

Object detection using faster-RCNN

Group 3 Xin Li, Sihan Wang, Yawen Zhao, Chen Zhao

UC San Diego

Abstract

Object detection is a computer technology using image processing and computer vision which deals with detection of instances about semantic objects of certain classes both in digital images and videos in the physical world. In this project, our group is aiming at solving object detection with large varieties in size, angle, posture. Instead of using inefficient region-based convolutional neural networks(R-CNNs), with producing candidate bounding box with CPU and passing CNN one by one, we choose Faster-RCNN^[1], a method approaching real-time rates and ignores the time spent on region proposals and thus realizes the speed improvement. We will test this method on Pascal VOC and the results will be shown below.

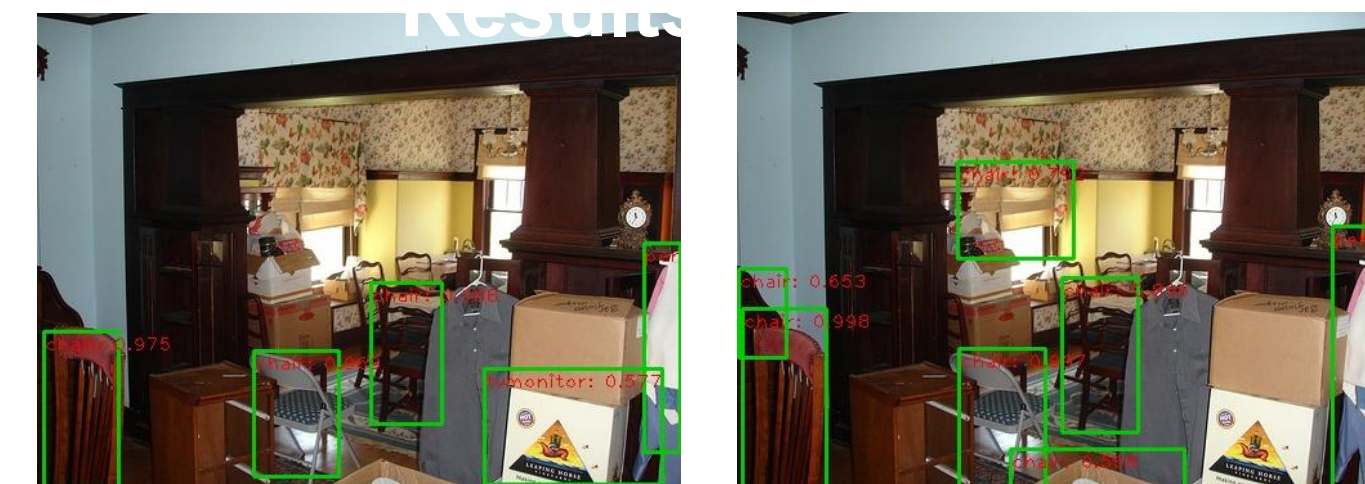
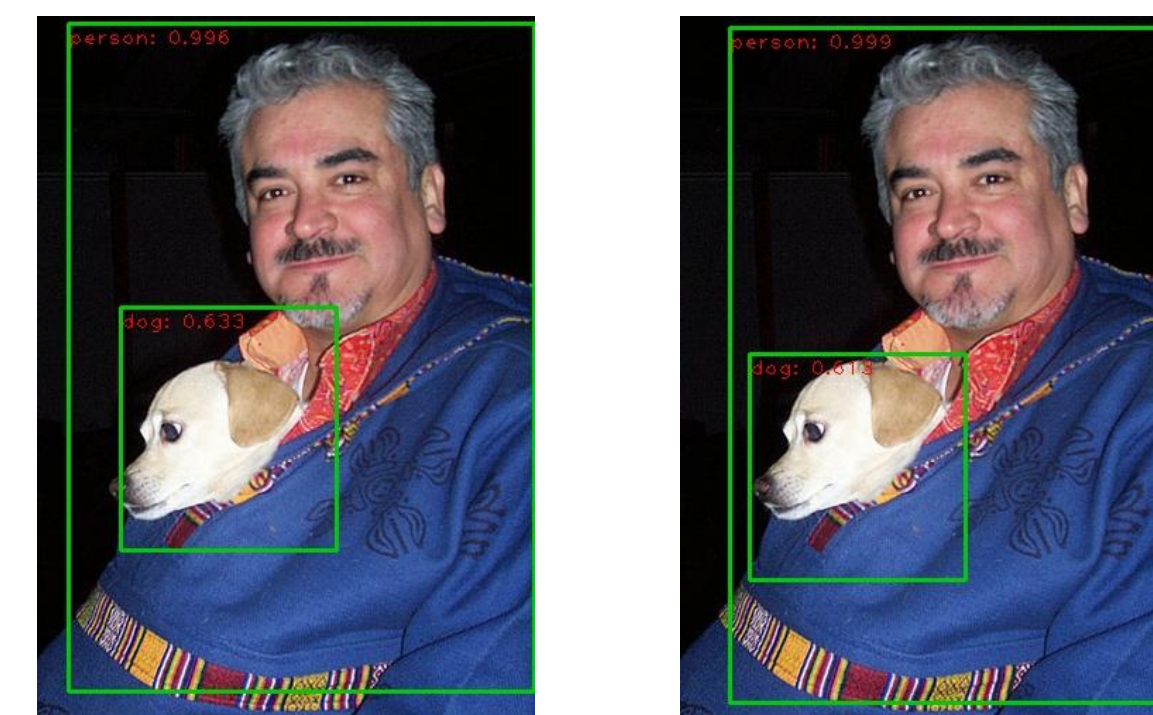
Methods

	GPU	Batch Size	Lr	Lr_decay	Epoch	Time
VGG16	1	1	1e-3	8	12	18hrs
ResNet101	1	1	1e-3	5	7	13hrs

We trained VGG16 and ResNet101 model on pascal 2012 dataset using 1080Ti^[3]. Due to GPU limit, we trained with batch size of 1. ResNet101 takes nearly 2 hours per epoch and VGG16 takes around 1.5 hrs per epoch.

Since the annotated test set for VOC 2012 has not been released, we are using VOC 2007 test set for benchmark. The advantage is that we can compare our result to the result trained on VOC 2007 training set.

Results

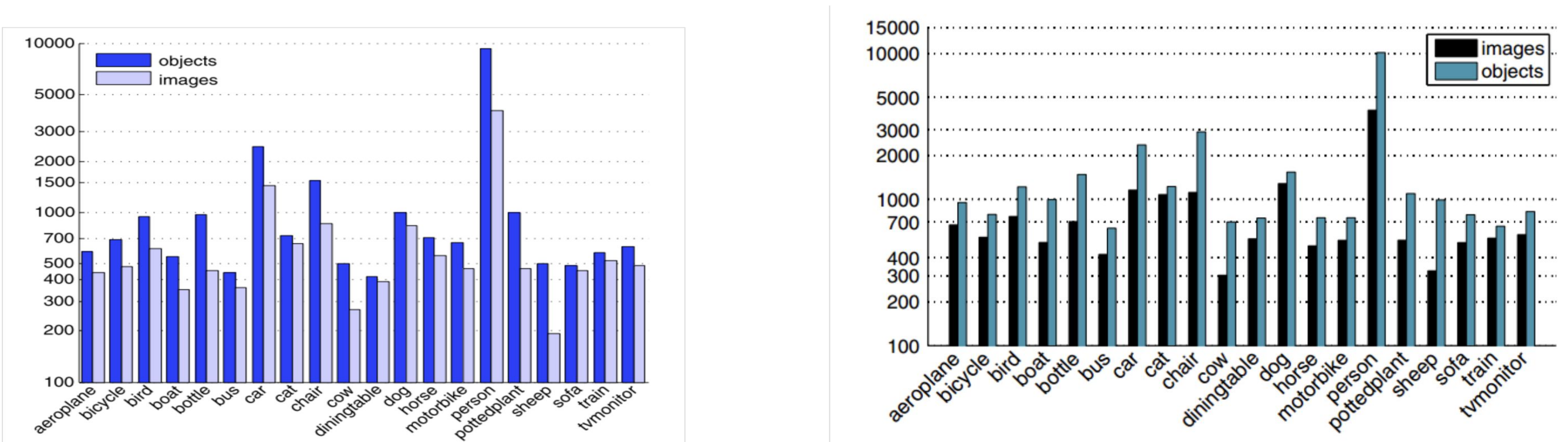


Above are object detection results of our trained models on two different images. The left side is result of VGG16, the right side represents the result of ResNet101. We use these two set of parameters for comparison because they have similar result in the first image. Both trained models are able to detect most of the objects. For VGG16, it has a misclassification on the television, and for ResNet101, it has a misclassification at the top. But comparing the bounding box location vs the real object, it is not hard to see ResNet101 is slightly better in precisely detecting project.

This observation is also reflected in mean average precision(mAP) of VOC 2007 test set. The resulted mAP on both models are better than those trained on VOC 2007 training set VGG16(0.724 vs 0.701), ResNet101(0.769 vs 0.75) which makes sense because VOC 2012 is a more complexed dataset.

Datasets

We use the datasets from Pascal VOC^[2] challenge. This dataset keeps updated till 2012. We choose the training dataset from the Pascal VOC 2012 and testing dataset from the Pascal VOC 2007. The left picture shows the details about the VOC 2007 and the tight one shows details about VOC 2012.



	train		val		test		Total	
	images	objects	images	objects	images	objects	images	objects
2007	2501	6301	2510	6307	4952	12032	9963	24640
2012	5717	13609	5823	13841	11540	27540	23080	54900
total	8218	19910	8333	20148	16492	39482	33043	79540

Models

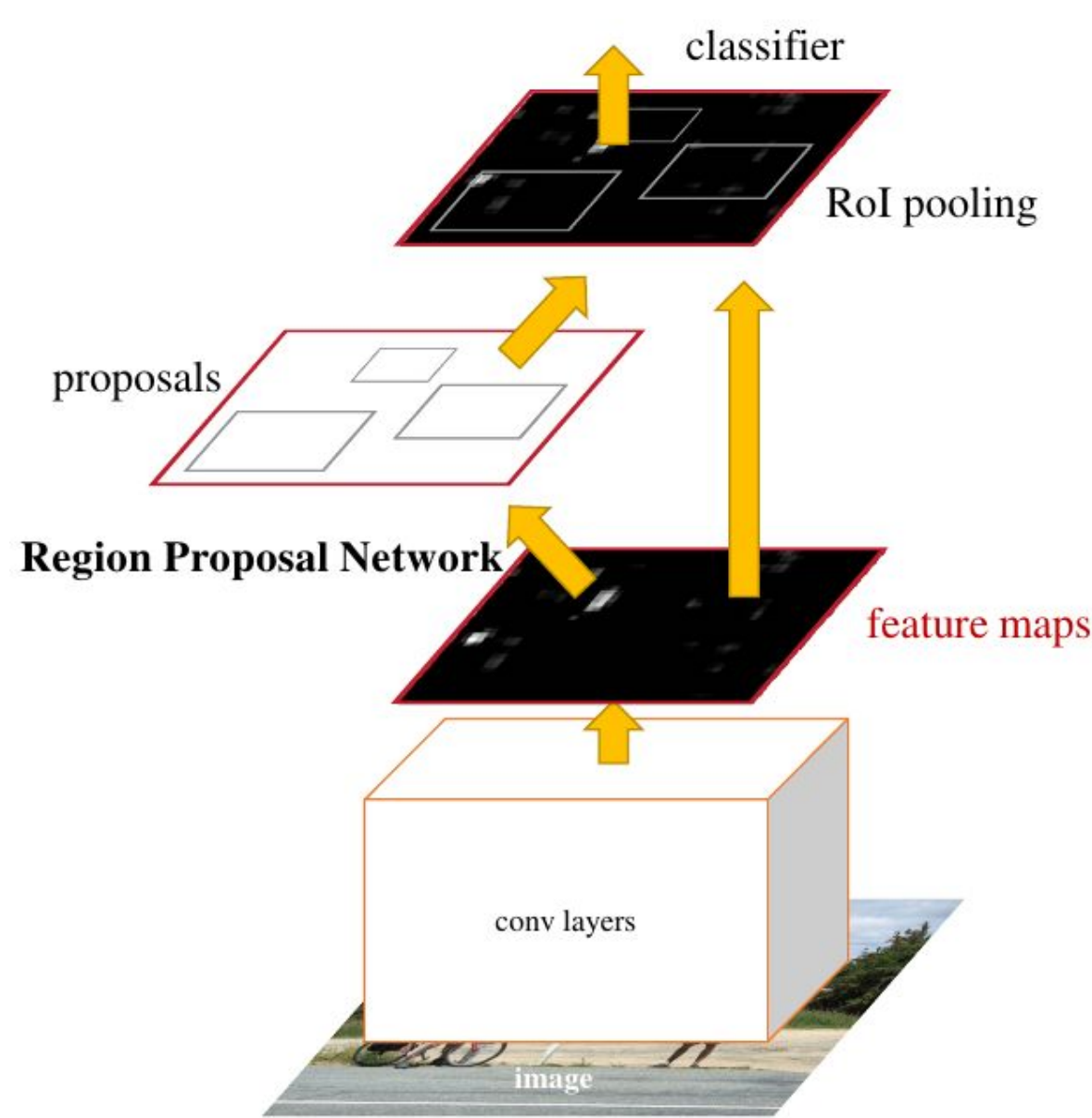
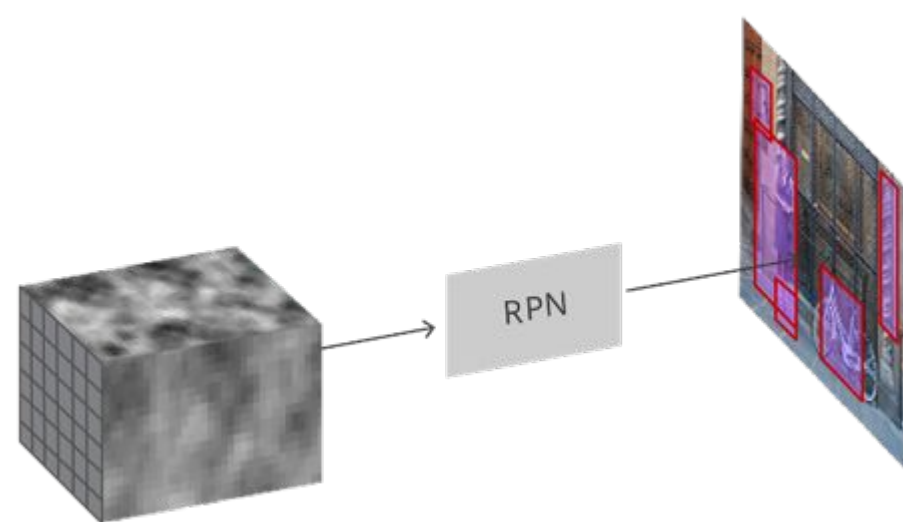
Faster RCNN consists of mainly 4 parts as follows:

- (1) Conv Layers
- (2) Region Proposal Network(RPN)
- (3) ROI(region of interest) Pooling
- (4) Classification

First, it uses a convolutional neural network to extract the features in an image.

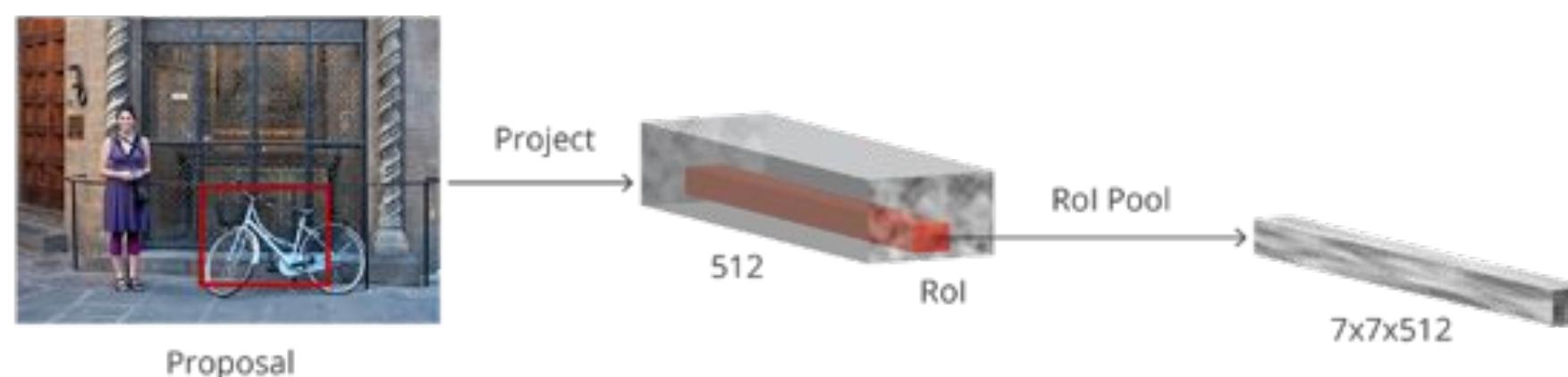


Intuitively, each image often contains many different objects and a large background, but only those features of objects are useful. So how to find those features for objects became the main problem.



In Faster-RCNN, they proposed a new idea to find those features for objects. They introduced another neural network which is called the "Region Proposal Network", RPN for short. RPN is a fully connected network, and it takes feature map and the possible size of bounding box as input. Then it outputs a set of good proposals for objects. The proposals are in the original image, and each of them is a 4D vector including center point (x, y), width and height.

Next, with all the proposals, Faster RCNN does a "region of interest pooling" to the original feature map. Now we have all the features that are highly likely to be from objects.



Using all the proposed features, we could finally use a classifier to find out the classes they belong to. The classifier is RCNN.

This RCNN has two different goals:

1. Classify proposals into one of the classes, plus a background class. This part outputs the probability for each class.
2. Better adjust the bounding box for the proposal according to the predicted class. This part output the best adjustment of our proposal, including box center, width and height adjustment.

Conclusions

In this project, we are using Faster-RCNN to realize object detections among Pascal VOC. Two different models, VGG16 and ResNet101, are trained and tested. Two models reach similar results in object detection. However, ResNet101 with more layers, takes longer time to train and achieves better results which is reflected by observation and also shows in mAP. Besides, some objects can not be detected or will be misclassified with the model we trained currently. For our future work, if we have more resources for training, we may train more models with different parameters to see if there will be any improvement. Besides, there are some popular directions to improve Faster-RCNN recently. First, improving feature extraction network, such as using IncRes v2, ResNet. Second, improving RPN, reduce the number of proposal and improve the accuracy. Third, improving ROI Pooling, with multi-task benefits algorithm, MASK R-CNN and multi-layer roi-pooling, DeepText.

References

1. <https://arxiv.org/abs/1506.01497>
2. <http://host.robots.ox.ac.uk/pascal/VOC/>
3. <https://github.com/jwyang/faster-rcnn.pytorch/tree/pytorch-1.0>