Paper summary: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks

Source: Excerpts from "1506.01497v3.pdf" by Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun

1. Paper Summary

This paper introduces **Faster R-CNN**, a novel object detection system that significantly improves upon previous state-of-the-art methods like SPPnet and Fast R-CNN by addressing the computational bottleneck of region proposal generation. The core innovation is the **Region Proposal Network (RPN)**, a fully convolutional network that shares full-image convolutional features with the detection network. This allows for "nearly cost-free" region proposals and enables a unified, end-to-end deep learning system for object detection to achieve near real-time performance (5fps with VGG-16) while maintaining or improving accuracy on benchmark datasets like PASCAL VOC and MS COCO. The RPN effectively acts as an "attention" mechanism, telling the subsequent detection network "where to look."

2. Main Themes and Innovations

2.1. Addressing the Region Proposal Bottleneck

- **Problem Identified:** Previous state-of-the-art object detection networks (e.g., Fast R-CNN) achieved near real-time speeds for the detection phase, but the **region proposal algorithms (e.g., Selective Search, EdgeBoxes) became the primary computational bottleneck.** These methods were typically CPU-bound and significantly slower (e.g., Selective Search: 2 seconds/image, EdgeBoxes: 0.2 seconds/image) than the GPU-accelerated detection networks.
- **Solution: Region Proposal Network (RPN):** The paper proposes an algorithmic change: "computing proposals with a deep convolutional neural network." This RPN is designed to share convolutional layers with the detection network, making the marginal cost of computing proposals "small (e.g., 10ms per image)" at test-time.

2.2. Region Proposal Network (RPN) Design

- **Fully Convolutional Network (FCN):** The RPN is an FCN that "simultaneously predicts object bounds and objectness scores at each position." It operates by sliding a small network over the convolutional feature map generated by shared layers.
- **Anchors for Multi-Scale/Aspect Ratio Handling:**RPNs efficiently predict proposals with a wide range of scales and aspect ratios using novel **"anchor" boxes**.
- An anchor is "centered at the sliding window in question, and is associated with a scale and aspect ratio."
- By default, the system uses "3 scales and 3 aspect ratios, yielding k = 9 anchors at each sliding position."
- This "pyramid of regression references" avoids the need for time-consuming image pyramids or filter pyramids.
- This design is crucial for "sharing features without extra cost for addressing scales."

- **Translation Invariance:** A key property of the anchor-based approach is its translation invariance. If an object moves, the RPN can still predict its location using the same function, which is not guaranteed by methods like MultiBox that use k-means to generate fixed anchors. This also leads to a significantly smaller model size compared to MultiBox.
- **Loss Function:** For training RPNs, a multi-task loss function is used, similar to Fast R-CNN. It combines:
- **Classification Loss (Lcls):** Binary classification (object vs. not object) for each anchor.
- **Regression Loss (Lreg):** Bounding box regression for positive anchors.
- The regression is "bounding-box regression from an anchor box to a nearby ground-truth box." Importantly, "a set of k bounding-box regressors are learned. Each regressor is responsible for one scale and one aspect ratio, and the k regressors do not share weights."

2.3. Unified Network and Feature Sharing

- **Faster R-CNN Architecture:** The entire system, Faster R-CNN, is composed of two modules: the RPN for proposing regions and the Fast R-CNN detector for object detection, both sharing a common set of convolutional layers. "Using the recently popular terminology of neural networks with 'attention' mechanisms, the RPN component tells the unified network where to look."
- **Alternating Training Scheme:** To enable shared features and optimize both RPN and Fast R-CNN, a pragmatic **"4-step alternating training algorithm"** is adopted:

1. Train RPN (initialized with ImageNet pre-trained model).
2. Train Fast R-CNN using proposals from Step 1 RPN (initialized with ImageNet pre-trained model).
3. Fine-tune RPN, but **fix the shared convolutional layers** and only train RPN-specific layers.
4. Fine-tune Fast R-CNN, but **fix the shared convolutional layers** and only train Fast R-CNN specific layers.

- This scheme leads to "a unified network with convolutional features that are shared between both tasks."

2.4. Performance and Efficiency

- **Speed:** With the very deep VGG-16 model, Faster R-CNN achieves a "frame rate of 5fps (including all steps) on a GPU," demonstrating its practicality for real-time object detection. The RPN itself adds only "10ms" to the computation time due to shared features. With the ZF net, it achieves 17fps.
- **Accuracy:** Achieves "state-of-the-art object detection accuracy on PASCAL VOC 2007, 2012, and MS COCO datasets."
- On PASCAL VOC 2007 test set, RPN+VGG shared features yield 69.9% mAP (compared to 66.9% for SS+Fast R-CNN).
- It performs well with "only 300 proposals per image," significantly fewer than traditional methods (e.g., 2000 for Selective Search), which also reduces downstream costs.
- **Robustness to Hyperparameters:** The choice of anchor scales and aspect ratios (3 scales, 3 ratios) is effective, and the model shows insensitivity to the balancing parameter λ in the loss function within a wide range.
- **Impact of RPN Components:**Removing the classification layer (no NMS/ranking) degrades mAP, showing that "the cls scores account for the accuracy of the highest ranked proposals."
- Removing the regression layer (proposals become anchor boxes) also drops mAP, indicating that "the high-quality proposals are mainly due to the regressed box bounds."

2.5. Competition Success and Generalizability

- **ILSVRC and COCO 2015 Competitions:** "Faster R-CNN and RPN are the foundations of the 1st-place winning entries in several tracks" including ImageNet detection, localization, COCO detection, and COCO segmentation.
- **Broader Impact:** The RPN and Faster R-CNN frameworks "have been adopted and generalized to other methods, such as 3D object detection, part-based detection, instance segmentation, and image captioning." It has also been built into "commercial systems such as at Pinterests."
- **Scalability with Deeper Networks:** The RPN's learning capabilities mean it "completely learn[s] to propose regions from data, and thus can easily benefit from deeper and more expressive features (such as the 101-layer residual nets adopted in [18])."

2.6. Comparison with One-Stage Detectors

- The paper compares Faster R-CNN (two-stage: proposal + detection) with a simulated one-stage system (e.g., OverFeat style using dense sliding windows).
- The two-stage system (Faster R-CNN) consistently outperforms the one-stage system in mAP (e.g., 58.7% vs. 53.9% with ZF model), justifying "the effectiveness of cascaded region proposals and object detection." The one-stage system is also slower due to more proposals.

2.7. Leveraging Large-Scale Data (MS COCO)

- Training on the large-scale MS COCO dataset significantly improves performance on PASCAL VOC.
- A model trained directly on COCO (without PASCAL VOC fine-tuning) yields 76.1% mAP on PASCAL VOC 2007, outperforming VOC07+12 trained models (73.2%).
- Fine-tuning a COCO-trained model on VOC data further boosts mAP to 78.8% on PASCAL VOC 2007, showing a "5.6%" increase due to the extra COCO data.

3. Key Facts and Figures

- **Core Innovation:** Region Proposal Network (RPN) sharing convolutional features with the detection network.
- **Speed (VGG-16):** 5 frames per second (fps) end-to-end on a GPU.
- **RPN Contribution to Time:** Only 10ms due to shared features.
- **Number of Proposals:** State-of-the-art accuracy achieved with just 300 proposals per image (vs. 2000 for Selective Search).
- **Anchor Design:** Default 3 scales (128^2, 256^2, 512^2 pixels) and 3 aspect ratios (1:1, 1:2, 2:1) per sliding window location, yielding 9 anchors (k=9).
- **PASCAL VOC 2007 mAP (VGG-16):**Fast R-CNN + Selective Search: 66.9%
- Faster R-CNN (RPN+VGG, shared): 69.9% (trained on VOC07)
- Faster R-CNN (RPN+VGG, shared): 73.2% (trained on VOC07+12)
- Faster R-CNN (RPN+VGG, shared): **78.8%** (trained on COCO+07+12)
- **MS COCO test-dev mAP (VGG-16):**Fast R-CNN (SS): 39.3% mAP@0.5, 19.3% mAP@[.5, .95]
- Faster R-CNN (RPN): 42.1% mAP@0.5, 21.5% mAP@[.5, .95] (trained on COCO train)
- Faster R-CNN (RPN): 42.7% mAP@0.5, 21.9% mAP@[.5, .95] (trained on COCO trainval)
- **Competition Wins:** Basis for 1st place in ILSVRC and COCO 2015 object detection, localization, and segmentation.

- **Code Availability:** Publicly available on GitHub (MATLAB and Python implementations).