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

Jiho Bak
School of Electrical and Electronic Engineering, Yonsei University

Abstract

Unlike previous classifier-based object detection models, YOLO(You Only Look Once) addresses object detection problem as regression problem. YOL -O predicts the bounding boxes and the class probabilities directly from the full image with it's single neural network. There are several following properties. First it's architecture enables extremely fast inference, up to 45 fps. The small version, Fast YOLO, even achieved the speed of 155 fps. Second, it reasons globally since it predicts from the convolution of whole image, unlike sliding window or region proposal methods.Third, it learns more generalized representations. This is examined in experiment by testing models on the other domain like art-work. Lastly, due to the trade-offs between accuracy and speed, it still lags behind sota object detection models in accuracy. Though YOLO makes more localization errors, it is less likely to predict false positives on background [2].

1 Unified Detection

The previous object detection models evaluate their classifiers at various locations. However intuitively, human glance at an image just once to recognize the objects not processing the image several times. Therefore, YOLO is designed by reframing the object detection as a single regression problem.

1.1 Object Detection with Regression

YOLO divides the input image into $S \times S$ grid. Each grid cell predicts B bounding boxes, it's confidence score, and the class probabilities.
Each predictor of bounding box needs 5 predictions, x, y, w, h, and it's confidence. (x, y) is the center coordinates of the box, and each w, h is width and height of the box. These x, y, w, h are normalized between 0 and 1. Confidence score is defined as $\Pr(\text{Object}) * \text{IOU}^{\text{truth}}_{\text{pred}}$. The target confidence score is the IOU(Intersection Over Union) between predicted box and the ground truth box. C conditional class probabilities,
$\Pr(\text{Class}_i|\text{Object})$, are predicted by each grid cell. So, the B bounding boxes from the same grid cell, has the identical class probabilities. The concept of this unified detection is visualized in Fig 1.

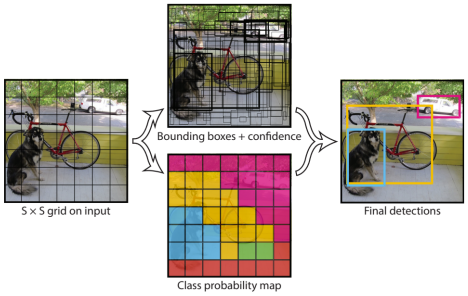

Figure 1: The concept of the grid division and bounding box, confidence, class probability prediction.

1.2 Network Architecture

As elaborated in Fig 2, the main architecture consists of 24 convolution layers and 2 fully-connected layers. This architecture is mainly from GoogLeNet, but using 1×1 convolution for dimension reduction, instead of inception modules. Fast YOLO uses 9 convolution layers instead of 24. The leaky rectified linear activation is applied to all layers except the final layer. For YOLO on PASCAL VOC, they used grid number of $S = 7$, and the bounding box number of $B = 2$, where $C = 20$. Therefore, the final prediction is a $7 \times 7 \times 30$ tensor.

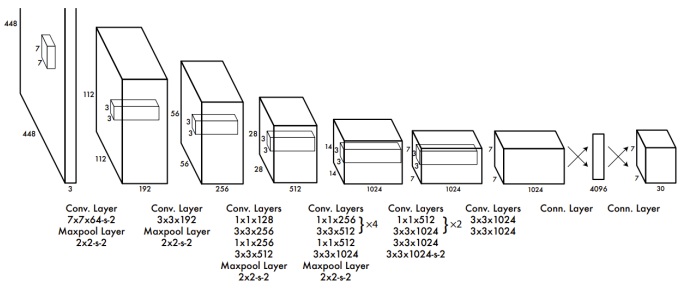

Figure 2: The main architecture.

2 Train

2.1 Training Details

The first 20 convolution layers were pretrained on ImageNet with extra average-pooling layer and fully-connected layer. The resolution of the ImageNet is 224×224 or 256×256, but the detection tasks often requires fine-grained resolution. So the resolution of the input is increased from 224×224 to 448×448, after the pretraining. The final network was trained for 135 epochs, in batchsize 64, on PASCAL 2007, 2012 datasets. To avoid overfitting, dropout and data augmentation were applied. For optimization, momentum of 0.9, decay of 0.0005 and manual learning rate scheduling were used.

2.2 Loss

The designed network have 98 bounding box predictors. For training, only one predictor is assigned to be "responsible" for single object. The predictor that has the highest IOU with the ground truth, is responsible for that object. Therefore, the loss is evaluated with only "responsible" predictors for that inference. Each predictor(bounding box) gets better with certain sizes, ratios, or classes [2].
Basically the sum-squared error is used. However this weights localization error and classification error equally. Also, since many grid cells do not contain objects commonly, the confidence scores of many cells are pushed to zero. This often overpowers the gradient from cells which contains objects.
Two remedies were applied for better loss. First, two hyper parameters $\lambda_{coord} = 5, \lambda_{noobj} = 0.5$ were weighted. To coordinate loss and to confidence score of no-object box, each. This increases the gradient of the coordinate prediction. And it decreases the gradient of the confidence score prediction that does not contain object. This prevents the pushing confidence to zero. Second, to address the matter of loss difference due to the size of the box, the square root is applied to width and height loss.

Loss Equation:

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

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

$$+ \quad \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} (C_i - \hat{C}_i)^2 + \lambda_{noobj} \quad \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{noobj} (C_i - \hat{C}_i)^2 \quad (3)$$

$$+ \quad \sum_{i=0}^{S^2} \mathbb{1}_{i}^{obj} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2 \quad (4)$$

The final equation of the elaborated loss is the sum of 1, 2, 3, 4. The indicator 1_{ij}^{obj} holds the value 1 if the jth bounding box of ith cell is "responsible" for that prediction. And the indicator 1_{ij}^{noobj} denotes the non-responsible predictor.

3 Limitation

There are three main limitations of YOLO. First, since each grid cell can only predict one class at a time, it imposes strong spatial constraints. So, YOLO struggles with small, grouped objects such as flocks of birds. Second, since YOLO directly learns the bounding box of the training data, it is hard to predict in case of unusual ratios or configurations of the bounding box. Finally, a small error in small bounding box has greater effect on IOU, of than in big bounding box's. Since the confidence score is defined by IOU, the size of the bounding box can lead to inappropriate loss evaluation.

4 Experiments

4.1 Real-Time System

The speed is the representative benefit of YOLO, fast object detection models were tested on PASCAL VOC 2007. Variations of YOLO and Fast YOLO, R-CNN series, and DPM models were tested. Only DPM and pure YOLO and Fast YOLO survived as real-time detector(Fig 3). The variation of YOLO such as 'YOLO VGG-16', improved the performance but the speed degradation was more severe. As a result, among real-time detectors, since DPM has poor performance, we can assure that YOLO is the only actual real-time object detector among them. YOLO is also tested in webcam for real-time object detection.

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

Figure 3: Comparison between Real-Time Systems.

4.2 VOC 2007 Error Analysis

To breakdown the effect of YOLO, they compared the YOLO and the state-of-the-art object detector Fast R-CNN on PASCAL VOC 2007. They analyzed the error by categorizing into 'Correct", 'Localization', 'Similar', 'Other', and 'Background'. The categorization method is as below, which was referenced from [1].

- Correct: correct class and IOU > 0.5
- Localization: correct class, 0.1 < IOU < 0.5
- Similar: class is similar, IOU > 0.1
- Other: class is wrong, IOU > 0.1
- Background: IOU < 0.1 for any object

As a result in Fig 4, YOLO has more localization error. However Fast R-CNN has almost 3x more background error. This indicates the stability of YOLO.

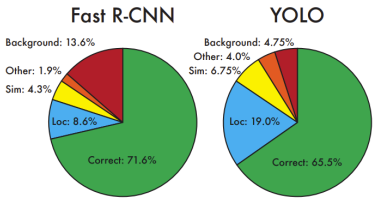

Figure 4: Error Analysis: Fast R-CNN vs YOLO

4.3 Combining Fast R-CNN and YOLO

If the bounding box of R-CNN is similarly predicted in YOLO, they gave a boost. This combination enables the advantage of YOLO to R-CNN, which to avoid background detection. Actually, this combination to best Fast R-CNN improved mAP by 3.2% on VOC 2007. The combination doesn't benefit in speed since two inference results are both used. However since YOLO is so fast, it can improve the performance almost without delay.
This combination and YOLO are both tested on PASCAL VOC 2012. YOLO scored 8-10% lower than R-CNN, but the combination improved the Fast R-CNN about 2.3% of mAP.

4.4 Generalizability: Person Detection in Artwork

To test the generalizability, trained YOLO and other models were tested on Picasso and People-Art dataset. YOLO outperformed other models(e.g. R-cNN, DPM). "Artwork and natural images are very different on a pixel level but they are similar in terms of the size and shape of objects, thus YOLO can still predict good bounding boxes and detections." [2]

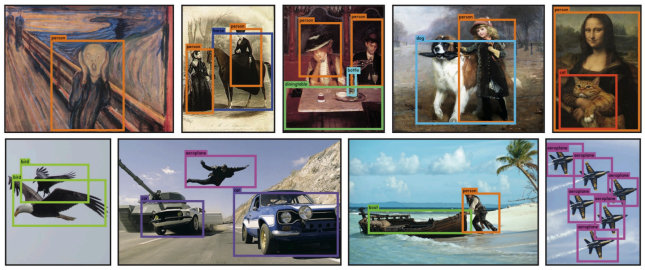

Figure 5: Qualitative Results of Artworks and Natural Images

5 Conclusion

YOLO addresses the object detection as regression problem. This concept enables the direct training from detection ground truth, which makes the system efficient and fast. As the result of the experiment, YOLO pushes the state-of-the-art in real-time object detection.

Reference

[1] Qieyun Dai Derek Hoiem, Yodsawalai Chodpathumwan. Diagnosing error in object detectors.

[2] Ross Girshick Ali Farhadi Joseph Redmon, Santosh Divvala. You only look once: Unified, real-time object detection. *CVPR*.