Deep Learning

A course about theory & practice

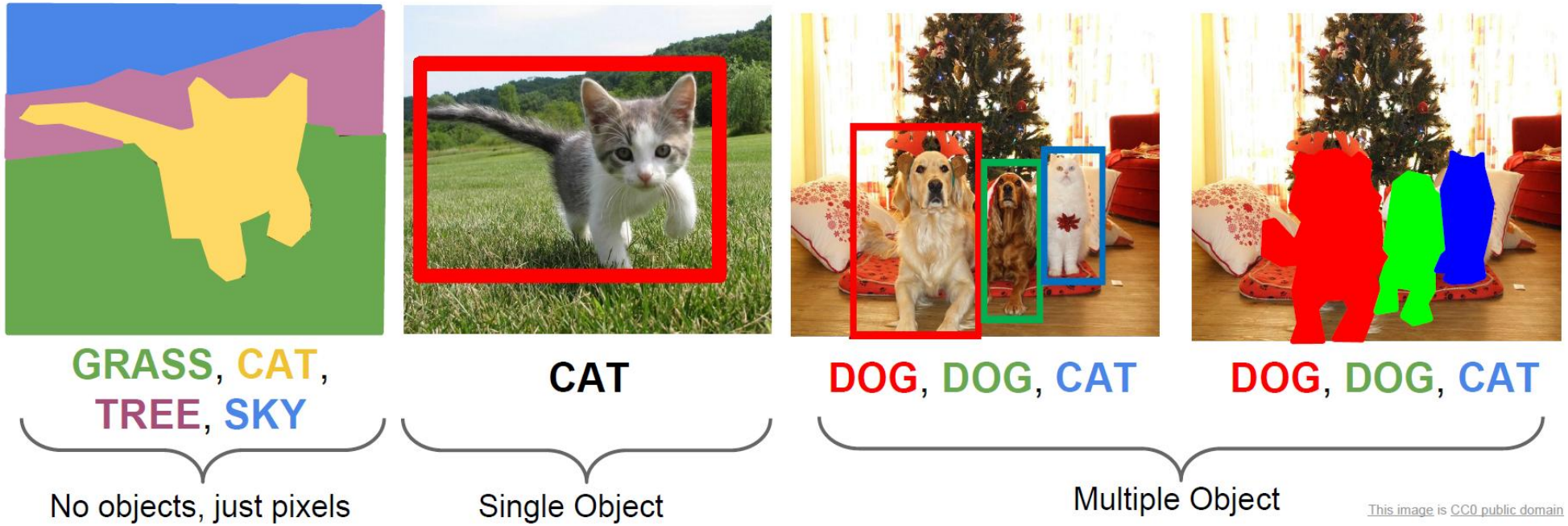
UNIVERSITÀ DI PAVIA

Image Segmentation and Object Detection

Marco Piastra

Deep Learning for different imaging tasks

Beyond simple image classification



[images from http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture11.pdf]

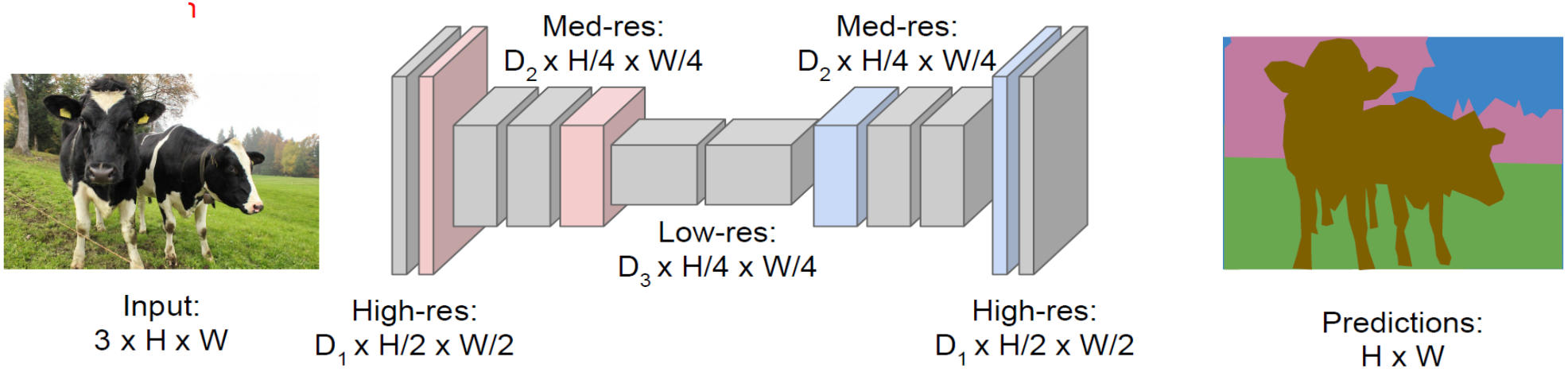
Semantic segmentation

Beyond simple image classification

- ***Similar network architecture, different arrangement***

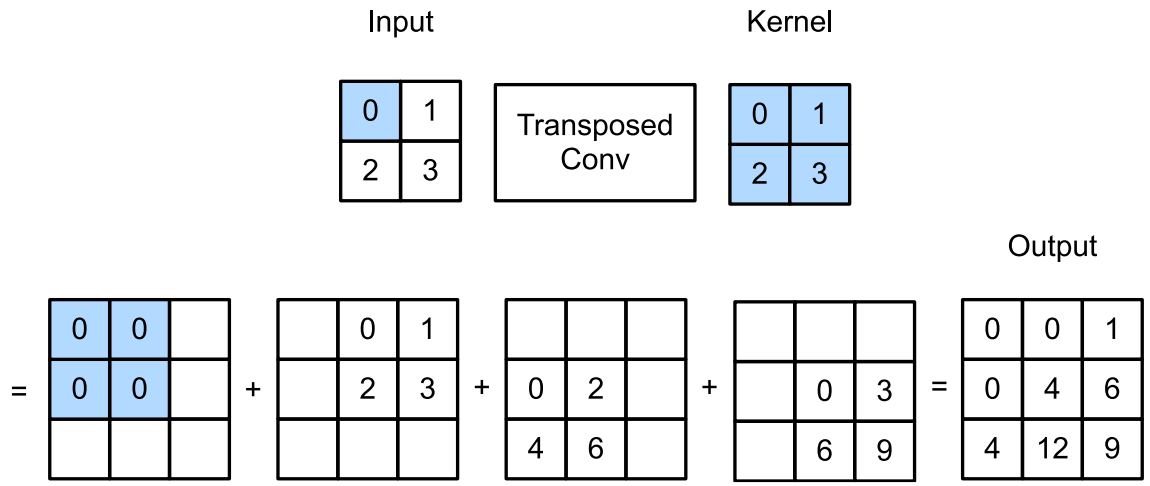
  Fully Convolutional Networks (FCN)

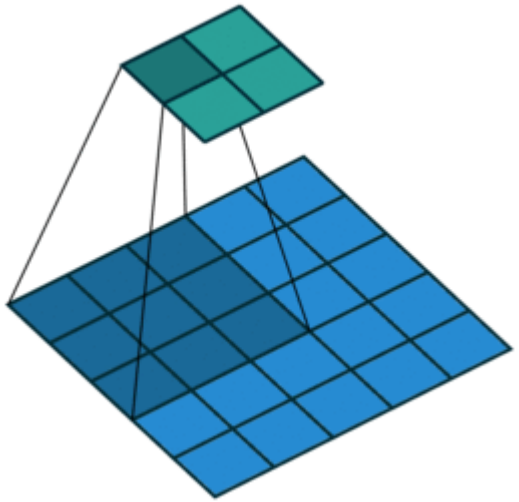
  Downsampling first, *upsampling* afterwards



[images from http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture11.pdf]

Transposed Convolution (a.k.a. 'Deconvolution')





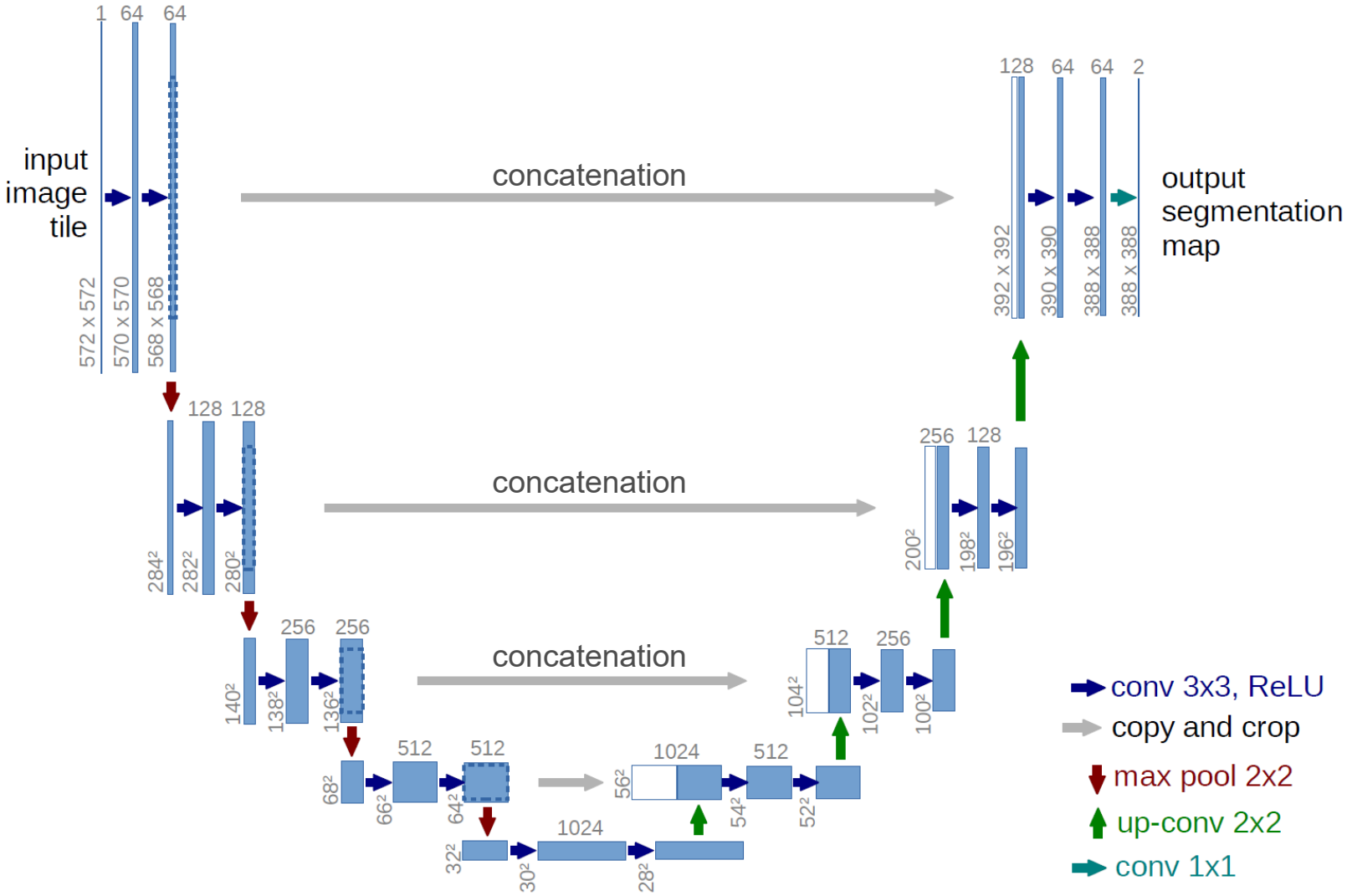
- **Convolutional layers working 'in reverse'**
  - Scalar input values are multiplied by the kernel tensor
  - The output feature map is obtained by summing up all contributions

[images from https://d2l.ai/chapter_computer-vision/transposed-conv.html]

Semantic segmentation

- ***U-Net* [2015]**

  Great precision
  Fast to train

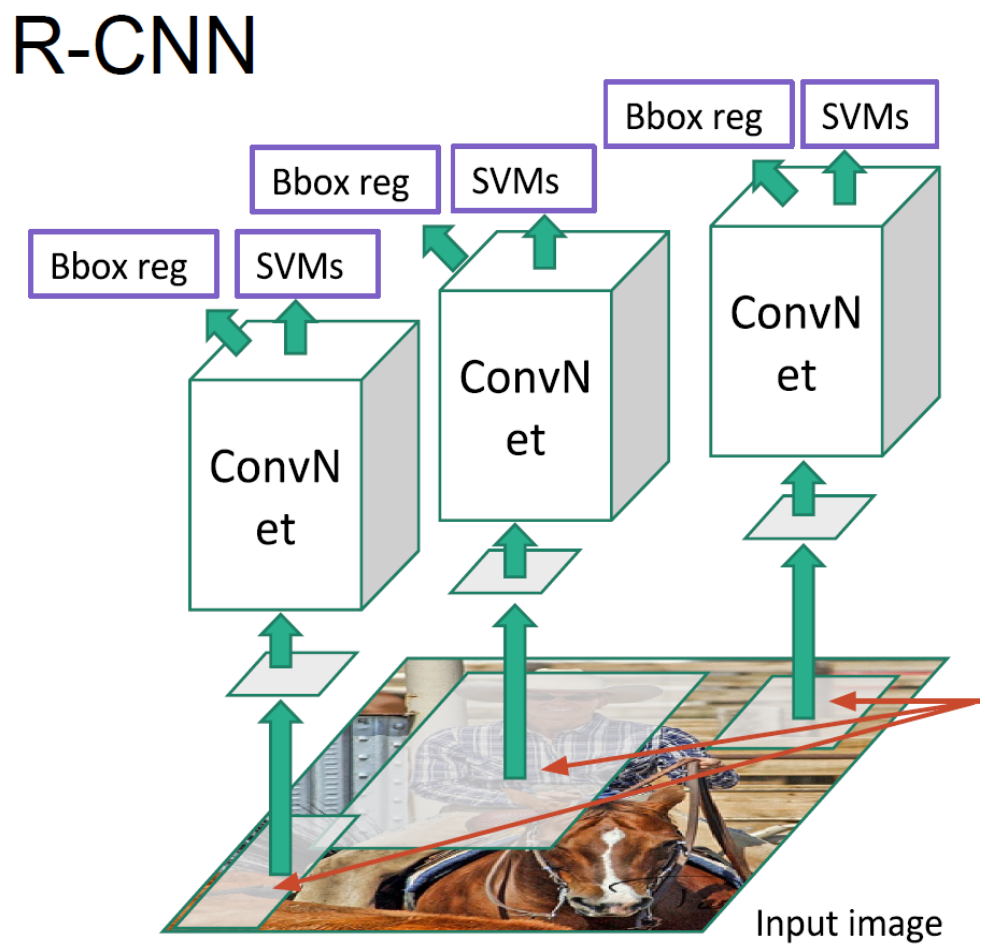


[image from https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/]

Object detection and positioning

Generate boxes and classifications

- ***Two-stage Process***
  - *Generate bounding box candidates*
  - *Pass each candidate through a DCNN*
  - *Select those candidates that are classified with higher certainty*



[images from http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture11.pdf]

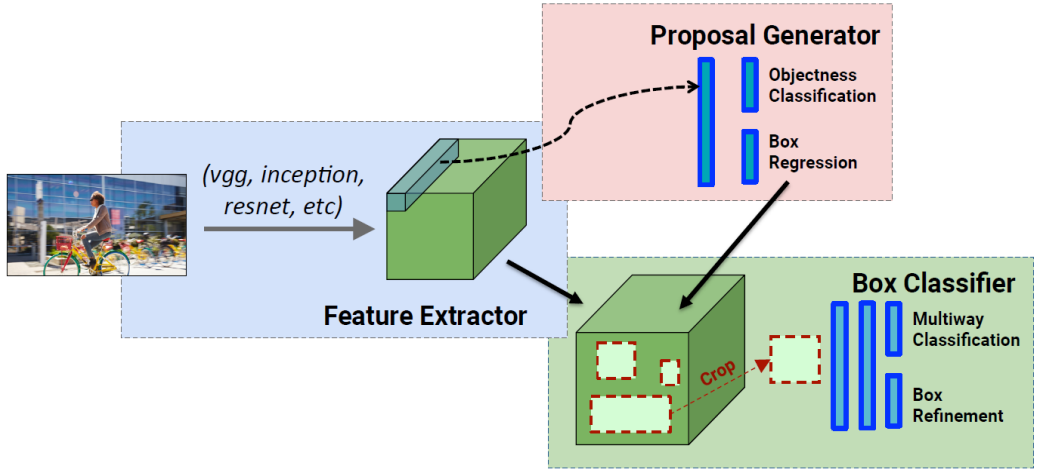
Object detection and positioning

Generate boxes and classifications

- ***Two-stage to One-stage process***

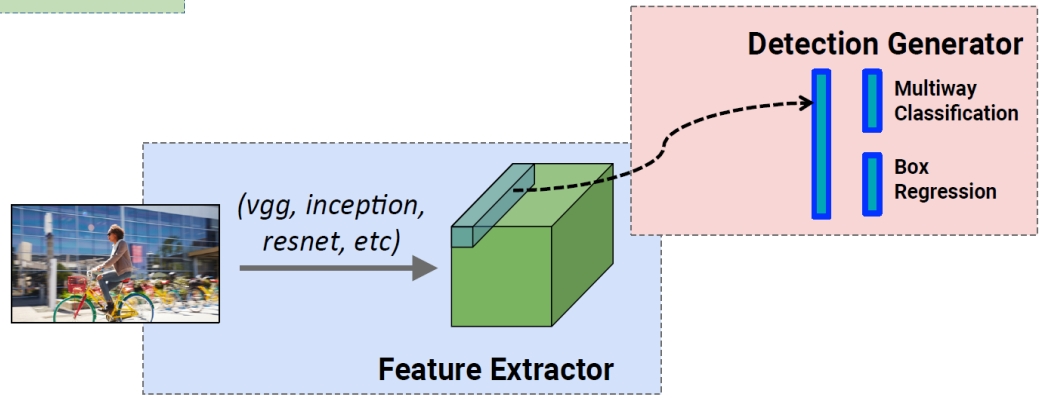
  Generate bounding box candidates and classifications in one go

[images from https://arxiv.org/pdf/1611.10012.pdf



Two stages

One stage



Object detection and positioning

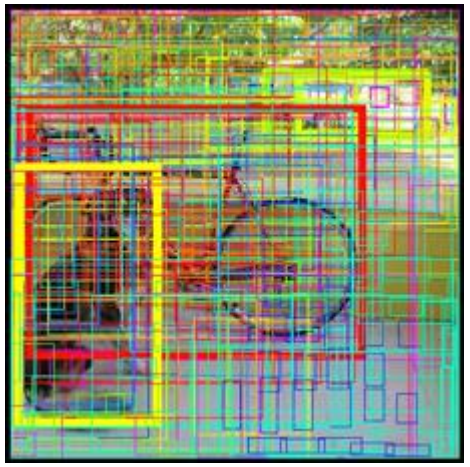
- ***YOLO and SSD: one-pass convolutional network for object detection***

  Generate boxes and classifications at once

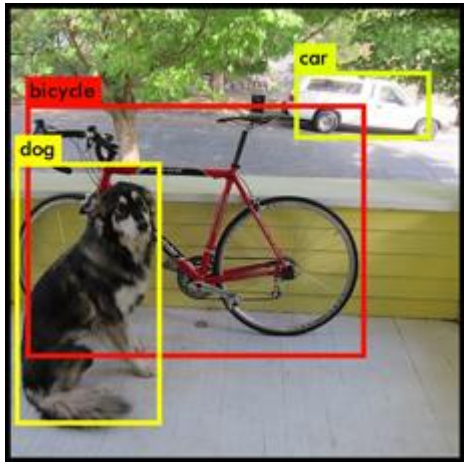
1) Impose a fixed grid over the input image

2) Generate possible bounding boxes

3) Classify each of them



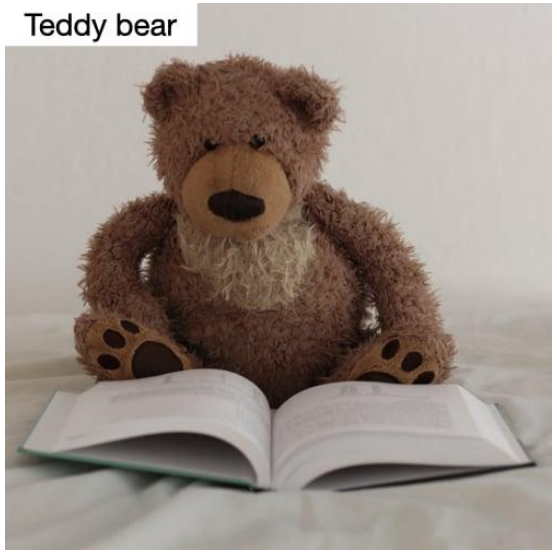
4) Keep the boxes at highest confidence

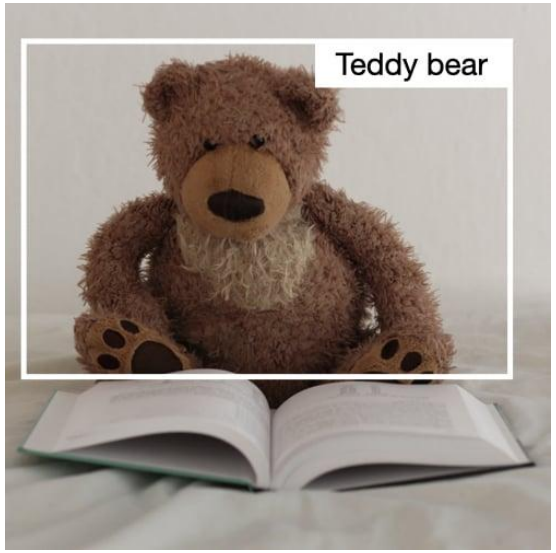


[images from http://machinethink.net/blog/object-detection-with-yolo/]

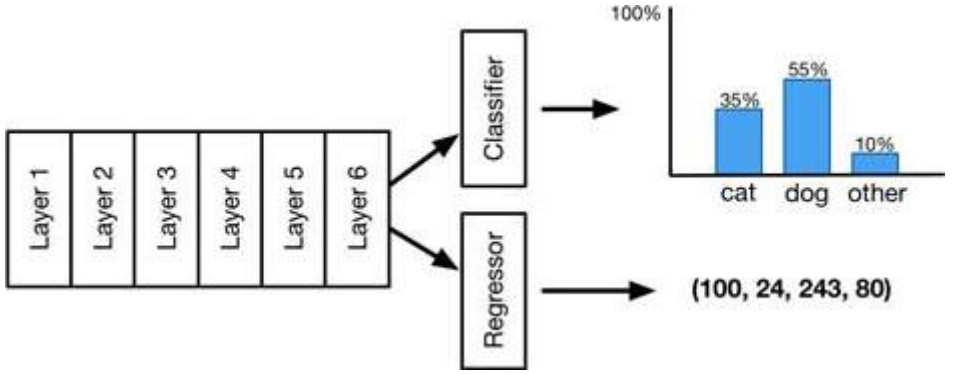
Object detection and positioning

- **From classification to localization**





[images from https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks]



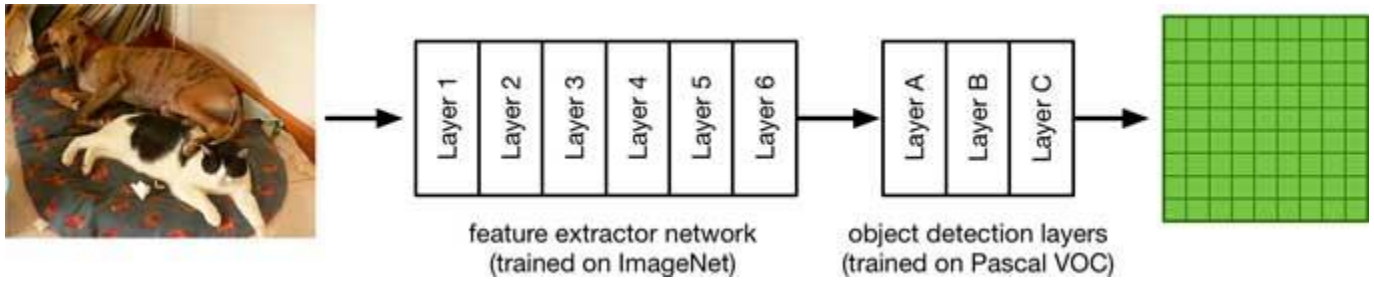
Loss: cross-entropy

Loss: mean square error

Object detection and positioning

YOLO v1 to YOLO v7

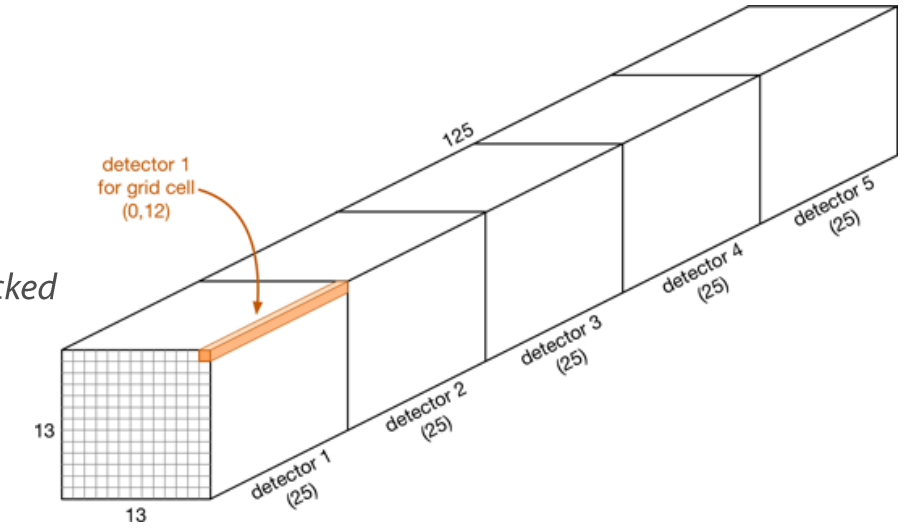
- **Grid detectors**



Classification (transfer learning)

Multi-class detectors (arranged in a grid)

(Detector blocks are shown as stacked but they work in parallel)



1) Impose a fixed grid over the input image (13 x 13 in the example)

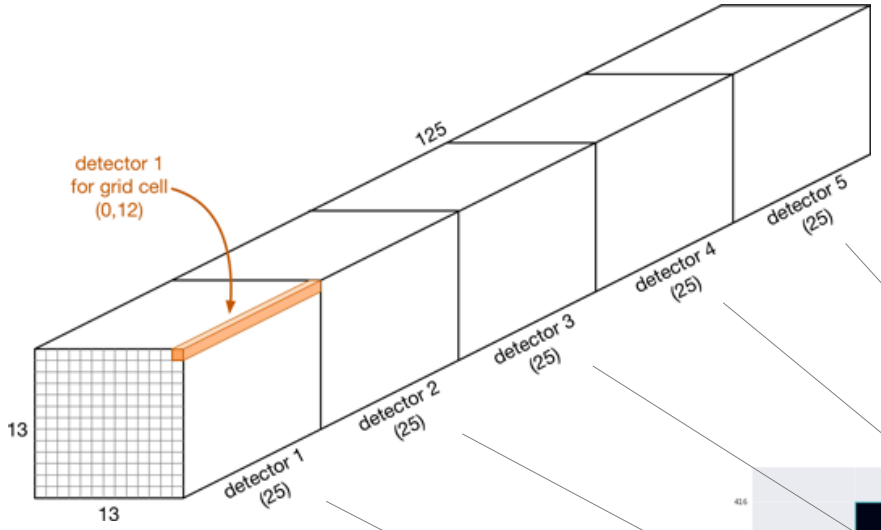
2) Assign 5 multiclass detectors to each cell of the grid

3) Each multiclass detector works on a specific *anchor* shape (*see next slide*)

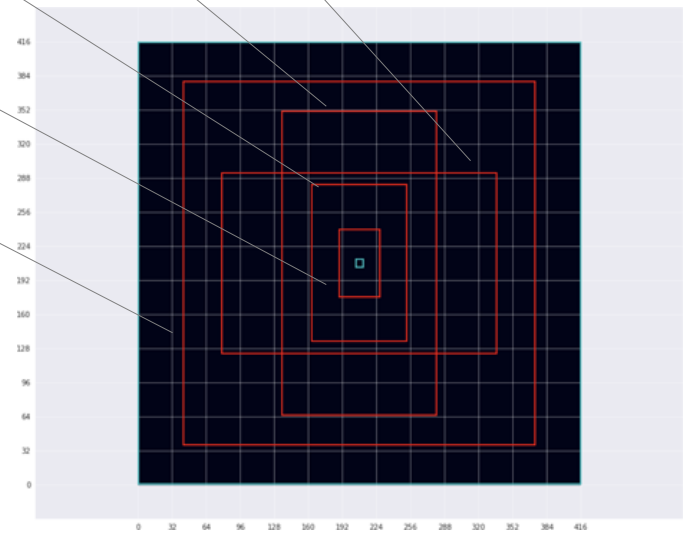
[images from https://machinethink.net/blog/object-detection/]

Object detection and positioning

- ***Grid detectors: one per <u>anchor</u>***



Anchor shapes and sizes
are pre-computed from the training set
(bounding box statistics)



Each detector produces:
- *class probabilities*
- *confidence score*

per its assigned anchor

[images from https://machinethink.net/blog/object-detection/]

Object detection and positioning

- **Given anchor, cell and class**

$$\langle c_x, c_y, p_w, p_h \rangle$$

top-left cell coordinates

anchor sizes

- **Each detector produces**

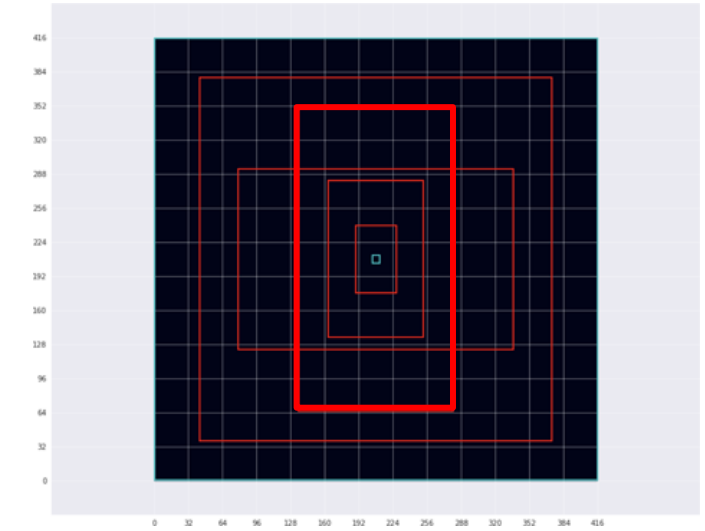
$$\langle t_x, t_y, t_w, t_h, p_o, p_c \rangle$$

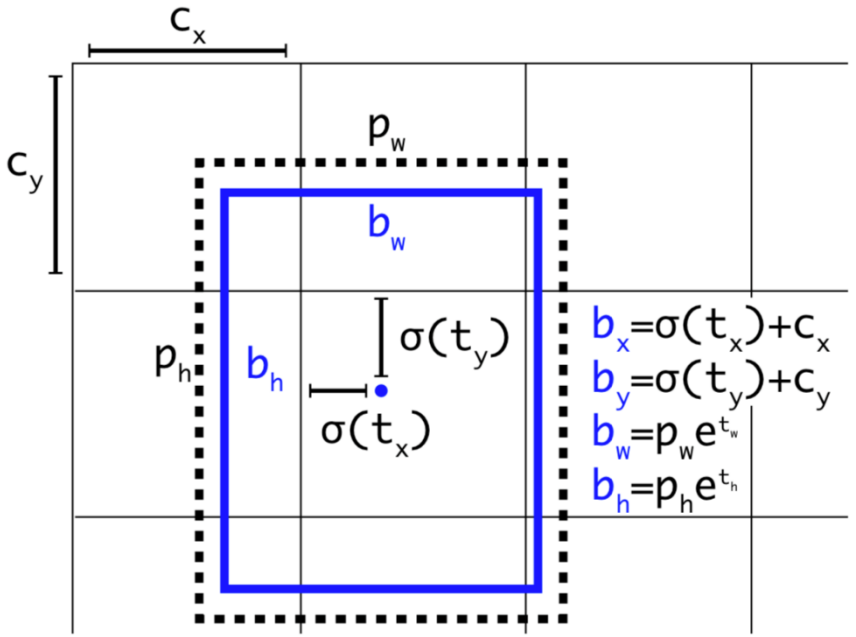
$$\begin{aligned} b_x &= \sigma(t_x) + c_x \\ b_y &= \sigma(t_y) + c_y \\ b_w &= p_w e^{t_w} \\ b_h &= p_h e^{t_h} \end{aligned}$$

bounding box coordinates

p_o 'objectness' probability

p_c class probability



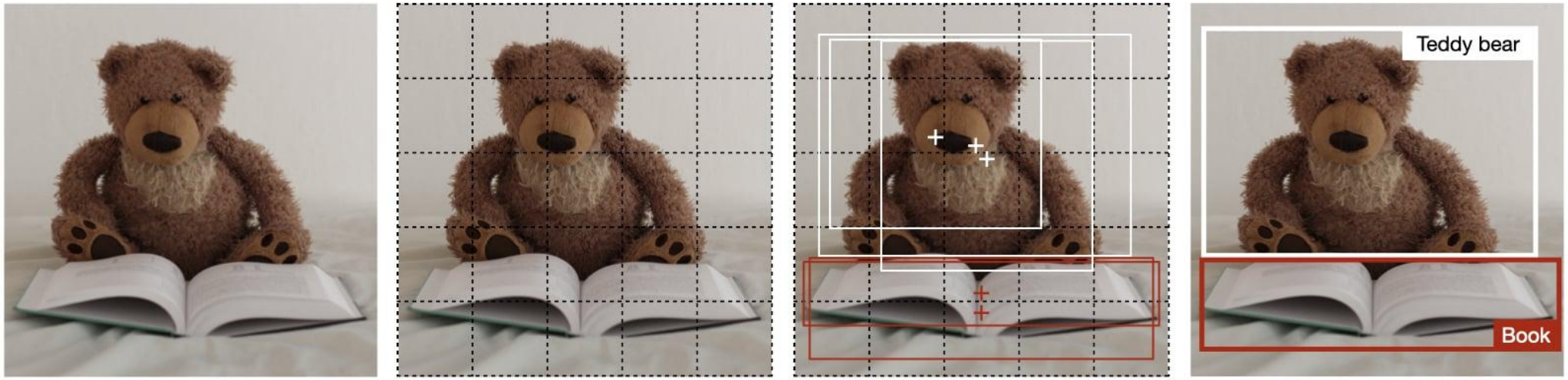


[images from https://wikidocs.net/167697]

Object detection and positioning

- ***From grid boxes to candidate boxes***

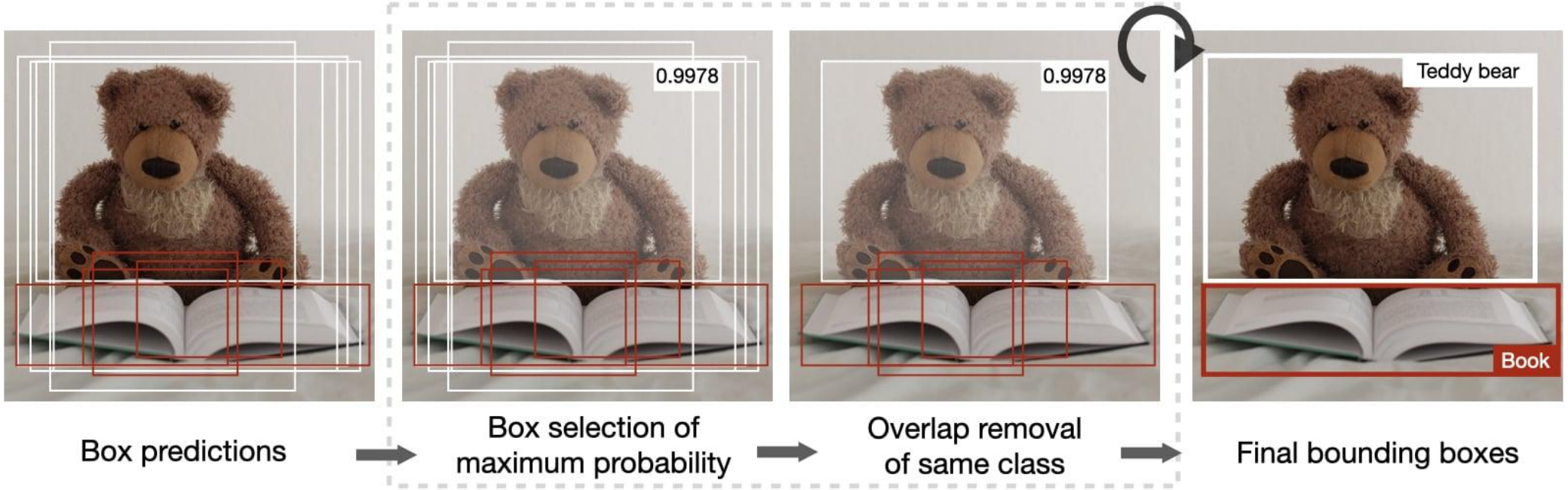
  Merging predictions



[images from https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks]

Object detection and positioning

- ***Further processing***



[images from https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks]

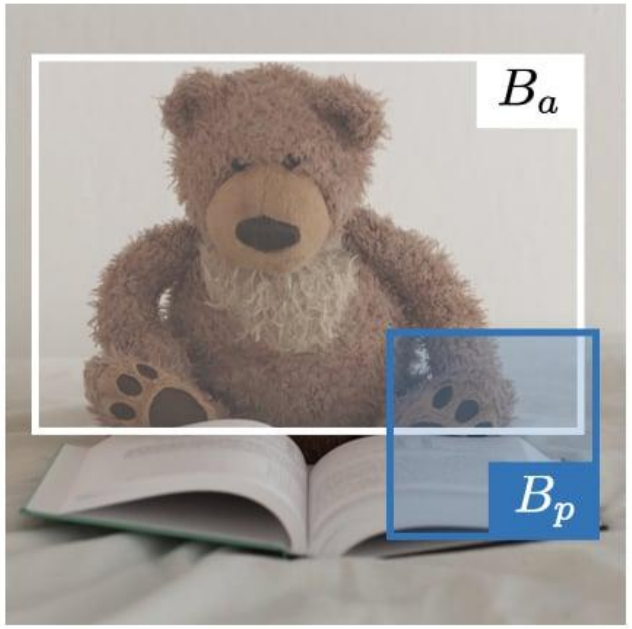
Object detection and positioning

- ***Measuring object detection accuracy***

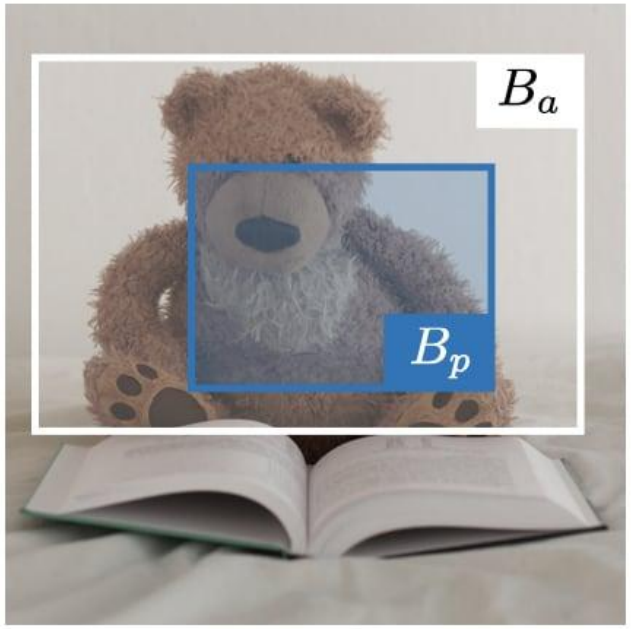
  Intersection over Union (IoU)

$$\mathrm{IoU}(B_p, B_a) := \frac{B_p \cap B_a}{B_p \cup B_a}$$

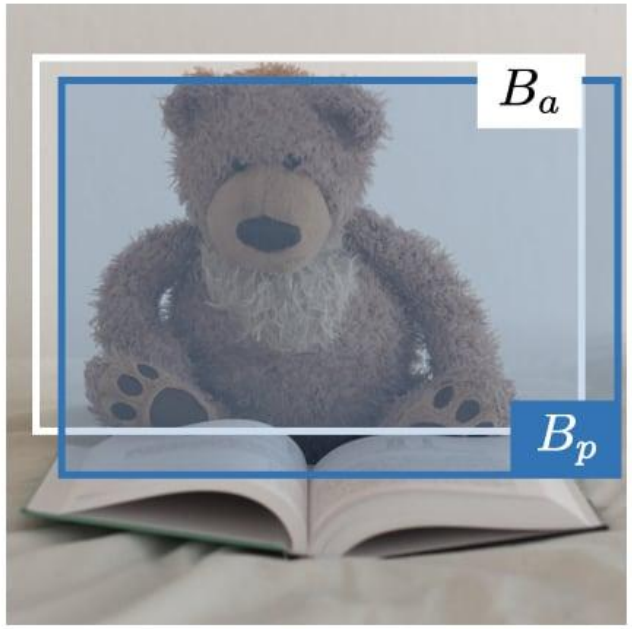
It's a post-localization accuracy measure (not a loss function)



$\mathrm{IoU}(B_p, B_a) = 0.1$



$\mathrm{IoU}(B_p, B_a) = 0.5$



$\mathrm{IoU}(B_p, B_a) = 0.9$

[images from https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks]