Preliminary Faster R-CNN Summary

Shishir Jakati
University of Massachusetts, Amherst
sjakati@umass.edu

Abstract

Detection and recognition are two important computer vision tasks that have been accelerated by deep learning. One major advancement in the recognition and detection pipeline has been the use of region proposal networks (RPN), which predict object position and objectness score simultaneously. Faster R-CNN[2] connects this RPN with fine tuned object detection layers to create an end-to-end trainable object detection and recognition pipeline.

1. Network Structure

1.1. Region Proposal Networks

The main purpose of RPNs is to produce a collection of bounding boxes which presumably identify an object present within the image. To do this, the RPN is trained as a classifier, and doubly as a regressor, which operates on the feature map produced by the initial convolutional layers in the Faster R-CNN architecture. The RPN is modeled with a fully convolutional network; with the goal that the computation may be shared with the To generate the collection of bounding boxes, the RPN utilizes a sliding window which then proceeds to feet this activated output to a "box-regression layer" and a "box-classification layer."

1.1.1 Translation-Invariant Anchors

When performing the sliding window search over the activated output feature map, the network must handle multiple region proposals. The specific number of these proposals is based entirely on the specified anchors, and their relative sizes. As presented in the paper, there are $k = 9$, predefined anchors used as reference for each sliding window.

These predefined anchors have the property of being translation invariant. This is the property that states that "If one translates an object in an image, the proposal should translate and the same function should be able to predict the proposal in either location." Ultimately, these scaled and translation-invariant anchors are called "Pyramid of Anchors."

1.1.2 Object Classifier and Box Regressors

As stated before, the *k* proposals are then fed into the sibling layers. The box regression layer is a fully connected layer that is trained to refine the proposed object boxes. These proposed boxes are originally scaled to one of the predefined anchors, and then transformed to better fit the object itself. Ultimately, there are *4k* possible points for the box regression layer to refine in one single sliding window.

Object classification is another responsibility which has been given to one of the sibling layers. In contrast to classical object detection, in which a network determines the specific class which an object belongs, this sibling layer simply provides the probabilities that a proposal is an object or not. This results in a total of *2k* objectness scores in one sliding window.

1.2. Layer Sharing

The detection network utilized by the framework is Fast R-CNN. This detection network is trained in unison with the RPN.

1.3. Training

The training procedure for the Faster R-CNN network is focused on end-to-end training using stochastic gradient descent, and is compartmentalized into specific training procedures for the RPNs and object classification networks.

The loss function for an RPN is defined as:

$$L(p_i, t_i) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*)$$

One of the most important terms in this loss function is the p_i term. This is the probability that a proposal is an object. This is directly correlated to the p_i^* term, which is the ground truth term of that anchor being a box; it is trivially defined to be 1 if the box is an object and 0 if the box is not an object.

To mitigate the problem of seperate convergence between the RPN and detection network, Faster R-CNN is trained using an alternating training process. First the RPN

is trained, which then produces proposals to train the detection network. This network is used to initialize the RPN and the process is continually repeated.

When training, many proposals will be produced for the same proposed object. To remove redundant proposals, the training procedure utilizes a 0.7 Intersection over Union (IoU) score as the Non Max Suppression (NMS) threshold. Following NMS, the top-*N* ranked proposals are used for detection.

2. Results

Faster R-CNN was evaluated on the PASCAL VOC 2007 detection benchmark, using a pre-trained ImageNet network. Additionally, Faster R-CNN was evaluated on MS COCO with alterations to the training routine.

2.1. PASCAL VOC

The goal of the Pattern Analysis, Statistical Modelling, and Computational Learning Visual Object Classes (PASCAL VOC) challenge "is to recognize objects from a number of visual object classes in realistic scenes." The relevance of this dataset to text detection research is found in the spatially diversity of the various classes presented in the challenge. Many text detection problems come from the inconsistencies between segments of text.

Faster R-CNN, when using the ZF network with shared layers produced a mAP of **58.9**. During this evaluation, the number of proposals was limited to 300. This mAP was state of the art. Additionally, when trained using the MS COCO, PASCAL VOC 2007 and PASCAL VOC 2012 datasets, Faster R-CNN using VGG achieved a mAP of **78.8**, which again was state of the art. Exhaustive results of the Faster R-CNN performance can be found in *Section 4* of the original Faster R-CNN paper.

2.2. MS COCO

One major distinction between PASCAL VOC and MS COCO is the number of classes. While the PASCAL VOC challenge is based on 20 separate classes, MS COCO contains 80 separate classes in their "natural context." [1]

Using a VGG-16 net, Faster R-CNN achieved a object detection rate (%) of **42.7** and **21.9** on mAP@0.5 and mAP@[0.5, 0.95] respectively.

3. Relevance to Research

When considering the problem of low-resolution text detection, and the more general problem of text detection, many frameworks may be used to segment different components of the detection pipeline. Many proposed solutions utilize two separate networks; one for detection and one for recognition. Other proposed solutions train object detection classifiers using words as classes. The Faster R-CNN proposal offers a computationally attractive solution, which is consequently compact.

Additionally, the Faster R-CNN framework offers an adaptable foundation for different object detection networks to be attached to the novel RPNs. When considering the problem of low-resolution detection, the Faster R-CNN framework allows for new generative networks, purpose built for low-resolution tasks, to become a component of the detection pipeline.

Overall, the Faster R-CNN framework introduced the novel RPNs, and performed well above previous state of the art detection pipelines. It may be adapted and modified to work well with low-resolution and character detection tasks.

References

[1] K. Gauen, R. Dailey, J. Laiman, Y. Zi, N. Asokan, Y. Lu, G. K. Thiruvathukal, M. Shyu, and S. Chen. Comparison of visual datasets for machine learning. In *2017 IEEE International Conference on Information Reuse and Integration (IRI)*, pages 346–355, Aug 2017. 2

[2] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 91–99. Curran Associates, Inc., 2015. 1