Review: Faster R-CNN (Object Detection)

SH Tsang Follow
Sep 14, 2018 · 6 min read

In this story, **Faster R-CNN** [1–2] is reviewed. In the previous Fast R-CNN [3] and R-CNN [4], region proposals are generated by selective search (SS) [5] rather than using convolutional neural network (CNN).

In Faster R-CNN [1–2], **both region proposal generation and objection detection tasks are all done by the same conv networks.** With such design, object detection is much faster.

To know deep learning object detection well, as a series of objection detection approaches, if there is enough time, it is better to read R-CNN, Fast R-CNN and Faster R-CNN in order, to know the evolution of objection detection, especially why region proposal network (RPN) is existed in this approach. I suggest to read my reviews about them if interested.

As Faster R-CNN is a state-of-the-art approach, it is published as **2015 NIPS** paper and **2017 TPAMI** paper with more than **4000 and 800 citations respectively** when I was writing this story. (SH Tsang @ Medium)

. . .

What are covered

1. **Region Proposal Network (RPN)**
2. **Detection Network**
3. **4-Step Alternating Training**
4. **Ablation Study**
5. **Detection Results**

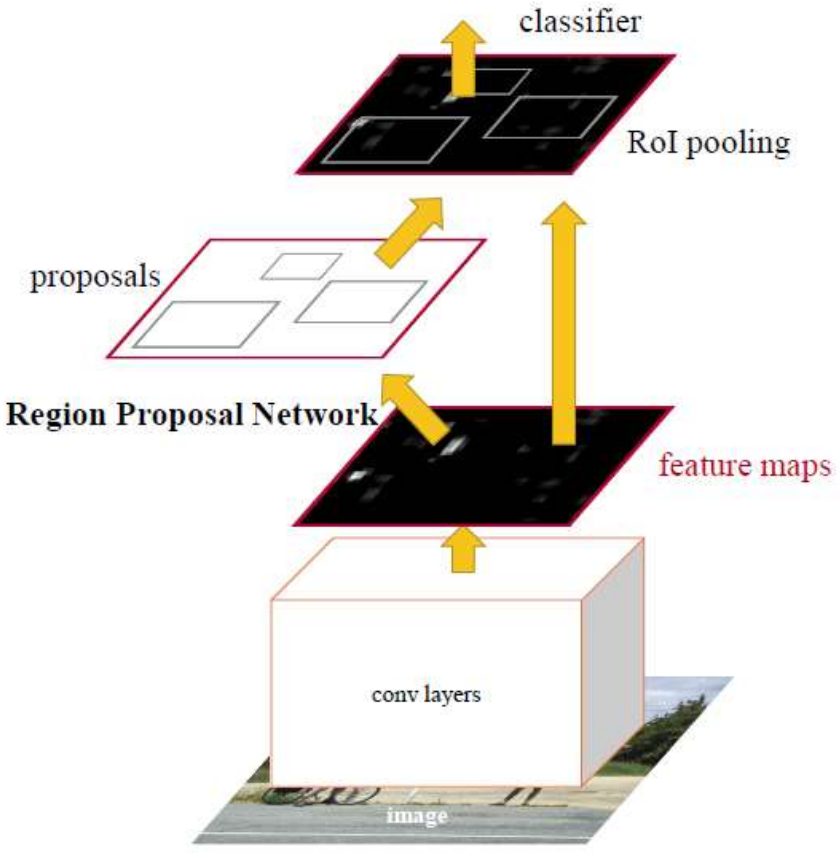
. . .

1. Region Proposal Network (RPN)

In brief, R-CNN [4] and Fast R-CNN [3] first generate region proposals by selective search (SS) [5], then a CNN-based network is used to classify the object class and detect the bounding box. (The main difference is that R-CNN input the region proposals at pixel level into CNN for detection while Fast R-CNN input the region proposals at feature map level.) **Thus, in R-CNN [4] and Fast R-CNN [3], the region proposal approach/network (i.e. SS) and the detection network are decoupled.**

Decoupling is not a good idea. Say for example, when SS has false negative, this error will hurt the detection network directly. It is better to couple them together such that they are correlated to each other.

In Faster R-CNN [1–2], RPN using SS [5] is replaced by RPN using CNN. And this CNN is shared with detection network. This CNN can be ZFNet or VGGNet in the paper. Thus, the overall network is as below:

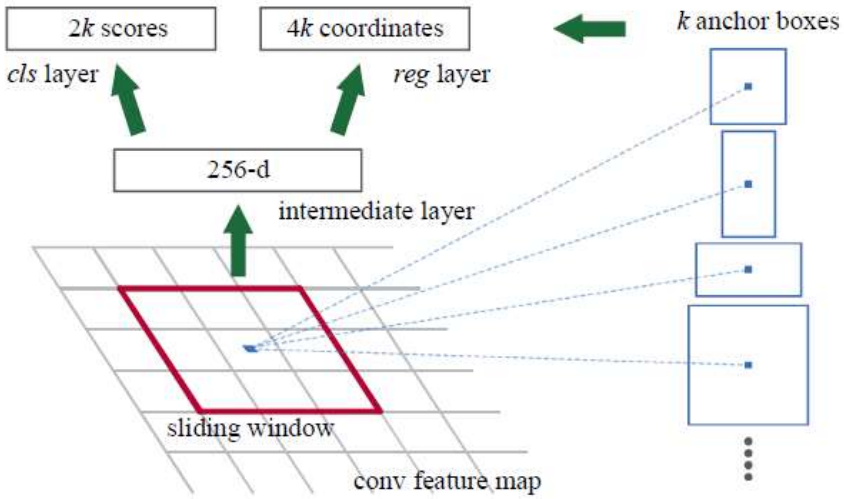


Faster R-CNN

1. First, the picture goes through conv layers and feature maps are extracted.
2. Then a **sliding window** is used in RPN for each location over the feature map.
3. For each location, **k (k=9) anchor** boxes are used (**3 scales of 128, 256 and 512, and 3 aspect ratios of 1:1, 1:2, 2:1**) for

generating region proposals.

4. A ***cls*** layer outputs *2k* scores **whether there is object or not** for *k* boxes.
5. A ***reg*** layer outputs *4k* for the **coordinates** (box center coordinates, width and height) of *k* boxes.
6. With a size of *W*×*H* feature map, there are *WHk* anchors in total.



The Output of RPN

The average proposal size for 3 scales of 128, 256 and 512, and 3 aspect ratios of 1:1, 1:2, 2:1 are:

Table 1: the learned average proposal size for each anchor using the ZF net (numbers for $s = 600$).

anchor	128^2, 2:1	128^2, 1:1	128^2, 1:2	256^2, 2:1	256^2, 1:1	256^2, 1:2	512^2, 2:1	512^2, 1:1	512^2, 1:2
proposal	188×111	113×114	70×92	416×229	261×284	174×332	768×437	499×501	355×715

Average Proposal Sizes

The loss function is:

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

RPN Loss Function

The first term is the classification loss over 2 classes (There is object or not). The second term is the regression loss of bounding boxes only when there is object (i.e. p_i* =1).

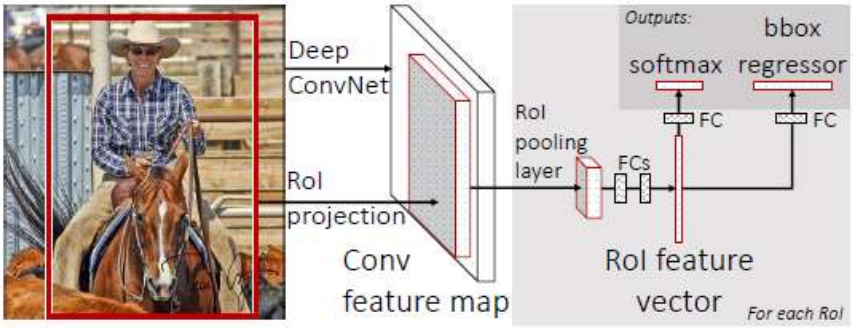
Thus, **RPN network is to pre-check which location contains object**. And **the corresponding locations and bounding boxes will pass to detection network** for detecting the object class and returning the bounding box of that object.

As regions can be highly overlapped with each other, non-maximum suppression (NMS) is used to reduce the number of proposals from about 6000 to N (N=300).

. . .

2. Detection Network

Except the RPN, the remaining part is similar to the Fast R-CNN. ROI pooling is performed first. And then the pooled area goes through CNN and two FC branches for class softmax and bounding box regressor. (If interested, please read my review about Fast R-CNN.)



Fast R-CNN Detection Network

. . .

3. 4-Step Alternating Training

Since the conv layers are shared to extract the feature maps with different outputs at the end, thus, training procedure is quite different:

1. Train (fine-tune) RPN with imagenet pre-trained model.
2. Train (fine-tune) a separate detection network with imagenet pre-trained model. (Conv layers not yet shared)
3. Use the detector network to initialize PRN training, fix the shared conv layers, only fine-tune unique layers of RPN.
4. Keep the conv layers fixed, fine-tune the unique layers of detector network.

. . .

4. Ablation Study

4.1. Region Proposal

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

As mentioned, with unshared conv layer (Only first 2 steps in alternating training), 58.7% mAP is obtained. **With shared conv layers, 59.9% mAP is obtained.** And it is better than prior arts SS and EB.

4.2 Scales and Ratios

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**

With 3 scales and 3 ratios, 69.9% mAP is obtained which is only little improvement over that of 3 scales and 1 ratio. But still 3 scales and 3 ratios are used.

4.3 λ in Loss Function

λ	0.1	1	10	100
mAP (%)	67.2	68.9	69.9	69.1

□□□□ **λ = 10 achieves the best result.**

. . .

5. Detection Results

5.1 PASCAL VOC 2007

method	# box	data	mAP	areo	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
SS	2000	07	66.9	74.5	78.3	69.2	53.2	36.6	77.3	78.2	82.0	40.7	72.7	67.9	79.6	79.2	73.0	69.0	30.1	65.4	70.2	75.8	65.8
SS	2000	07+12	70.0	77.0	78.1	69.3	59.4	38.3	81.6	78.6	86.7	42.8	78.8	68.9	84.7	82.0	76.6	69.9	31.8	70.1	74.8	80.4	70.4
RPN*	300	07	68.5	74.1	77.2	67.7	53.9	51.0	75.1	79.2	78.9	50.7	78.0	61.1	79.1	81.9	72.2	75.9	37.2	71.4	62.5	77.4	66.4
RPN	300	07	69.9	70.0	80.6	70.1	57.3	49.9	78.2	80.4	82.0	52.2	75.3	67.2	80.3	79.8	75.0	76.3	39.1	68.3	67.3	81.1	67.6
RPN	300	07+12	73.2	76.5	79.0	70.9	65.5	52.1	83.1	84.7	86.4	52.0	81.9	65.7	84.8	84.6	77.5	76.7	38.8	73.6	73.9	83.0	72.6
RPN	300	COCO+07+12	**78.8**	**84.3**	**82.0**	**77.7**	**68.9**	**65.7**	**88.1**	**88.4**	**88.9**	**63.6**	**86.3**	**70.8**	**85.9**	**87.6**	**80.1**	**82.3**	**53.6**	**80.4**	**75.8**	**86.6**	**78.9**

Detailed Results

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**

Overall Results

With training data using COCO, VOC 2007 (trainval) and VOC 2012 (trainval) dataset, 78.8% mAP is obtained.

5.2 PASCAL VOC 2012

method	# box	data	mAP	areo	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
SS	2000	12	65.7	80.3	74.7	66.9	46.9	37.7	73.9	68.6	87.7	41.7	71.1	51.1	86.0	77.8	79.8	69.8	32.1	65.5	63.8	76.4	61.7
SS	2000	07++12	68.4	82.3	78.4	70.8	52.3	38.7	77.8	71.6	89.3	44.2	73.0	55.0	**87.5**	80.5	80.8	72.0	35.1	68.3	**65.7**	80.4	64.2
RPN	300	12	67.0	82.3	76.4	71.0	48.4	45.2	72.1	72.3	87.3	42.2	73.7	50.0	86.8	78.7	78.4	77.4	34.5	70.1	57.1	77.1	58.9
RPN	300	07++12	70.4	84.9	79.8	74.3	53.9	49.8	77.5	75.9	88.5	45.6	77.1	55.3	86.9	81.7	80.9	79.6	40.1	72.6	60.9	81.2	61.5
RPN	300	COCO+07++12	**75.9**	**87.4**	**83.6**	**76.8**	**62.9**	**59.6**	**81.9**	**82.0**	**91.3**	**54.9**	**82.6**	**59.0**	**89.0**	**85.5**	**84.7**	**84.1**	**52.2**	**78.9**	65.5	**85.4**	**70.2**

Detailed Results

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**

Overall Results

With training data using COCO, VOC 2007 (trainval+test) and VOC 2012 (trainval) dataset, 75.9% mAP is obtained.

5.3 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**

Overall Results

42.1% mAP is obtained with IoU @ 0.5 using COCO train set for training.
21.5% mAP is obtained with IoU from 0.5 to 0.95 with step size of 0.05.

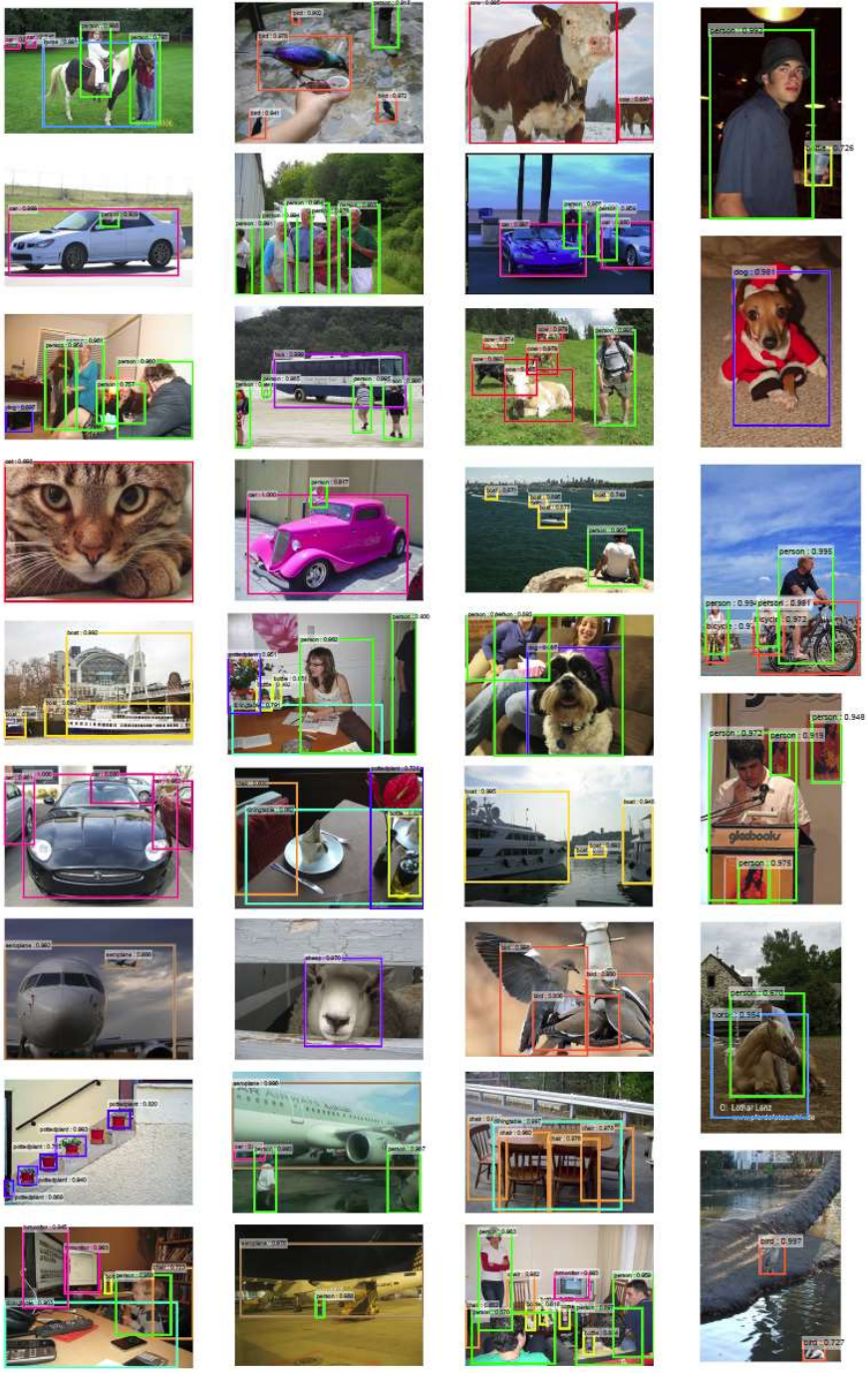
5.4 Detection Time

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**

Detection Time

Using SS as RPN and VGGNet as detection network: 0.5 fps / 1830ms
Using **VGGNet as RPN and detection network: 5fps / 198ms**
Using **ZFNet as RPN and detection network: 17fps / 59ms**
which is much faster than SS.

5.5. Some Examples



VOC 2007

. . .

References

1. [2015 NIPS] [Faster R-CNN]
   Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks

2. [2017 TPAMI] [Faster R-CNN]
   Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks

3. [2015 ICCV] [Fast R-CNN]
   Fast R-CNN

4. [2014 CVPR] [R-CNN]
   Rich feature hierarchies for accurate object detection and semantic segmentation

5. [2013 IJCV] [Selective Search]
   Selective Search for Object Recognition

My Reviews

1. Review: Fast R-CNN (Object Detection)

2. Review: R-CNN (Object Detection)