Convolution Neural Network 2

Dr. Kelly Trinh

Housekeeping

- Data for Assessment 3
  - Data must NOT be the one already covered in Practical sessions and AWS Jupyter Lab (such as MNIST, CIFAR, dogs and cats).

Today

- Image Localisation
- Image Detection

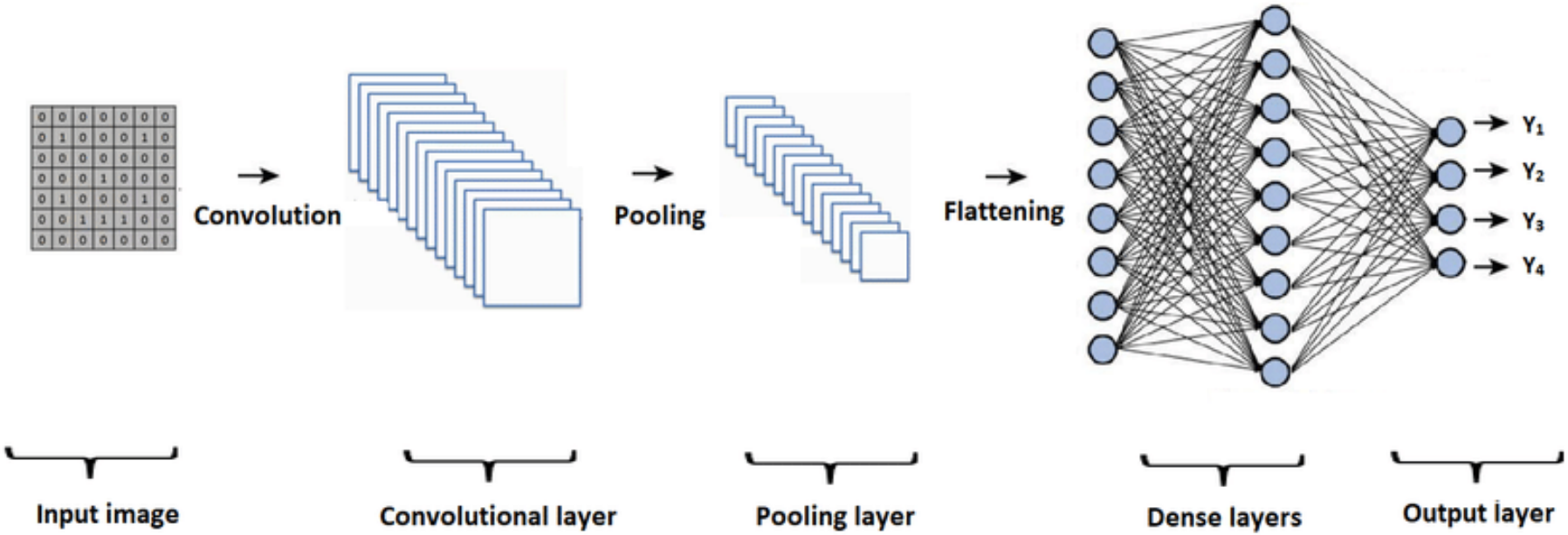
Image Classification

Image Classification (Convnet)

Input: RGB image

Output: Object class (a dog)

Object classification-CNN



In this example: output $\in \{dog, cat\}$

Object localisation

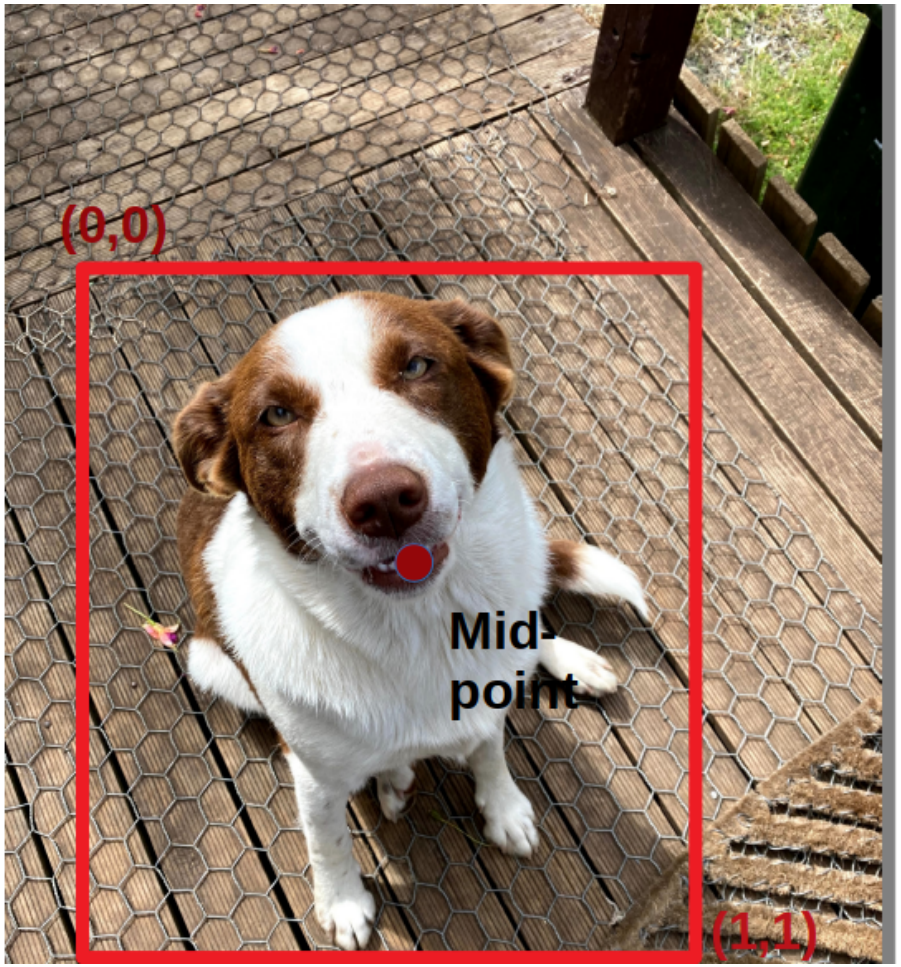
- Object localisation : identifying object and location with bounding box.

Image Localisation (Convnet)

Input: RGB image

Output:
- **P**robability if there is an object in the image
- bounding box (x,y, width, height)
- object class (a dog)

E.g:
X= 0.5
Y= 0.45
W= 0.3
H= 0.7



Object localisation

Image classification

$$o = \begin{bmatrix} p(c_1) \\ p(c_2) \end{bmatrix}$$

An example of loss function

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N} c_i log(p(c_i)) + (1 - c_i) log(1 - p(c_i))$$

Image localisation

$$o = \begin{bmatrix} p_c \\ x \\ y \\ w \\ h \\ c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} \text{is the object in an image?} \\ \text{x coordinate relative to an image, } x \in [0,1] \\ \text{y coordinate relative to an image , } y \in [0,1] \\ \text{width of the image relative to an image , } w > 0 \\ \text{height of the image relative to an image , } h > 0 \\ \text{confidence level that image belongs to class 1 or not} \\ \text{confidence level image belongs to class 2 or not} \end{bmatrix}$$

An example of loss function

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

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

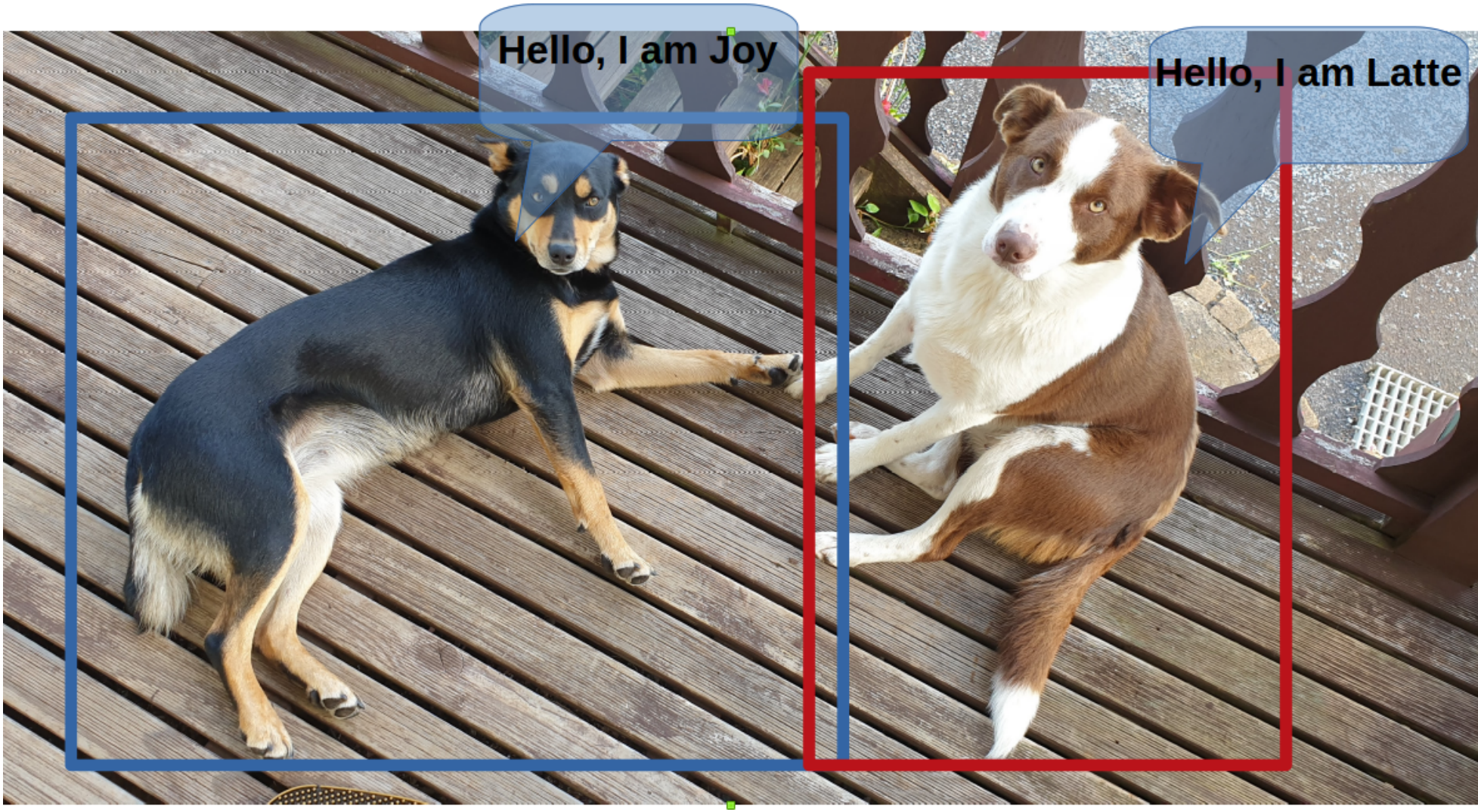
$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2$$

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

$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2 \quad (3)$$

Object detection

- Object detection : identifying objects and locations of each object with bounding boxes.



Object detection-Challenges

- Multiple outputs: multiple objects per image

Object detection-Challenges

- Multiple outputs: multiple objects per image
- Multiple types of outputs: predicting objects (category labelling) and location (bounding boxes)

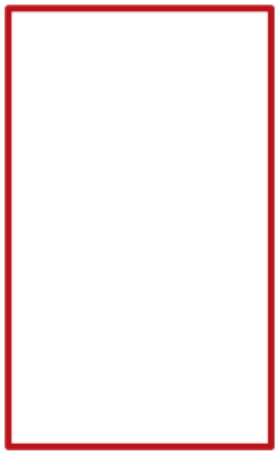
Object detection-Challenges

- Multiple outputs: multiple objects per image
- Multiple types of outputs: predicting objects (category labelling) and location (bounding boxes)
- Large images: classification generally work with 224×224, need higher resolution for detection 800×600. Therefore, always beware of computation cost and memory when doing image detection.

Object detection-Challenges

- Multiple outputs: multiple objects per image
- Multiple types of outputs: predicting objects (category labelling) and location (bounding boxes)
- Large images: classification generally work with 224×224, need higher resolution for detection 800×600. Therefore, always beware of computation cost and memory when doing image detection.

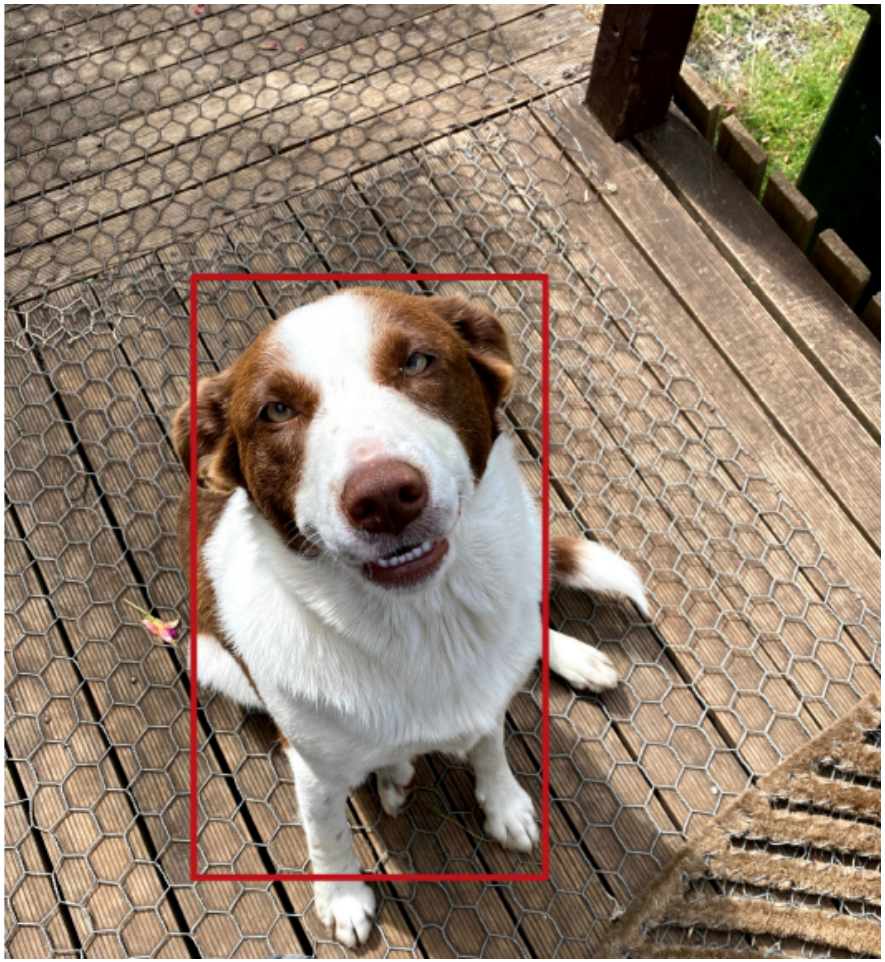
Naive approach: sliding window detection



Fixed window size

Apply a CNN to many crops of image. and CNN will classify each window as an object or a background

Naive approach: sliding window detection



How many possible boxes are there?

- An image of size $H \times W$, a box of size $h \times w$:
- Possible x positions: $W-w+1$
- Possible y positions: $H-h+1$
- Possible positions: $(W-w+1) \times (H-h+1)$
- Total possible boxes:

$$\sum_{w=1}^{W} \sum_{h=1}^{H} (W-w+1) \times (H-h+1)$$
$$= \frac{H(H+1)W(W+1)}{4}$$

- An image 600x600 will have approximately 32 million possible bounding boxes!!!

Image detection approaches

- Convolution Implementation of Sliding windows
- Regional Based CNN (R-CNN) ([Girshick et al., 2014])
- Fast regional CNN (R-CNN)([Girshick, 2015], [Ren et al., 2015])
- Mask R-CNN ([He et al., 2017])
- Single shot detector [Liu et al., 2015]
- YOLO (YOLOv1, YOLOv2, YOLOv3, YOLOv4) ([Redmon et al., 2015], [Redmon and Farhadi, 2018])

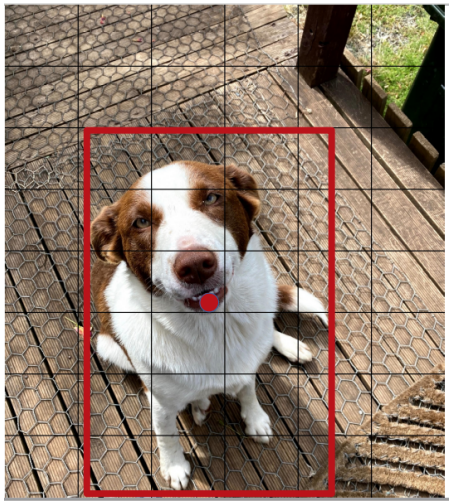
Image Detection

- Bounding boxes
- Intersection over Union
- Non-Max Suppression
- Mean Average Precision

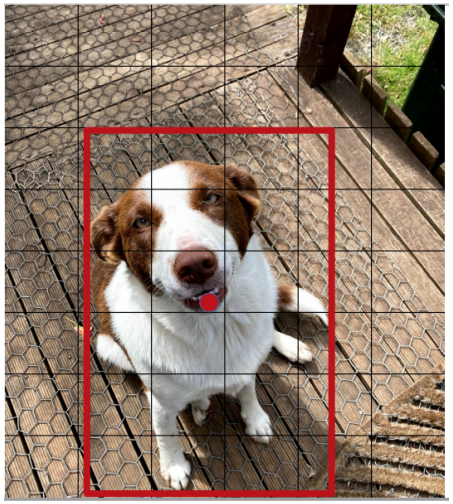
How to select a better bounding box?

YOLO algorithm-Bounding Box

- Divide an image into $s \times s$ grids (e.g. 6×6)

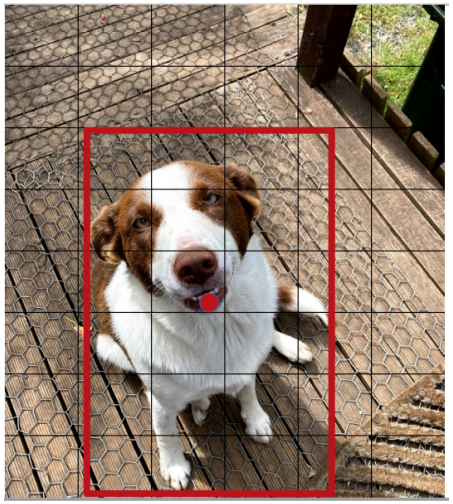


YOLO algorithm-Bounding Box



- Divide an image into $s \times s$ grids (e.g. 6×6)
- The cell contains the mid-point of an object is responsible for detecting the object

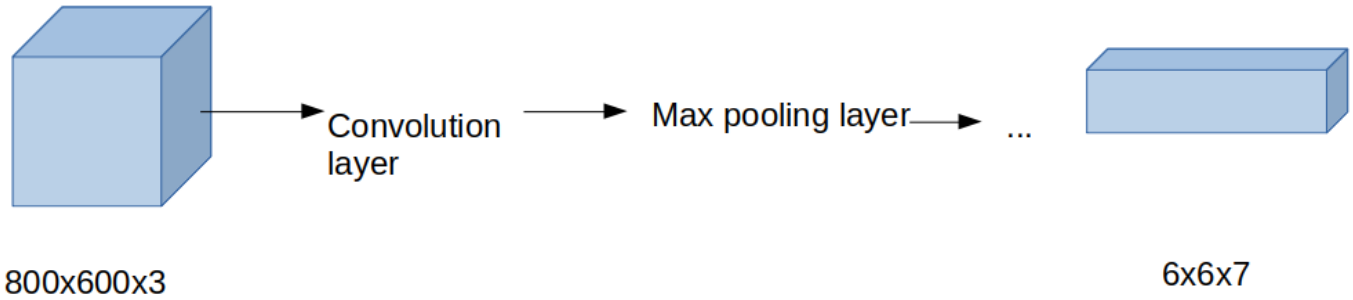
YOLO algorithm-Bounding Box

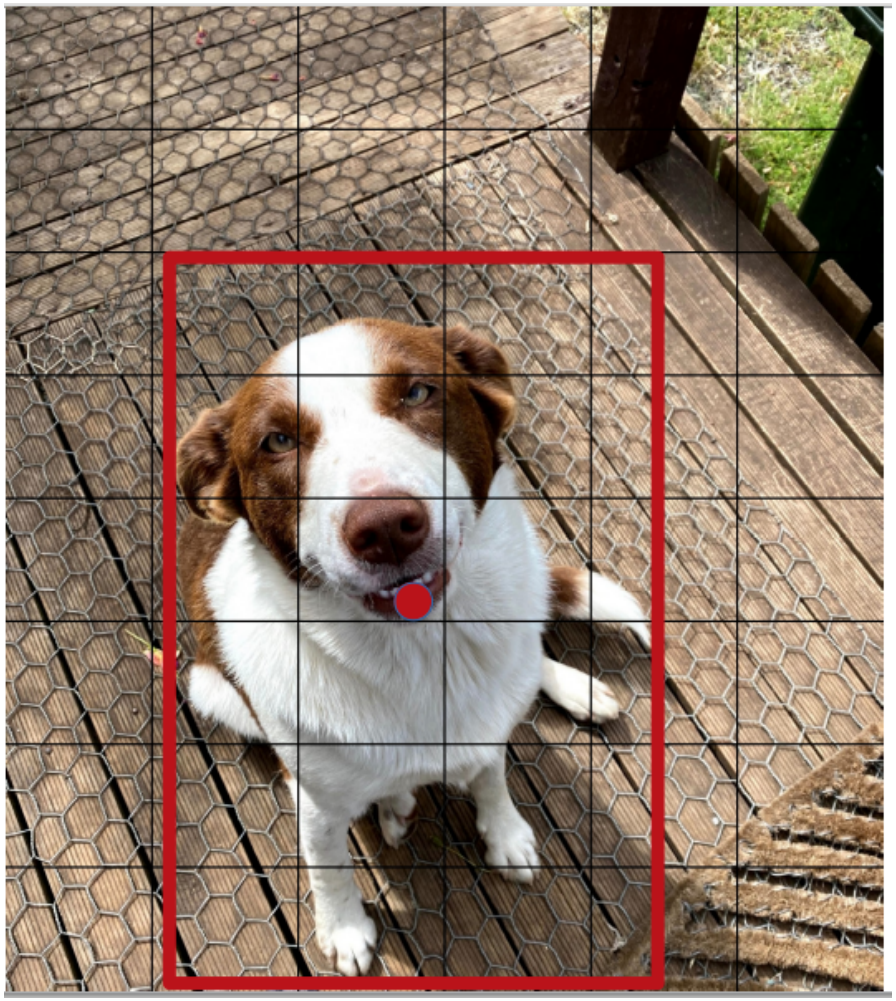


- Divide an image into $s \times s$ grids (e.g. 6×6)
- The cell contains the mid-point of an object is responsible for detecting the object
- Each grid cell provide

$$o = \begin{bmatrix} p_o \\ x \\ y \\ w \\ h \\ p_{c_1} \\ p_{c_2} \end{bmatrix} = \begin{bmatrix} \text{is the object in the grid?} \\ \text{x coordinate relative to the grid cell, } x \in [0,1] \\ \text{y coordinate relative to the grid cell, } y \in [0,1] \\ \text{width of the image relative to the grid cell, } w > 0 \\ \text{height of the image relative to the grid cell, } h > 0 \\ \text{confidence level that image belongs to class 1 or not} \\ \text{confidence image belongs to class 2 or not} \end{bmatrix}$$

- Total output dimension is $6 \times 6 \times 7$

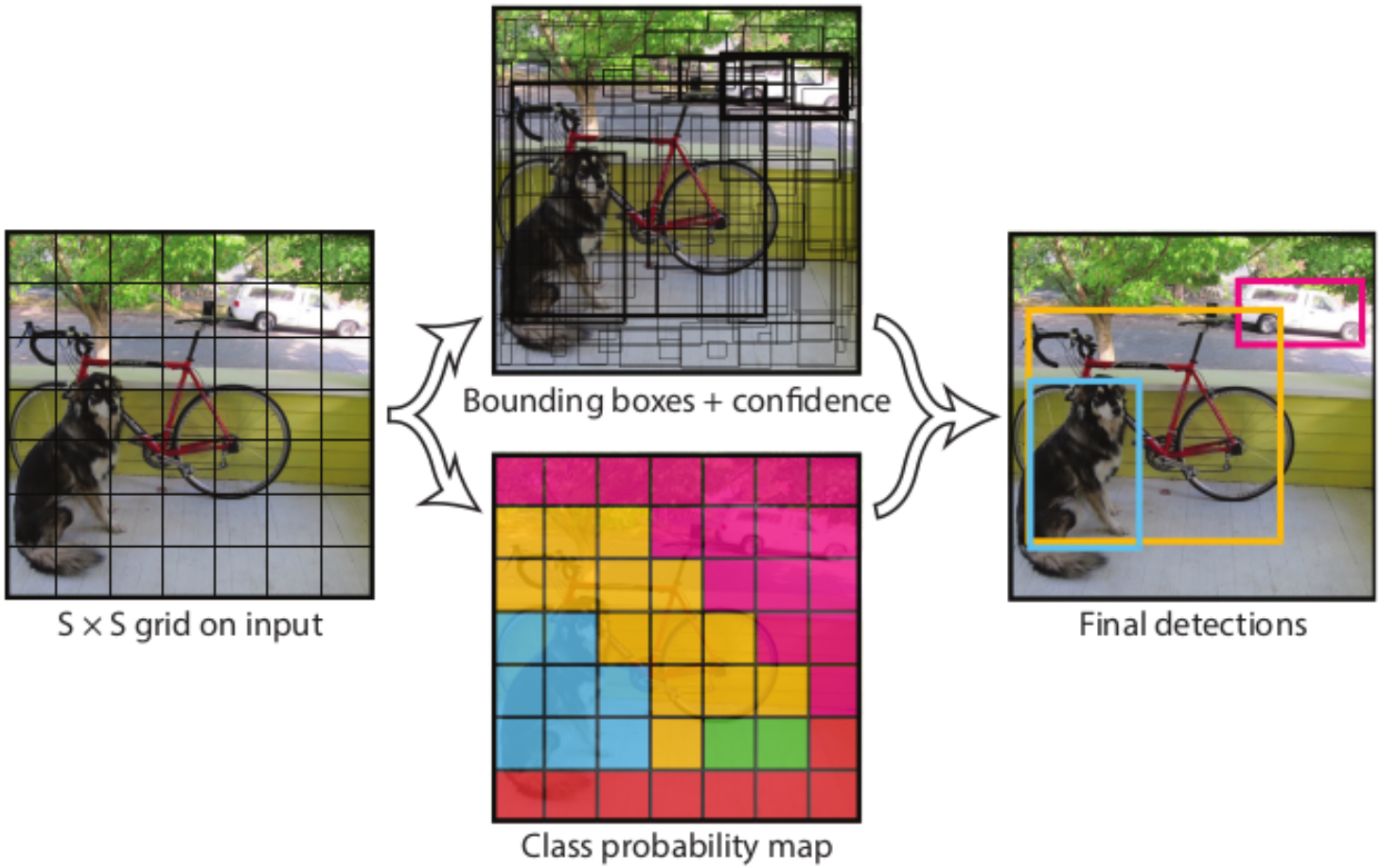




- Each grid cell provide

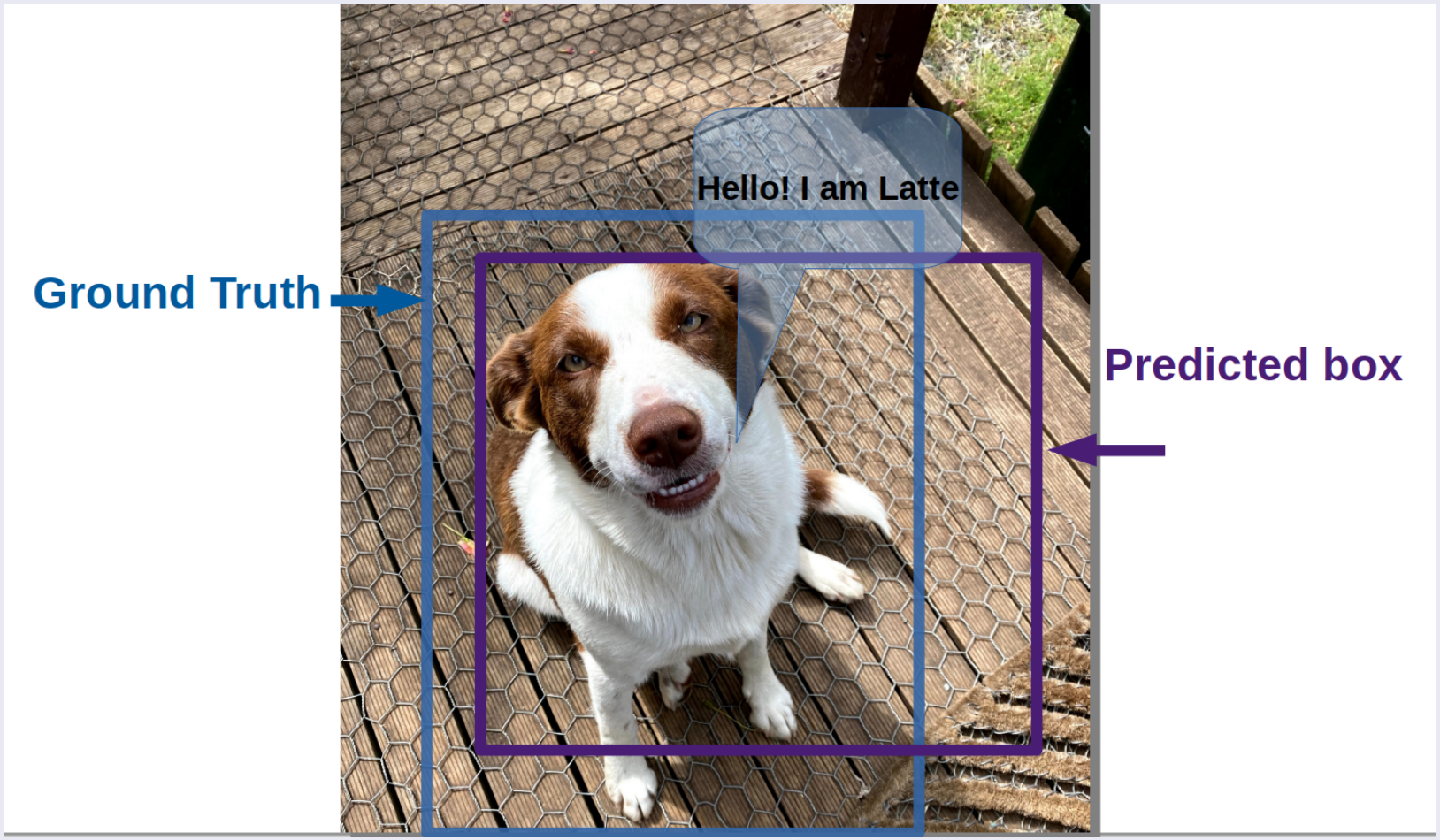
$$\text{bounding box} = \begin{bmatrix} x \\ y \\ w \\ h \end{bmatrix} = \begin{bmatrix} 0.9 \\ 0.9 \\ 3.2 \\ 6 \end{bmatrix}$$

YOLO-v1



Source:[Redmon et al., 2015]

How to determine if it is a good predicted bounding box?

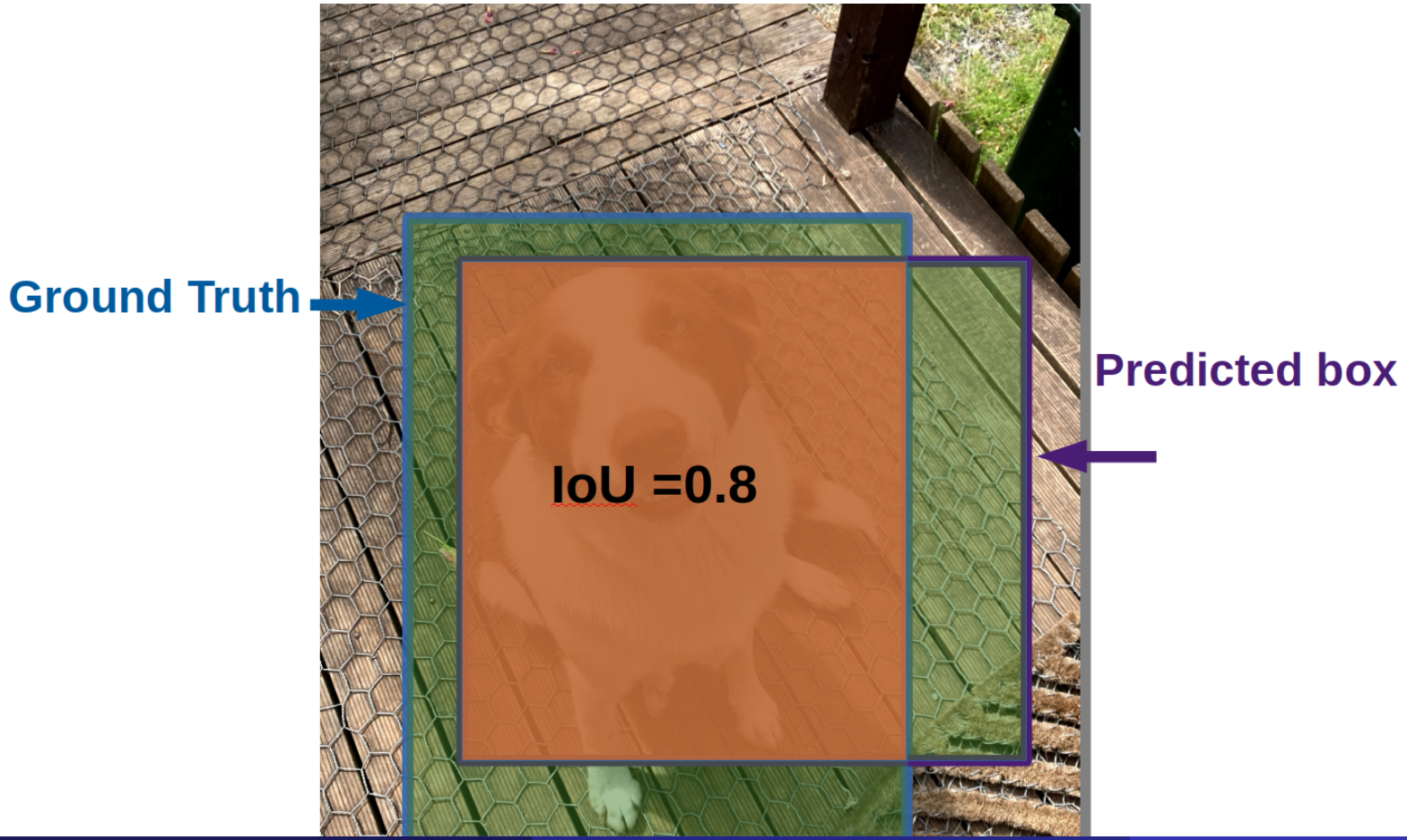


How to evaluate bounding boxes? Intersection over Union

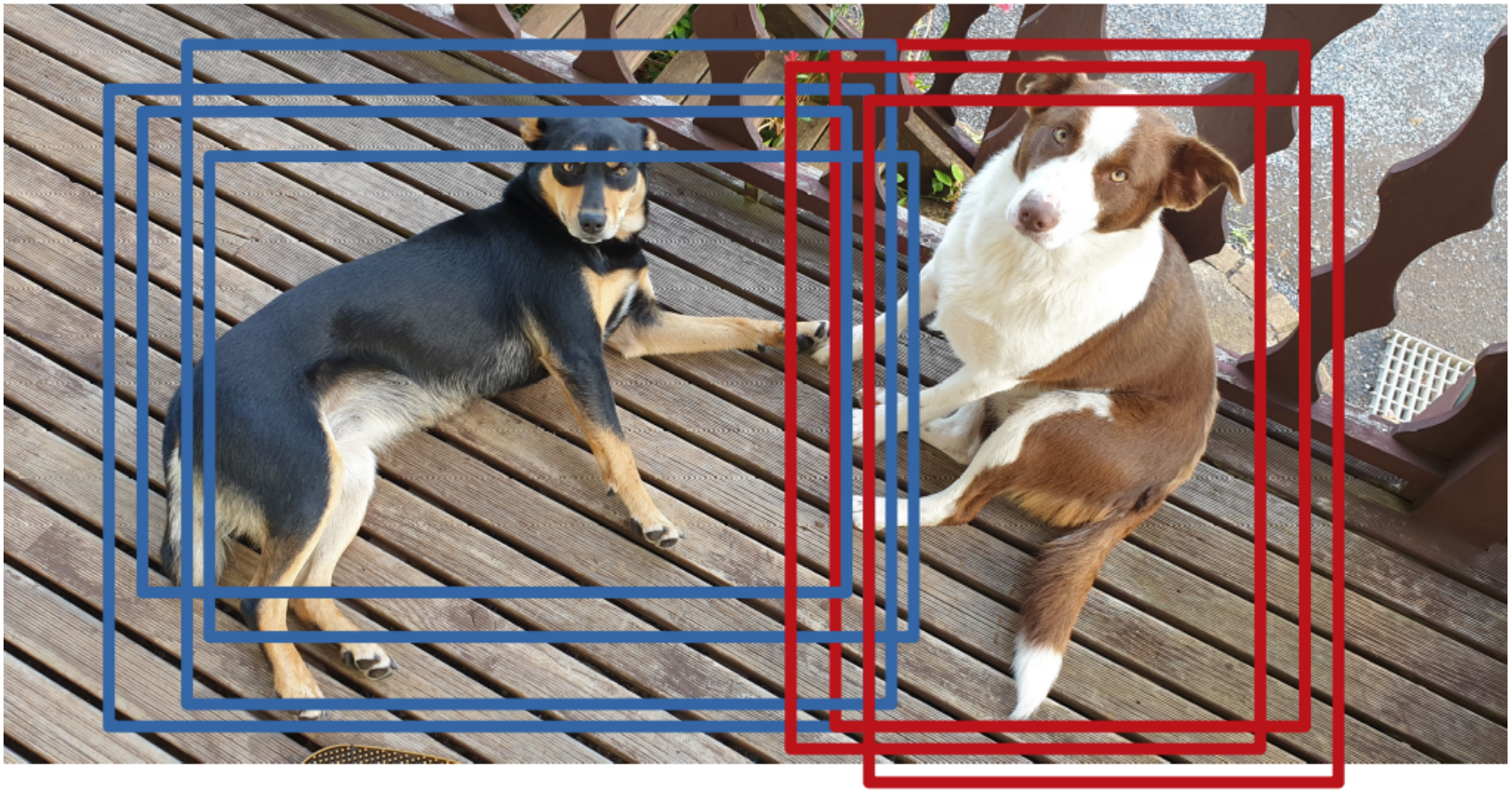
- How to compare the predicted bounding box to the ground truth box?
- Intersection over Union (IoU)

$$IoU = \frac{\text{Area of Intersection}}{\text{Area of Union}}$$

- $IoU > 0.5$ is okay; $IoU > 0.7$ is good, $IoU > .9$ is almost perfect

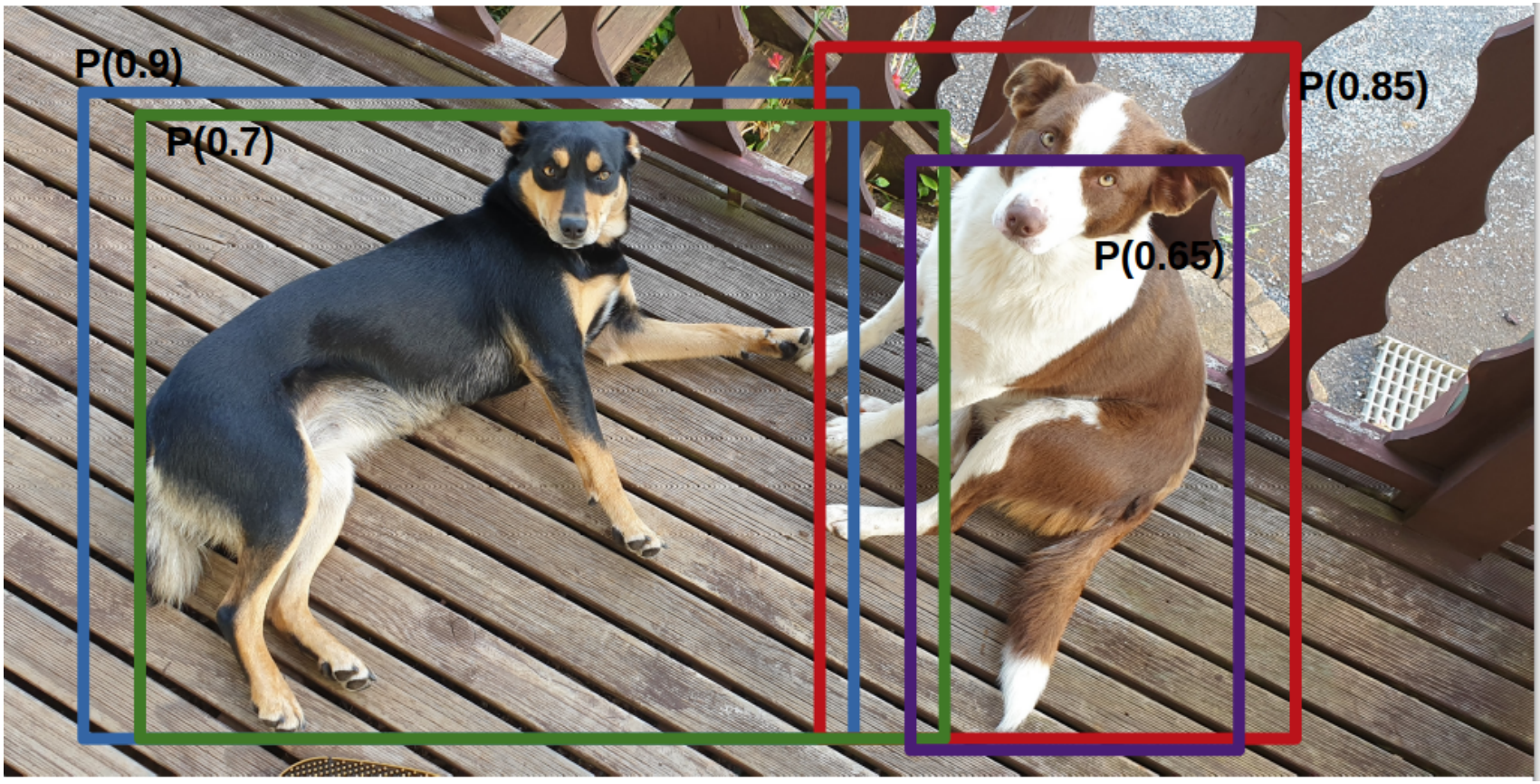


Cleaning multiple bounding boxes

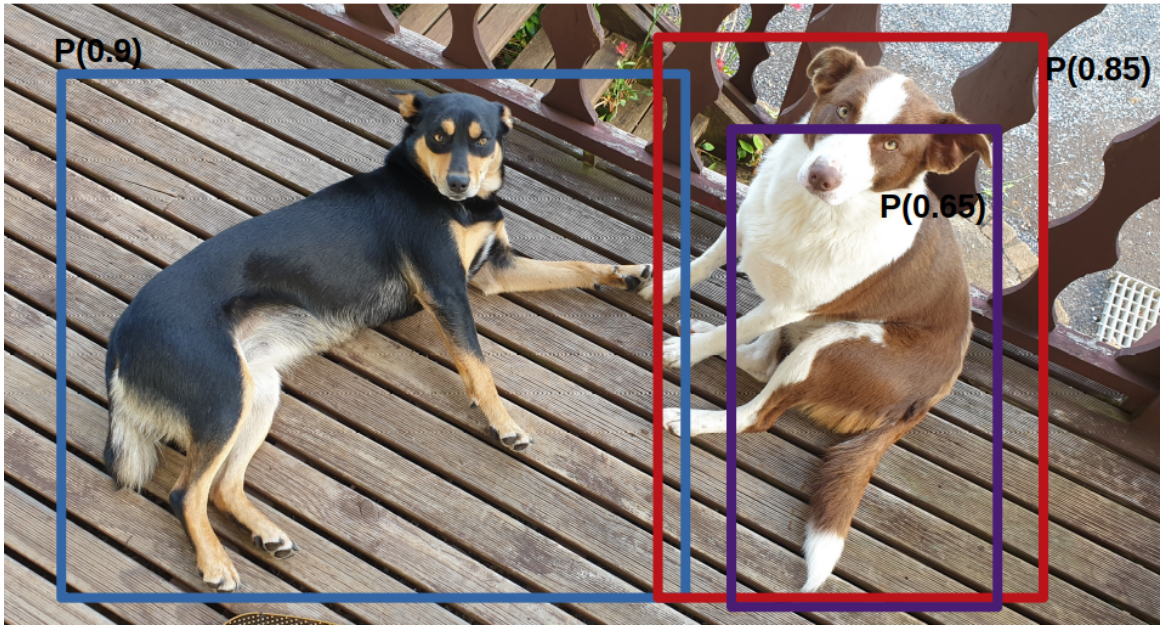


- Object detectors often produce many bounding boxes overlapping. We only need one bounding box to detect an object.

Non-max suppression



- Object detectors often produce many bounding boxes overlapping
- Solution: Non-Max Suppression (NMS)
  1. Discard all boxes with $p_c \leq$ threshold (e.g. 0.6)
  2. Select next highest-scoring box
  3. Eliminate any remaining boxes with IoU $\geq$ threshold (e.g. 0.7)
  4. If any boxes remain, return step 1

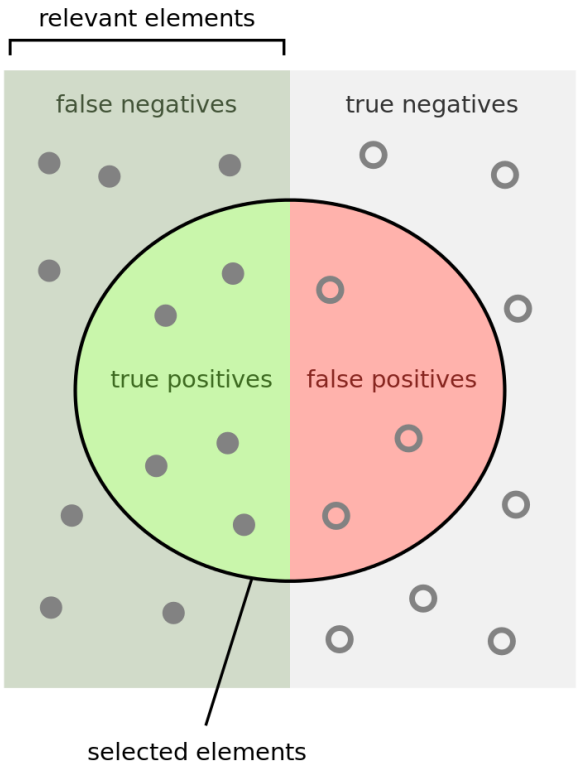

P(0.9)
P(0.85)
P(0.65)


P(0.9)
P(0.85)

How to evaluate the performance of image detection algorithm?

- Precision versus Recall
- Mean Average Precision (mAP)

Precision and Recall



How many selected items are relevant?

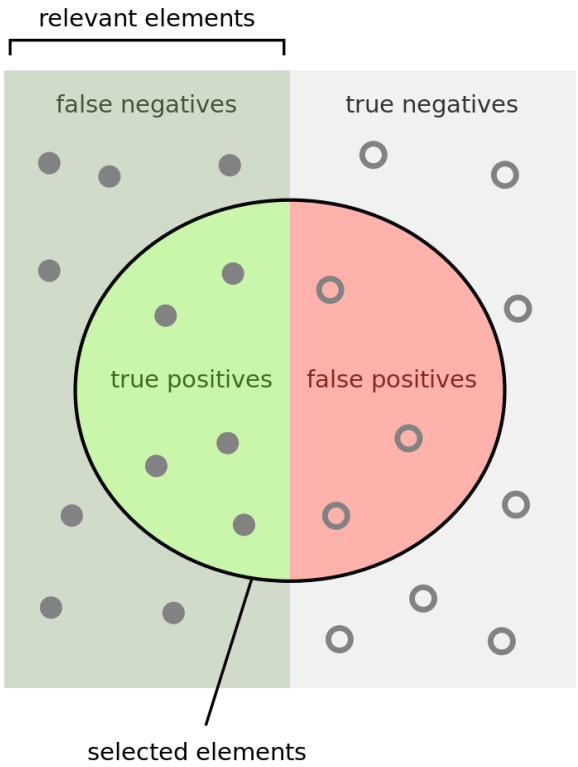
$$\text{Precision} = \frac{\text{true positives}}{\text{selected elements}}$$

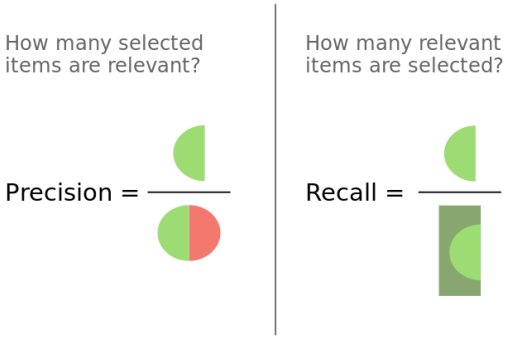
How many relevant items are selected?

$$\text{Recall} = \frac{\text{true positives}}{\text{relevant elements}}$$

Source: Wikipedia

Precision and Recall

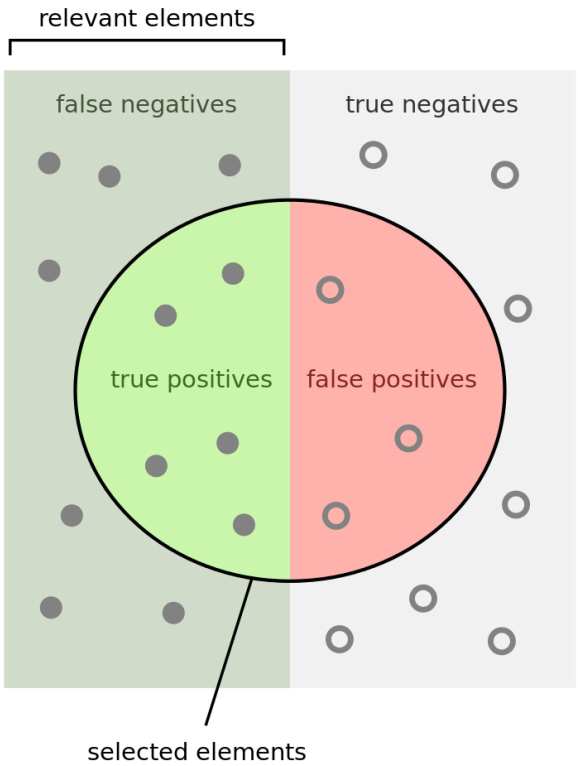


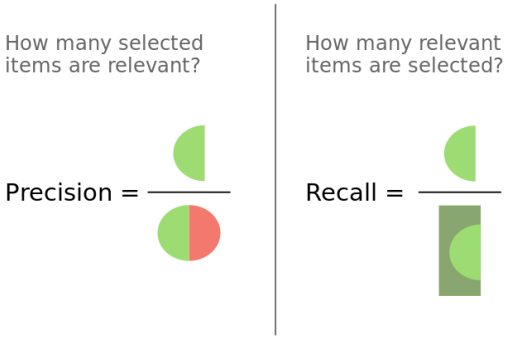


- True positive: the bounding boxes that shares $IoU > .5$ with the ground truth bounding box

Source: Wikipedia

Precision and Recall

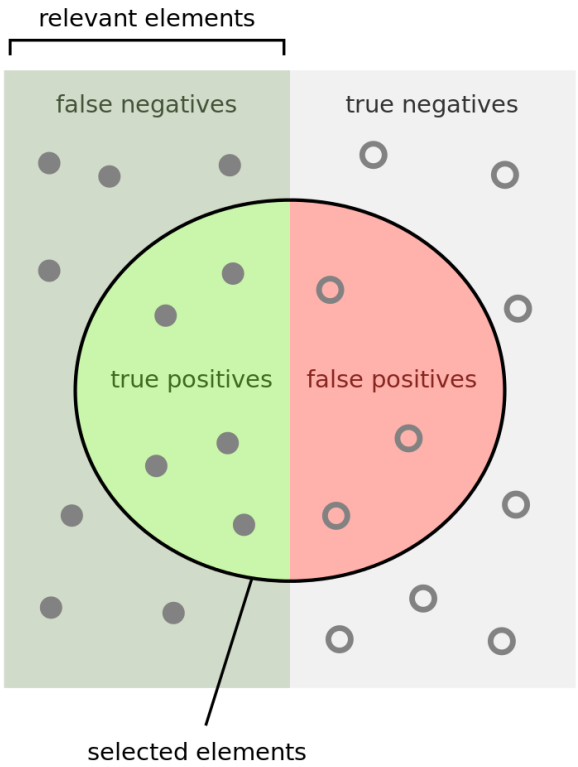


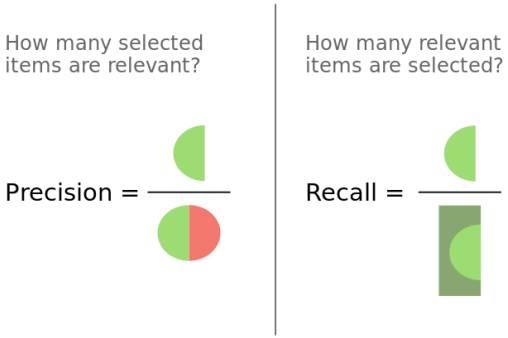


- True positive: the bounding boxes that shares $IoU > .5$ with the ground truth bounding box
- False positive: the bounding boxes that shares $IoU < .5$ with the ground truth bounding box

Source: Wikipedia

Precision and Recall

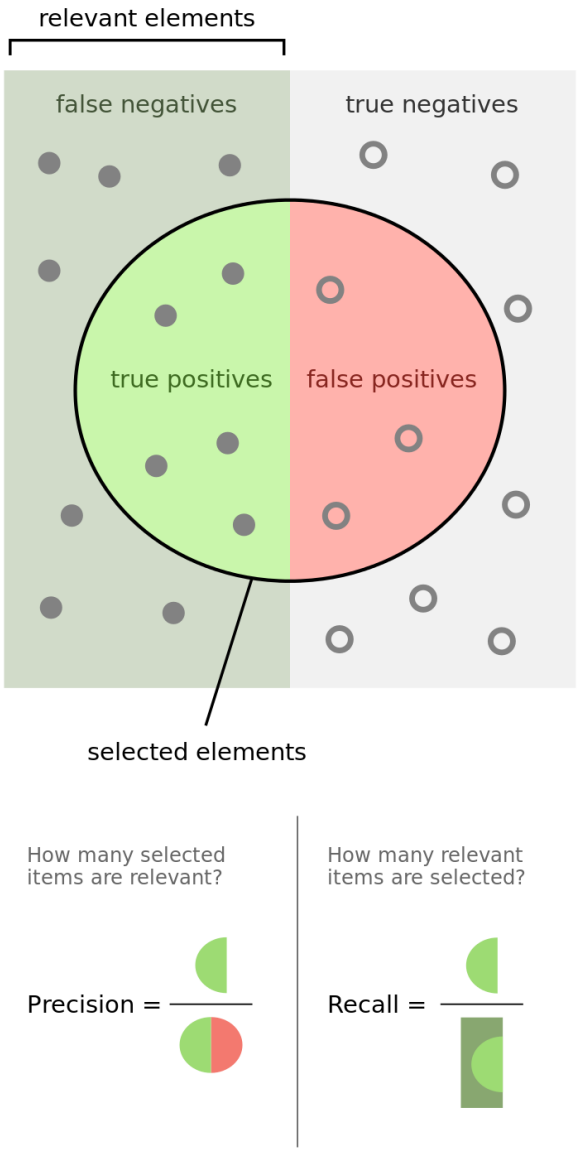




- True positive: the bounding boxes that shares $IoU > .5$ with the ground truth bounding box
- False positive: the bounding boxes that shares $IoU < .5$ with the ground truth bounding box
- False negative: we did not produce the bounding boxes for the ground truth bounding boxes

Source: Wikipedia

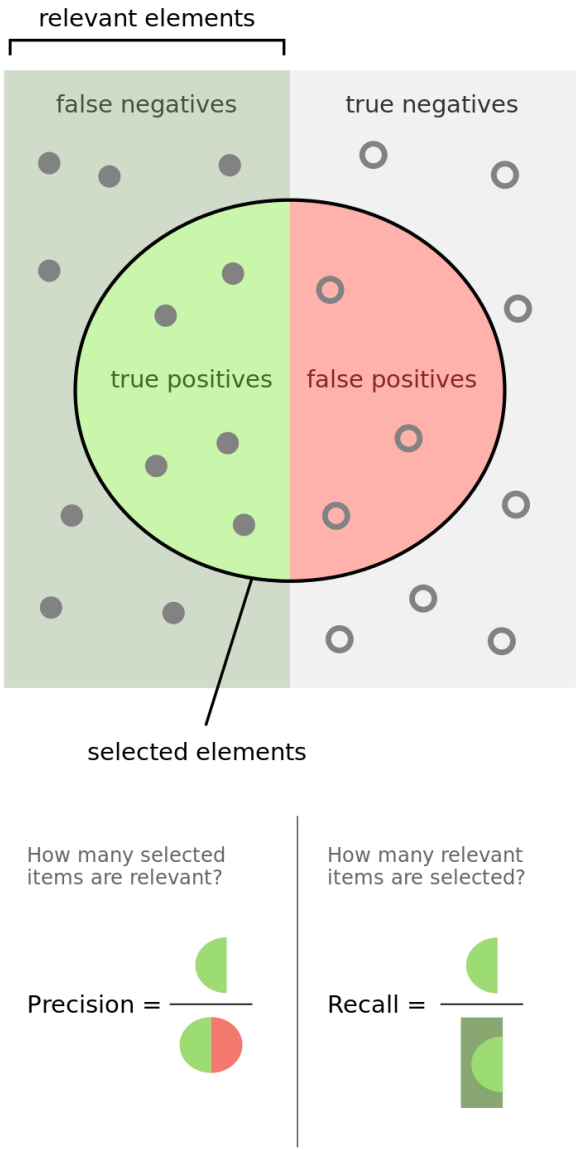
Precision and Recall



- True positive: the bounding boxes that shares $IoU > .5$ with the ground truth bounding box
- False positive: the bounding boxes that shares $IoU < .5$ with the ground truth bounding box
- False negative: we did not produce the bounding boxes for the ground truth bounding boxes
- Precision: Of all bounding box predictions, what fraction was actually correct?

Source: Wikipedia

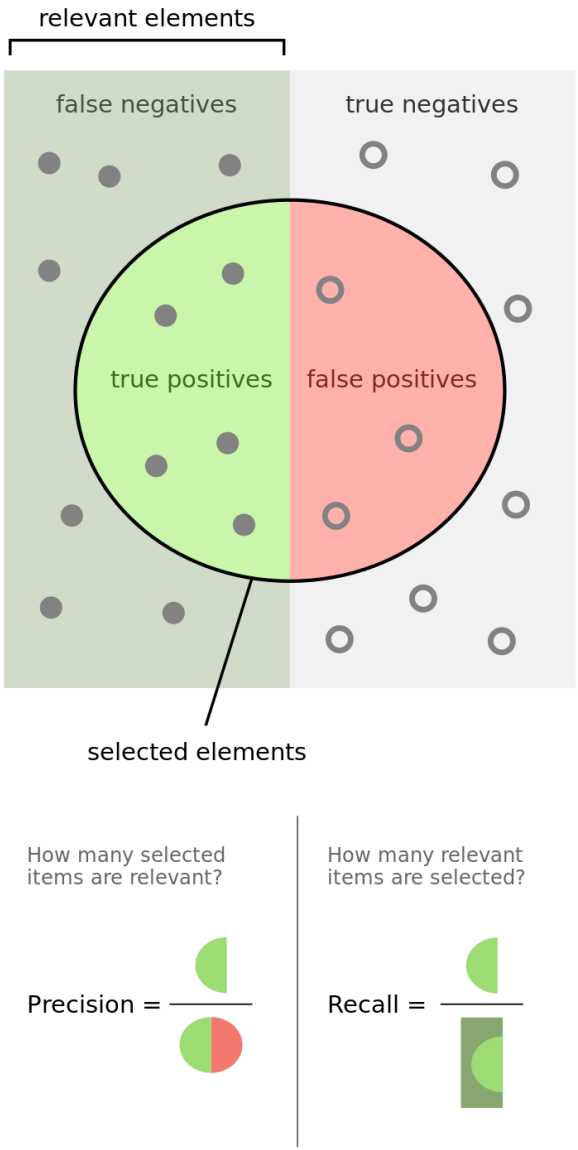
Precision and Recall



Source: Wikipedia

- True positive: the bounding boxes that shares $IoU > .5$ with the ground truth bounding box
- False positive: the bounding boxes that shares $IoU < .5$ with the ground truth bounding box
- False negative: we did not produce the bounding boxes for the ground truth bounding boxes
- Precision: Of all bounding box predictions, what fraction was actually correct?
- Recall: Of all ground truth bounding boxes, what fraction was actually correct?

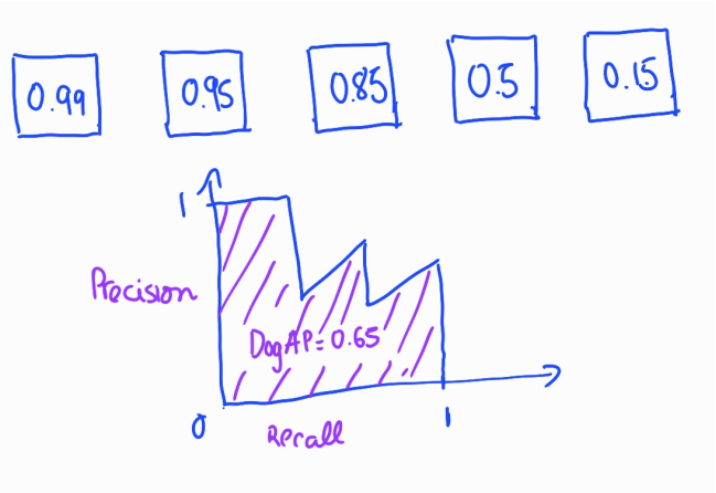
Precision and Recall



- True positive: the bounding boxes that shares $IoU > .5$ with the ground truth bounding box
- False positive: the bounding boxes that shares $IoU < .5$ with the ground truth bounding box
- False negative: we did not produce the bounding boxes for the ground truth bounding boxes
- Precision: Of all bounding box predictions, what fraction was actually correct?
- Recall: Of all ground truth bounding boxes, what fraction was actually correct?

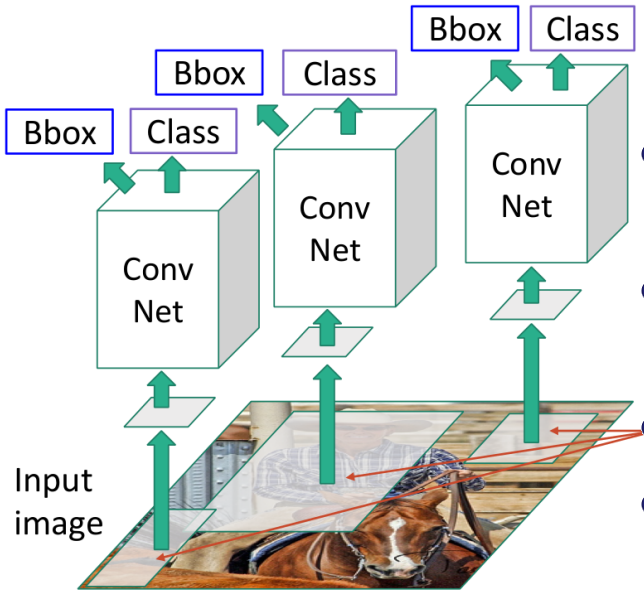
Source: Wikipedia

MAP



- The area under the PR curve is **average precision**
  - DogAP = 0.65
  - CatAP = 0.5
  - mAP = (0.55+0.6)/2.
  - The mAP is calculated at a specific IoU threshold of 0.5
- We need to redo all computation for IoU .5, 0.55, ... 0.95. and take average this and this will be for the final results.$mAP@0.5 : 0.05 : 0.95$

Regional proposals



Source: Justin Johnson

- Run region proposal method to extract around 2000 region proposals
- Resize each region to 224x224 and run independently through CNN to predict class scores and bounding boxes
- Use scores to select a subset of region proposals to output
- Compare with ground-truth boxes

Code to practice with image detection

- YOLO on SageMaker
- SSD on SageMaker
- Mask-RCNN on SageMaker

Lab session

Built-in Computer Vision in SageMaker for Image Classification

Next week

- Transfer learning
- QA session and Review A2

References I

Girshick, R. (2015).

Fast r-cnn.
In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014).

Rich feature hierarchies for accurate object detection and semantic segmentation.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. B. (2017).

Mask R-CNN.
CoRR, abs/1703.06870.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S. E., Fu, C., and Berg, A. C. (2015).

SSD: single shot multibox detector.
CoRR, abs/1512.02325.

Redmon, J., Divvala, S. K., Girshick, R. B., and Farhadi, A. (2015).

You only look once: Unified, real-time object detection.
CoRR, abs/1506.02640.

Redmon, J. and Farhadi, A. (2018).

Yolov3: An incremental improvement.
CoRR, abs/1804.02767.

Ren, S., He, K., Girshick, R. B., and Sun, J. (2015).

Faster R-CNN: towards real-time object detection with region proposal networks.
CoRR, abs/1506.01497.