Week7 Pre-Report: You Only Look Once - Unified, Real-Time Object Detection

Euijin Hong[1]
[1]Department of Electrical and Electronic Engineering, Yonsei University.

1 Introduction and Background

Object detection systems at that time operate by introducing classifiers to perform detection. These systems take the classifiers of a certain object and assess with taking various locations and scales in the image. DPM (deformable parts models) [2] use a sliding window strategy, where the classifier evaluates every evenly spaced locations in an entire image. R-CNN [3] utilizes an approach of region proposal methods, initially generating the potential bounding boxes and then running a classifier on these boxes. After this, bounding boxes are refined, duplicate predictions are removed, and boxes from other objects are rescored by post-processing procedure.

However, these complex detection flow is difficult and slow to train and optimize since they consist of separate models. In order to enhance the detection speed, YOLO algorithm newly suggests a single regression model of object detection, which yields bounding box coordinates and class probabilities straight from the image pixels. As shown in Figure 1. the coordinates of multiple bounding boxes and class probabilities are simultaneously predicted by a single convolutional network. YOLO can learn from full-image end-to-end and directly optimize its performance.

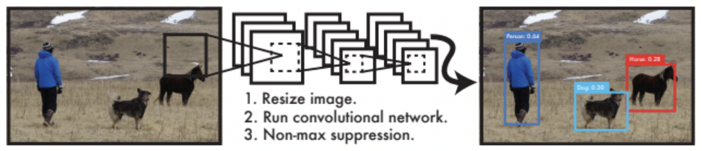

Figure 1: The YOLO detection system resizes the image to 448×448, runs a single convolutional network, and thresholds the resulting detections based on the model's confidence.

YOLO algorithm is beneficial than other traditional algorithms in several aspects. First, the model is extremely fast, its baseline model showing about 45fps in a Titan X GPU, while the fast version model achieves 150fps in the same GPU, while scoring mAP twice as big as other real-time detectors. Second, the model performs inference using a global region of an image, thus encodes the contextual and appearance information, and shows good results in distinguishing the background. Third, YOLO model learns generalizable representations of an object, which makes the model less likely to break down in applying to unexpected inputs or new domains. Although it shows less accuracy than the state-of-the-arts model, especially when detecting small objects, trade-off with its inference speed seems bearable.

2 Unified Detection: Theory and Configuration

The image is divided into N grids in YOLO algorithm. Each of the grid occupies the location in $S \times S$ dimensional space. A specific grid cell is responsible for the object detection when the center of the object locates on that grid. The N grids are related to detecting and localizing the object in the grid. These grids inference the adjacent B bounding boxes related to the cell coordinates, and their confidence scores. For instance, when each cell in a 19 x 19 grid is related to predicting five bounding boxes, 19 x 19 x 5 = 1805 bounding boxes are generated in total. Besides, the grids predict the label type of an object and the probability whether the object will exist in the bounding box.

A YOLO detector generates a prediction of class type of the object, its bounding box, and the confidence value (the probability the prediction gets correct). Each bounding box have parameters as follows:

- The center position of the bounding box in the image (b_x, b_y)
- The width and height of the box(b_w, b_h)
- The class of object (C)
- The probability of a class of object in the box (p_c)

The confidence is defined as $Pr(Object) * IOU_{pred}^{truth}$ where the score becomes zero when there are no object in the box, while it becomes the same as the intersecton over unit (IOU) value between the ground truth and the predicted box.

The conditional class probability is also predicted in every grid cell as $C = Pr(Class_i|Object)$. Each grid cell produces one set of class probability, regardless of the number of bounding box B.

During the test, YOLO multiplies the conditional class probability with the predicted individual box confidence. This gives the confidence scores for each box specifically for class. The scores contains both the probability whether the specific class to appear in the predicted box and how much the predicted box will fit the object. The final output has a size of $7 \times 7 \times 30$.

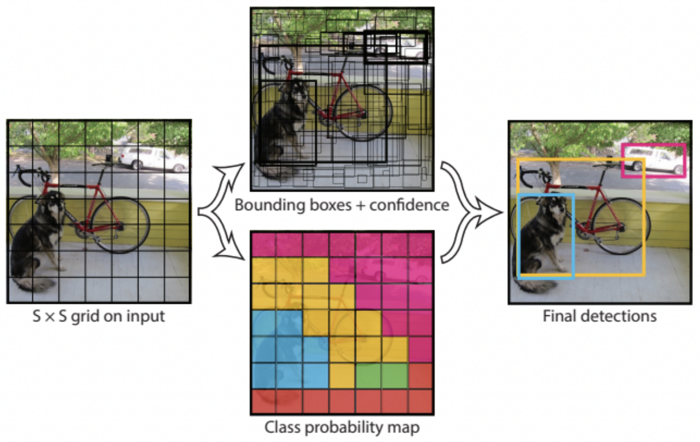

Figure 2: The system views detection as a regression based approach. The image is divided to $S \times S$ grid cells, with each of them predicting the bounding box B, confidence of those boxes, and the class probabilities C. The predictions are encoded in a tensor with dimension $S \times S \times (B * 5 + C)$.

2.1 Network Design

The model is implemented as a convolutional neural network and evaluated on Pascal VOC detection dataset [1]. The convolutional layers on the front of the network extract the features of the image and the fully connected layers inference the coordinates and the output probabilities.

The network as shown in Figure 3. consists of 24 convolutional layers and 2 fully connected layers. The model is derived from GoogLeNet model in image classification [5]. The inception modules in GoogLeNet is replaced by a 1×1 reduction layers followed by a 3×3 convolutional layers.

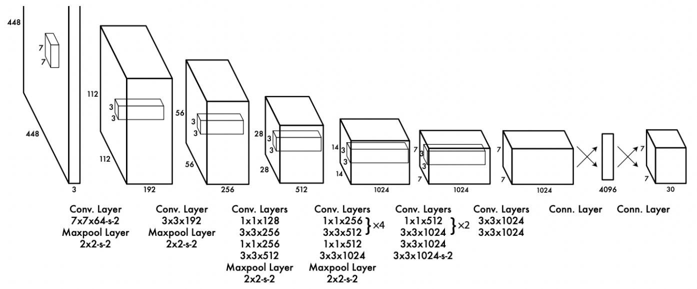
448
448
7
7
3
112
112
3
3
192
56
56
3
3
256
28
28
3
3
512
14
14
3
3
1024
7
7
3
3
1024
7
7
1024
4096
7
7
30
Conv. Layer 7x7x64-s-2 Maxpool Layer 2x2-s-2
Conv. Layer 3x3x192 Maxpool Layer 2x2-s-2
Conv. Layers 1x1x128 3x3x256 1x1x256 3x3x512 Maxpool Layer 2x2-s-2
Conv. Layers 1x1x256 3x3x512 ×4 1x1x512 3x3x1024 Maxpool Layer 2x2-s-2
Conv. Layers 1x1x512 3x3x1024 ×2 3x3x1024 3x3x1024-s-2
Conv. Layers 3x3x1024 3x3x1024
Conn. Layer
Conn. Layer

Figure 3: The YOLO detection network architecture. It consists of 24 convolutional layers followed by 2 fully connected layers. 1×1 convolutional layers reduce the feature space of the former layers. Pretraining is done by reducing the resolution to half, while testing is done in twice of the resolution.

Also, there is a fast version of YOLO, which outperforms the traditional object detection speed. The model uses a network with fewer convolutional layers and fewer filters in the layers. The rest of the training and testing parameters are identical to those of basic YOLO. The final prediction output has a size of $7 \times 7 \times 30$.

This is an extended abstract. The full paper is available at the Computer Vision Foundation webpage.

2.2 Training

Pretraining of the YOLO model is based on the ImageNet 1000-class competition dataset [4]. The first 20 convolutional layers in Figure 3. followed by an averabe-pooling layer and a fully connected layer, are used in pretraining. A single crop top-5 accuracy is obtained as 88% on the ImageNet 2012 validation set. The model uses a Darknet framework for every training and inference.

The model is then converted to perform detection, while the four convolutional layers and two fully connected layers are added with random-initialziaed weights, and doubling the resolution for obtaining fine-grained visual information.

The final layer predicts the bounding boxes and the class probabilities. The bounding boxes and their x, y coordinates are normalized by the width and height of an image.

The final layer uses a linear activation function while the other layers use a leaky rectified linear activation as follows:

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

Sum-squared error is used for generating the output because it is easy to optimize. Also, when the grid cell does not contain any object, gradient overpowering can occur since it pushes the confidence to zero, which can lead to model instability. To deal with this problem, researchers increased the loss from predicting bounding box coordinates and decreased the loss of confidence predictions, by setting the parameters $\lambda_{coord} = 5$ and $\lambda_{noobj} = 0.5$. Furthermore, they adjusted YOLO to predict the square root of the bounding box width and height to reflect that differences in location matter more in small bounding boxes.

Since YOLO predicts multiple bounding boxes per grid cell while what we want is one bounding box predictor being responsible for each object, researchers assign one predictor by selecting the prediction with the highest IOU with the ground truth. This allows each predictor to advance in predicting certain sizes, aspect ratios, or classes of object, improving overall recall.

The multi-part loss function is used for optimizing in training:

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

where $\mathbf{1}_i^{obj}$ denotes if object appears in cell i and $\mathbf{1}_{ij}^{obj}$ denotes that the jth bounding box predictor in cell i is "responsible" for that prediction.

The loss function gives penalty to classification error when an object exists in the grid cell following the conditional class probability. It also gives penalty to the bounding box coordinate error when the predictor is "responsible" for the ground truth box.

2.3 Inference

Predicting detections in a test image is performed by one network evaluation, as in training, thus the inference speed is very fast. YOLO predicts 98 bounding boxes and class probabilities in a single image.

Grid design of YOLO enables spatial diversity when predicting bounding boxes. Frequently, the network can easily identify the grid cell containing an object and only predicts one box for that object. Nevertheless, certain objects that are either large or located close to the border of multiple cells can be well localized by more than one cell. To address this issue, non-maximal suppression can be utilized to remove these multiple detections. Although it is not as crucial for performance as it is for R-CNN or DPM, non-maximal suppression results in a 23% increase in mAP.

2.4 Limitations of YOLO

The spatial constraint on bounding box predictions of YOLO limits the number of close objects that can be predicted. So, it shows low accuracy in predicting small objects in a group. Due to the fact that YOLO learns to predict bounding boxes based on the given data, it may face difficulties in generalizing to objects that possess new or unusual aspect ratios or configurations. Additionally, YOLO relies on relatively coarse features for the prediction of bounding boxes, as a result of multiple downsampling layers in our architecture. Lastly, the loss function treats errors in small and large bounding boxes equally, despite the fact that a small error in a small box can have a significant impact on the Intersection over Union (IOU) value. The primary cause of errors in the model is incorrect localizations.

3 Experiments

The researchers evaluate and compare YOLO with other real-time detection systems on the PASCAL VOC 2007 dataset. Then, they analyzed the errors made by YOLO and Fast R-CNN [3] on the same dataset, and demonstrate that YOLO can be utilized to rescore Fast R-CNN detections, thereby reducing false positives and enhancing performance. The performance of YOLO is then compared to state-of-the-art methods on the VOC 2012 dataset, and its superior ability to generalize to new domains is illustrated using two artwork datasets.

The researchers compared the performance and speed of fast object detectors, as shown in Figure 4 (left), highlighting that Fast YOLO is the fastest and most accurate detector on record for PASCAL VOC detection, surpassing any other real-time detector by twice the accuracy. Additionally, YOLO is 10 mAP more accurate than the fast version while still maintaining real-time speed.

Also, they analyzed the error of YOLO and R-CNN, as shown in Figure 4 (right), based on the following standards:

- 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

We can see that most of the errors in YOLO is localization errors, while showing less background errors compared to the R-CNN's result.

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

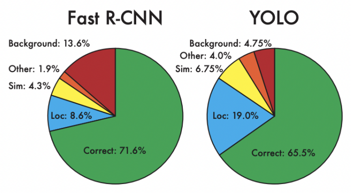

Figure 4: (left): Object Detection Systems in Real-Time on the PASCAL VOC 2007 Dataset, (right): Fast R-CNN vs. YOLO: Error Analysis

Furthermore, researchers complemented the pros and cons of the two models, Fast R-CNN and YOLO, by combining the two models. They could increase the mAP by 3.2% in a VOC 2007 test set.

Moreover, they assessed the generalizability of the YOLO model by performing object detection tasks on person detection in artworks, and real-time detection in the wild, showing satisfactory performances shown in Figure 5(left) and (right).

	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	

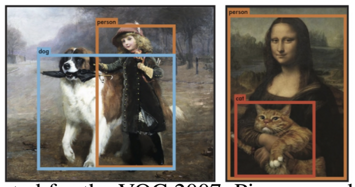

Figure 5: (left): AP outcomes are reported for the VOC 2007, Picasso, and People-Art Datasets, while in Picasso dataset, both AP and the highest F1 score are evaluated, (right): YOLO running on sample artwork images from the internet.

4 Conclusion

YOLO is a simple and unified model for object detection that can be trained directly on full images. It is trained on a loss function that corresponds directly to detection performance, and the entire model is trained jointly. YOLO is the fastest general-purpose object detector available, and it sets the state-of-the-art in real-time object detection. Additionally, YOLO can generalize well to new domains, which makes it an ideal choice for applications that require fast and robust object detection.

[1] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111:98–136, 2015.

[2] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2009.

[3] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[4] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015.

[5] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.