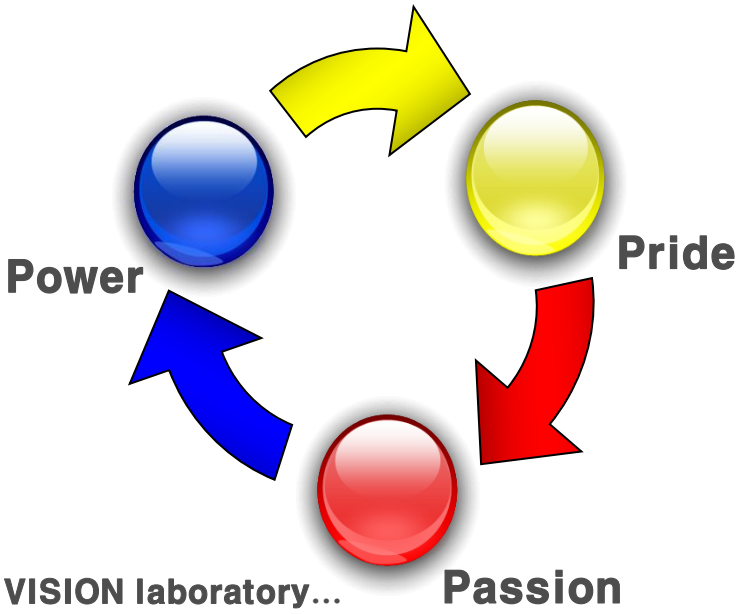


You Only Look Once:
Unified, Real-Time Object Detection

Vision System Lab, Gyumin Park
yywnnaa@gmail.com
Jan 31, 2024

Soongsil University
VISION SYSTEM LABOARTORY

1. Introduction

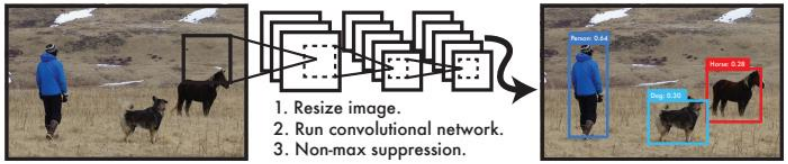


Figure 1: The YOLO Detection System. Processing images with YOLO is simple and straightforward. Our system (1) resizes the input image to 448×448, (2) runs a single convolutional network on the image, and (3) thresholds the resulting detections by the model's confidence.

- you only look once (YOLO) at an image to predict what objects are present and where they are
- refreshingly simple
- single neural network
- extremely fast
- less likely to predict false positives on background

VISION SYSTEM LABORATORY

1. Introduction

several benefits over traditional methods

1. extremely fast

don’t need a complex pipeline

achieves high mAP

2. reasons globally about the image when making predictions (next page)

sees the entire image during training and test time

makes less than half the number of background errors compared to Fast R-CNN

3. learns generalizable representations of objects

highly generalizable

less likely to break down when applied to new domains or unexpected inputs

less accuracy than sota

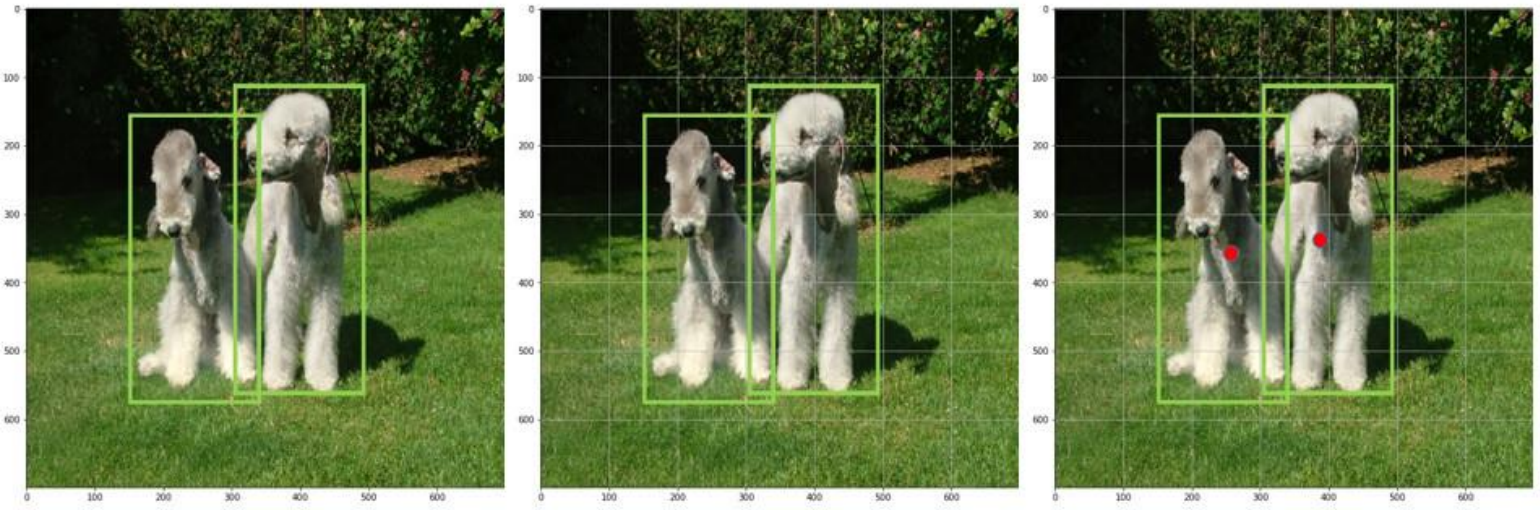
quickly identify objects in images

struggles to precisely localize some objects, especially small ones

VISION SYSTEM LABORATORY

2. Unified Detection

- single neural network object detection
- uses features from the entire image to predict each bounding box
- also predicts all bounding boxes across all classes for an image simultaneously
- reasons globally about the full image and all the objects in the image
- end-to-end training, real time speeds, high average precision



- **divides the input image into an S × S grid**
- If the center of an object falls into a grid cell, that grid cell is **responsible** for detecting that object

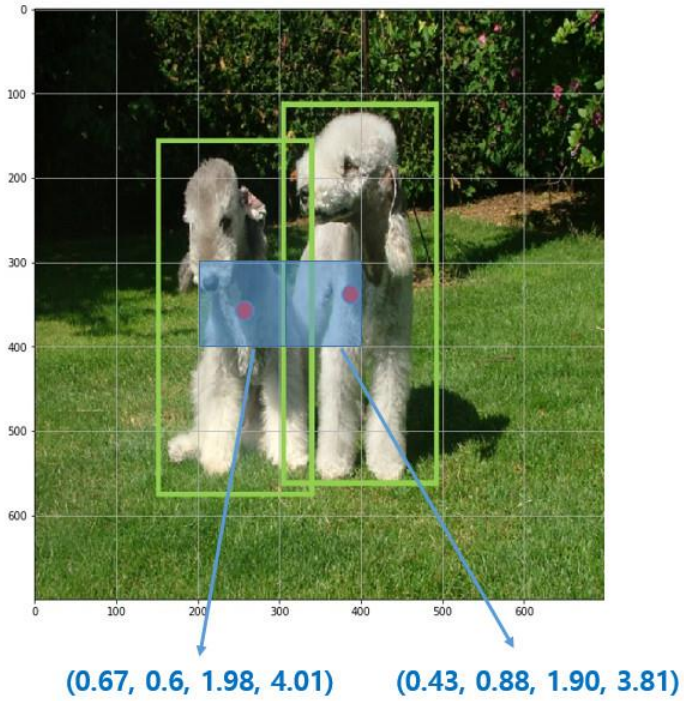
2. Unified Detection

- **Each grid cell predicts B bounding boxes and confidence scores for those boxes**
- **confidence scores** : $\Pr(\text{Object}) * \bar{\text{IOU}}_{\text{pred}}^{\text{truth}}$
- reflect how confident the model is that the box contains an object and also how accurate it thinks the box is that it predicts

- confidence score : to equal the IOU between the predicted box and the ground truth
- no object exists - confidence score : zero

VISION SYSTEM LABORATORY

2. Unified Detection



- Each bounding box consists of 5 predictions:
- x, y, w, h, and confidence

- (x, y) : represent the center of the box relative to the bounds of the grid cell
- width, height : predicted relative to the whole image

- confidence prediction : represents the IOU between the predicted box and any ground truth box

- Each grid cell predicts C conditional class probabilities (conditioned on the grid cell containing an object)
- only predict one set of class probabilities per grid cell, regardless of the number of boxes B

VISION SYSTEM LABORATORY

2. Unified Detection

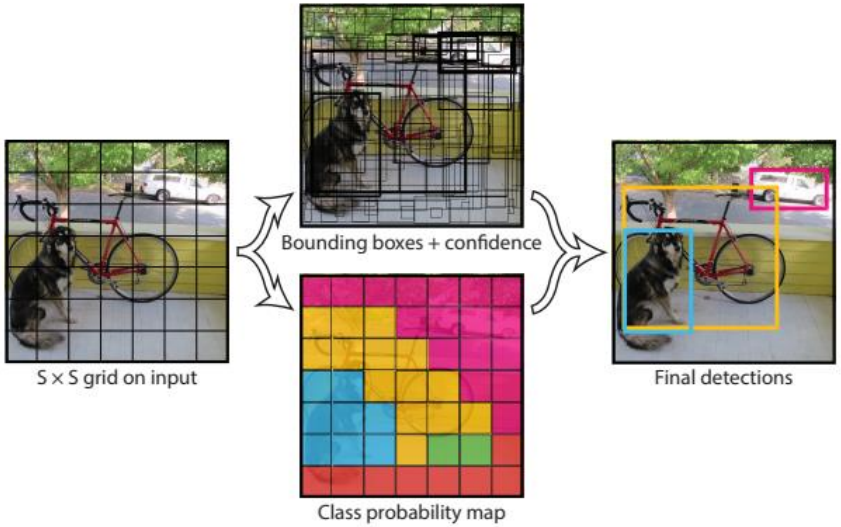


Figure 2: The Model. Our system models detection as a regression problem. It divides the image into an $S \times S$ grid and for each grid cell predicts B bounding boxes, confidence for those boxes, and C class probabilities. These predictions are encoded as an $S \times S \times (B * 5 + C)$ tensor.

evaluating YOLO on PASCAL VOC,

S = 7, B = 2, C = 20

(PASCAL VOC has 20 labelled classes)

final prediction : 7 × 7 × 30 tensor

7 x 7 x (2 x 5 + 20)

$$pred_{cell} = [c_1, c_2, \ldots, c_{20}, p_{c_1}, x, y, w, h, p_{c_2}, x, y, w, h]$$

Class probabilities

Box 1

Box 2

Box1 confidence score

Box2 confidence score

VISION SYSTEM LABORATORY

2. Unified Detection

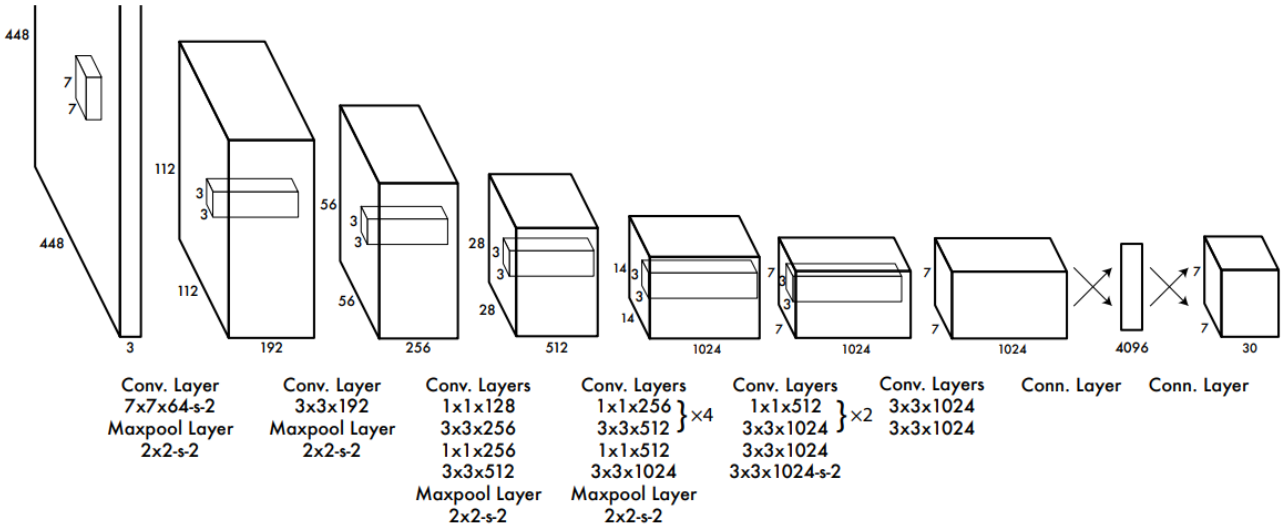


Figure 3: The Architecture. Our detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating 1×1 convolutional layers reduce the features space from preceding layers. We pretrain the convolutional layers on the ImageNet classification task at half the resolution (224×224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

- pretrain convolutional layers on the ImageNet 1000-class competition dataset
- 88% accuracy, comparable to the GoogLeNet models

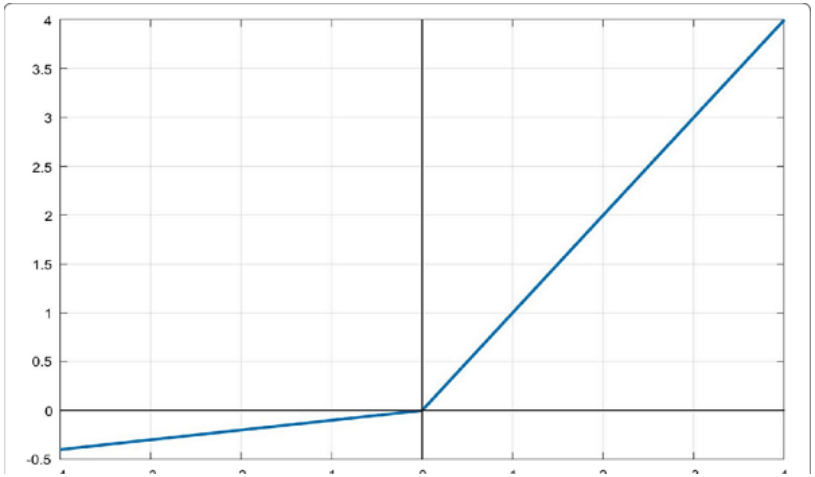
- convert the model to perform detection – “adding both convolutional and connected layers to pretrained networks can improve performance” - add four convolutional layers and two FC layers with randomly initialized weights
- Detection often requires fine-grained visual information so increase the input resolution of the network from 224 × 224 to 448 × 448

2. Unified Detection

- final layer predicts both class probabilities and bounding box coordinates
- normalize width and height - fall between 0 and 1

- final layer - linear activation function
- all other layers - leaky rectified linear activation

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



Leaky ReLU

- small slope when negative
- allows it to convey information while preserving some sort of linearity
- helps the model converge and learn faster

VISION SYSTEM LABORATORY

Loss Function

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

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

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

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

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

sum-squared error : easy to optimize

VISION SYSTEM LABORATORY

loss

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

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

- S^2 : number of grid cells (7^2 = 49)
- B : number of bounding boxes per grid cell (=2)
- λ_{coord} : = 5, parameter for weighting cells containing objects (in every image many grid cells do not contain any object)
- $1_{i,j}^{obj}$: index parameter assigned as 1 if the j-th bounding box in the i-th grid cell is responsible for predicting an object, and 0 otherwise
- $\hat{x}_i, \hat{y}_i, \hat{w}_i, \hat{h}_i$: x, y coordinates and width, height of the ground truth box
- x_i, y_i, w_i, h_i : x, y coordinates, width, height of predicted bounding box

loss

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

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

- λ_{noobj} : = 0.5, Weight parameter to multiply grid cells that do not contain objects

Compared to setting λobj=5, the influence of grid cells that do not contain objects was reduced by setting it significantly smaller.

- $1_{i,j}^{noobj}$: An index parameter that is 1 when the jth bounding box of the ith grid cell is not responsible to predict an object, and 0 otherwise
- C_i : 1 if the object is included, 0 otherwise.
- $\hat{C}_i$: Confidence score of predicted bounding box

VISION SYSTEM LABORATORY

loss

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

- $p_i(c)$: Actual class probabilities
- $\hat{p}_i(c)$: Predicted class probabilities

VISION SYSTEM LABORATORY

2.2. Training

- 135 epochs
- batch size of 64
- a momentum of 0.9
- a decay of 0.0005

- For the first epochs - slowly raise the learning rate from 10−3 to 10−2 .
- (start at a high learning rate - model often diverges due to unstable gradients)
- continue training with 10−2 for 75 epochs, then 10−3 for 30 epochs, and finally 10−4 for 30 epochs

To avoid overfitting - use **dropout** and **extensive data augmentation**

- dropout rate = .5
- random scaling and translations of up to 20% of the original image size
- randomly adjust the exposure and saturation of the image by up to a factor of 1.5 in the HSV color space

- non-maximal suppression adds 2- 3% in mAP

2.4. Limitations of YOLO

Spatial constraints

- each grid cell only predicts two boxes and can only have one class
- limits the number of nearby objects
- struggles with small objects that appear in groups, such as flocks of birds

Difficulty in Generalization

- struggles to generalize to objects in new or unusual aspect ratios or configurations

Limitations of the Loss Function

- loss function treats errors the same in small bounding boxes versus large bounding boxes
- a small error in a small box has a much greater effect on IOU
- main source of error is incorrect localizations

3. Comparison to Other Detection Systems

compare the YOLO detection system to several top detection frameworks, highlighting key similarities and differences

- Deformable Parts Models: DPM uses a disjoint pipeline for tasks like static feature extraction, region classification, and bounding box prediction. YOLO replaces this with **a single convolutional neural network**, offering **faster speed** and **higher accuracy**.

- R-CNN: R-CNN uses a complex pipeline for region proposals, slowing down the system. In contrast, YOLO places spatial constraints on grid cell proposals and reduces the number of bounding box proposals, enhancing efficiency.

- Other Fast Detectors: Fast and Faster R-CNN, DPM, etc., optimize individual components to improve speed but fall short of real-time performance. YOLO **throws out the pipeline entirely and is fast by design**.

VISION SYSTEM LABORATORY

3. Comparison to Other Detection Systems

- Detectors for Single Classes: Detectors for single classes like faces or people can be highly optimized due to less variation. On the other hand, YOLO is trained to **detect a variety of objects** simultaneously.

- Deep MultiBox: MultiBox trains a convolutional neural network for region proposals but cannot perform general object detection and requires further image patch classification. In contrast, YOLO is a **complete detection system**.

- OverFeat: OverFeat performs efficient sliding window detection but cannot reason about global context, requiring significant post-processing.

- MultiGrasp: MultiGrasp is a simpler task as it only needs to predict a single graspable region. On the contrary, YOLO **predicts both bounding boxes and class probabilities for multiple objects** in an image.

VISION SYSTEM LABORATORY

4.1. Comparison to Other Real-Time Systems

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.

- Fast YOLO - the fastest object detection method
- YOLO - 63.4% mAP, real-time performance
- train YOLO using VGG-16 : more accurate, significantly slower than YOLO

4.2. VOC 2007 Error Analysis

Fast R-CNN is one of the highest performing detectors on PASCAL and it's detections are publicly available

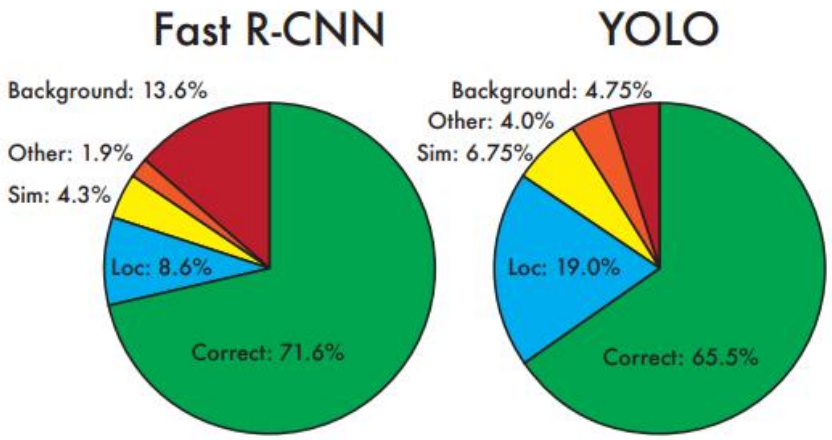


Figure 4: Error Analysis: Fast R-CNN vs. YOLO These charts show the percentage of localization and background errors in the top N detections for various categories (N = # objects in that category).

- Correct: correct class and IOU > .5
- Localization: correct class, .1 < IOU < .5
- Similar: class is similar, IOU > .1
- Other: class is wrong, IOU > .1
- Background: IOU < .1 for any object

VISION SYSTEM LABORATORY

4.3. Combining Fast R-CNN and YOLO

	mAP	Combined	Gain
Fast R-CNN	71.8	-	-
Fast R-CNN (2007 data)	**66.9**	72.4	.6
Fast R-CNN (VGG-M)	59.2	72.4	.6
Fast R-CNN (CaffeNet)	57.1	72.1	.3
YOLO	63.4	**75.0**	**3.2**

Table 2: Model combination experiments on VOC 2007. We examine the effect of combining various models with the best version of Fast R-CNN. Other versions of Fast R-CNN provide only a small benefit while YOLO provides a significant performance boost.

- By using YOLO to eliminate background detections from Fast R-CNN we get a significant boost in performance
- doesn't benefit from the speed of YOLO

VISION SYSTEM LABORATORY

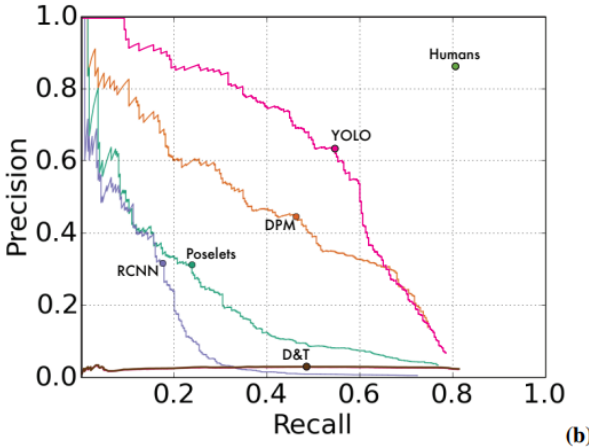
4.4. VOC 2012 Results

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.

- struggles with small objects compared
- categories like bottle, sheep, and tv/monitor YOLO scores 8-10% lower than R-CNN or Feature Edit
- categories like cat and train YOLO achieves higher performance
- combined Fast R-CNN + YOLO model is one of the highest performing detection methods

4.5. Generalizability: Person Detection in Artwork



(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.

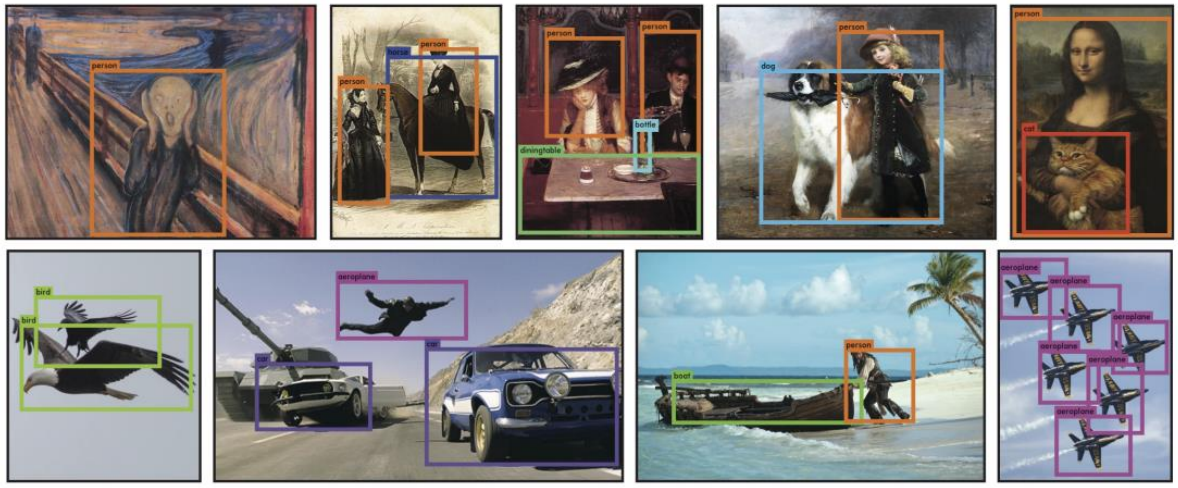


Figure 6: Qualitative Results. YOLO running on sample artwork and natural images from the internet. It is mostly accurate although it does think one person is an airplane.

testing person detection on artwork

R-CNN drops off considerably when applied to artwork

DPM maintains its AP well when applied to artwork

YOLO has good performance

YOLO models the size and shape of objects, as well as relationships between objects and where objects commonly appear

Artwork and natural images are similar in terms of the size and shape of objects, thus YOLO can still predict good bounding boxes and detections

6. Conclusion

YOLO, a unified model for object detection

- simple to construct
- trained on a loss function that directly corresponds to detection performance
- entire model is trained jointly

- Fast YOLO is the fastest general-purpose object detector in the literature and YOLO pushes the state-of-the-art in real-time object detection
- YOLO generalizes well to new domains making it ideal for applications that rely on fast, robust object detection