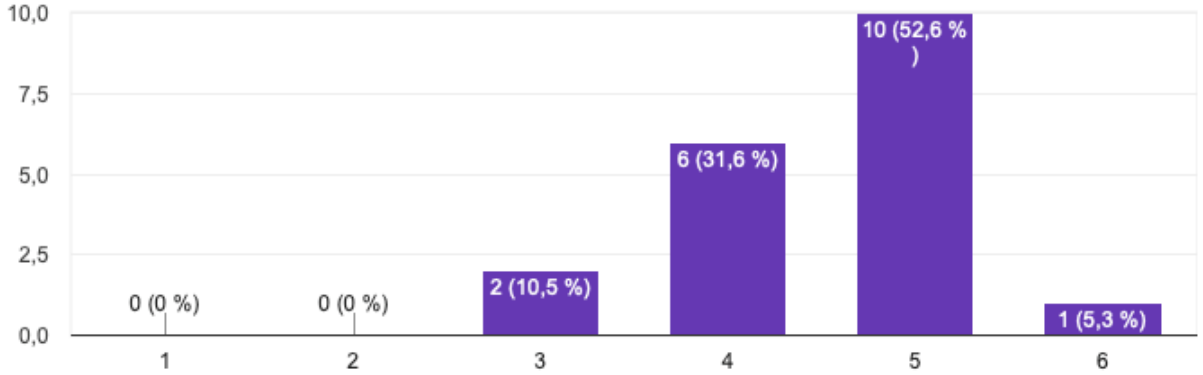
Assignment Lecture 4

- Assignment 2 Feedback
- Final Project
- Object Detection Performance Metrics
- You Only Look Once (YOLO)
  - YOLOv1
  - YOLOv2

Assignment 2 Feedback

With a scale from 1 to 6, 6 being the best, how would you rate the assignment?

19 svar

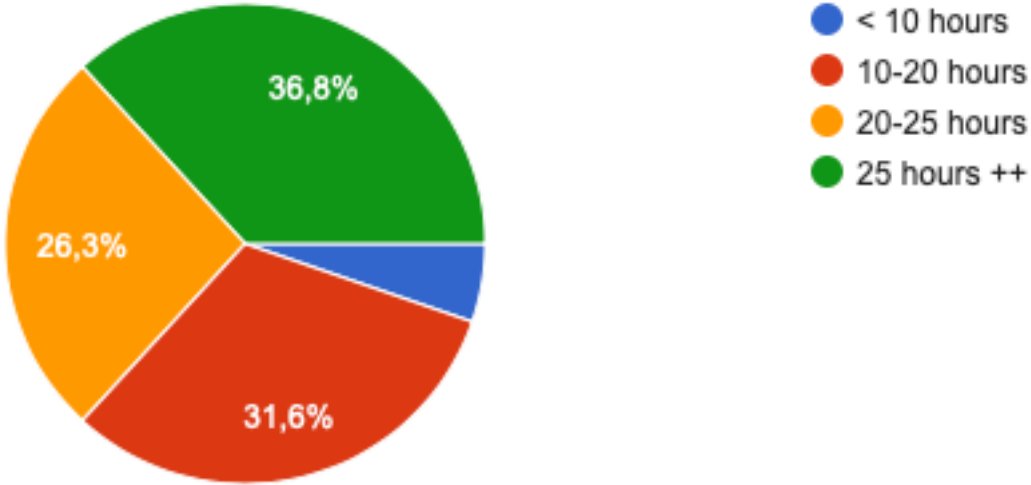


NTNU

TLDR; what was good

- Piazza
- Educational & fun
- Step-by-step improvements
- Tricks of trade
- Good with starter code

Assignment 2 workload



TLDR; what can be improved upon

- Long lines for studass
- Clarity of tasks
- Hard to debug
- Total work load
- Consistency between lectures & assignments
- Gradient input -> hidden was confusing

Final Project

- Will count 16% of your grade

- Will be 3 deliveries:
  - (2 points) Project Proposal
  - (3 points) Presentation Slides Draft
  - (11 points) Final Presentation Slides + Presentation

Project Proposal

- Delivery deadline: 24. March
- Summarize what you want to do in the project

- This is to ensure that the scope of your project is not too large or too small

- Full score if done properly and delivered.

Presentation Slides Draft

- Delivery Deadline: 7. April
- Early draft of your slides

- Should include goal of project, relevant literature, what dataset you are using, chosen architecture, current results++

- To ensure that you start early on your project

- Full score if done properly and delivered

Final Presentation

- Present your project. Will count 11% of your grade
- Date: 28/29 April
- 9-11 minutes per group

Project suggestions

- You can choose your own project, as long as it is computer vision related!

- Or, choose between our suggestions:
  - Image-to-Image translation with GANs
  - Reinforcement Learning
  - Object Detection
  - End-to-end learning of autonomous car
  - Medical Image Segmentation
  - Visual Odometry
- More details will come early next week!

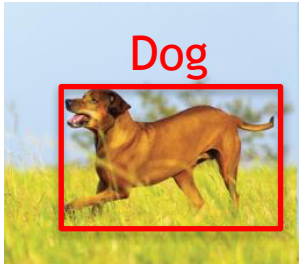
What we have done so far:

Image Classification

Single object in image

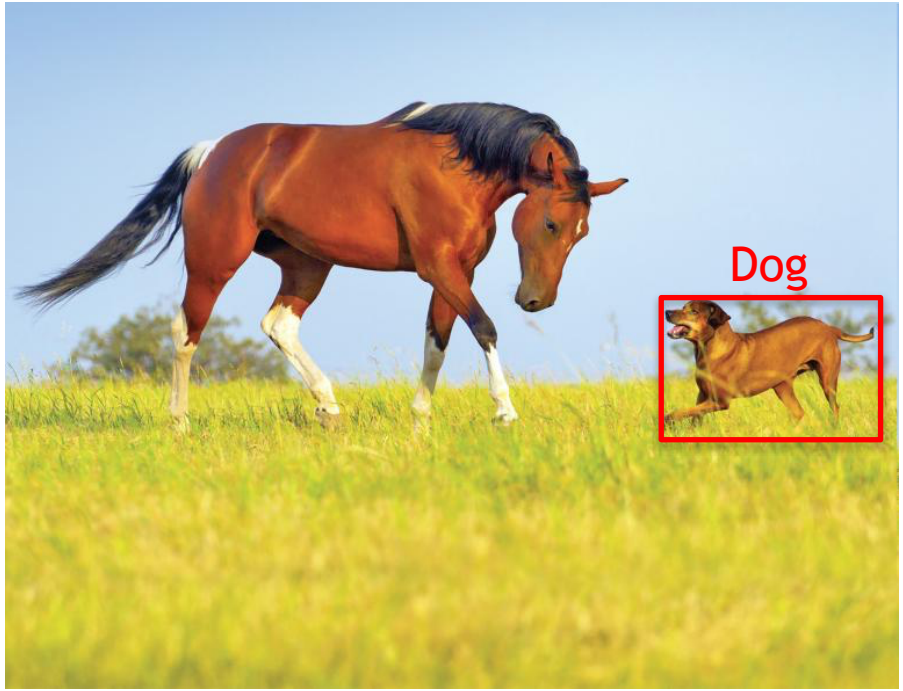
Object Localization

Image Classification + Localization

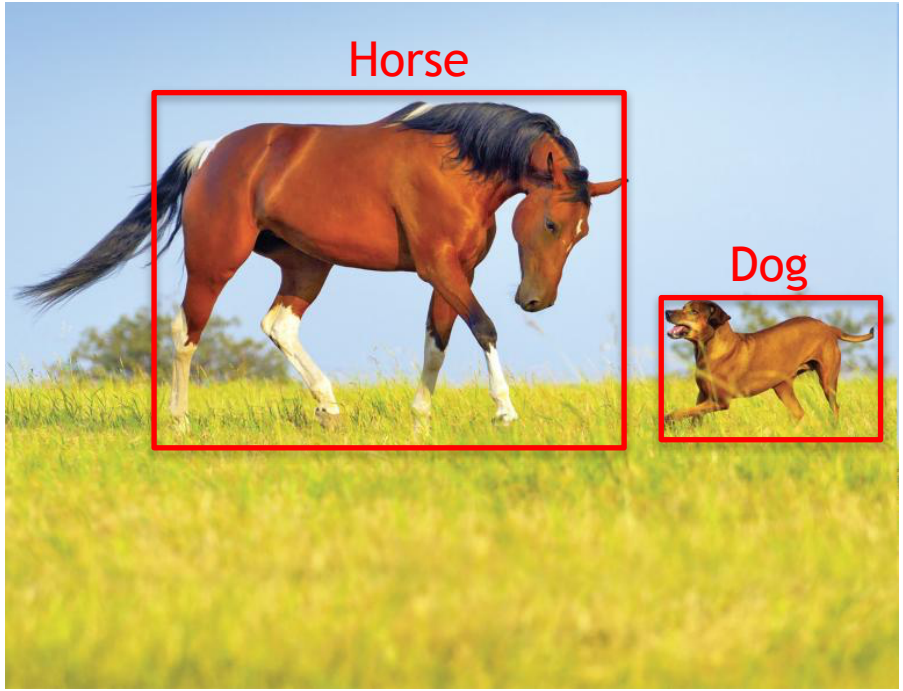


Single object in image

This assignment: Object Detection



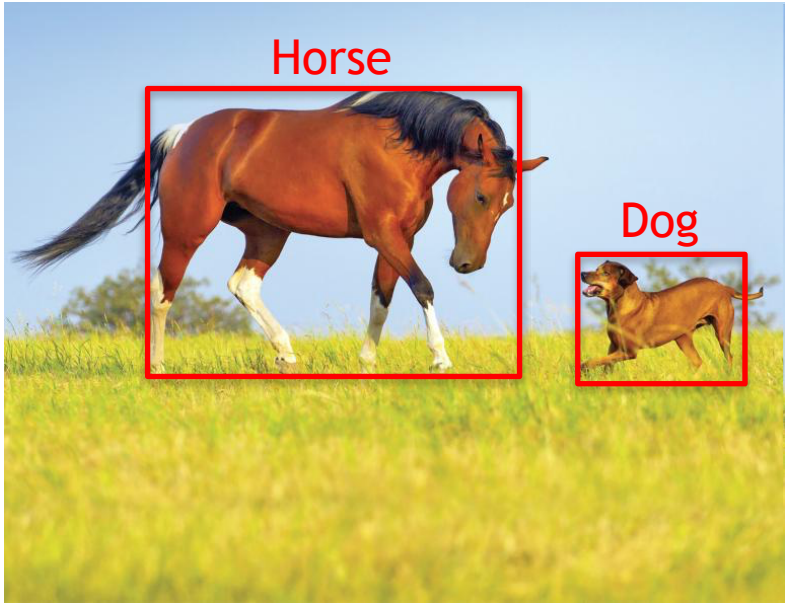
This assignment: Object Detection



One or more object(s) in image

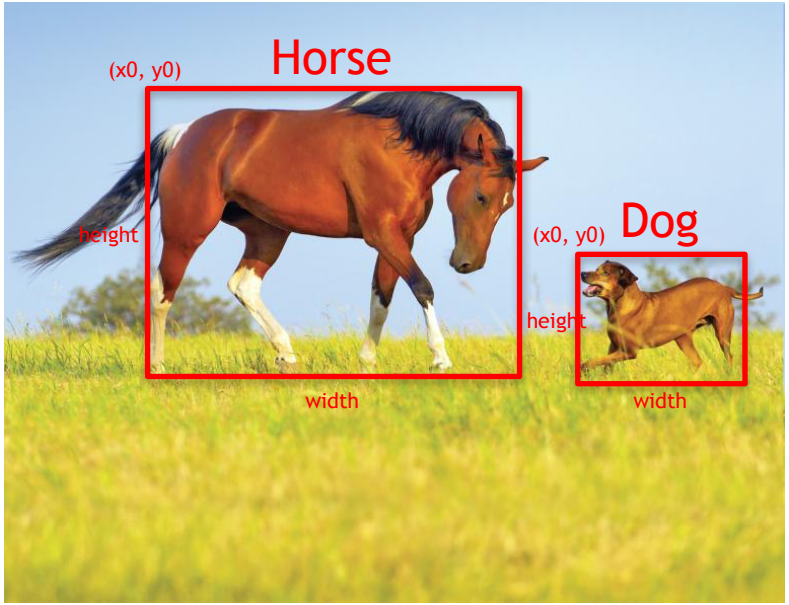
What do we predict?

- Object class (one-hot encoded)



What do we predict?

- Object class (one-hot encoded)
- N bounding boxes
  - x0, y0, width, height
  - x0, y0 is usually top left corner
- Bounding box confidence

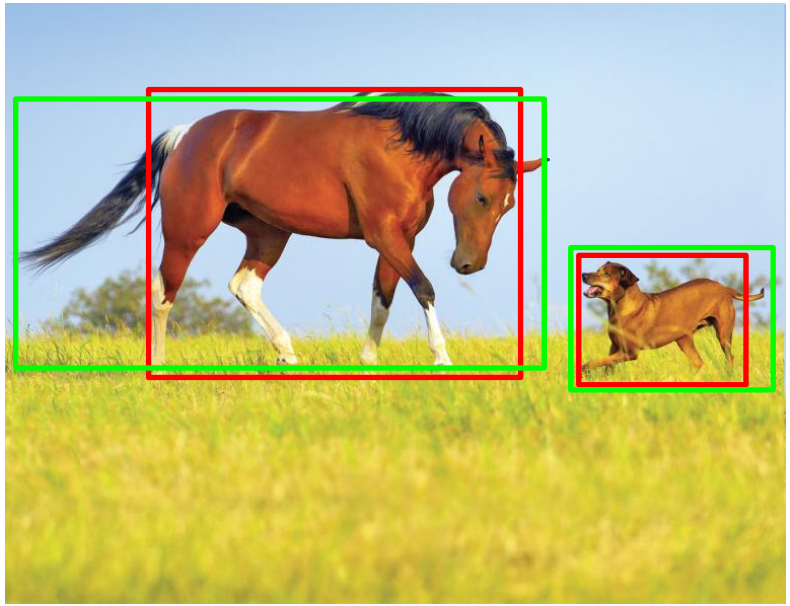


Object Detection Performance Metrics

How to measure the performance?

- Object detection
  - Multiple targets
  - Variable number of targets
- **We need to match bounding boxes and check:**
  - Is the bounding box correct?
  - Is the predicted class correct?

= Ground truth
= Predicted bounding box



How to find bounding box match?

What predicted bounding box corresponds to the ground truth bounding box?

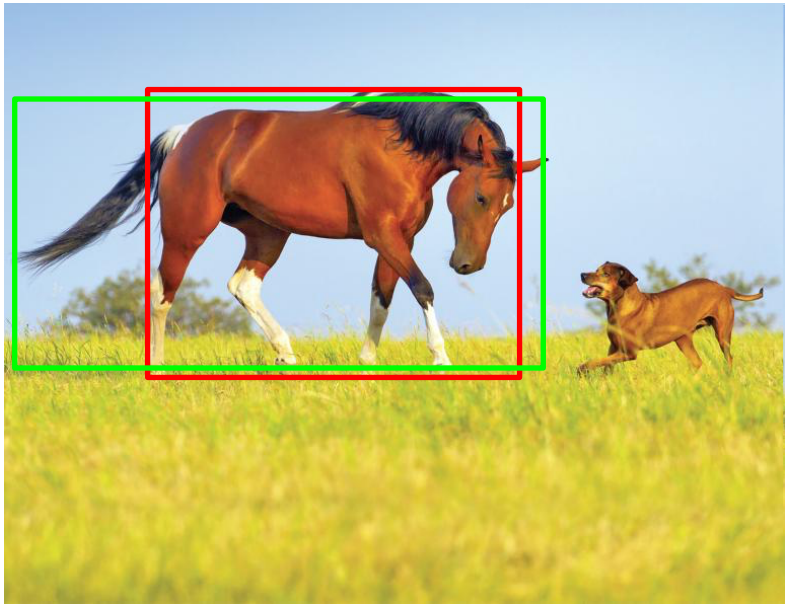
We use Intersection over Union! (IoU)

Intersection over Union (IoU)

$$IoU = \frac{Intersection}{Union}$$

$$0 <= IoU <= 1$$

= Ground truth
= Predicted bounding box



Intersection over Union (IoU)

= Intersection
= Union

union $= area_{bbox1} + area_{bbox2} - intersection$

$$IoU = \frac{Intersection}{Union}$$

$$0 <= IoU <= 1$$

Intersection over Union (IoU)

= Intersection
= Union

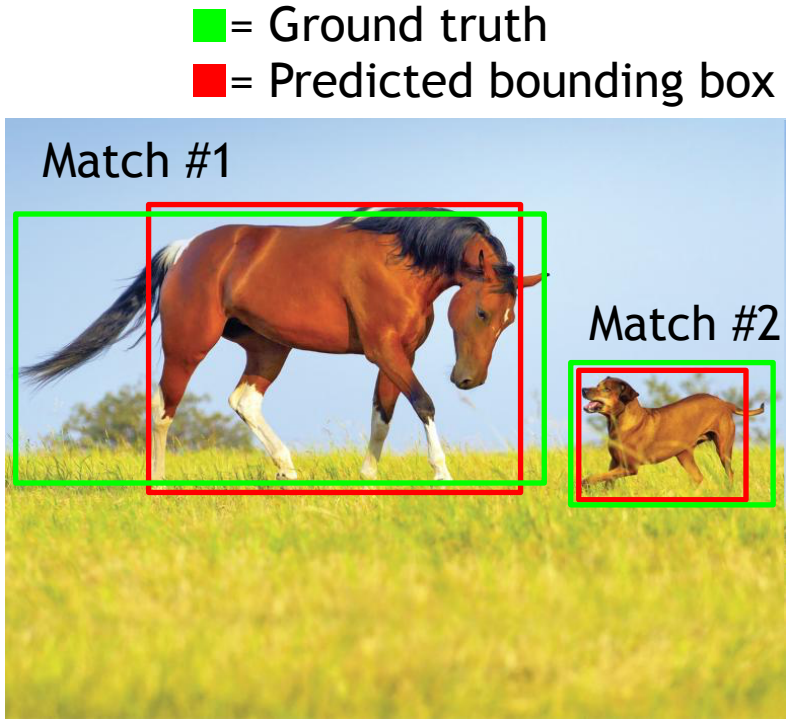
$$IoU = \frac{Intersection}{Union}$$

$$0 <= IoU <= 1$$

Usually a hit if IoU >= 0.5

Matching of bounding boxes

- Given a set of predicted & ground truth bounding boxes..
  - We match the bounding boxes with the highest IoU ratio

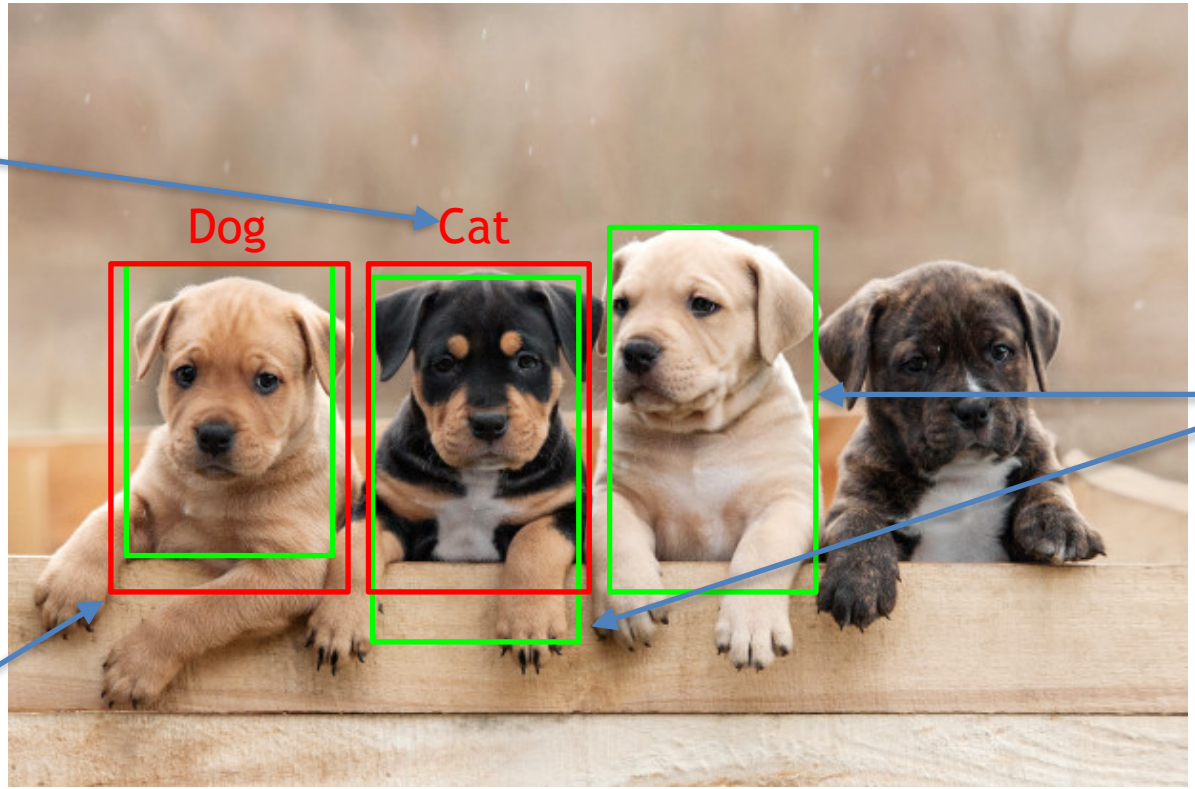


Bounding Box Match

- The match is a…
  - **True Positive** if a proposal was made for class c and there was an object of class c
  - **False Positive** if a proposal was made for class c, but there is no object of class c.
  - **False Negative** if no proposal was made for class c, but there is an object of class c.

TP, FP, FN

= Ground truth
= Predicted bounding box



Performance Metrics: Precision & Recall

- Precision:
  - How many of the predicted boxes are correct

$$\text{Precision} = \frac{TP}{TP + FP}$$

TP=True Positive
FP=False Positive
FN=False Negative

- Recall:
  - Out of all the ground truth boxes, how many are we able to detect

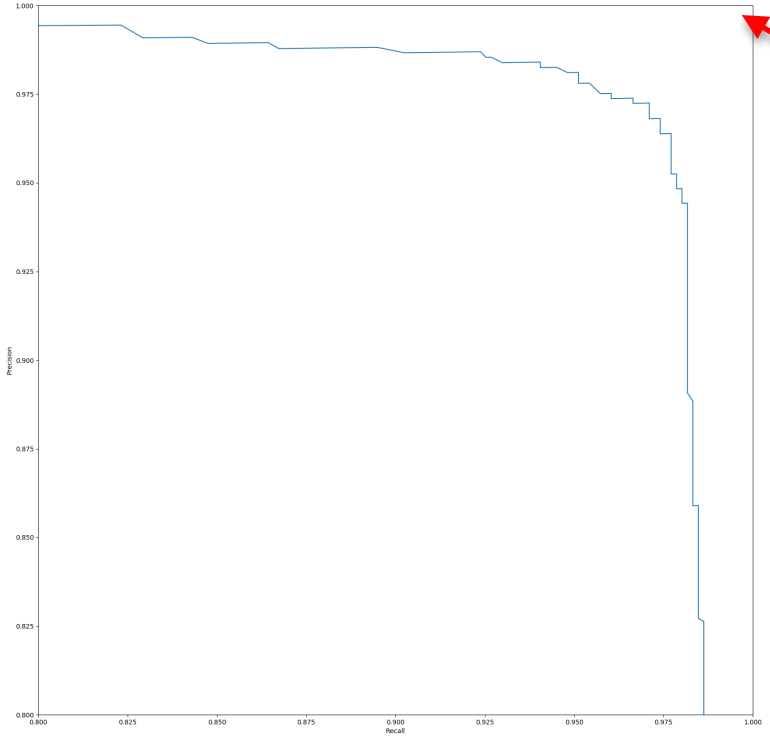
$$\text{Recall} = \frac{TP}{TP + FN}$$

Precision-Recall Curve (Single class)

- For all possible confidence thresholds t:
  - Find all predictions with a confidence >= t
  - Calculate precision & recall for these predictions

Precision-Recall Curve (Single class)

Precision



Recall

The closer top-right corner, the better the predictor

Precision-Recall curve is easy to read

- … but, we desire a single number to measure the performance
  - (e.g: classification accuracy for image classification)

Standard metric to use is **Average Precision**

Average Precision(AP)

- Formal definition:

$$\text{AP} = \frac{1}{11} \sum_{r \in 0, 0.1, \ldots, 1.0} \max_{\hat{r}; \hat{r} \geq r} p(\hat{r})$$

- r: recall
- p(r) = precision at recall level r
  (from precision-recall curve)

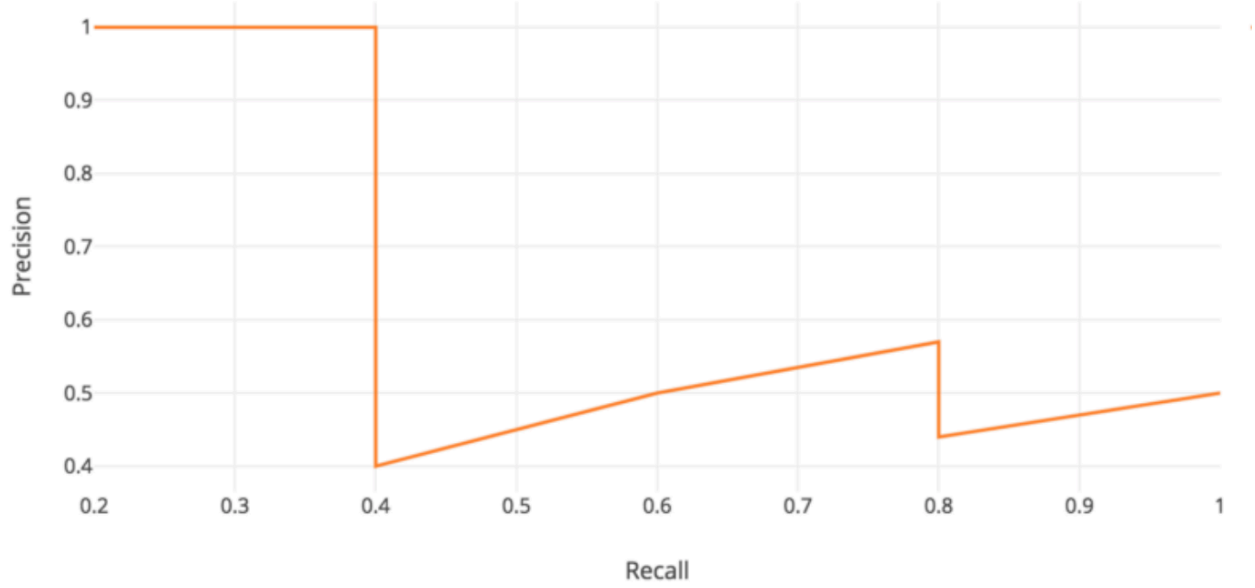
Average Precision(AP) - Pseudo Code

- Given a precision-recall curve, the AP is:

- avg_precision = 0
- for *recall level* **r** in [0.0, 0.1, .02,… 1.0]
  - find largest *precision* **p** with recall value >= **r**
  - avg_precision += p
- avg_precision = avg_precision / 11

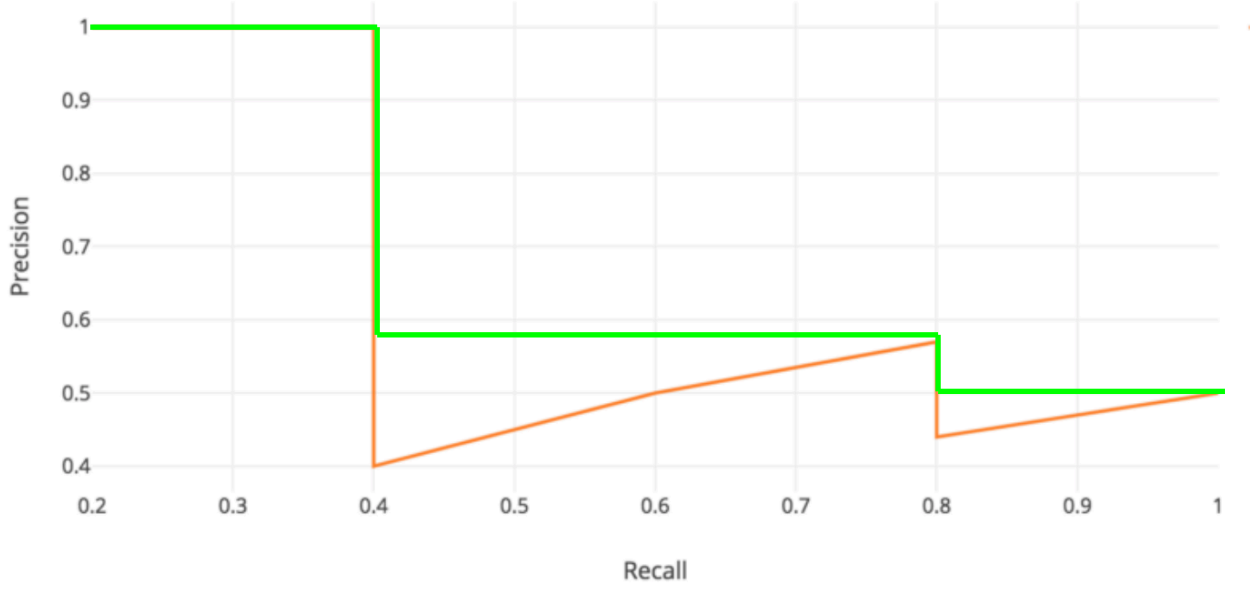
Average Precision(AP)

- What is AP doing?



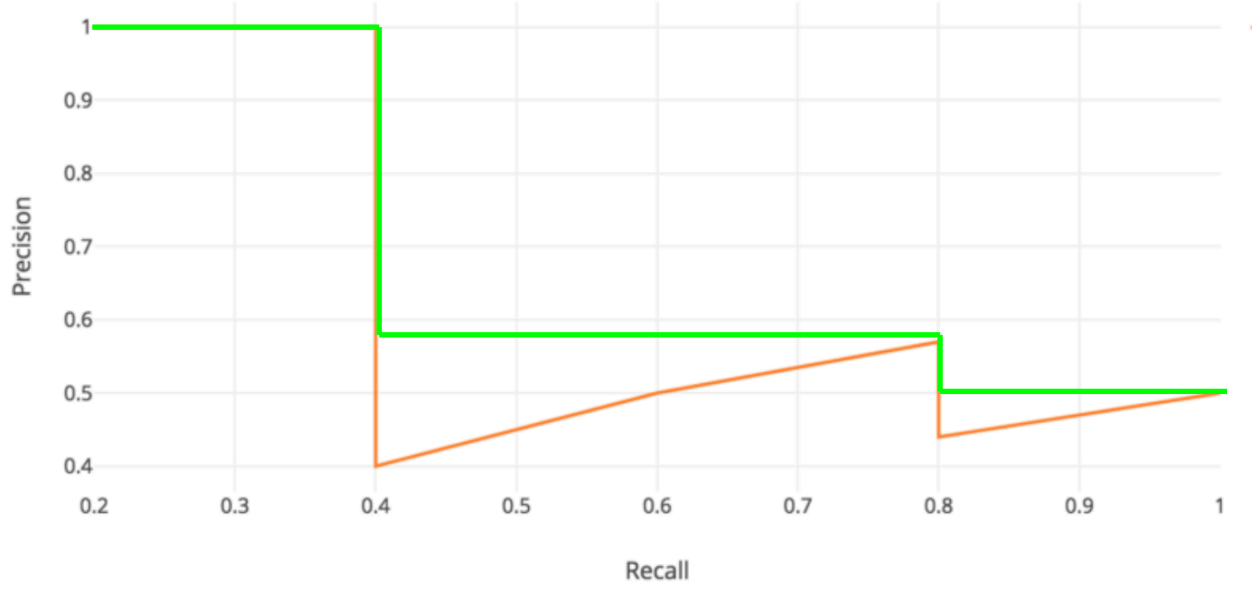
Average Precision(AP)

- What is AP doing? **Smooths the zig-zag pattern**



Average Precision(AP)

- Why? To **reduce the impact** of wiggles/noise in predictions



Average Precision(AP) Example

Calculating the AP given this precision-recall curve:

We iterate over all recall levels from [0.0, 0.1, 0.2… 1.0]

Rank	Correct?	Precision	Recall
1	True	1.0	0.2
2	True	1.0	0.4
3	False	0.67	0.4
4	False	0.5	0.4
5	False	0.4	0.4
6	True	0.5	0.6
7	True	0.57	0.8
8	False	0.5	0.8
9	False	0.44	0.8
10	True	0.5	1.0

Recall	Max
0	
0.1	
0.2	
0.3	
0.4	
0.5	
0.6	
0.7	
0.8	
0.9	
1.0	

Example Source

Average Precision(AP) Example

Calculating the AP given this precision-recall curve:

We iterate over all recall levels from [0.0, 0.1, 0.2… 1.0]

recall= 1.0

$$AP = \frac{0.5}{11}$$

Rank	Correct?	Precision	Recall
1	True	1.0	0.2
2	True	1.0	0.4
3	False	0.67	0.4
4	False	0.5	0.4
5	False	0.4	0.4
6	True	0.5	0.6
7	True	0.57	0.8
8	False	0.5	0.8
9	False	0.44	0.8
10	True	0.5	1.0

Recall	Max
0	
0.1	
0.2	
0.3	
0.4	
0.5	
0.6	
0.7	
0.8	
0.9	
1.0	0.5

Example Source

Average Precision(AP) Example

Calculating the AP given this precision-recall curve:

We iterate over all recall levels from [0.0, 0.1, 0.2… 1.0]

recall= 0.9

$$AP = \frac{0.5 \times 2}{11}$$

Rank	Correct?	Precision	Recall
1	True	1.0	0.2
2	True	1.0	0.4
3	False	0.67	0.4
4	False	0.5	0.4
5	False	0.4	0.4
6	True	0.5	0.6
7	True	0.57	0.8
8	False	0.5	0.8
9	False	0.44	0.8
10	True	0.5	1.0

Recall	Max
0	
0.1	
0.2	
0.3	
0.4	
0.5	
0.6	
0.7	
0.8	
0.9	0.5
1.0	0.5

Example Source

Average Precision(AP) Example

Calculating the AP given this precision-recall curve:

We iterate over all recall levels from [0.0, 0.1, 0.2… 1.0]

recall= 0.8

$$AP = \frac{0.5 \times 2 + 0.57}{11}$$

Rank	Correct?	Precision	Recall
1	True	1.0	0.2
2	True	1.0	0.4
3	False	0.67	0.4
4	False	0.5	0.4
5	False	0.4	0.4
6	True	0.5	0.6
7	True	0.57	0.8
8	False	0.5	0.8
9	False	0.44	0.8
10	True	0.5	1.0

Recall	Max
0	
0.1	
0.2	
0.3	
0.4	
0.5	
0.6	
0.7	
0.8	0.57
0.9	0.5
1.0	0.5

Example Source

Average Precision(AP) Example

Calculating the AP given this precision-recall curve:

We iterate over all recall levels from [0.0, 0.1, 0.2… 1.0]

recall= … 0.5

$$AP = \frac{0.5 \times 2 + 0.57 \times 4}{11}$$

Rank	Correct?	Precision	Recall
1	True	1.0	0.2
2	True	1.0	0.4
3	False	0.67	0.4
4	False	0.5	0.4
5	False	0.4	0.4
6	True	0.5	0.6
7	True	0.57	0.8
8	False	0.5	0.8
9	False	0.44	0.8
10	True	0.5	1.0

Recall	Max
0	
0.1	
0.2	
0.3	
0.4	
0.5	0.57
0.6	0.57
0.7	0.57
0.8	0.57
0.9	0.5
1.0	0.5

Example Source

Average Precision(AP) Example

Calculating the AP given this precision-recall curve:

We iterate over all recall levels from [0.0, 0.1, 0.2… 1.0]

recall= 0.4

$$AP = \frac{0.5 \times 2 + 0.57 \times 4 + 1.0}{11}$$

Rank	Correct?	Precision	Recall
1	True	1.0	0.2
2	True	1.0	0.4
3	False	0.67	0.4
4	False	0.5	0.4
5	False	0.4	0.4
6	True	0.5	0.6
7	True	0.57	0.8
8	False	0.5	0.8
9	False	0.44	0.8
10	True	0.5	1.0

Recall	Max
0	
0.1	
0.2	
0.3	
0.4	1.0
0.5	0.57
0.6	0.57
0.7	0.57
0.8	0.57
0.9	0.5
1.0	0.5

Example Source

Average Precision(AP) Example

Calculating the AP given this precision-recall curve:

We iterate over all recall levels from [0.0, 0.1, 0.2… 1.0]

recall= 0.0

$$AP = \frac{0.5 \times 2 + 0.57 \times 4 + 1.0 \times 5}{11}$$

Rank	Correct?	Precision	Recall
1	True	1.0	0.2
2	True	1.0	0.4
3	False	0.67	0.4
4	False	0.5	0.4
5	False	0.4	0.4
6	True	0.5	0.6
7	True	0.57	0.8
8	False	0.5	0.8
9	False	0.44	0.8
10	True	0.5	1.0

Recall	Max
0	1.0
0.1	1.0
0.2	1.0
0.3	1.0
0.4	1.0
0.5	0.57
0.6	0.57
0.7	0.57
0.8	0.57
0.9	0.5
1.0	0.5

Example Source

Mean Average Precision(mAP)

- Average precision works only for a single class
- mAP is the mean average precision over all K classes

$$\text{mAP} = \frac{1}{K} \sum_{c}^{K} AP_c$$

Mean Average Precision Recap

1. Match predicted bounding boxes & ground truth bounding boxes
   - Match the bounding-boxes with the highest IoU rate
   - Ignore any match with IoU rate < 0.5
   - Maximum 1 match per bounding box
2. Find the precision-recall curve for all possible confidence thresholds
   - In task2 we give you the confidence thresholds to use
3. Calculate the average precision for each class
   - Use 11 recall levels: [0.0, 0.1, …, 1.0]

mAP in papers

Average Precision for different IoU thresholds

	backbone	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}
Faster R-CNN+++ [19]	ResNet-101-C4	34.9	55.7	37.4
Faster R-CNN w FPN [27]	ResNet-101-FPN	36.2	59.1	39.0
Faster R-CNN by G-RMI [21]	Inception-ResNet-v2 [41]	34.7	55.5	36.7
Faster R-CNN w TDM [39]	Inception-ResNet-v2-TDM	36.8	57.7	39.2
Faster R-CNN, RoIAlign	ResNet-101-FPN	37.3	59.6	40.3
Mask R-CNN	ResNet-101-FPN	38.2	60.3	41.7
Mask R-CNN	ResNeXt-101-FPN	**39.8**	**62.3**	**43.4**

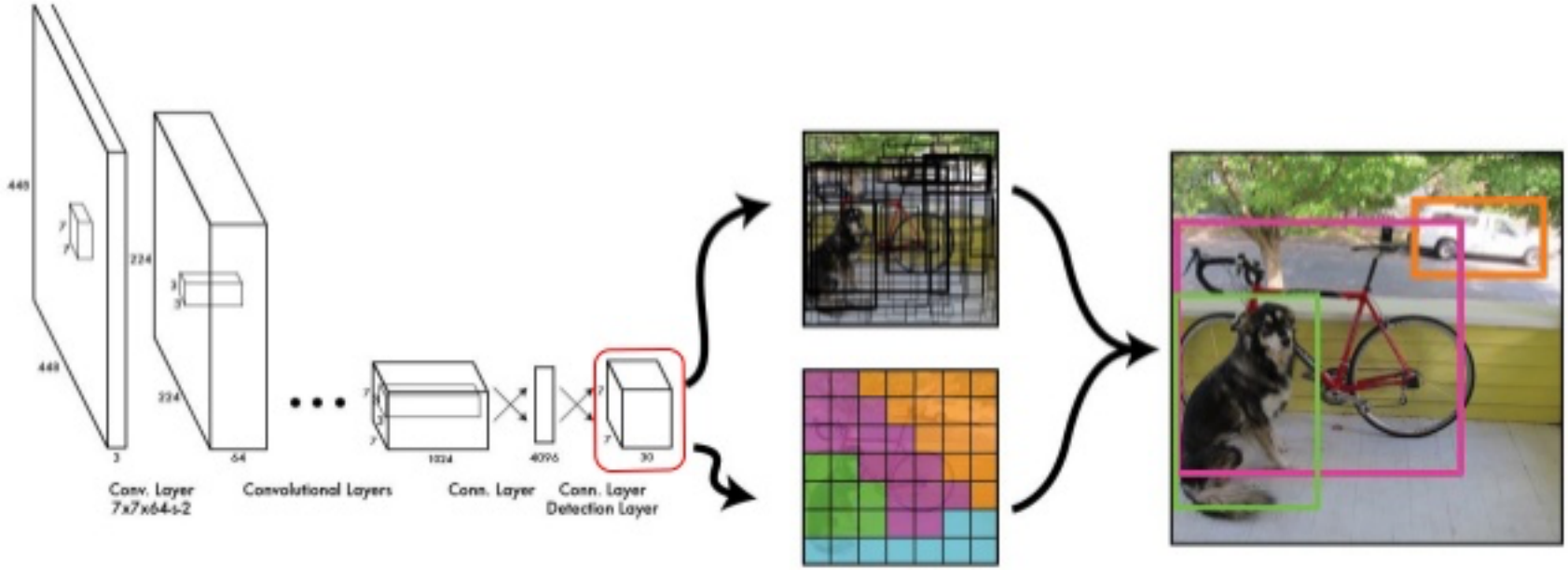
Useful Resources Task 1/2

- http://host.robots.ox.ac.uk/pascal/VOC/pubs/everingham10.pdf - Section 4.2
- https://medium.com/@jonathan_hui/map-mean-average-precision-for-object-detection-45c121a31173

You Only Look Once (YOLO)

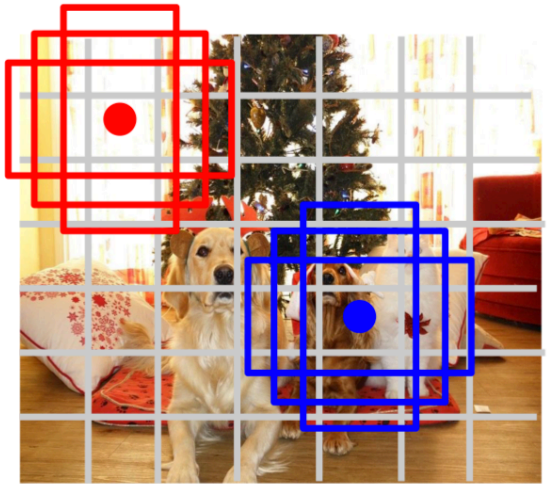
Real-time object detection

YOLOv1


448
448
7
3
224
224
3
3
64
1024
4096
7
7
30
Conv. Layer
7x7x64-2
Convolutional Layers
Conn. Layer
Conn. Layer
Detection Layer

YOLOv1

3xHxW



Divide image into grid
SxS

Each grid cell:

- B base bounding boxes
- Predict probability for all classes C

Output:

SxSx(5xB + C)

YOLOv1

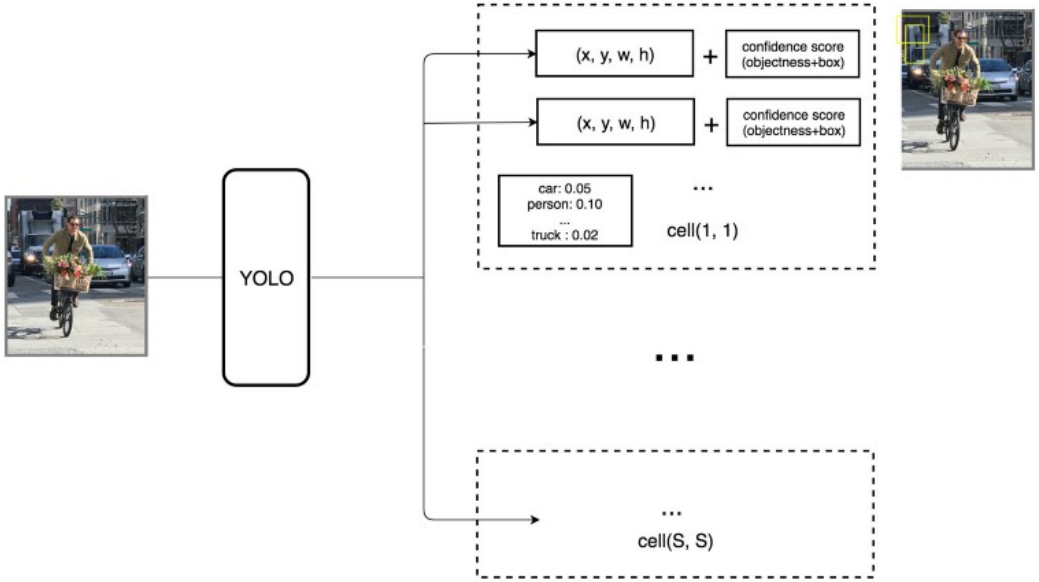


Figure Source

YOLOv1

B bounding boxes per grid cell

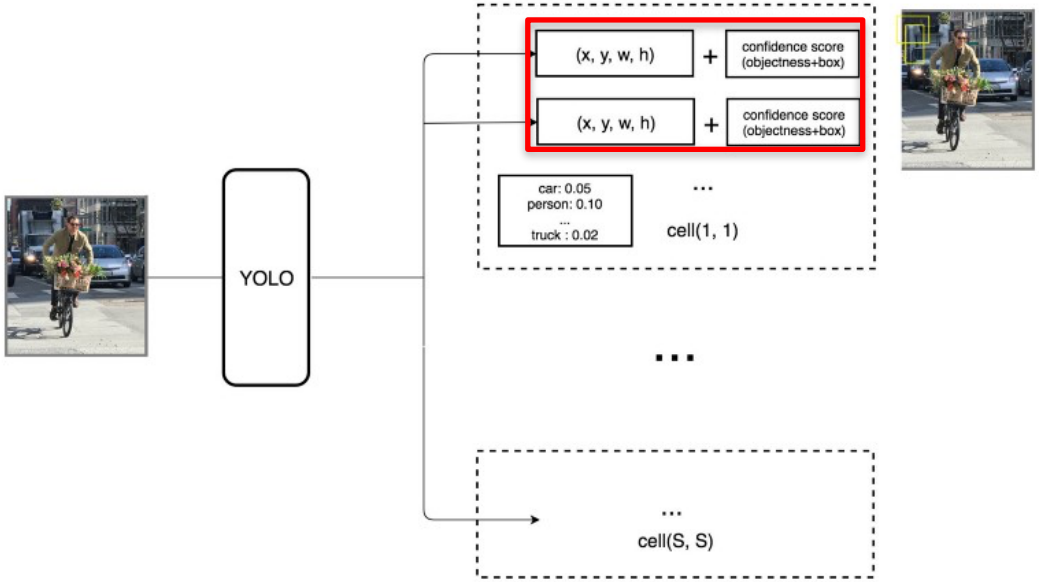


Figure Source

YOLOv1

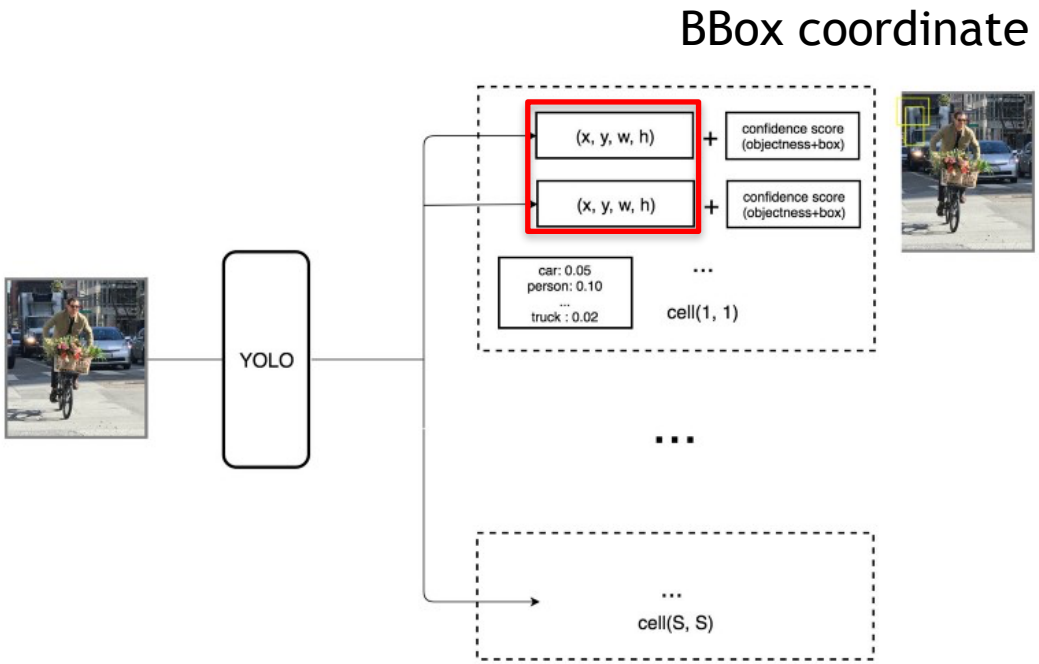


Figure Source

YOLOv1

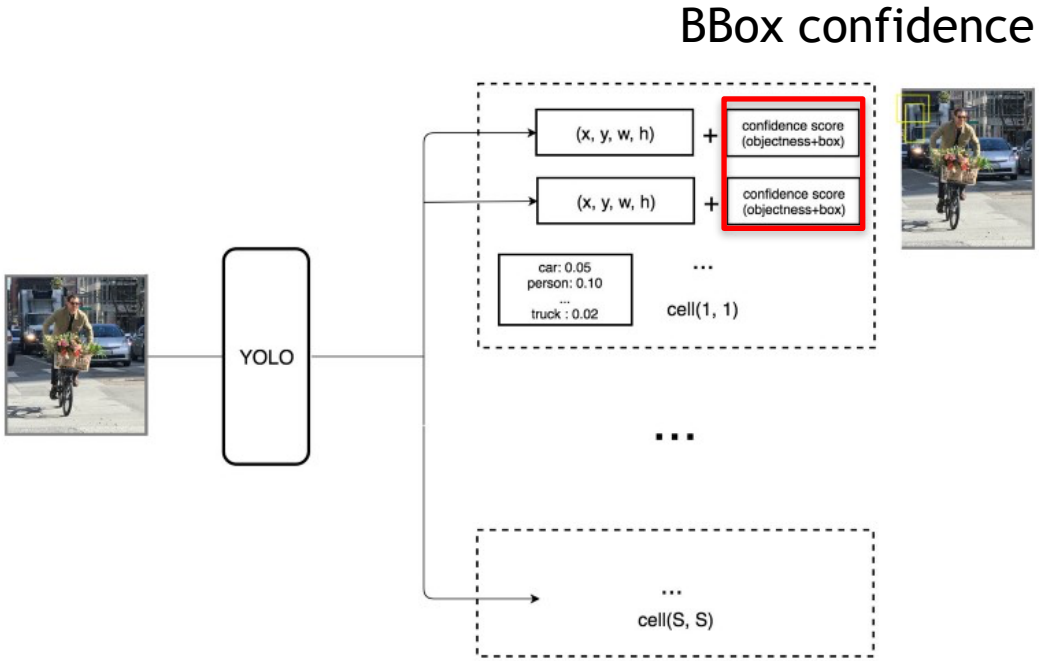
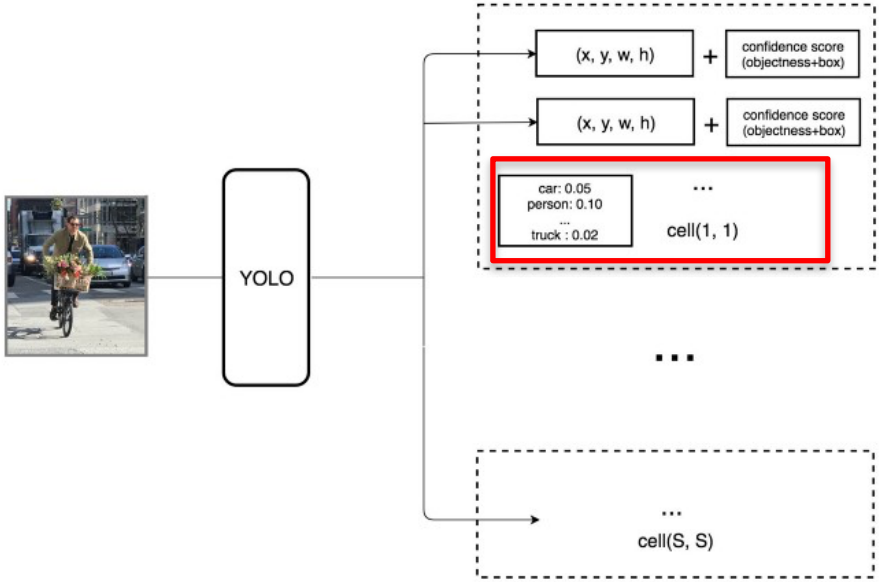


Figure Source

YOLOv1



1 predicted class per grid cell

Figure Source

CNN Output Processing

CNN Output Processing

- We want to:
  - Remove any predicted bounding box with low confidence
    - Confidence thresholding
  - Remove any predicted bounding box that are overlapping significantly
    - Non-Maximum Suppression

Confidence Thresholding

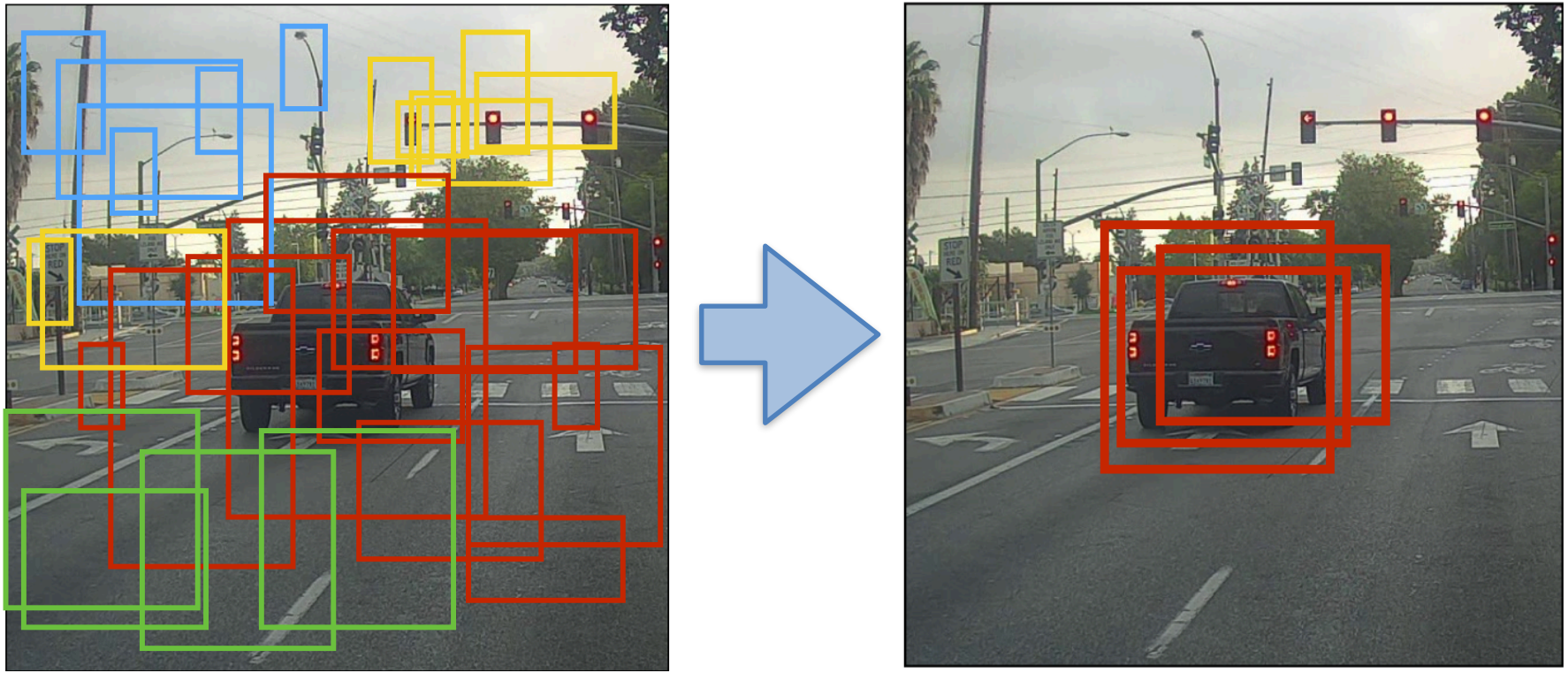
- Remove predictions with low confidence

- Total bounding box confidence:

$$\text{total confidence} = P(\text{box confidence}) * P(\text{class confidence})$$

- If no class has total confidence > threshold:
  - Remove prediction

After Confidence Thresholding



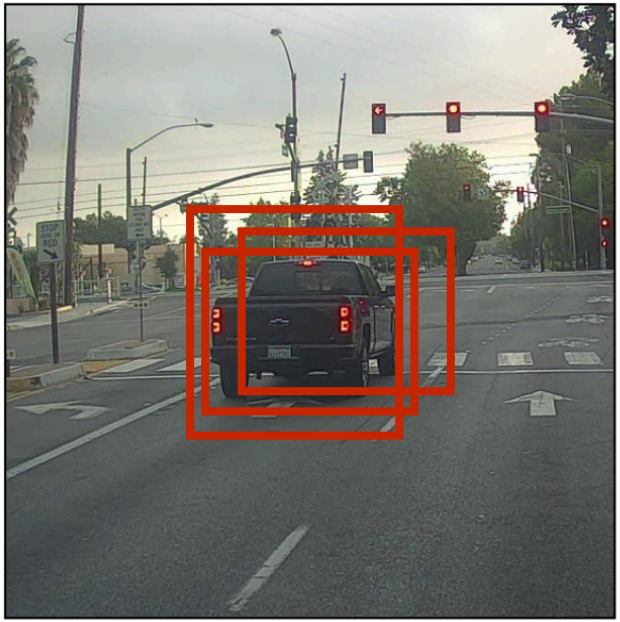
Non-Maximum Suppression

- Remove largely overlapping boxes

- Sort predicted bounding boxes by confidence score (descending)
- Iterate over all predicted bounding boxes:
  - Remove any bounding box where the IoU with the current box is larger than iou_threshold

After Non-Maximum Suppression

Before non-max suppression



Non-Max Suppression

After non-max suppression

The Convolutional Network

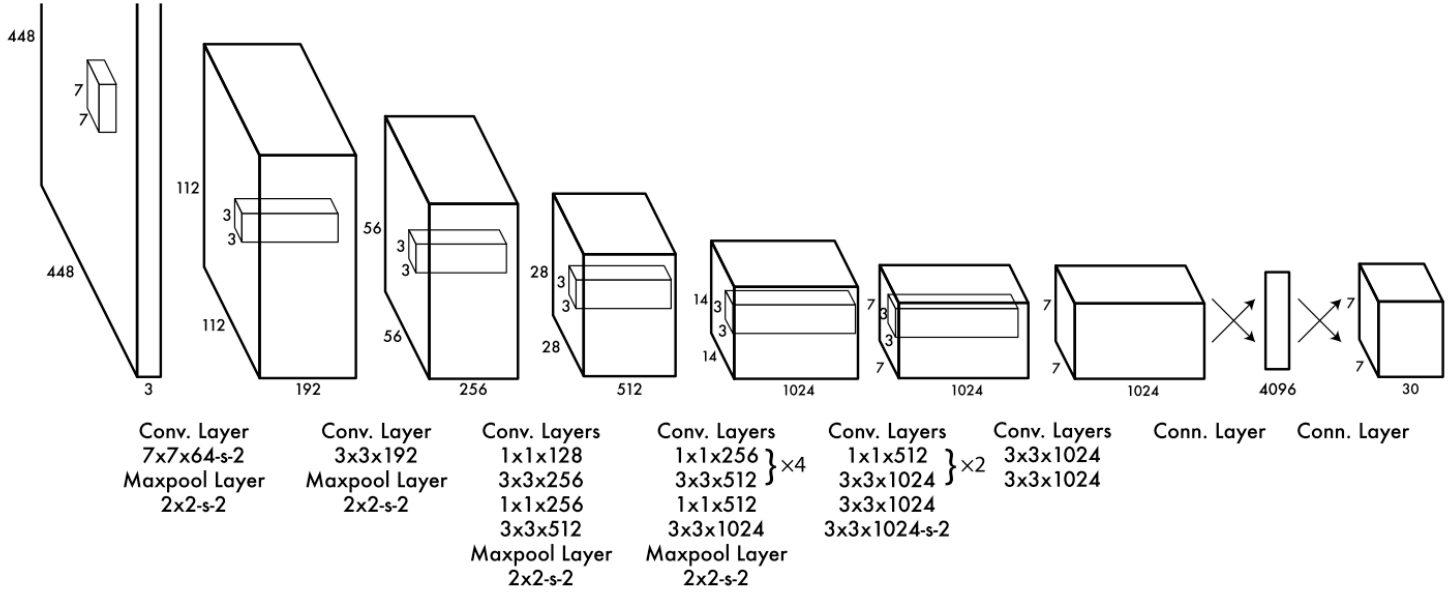


Figure 3: The Architecture. Our detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating 1×1 convolutional layers reduce the features space from preceding layers. We pretrain the convolutional layers on the ImageNet classification task at half the resolution (224×224 input image) and then double the resolution for detection.

End-to-End Learning

- The YOLO loss function enables:
  - End-to-end learning
  - Fast convergence

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

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

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

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

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

End-to-End Learning

Loss for the bounding box coordinate prediction

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

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

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

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

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

End-to-End Learning

Loss for predicted bounding box confidence

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

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

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

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

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

End-to-End Learning

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

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

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

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

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

Classification loss for each grid cell

Results

Real-Time Detectors	Train	mAP	FPS
100Hz DPM [30]	2007	16.0	100
30Hz DPM [30]	2007	26.1	30
Fast YOLO	2007+2012	52.7	**155**
YOLO	2007+2012	**63.4**	45

Table 1: Real-Time Systems on PASCAL VOC 2007. Compar-

Pros

- Fast. Works in real time
- Efficient training with end-to-end learning
- Generalizes well to other domains
- “Human like”

Limitations

- Spatially constrained by grid cells, 1 class per cell.
- Poor generalization to weird shapes/sizes
- Small error in large boxes are treated the same as large errors on small boxes

Assignment Task 3/4

- Theory questions

- You will implement the post-processing pipeline of YOLOv2:
  - Confidence score thresholding
  - Non-Maximum Suppression

YOLOv1

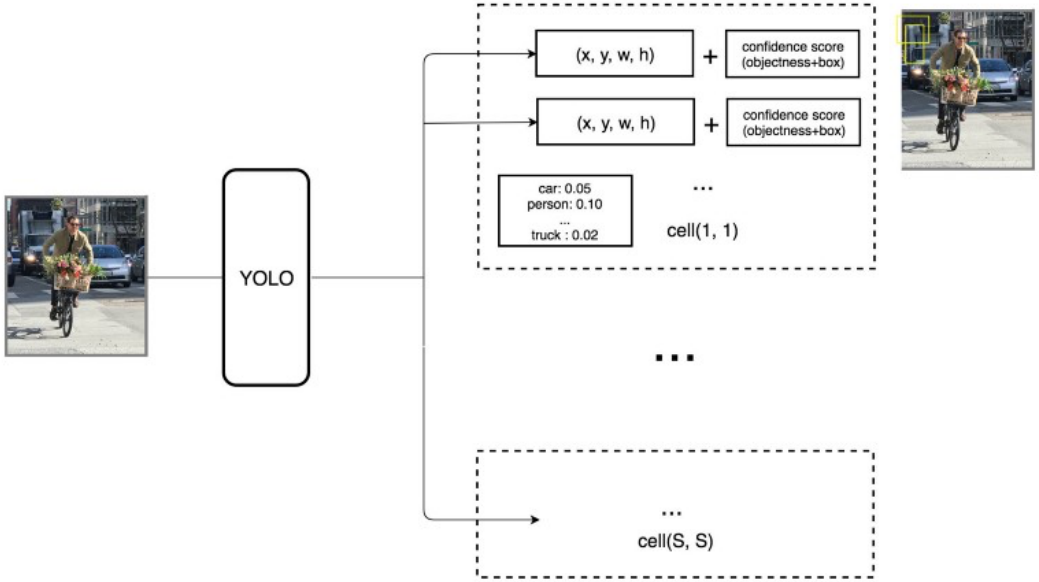
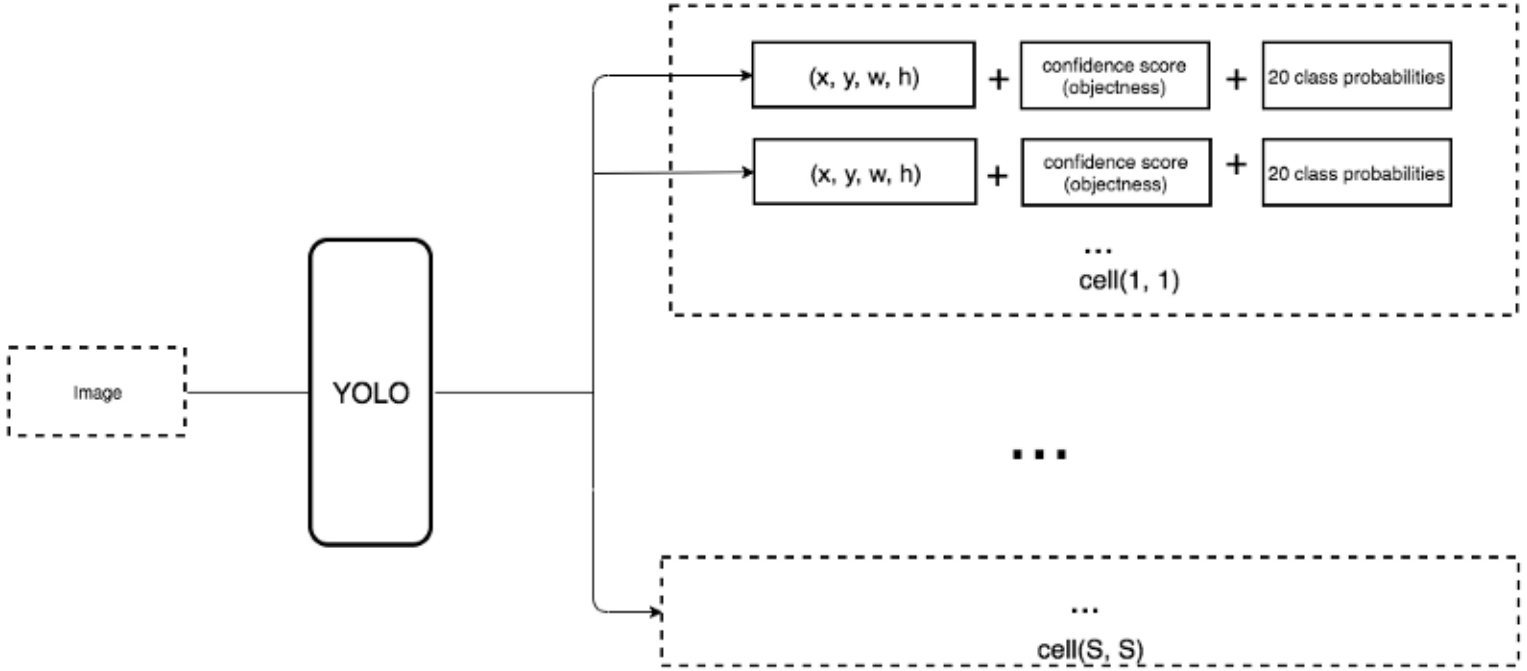
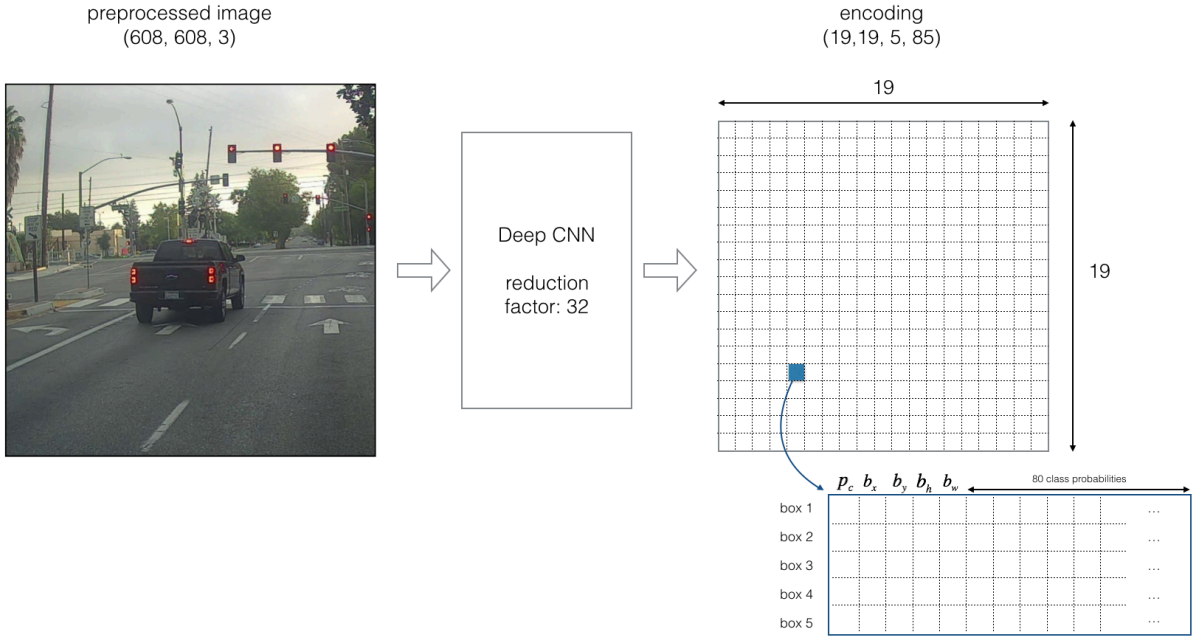


Figure Source

YOLOv2


(x, y, w, h)
+
confidence score (objectness)
+
20 class probabilities
(x, y, w, h)
+
confidence score (objectness)
+
20 class probabilities
...
cell(1, 1)
Image
YOLO
...
...
cell(S, S)

CNN Output

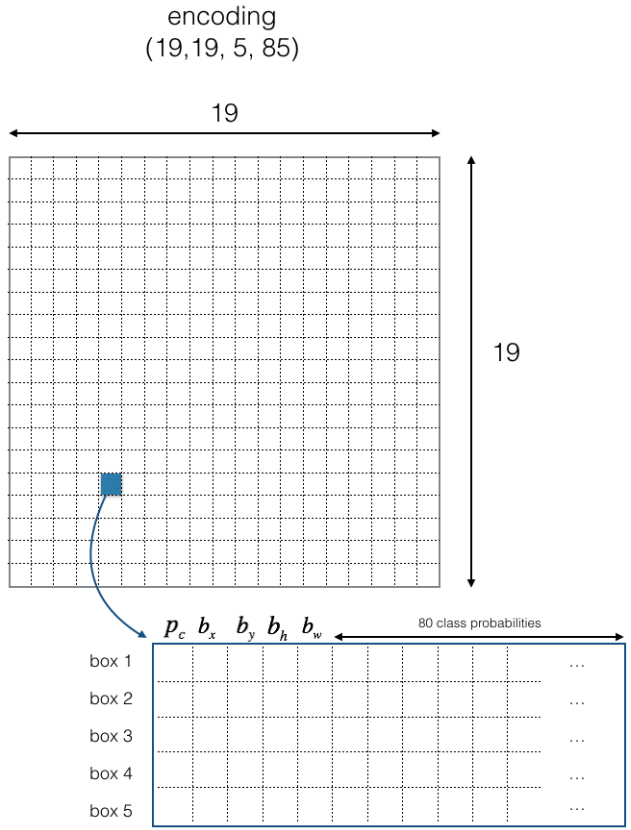


This encoding is used in YOLOv2, but is almost the same as YOLOv1

CNN Output

- (SxS) = 19x19 grid each with
  - 5 bounding box predictions

- Bounding box prediction includes:
  - x, y, width and height
  - confidence score that bounding box is correct
  - 80 classes

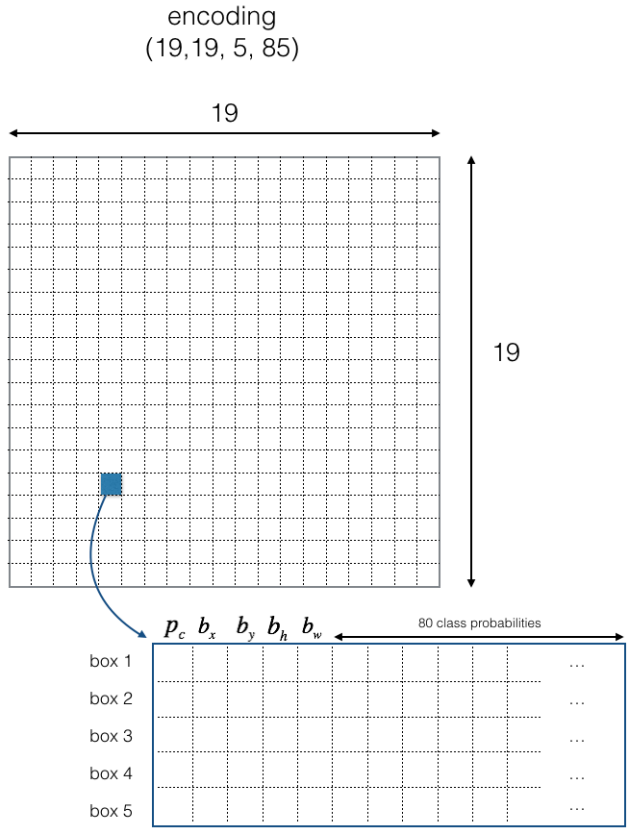


This encoding is used in YOLOv2, but is almost the same as YOLOv1

CNN Output

- The 5 bounding box predictions:
  - specialized for different shapes
  - specialized for different sizes

- The confidence score for bounding box should be the same as the IoU between the predicted bounding box and the ground truth bounding box



This encoding is used in YOLOv2, but is almost the same as YOLOv1

Useful Resources Task 3/4

- https://medium.com/@jonathan_hui/real-time-object-detection-with-yolo-yolov2-28b1b93e2088
- https://arxiv.org/abs/1506.02640
- https://arxiv.org/abs/1612.08242
- https://www.coursera.org/lecture/convolutional-neural-networks/non-max-suppression-dvrjH