Case Study: State-of-the-Art Object Detection Techniques

Liubov Tovbin
Computer Engineering Department
San Jose State University San Jose, CA
liubov.tovbin@sjsu.edu

***Abstract*—Object detection is a crucial task for many artificial intelligence applications. Ability to identify one or several objects and pinpoint their location on a picture is the fundamental component of computer vision. In this work, the author takes an opportunity to explore the topic of object detection in depth. R-CNN (Regional CNN) and YOLO (You Only Look Once) object detection techniques are the milestones in recent years' research. This paper discusses the ideas behind R-CNN and two methods that build upon it, Fast R-CNN and Faster R-CNN. YOLO is currently the fastest method that suits for real-time object detection applications. The paper provides a closer look at YOLO method's architecture. Also, it gives an overview of the various performance evaluation metrics and benchmarks accepted among researchers.**

Keywords*— *Object Detection, R-CNN, Fast R-CNN, Faster R-CNN, YOLO

I. Introduction

Object detection means localization and classification of objects on a picture. Given an image, an object detection system should be able to determine the essence of objects depicted on the image and their spatial relations. The task of object detection is a part of computer visoin which is a more complicated task who's ultimate goal is to mimic and, in some circumstances, outperform human vision abilities.

In a recent decade, the research on object detection techniques made significant advantages. Two factors contributed to this technology leap. First, computational powers grew up. Second, the amount of data that is available for training and testing models multiplied enormously. Three accessible image databases are available today for researchers: ImageNet, PASCAL VOC and MS COCO. Every year, various challenges on datasets from these databases are organized. The next section provides an overview of the popular benchmarks for competitions and introduces performance evaluation metrics for object detection techniques. The rest of the paper is organized as follows.

Section III gives an overview of the breakthrough technologies on object detection: R-CNN, its derivatives, Fast R-CNN, and Faster R-CNN, as well as the YOLO object detection method, that is so fast that can be used in real-time object detection applications.

Section IV provides a closer look at YOLO's architecture and training methods.

Section V concludes this paper and talks about a few methods that were not discussed here.

II. Algorithm Evaluation Methods

Several evaluation benchmarks for object detection algorithms are popular among researchers — for example, ImageNet, the database that contains more than 14 million indexed images [1]. From the year 2010, the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) is held every year. One more popular dataset for performance evaluation of object detection algorithms is PASCAL Visual Object Classes (VOC) dataset [2]. PASCAL stands for Pattern Analysis, Statistical modeling and Computational Learning. The challenges on this dataset were held from 2008 to 2012 years. MS COCO (Microsoft Common Objects in Context) is another large-scale object detection dataset [3]. It contains more than 200k labeled images. Numerous competitions on this dataset are held as well.

Object detection task combines two objectives. First, is a classification problem: to determine whether or not an object is present on a picture. Second, is to locate the object. It is important to remember that an image may contain objects of different classes. The objects may not be evenly distributed among the classes, meaning there might be more cats than bicycles, for example. Consequently, *accuracy* is not appropriate as an evaluation metric as it only measures the percentage of the correct predictions [2].

Both classification and localization tasks should be assessed independently. Also, the evaluation metrics should not be dependent on the algorithm for detection. So for the classification evaluation, the "confidence scores" for each class in an image should be provided. The higher the confidence score, the higher the probability that an object belongs for a specific class. As for detection evaluation, a list of bounding boxes locations around the object should be provided with a confidence score for each box [2].

The following metrics are used to assess the performance of object detection algorithms.

Precision (P), is a ratio of true positive predictions (TP) over a total number of positive predictions.

$$P = \frac{TP}{TP+FP}$$

where *FP* stands for false positive.

Recall (R), or Sensitivity is a ratio of true positive predictions over a total number of positive instances in a dataset.

$$R = \frac{TP}{TP+FN}$$

where *FN* stands for false negative.

Precision and recall are not independent metrics. The precision-recall curve is used to calculate an *Average Precision* (AP):

$$AP = \frac{1}{11} \sum_{R} P\,max(R)$$

where *Pmax(R)* is a maximum precision at the recall *R* that is greater than a threshold. *AP* is calculated for each class. The sum is over eleven evenly separated values of R:

$$0.0,\ 0.1,\ 0.2,\ 0.3,\ \ldots,\ 1.0$$

A similarity score between the predicted bounding box and the actual location should be calculated to assess an object localization algorithm. Jaccard index, or Intersection over Union (IoU), is a metric to measure the similarity between two sets.

$$J(A,\ B) = \frac{A \cap B}{A \cup B}$$

The bounding box with IoU that is greater than a predefined *IoU threshold* is considered a correct prediction.

Since the outcome of the prediction depends on IoU threshold, precision and recall values in the same experiment would be different if IoU threshold changes. Same goes for the average precision score. To account for these facts, *mean Average Precision* (mAP) was introduced. The mAP is calculated over all classes and over the all IoU thresholds that are required by a competition.

Next section presents an overview of the several state-of-the-art methods for object detection.

III. Object Detection Techniques Overview

A. R-CNN

R-CNN stands for Regions with Convolutional Neural Network features. Before, object detection systems utilized CNN and sliding window approach. In a sliding window, a filter designed to recognize a specific feature runs across the entire image in even steps. This procedure is time-consuming.

The idea behind R-CNN is to speed up the process by proposing regions on the image that should be fed into a CNN for feature detection. The algorithm was developed in 2014 by UC Berkeley researchers [4]. The authors used a Selective Search [5] as a region proposal method, but they mention that R-CNN technique is not tied to any particular region proposal method.

The Selective Search generates about 2000 regions (bounding boxes), each of them potentially contains a distinct object. In the next step, a deep CNN is used to extract features from each region. Finally, the extracted features are fed into a linear SVM for final object classification.

B. Fast R-CNN

Fast R-CNN is an improvement proposed later in 2015 by the author of R-CNN [6]. As the name suggests, the Fast R-CNN aims to improve the speed of R-CNN as well as accuracy.

The idea behind the Fast R-CNN is that first, the entire image is fed into a CNN to produce a convolutional feature map. Then, the boundaries of predefined regions are used to extract feature vector from the feature map for each region.

C. Faster R-CNN

The authors of Faster R-CNN [7] rightly noticed that region proposal operation is a bottleneck for the previous object detection algorithms. Their solution was to utilize Region Proposal Network (RPN) to speed up the Fast R-CNN. The main idea was to combine RPN for region proposal and Fast R-CNN for object detection into one network. The convolutional features are shared between these two tasks which allow achieving the goal "nearly cost-free."

Faster R-CNN can process 5 frames per second, which is a strong push towards real-time object detection.

D. YOLO, You Only Look Once

Previous methods required fine-tuning to make a final prediction regarding the object's location. YOLO system, as the name suggests, accepts an entire image as input only once [8]. YOLO stands for "You Only Look Once." This system for real-time object detection was developed in 2016 by University of Washington researchers [8].

The following section takes a closer look at YOLO's system architecture.

IV. You-Only-Look-Once Object Dtection Method

The YOLO system has three hyperparameters: S, B, and C.

- S is the size of a grid cell
- B is the number of bounding boxes that each cell predicts
- C is the number of possible classes to classify an object on the image

The first step is to divide an input image into cells of a size $S \times S$. Each cell is responsible for the prediction of B bounding boxes and C conditional probabilities, one for every possible class. For every bounding box, five parameters get predicted:

- Two numbers for the coordinates of the box's center that represent an offset from a grid cell location. The numbers are parametrized to fall into $(0, 1)$ interval.
- Two numbers for box's width and height that are normalized by the entire image dimensions and fall into $(0, 1)$ interval as well.
- One number for the confidence score which is an IoU ratio.

The authors evaluated the YOLO system on PASCAL VOC dataset which has 20 classes, thus $C = 20$. The other two hyperparameters were set to $S = 7$ and $B = 2$. The predictions are encoded in tensor:

$$S \times S \times (5 * B + C) = 7 \times 7 \times 30$$

where number 5 represents five predictions for each bounding box mentioned above: two coordinates, width, height, and IoU score.

The system architecture is a CNN of 24 convolutional layers each followed by a max-pooling layer and 2 fully connected layers at the end. The convolutional layers perform the initial image processing and extract features. Afterward, two fully connected layers generate the final prediction.

An important technical nuance is that a slightly different architecture is used to pre-train the system. The authors pretrained their system on ImageNet dataset using only 20 convolutional layers, followed by an average-pooling layer and one fully connected layer. After pre-training, they add another four convolutional and one fully connected layers with randomly initialized weights.

The activation function in all layers except the last one is a *leaky rectified linear function,* $\theta(s)$:

$$\theta(s > 0) = s$$

$$\theta(s \le 0) = 0.1s$$

The activation function in the last layer is a linear function

The optimization algorithm aims to minimize the sum of squared errors since it is "easy to optimize."

YOLO is a very fast algorithm comparing to others. It can process 45 frames per second that make it suitable for real-time object detection. The Fast YOLO, which is a smaller version of original YOLO architecture, can process 155 frames per second. This speedup comes in exchange of mAP score, 52.7% versus 63.4% in the original YOLO system as measured on PASCAL dataset. However, the improvement makes the Fast YOLO the fastest object detection method [8].

V. Conclusion

In this paper, the author reviewed state-of-the-art technologies for object detection and gave an overview of image databases available for researchers. The accepted performance evaluation metrics were discussed as well.

R-CNN and YOLO object detection methods are the milestones in recent years' research. The Fast YOLO, a smaller variation of YOLO, can process up to 155 frames per second, which makes it the fastest object detection method today. However, it has limitations: detection of small objects. Timely detection of a small object might be a crucial task. For example, an airplane vision system must be able to detect a bird in the sky. An airport cleaning robot must be able to detect a small piece of derby on a runway. P. Pham et al. [12] evaluate YOLO, SSD (Single Shot Multibox Detector), and Faster R-CNN on small object detection at real-time.

Two additional object detection methods are left for the future review. One is the Single Shot Multibox Detector (SSD), introduced in 2015 [10], and another one is Mask R-CNN, introduced in 2018 [11]. SSD achieves the rate of 58 frames per second and outperforms the Fastest R-CNN on PASCAL VOC. Mask R-CNN, which extends the Faster R-CNN, is "simple, flexible and general" as its authors suggest.

References

[1] http://image-net.org/about-overview

[2] M. Everingham et al., "The PASCAL Visual Object Classes Challenge: A Retrospective," *Springer Science+Business,* Media New York 2014

[3] T. Lin et al. "Microsoft COCO: Common Objects in Context," Available online: https://arxiv.org/pdf/1405.0312.pdf, accessed on May 9, 2019.

[4] R. Girshick, J. Donahue, T. Darrell, J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation Tech report (v5)," Available online: https://arxiv.org/pdf/1311.2524.pdf, accessed on May 8, 2017.

[5] J. Uijlings, K. van de Sande, T. Gevers, A. Smeulders. "Selective search for object recognition." *IJCV,* 2013.

[6] R. Girshick, "Fast R-CNN," Available online: https://arxiv.org/pdf/1504.08083.pdf, accessed on May 8, 2019.

[7] S. Ren, K. He, R. Girshick, J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," Available online: https://arxiv.org/pdf/1506.01497.pdf, accessed on May 8, 2019.

[8] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," Available online: https://arxiv.org/pdf/1506.02640v5.pdf, accessed on May 8, 2019.

[9] P. Pham et al, "Evaluation of Deep Models for Real-Time Small Object Detection," In D. Liu et al. (Eds.): *ICONIP 2017, Part III, LNCS* 10636, pp. 516–526, 2017. https://doi.org/10.1007/978-3-319-70090-8_53

[10] W. Liu et al, "SSD: single shot multibox detector.", In: B.Leibe, J.Matas, N. Sebe, M. Welling, (eds.) *ECCV 2016. LNCS*, vol. 9905, pp. 21–37. Springer, Cham (2016). doi:10. 1007/978-3-319-46448-0 2

[11] K.g He, G. Gkioxari, P. Dollar, R. Girshick, "Mask R-CNN," Available online: https://arxiv.org/pdf/1703.06870.pdf, accessed on May 10, 2019.

[12] P. Pham et al, "Evaluation of Deep Models for Real-Time Small Object Detection," In D. Liu et al. (Eds.): *ICONIP 2017, Part III, LNCS* 10636, pp. 516–526, 2017 https://doi.org/10.1007/978-3-319-70090-8_53