You Only Look Once: Unified, Real-Time Object Detection[1]

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi

June 2, 2022

[1] Joseph Redmon et al. "You only look once: Unified, real-time object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 779–788.

Outline

The Need for Object Detection Algorithms

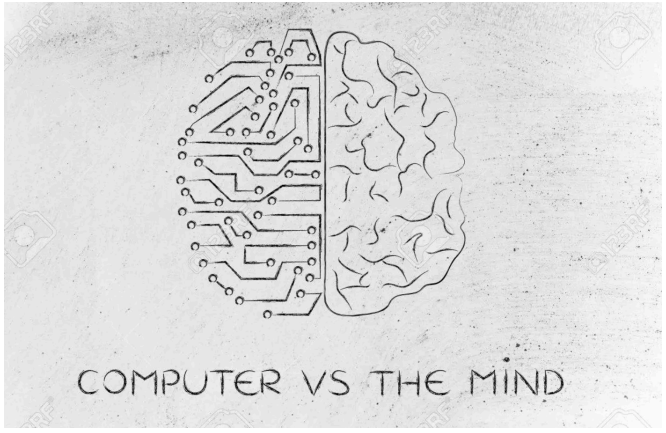


Figure: Computer Versus Human Mind[2]

[2] URL: https://www.123rf.com/photo_58580449_computer-vs-the-mind-artificial-intelligence-and-human-brain-comparison-design.html.

Existing Algorithms for Object Detection

- Reusing classifiers for object detection
- For example, the Deformable Parts Model (DPM) system and Region-Based Convolutional Neural Networks (R-CNNs)
- Complex, slow, and hard to optimize

The YOLO System

- Unified approach for real-time object detection
- Object detection as a regression problem
- Single neural network looks only once at the image and makes predictions
- End-to-end optimized

Advantages and Drawbacks of YOLO

Advantages:

- Fast (45 fps, fast YOLO is even faster at 155 fps)
- Makes less false positive predictions in the background compared to fast R-CNN
- Generalizes on images from new domains such as artwork compared to DPM and R-CNN

Drawback: Large localization error

The Model

- Input image: $S \times S$ grid
- Each grid predicts B bounding boxes and one set of class probabilities

$$Pr(Class_i|Object)$$

- Each bounding box consists of five predictions: (x, y) coordinates of the center of box, width w and height h of the box, and a confidence score

$$Pr[Object] \times IOU_{pred}^{truth}$$

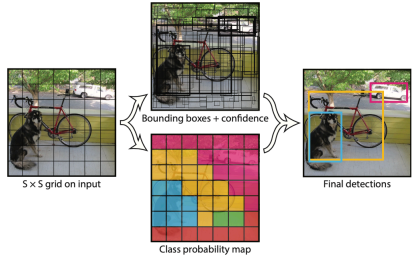


Figure: The Model

The Model

At inference, the class-specific confidence scores for each bounding box:

$$Pr(Class_i|Object) * Pr[Object] \times IOU_{pred}^{truth} = Pr(Class_i) \times IOU_{pred}^{truth}$$

The Network Architecture

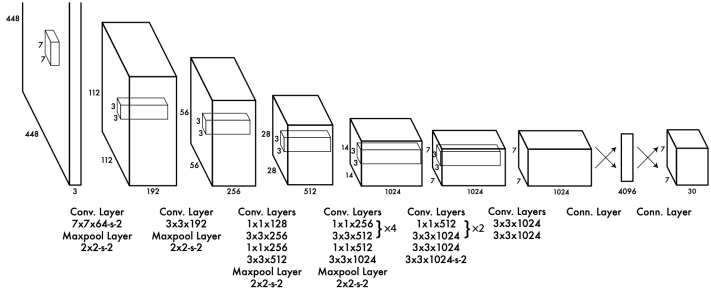


Figure: The YOLO Network Architecture

- 24 convolutional layers followed by 2 fully connected layers
- Alternating 3×3 convolutional layer followed by a 1×1 convolutional layer
- Final output is a tensor of predictions: $S \times S \times (B \times 5 + C) = 7 \times 7 \times 30$
- Fast YOLO: 9 convolutional layers with fewer filters in each layer

Training Dataset

- First 20 conv. layers pretrained on ImageNET at resolution (224×224 input image)
- Four conv. layers followed by two FC layers added for detection at double the resolution (448×448 input image)
- Trained on PASCAL VOC 2007 and 2012 (PASCAL VOC contains $C = 20$)

Activation Function

- Linear activation function for the final layer
- Leaky rectified linear activation for all other layers

$$\phi(x) = \begin{cases} x, & x > 0 \\ 0.1x, & \text{otherwise} \end{cases} \tag{1}$$

Loss Function

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

Hyper-Parameters

- Number of training epochs
- Batch size
- Momentum
- Learning rate
- Dropout
- Data augmentation

Inference

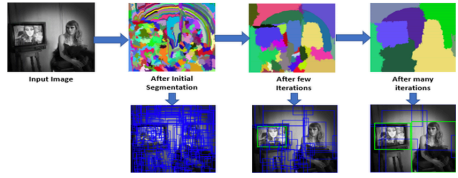
- Tested on the PASCAL VOC 2012
- Requires only one network evaluation
- But, an object can be detected by multiple cells (spatial diversity)

Limitations of YOLO

- A cell can detect limited number of nearby objects
- Not good at predicting grouped objects
- Fails to generalize to objects with unusual aspect ratios
- Extracts coarse features from the input image
- Equally penalizes small errors in large boxes and small errors in small boxes
- Localization error dominates the loss.

Comparison to Other Detection Systems

- Deformable Parts Model (DPM): Sliding window method for bounding boxes and conventional image processing for part object detection
- Region Based CNN (RCNN): Selective search for bounding boxes and other machine learning steps for classification and feature extraction
- Deep MultiBox: CNN used to predict bounding boxes. This is a single step in multi-class classification
- Other comparisons (OverFeat, MultiGrasp): CNN localization with local perspective only, and detecting grasping regions only in single object images.



Experiments and Real-Time Performance

- Pascal VOC 2007, 2012 Dataset
- Different Versions of R-CNN tested

Real-Time Detectors	Train	mAP	FPS
100Hz DPM [31]	2007	16.0	100
30Hz DPM [31]	2007	26.1	30
Fast YOLO	2007+2012	52.7	**155**
YOLO	2007+2012	**63.4**	45
Less Than Real-Time			
Fastest DPM [38]	2007	30.4	15
R-CNN Minus R [20]	2007	53.5	6
Fast R-CNN [14]	2007+2012	70.0	0.5
Faster R-CNN VGG-16[28]	2007+2012	73.2	7
Faster R-CNN ZF [28]	2007+2012	62.1	18
YOLO VGG-16	2007+2012	66.4	21

Table 1: Real-Time Systems on PASCAL VOC 2007. Comparing the performance and speed of fast detectors. Fast YOLO is the fastest detector on record for PASCAL VOC detection and is still twice as accurate as any other real-time detector. YOLO is 10 mAP more accurate than the fast version while still well above real-time in speed.

Error Analysis

- Correct: class is correct, and Intersection over union (IOU) is greater than 0.5.
- Localization: class is correct, and IOU is greater than 0.1 but less than 0.5.
- Similar: class is similar, and IOU is greater than 0.1.
- Other: class is wrong, and IOU is greater than 0.1.
- Background: for all IOU less than 0.1.
- **Combining YOLO and Fast RCNN is beneficial: a raise from 71.6% to 75%.**

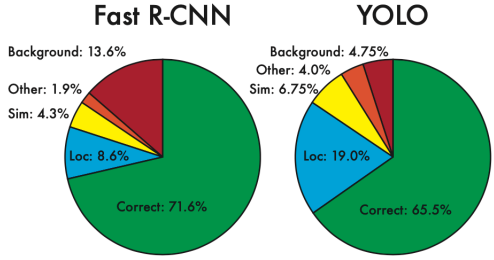


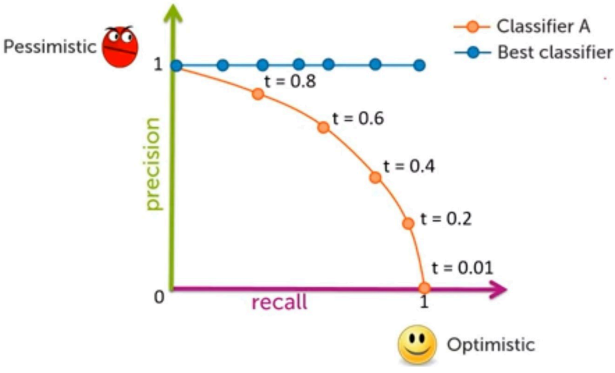
Figure 4: Error Analysis: Fast R-CNN vs. YOLO These charts show the percentage of localization and background errors in the top N detections for various categories (N = # objects in that category).

VOC 2012 Discussion (Visual Object Classes)

VOC 2012 test	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
MR_CNN_MORE_DATA [11]	**73.9**	**85.5**	**82.9**	**76.6**	**57.8**	**62.7**	**79.4**	77.2	86.6	**55.0**	**79.1**	**62.2**	87.0	**83.4**	**84.7**	78.9	45.3	73.4	65.8	80.3	74.0
HyperNet_VGG	71.4	84.2	78.5	73.6	55.6	53.7	78.7	**79.8**	87.7	49.6	74.9	52.1	86.0	81.7	83.3	**81.8**	**48.6**	**73.5**	59.4	79.9	65.7
HyperNet_SP	71.3	84.1	78.3	73.3	55.5	53.6	78.6	79.6	87.5	49.5	74.9	52.1	85.6	81.6	83.2	81.6	48.4	73.2	59.3	79.7	65.6
Fast R-CNN + YOLO	70.7	83.4	78.5	73.5	55.8	43.4	79.1	73.1	**89.4**	49.4	75.5	57.0	**87.5**	80.9	81.0	74.7	41.8	71.5	68.5	**82.1**	67.2
MR_CNN_S_CNN [11]	70.7	85.0	79.6	71.5	55.3	57.7	76.0	73.9	84.6	50.5	74.3	61.7	85.5	79.9	81.7	76.4	41.0	69.0	61.2	77.7	72.1
Faster R-CNN [28]	70.4	84.9	79.8	74.3	53.9	49.8	77.5	75.9	88.5	45.6	77.1	55.3	86.9	81.7	80.9	79.6	40.1	72.6	60.9	81.2	61.5
DEEP_ENS_COCO	70.1	84.0	79.4	71.6	51.9	51.1	74.1	72.1	88.6	48.3	73.4	57.8	86.1	80.0	80.7	70.4	46.6	69.6	**68.8**	75.9	71.4
NoC [29]	68.8	82.8	79.0	71.6	52.3	53.7	74.1	69.0	84.9	46.9	74.3	53.1	85.0	81.3	79.5	72.2	38.9	72.4	59.5	76.7	68.1
Fast R-CNN [14]	68.4	82.3	78.4	70.8	52.3	38.7	77.8	71.6	89.3	44.2	73.0	55.0	**87.5**	80.5	80.8	72.0	35.1	68.3	65.7	80.4	64.2
UMICH_FGS_STRUCT	66.4	82.9	76.1	64.1	44.6	49.4	70.3	71.2	84.6	42.7	68.6	55.8	82.7	77.1	79.9	68.7	41.4	69.0	60.0	72.0	66.2
NUS_NIN_C2000 [7]	63.8	80.2	73.8	61.9	43.7	43.0	70.3	67.6	80.7	41.9	69.7	51.7	78.2	75.2	76.9	65.1	38.6	68.3	58.0	68.7	63.3
BabyLearning [7]	63.2	78.0	74.2	61.3	45.7	42.7	68.2	66.8	80.2	40.6	70.0	49.8	79.0	74.5	77.9	64.0	35.3	67.9	55.7	68.7	62.6
NUS_NIN	62.4	77.9	73.1	62.6	39.5	43.3	69.1	66.4	78.9	39.1	68.1	50.0	77.2	71.3	76.1	64.7	38.4	66.9	56.2	66.9	62.7
R-CNN VGG BB [13]	62.4	79.6	72.7	61.9	41.2	41.9	65.9	66.4	84.6	38.5	67.2	46.7	82.0	74.8	76.0	65.2	35.6	65.4	54.2	67.4	60.3
R-CNN VGG [13]	59.2	76.8	70.9	56.6	37.5	36.9	62.9	63.6	81.1	35.7	64.3	43.9	80.4	71.6	74.0	60.0	30.8	63.4	52.0	63.5	58.7
YOLO	57.9	77.0	67.2	57.7	38.3	22.7	68.3	55.9	81.4	36.2	60.8	48.5	77.2	72.3	71.3	63.5	28.9	52.2	54.8	73.9	50.8
Feature Edit [33]	56.3	74.6	69.1	54.4	39.1	33.1	65.2	62.7	69.7	30.8	56.0	44.6	70.0	64.4	71.1	60.2	33.3	61.3	46.4	61.7	57.8
R-CNN BB [13]	53.3	71.8	65.8	52.0	34.1	32.6	59.6	60.0	69.8	27.6	52.0	41.7	69.6	61.3	68.3	57.8	29.6	57.8	40.9	59.3	54.1
SDS [16]	50.7	69.7	58.4	48.5	28.3	28.8	61.3	57.5	70.8	24.1	50.7	35.9	64.9	59.1	65.8	57.1	26.0	58.8	38.6	58.9	50.7
R-CNN [13]	49.6	68.1	63.8	46.1	29.4	27.9	56.6	57.0	65.9	26.5	48.7	39.5	66.2	57.3	65.4	53.2	26.2	54.5	38.1	50.6	51.6

Table 3: PASCAL VOC 2012 Leaderboard. YOLO compared with the full `comp4` (outside data allowed) public leaderboard as of November 6th, 2015. Mean average precision and per-class average precision are shown for a variety of detection methods. YOLO is the only real-time detector. Fast R-CNN + YOLO is the forth highest scoring method, with a 2.3% boost over Fast R-CNN.

Review on Precision and Recall



Precision attempts to answer the following question:

What proportion of positive identifications was actually correct?

Precision is defined as follows:

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

Recall attempts to answer the following question:

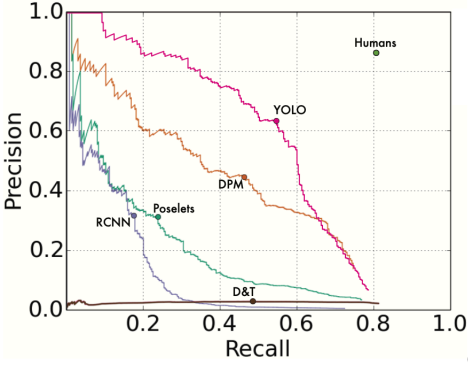
What proportion of actual positives was identified correctly?

Mathematically, recall is defined as follows:

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

Generalizability: Person Detection in Artwork

Picasso Dataset and People-Art Dataset are used in this testing.



(a) Picasso Dataset precision-recall curves.

	VOC 2007	Picasso		People-Art
	AP	AP	Best F_1	AP
YOLO	**59.2**	**53.3**	**0.590**	**45**
R-CNN	54.2	10.4	0.226	26
DPM	43.2	37.8	0.458	32
Poselets [2]	36.5	17.8	0.271	
D&T [4]	-	1.9	0.051	

(b) Quantitative results on the VOC 2007, Picasso, and People-Art Datasets. The Picasso Dataset evaluates on both AP and best F_1 score.

Figure 5: Generalization results on Picasso and People-Art datasets.

YOLO Real-Time Detection in the Wild

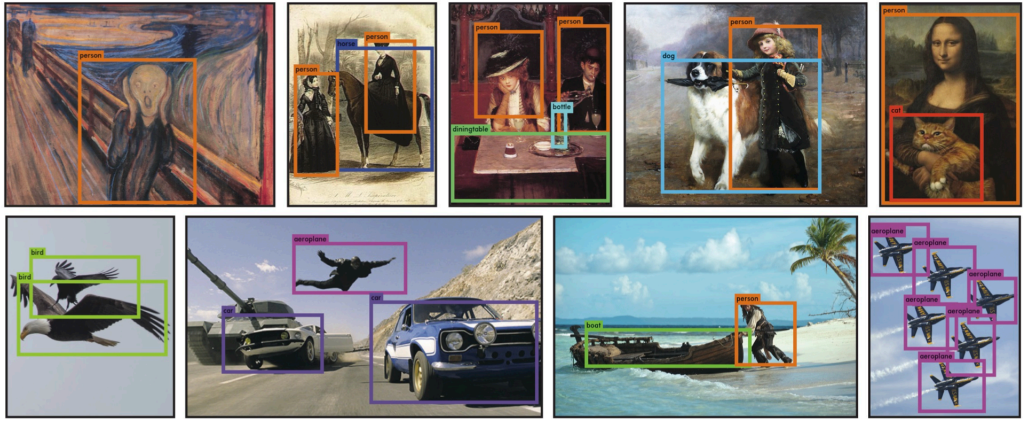


Figure 6: Qualitative Results. YOLO running on sample artwork and natural images from the internet. It is mostly accurate although it does think one person is an airplane.

In Conclusion, "You Only Look Once"