You Only Look Once: Object Detection

Replication of an advanced research paper

Hemant Hajare
Electrical Engineering
Indian Institute of Technology, Bombay
Mumbai, India
20D070037@iitb.ac.in

Jahnvi Rohela
Electrical Engineering
Indian Institute of Technology, Bombay
Mumbai, India
20D070040@iitb.ac.in

Tejaswee Sulekh
Eletcrical Engineering
Indian Institute of Technology, Bombay
Mumbai, India
20D070082@iitb.ac.in

***Abstract*—YOLO (You Only Look Once) is an extremely fast, novel approach to perform object detection using regressors, instead of traditional classifiers, wherein a single neural network predicts bounding boxes and class probabilities directly from full image in one evaluation. YOLO, while making larger number of localisation errors is less likely to predict false positives on the background and at the same time learns a general representation of objects and thus can be used in a number of application after single training.**

***Index Terms*—Object detection, R-CNN, YOLO**

I. Introduction

YOLO takes inspiration from human ability to take a quick look at an image and decipher the contextual information in them. A framework like YOLO would have several applications in general purpose, real-time, responsive robotic systems. Unlike more recent approaches like R-CNN which use region proposal methods to generate potential bounding boxes in an image and then run classifier on theses proposed boxes, YOLO reframes object detection as a single regression problem straight form image pixels to bounding box coordinates and class probabilities. It trains on full images and directly optimises detection performance using a single convolutional network simultaneously to predict multiple bounding boxes and class probabilities for those boxes making it very fast in the absence of a complex pipeline. Since YOLO sees entire image it encodes contextual information about classes as well as their appearance unlike sliding window and regional proposal-based techniques and thus makes half the number of background errors. YOLO is a highly generalizable and when trained on natural images and tested on art images YOLO outperforms R-CNN by a wide margin.

II. Methods

YOLO combines separate components of object detection into one single neural network using features from the entire image to predict each bounding box and to predict all bounding boxes across all classes for an image simultaneously. It divides an image into S $\times$ S grid. A grid cell is responsible for detecting an object if the centre of the object falls into the grid cell. A confident score is calculated for B bounding boxes predicted by each of the grid cells which reflects confidence of the model about having an object within each bounding box and accuracy of the predicted box.

Mathematically, confidence is defined as,

$$Confidence{=}Pr(Object){*}IOU_{pred}^{truth}$$

Where IOU is defined as intersection over union between the predicted box and the ground truth.

5 prediction are made for each bounding box: x, y, w, h, and confidence. The (x,y) coordinates represent the centre of the box relative to the bounds of the grid cell. w is the width and h is the height of the are predicted with respect to the whole image.

There is a conditional class probability $C = Pr(Class_i|Object)$ associated with each grid cell conditioned on the grid cell containing an object. Only one set of class probabilities are predicted per grid cell regardless the number of boxes B such that on multiplication of the conditional class probabilities and the individual box confidence predictions we get the class confidence score of each block.

$$Pr(Class_i|Object){*}Pr(Object){*}IOU_{pred}^{truth}{=}Pr(Class_i){*}IOU_{pred}^{truth}$$

A. Network Design

The network design is such that the initial convolutions layers and the fully connected layers extract the features and predict the output respectively. The network takes its inspiration from GoogLeNet model which is used for image classification. The network consists of 24 convolutional layers and 2 fully connected layers, and uses a 1 $\times$ 1 reduction layer followed by 3 $\times$ 3 convolutional layers. The final output of the YOLOv3 network is a list of bounding boxes and their associated class probabilities. The full network is shown in Figure 3.

III. Testing and Results

We trained and tested the YOLO and Fast RCNN model on COCO dataset, which is a large-scale image recognition dataset for object detection, segmentation, and captioning tasks

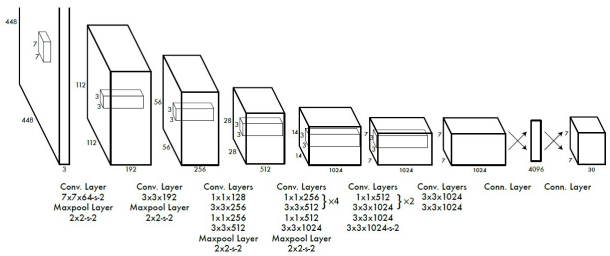


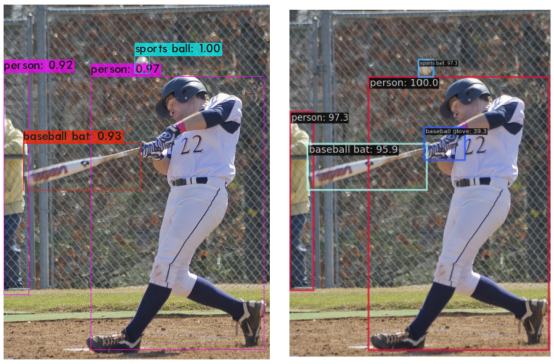
Fig. 1: Network Design for YOLO

and contains over 330,000 images, each annotated with 80 object categories, by using pre-trained weights. The findings are presented as shown below.

From Table 1, it can be observed that Fast-RCNN is almost 3 times faster than YOLO. However, YOLO seems better at detecting smaller objects because it considers the entire image as a single entity and uses a grid-based approach to object detection. It is evident in Fig. 4 (a) and (b), where bottle is detected by YOLO but not by Fast-RCNN.

TABLE I: Comparison of Timing Performance

Test Case Image	MODEL	
	Time Taken (ms), YOLO	*Time Taken (ms), R-CNN*
000000000001.jpg	714.91	341.07
000000000016.jpg	718.24	231.36
000000000019.jpg	906.07	257.97
000000000069.jpg	701.44	254.62
000000007039.jpg	695.96	258.87
000000007101.jpg	903.55	228.40
000000014350.jpg	695.47	243.87
000000014358.jpg	736.06	226.32
000000014390.jpg	705.83	229.94
Mean time	753.05	252.49



(a) YOLO (b) Fast-RCNN

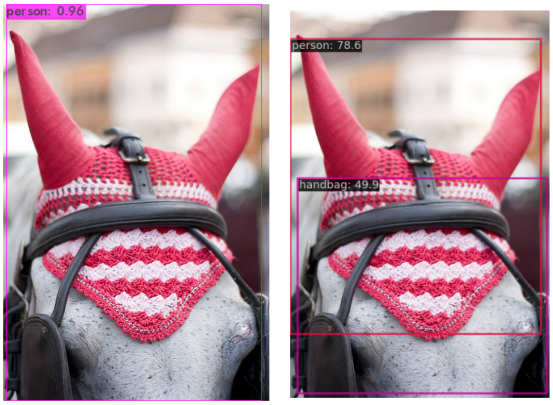
Fig. 2: Comparison between YOLO and R-CNN

(a) YOLO (b) Fast-RCNN

Fig. 3: Comparison between YOLO and R-CNN

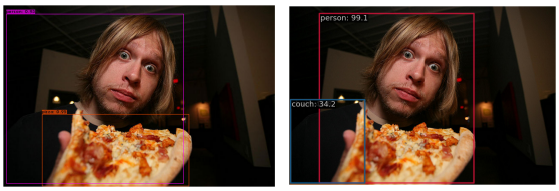
(a) YOLO (b) Fast-RCNN

Fig. 4: Comparison between YOLO and R-CNN



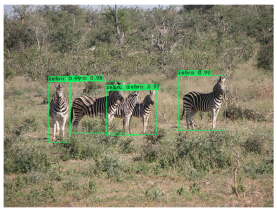
(a) YOLO (b) Fast-RCNN

Fig. 5: Comparison between YOLO and R-CNN

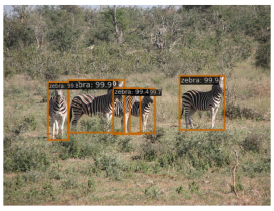


(a) YOLO (b) Fast-RCNN

Fig. 6: Comparison between YOLO and R-CNN

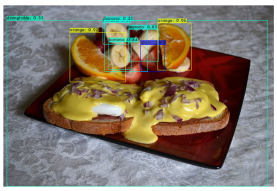


(a) YOLO

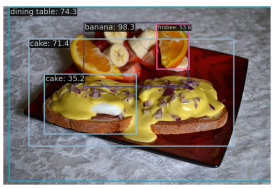


(b) Fast-RCNN

Fig. 7: Comparison between YOLO and R-CNN



(a) YOLO



(b) Fast-RCNN

Fig. 8: Comparison between YOLO and R-CNN

(a) YOLO

(b) Fast-RCNN

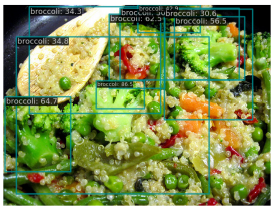
Fig. 9: Comparison between YOLO and R-CNN

(a) YOLO

(b) Fast-RCNN

Fig. 10: Comparison between YOLO and R-CNN

(a) YOLO



(b) Fast-RCNN

Fig. 11: Comparison between YOLO and R-CNN

(a) YOLO

(b) Fast-RCNN

Fig. 12: Comparison between YOLO and R-CNN

IV. Conclusions

We review YOLO, a simple unified model for object detection which can be trained on full images and thus generalise to new domains making it ideal for applications that rely robust object detection. Its comparison with RCNN shows that perfoms better classification but is somewhat slower.

Acknowledgement

We would like to thank Indian Institute of Technology for giving us an opportunity to take the course EE769 Introduction to Machine Learning. We would like to thank Prof. Amit Sethi for guiding us academically throughout the course. Lastly, we would like to thank all the people enrolled in this course who made learning interactive and competitive.

References

[1] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi"You Only Look Once : Unified, Real-Time Objection Detection," University of Washington, Allen Institute for AI, Facebook AI Research, arXiv: 1506.02640v5 9 May 2016.