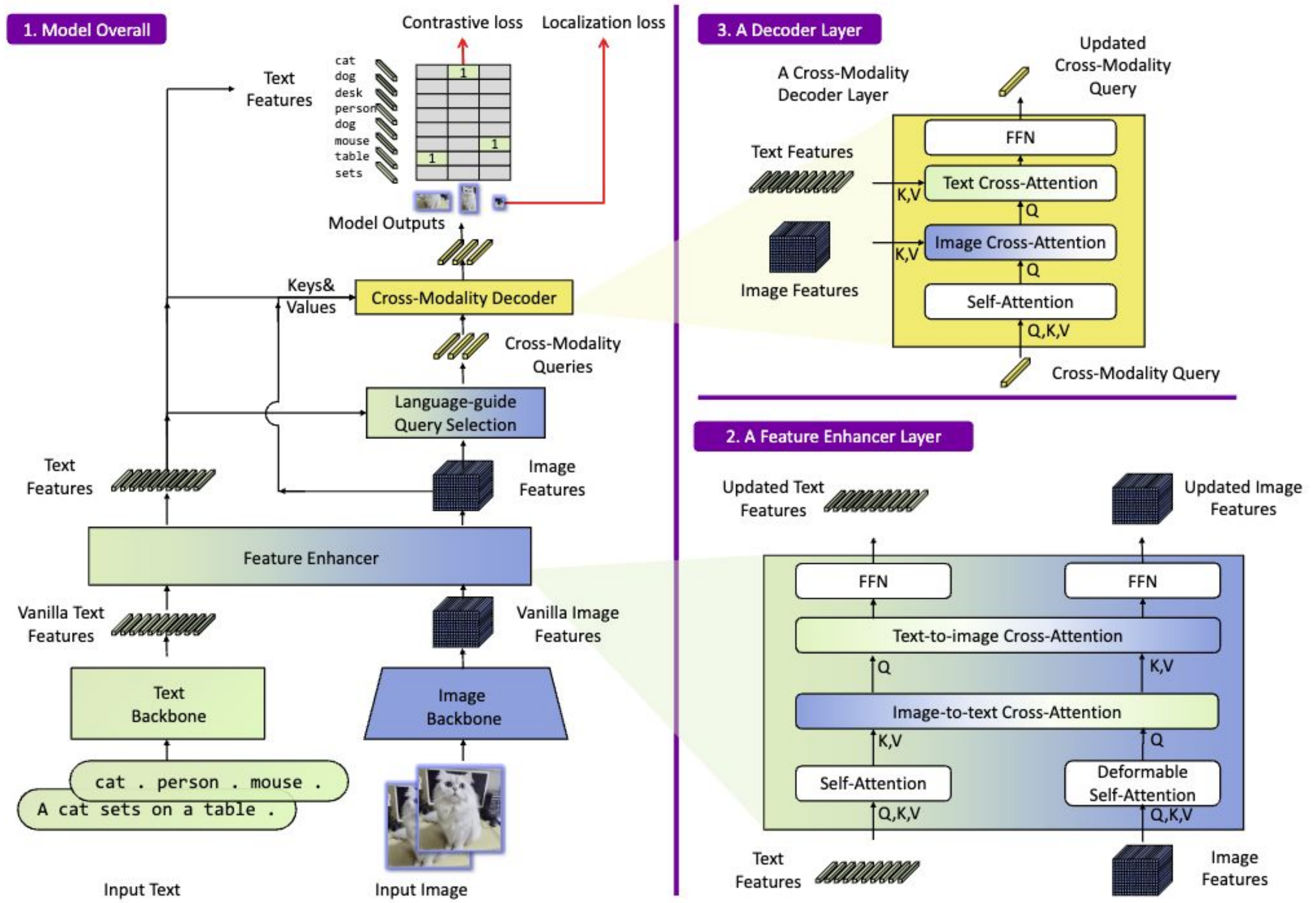


Figure 3. The framework of Grounding DINO. We present the overall framework, a feature enhancer layer, and a decoder layer in block 1, block 2, and block 3, respectively.

- 0.8M bounding box annotations
- 4M text-image pairs from Object365 and COCO
- 3M text-image pairs from Google's CC3M dataset

Image from [Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection]()

Summary

Object detection combines classification and bounding box regression

- Regression loss functions (e.g. L1, L2) quantify the positioning error of bounding boxes, cross-entropy quantifies the classification error

Anchor box-based detectors

- Anchor boxes define potential object locations, scales, and aspect ratios
- Two stage detectors use region proposal networks to filter candidate anchor boxes, and tend to have higher accuracy at higher computation cost
- The number and variety of anchor boxes influences the performance of the model

Anchor-free and grounded detection

- DETR removes anchor boxes by using object queries and the attention mechanism
- Grounding DINO enables language-guided detection, with an accuracy tradeoff vs fully supervised models

SPONSORED BY THE
Federal Ministry of Education and Research

References and further reading

Focal Loss for Dense Object Detection

- https://arxiv.org/abs/1708.02002

End-to-End Object Detection with Transformers

- https://arxiv.org/abs/2005.12872
- https://www.youtube.com/watch?v=utxbUIo9CyY

Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks

- https://arxiv.org/abs/1506.01497
- https://pyimagesearch.com/2023/11/13/faster-r-cnns/

Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection

- https://arxiv.org/abs/2303.05499

SPONSORED BY THE

Federal Ministry of Education and Research