Contents

1 Introductionl **2**

1.1 Unified Detection 4

1. Introductionl

Humans glance at an image and instantly know what objects are in the image, where they are, and how they interact. The human visual system is fast and accurate, allowing us to perform complex tasks like driving with little conscious thought. Fast, accurate algorithms for object detection would allow computers to drive cars without specialized sensors, enable assistive devices to convey real-time scene information to human users, and unlock the potential for general purpose, responsive robotic systems. Systems like deformable parts models (DPM) use a sliding window approach where the classifier is run at evenly spaced locations over the entire image . More recent approaches like R-CNN use region proposal.

1. Resize image.
2. Run convolutional network.
3. Non-max suppression.

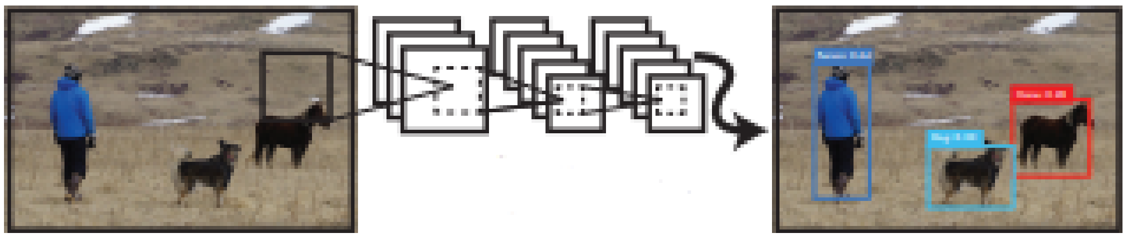
Figure 1: The YOLO Detection System.

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.

methods to first generate potential bounding boxes in an image and then run a classifier on these proposed boxes. After classification, post-processing is used to refine the bounding boxes, eliminate duplicate detections, and rescore the boxes based on other objects in the scene [13]. These complex pipelines are slow and hard to optimize because each individual component must be trained separately. We reframe object detection as a single regression problem, straight from image pixels to bounding box coordinates and class probabilities. Using our system, you only look once (YOLO) at an image to predict what objects are present and where they are. YOLO is refreshingly simple: see Figure 1.

A single convolutional network simultaneously predicts multiple bounding boxes and class probabilities for those boxes. YOLO trains on full images and directly optimizes detection performance. This unified model has several benefits over traditional methods of object detection. First, YOLO is extremely fast. Since we frame detection as a regression

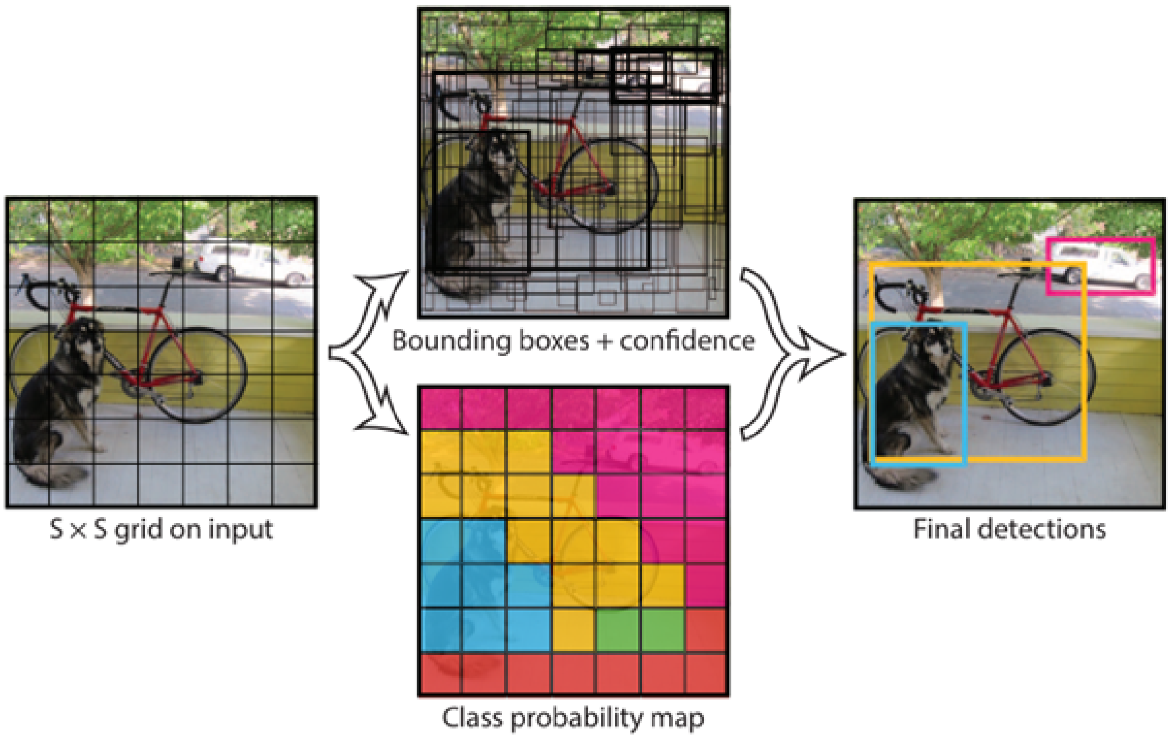

Figure 2: S*S grid on input, etc.

Table 1: Real-Time Detectors

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

Table 2: Less Than Real-Time

Less Than Real-Time Detectors	Train	mAP	FPS
Fastest DPM	2007	30.4	15
R-CNN Minus R	2007	53.5	6
Fast R-CNN	2007+2012	70.0	0.5
Faster R-CNN VGG-16	2007+2012	73.2	7
Faster R-CNN ZF	2007+2012	62.1	18
YOLO VGG-16	2007+2012	66.4	21

problem we don't need a complex pipeline. We simply run our neural network on a new image at test time to predict detections. Our base network runs at 45 frames per second with no batch processing on a Titan X GPU and a fast version runs at more than 150 fps.

$$\lambda_{coord} = \sum_{i=0}^{s^2} \sum_{j=0}^{M} (\sqrt{w_i} - \sqrt{w_j})^2 \tag{1}$$

Fast R-CNN, a top detection method [14], mistakes background patches in an image for objects because it can't see the larger context. YOLO makes less than half the number of background errors compared to Fast R-CNN. Third, YOLO learns generalizable representations of objects. When trained on natural images and tested on artwork, YOLO outperforms top detection methods like DPM and R-CNN by a wide margin. Since YOLO is highly generalizable it is less likely to break down when applied to new domains or unexpected inputs.

- CNN is a type of deep learning algorithm.
- Is commonly used for image recognition tasks.
- Have convolutional layers for feature extraction.
- Also pooling layers for dimensionality reduction.

YOLO still lags behind state-of-the-art detection systems in accuracy. While it can quickly identify objects in images it struggles to precisely localize some objects, especially small ones. We examine these tradeoffs further in our experiments. All of our training and testing code is open source. A variety of trained models are also available to download.

1.1 Unified Detection

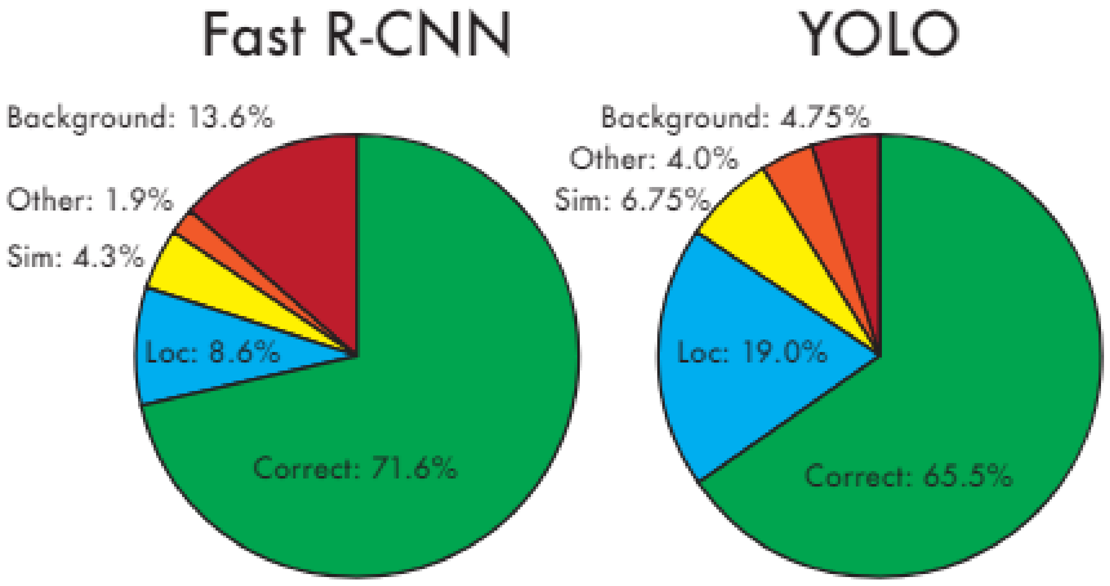

Figure 3: Fast R-CNN.

We unify the separate components of object detection into a single neural network. Our network uses features from the entire image to predict each bounding box. It also predicts all bounding boxes across all classes for an image simultaneously. This means our network reasons globally about the full image and all the objects in the image.

$$s_n = \begin{cases} 0, & \frac{g1(1-r^n)}{(1-r)} \\ 1, & e^{-g1\Delta r} f(k\Delta r) \end{cases} \tag{2}$$

These confidence scores reflect how confident the model is that the box contains an object and also how accurate it thinks the box is that it predicts. Formally we define confidence as Pr(Object)IOUtruth pred . If no object exists in that cell, the confidence scores should be zero. Otherwise we want the confidence score to equal the intersection over union (IOU) between the predicted box and the ground truth. Each bounding box consists of 5 predictions: x, y, w, h, and confidence.

$$\lambda_{coord} \sum_{i=0}^{s} \sum_{i=0}^{B} 1_{ij}^{obj} [(x_i - \widehat{x}_i)^2 + (y_i - \widehat{y}i)^2] + \lambda_{cord} \sum_{i=0}^{s} \sum_{i=0}^{2} 10^b [(\sqrt{w_i} - \sqrt{w_i)^2 + (\sqrt{h_i} - \sqrt{h_i})^2} \tag{3}$$