Study Notes for Faster R-CNN

Qinghong Lin
2018/10

Abstract

• Fast R-CNN reduce the running time except region proposal computation.
• RPN(*Region Proposal Network*) shares full-image convolutional features and nearly cost-free region proposals.
- RPN is FCN simultaneously predicts object bounds and objectness scores at each postion.
- RPN is trained **end-to-end** to generate high-quality region proposals.
• Merge RPN and Fast R-CNN into a single network by sharing their convolutional features
- using **"attention"** mechanism can tells the network where to look.

1.Introdution

• object detection are driven by region proposal methods and region-based convolutional neural networks.
- proposals are the test-time computational bottleneck in now.

• This paper show that computing proposals with a deep convolutional neural network where nearly **cost-free.**
• observation from *convolutional feature maps* used by region-based detectors, can also be used for generating region proposals.
• so construct an **RPN** by adding a few additional convolutional layers.
- RPN is a kind of *FCN* can be trained *end-to-end*.

• RPNs are effficiently predict region proposals with a wide range of scales and aspect ratios.
• **"anchor" boxes:** serve as references at multiple scales and aspect ratios.
- avoids enumerating images or filters of multiple scales or aspect ratios and benefits running speed.

• To unify RPNs with Fast R-CNN, propose **a training scheme** that **alternates** between fine-tuning for the region proposal task and then fine-tuning for object detection.

• RPNs completely learn to propose regions from data and easily benefit from deeper and more expressive features.

2.Related work

Object Proposals

• widely used methods mainly based on grouping super-pixels and sliding windows.
• Object proposal methods as external modules independent of the detectors.

Deep Networks for Object Detection

• *OverFeat method*: a FC layer is trained to predict the box coordinates for the localization task that assumes a single object.

•*MultiBox method:* generate region proposals from a network whose last FC layer simultaneously predicts multiple class-agnostic boxes, generalizing the "single-box" fashion of OverFeat.

• Shared computation of convolutions

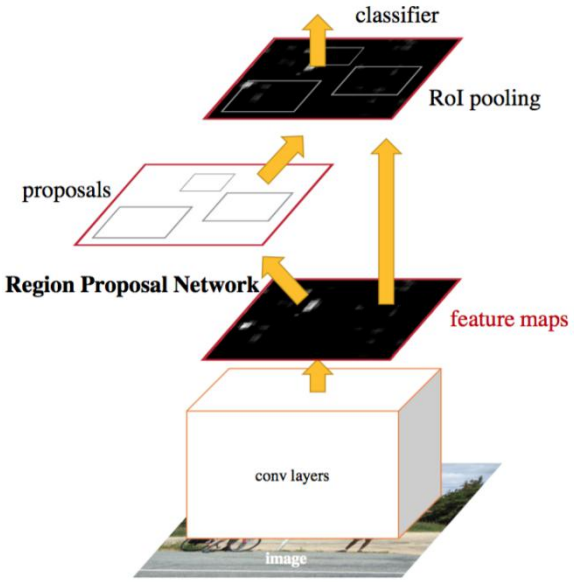
- The OverFeat computes from an image pyramid for classification, localization and detection.

- *Adaptively-sized pooling(SPP)*

3.Faster R-CNN

• Faster R-CNN contain two modules

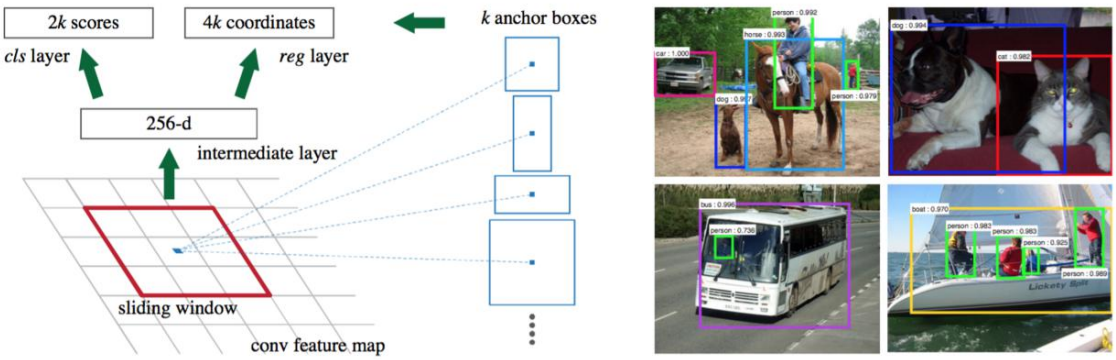
- first module is **a deep fully convolutional network** that proposes regions.
- second module is **Fast R-CNN detector** that uses the proposed regions.
- **'attention' mechanisms** tells the Fast R-CNN where to look.'



3.1.Region Proposal Networks

• RPN taks an image as input and outs a set of rectangular object proposals.

• model this process with **FCN** because our goal is to share computation with Fast R-CNN.



• Slide a small network over the conv feature map.

• **Small network process**

- input an n×n spatial windows of the conv feature map.
- each sliding window map to a lower-dim feature.
- feature go into two sibling FC layers:a box-regression layer and a box-classification layer.

3.1.1.Anchors

• *k* is the maximum possible proposals for each location

- *reg* has 4k outputs.
- *cls* has 2k outputs scores estimate probability of object or not object.

• **An anchor** is centered at the sliding window and associated with a scale and aspect ratio.

- using *3 scales and 3 aspect ratios*, *k=9* anchors at each sliding position.
- feature map of size *W*×*H*, has *WHk* anchors in total.

Translation-Invariant Anchors

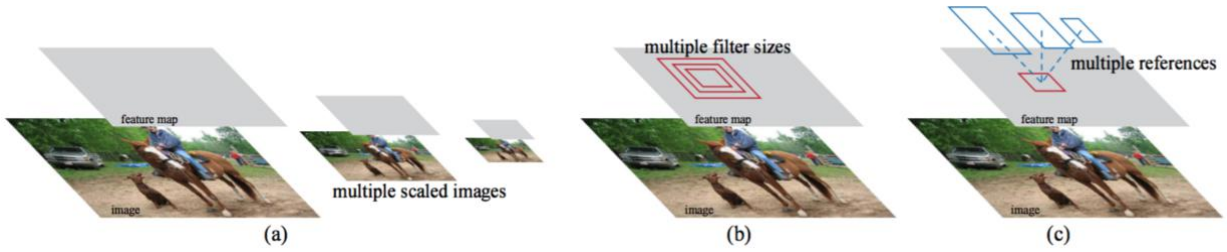
• Translation invariant: If one translates an object in an image, the proposal should translate and the same function should be able to predict the proposal in either location.

• this method also reduces the model size.

Multi-Scale Anchors as Regression References

• anchors presents a novel scheme for addressing multiple scales(and aspect ratios).

There are two popular ways



• **Based on image pyramids**

- images are resized at multiple scales and feature maps are computed for each scale.
- useful but time-consuming.

• **use sliding windows of multiple scales**(pyramid of filters)

- using different filter size.
- most cost-efficient.

• **anchor-based method**(pyramid of ahchors)

- cost-efficient only rely on images and feature maps of a single scale.
- key component for **sharing features** without extra cost for addressing scales.

3.1.2.Loss Function

• assign a binary class label to each anchor for training RPNs.

(i)the anchor with the highest IoU with a ground-truth box.

(ii)an anchor has an IoU overlap higher than 0.7 with any ground-truth box.

using(i), (ii) may find no positive sample in some rare cases.

• Loss Function

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

i : index of an anchor in a mini-batch

pi: the predicted probability of anchor *i* being an object

*pi**: equal to 1 if anchor is positive and 0 when negative

ti: vector represent 4 parameterized coordinates of the bounding box.

*ti**: *ti* for the ground-truth box

- *Lcls*: log loss over two classes(object vs. not object)
- *Lreg(ti,ti*)=R(ti-ti*)*

R: the robust loss function(smooth L1) defined in

*pi*Lreg*: regression loss only for positive anchors

{pi}: the outputs of the cls

{ti}: the outputs of the reg

- two terms are normalized by *Ncls* and *Nreg and* weighted by a balancing parameter λ.

$$\begin{aligned} t_x &= (x - x_a)/w_a, \quad t_y = (y - y_a)/h_a, \\ t_w &= \log(w/w_a), \quad t_h = \log(h/h_a), \\ t_x^* &= (x^* - x_a)/w_a, \quad t_y^* = (y^* - y_a)/h_a, \\ t_w^* &= \log(w^*/w_a), \quad t_h^* = \log(h^*/h_a), \end{aligned} \quad (2)$$

- *x,y,w,h* denote the box's center coordinates and its width and height.
- *x, xa, x** are for the predicted box, anchor box and ground-truth box

• Our method for **bounding-box regression** can predict boxes of various sizes though the features are fixed.

3.1.3.Training RPNs

• RPN can be trained end-to-end by back-propagation and SGD.("image-centric")

• anchors will bias towards negative samples as they are dominate.

• randomly initialize layers by drawing weights from a zero-mean Gaussian distribution.

3.2.Sharing Features for RPN and Fast R-CNN

• paper **has described** how to train a region proposal network **without considering** how CNN use these proposals

• we need to develop a technique that **sharing convolutional layers** between the two network rather than learning two separate networks.

• Three ways for training networks with features shared

(i)Alternating training:

- first train RPN, and use the proposals to train Fast R-CNN.
- the network tuned by Fast R-CNN is then used to initialize RPN and iterate this process.

(ii)Approximate joint training:

- merge RPN and Fast R-CNN into one network during training
- each SGD iteration, the forward pass generates region proposals which are treated just like fixed, pre-computed proposals when training a Fast R-CNN detector.

- the backward propagation takes place as usual

- RPN loss and the Fast R-CNN loss are combined for the shared layers

- ignores the derivative w.r.t the proposal boxes' coordinates that are also network response.

(iii)Non-approximate joint training:

-The RoI pooling accepts the conv features and boxes as input so solver should involve gradients w.r.t the box coordinates.

-we need an RoI pooling layer that is differentiable w.r.t the box coordinates.

4-Step Alternating Training

1)train the RPN as 3.1.3

2)train Fast R-CNN using the proposals generated by RPN, two networks do not share now.

3)use Fast R-CNN to initialize RPN training, but we **fix** the shared convolutional layers and only **fine-tune** the layers unique to RPN.

4)keeping the shared conv layers **fixed** and **fine-tune** the unique layers of Fast R-CNN.

3.3.Implementation Details

• Multi-scale feature extraction(using an image pyramid) may improve accuracy but doesn't exhibit a good speed-accuracy trade-off.

• Paper's solution

- doesn't need an image pyramid or filter pyramid to predict regions of multiple scales, saving considerable running time.

- allows predictions that are larger than the underlying receptive field.

• cross-boundary anchors

- introduce large, difficult to correct error terms in the objective, and training doesn't converge.

- ignore all cross-boundary anchors so they don't contribute to the loss.

• non-maximum suppression(NMS) to reduce redundancy.

- not harm the ultimate detection accuracy but substantially reduces the number of proposals

• After NMS,use the top-*T* ranked proposal regions for detection.

4.Experiments

4.1 Experiments on PASCAL VOC

train-time region proposals method	# boxes	test-time region proposals method	# proposals	mAP (%)
SS	2000	SS	2000	58.7
EB	2000	EB	2000	58.6
RPN+ZF, shared	2000	RPN+ZF, shared	300	**59.9**
ablation experiments follow below				
RPN+ZF, unshared	2000	RPN+ZF, unshared	300	58.7
SS	2000	RPN+ZF	100	55.1
SS	2000	RPN+ZF	300	56.8
SS	2000	RPN+ZF	1000	56.3
SS	2000	RPN+ZF (no NMS)	6000	55.2
SS	2000	RPN+ZF (no *cls*)	100	44.6
SS	2000	RPN+ZF (no *cls*)	300	51.4
SS	2000	RPN+ZF (no *cls*)	1000	55.8
SS	2000	RPN+ZF (no *reg*)	300	52.1
SS	2000	RPN+ZF (no *reg*)	1000	51.3
SS	2000	RPN+VGG	300	59.2

Ablation Experiments on RPN

• Show the effect of **sharing conv layers** between the RPN and Fast R-CNN detection network.

- (59.9% v.s. 58.7%)
- stop after the second step in the 4-step training process.
- proposal quality is improved when the detector-tuned features are used to fine-tune the RPN.

• Show **the RPN's influence on training** the Fast R-CNN detection network.

- 300 proposals: (58.7% v.s. 56.8%) loss for inconsistency between the training/testing proposals.
- 100 proposals: (58.7% v.s. 55.1%) indicating that the top-ranked RPN proposals are accurate.
- 6000 proposals: **NMS** not harm for the mAP and may reduce false alarms.

• Show when the ***cls*** is removed

- 55.8%(N = 1000) nearly unchange, but degrades to 44.6%(N =100)
- Show the *cls* scores account for the accuracy of the highest ranked proposals.

• Show when the ***reg*** is removed

- mAP drops to 52.1%. show the high-quality proposals are mainly due to the regressed box bounds and anchor boxes not sufficient for multiple detection.

• Evaluate the effects of more powerful networks,

- Improves from 56.8%(using RPN+ZF) to 59.2%(using RPN+VGG)
- we expect RPN+VGG to be better than SS.

Performance of VGG-16

• the results of VGG-16 for both proposal and detection

-PASCAL VOC 2007 test set

method	# proposals	data	mAP (%)
SS	2000	07	66.9†
SS	2000	07+12	70.0
RPN+VGG, unshared	300	07	68.5
RPN+VGG, shared	300	07	69.9
RPN+VGG, shared	300	07+12	**73.2**
RPN+VGG, shared	300	COCO+07+12	**78.8**

-PASCAL VOC 2012 test set

method	# proposals	data	mAP (%)
SS	2000	12	65.7
SS	2000	07++12	68.4
RPN+VGG, shared†	300	12	67.0
RPN+VGG, shared‡	300	07++12	**70.4**
RPN+VGG, shared§	300	COCO+07++12	**75.9**

• running time of the entire object detection system.

model	system	conv	proposal	region-wise	total	rate
VGG	SS + Fast R-CNN	146	1510	174	1830	0.5 fps
VGG	RPN + Fast R-CNN	141	**10**	47	**198**	**5 fps**
ZF	RPN + Fast R-CNN	31	**3**	25	**59**	**17 fps**

Sensitivities to Hyper-parameters

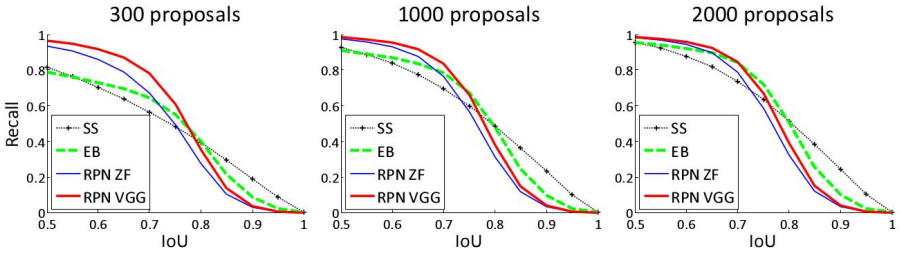
• different settings of anchors

-using anchors of multiple sizes as the regression references is an effective solution.

settings	anchor scales	aspect ratios	mAP (%)
1 scale, 1 ratio	128^2	1:1	65.8
	256^2	1:1	66.7
1 scale, 3 ratios	128^2	{2:1, 1:1, 1:2}	68.8
	256^2	{2:1, 1:1, 1:2}	67.9
3 scales, 1 ratio	$\{128^2, 256^2, 512^2\}$	1:1	**69.8**
3 scales, 3 ratios	$\{128^2, 256^2, 512^2\}$	{2:1, 1:1, 1:2}	**69.9**

Analysis of Recall-to-IoU

• recall of proposals at different IoU ratios with ground-truth boxes.



• RPN has a good mAP when using as few as 300 proposal mainly attributed to the *cls* term of the RPN.

One-Stage Detection vs. Two-Stage Proposal + Detection

• **OverFeat** is a ***one-stage, class-specific*** detection pipeline and **ours** is a ***two-stage cascade*** consisting of class-agnostic proposals and class-specific detections.

	proposals		detector	mAP (%)
Two-Stage	RPN + ZF, unshared	300	Fast R-CNN + ZF, 1 scale	58.7
One-Stage	dense, 3 scales, 3 aspect ratios	20000	Fast R-CNN + ZF, 1 scale	53.8
One-Stage	dense, 3 scales, 3 aspect ratios	20000	Fast R-CNN + ZF, 5 scales	53.9

• Justify the effectivenes of **cascaded** region proposals and object detection.

• one-stage system is slower as it has considerably more proposals to process.

4.2 Experiments on MS COCO

method	proposals	training data	COCO val		COCO test-dev	
			mAP@.5	mAP@[.5, .95]	mAP@.5	mAP@[.5, .95]
Fast R-CNN [2]	SS, 2000	COCO train	-	-	35.9	19.7
Fast R-CNN [impl. in this paper]	SS, 2000	COCO train	38.6	18.9	39.3	19.3
Faster R-CNN	RPN, 300	COCO train	41.5	21.2	42.1	21.5
Faster R-CNN	RPN, 300	COCO trainval	-	-	**42.7**	**21.9**

• RPN performs excellent for improving the localization accuracy at higher IoU thresholds.

Faster R-CNN in ILSVRC & COCO 2015 competitions

• Replacing VGG-16 with a ResNet-101, mAP from 41.5%/21.2% to 48.4%/27.2%.

4.3 From MS COCO to PASCAL VOC

• **Large-scale data** is of crucial importance for improving deep neural networks. and we concern how the MS COCO dataset can help with the detection performance on PASCAL VOC.

training data	2007 test	2012 test
VOC07	69.9	67.0
VOC07+12	73.2	-
VOC07++12	-	70.4
COCO (no VOC)	76.1	73.0
COCO+VOC07+12	**78.8**	-
COCO+VOC07++12	-	**75.9**

• Evaluate the COCO detection model on the PASCAL VOC dataset without fine-tuning because COCO are a superset of those on PASCAL VOC.

- 76.1% better than trained on VOC07+12(73.2%)

• Fine-tune the COCO detection model on the VOC dataset.

- 78.8% mAP
- show that the model trained on COCO+VOC has the best AP on PASCAL VOC2007.

5 Conclusion

• RPNs for efficient and accurate region proposal generation.

•By sharing convolutional features with the down-stream detection network, the region proposal step is nearly **cost-free**.

• Our method enables a unified, deep-learning-based object detection system to run ar near real-time frame rates.