Faster R-CNN with RoI Refinement

Peng Yuan(pengy@stanford.edu), Yangxin Zhong(yangxin@stanford.edu), Yang Yuan(yyuan16@stanford.edu)

Introduction

Object detection is a computer vision task that aims to detect instances of semantic objects of certain classes in digital images (and videos). Given an image, object detection system detects what objects are in it and where they locate. It plays an important role in face detection, self-driving cars, video surveillance and many other applications.

In this project, we investigate the class of object detection algorithm that makes Region of Interest (RoI) proposals first and then classifies object in the proposal regions and pinpoints the bounding box for each object.

Previous Object Detection Models:

- R-CNN[1], Fast R-CNN[2]: Independent RoI Alg.
- Faster R-CNN[3]: RoI proposing within network
- Previous models suffer from using very rough RoIs for classifying and bounding box regression.

Inspired by Recurrent Neural Network (RNN), we propose model to get iteratively better RoI proposals through multiple refining iterations or through a LSTM model. We base our model on the state-of-the-art Faster R-CNN[3].

Dataset

We will use PASCAL VOC 2007 dataset[4] to evaluate our models, which has 5011 images in the training set and 4952 images in the test set. There are 20 object classes in VOC 2007 including: person, bird, cat, car, bicycle, chair… On each image, objects are represented by their ground truth class labels along with bounding boxes.

We will use mean Average Precision (mAP) metric to evaluate detection performance. To get a more intuitive understanding of the performance of our model and its iterative refining effect on bounding box regression, we will also visualize the predicted boxes at each iterative step on the original images.

Models

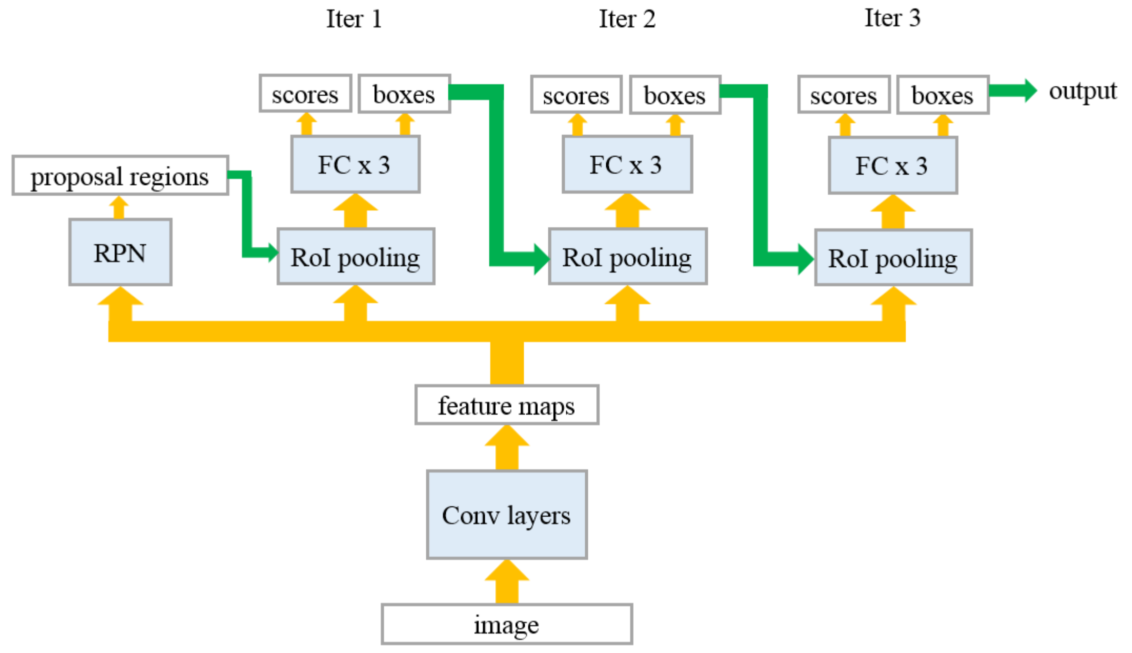

Fig. 1 Faster R-CNN with Iterative RoI Refinement

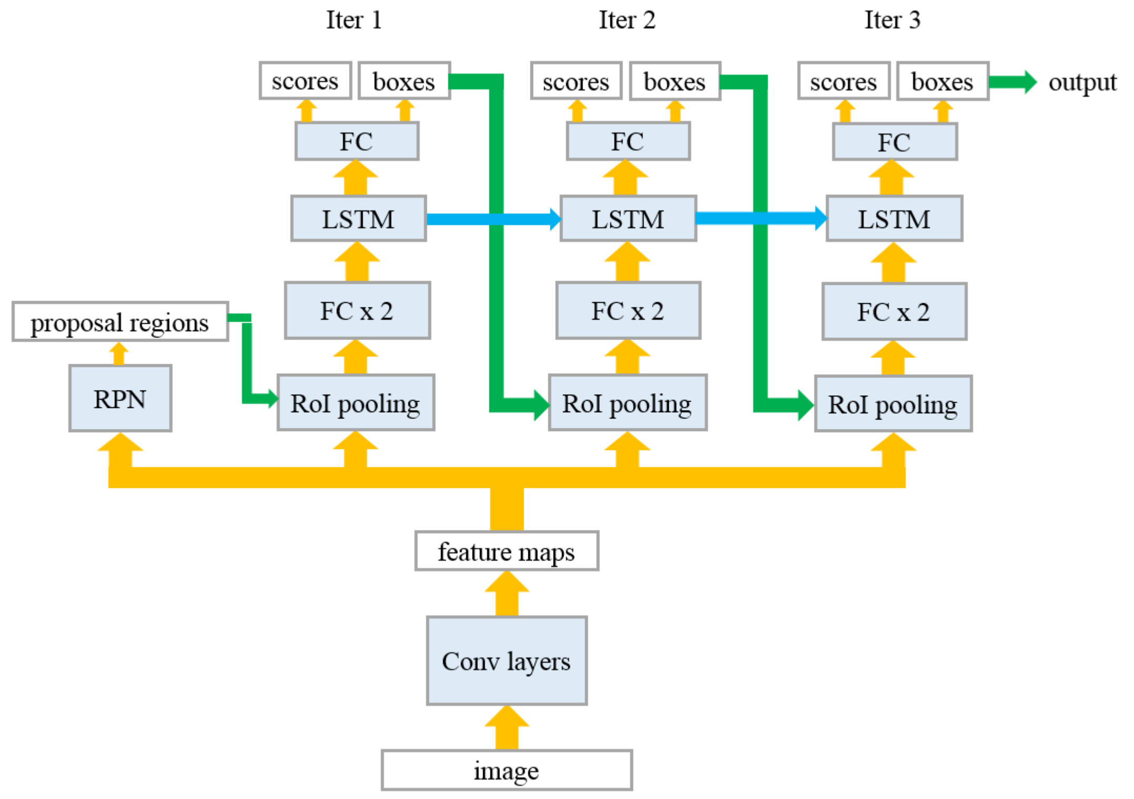

Fig. 2 Faster R-CNN with LSTM RoI Refinement

Faster R-CNN with Iterative RoI Refinement:

$$\begin{matrix} f = VGG(image) \\ RoI_0 = RPN(f) \end{matrix} \quad \Rightarrow \quad \text{for } i = 1, 2, ...T \begin{cases} r_i = RoIPooling(f, RoI_{i-1}) \\ scores_i,\ boxes_i = FC^3(r_i) \\ RoI_i = boxes_i \\ loss_i = loss_{cross-entropy}(scores_i, c_i) \\ \qquad + \lambda[c_i \neq bg]loss_{smooth_{L_1}}(boxes_i, b_i) \end{cases} \Rightarrow \begin{cases} output = scores_T, boxes_T \\ loss_{multi} = \frac{1}{T}\sum_{i=1}^{T} loss_i \end{cases}$$

Faster R-CNN with LSTM RoI Refinement :

$$\begin{matrix} f = VGG(image) \\ RoI_0 = RPN(f) \end{matrix} \quad \Rightarrow \quad \text{for } i = 1, 2, ...T \begin{cases} r_i = RoIPooling(f, RoI_{i-1}) \\ a_i = FC^2(r_i) \\ h_i = LSTM(h_{i-1}, a_i) \\ scores_i,\ boxes_i = FC(h_i) \\ RoI_i = boxes_i \\ loss_i = loss_{cross-entropy}(scores_i, c_i) \\ \qquad + \lambda[c_i \neq bg]loss_{smooth_{L_1}}(boxes_i, b_i) \end{cases} \Rightarrow \begin{cases} output = scores_T, boxes_T \\ loss_{multi} = \frac{1}{T}\sum_{i=1}^{T} loss_i \end{cases}$$

References

[1] R. Girshick, J. Donahue, T. Darrell, and J. 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.

[2] R. Girshick. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, pages 1440–1448, 2015.

[3] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems, pages 91–99, 2015.

[4] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. International journal of computer vision, 88(2):303–338, 2010.

Evaluation

We trained and tested our models on PASCAL VOC 2007 dataset, with learning curves in Fig. 3 and 4, and test results in Table 1. In addition, Fig. 5 shows the visualization of our method.

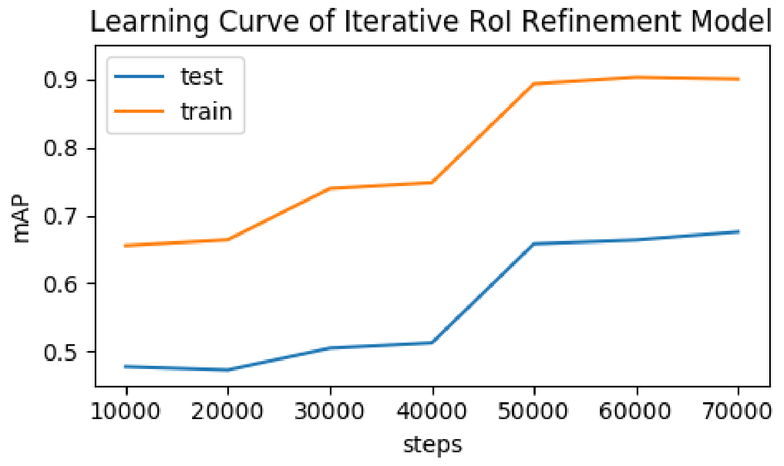

Fig. 3 Learning curve of Faster R-CNN with Iterative RoI Refinement

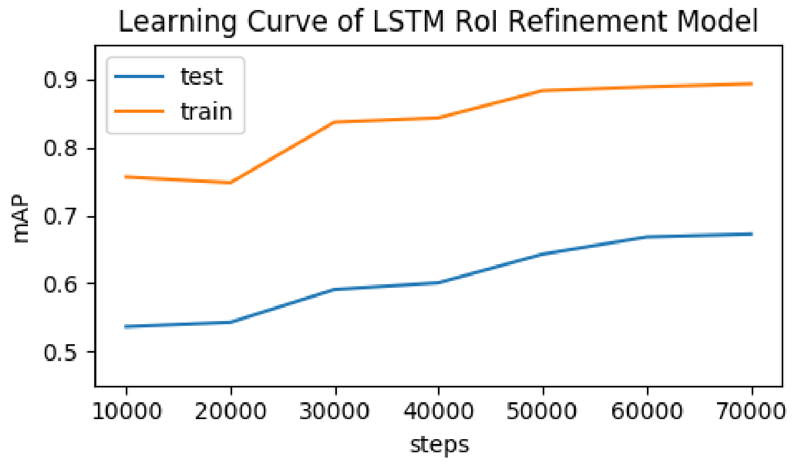

Fig. 4 Learning curve of Faster R-CNN with LSTM RoI Refinement

Table 1 Best performance (mAP) of different models on test set with Iteration number T = 1,2, 3

Vanilla	Iterative RoI Refinement		LSTM RoI Refinement	
67.02%	T = 2	67.53%	T = 2	67.12%
	T = 3	67.56%	T = 3	67.23%
	T = 4	67.42%	T = 4	67.09%

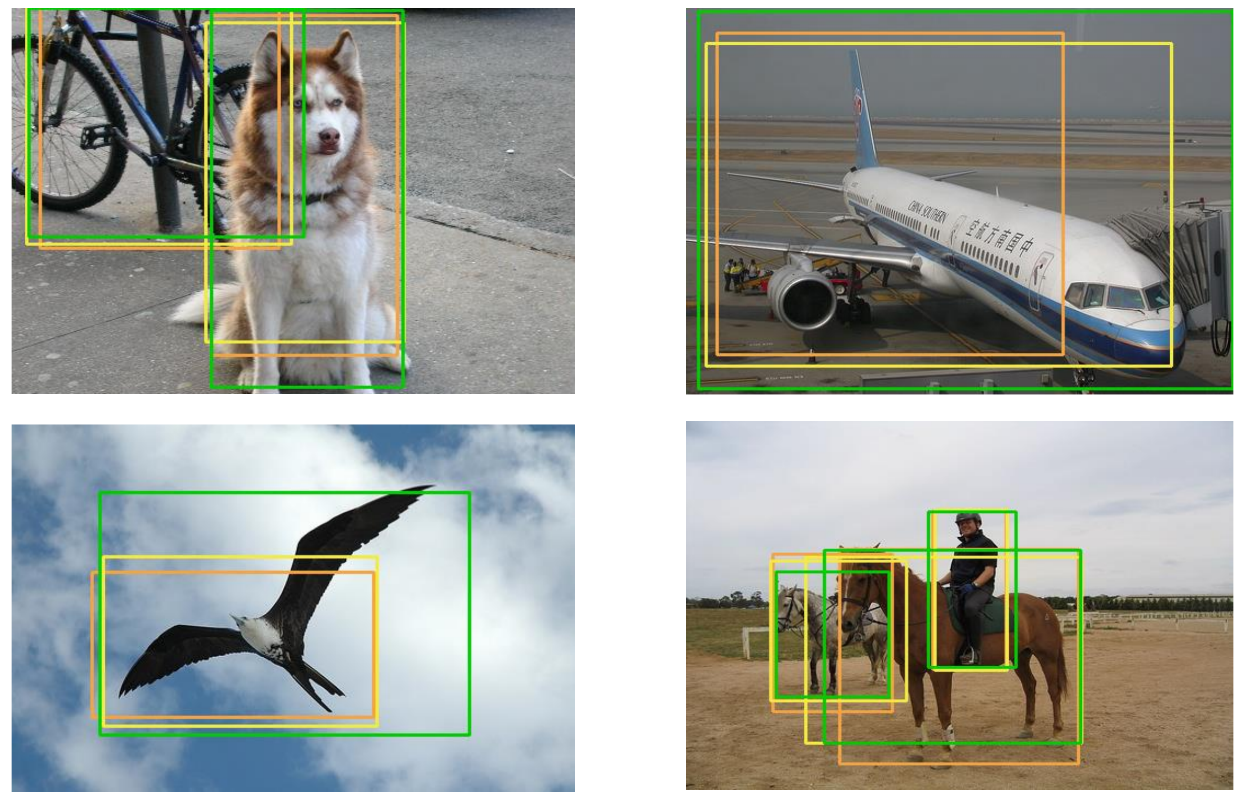

Fig. 5 Iterative refinement effect of our model (Iter 1, 2, 3 = orange, yellow, green)

Discussion:

- As shown in Fig. 5, the iterative refinement shows significant improvement in boxing the objects in some images.
- Both iterative and LSTM RoI refinement shows best mAP at 3 iterations; the iteration number is not "the more the better".
- Both learning curves show large gap between learn and test mAP, this suggests the models may have overfitting issue.
- Iterative RoI refinement works better than LSTM RoI refinement, this may because such iterative length is too short for LSTM to show its strength.

Future Works:

- Currently the RoI pooling layer can not be backpropagated between different iterations, our major future work is to make this layer capable of backpropagation between iterations.