

Object detection: Introduction



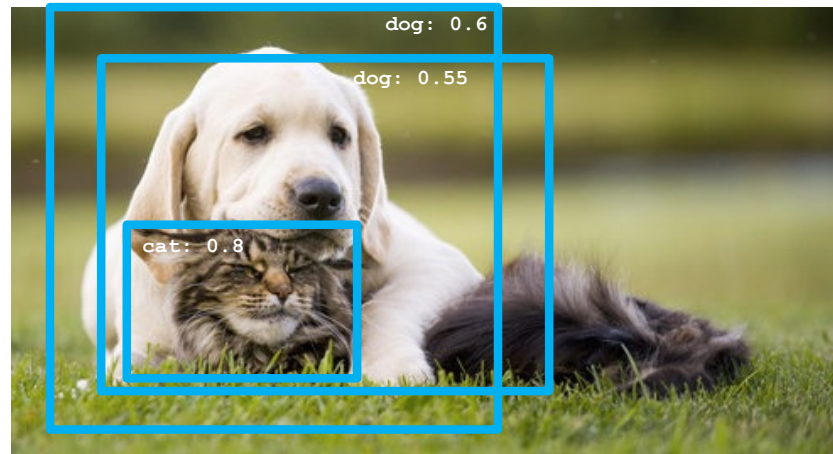
[Image source](#)

Object detection: Introduction

- Evaluating detectors
 - Intersection over union (IoU)
 - Non-maximum suppression
 - Recall, precision, AP, mAP
- Early datasets
- Early detection architectures: two-stage vs. single-stage
- YOLO detector

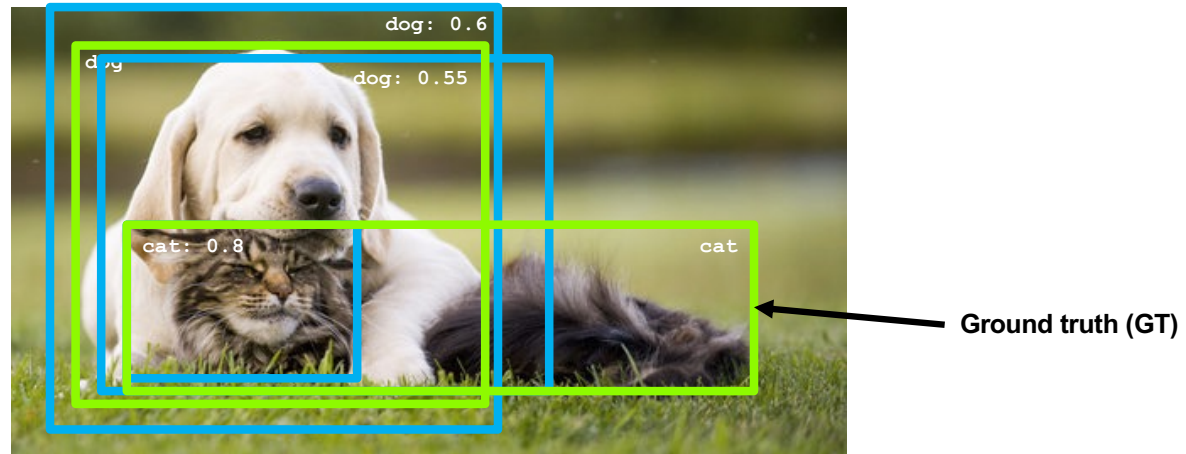
Task definition and evaluation

- What does a detector predict?
 - Bounding boxes, class labels, and confidence scores
- How can we evaluate detector output?



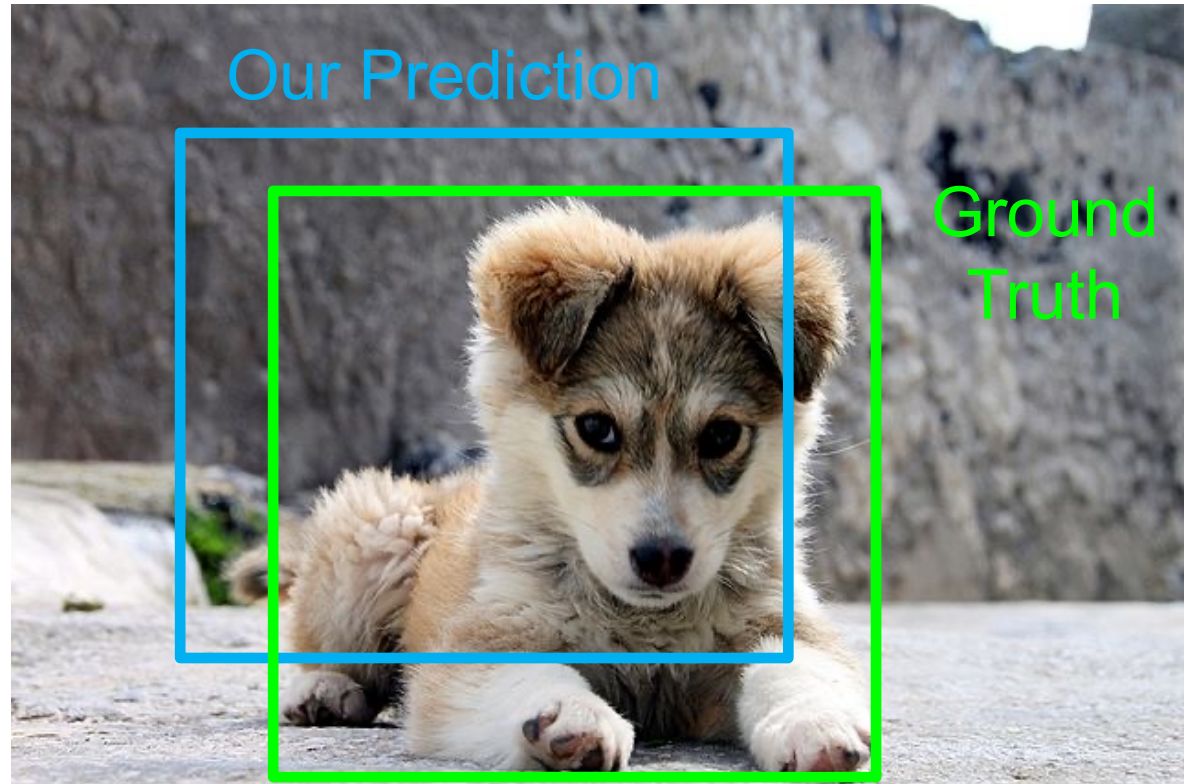
Task definition and evaluation

- What does a detector predict?
 - Bounding boxes, class labels, and confidence scores
- How can we evaluate detector output?
 - Step 1: for each predicted bounding box, determine whether it's a **true positive** or **false positive**



Comparing Boxes: Intersection over Union (IoU)

How can we compare our prediction to the ground-truth box?



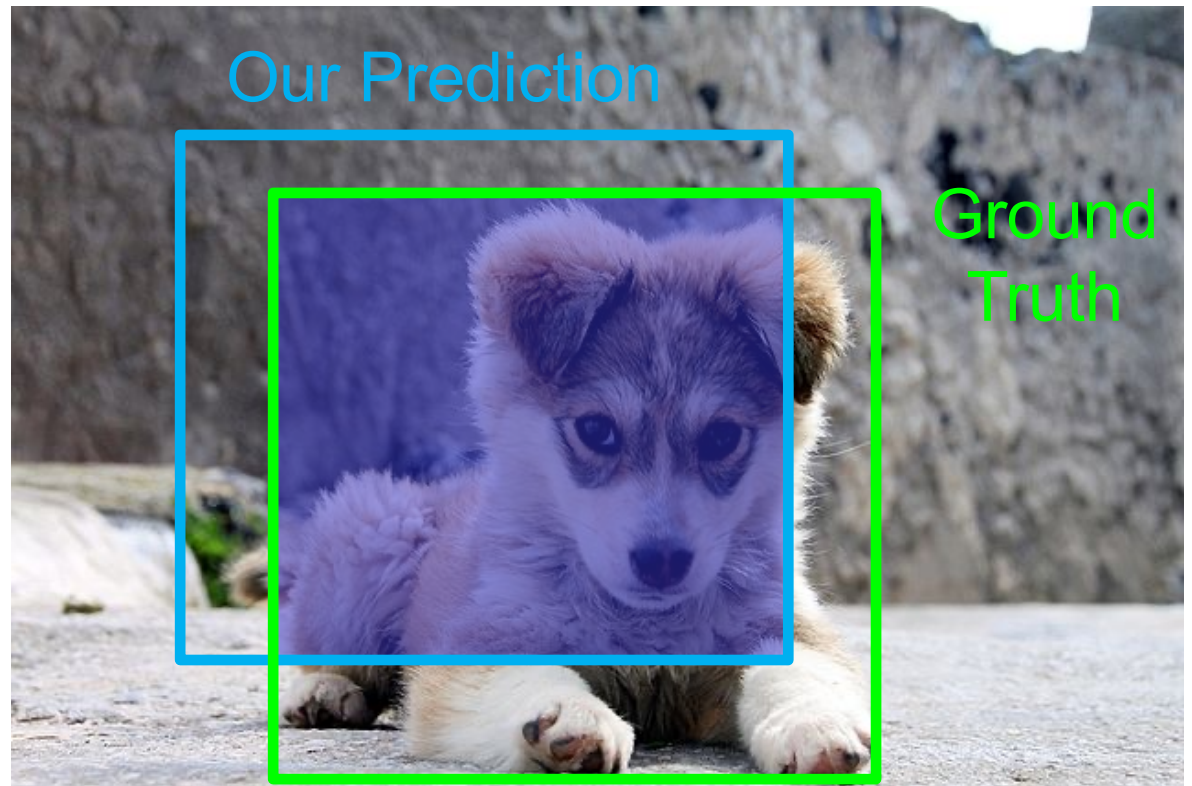
Source: [J. Johnson](#)

Comparing Boxes: Intersection over Union (IoU)

How can we compare our prediction to the ground-truth box?

Intersection over Union (IoU):

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



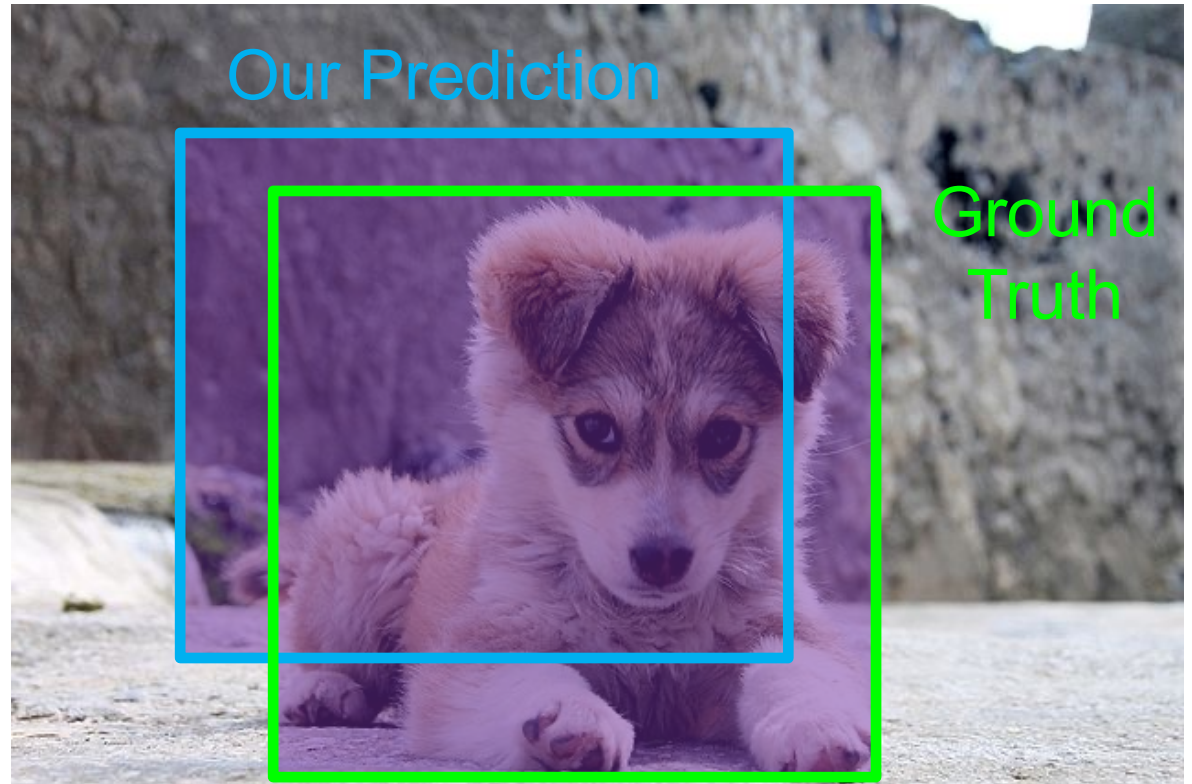
Source: [J. Johnson](#)

Comparing Boxes: Intersection over Union (IoU)

How can we compare our prediction to the ground-truth box?

Intersection over Union (IoU):

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



Source: [J. Johnson](#)

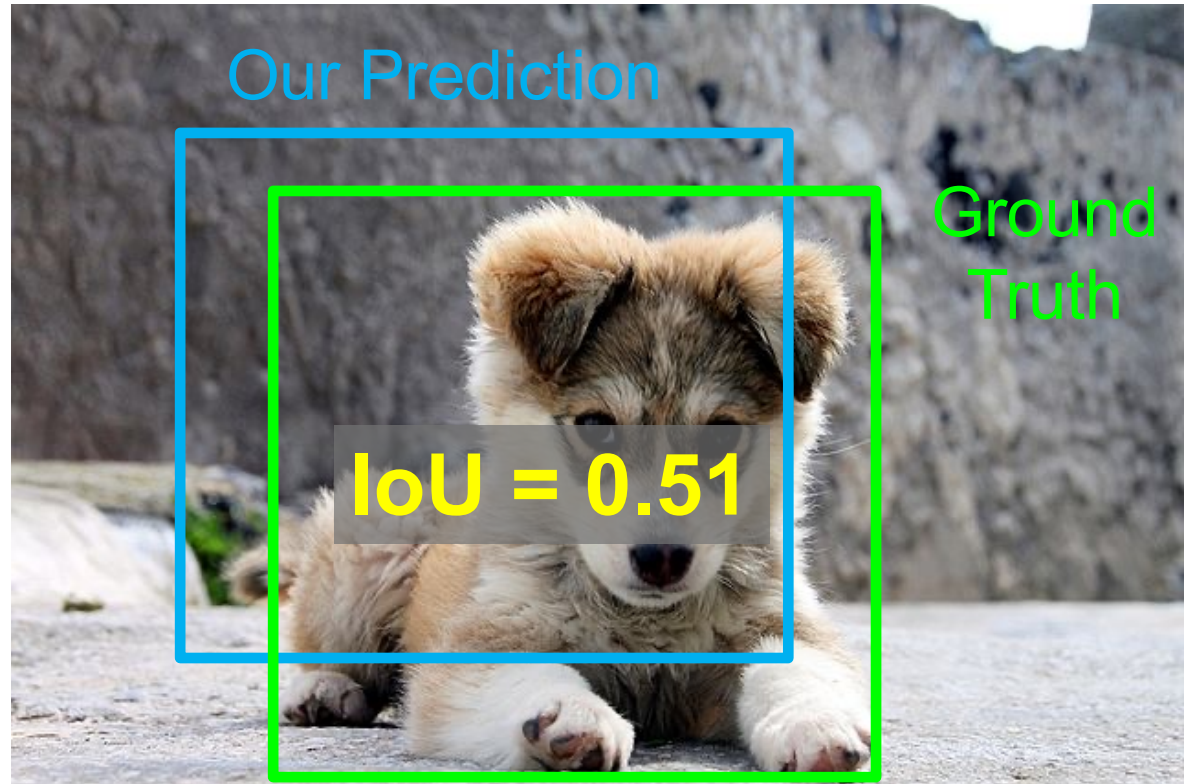
Comparing Boxes: Intersection over Union (IoU)

How can we compare our prediction to the ground-truth box?

Intersection over Union (IoU):

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

IoU > 0.5 is “decent”



Source: [J. Johnson](#)

Comparing Boxes: Intersection over Union (IoU)

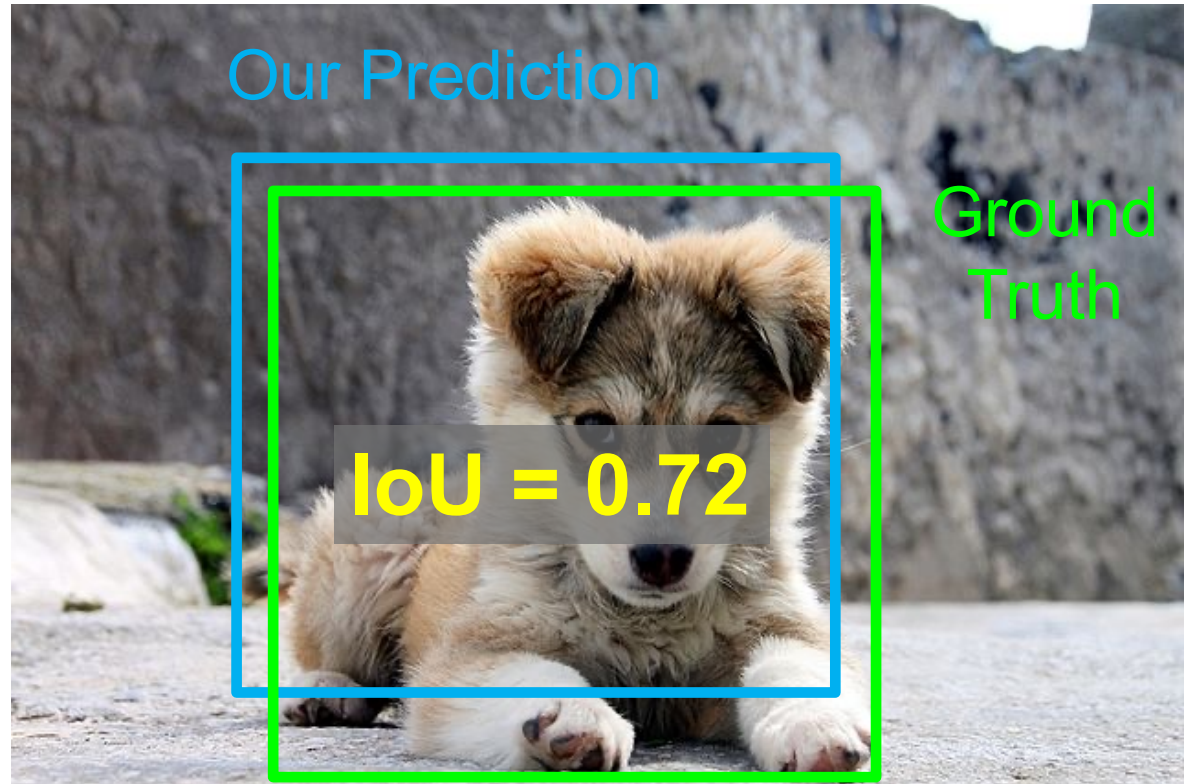
How can we compare our prediction to the ground-truth box?

Intersection over Union (IoU):

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

IoU > 0.5 is “decent”

IoU > 0.7 is “pretty good”



Comparing Boxes: Intersection over Union (IoU)

How can we compare our prediction to the ground-truth box?

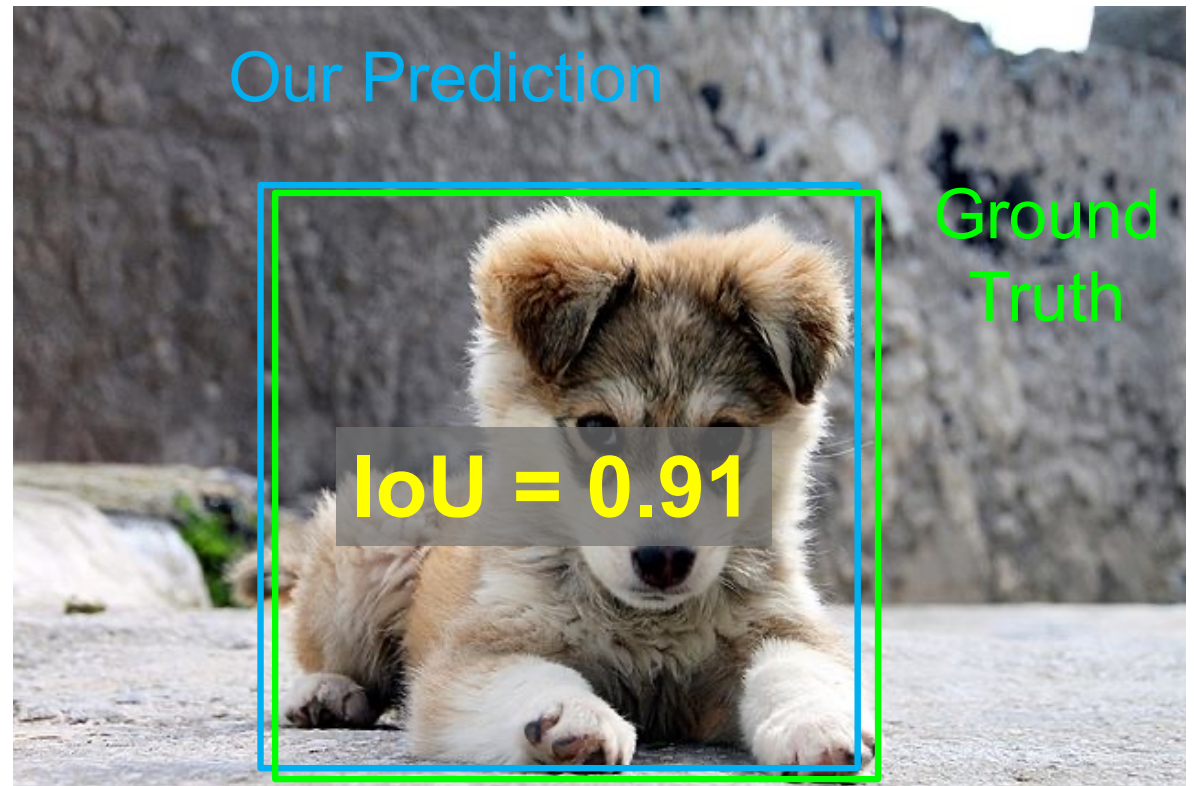
Intersection over Union (IoU):

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

IoU > 0.5 is “decent”

IoU > 0.7 is “pretty good”

IoU > 0.9 is “almost perfect”

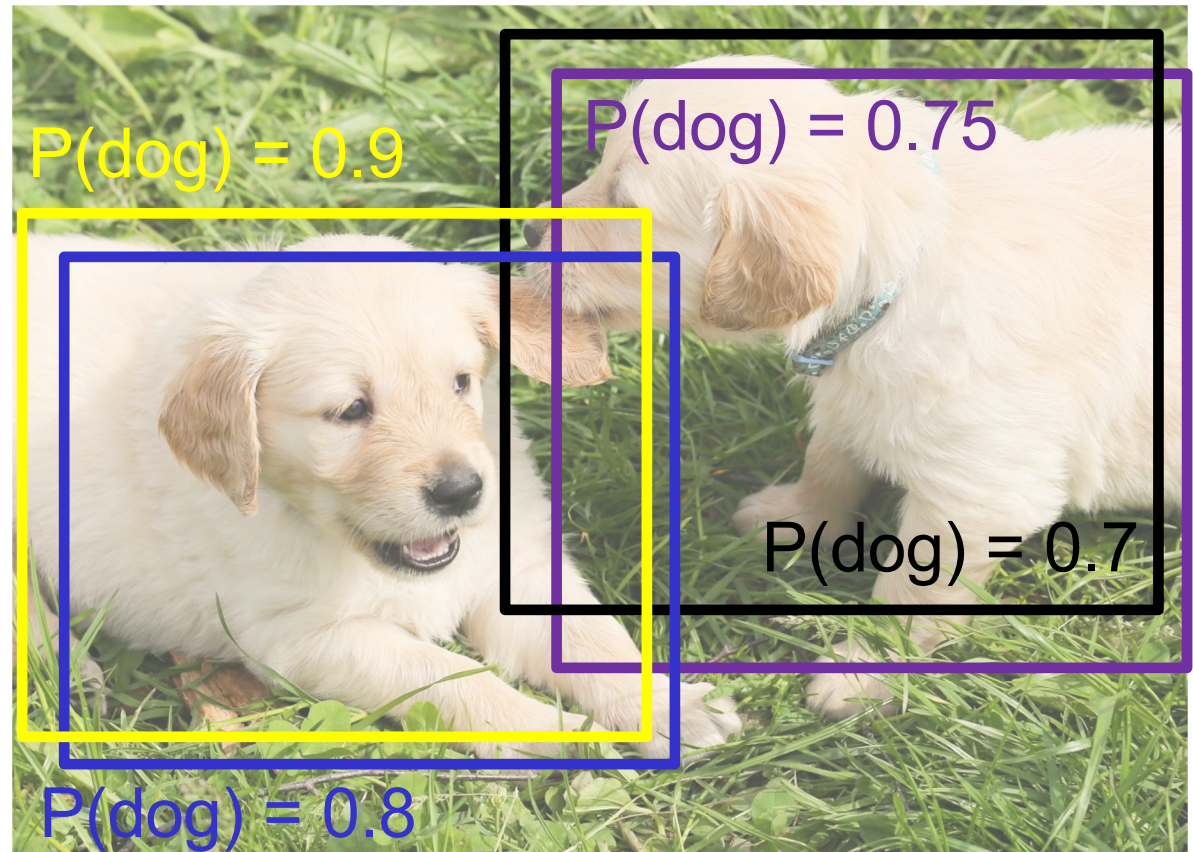


Source: [J. Johnson](#)

Non-maximum suppression

Problem: Detectors often output many overlapping detections

Solution: Post-process raw detections using Non-Maximum Suppression (NMS)



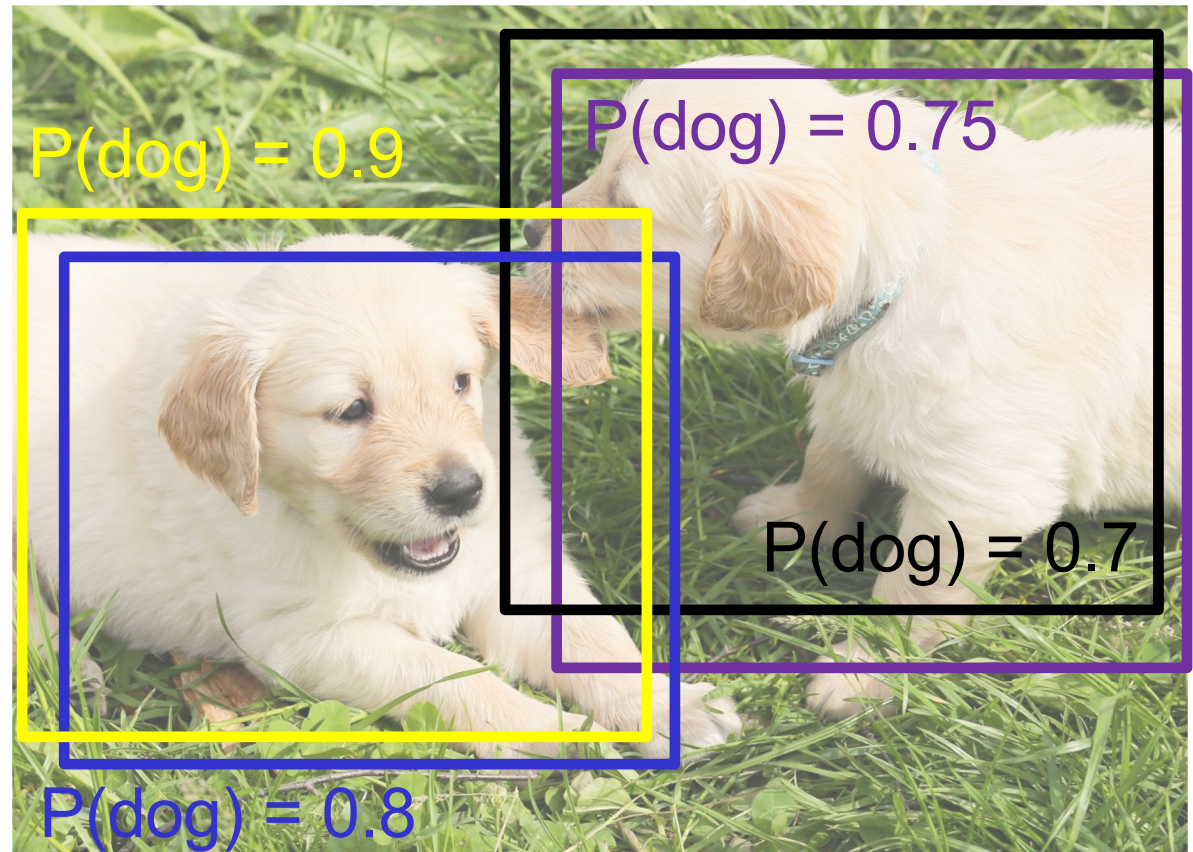
Non-maximum suppression

Problem: Detectors often output many overlapping detections

Solution: Post-process raw detections using Non-Maximum Suppression (NMS)

Typical NMS procedure:

1. Select next highest-scoring box
2. Eliminate lower-scoring boxes with IoU > threshold (e.g. 0.7)
3. If boxes remain, GOTO 1



Non-maximum suppression

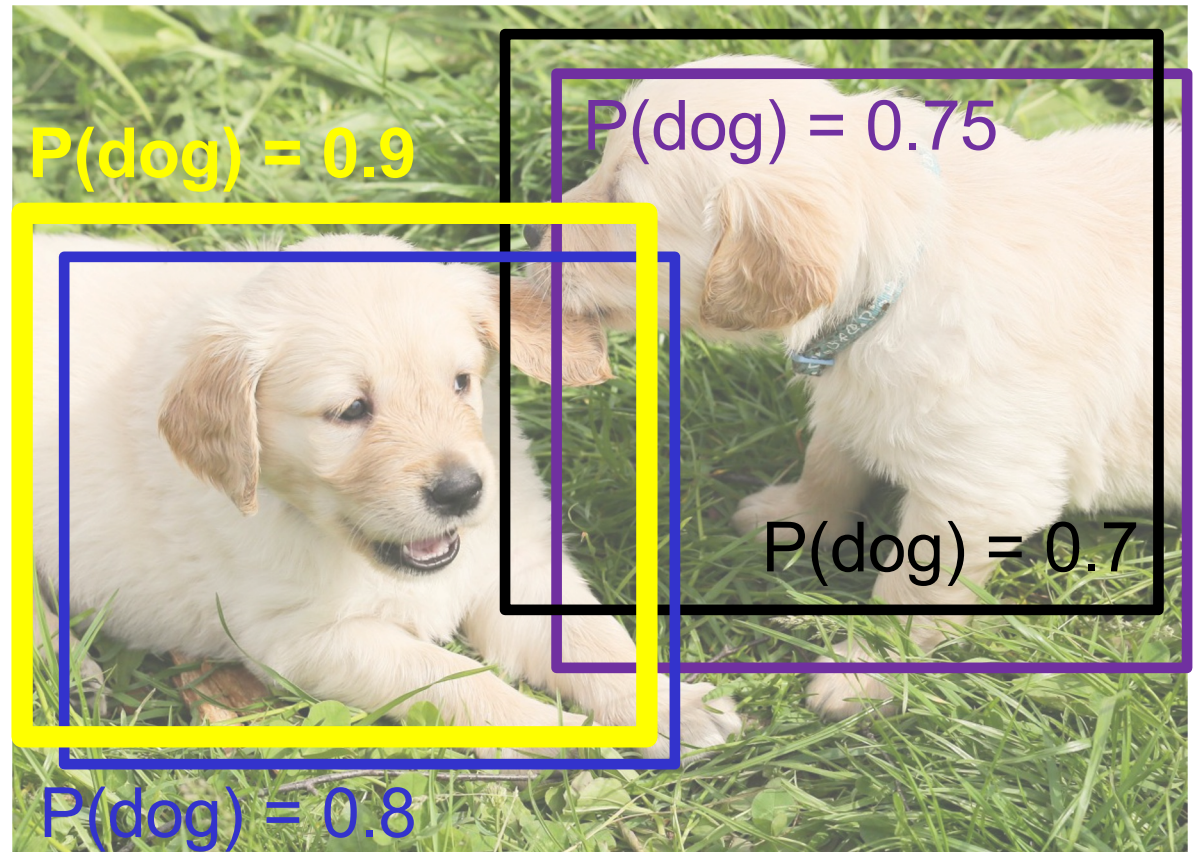
Problem: Detectors often output many overlapping detections

Solution: Post-process raw detections using Non-Maximum Suppression (NMS)

Typical NMS procedure:

1. Select next highest-scoring box
2. Eliminate lower-scoring boxes with IoU > threshold (e.g. 0.7)
3. If boxes remain, GOTO 1

Source: [J. Johnson](#)



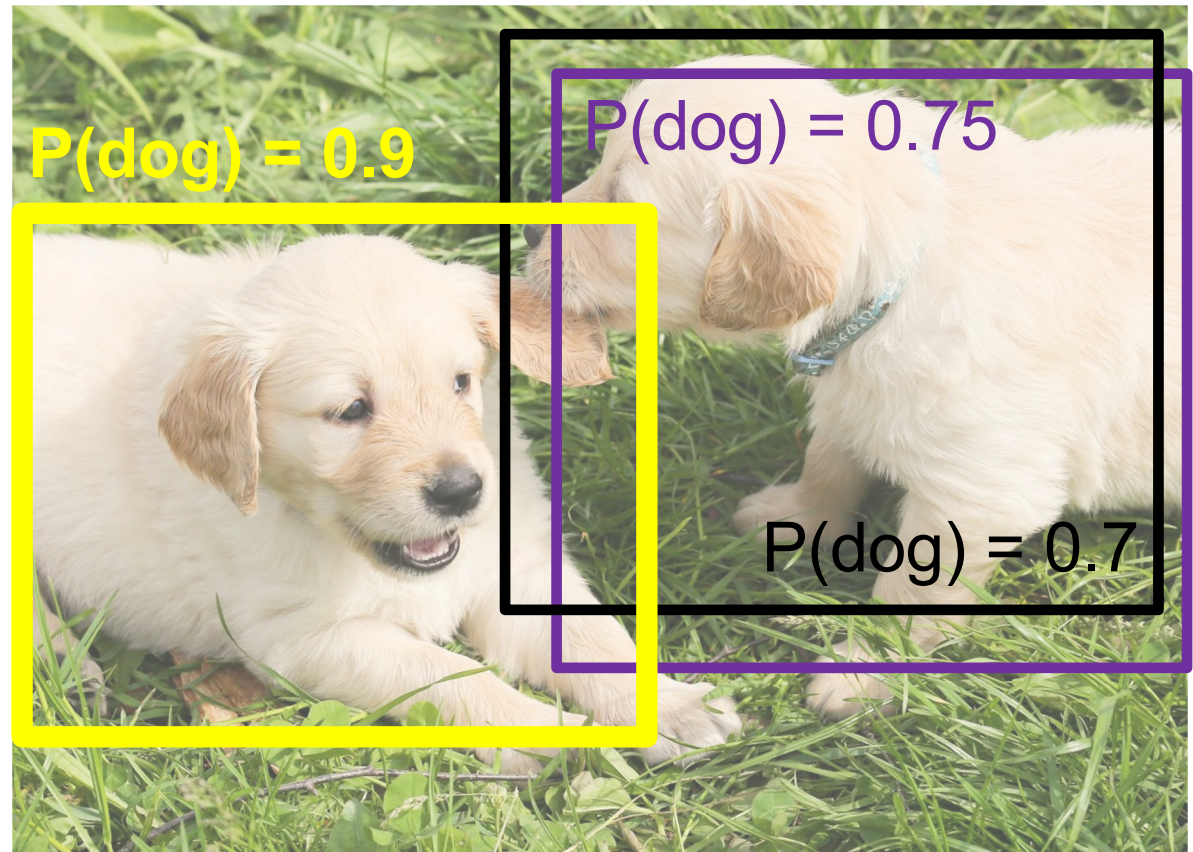
Non-maximum suppression

Problem: Detectors often output many overlapping detections

Solution: Post-process raw detections using Non-Maximum Suppression (NMS)

Typical NMS procedure:

1. Select next highest-scoring box
2. Eliminate lower-scoring boxes with IoU > threshold (e.g. 0.7)
3. If boxes remain, GOTO 1



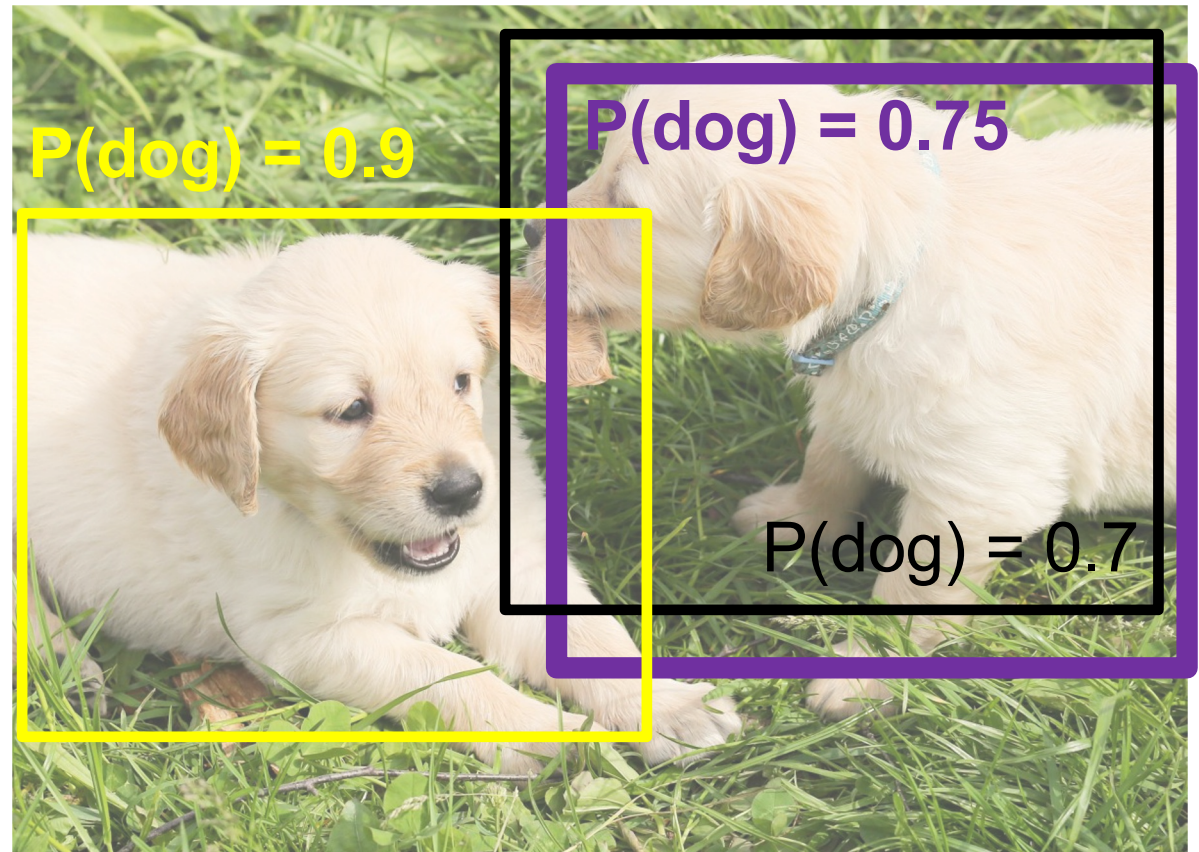
Non-maximum suppression

Problem: Detectors often output many overlapping detections

Solution: Post-process raw detections using Non-Maximum Suppression (NMS)

Typical NMS procedure:

1. Select next highest-scoring box
2. Eliminate lower-scoring boxes with IoU > threshold (e.g. 0.7)
3. If boxes remain, GOTO 1



Non-maximum suppression

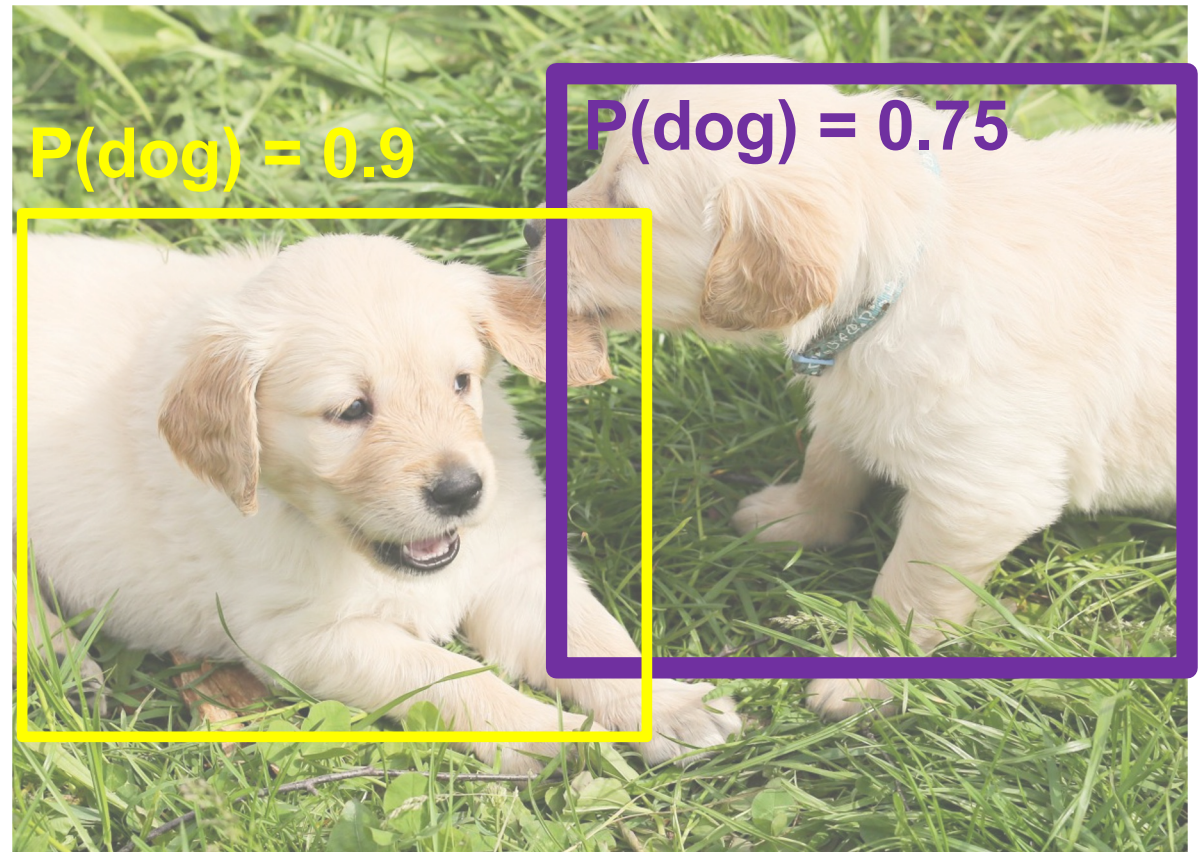
Problem: Detectors often output many overlapping detections

Solution: Post-process raw detections using Non-Maximum Suppression (NMS)

Typical NMS procedure:

1. Select next highest-scoring box
2. Eliminate lower-scoring boxes with IoU > threshold (e.g. 0.7)
3. If boxes remain, GOTO 1

Source: [J. Johnson](#)



Non-maximum suppression

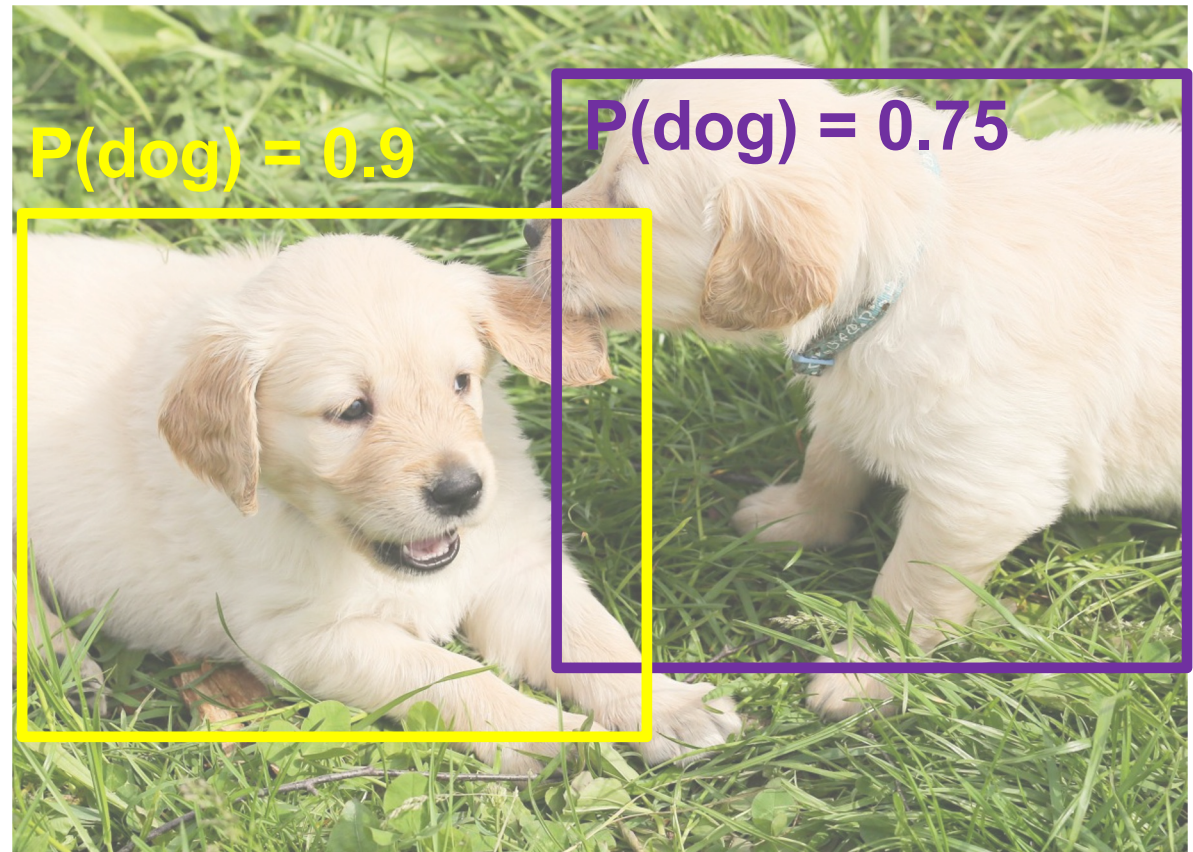
Problem: Detectors often output many overlapping detections

Solution: Post-process raw detections using Non-Maximum Suppression (NMS)

Typical NMS procedure:

1. Select next highest-scoring box
2. Eliminate lower-scoring boxes with IoU > threshold (e.g. 0.7)
3. If boxes remain, GOTO 1

Source: [J. Johnson](#)



Limitations of NMS



Source: [J. Johnson](#)

Object detection: Introduction

- Evaluating detectors
 - Intersection over union (IoU)
 - Non-maximum suppression
 - Recall, precision, AP, mAP

Evaluating object detectors

1. Run object detector on all test images (with NMS)
2. For each category, compute the **Precision vs. Recall Curve** and **Average Precision (AP)**, or area under the Precision vs. Recall Curve
3. Average the APs of all categories to obtain **mean AP**, or **mAP**

Evaluating object detectors

1. Run object detector on all test images (with NMS)
2. For each category, compute the **Precision vs. Recall Curve**:

Curve:

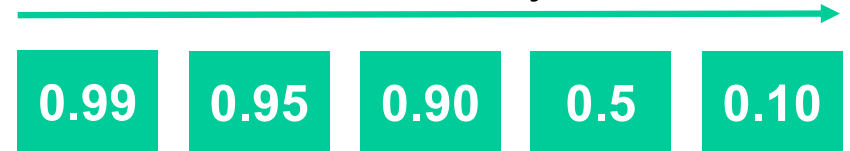
1. For each detection (highest to lowest score)
 1. If it matches some GT box with IoU > 0.5, mark it as positive and eliminate the GT
 2. Otherwise mark it as negative
 3. Plot a point on PR Curve:

$$\text{Precision} = \frac{\text{true positive detections}}{\text{total detections so far}}$$

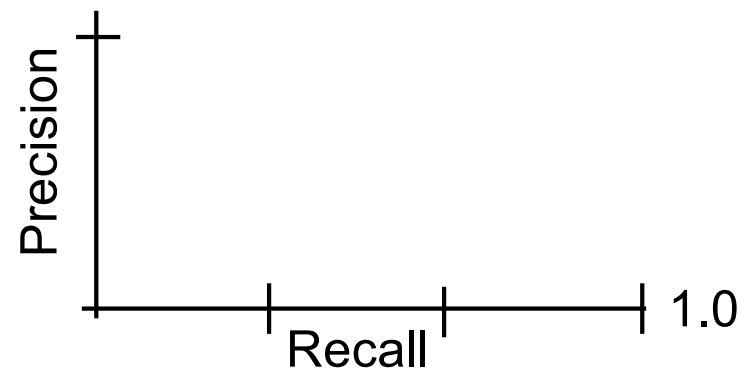
$$\text{Recall} = \frac{\text{true positive detections}}{\text{true positive test instances}}$$

Source: [J. Johnson](#)

All detections sorted by score



All ground-truth boxes



Evaluating object detectors

1. Run object detector on all test images (with NMS)
2. For each category, compute the **Precision vs. Recall Curve**:

Curve:

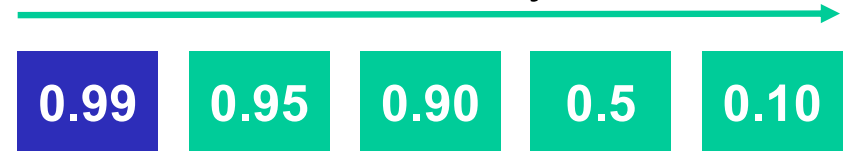
1. For each detection (highest to lowest score)
 1. If it matches some GT box with $IoU > 0.5$, mark it as positive and eliminate the GT
 2. Otherwise mark it as negative
 3. Plot a point on PR Curve:

$$Precision = \frac{\text{true positive detections}}{\text{total detections so far}}$$

$$Recall = \frac{\text{true positive detections}}{\text{true positive test instances}}$$

Source: [J. Johnson](#)

All detections sorted by score



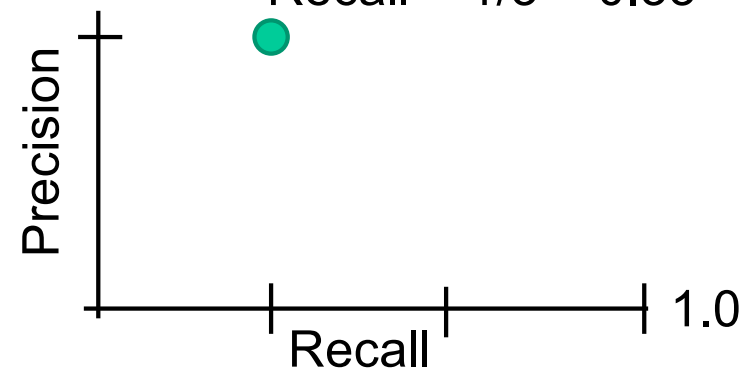
Match: $IoU > 0.5$



All ground-truth boxes

$$Precision = 1/1 = 1.0$$

$$Recall = 1/3 = 0.33$$



Evaluating object detectors

1. Run object detector on all test images (with NMS)
2. For each category, compute the **Precision vs. Recall Curve**:

Curve:

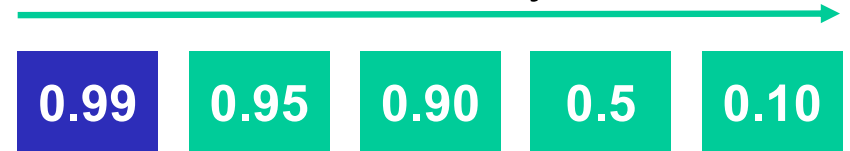
1. For each detection (highest to lowest score)
 1. If it matches some GT box with IoU > 0.5, mark it as positive and eliminate the GT
 2. Otherwise mark it as negative
 3. Plot a point on PR Curve:

$$\text{Precision} = \frac{\text{true positive detections}}{\text{total detections so far}}$$

$$\text{Recall} = \frac{\text{true positive detections}}{\text{true positive test instances}}$$

Source: [J. Johnson](#)

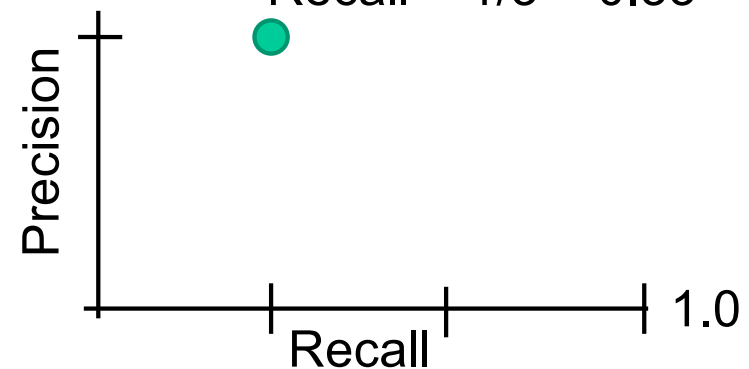
All detections sorted by score



All ground-truth boxes

$$\text{Precision} = 1/1 = 1.0$$

$$\text{Recall} = 1/3 = 0.33$$



Evaluating object detectors

1. Run object detector on all test images (with NMS)
2. For each category, compute the **Precision vs. Recall Curve**:

Curve:

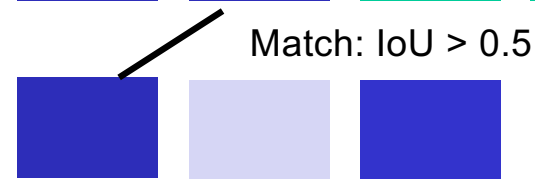
1. For each detection (highest to lowest score)
 1. If it matches some GT box with $IoU > 0.5$, mark it as positive and eliminate the GT
 2. Otherwise mark it as negative
 3. Plot a point on PR Curve:

$$Precision = \frac{\text{true positive detections}}{\text{total detections so far}}$$

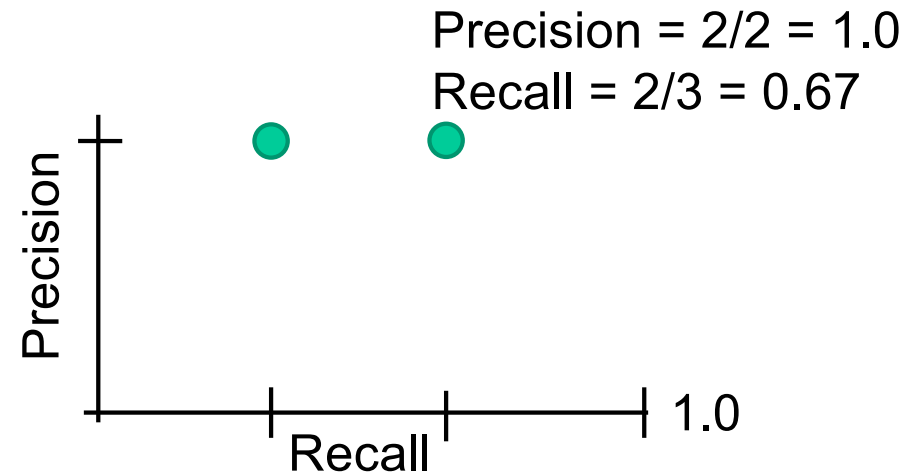
$$Recall = \frac{\text{true positive detections}}{\text{true positive test instances}}$$

Source: [J. Johnson](#)

All detections sorted by score



All ground-truth boxes



Evaluating object detectors

1. Run object detector on all test images (with NMS)
2. For each category, compute the **Precision vs. Recall Curve**:

Curve:

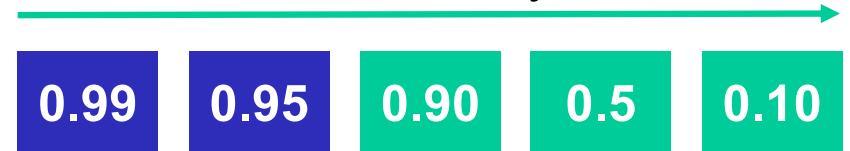
1. For each detection (highest to lowest score)
 1. If it matches some GT box with IoU > 0.5, mark it as positive and eliminate the GT
 2. Otherwise mark it as negative
 3. Plot a point on PR Curve:

$$\text{Precision} = \frac{\text{true positive detections}}{\text{total detections so far}}$$

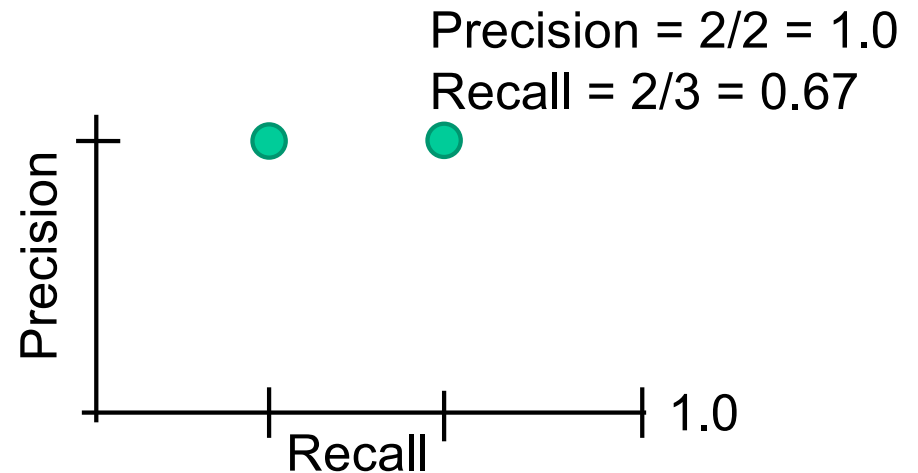
$$\text{Recall} = \frac{\text{true positive detections}}{\text{true positive test instances}}$$

Source: [J. Johnson](#)

All detections sorted by score



All ground-truth boxes



Evaluating object detectors

1. Run object detector on all test images (with NMS)
2. For each category, compute the **Precision vs. Recall Curve**:

Curve:

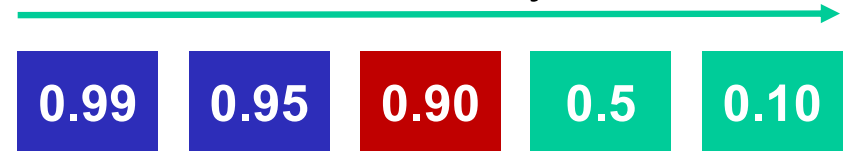
1. For each detection (highest to lowest score)
 1. If it matches some GT box with IoU > 0.5, mark it as positive and eliminate the GT
 2. Otherwise mark it as negative
 3. Plot a point on PR Curve:

$$\text{Precision} = \frac{\text{true positive detections}}{\text{total detections so far}}$$

$$\text{Recall} = \frac{\text{true positive detections}}{\text{true positive test instances}}$$

Source: [J. Johnson](#)

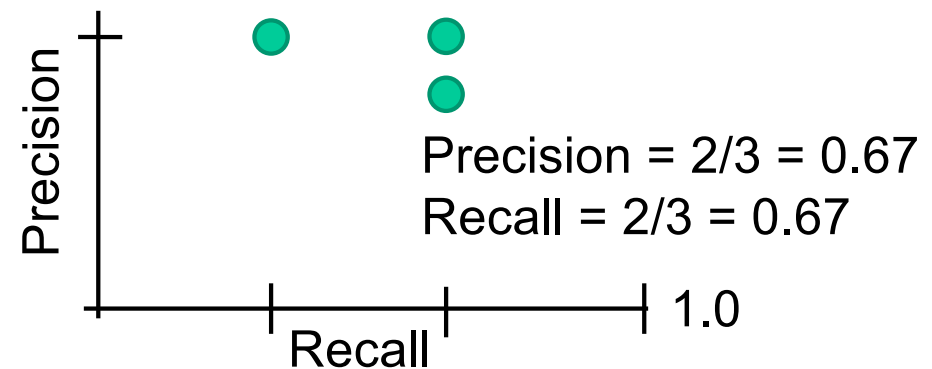
All detections sorted by score



No match > 0.5 IoU with GT



All ground-truth boxes



Evaluating object detectors

1. Run object detector on all test images (with NMS)
2. For each category, compute the **Precision vs. Recall Curve**:

Curve:

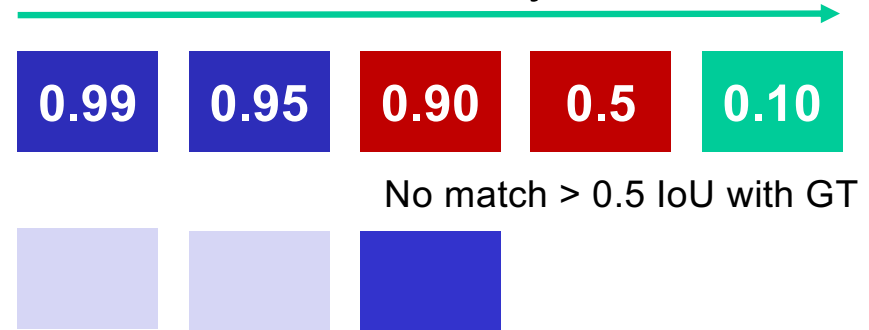
1. For each detection (highest to lowest score)
 1. If it matches some GT box with IoU > 0.5, mark it as positive and eliminate the GT
 2. Otherwise mark it as negative
 3. Plot a point on PR Curve:

$$\text{Precision} = \frac{\text{true positive detections}}{\text{total detections so far}}$$

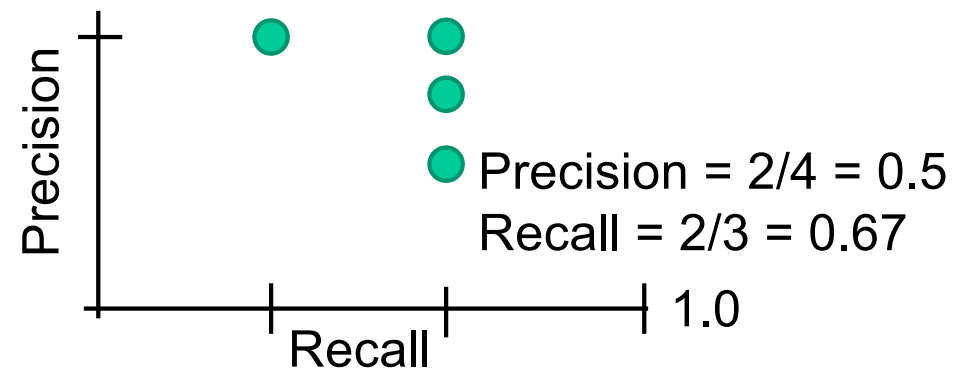
$$\text{Recall} = \frac{\text{true positive detections}}{\text{true positive test instances}}$$

Source: [J. Johnson](#)

All detections sorted by score



All ground-truth boxes



Evaluating object detectors

1. Run object detector on all test images (with NMS)
2. For each category, compute the **Precision vs. Recall Curve**:

Curve:

1. For each detection (highest to lowest score)
 1. If it matches some GT box with IoU > 0.5, mark it as positive and eliminate the GT
 2. Otherwise mark it as negative
 3. Plot a point on PR Curve:

$$\text{Precision} = \frac{\text{true positive detections}}{\text{total detections so far}}$$

$$\text{Recall} = \frac{\text{true positive detections}}{\text{true positive test instances}}$$

Source: [J. Johnson](#)

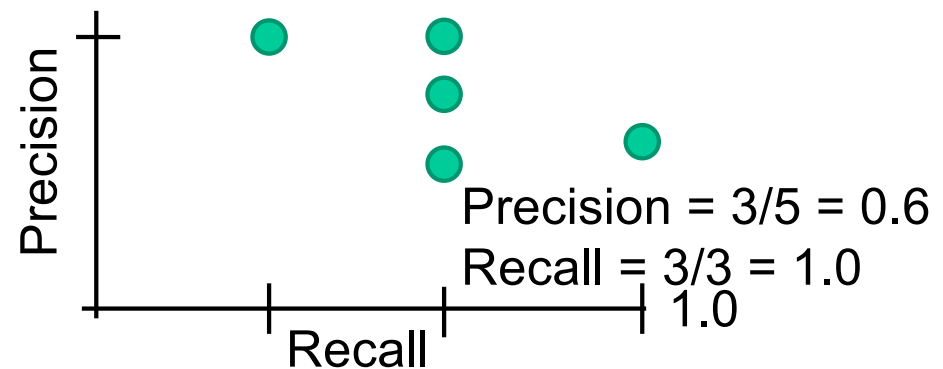
All detections sorted by score



Match: > 0.5 IoU



All ground-truth boxes



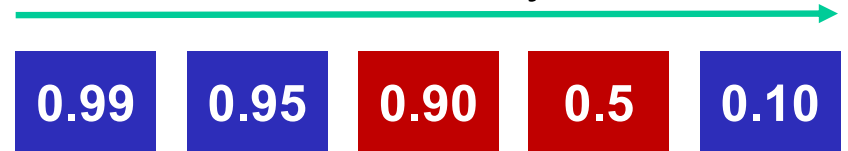
Evaluating object detectors

1. Run object detector on all test images (with NMS)
2. For each category, compute the **Precision vs. Recall Curve**:

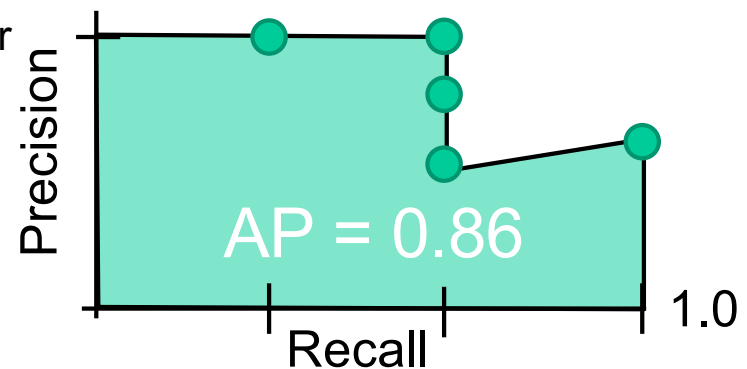
Curve:

1. For each detection (highest to lowest score)
 1. If it matches some GT box with $\text{IoU} > 0.5$, mark it as positive and eliminate the GT
 2. Otherwise mark it as negative
 3. Plot a point on PR Curve
2. Compute **Average Precision (AP)** or area under the Precision vs. Recall Curve:

All detections sorted by score

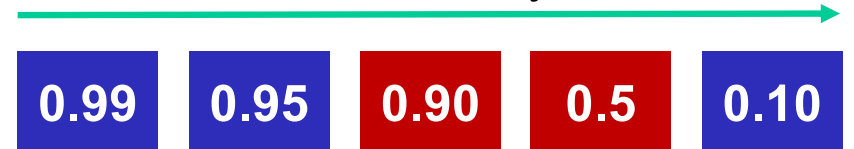


All ground-truth boxes



Evaluating object detectors

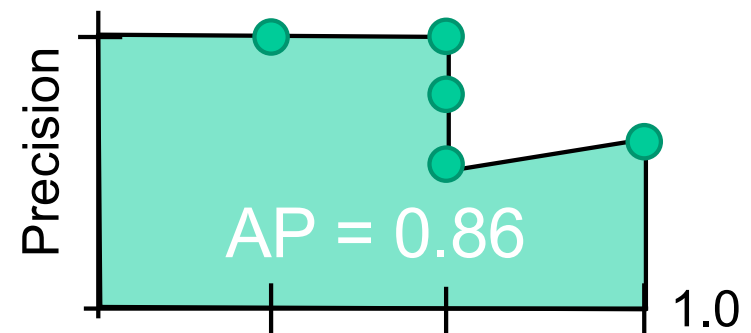
All detections sorted by score



All ground-truth boxes

How to get AP = 1.0?

- Hit all GT boxes with IoU > 0.5, and have no “false positive” detections ranked above any “true positives”



Evaluating object detectors

1. Run object detector on all test images (with NMS)
2. For each category, compute the **Precision vs. Recall Curve**:
 1. For each detection (highest to lowest score)
 1. If it matches some GT box with $\text{IoU} > 0.5$, mark it as positive and eliminate the GT
 2. Otherwise mark it as negative
 3. Plot a point on PR Curve
 2. Compute **Average Precision (AP)** or area under the Precision vs. Recall Curve
3. Average the APs of all categories to obtain **mean AP**, or **mAP**

Source: [J. Johnson](#)

Object detection: Introduction

- Evaluating detectors
 - Intersection over union (IoU)
 - Non-maximum suppression
 - Recall, precision, AP, mAP
- Early datasets

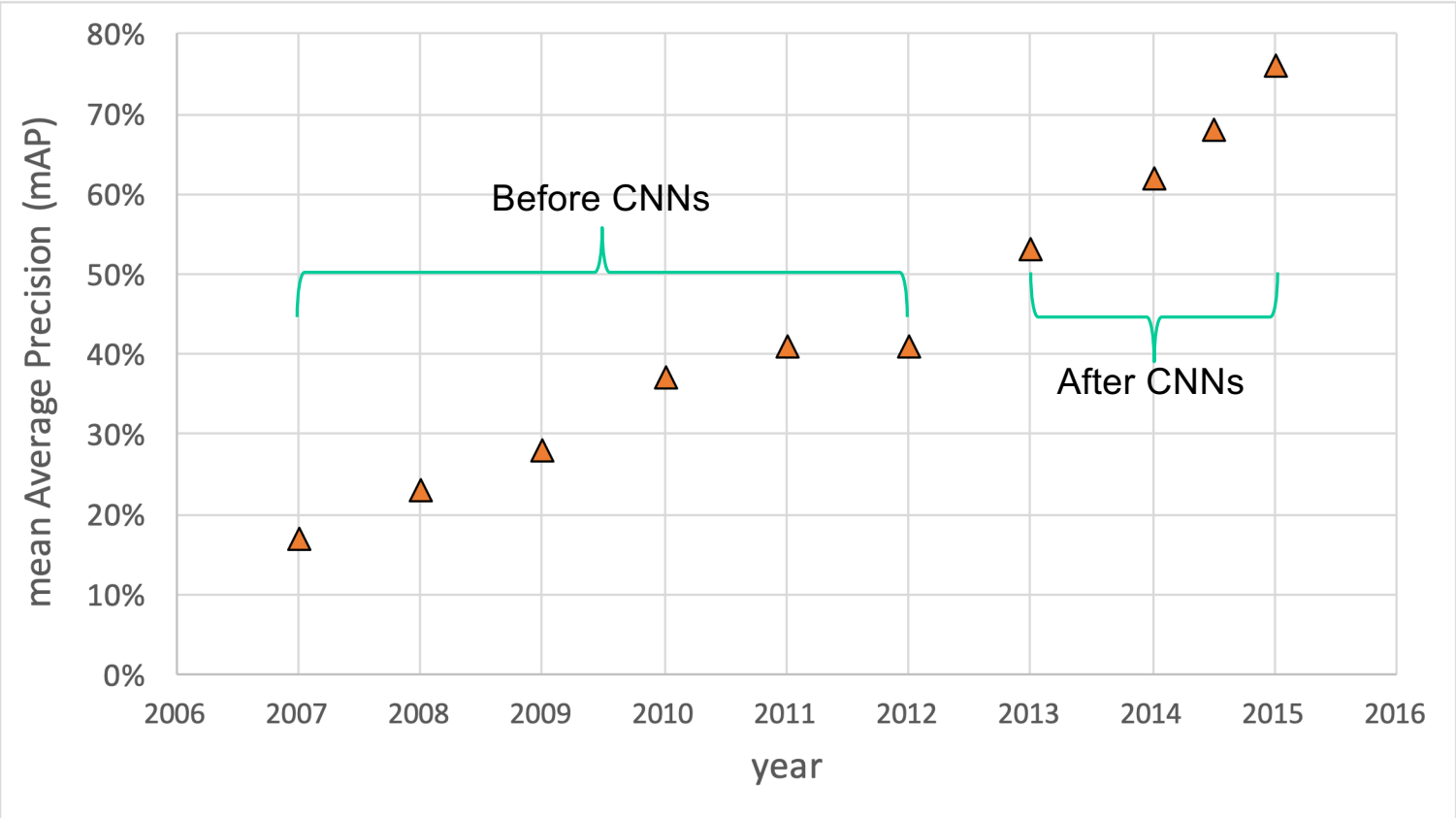
PASCAL VOC Challenge (2005-2012)



- 20 challenge classes:
 - *Person*
 - *Animals*: bird, cat, cow, dog, horse, sheep
 - *Vehicles*: airplane, bicycle, boat, bus, car, motorbike, train
 - *Indoor*: bottle, chair, dining table, potted plant, sofa, tv/monitor
- Dataset size (by 2012): 11.5K training/validation images, 27K bounding boxes, 7K segmentations

<http://host.robots.ox.ac.uk/pascal/VOC/>

Detection progress on PASCAL



More recent benchmark: COCO

What is COCO?



COCO is a large-scale object detection, segmentation, and captioning dataset. COCO has several features:

- ✓ Object segmentation
- ✓ Recognition in context
- ✓ Superpixel stuff segmentation
- ✓ 330K images (>200K labeled)
- ✓ 1.5 million object instances
- ✓ 80 object categories
- ✓ 91 stuff categories
- ✓ 5 captions per image
- ✓ 250,000 people with keypoints

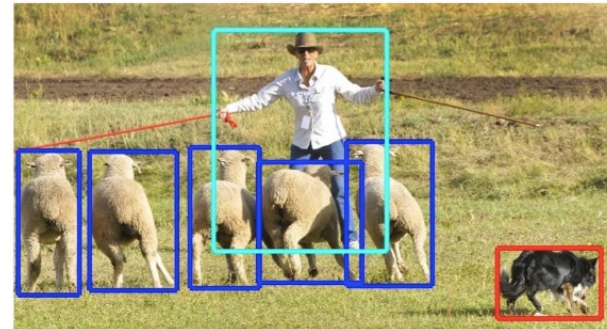


<http://cocodataset.org/#home>

COCO dataset: Tasks



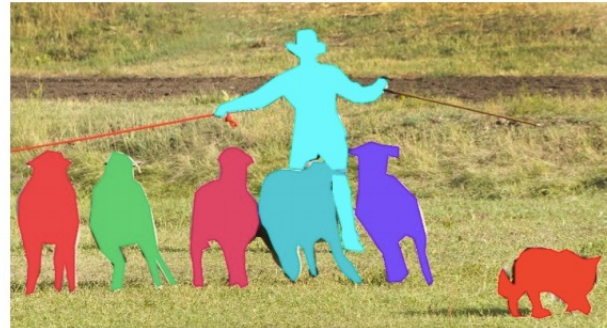
image classification



object detection



semantic segmentation



instance segmentation

- Also: keypoint prediction, captioning, question answering...

COCO detection metrics

Average Precision (AP):

AP % AP at IoU=.50:.05:.95 (primary challenge metric)
AP^{IoU=.50} % AP at IoU=.50 (PASCAL VOC metric)
AP^{IoU=.75} % AP at IoU=.75 (strict metric)

AP Across Scales:

AP^{small} % AP for small objects: area < 32²
AP^{medium} % AP for medium objects: 32² < area < 96²
AP^{large} % AP for large objects: area > 96²

Average Recall (AR):

AR^{max=1} % AR given 1 detection per image
AR^{max=10} % AR given 10 detections per image
AR^{max=100} % AR given 100 detections per image

AR Across Scales:

AR^{small} % AR for small objects: area < 32²
AR^{medium} % AR for medium objects: 32² < area < 96²
AR^{large} % AR for large objects: area > 96²

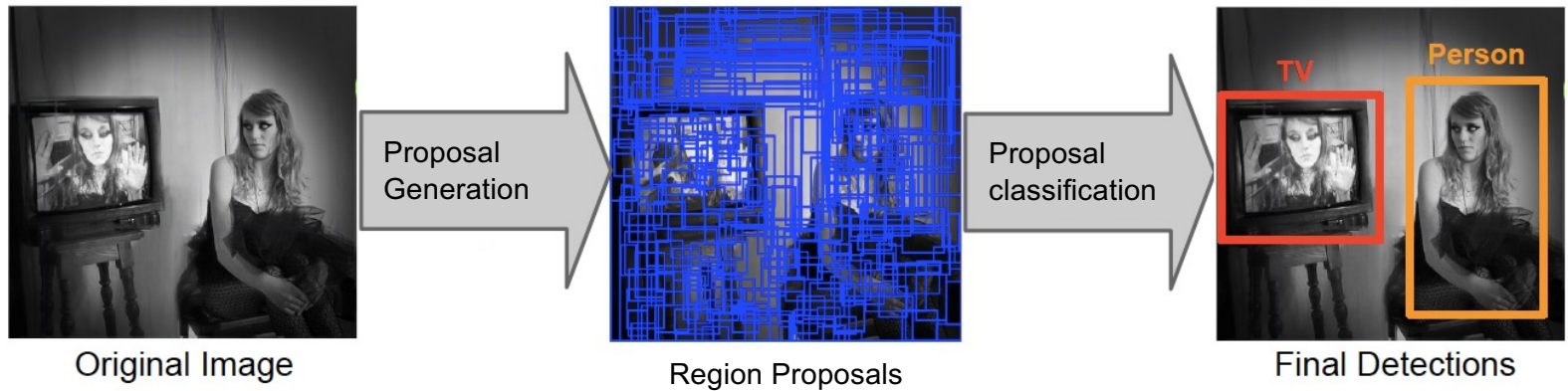
- Leaderboard: <http://cocodataset.org/#detection-leaderboard>
- Not updated since 2020

Object detection: Introduction

- Evaluating detectors
 - Intersection over union (IoU)
 - Non-maximum suppression
 - Recall, precision, AP, mAP
- Early datasets
- Early detector architectures: Two-stage vs. single-stage
- YOLO

Detection architectures: Two-stage vs. single-stage

Two-stage

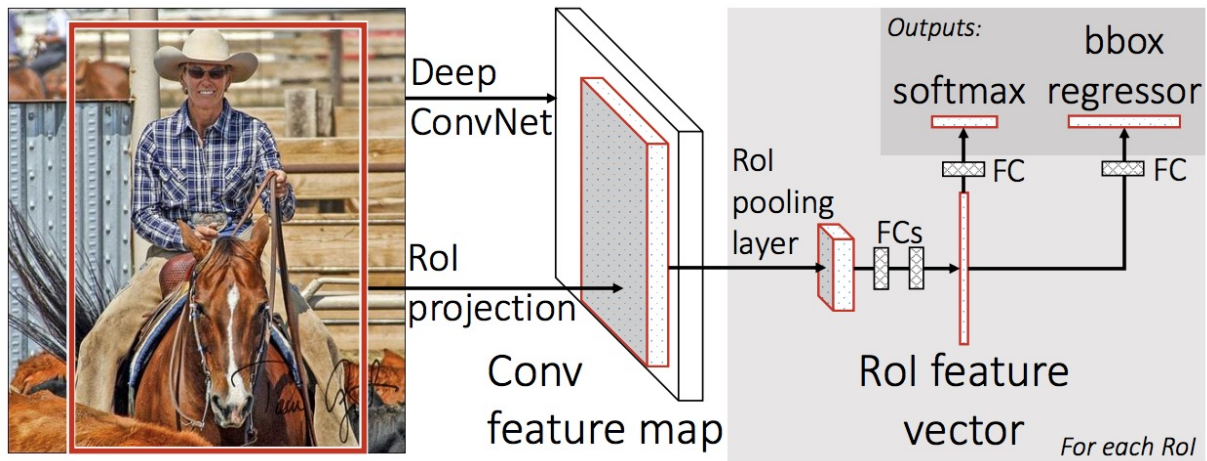


Single-stage

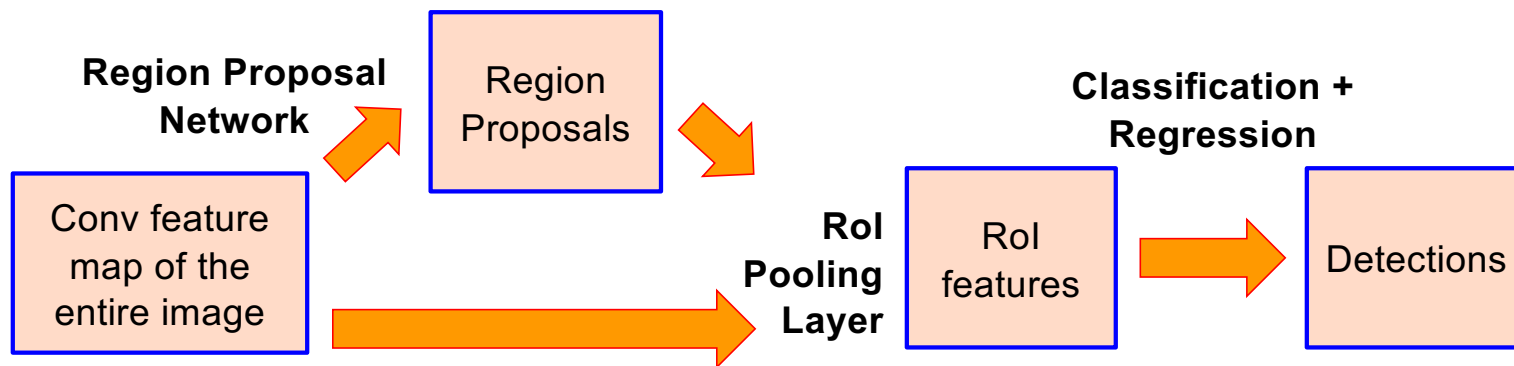


[Image source](#)

Two-stage detectors (R-CNN family)

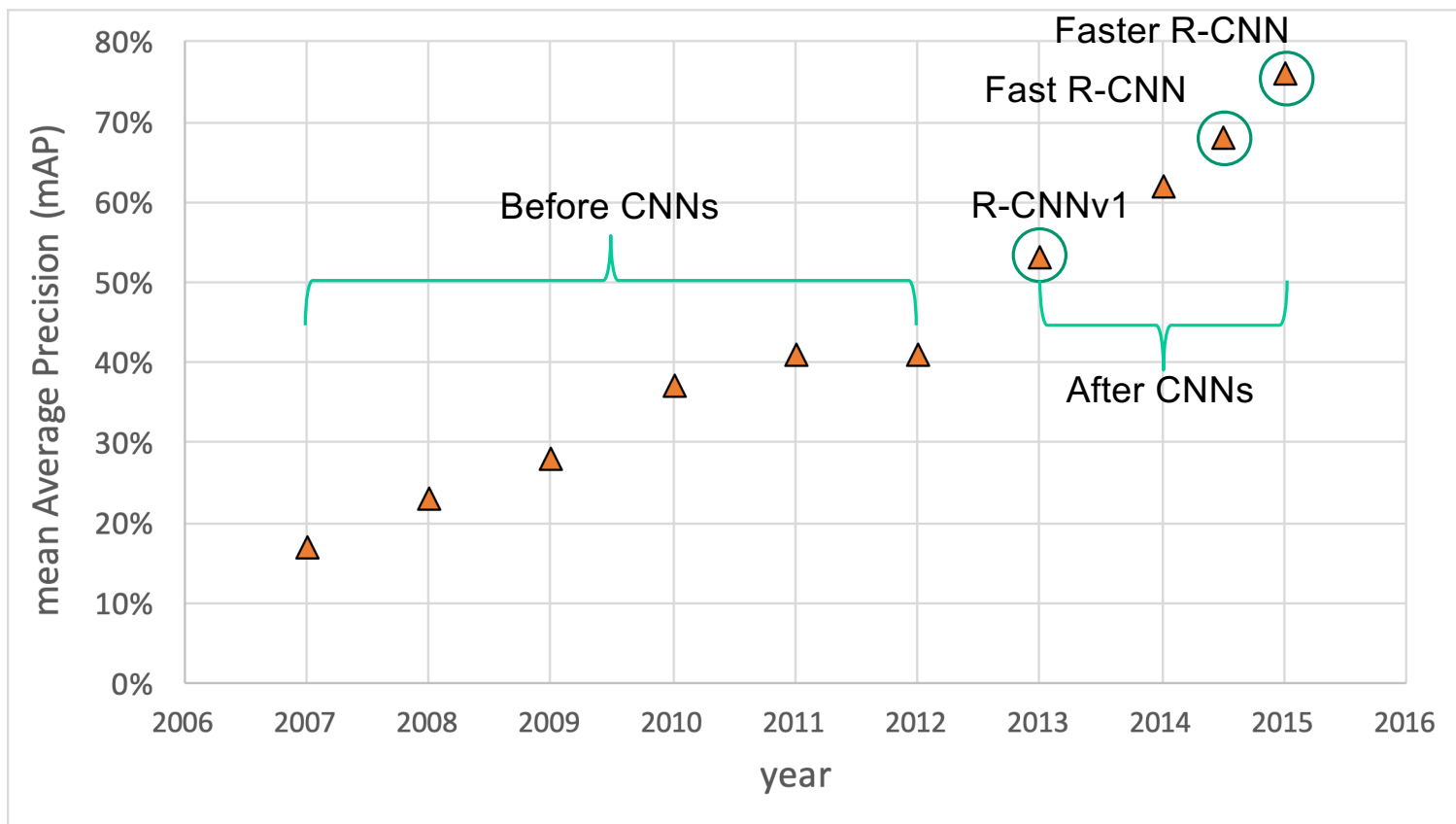


R. Girshick.
[Fast R-CNN](#).
ICCV 2015

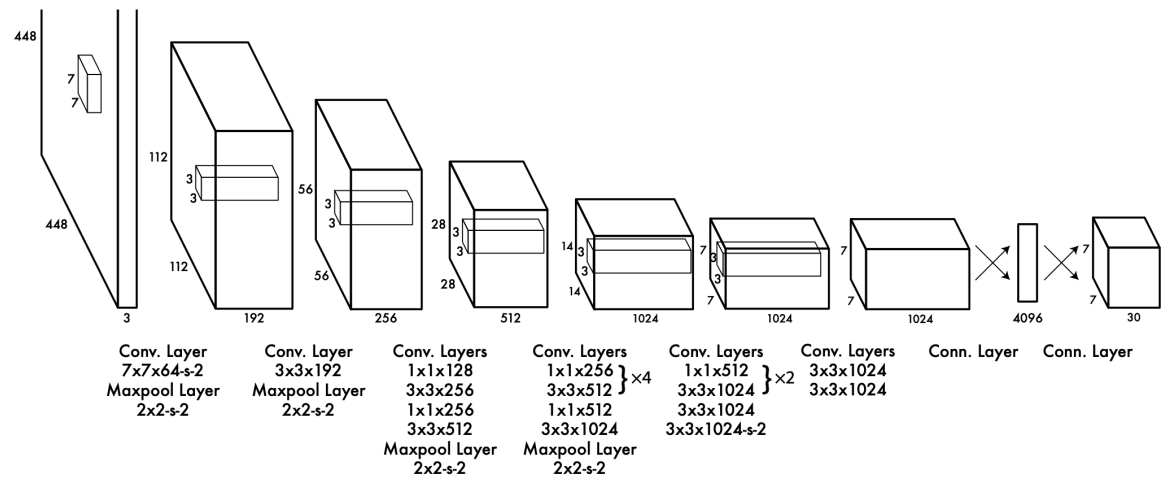
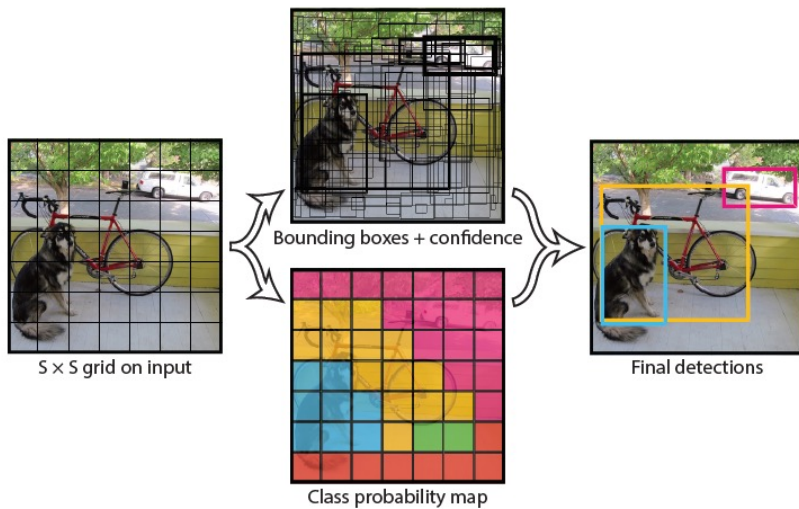
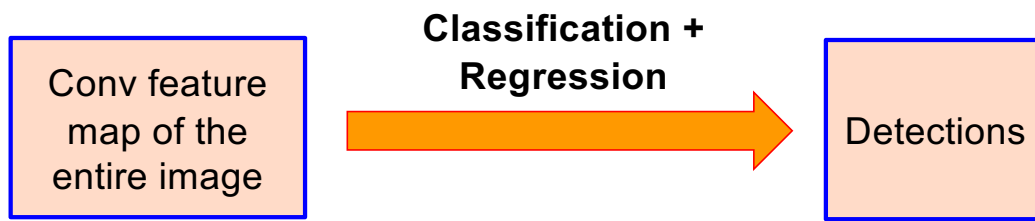


S. Ren et al. [Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks](#). NeurIPS 2015

Detection progress on PASCAL



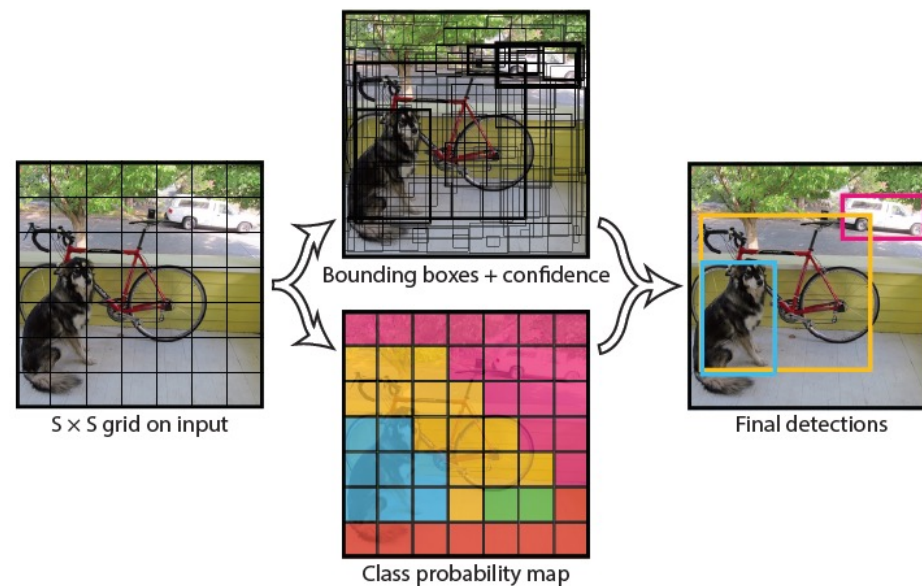
Single-stage detectors (YOLO)



J. Redmon et al. [You Only Look Once: Unified, Real-Time Object Detection](#). CVPR 2016

YOLO

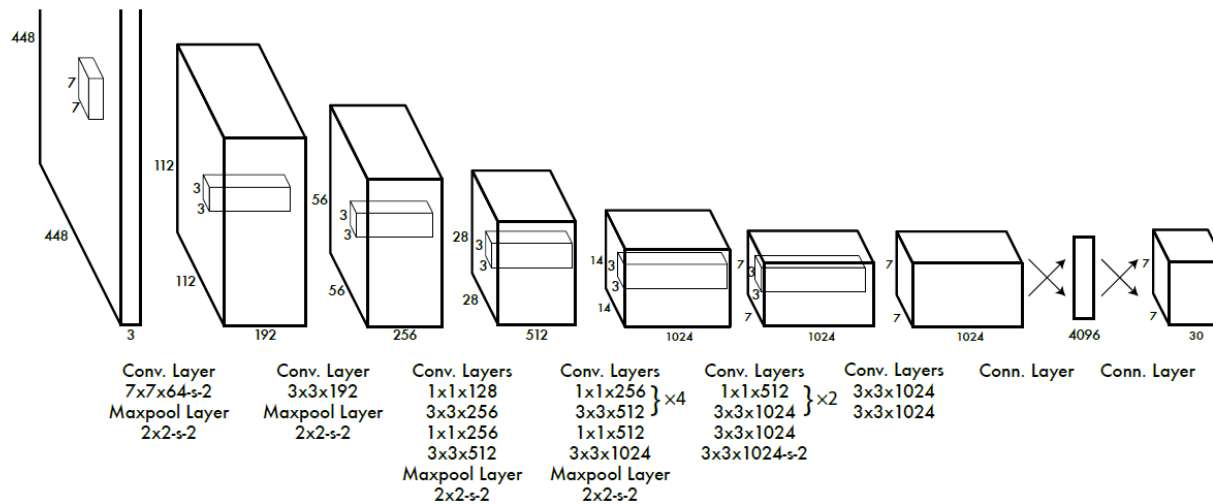
- Divide the image into a coarse grid and directly predict class label and a few candidate boxes for each grid cell



J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, [You Only Look Once: Unified, Real-Time Object Detection](#), CVPR 2016

YOLO

1. Take conv feature maps at 7x7 resolution
2. Add two FC layers to predict, at each location, a score for each class and 2 bboxes w/ confidences
 - For PASCAL, output is $7 \times 7 \times 30$ ($30 = 20 + 2 * (4 + 1)$)



J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, [You Only Look Once: Unified, Real-Time Object Detection](#), CVPR 2016

YOLO loss

$$\begin{aligned} & \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\ & + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \\ & + \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} (C_i - \hat{C}_i)^2 \\ & + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{noobj}} (C_i - \hat{C}_i)^2 \\ & + \sum_{i=0}^{S^2} \mathbb{1}_i^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2 \end{aligned}$$

Regression

Object/no object confidence

Class prediction

YOLO loss

$$\begin{aligned}
 & \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\
 & + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \\
 & + \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} (C_i - \hat{C}_i)^2 \\
 & + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{noobj}} (C_i - \hat{C}_i)^2 \\
 & + \sum_{i=0}^{S^2} \mathbb{1}_i^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2
 \end{aligned}$$

Cell i contains object, predictor j is responsible for it

Small deviations matter less for larger boxes than for smaller boxes

Confidence for object

Confidence for no object

Class probability

Down-weight loss from boxes that don't contain objects ($\lambda_{\text{noobj}} = 0.5$)

YOLO: Results

- Each grid cell predicts only two boxes and can only have one class – this limits the number of nearby objects that can be predicted
- Localization accuracy suffers compared to Fast(er) R-CNN due to coarser features, errors on small boxes
- 7x speedup over Faster R-CNN (45-155 FPS vs. 7-18 FPS)

