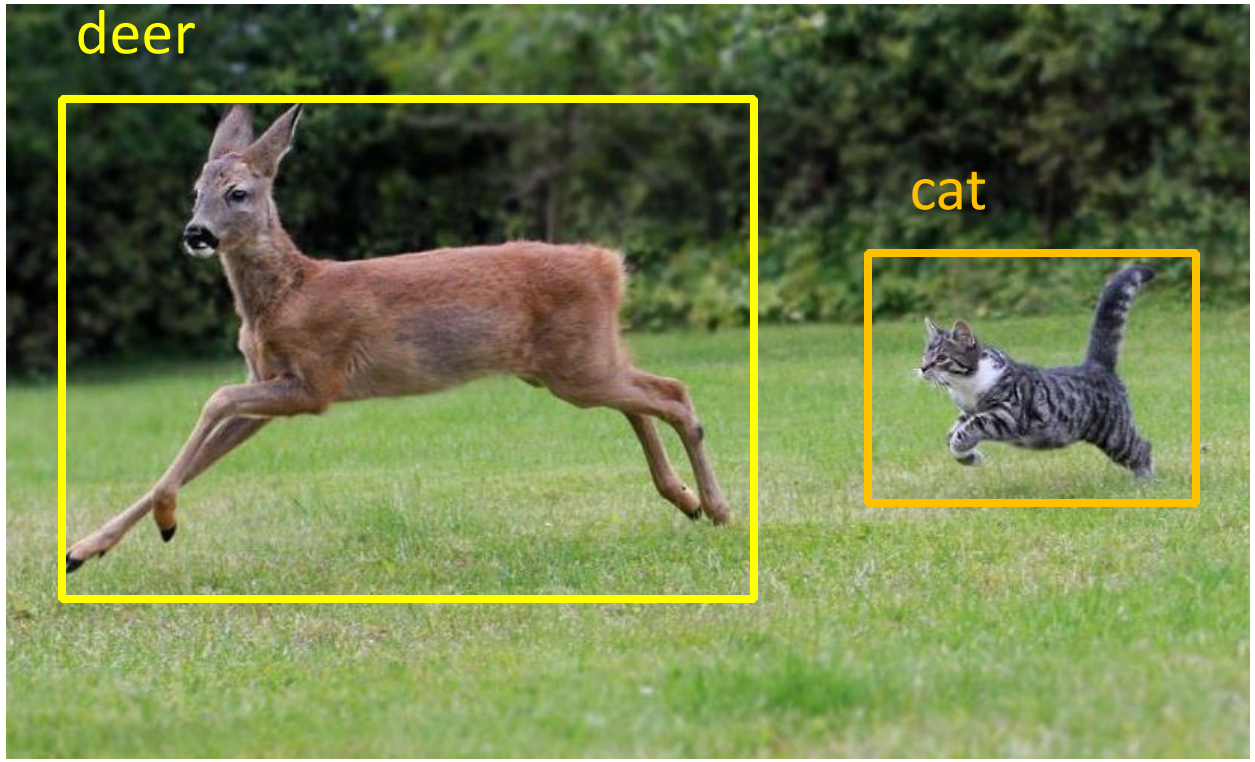
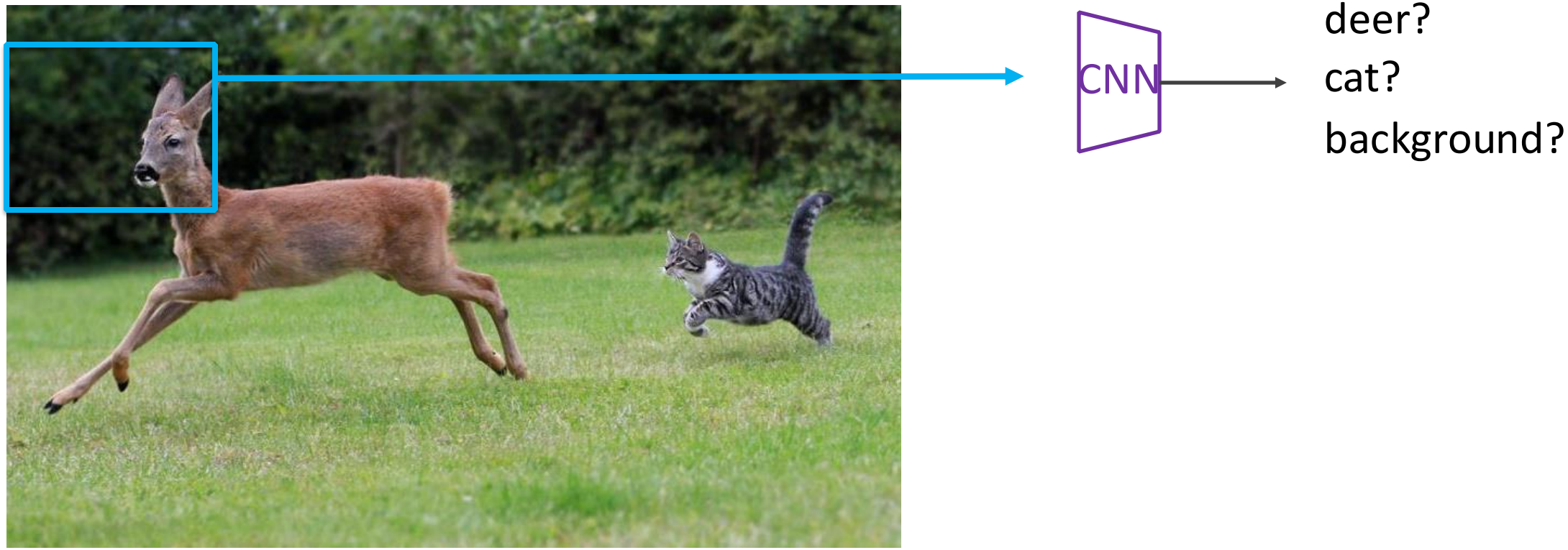
Deep Learning for Vision & Language

Convolutional Neural Networks for Object Detection

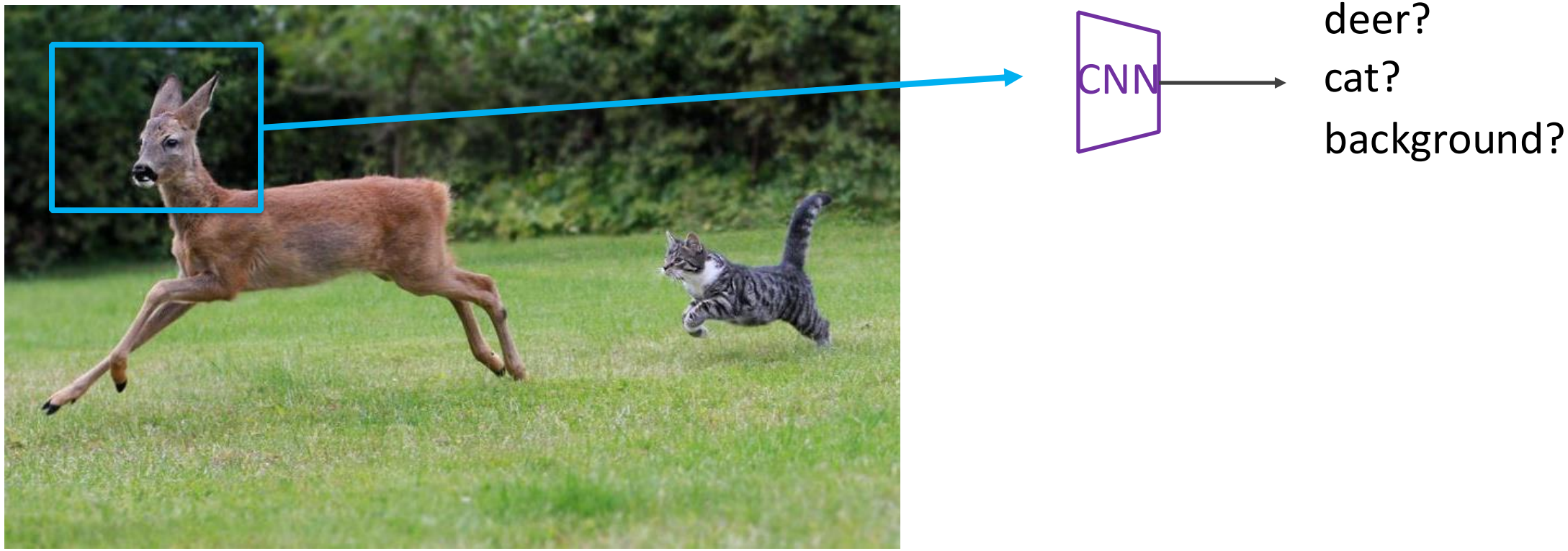
Object Detection



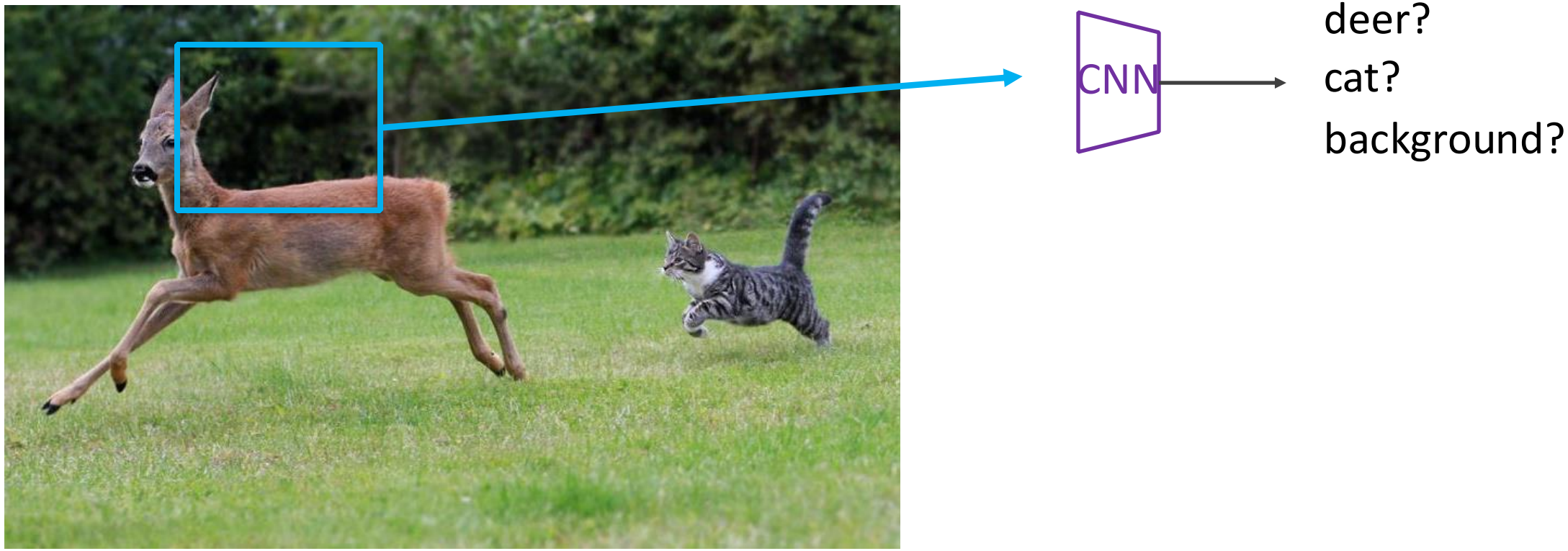
Object Detection as Classification



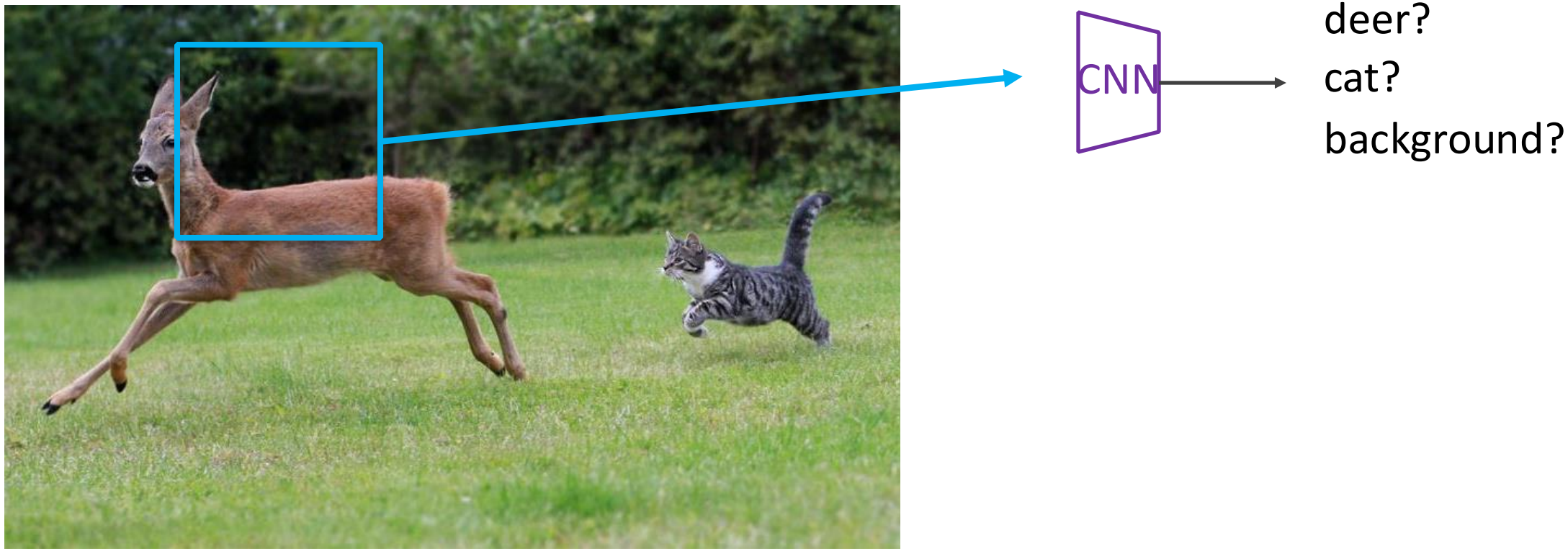
Object Detection as Classification



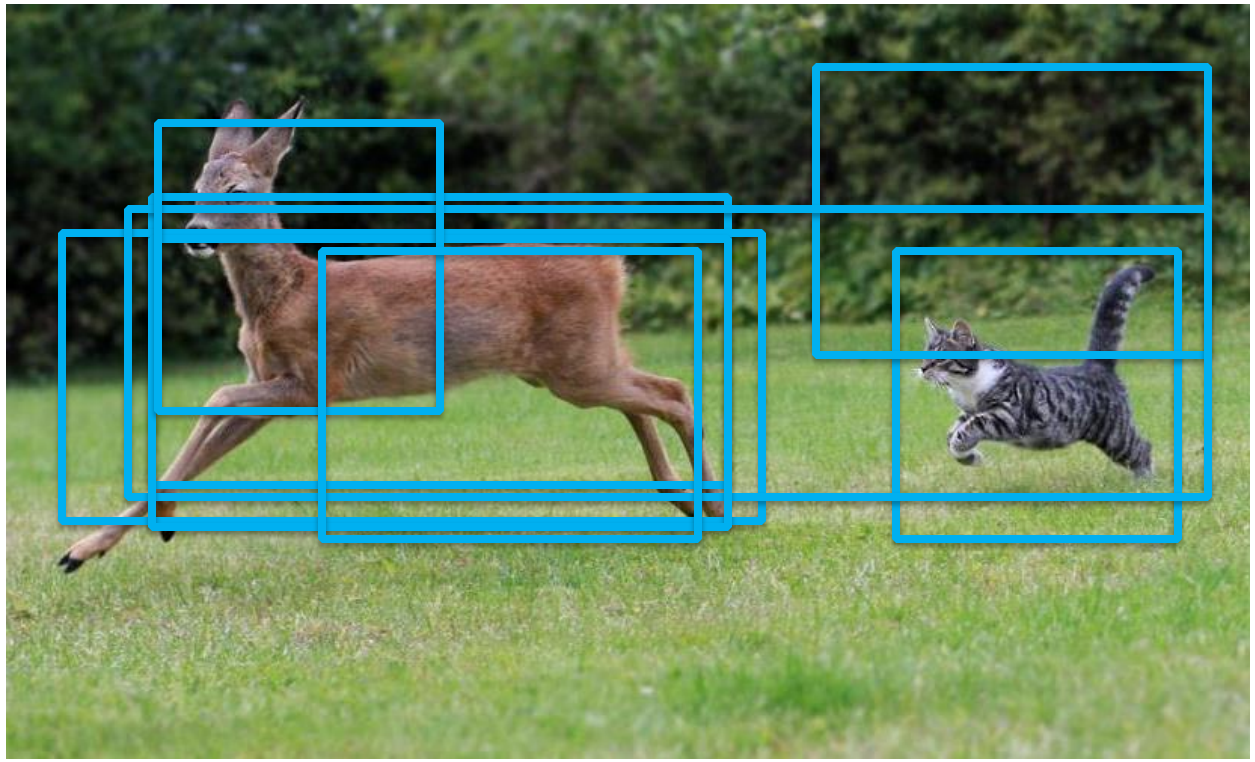
Object Detection as Classification



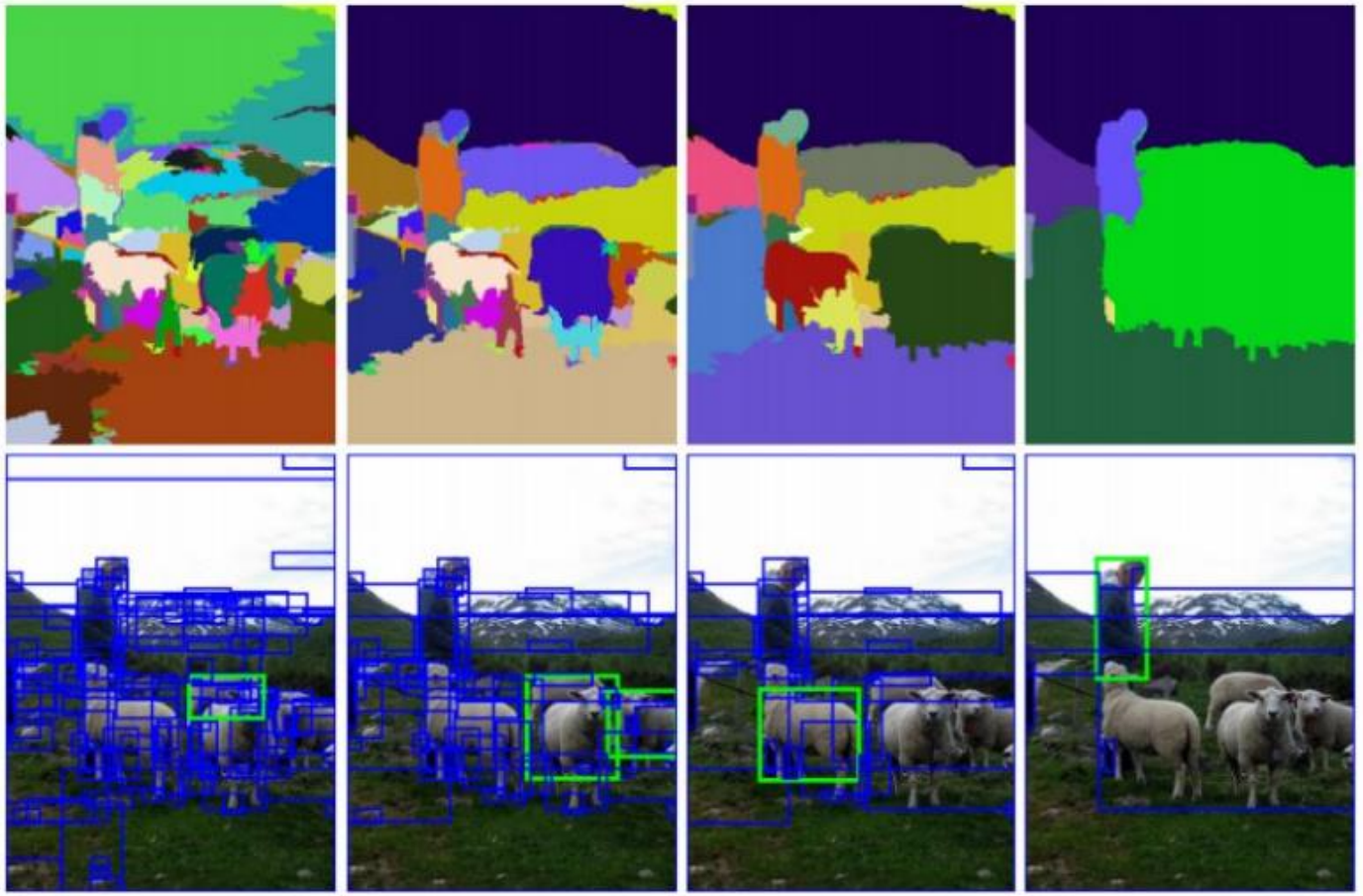
Object Detection as Classification with Sliding Window



Object Detection as Classification with Box Proposals

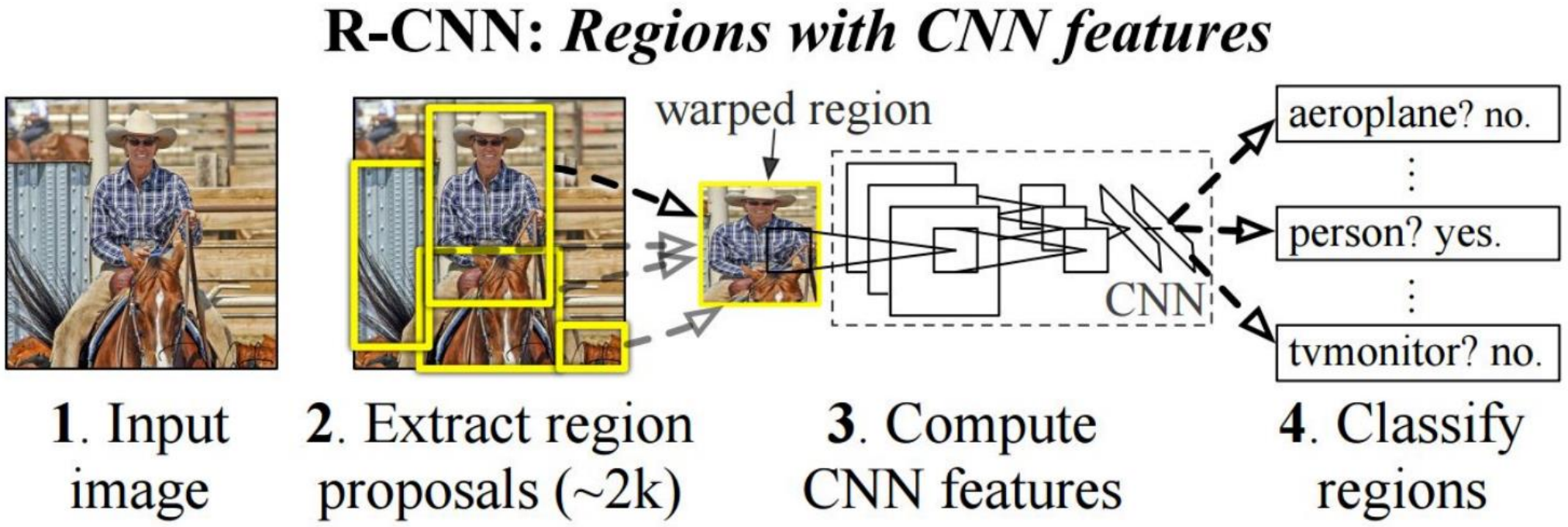


Box Proposal Method – SS: Selective Search



Segmentation As Selective Search for Object Recognition. van de Sande et al. ICCV 2011

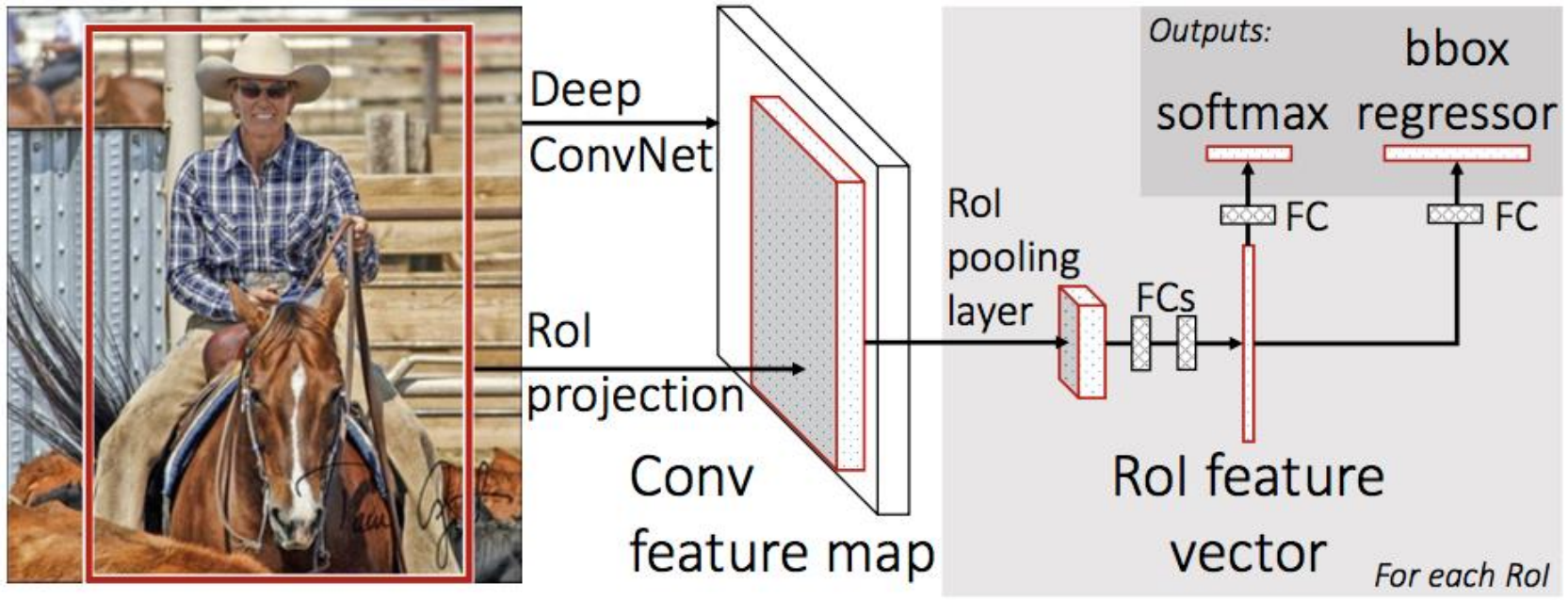
RCNN



https://people.eecs.berkeley.edu/~rbg/papers/r-cnn-cvpr.pdf

Rich feature hierarchies for accurate object detection and semantic segmentation. Girshick et al. CVPR 2014.

Fast-RCNN



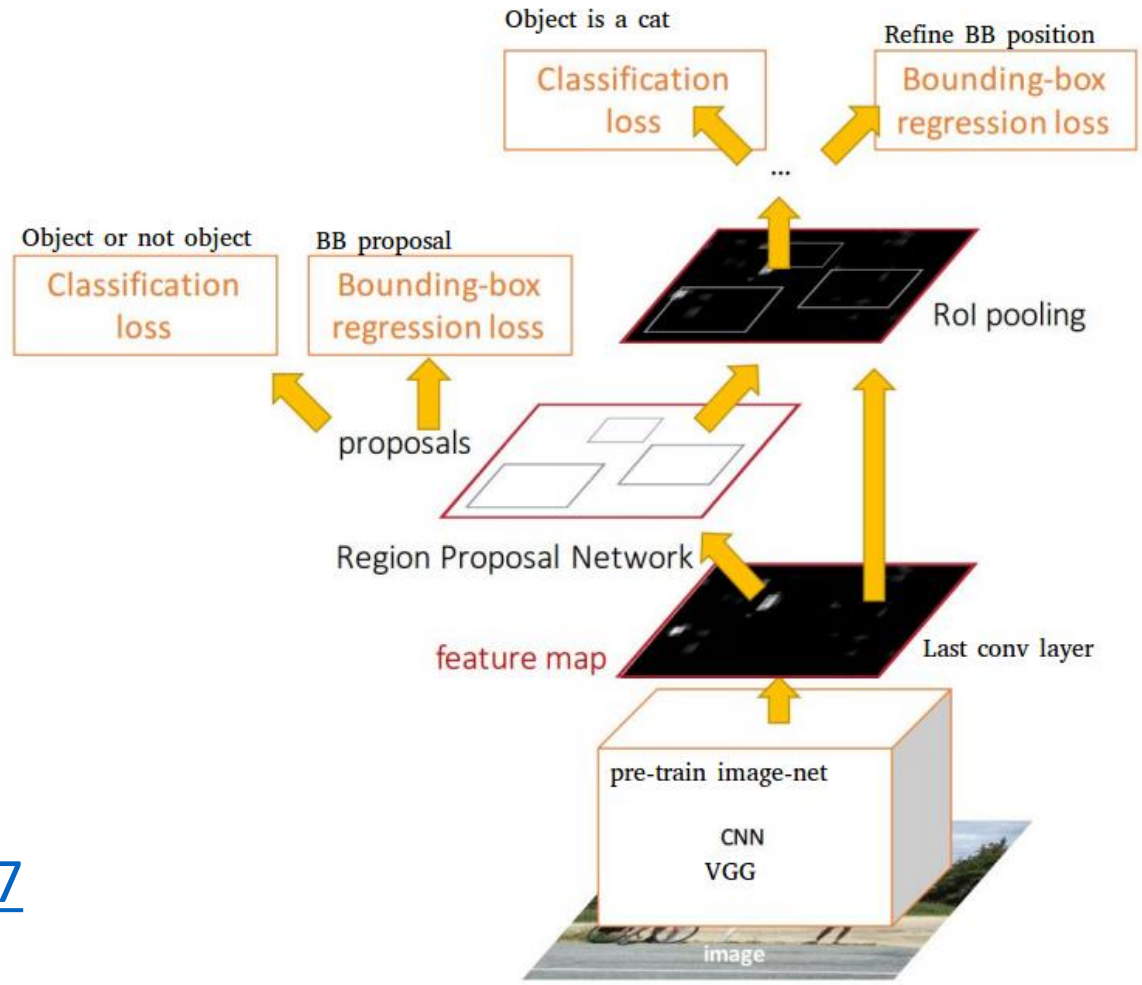
Idea: No need to recompute features for every box independently,
Regress refined bounding box coordinates.

https://arxiv.org/abs/1504.08083

Fast R-CNN. Girshick. ICCV 2015.

https://github.com/sunshineatnoon/Paper-Collection/blob/master/Fast-RCNN.md

Faster-RCNN



Idea: Integrate the Bounding Box Proposals as part of the CNN predictions

https://arxiv.org/abs/1506.01497

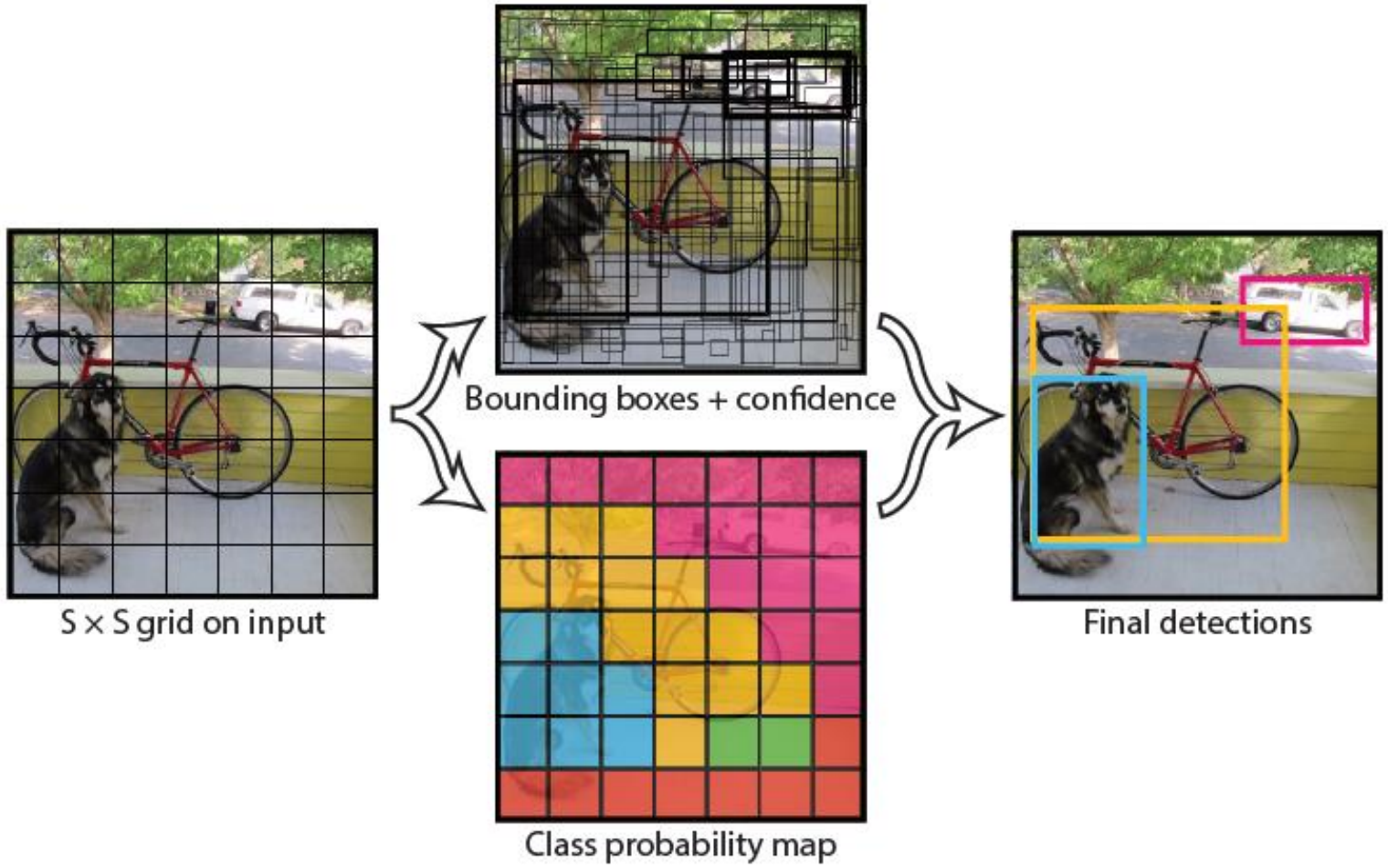
Ren et al. NIPS 2015.

Single-shot Object Detectors

- No two-steps of box proposals + Classification
- Anchor Points for predicting boxes

YOLO- You Only Look Once

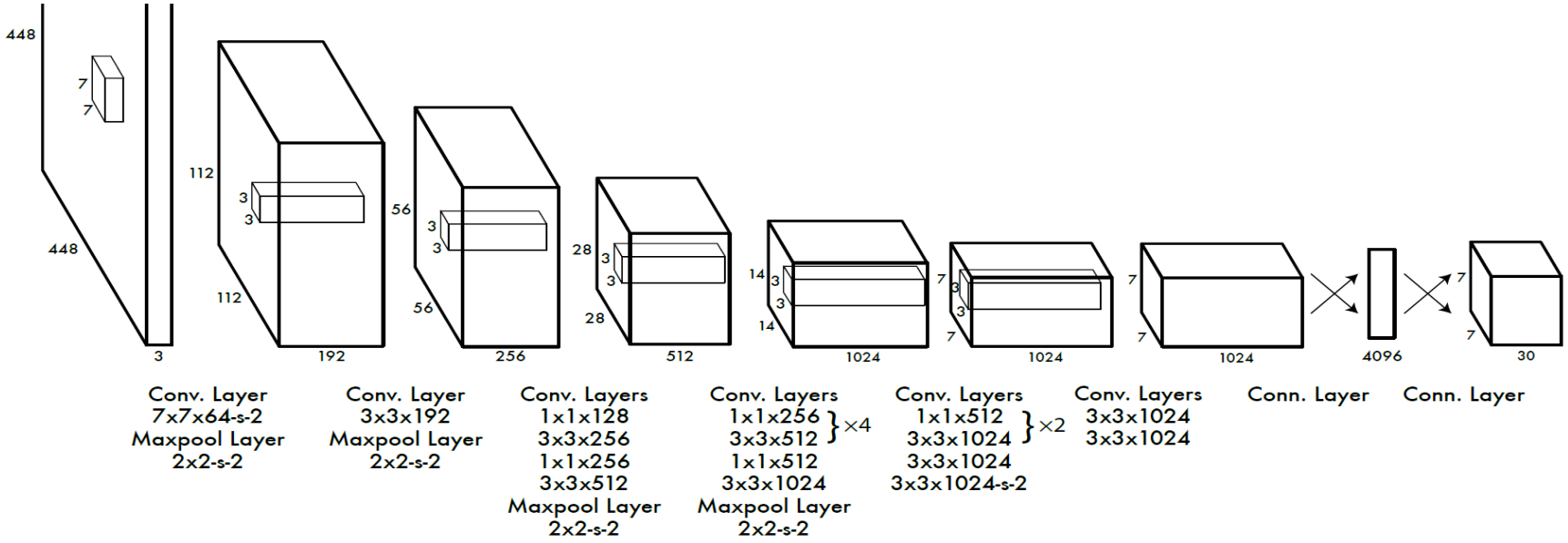
Idea: No bounding box proposals.
Predict a class and a box for every location in a grid.



https://arxiv.org/abs/1506.02640

Redmon et al. CVPR 2016.

YOLO- You Only Look Once



Divide the image into 7x7 cells.
Each cell trains a detector.
The detector needs to predict the object's class distributions.
The detector has 2 bounding-box predictors to predict bounding-boxes and confidence scores.

https://arxiv.org/abs/1506.02640

Redmon et al. CVPR 2016.

YOLO - Loss Function

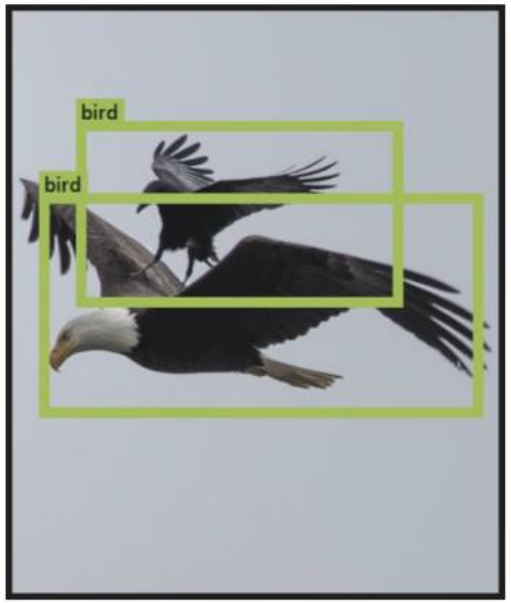
$$\lambda_{\textbf{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]$$

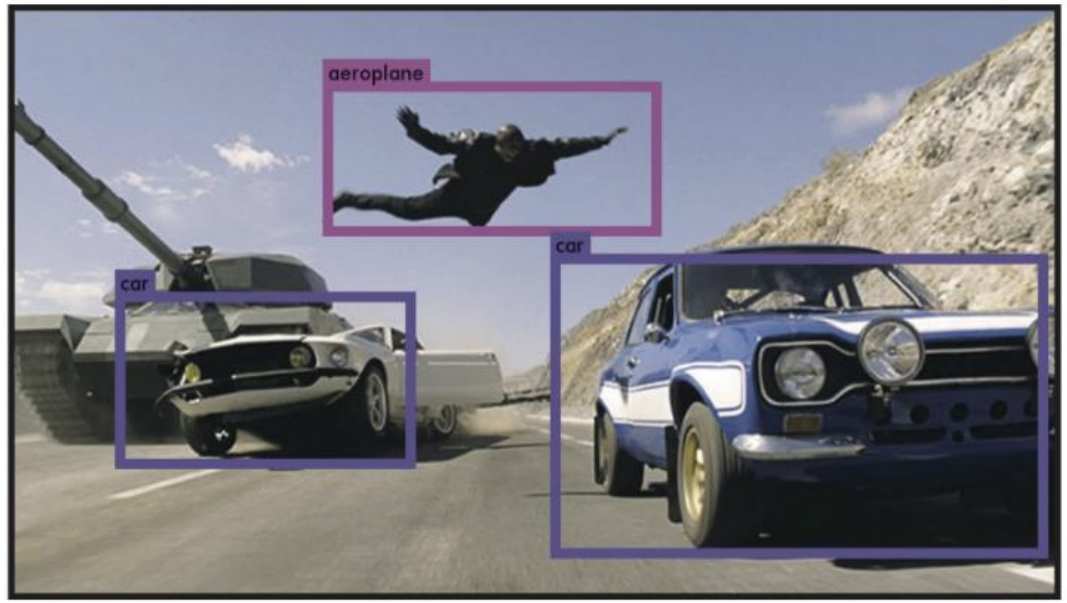
$$+ \lambda_{\textbf{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$$

$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2$$

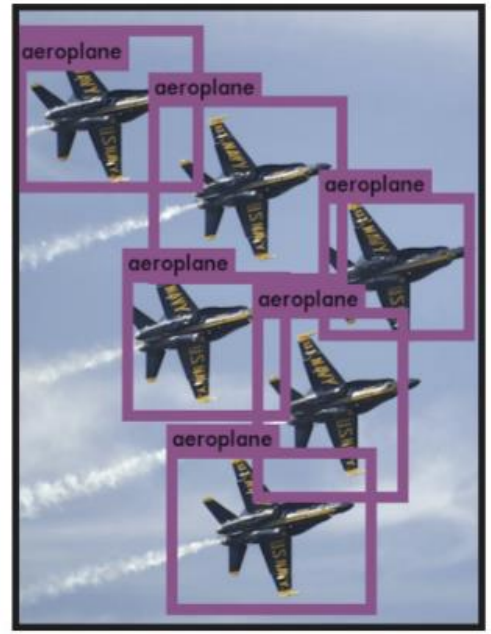
$$+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2$$

$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} \left(p_i(c) - \hat{p}_i(c) \right)^2$$

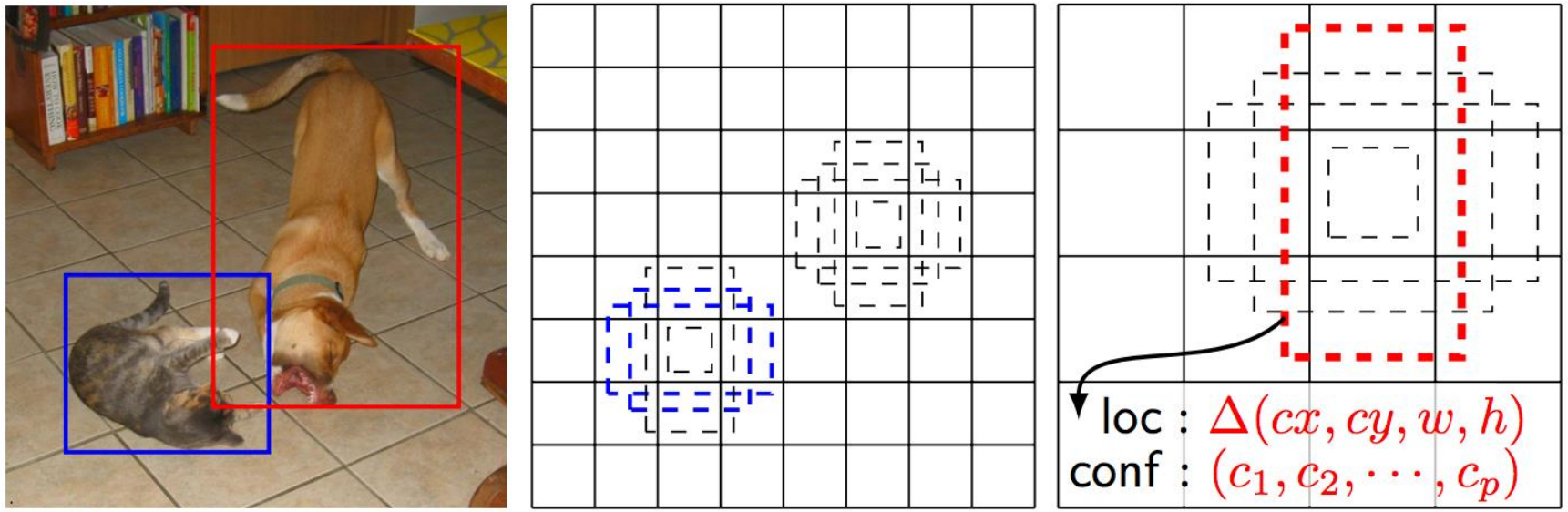

bird
bird


aeroplane
car
car


person
boat


aeroplane
aeroplane
aeroplane
aeroplane
aeroplane
aeroplane

SSD: Single Shot Detector

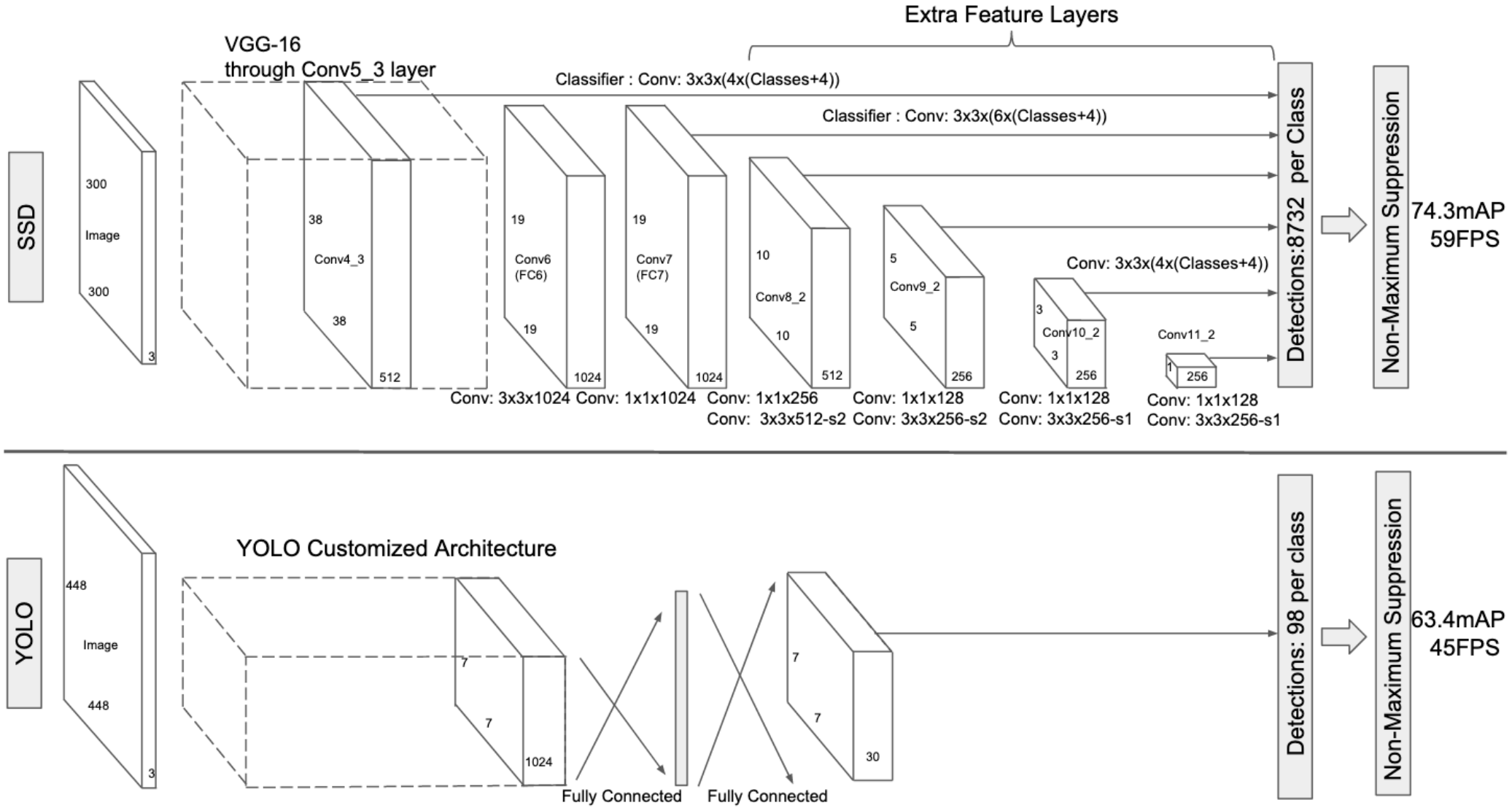


(a) Image with GT boxes (b) 8×8 feature map (c) 4×4 feature map

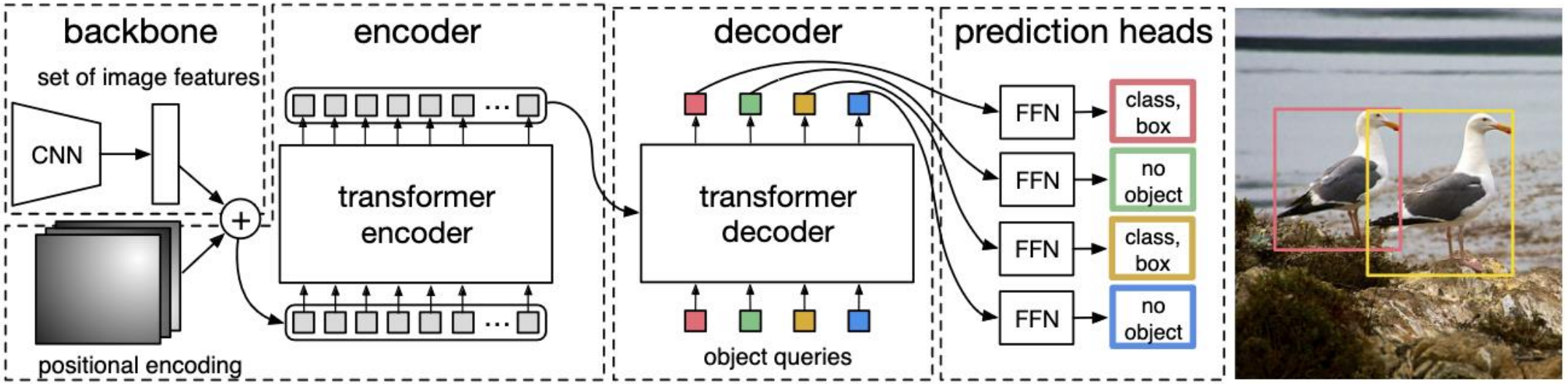
Idea: Similar to YOLO, but denser grid map, multiscale grid maps. + Data augmentation + Hard negative mining + Other design choices i n the network.

Liu et al. ECCV 2016.

SSD vs YOLO


Extra Feature Layers
VGG-16
through Conv5_3 layer
Classifier : Conv: 3x3x(4x(Classes+4))
Classifier : Conv: 3x3x(6x(Classes+4))
SSD
300
Image
300
3
38
Conv4_3
38
512
19
Conv6
(FC6)
19
1024
19
Conv7
(FC7)
19
1024
10
Conv8_2
10
512
5
Conv9_2
5
256
Conv: 3x3x(4x(Classes+4))
3
Conv10_2
3
256
Conv11_2
1
256
Detections:8732 per Class
Non-Maximum Suppression
74.3mAP
59FPS
Conv: 3x3x1024
Conv: 1x1x1024
Conv: 1x1x256
Conv: 3x3x512-s2
Conv: 1x1x128
Conv: 3x3x256-s2
Conv: 1x1x128
Conv: 3x3x256-s1
Conv: 1x1x128
Conv: 3x3x256-s1
YOLO
448
Image
448
3
YOLO Customized Architecture
7
7
1024
Fully Connected
Fully Connected
7
7
30
Detections: 98 per class
Non-Maximum Suppression
63.4mAP
45FPS

Object Detection with Transfromers (DETR) (2020)



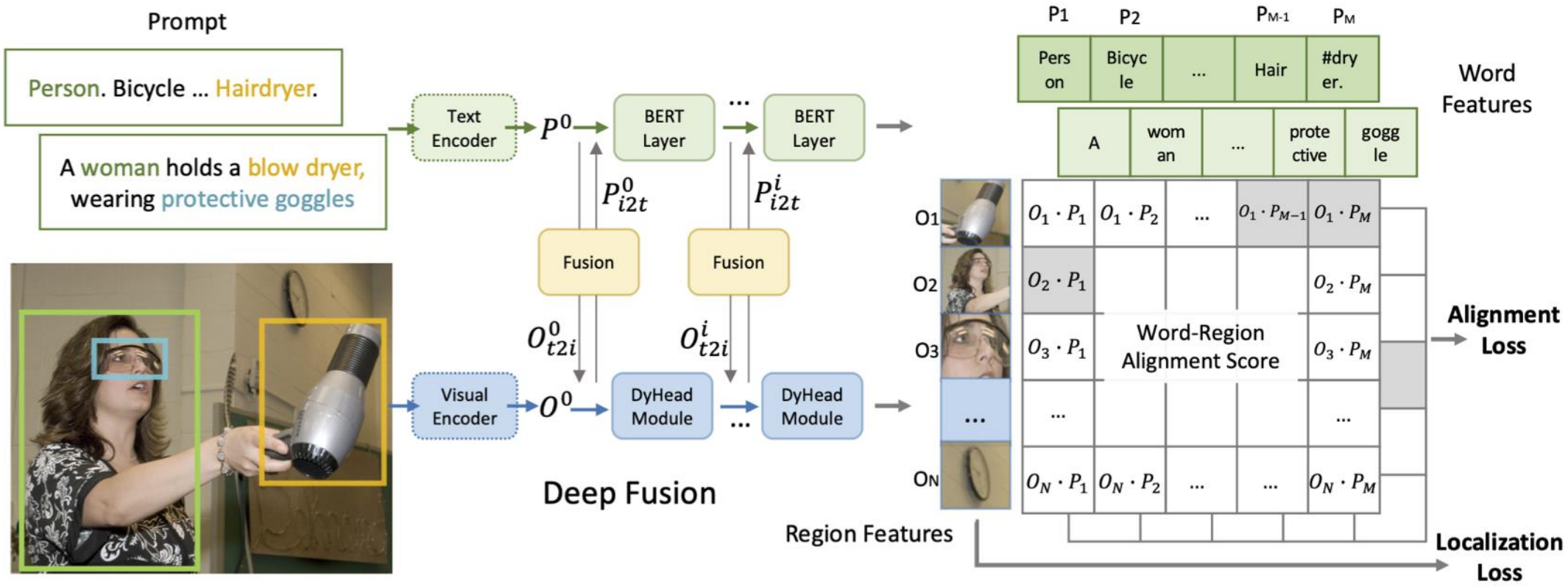
$$\mathcal{L}_{\text{Hungarian}}(y, \hat{y}) = \sum_{i=1}^{N} \left[-\log \hat{p}_{\hat{\sigma}(i)}(c_i) + \mathbb{1}_{\{c_i \neq \varnothing\}} \mathcal{L}_{\text{box}}(b_i, \hat{b}_{\hat{\sigma}}(i)) \right]$$

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

https://arxiv.org/abs/2005.12872

https://github.com/facebookresearch/detr

GLIP: CLIP but also outputs boxes (2021)



https://arxiv.org/abs/2112.03857

https://github.com/microsoft/GLIP

More recent Object Detectors

- OwL-ViT (2022): https://arxiv.org/abs/2205.06230
- GLIP-v2 (2022): https://arxiv.org/abs/2206.05836
- DetCLIP (2022): https://arxiv.org/abs/2209.09407
- YOLO-World (2024): https://github.com/AILab-CVC/YOLO-World

Questions