



Training Region-based Object Detectors with Online Hard Example Mining

Abhinav Shrivastava¹

Abhinav Gupta¹

Ross Girshick²

¹Carnegie Mellon University

²Facebook AI Research

`{ashrivas, abhinavg}@cs.cmu.edu`

`rbg@fb.com`

Related Papers

Rich feature hierarchies for accurate object detection and semantic segmentation(R-CNN)
CVPR-2014 **8929** citations

Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition
TPAMI-2015 **2870** citations

Fast R-CNN ICCV-2015 **5880** citations

Faster R-CNN Towards Real-Time Object Detection with Region Proposal Networks
NIPS-2015 **10443** citations

Training Region-based Object Detectors with Online Hard Example Mining
CVPR-2016 **556** citations

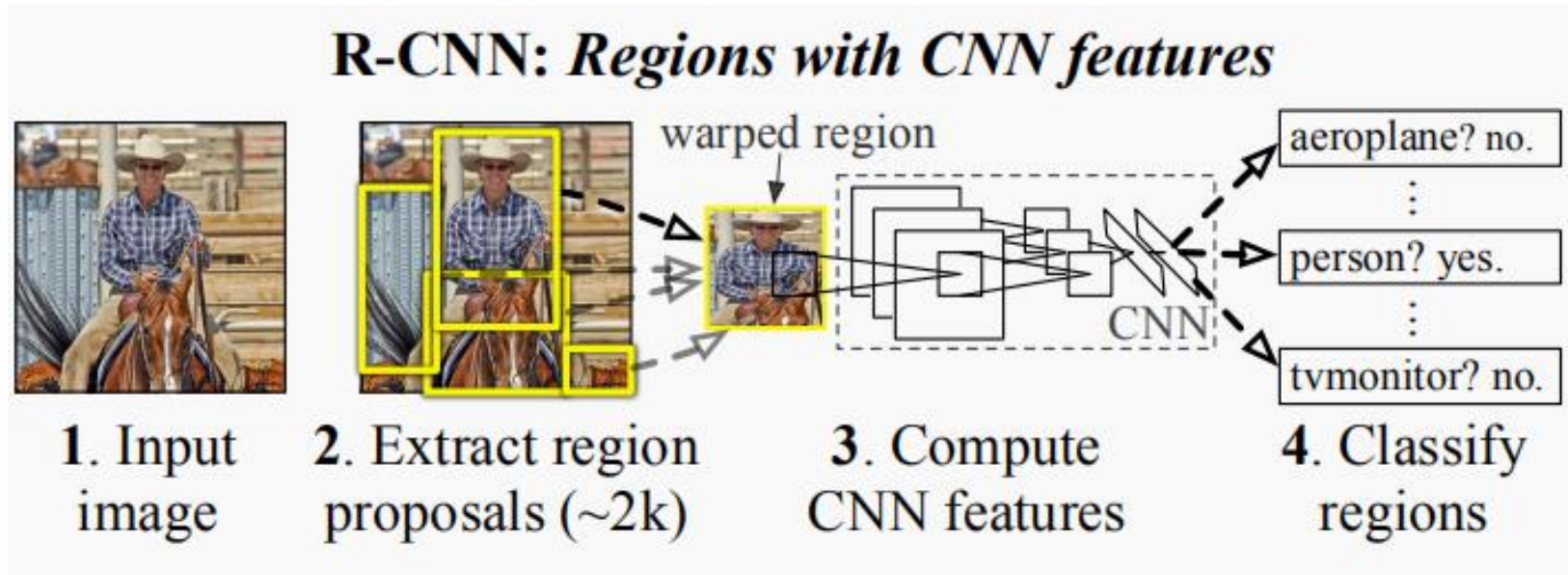
■ 相关学者

Kaiming He (何凯明) Jian Sun (孙剑) Shaoqing Ren (任少庆)

Ross Girshick (rbg) --Facebook AI Research (FAIR)

R-CNN

- One can apply high-capacity convolutional neural networks (CNNs) to bottom-up region proposals in order to localize and segment objects
- When labeled training data is scarce, supervised pre-training for an auxiliary task, followed by domain-specific fine-tuning, yields a significant performance boost



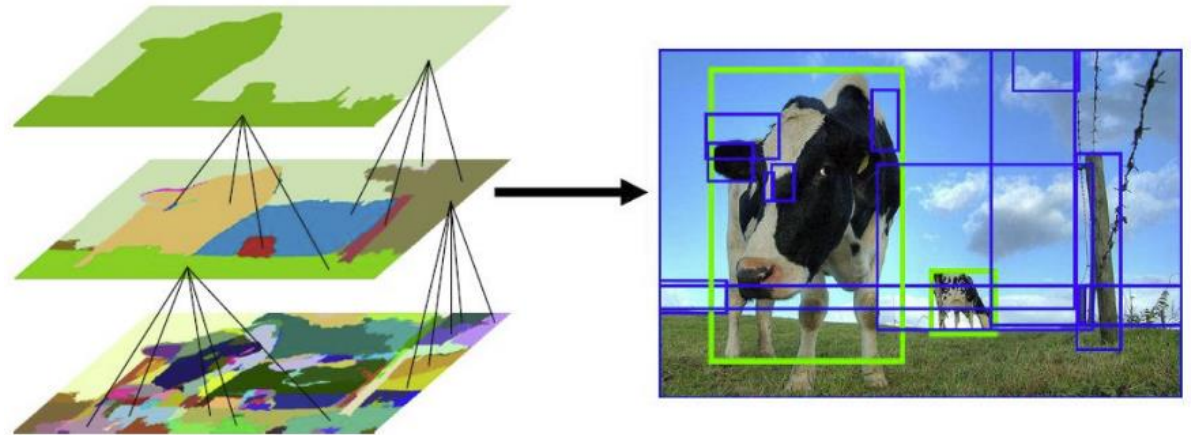
Region proposal

■ Sliding window

This process will be extremely slow if we use deep learning CNN for image classification at each location.

■ Selective search

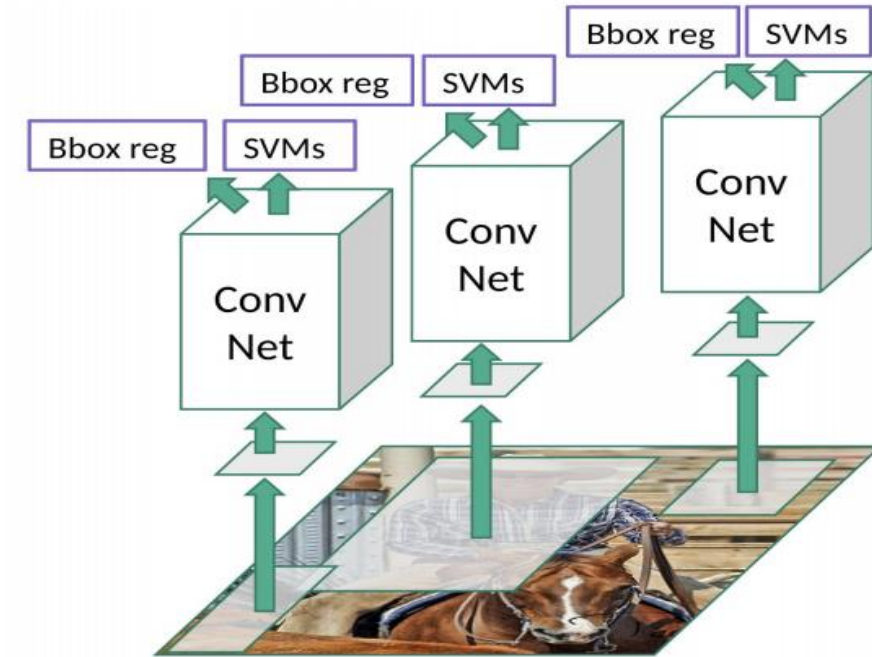
Selective Search for Object Recognition IJCV-2013



R-CNN

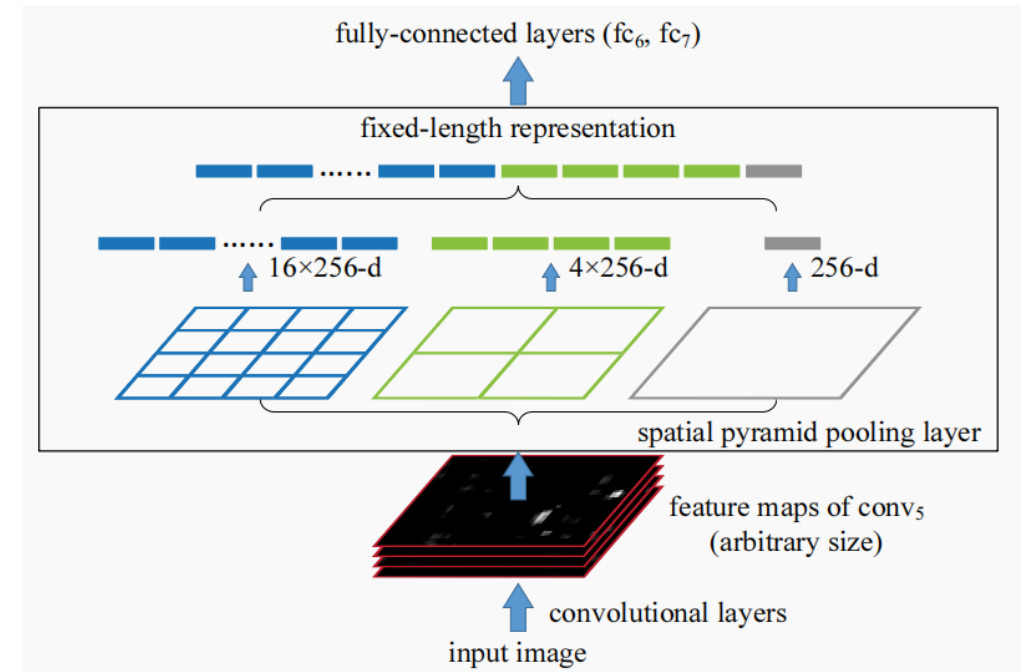
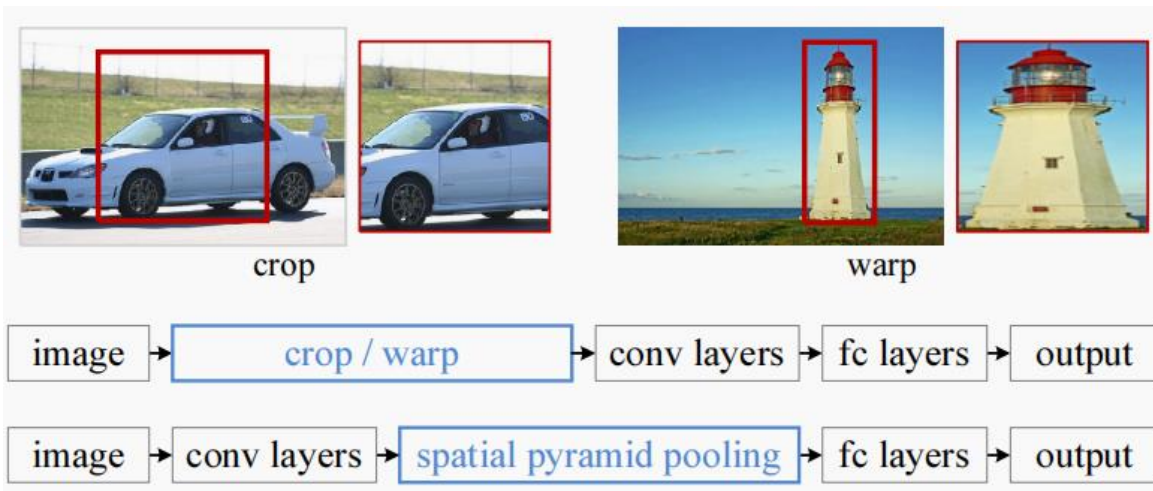
■ Problems with R-CNN:

1. It still takes a huge amount of time to train the network as you would have to classify 2000 region proposals per image. (**Expensive in Space and Time**)
2. It cannot be implemented real time as it takes around 47 seconds for each test image. (**Slow Object Detection**)
3. The selective search algorithm is a fixed algorithm. Therefore, no learning is happening at that stage. This could lead to the generation of bad candidate region proposals. (**Fixed the region proposals**)



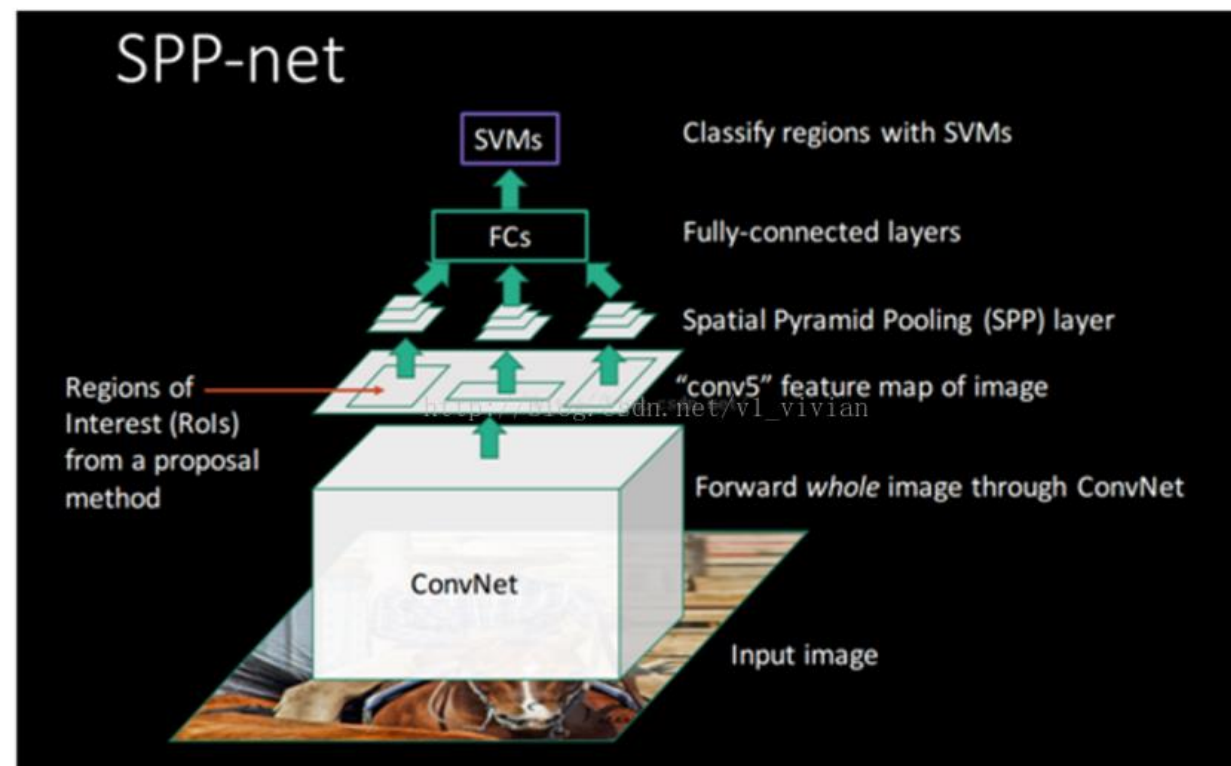
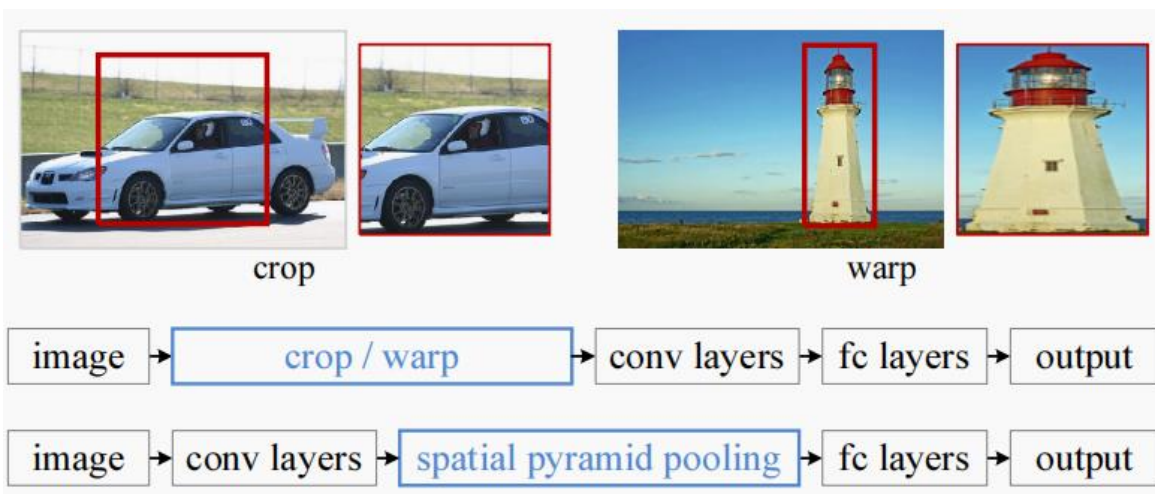
SPP-Net(Spatial Pyramid Pooling)

- Remove the fixed-size constraint of the network
- SPP uses multi-level spatial bins, Multi-level pooling has been shown to be robust to object deformations
- SPP can pool features extracted at variable scales thanks to the flexibility of input scales.



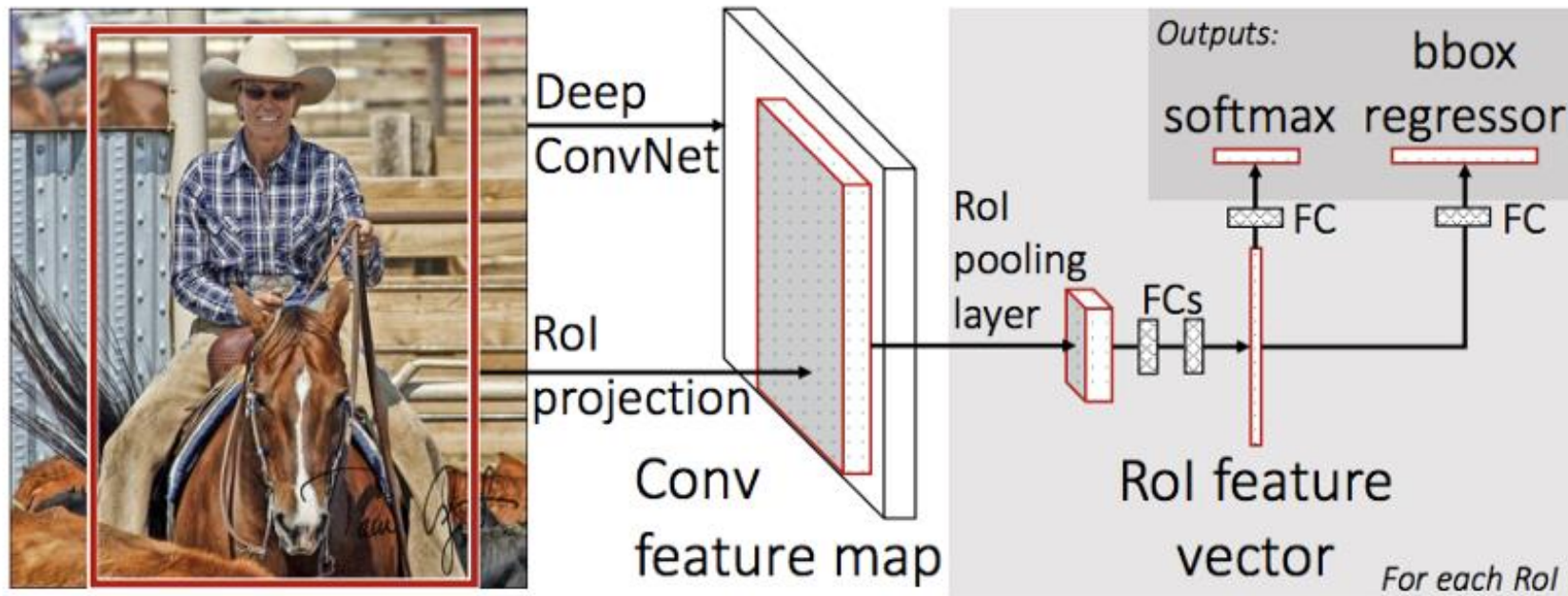
SPP-Net(Spatial Pyramid Pooling)

- Remove the fixed-size constraint of the network
- SPP uses multi-level spatial bins, Multi-level pooling has been shown to be robust to object deformations
- SPP can pool features extracted at variable scales thanks to the flexibility of input scales.



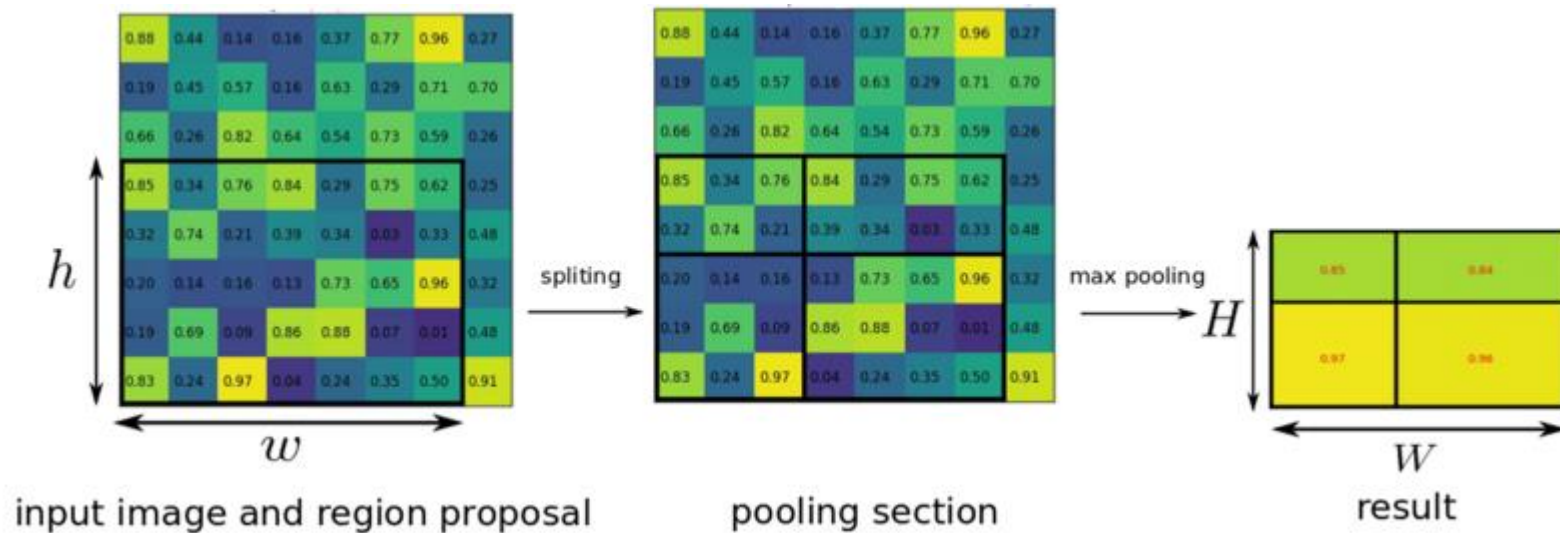
Fast R-CNN

- Instead of feeding the region proposals to the CNN, we feed the input image to the CNN to generate a convolutional feature map.
- Identify the region of proposals and warp them into squares and by using a RoI pooling layer we reshape them into a fixed size



Fast R-CNN

- The RoI pooling layer uses max pooling to convert the features inside any valid region of interest into a small feature map with a fixed spatial extent of $H \times W$, where H and W are layer hyper-parameters that are independent of any particular RoI.



Fast R-CNN

■ Multi-task loss

Since Fast R-CNN is an end-to-end learning architecture to learn the class of object as well as the associated bounding box position and size, the loss is multi-task loss.

$$L(p, u, t^u, v) = L_{\text{cls}}(p, u) + \lambda[u \geq 1]L_{\text{loc}}(t^u, v),$$

L_{cls} is the log loss for true class u .

L_{loc} is the loss for bounding box.

$$L_{\text{loc}}(t^u, v) = \sum_{i \in \{x, y, w, h\}} \text{smooth}_{L_1}(t_i^u - v_i),$$

in which

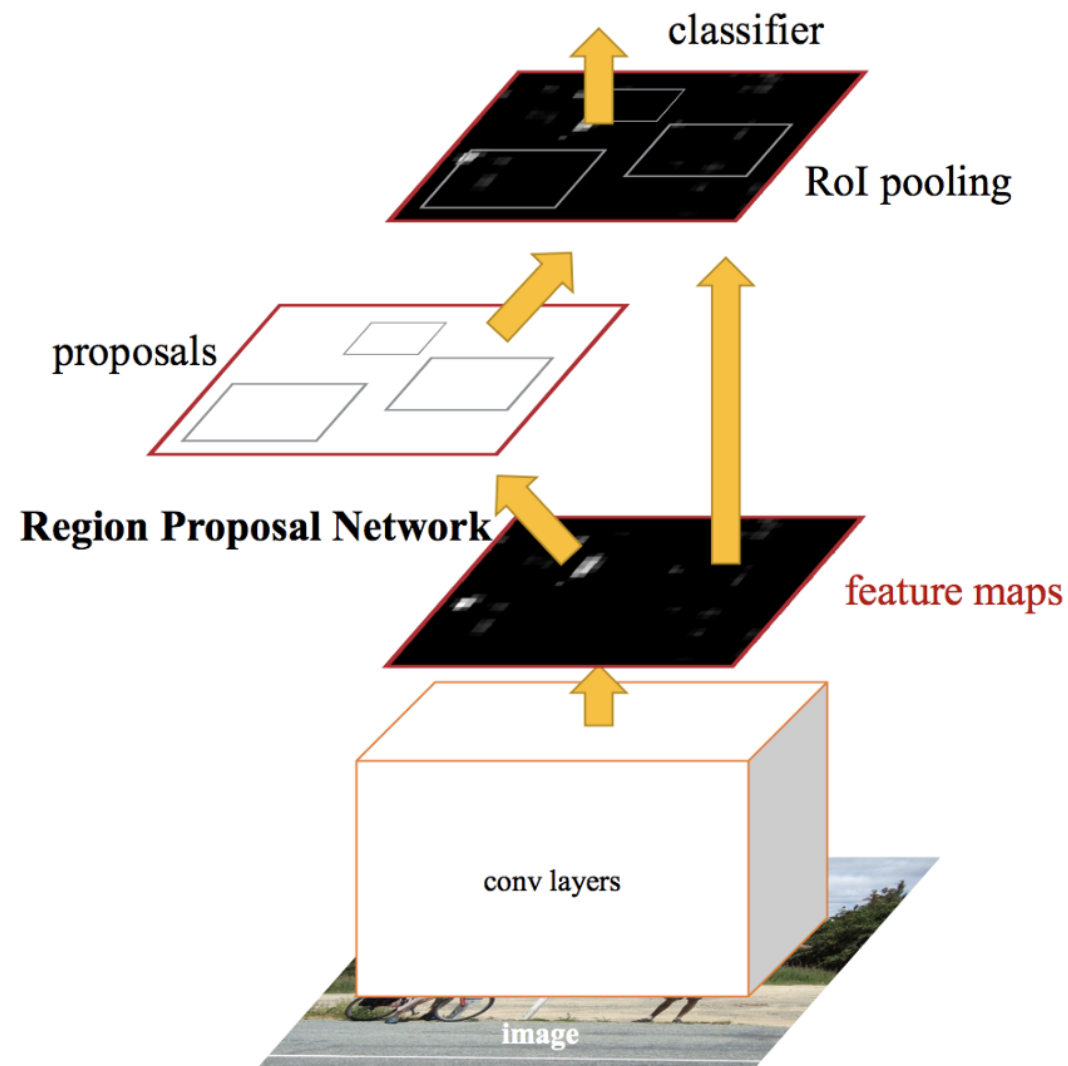
$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise,} \end{cases}$$

Faster R-CNN

■ Region Proposal Network

In R-CNN and Fast R-CNN , the region proposal approach/network and the detection network are decoupled.

In Faster R-CNN , RPN using SS is replaced by RPN using CNN.





Training Region-based Object Detectors with Online Hard Example Mining

Abhinav Shrivastava¹

Abhinav Gupta¹

Ross Girshick²

¹Carnegie Mellon University

²Facebook AI Research

`{ashrivas, abhinavg}@cs.cmu.edu`

`rbg@fb.com`


Hard example mining (Bootstrapping)

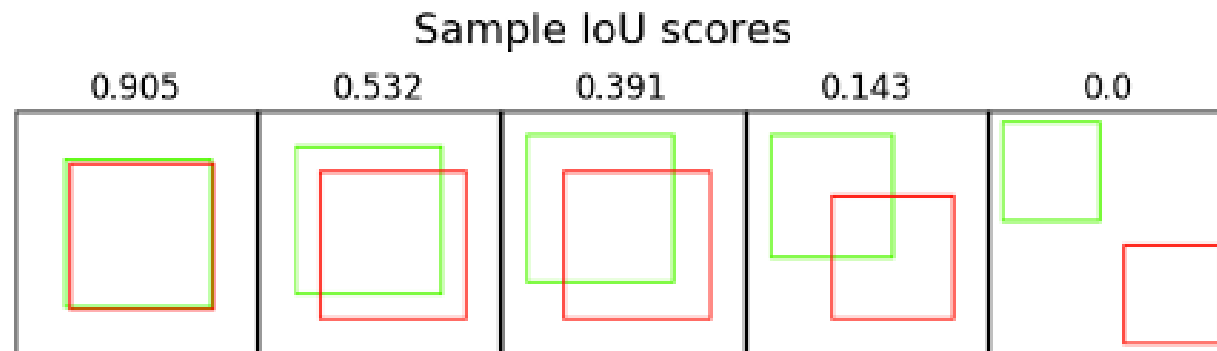
- The key idea was to gradually grow, or bootstrap, the set of background examples by selecting those examples for which the detector triggers a false alarm
- An iterative training algorithm
 1. updating the detection model given the current set of examples
 2. then using the updated model to find new false positives to add to the bootstrapped training set
- The **problem** for CNN model

Freezing the model for even a few iterations at a time would dramatically slow progress.

IoU(Intersection over Union)

- Intersection over Union (IoU) is a metric that allows us to evaluate how similar our predicted bounding box is to the ground truth bounding

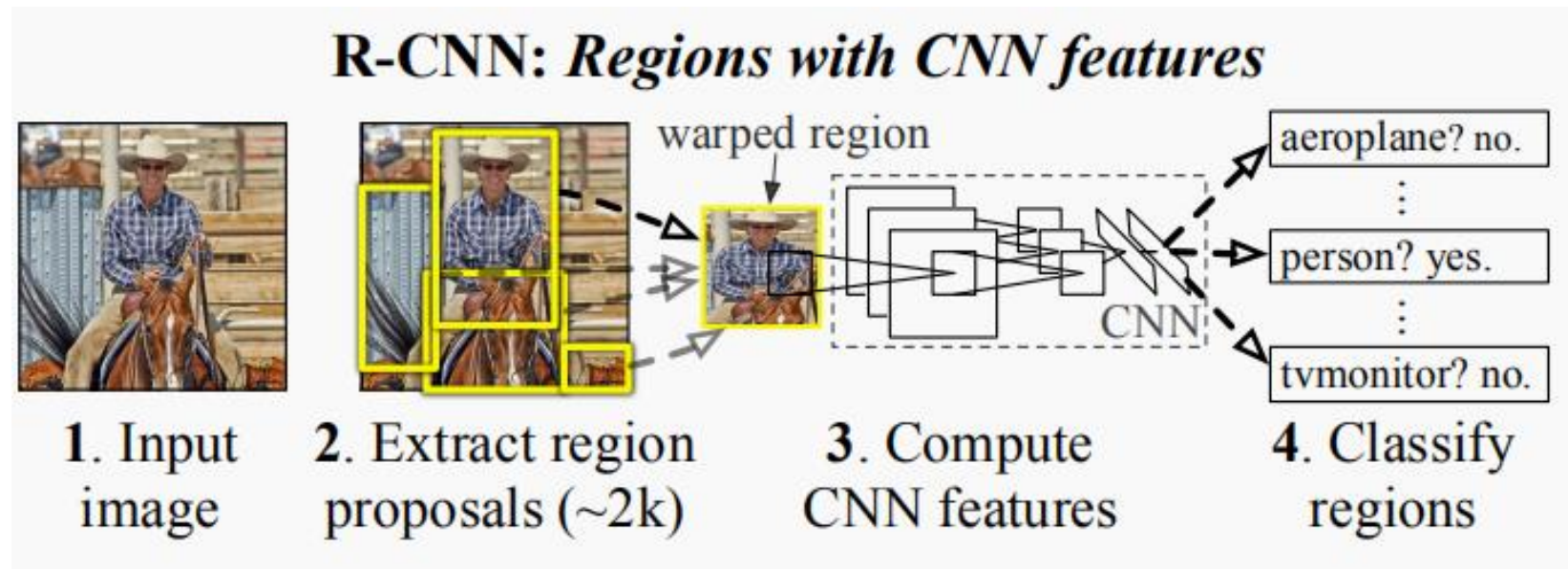
$$\text{IoU} = \frac{\text{Overlapping Region}}{\text{Combined Region}}$$




R-CNN

- Its intersection over union (IoU) overlap with a ground-truth bounding box should be at least 0.5 (**Positive Samples**)
- A region is labeled background if its maximum IoU with ground truth is in the interval $[bg_{IoU}, 0.5)$ (**Negative samples**)
- **Balancing fg-bg Rols**

Rebalance the foreground-to-background ratio in each mini-batch to a target of 1 : 3



OHEM (online hard example mining)

■ Intuition

It has always been detection datasets contain an overwhelming number of easy examples and a small number of hard examples. Automatic selection of these hard examples can make training more effective and efficient.

The key is that although each SGD iteration samples only a small number of images, each image contains thousands of example Rols from which we can select the hard examples rather than a heuristically sampled subset.

Method

■ Specific Method

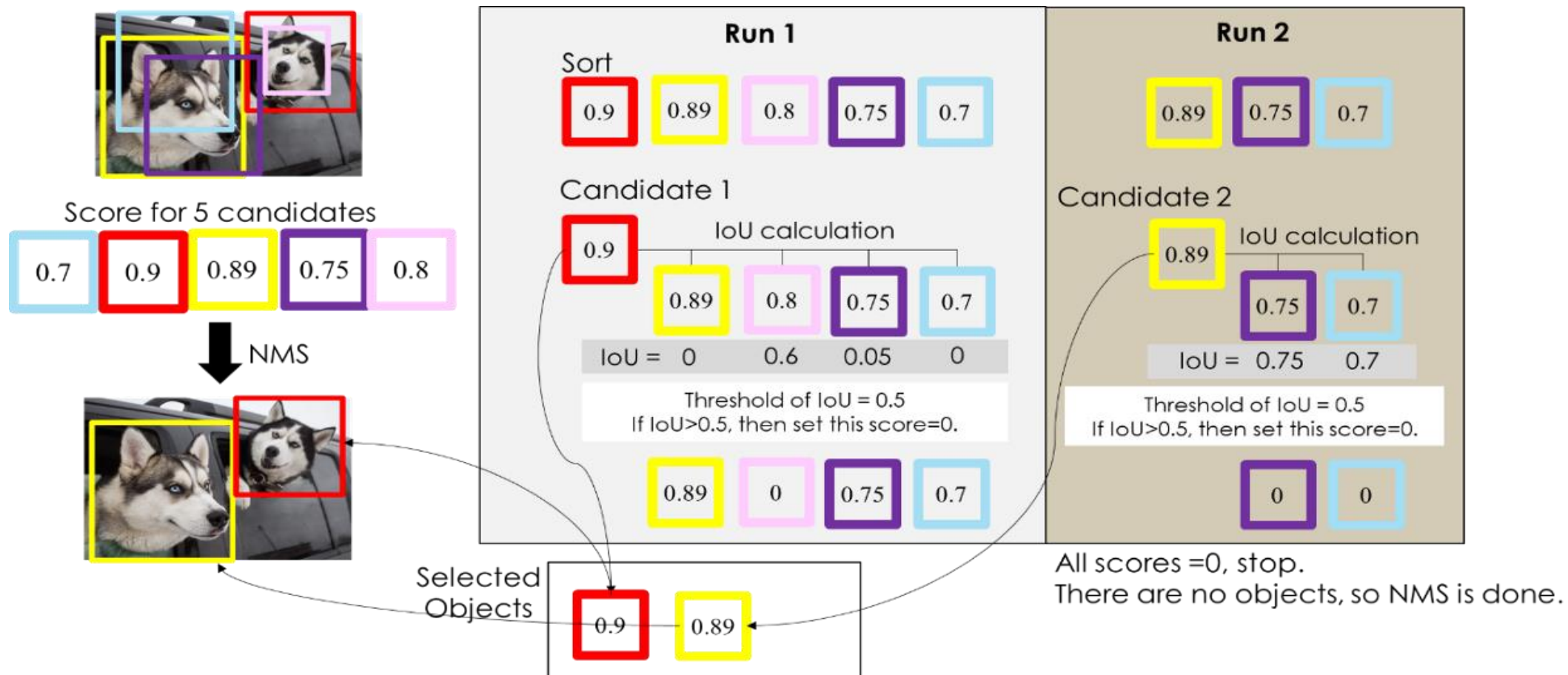
1. For an input image at SGD iteration t , we first compute a conv feature map using the conv network.
2. Then the RoI network uses this feature map and the all the input Rols (R), instead of a sampled mini-batch , to do a forward pass.
3. Hard examples are selected by sorting the input Rols by loss and taking the B/N examples for which the current network performs worst

■ Co-located Rols with high overlap are likely to have correlated losses

Using standard non-maximum suppression (NMS) to perform deduplication

NMS (Non-Maximum Suppression)

- 对于Bounding Box的列表B及其对应的置信度S 选择具有最大score的检测框M,将其从B集合中移除并加入到最终的检测结果D中.通常将B中剩余检测框中与M的IoU大于阈值Nt的框从B中移除.重复这个过程,直到B为空.



Architecture of OHEM

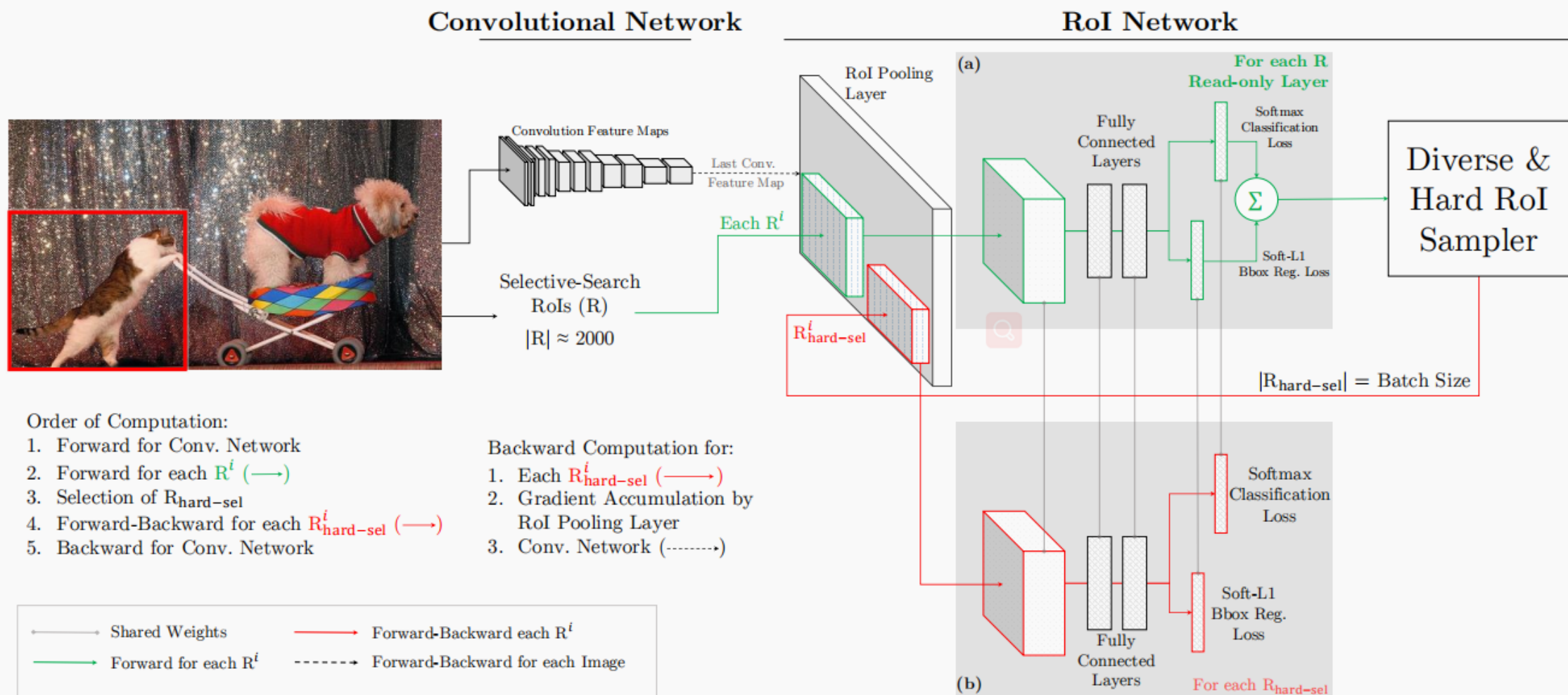


Figure 2: Architecture of the proposed training algorithm. Given an image, and selective search RoIs, the conv network computes a conv feature map. In (a), the *readonly* RoI network runs a forward pass on the feature map and all RoIs (shown in green arrows). Then the Hard RoI module uses these RoI losses to select B examples. In (b), these hard examples are used by the RoI network to compute forward and backward passes (shown in red arrows).

100%

■ Dataset

VGG16 from

PASCAL VOC07 dataset

- OHEM vs. heuristic sampling(1-4)
- Robust gradient estimates(5-6)
- Use all Rols(7-10)

Experiment		Model	N	LR	B	bg_lo	07 mAP
1	Fast R-CNN [14]	VGGM	2	0.001	128	0.1	59.6
2		VGG16					67.2
3	Removing hard mining heuristic (Section 5.2)	VGGM	2	0.001	128	0	57.2
4		VGG16					67.5
5	Fewer images per batch (Section 5.3)	VGG16	1	0.001	128	0.1	66.3
6						0	66.3
7	Bigger batch, High LR (Section 5.4)	VGGM	1	0.004	2048	0	57.7
8			2				60.4
9		VGG16	1	0.003	2048	0	67.5
10			2				68.7
11	Our Approach	VGG16	1	0.001	128	0	69.7
12		VGGM	2	0.001	128	0	62.0
13		VGG16					69.9

Experiment

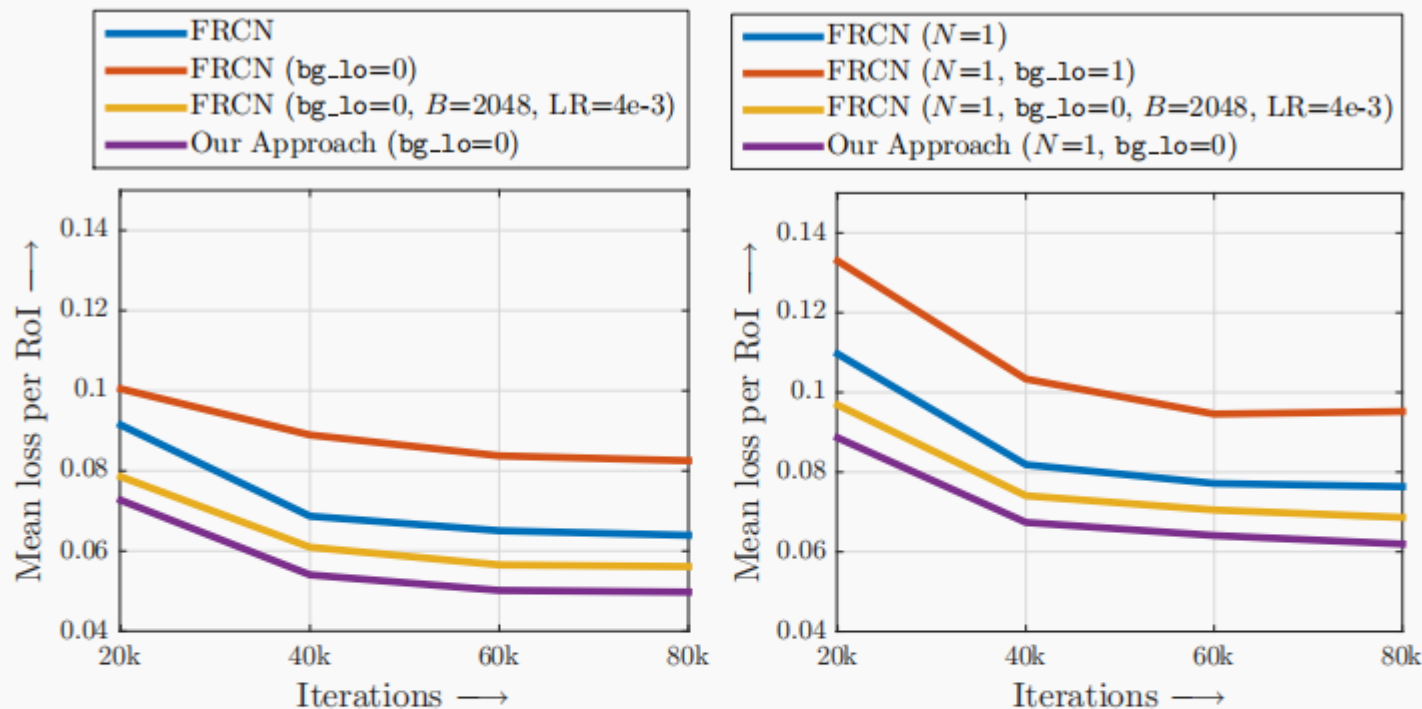


Figure 3: Training loss is computed for various training procedures using VGG16 networks discussed in Section 5. We report mean loss per RoI. These results indicate that using hard mining for training leads to lower training loss than any of the other heuristics.

Computational cost

Table 2: Computational statistics of training FRCN [14] and FRCN with OHEM (using an Nvidia Titan X GPU).

	VGGM		VGG16		
	FRCN	Ours	FRCN	FRCN*	Ours*
time (sec/iter)	0.13	0.22	0.60	0.57	1.00
max. memory (G)	2.6	3.6	11.2	6.4	8.7

*: uses gradient accumulation over two forward/backward passes

Experiment

Table 3: **VOC 2007 test** detection average precision (%). All methods use VGG16. Training set key: **07**: VOC07 trainval, **07+12**: union of **07** and VOC12 trainval. All methods use bounding-box regression. Legend: **M**: using multi-scale for training and testing, **B**: multi-stage bbox regression. FRCN* refers to FRCN [14] with our training schedule.

method	M	B	train set	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	persn	plant	sheep	sofa	train	tv
FRCN [14]			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
FRCN*			07	67.2	74.6	76.8	67.6	52.9	37.8	78.7	78.8	81.6	42.2	73.6	67.0	79.4	79.6	74.1	68.3	33.4	65.9	68.7	75.4	68.1
Ours			07	69.9	71.2	78.3	69.2	57.9	46.5	81.8	79.1	83.2	47.9	76.2	68.9	83.2	80.8	75.8	72.7	39.9	67.5	66.2	75.6	75.9
FRCN*	✓	✓	07	72.4	77.8	81.3	71.4	60.4	48.3	85.0	84.6	86.2	49.4	80.7	68.1	84.1	86.7	80.2	75.3	38.7	71.9	71.5	77.9	67.8
MR-CNN [13]	✓	✓	07	74.9	78.7	81.8	76.7	66.6	61.8	81.7	85.3	82.7	57.0	81.9	73.2	84.6	86.0	80.5	74.9	44.9	71.7	69.7	78.7	79.9
Ours	✓	✓	07	75.1	77.7	81.9	76.0	64.9	55.8	86.3	86.0	86.8	53.2	82.9	70.3	85.0	86.3	78.7	78.0	46.8	76.1	72.7	80.9	75.5
FRCN [14]			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
Ours			07+12	74.6	77.7	81.2	74.1	64.2	50.2	86.2	83.8	88.1	55.2	80.9	73.8	85.1	82.6	77.8	74.9	43.7	76.1	74.2	82.3	79.6
MR-CNN [13]	✓	✓	07+12	78.2	80.3	84.1	78.5	70.8	68.5	88.0	85.9	87.8	60.3	85.2	73.7	87.2	86.5	85.0	76.4	48.5	76.3	75.5	85.0	81.0
Ours	✓	✓	07+12	78.9	80.6	85.7	79.8	69.9	60.8	88.3	87.9	89.6	59.7	85.1	76.5	87.1	87.3	82.4	78.8	53.7	80.5	78.7	84.5	80.7

Table 4: **VOC 2012 test** detection average precision (%). All methods use VGG16. Training set key: **12**: VOC12 trainval, **07++12**: union of VOC07 trainval, VOC07 test, and VOC12 trainval. Legend: **M**: using multi-scale for training and testing, **B**: iterative bbox regression.

method	M	B	train set	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	persn	plant	sheep	sofa	train	tv
FRCN [14]			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
Ours ¹			12	69.8	81.5	78.9	69.6	52.3	46.5	77.4	72.1	88.2	48.8	73.8	58.3	86.9	79.7	81.4	75.0	43.0	69.5	64.8	78.5	68.9
MR-CNN [13]	✓	✓	12	70.7	85.0	79.6	71.5	55.3	57.7	76.0	73.9	84.6	50.5	74.3	61.7	85.5	79.9	81.7	76.4	41.0	69.0	61.2	77.7	72.1
Ours ²	✓	✓	12	72.9	85.8	82.3	74.1	55.8	55.1	79.5	77.7	90.4	52.1	75.5	58.4	88.6	82.4	83.1	78.3	47.0	77.2	65.1	79.3	70.4
FRCN [14]			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
Ours ³			07++12	71.9	83.0	81.3	72.5	55.6	49.0	78.9	74.7	89.5	52.3	75.0	61.0	87.9	80.9	82.4	76.3	47.1	72.5	67.3	80.6	71.2
MR-CNN [13]	✓	✓	07++12	73.9	85.5	82.9	76.6	57.8	62.7	79.4	77.2	86.6	55.0	79.1	62.2	87.0	83.4	84.7	78.9	45.3	73.4	65.8	80.3	74.0
Ours ⁴	✓	✓	07++12	76.3	86.3	85.0	77.0	60.9	59.3	81.9	81.1	91.9	55.8	80.6	63.0	90.8	85.1	85.3	80.7	54.9	78.3	70.8	82.8	74.9

¹<http://host.robots.ox.ac.uk:8080/anonymous/XNDVK7.html>, ²<http://host.robots.ox.ac.uk:8080/anonymous/H49PTT.html>,

Other Skill for class imbalance

■ Focal loss

Focal Loss for Dense Object Detection ICCV-2017 1088 citation

$$\text{CE}(p, y) = \begin{cases} -\log(p) & \text{if } y = 1 \\ -\log(1 - p) & \text{otherwise.} \end{cases}$$

$$p_t = \begin{cases} p & \text{if } y = 1 \\ 1 - p & \text{otherwise,} \end{cases}$$

rewrite $\text{CE}(p, y) = \text{CE}(p_t) = -\log(p_t)$.

$$\text{CE}(p_t) = \alpha_t \log(p_t)$$

$$\text{FL}(p_t) = -(1 - p_t)^\gamma \log(p_t)$$

$$\text{FL}(p_t) = \alpha_t (1 - p_t)^\gamma \log(p_t)$$

