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

Abstract

Region Proposal Network
full-image convolutional features with the detection network
nearly cost-free region proposals

merge RPN and Fast R-CNN into a single network

Introduction

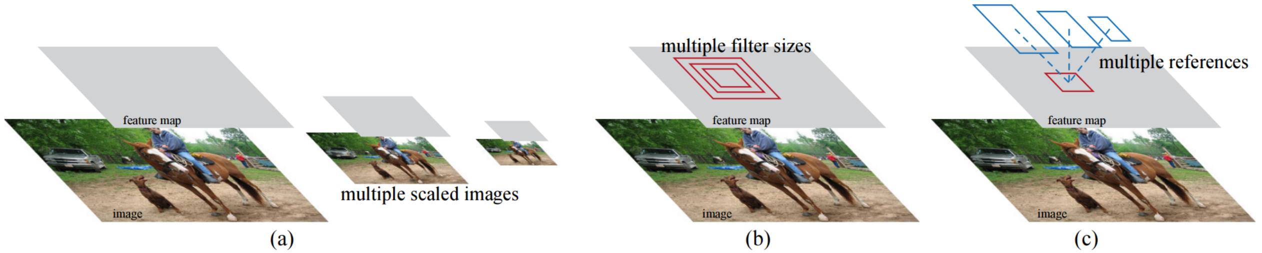
object detection
region proposal methods
region-based convolutional neural networks
cost has been drastically reduced thanks to sharing convolutions across proposals

Fast R-CNN [2], achieves near real-time rates using very deep networks

Region proposal methods
Selective Search
EdgeBoxes

implement it for the GPU
ignores the down-stream detection network -> misses important opportunities for sharing computation.
marginal cost for computing proposals is small

convolutional feature maps used by region-based detectors, like Fast RCNN, can also be used for generating region proposals



(a) (b) (c)

Our scheme can be thought of as a pyramid of regression references (Figure 1, c), which avoids enumerating images or filters of multiple scales or aspect ratios.

not only a cost-efficient solution
but also an effective way of improving object detection accuracy

Related Work

Object Proposals
Comprehensive surveys and comparisons of object proposal methods
based on
grouping super-pixels / sliding windows
Object proposal methods were adopted as external modules independent of the detectors

Deep Networks for Object Detection
R-CNN -> train CNN end-to-end to classify the proposal regions into object categories or background
R-CNN mainly plays as a classifier, and it does not predict object bounds

Its accuracy depends on the performance of the region proposal module

OverFeat
a fully-connected layer is trained to predict the box coordinates for the localization task that assumes a single object
convolutional features from an image pyramid

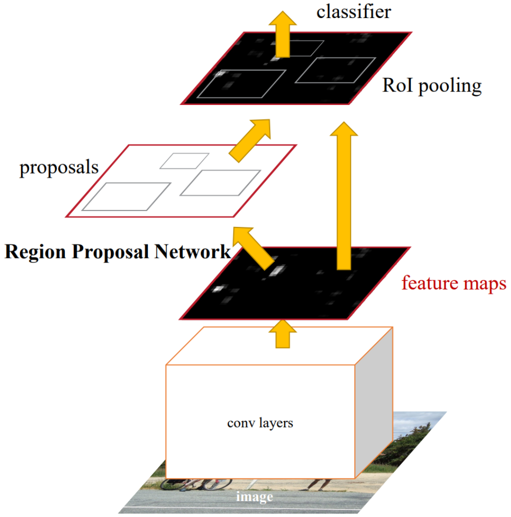
MultiBox
generate region proposals from a network
last fully-connected layer simultaneously predicts multiple class-agnostic boxes

Adaptively-sized pooling
efficient region-based object detection
semantic segmentation

Fast R-CNN
training on shared convolutional features -> shows compelling accuracy and speed.

Faster R-CNN

two modules
1 a deep fully convolutional network that proposes regions
2 Fast R-CNN detector



'attention' mechanisms

Region Proposal Networks

outputs a set of rectangular object proposals, each with an objectness score

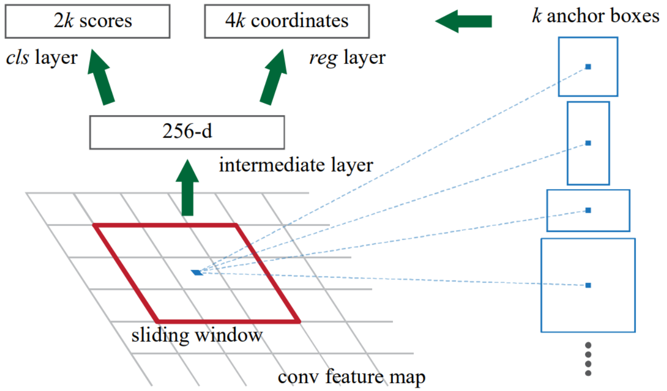
assume that both nets share a common set of convolutional layers slide a small network over the convolutional feature map output by the last shared convolutional layer

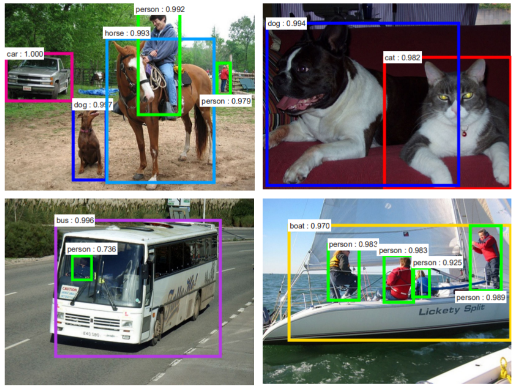
sliding window is mapped to a lower-dimensional feature

two sibling fully-connected layers—a box-regression layer (reg) and a box-classification layer

because the mini-network operates in a sliding-window fashion, the fully-connected layers are shared across all spatial locations

architecture is naturally implemented with an n×n convolutional layer followed by two sibling 1 × 1 convolutional layers (for reg and cls, respectively)





Anchors

4k -> 좌표들

2k -> probability of object or not object for each proposal

two-class

An anchor is centered at the sliding window in question, and is associated with a scale and aspect ratio

3 scales and 3 aspect ratios

Translation-Invariant Anchors

reduces the model size

less risk of overfitting on small datasets

Multi-Scale Anchors as Regression References

1 The images are resized at multiple scales, and feature maps (HOG [8] or deep convolutional features [9], [1], [2]) are computed for each scale

-> This way is often useful but is time-consuming

2 sliding windows of multiple scales (and/or aspect ratios) on the feature maps

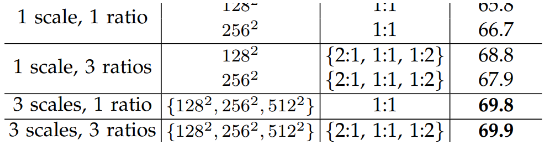
adopted jointly with the first way

Our method classifies and regresses bounding boxes with reference to anchor boxes of multiple scales and aspect ratios

relies on images and feature maps of a single scale, and uses filters (sliding windows on the feature map) of a single size

design of multiscale anchors is a key component for sharing features without extra cost for addressing scales

settings	anchor scales	aspect ratios	mAP (%)
	128^2	1:1	65.8



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**

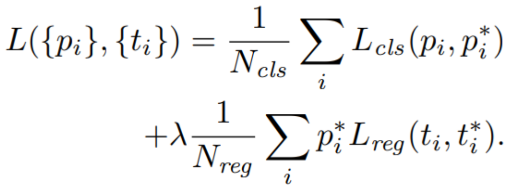
Loss Function

binary class label (of being an object or not) to each anchor

the anchor/anchors with the highest Intersection-overUnion (IoU) overlap with a ground-truth box

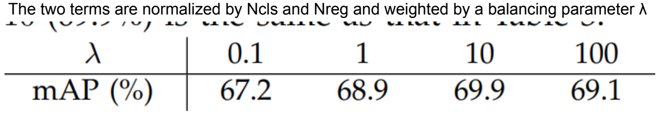
an anchor that has an IoU overlap higher than 0.7 with any ground-truth box

still adopt the first condition for the reason that in some rare cases the second condition may find no positive sample



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

The two terms are normalized by Ncls and Nreg and weighted by a balancing parameter λ



10 (69.9%) is the same as that in Table 3.

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

results are insensitive to the values of λ in a wide range

bounding-box regression is performed on features pooled from arbitrarily sized RoIs

the regression weights are shared by all region sizes

the features used for regression are of the same spatial size (3 × 3) on the feature maps

Each regressor is responsible for one scale and one aspect ratio, and the k regressors do not share weights.

Training RPNs

backpropagation and stochastic gradient descent (SGD)

image-centric sampling strategy

possible to optimize for the loss functions of all anchors, but this will bias towards negative samples as they are dominate

-> randomly sample 256 anchors in an image to compute the loss function of a mini-batch, where the sampled positive and negative anchors have a ratio of up to 1:1

pad the mini-batch with negative ones.

randomly initialize all new layers by drawing weights from a zero-mean Gaussian distribution with standard deviation 0.01

All other layers (i.e., the shared convolutional layers) are initialized by pretraining a model for ImageNet classification [36], as is standard practice

Sharing Features for RPN and Fast R-CNN

For the detection network, we adopt Fast R-CNN

RPN and Fast R-CNN with shared convolutional layers

three ways
Alternating training
Approximate joint training
Non-approximate joint training

Alternating training
train RPN, and use the proposals to train Fast R-CNN
tuned by Fast R-CNN is then used to initialize RPN, and this process is iterated.

Approximate joint training
RPN and Fast R-CNN merged into one network during training

forward pass generates region proposals
backward propagation takes place as usual
RPN loss and the Fast R-CNN loss are combined

ignores the derivative
boxes' coordinates are so approximate

Non-approximate joint training
RoI pooling layer in Fast R-CNN accepts the convolutional features and also the predicted bounding boxes as input
gradients w.r.t. the box coordinates are ignored in the above approximate joint training
RoI warping" layer as developed in [15], which is beyond the scope of this paper

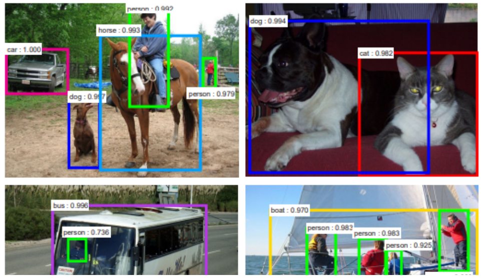
4-Step Alternating Training
1 train the RPN
2 train a separate detection network by Fast R-CNN using the proposals generated by the step-1 RPN
at this point two networks do not share convolutional layers
3 use the detector network to initialize RPN training but fix the shared convolutional layers and only fine-tune the layers unique to RPN
4 keeping the shared convolutional layers fixed, we fine-tune the unique layers of Fast R-CNN

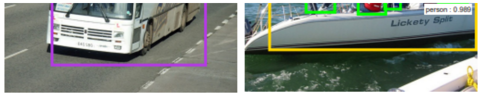
As such, both networks share the same convolutional layers and form a unified network

Implementation Details

train and test both region proposal and object detection networks on images of a single scale
Multi-scale feature extraction may improve accuracy but does not exhibit a good speed-accuracy trade-off

our solution does not need an image pyramid or filter pyramid to predict regions of multiple scales





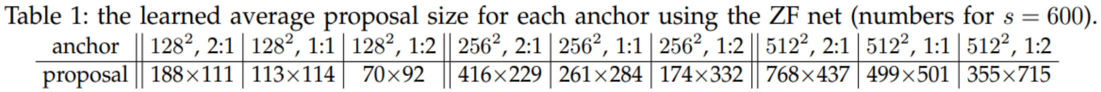


Table 1: the learned average proposal size for each anchor using the ZF net (numbers for $s = 600$).

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

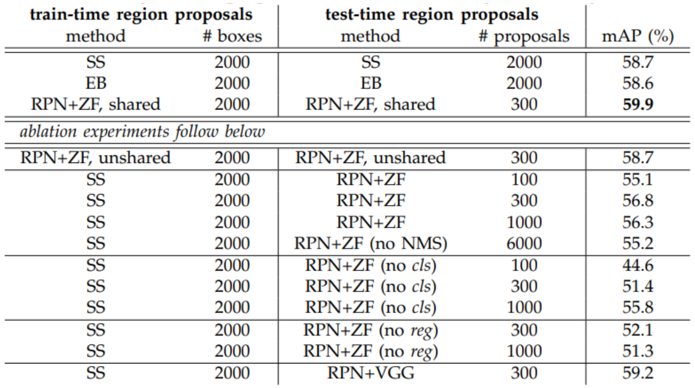
For a typical 1000 × 600 image, there will be roughly 20000 (≈ 60 × 40 × 9) anchors in total.
With the cross-boundary anchors ignored, there are about 6000 anchors per image for training.

If the boundary-crossing outliers are not ignored in training, they introduce large, difficult to correct error terms in the objective, and training does not converge

RPN proposals highly overlap
-> non-maximum suppression (NMS) on the proposal regions based on their cls scores.
fix the IoU threshold for NMS at 0.7, which leaves us about 2000 proposal regions per image
NMS does not harm the ultimate detection accuracy, but substantially reduces the number of proposals
After NMS, we use the top-N ranked proposal regions for detection
Fast R-CNN using 2000 RPN proposals, but evaluate different numbers of proposals at test-time

Experiments

Experiments on PASCAL VOC



train-time region proposals		test-time region proposals		
method	# boxes	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

Using RPN yields a much faster detection system

Ablation Experiments on RPN.
several ablation studies.

effect of sharing convolutional layers -> stop after the second step in the 4-step training process.
disentangle the RPN's influence on training the Fast R-CNN detection network
the RPN still leads to a competitive result

we separately investigate the roles of RPN's cls and reg outputs by turning off either of them at test-time
When the cls layer is removed at testtime

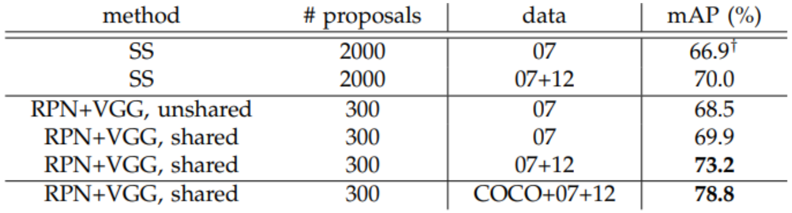
mAP is nearly unchanged with N = 1000 (55.8%), but degrades considerably to 44.6% when N = 100.

when the reg layer is removed at test-time
mAP drops to 52.1%

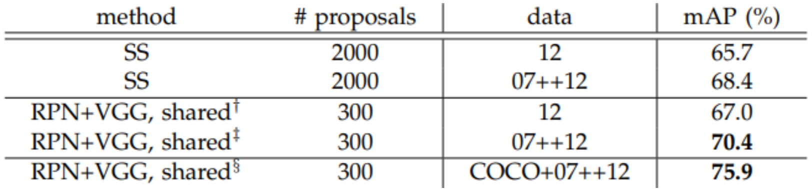
highquality proposals are mainly due to the regressed box bounds
The anchor boxes, though having multiple scales and aspect ratios, are not sufficient for accurate detection

We also evaluate the effects of more powerful networks on the proposal quality of RPN alone
mAP improves from 56.8% (using RPN+ZF) to 59.2% (using RPN+VGG)
proposal quality of RPN+VGG is better than that of RPN+ZF

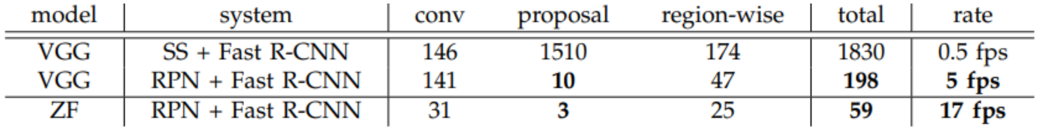
Performance of VGG-16



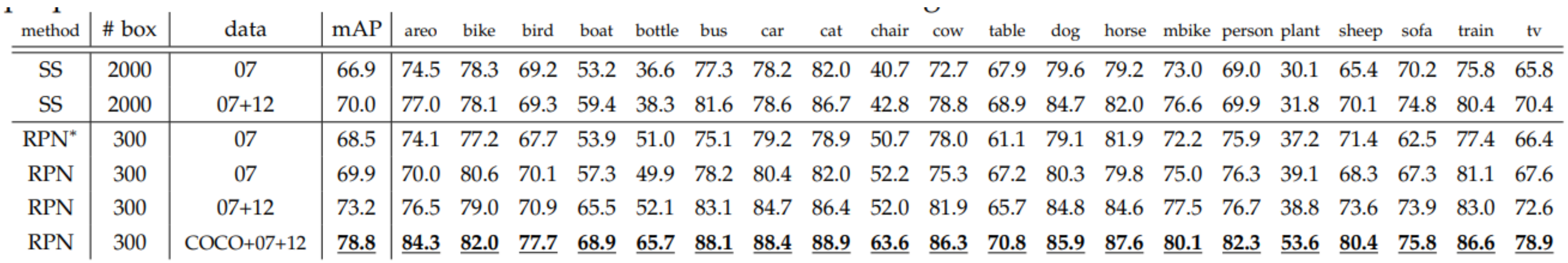
method	# proposals	data	mAP (%)
SS	2000	07	66.9†
SS	2000	07+12	70.0
RPN+VGG, unshared	300	07	68.5
RPN+VGG, shared	300	07	69.9
RPN+VGG, shared	300	07+12	**73.2**
RPN+VGG, shared	300	COCO+07+12	**78.8**



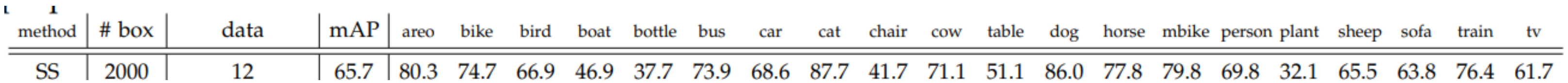
method	# proposals	data	mAP (%)
SS	2000	12	65.7
SS	2000	07++12	68.4
RPN+VGG, shared†	300	12	67.0
RPN+VGG, shared‡	300	07++12	**70.4**
RPN+VGG, shared§	300	COCO+07++12	**75.9**



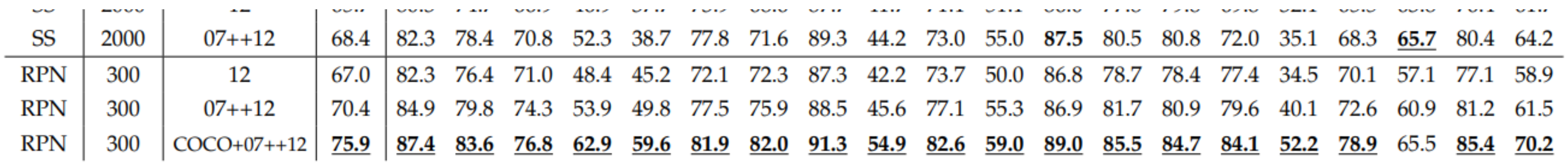
model	system	conv	proposal	region-wise	total	rate
VGG	SS + Fast R-CNN	146	1510	174	1830	0.5 fps
VGG	RPN + Fast R-CNN	141	**10**	47	**198**	**5 fps**
ZF	RPN + Fast R-CNN	31	**3**	25	**59**	**17 fps**



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**

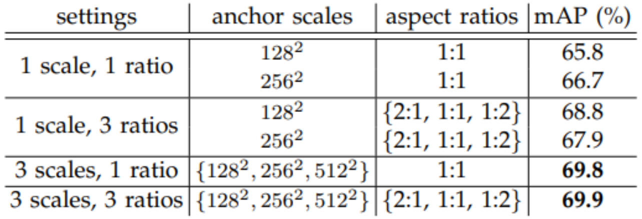


method	# box	data	mAP	areo	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
SS	2000	12	65.7	80.3	74.7	66.9	46.9	37.7	73.9	68.6	87.7	41.7	71.1	51.1	86.0	77.8	79.8	69.8	32.1	65.5	63.8	76.4	61.7



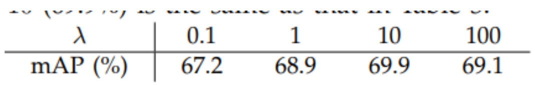
SS	2000	07++12	68.4	82.3	78.4	70.8	52.3	38.7	77.8	71.6	89.3	44.2	73.0	55.0	**87.5**	80.5	80.8	72.0	35.1	68.3	**65.7**	80.4	64.2
RPN	300	12	67.0	82.3	76.4	71.0	48.4	45.2	72.1	72.3	87.3	42.2	73.7	50.0	86.8	78.7	78.4	77.4	34.5	70.1	57.1	77.1	58.9
RPN	300	07++12	70.4	84.9	79.8	74.3	53.9	49.8	77.5	75.9	88.5	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**

Sensitivities to Hyper-parameters



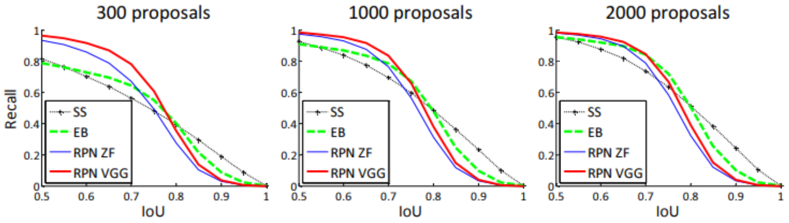
settings	anchor scales	aspect ratios	mAP (%)
1 scale, 1 ratio	128^2	1:1	65.8
	256^2	1:1	66.7
1 scale, 3 ratios	128^2	{2:1, 1:1, 1:2}	68.8
	256^2	{2:1, 1:1, 1:2}	67.9
3 scales, 1 ratio	$\{128^2, 256^2, 512^2\}$	1:1	**69.8**
3 scales, 3 ratios	$\{128^2, 256^2, 512^2\}$	{2:1, 1:1, 1:2}	**69.9**

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



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

Analysis of Recall-to-IoU



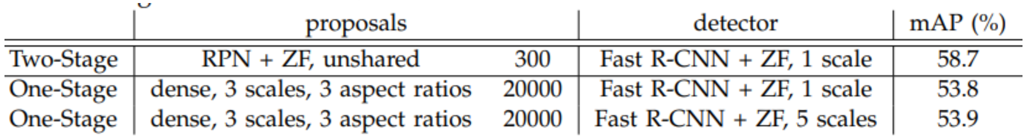
RPN has a good ultimate detection mAP when using as few as 300 proposals

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

OverFeat, the region-wise features come from a sliding window of one aspect ratio over a scale pyramid

In RPN, the features are from square (3×3) sliding windows and predict proposals relative to anchors with different scales and aspect ratios

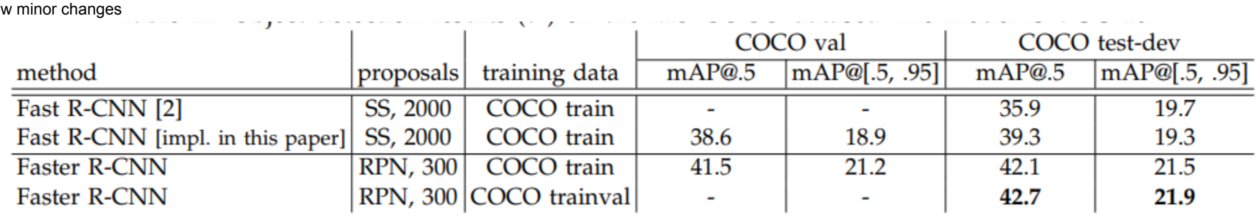


	proposals		detector	mAP (%)
Two-Stage	RPN + ZF, unshared	300	Fast R-CNN + ZF, 1 scale	58.7
One-Stage	dense, 3 scales, 3 aspect ratios	20000	Fast R-CNN + ZF, 1 scale	53.8
One-Stage	dense, 3 scales, 3 aspect ratios	20000	Fast R-CNN + ZF, 5 scales	53.9

This experiment justifies the effectiveness of cascaded region proposals and object detection
replacing SS region proposals with sliding windows leads to ~6% degradation in both papers
one-stage system is slower

Experiments on MS COCO

a few minor changes



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**

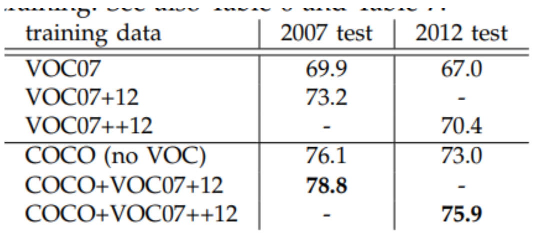
RPN performs excellent for improving the localization accuracy at higher IoU thresholds

Faster R-CNN in ILSVRC & COCO 2015 competitions

RPN completely learns to propose regions by neural networks

From MS COCO to PASCAL VOC

Large-scale data is of crucial importance for improving deep neural networks
MS COCO dataset can help with the detection performance on PASCAL VOC
COCO detection model on the PASCAL VOC dataset, without fine-tuning on any PASCAL VOC data
categories on COCO are a superset of those 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**

We note that the test-time speed of obtaining these strong results is still about 200ms per image

Conclusion

region proposal step is nearly cost-free
learned RPN also improves region proposal quality and thus the overall object detection accuracy