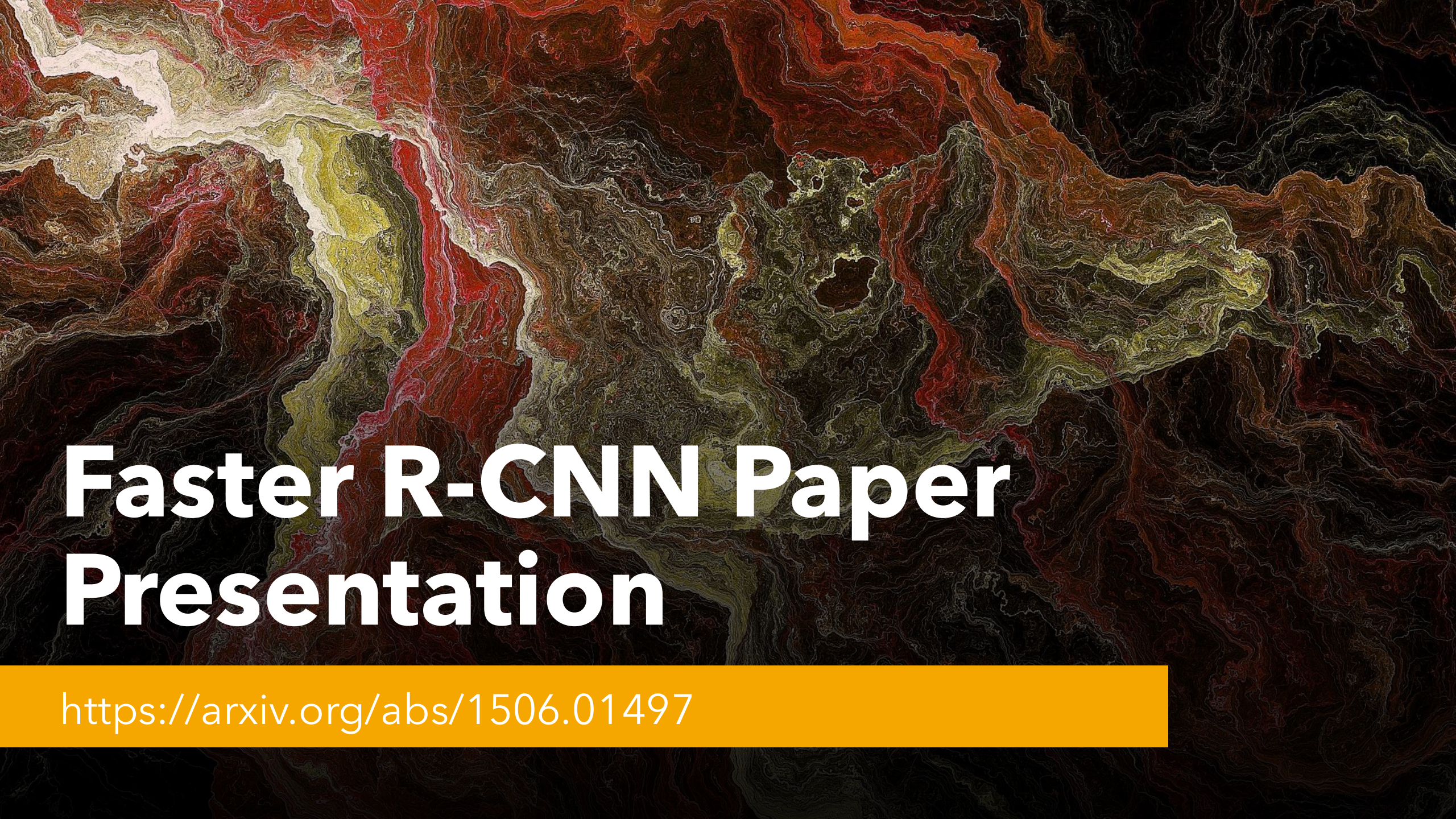

Faster R-CNN Paper
Presentation
https://arxiv.org/abs/1506.01497

Introduction

- Faster R-CNN builds upon the idea of Fast R-CNN detection network, but main difference is that it shares computation (i.e. convolution layers) with the region proposal and detection network as opposed to Fast R-CNN.
- The Region proposal network(RPNs) are constructed upon the shared conv layers by adding few more convnets and simultaneously preforms regression for bounding box and calculates the objectness score at each region i.e. whether the region has object or not.
- Unification b/w RPNs and Detection network is achieved by using alternate training method. (more later!!)

Anchor Boxes

- These serve as references for multiple scales and aspect ratios (pyramid of references)as opposed to previous methods which is uses pyramid of images or pyramid of filters.

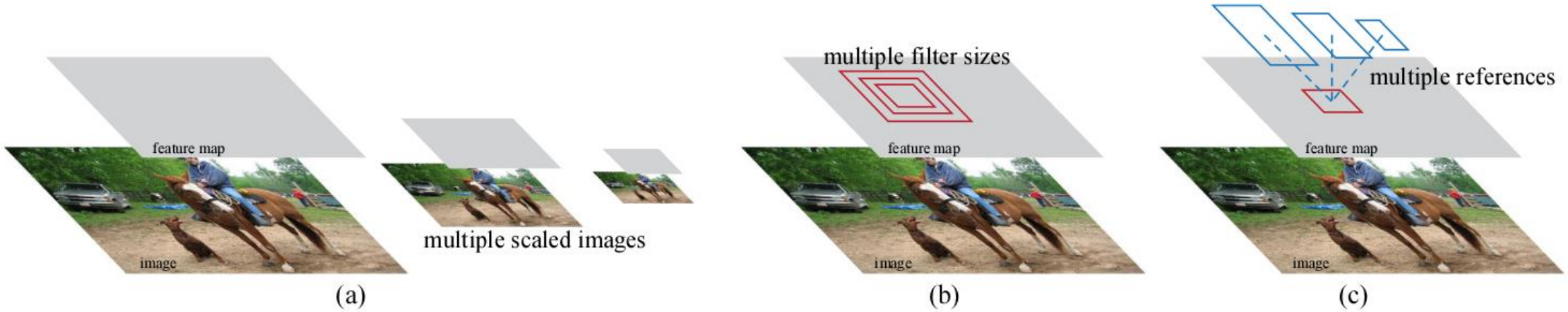


Figure 1: Different schemes for addressing multiple scales and sizes. (a) Pyramids of images and feature maps are built, and the classifier is run at all scales. (b) Pyramids of filters with multiple scales/sizes are run on the feature map. (c) We use pyramids of reference boxes in the regression functions.

Faster R-CNNs

- Feature maps are generated from the shared conv layers which are used by both the Detection network and RPN.
- For RPN a small n x n window is slided over the conv feature map of last o/p layer.
- Each sliding window is mapped to a lower dimensional feature and then fed into a classification and regression layer for bounding box and objectness score.

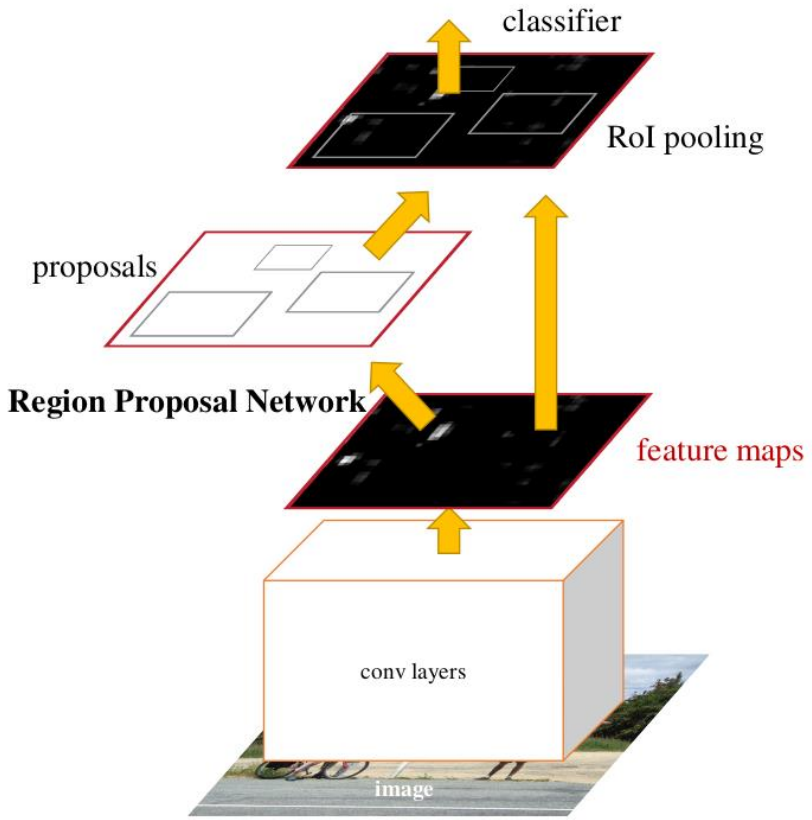


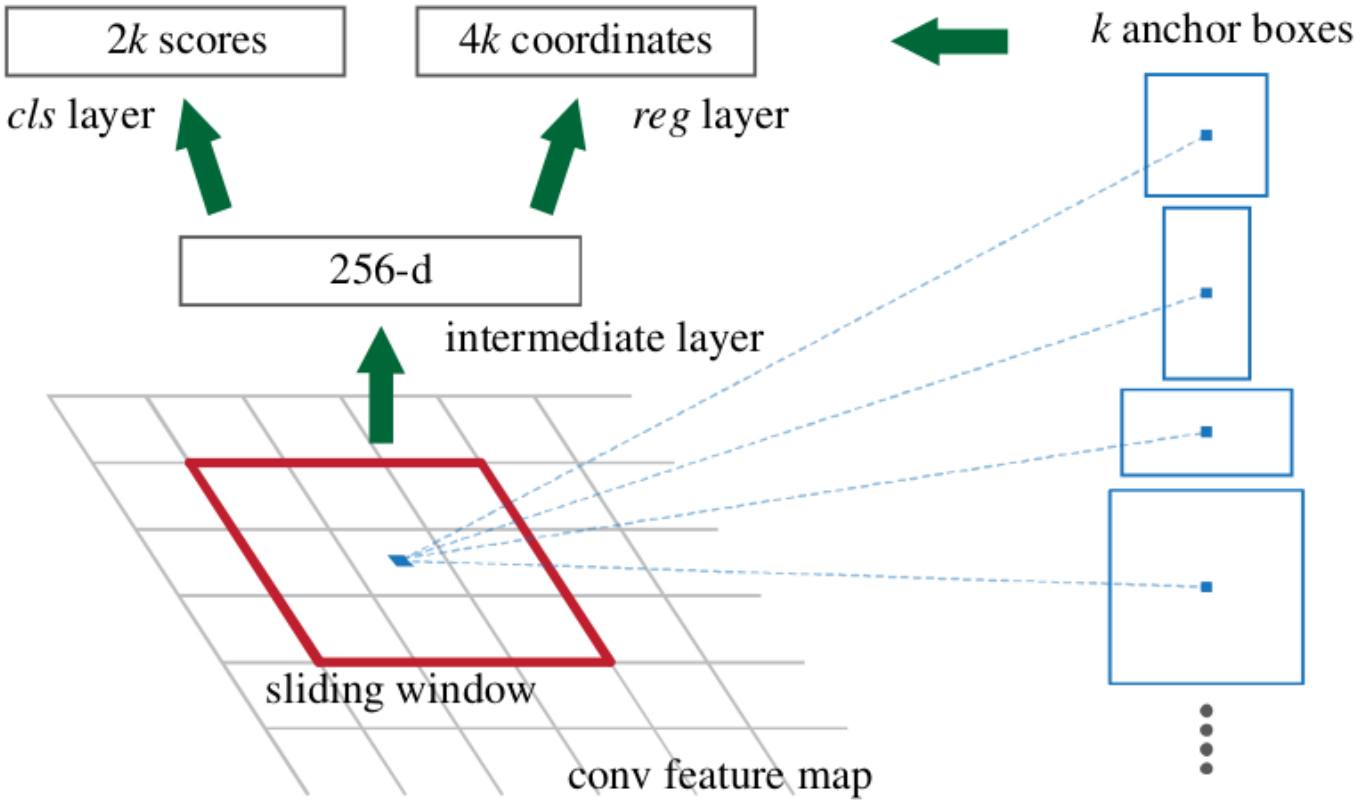
Figure 2: Faster R-CNN is a single, unified network for object detection. The RPN module serves as the 'attention' of this unified network.

Region Proposal Network

- At each location multiple region are proposed (k proposals).
- For each proposal reg layer generate 4 coordinate for bounding box.(4k o/p)
- Similarly for cls layer each proposal has 2 o/p whether it has object or not.

Note: Here k => no. of anchors, k=9 in paper.

- For a feature map of W X H there are total WHk anchors



Translation Invariant Property of Anchors

- Even if the objects is translated in the image the same proposal would also translate and can be calculated using the function as before.
- Hence, it can predict proposals in any location in the image.
- This property also results in less no of parameters and smaller model size w.r.t to other conventional means.

Anchors as Regression References

- The conventional means of doing multi-scale predictions is either through image pyramid or through filter pyramid which expensive due to time spend in rescaling and calculating feature map for each of the scale.
- Instead the pyramid of anchors is cost effective due to its single image size and filter size.
- This is the key feature for sharing the conv layers b/w Detections Network(Fast R-CNN here) and RPN.

Loss Function

- Anchors with highest IoU with ground truth and those with IoU more than 0.7 with ground truth box are assigned positive label (1) and vice versa for negative label (i.e. IoU < 0.3). (Note: Two conditions are imposed for robustness)
- Negative are ignored for training objective.
- The loss function is defined as:

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

- Here the cls loss is log loss and reg

loss is only active if there is positive anchor.

- Both losses are made equally weighted using lambda = 10 though results were insensitive to it.

Training of RPNs

- To avoid bias toward the negative anchors they randomly sample 256 anchors with a ratio of 1:1 for +ve and –ve anchors. Padding with –ve ones is used if +ve samples are < 128.
- The share conv layers are initialized with weights from ImageNet model(transfer learning) and the later layers are randomly initializes with 0-mean Gauss distribution with s.d. = 0.01

Training the unified network

- Alternate Training: First RPN is trained , then the learned proposal are used to train Fast R-CNN. The tuned net by Fast R-CNN is used to reinitialize RPN and the process gets iterated.
- Approximate joint training: FP generates region proposals which gets fixed to train Fast R-CNN. In BP in the shared layer loss from both nets are combined. Name Approximate is because it ignore the derivative of box's coordinates. Reduced training time as compared with above method.
- Non-Approximate joint training: Tries to incorporate the above ignored gradient.
- 4-step alternate training is used here:

1. Train RPN as described above.Intialized using Image-Net
2. Train Fast R-CNN using the proposal in 1. Initialized using Image-Net , no sharing of layers till now.
3. Use detector to reintialize the RPN training. Shared convs are fixed here.
4. Keeping shared convs fixed Fast R-CNN are tuned.

Faster RCNN implementation

3 scales

Multi-scale feature extraction (using an image pyramid) may improve accuracy but does not exhibit a good speed-accuracy trade-off

3 ratios 1:1,1:2,1:3

Cross boundary anchors are ignored

Metric - mAP (mean average precision)

NMS used to reduce redundancy

Two networks ZF net - 5 conv, 3 FC VGG-16 - 13 conv, 3 FC

RPN + ZF vs SS and EB

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**

RPN + ZF performs better with fewer proposals

It performs better due to sharing of convolutional computations

Ablation experiments

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

For unshared , we stop after second step in 4 step training process

For SS, loss in mAP due to inconsistency

Even when top 100 is used proposals are accurate enough

When 6000 are used there is not much difference showing effectiveness of NMS.

Ablation experiments

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

When cls is removed performance of first 100 decreases,cls scores account for accuracy of high ranked proposals

High quality proposals due to reg , anchor boxes not sufficient

Proposal quality of RPN + VGG better than RPN + ZF

Table 3: Detection results on **PASCAL VOC 2007 test set**. The detector is Fast R-CNN and VGG-16. Training data: "07": VOC 2007 trainval, "07+12": union set of VOC 2007 trainval and VOC 2012 trainval. For RPN, the train-time proposals for Fast R-CNN are 2000. †: this number was reported in [2]; using the repository provided by this paper, this result is higher (68.1).

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**

RPN + VGG more accurate than SS

SS predefined , RPN benefits from better networks

Computation 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**

Proposals faster for RPN

Conv slower for VGG due to more complexity

Region wise faster due to lower number of proposals

Scales and Aspect 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**

Anchors of multiple sizes effective

Scales and aspect ratios not disentangled

Table 9: Detection results of Faster R-CNN on PASCAL VOC 2007 test set using **different values of** λ in Equation (1). The network is VGG-16. The training data is VOC 2007 trainval. The default setting of using $\lambda = 10$ (69.9%) is the same as that in Table 3.

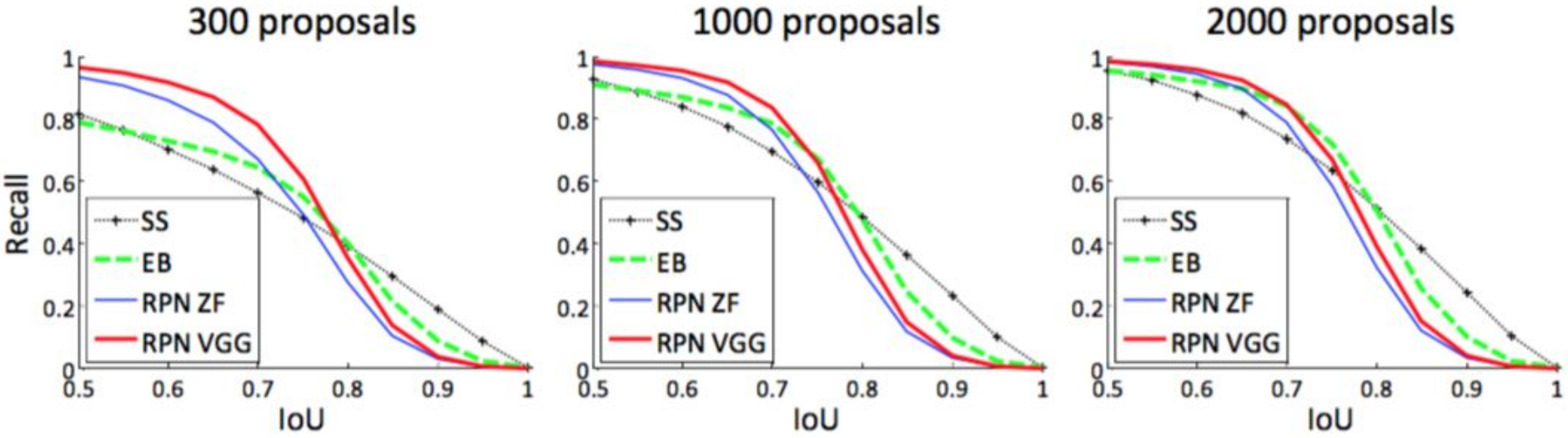
λ	0.1	1	10	100
mAP (%)	67.2	68.9	69.9	69.1

Result mostly insensitive to λ in wide range

λ = 10 most useful as it makes both terms equal

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

Recall vs IoU



RPN performs gracefully for reduction of number of proposals due to cls

Table 10: **One-Stage Detection *vs.* Two-Stage Proposal + Detection**. Detection results are on the PASCAL VOC 2007 test set using the ZF model and Fast R-CNN. RPN uses unshared features.

	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

Overfeat : one stage , class - specific

Faster RCNN : two stage cascade

This model performs better , running faster with lower number of proposals

Table 11: Object detection results (%) on the **MS COCO** dataset. The model is VGG-16.

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**

mAP@0.5 : threshold of 0.5 for iou

mAP@[.5, .95]) : mAP averaged over different thresholds starting from 0.5 upto 0.95

RPN performs very well for improving localization accuracy at high thresholds

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**

Extra data increases mAP

Conclusion

Region proposal step is nearly cost free !