Papers On Object Detection

Gong Qiqi

Paper List

- **1. Training Region-based Object Detectors with Online Hard Example Mining**
- **2. R-FCN: Object Detection via Region-based Fully Convolutional Networks**
- **3. Feature pyramid networks for object detection (FPN)**
- **4. Cascade R-CNN: Delving into High Quality Object Detection**
- 5. You Only Look Once: Unified, Real-Time Object Detection (YOLO)
- 6. SSD: Single Shot MultiBox Detector
- 7. Focal Loss for Dense Object Detection

Training Region-based Object Detectors with Online Hard Example Mining(OHEM)

Author: Abhinav Shrivastava; Abhinav Gupta; Ross Girshick

Online Hard Example Mining -- OHEM

- Basic Info.:
  - 2016 CVPR
- Author Introduction:
  - Ross Girshick (RBG) : Facebook AI Research
  - Abhinav Shrivastava, Abhinav Gupta: CMU

Online Hard Example Mining -- OHEM

- Motivation:
  - Unbalanced labels for background examples and foreground examples
  - Overwhelming number of easy examples and a small number of hard examples
- Contribution:
  - Removes need for several heuristics and hyperparameters
  - Consistent and significant boosts in mAP
  - Effectiveness increased
- Inspiration:
  - Could be useful when samples of positive examples are small

Online Hard Example Mining -- OHEM

- Overview of Fast R-CNN:

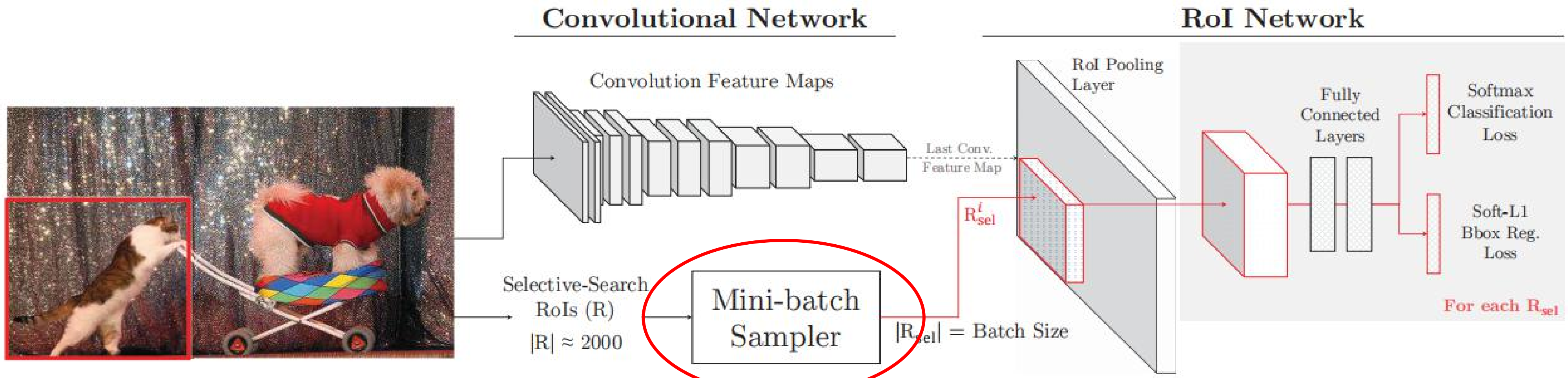


Figure 1: Architecture of the Fast R-CNN approach (see Section 3 for details).

hyperparameters are needed!

Online Hard Example Mining -- OHEM

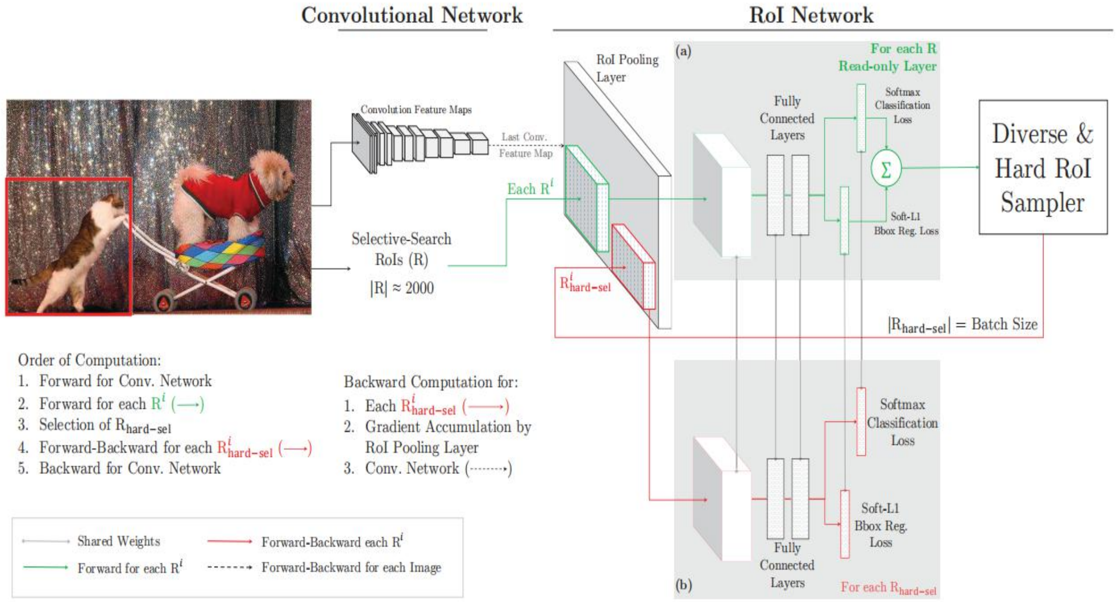
- Overview of Fast R-CNN:
  - Foreground RoIs (**fg**): IoU>=0.5
  - Background RoIs (**bg**): IoU∈[0.1, 0.5)
    - 0.1 is a threshold (hyperparameter) here
    - **Method proposed in this paper elinimate this parameter**
  - Balance fg-bg RoIs:
    - In paper of Fast R-CNN, **ratios of examples of fg to bg is SET to 1:3**
    - **Proposed method elinimate this hyperparameter ratio**

Online Hard Example Mining -- OHEM

- Definition:
  - Hard examples mining:
    - for some period, the model is *fixed* to find new examples
    - for some period, model is trained on *fixed* training set
  - Hard examples: examples with high loss
  - Easy examples: examples with low loss

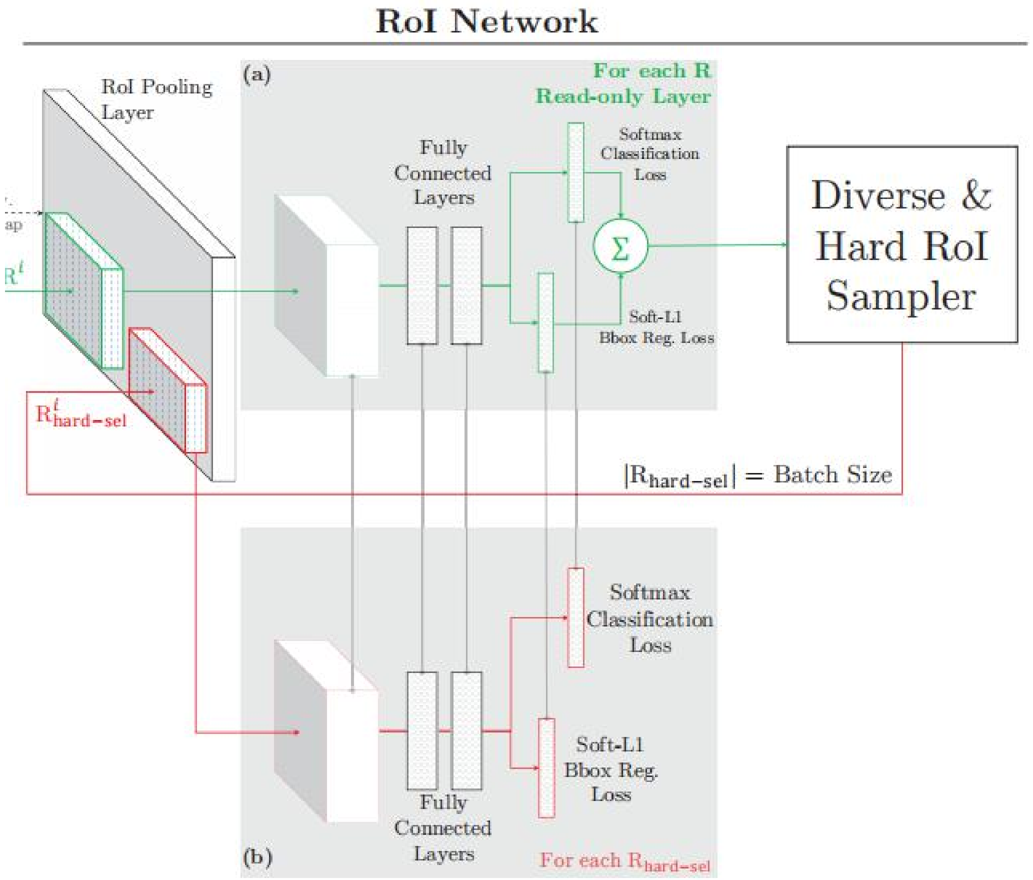
Online Hard Example Mining -- OHEM

- Implementation:
  - Compute feature map first
  - Use **all** RoIs as input RoIs
  - Take hard examples
    - Set loss of easy eg.s as 0
    - Use NMS to remove high overlapped examples



Online Hard Example Mining -- OHEM

- Implementation:



Readonly Layer (a):

- Only perform forward passes

Hard RoI Sampler:

- Take B hard examples for N images

Layer (b):

- Use hard examples to compute forward and backward passes
- Weights are shared between (a) and (b)

Online Hard Example Mining -- OHEM

- A Question:
  - What does Online mean?
  - An explanation
    - 在线学习中，每次录入一条数据（而非一个batch），训练完后直接更新模型
    - 而离线学习是一个batch全部录入完成后，才更新模型

R-FCN: Object Detection via Region-based Fully Convolutional Networks

Author: Jifeng Dai; Yi Li; Kaiming He; Jian Sun

R-FCN

- Basic Info.:
  - 2016 NIPS
- Author Introduction:
  - MSRA(微软亚洲研究院)
  - Jifeng Dai(代季峰)
  - Kaiming He(何恺明) :现 Facebook AI Research
  - Jian Sun(孙剑)

R-FCN

- Background:
  - VGG and AlexNet were widely used in object detection
  - ResNet was put forward
- Motivation:
  - Two-stage object detction algorithm can be speeded up by sharing computation
  - Translation invariance of classification VS Translation variance of detection

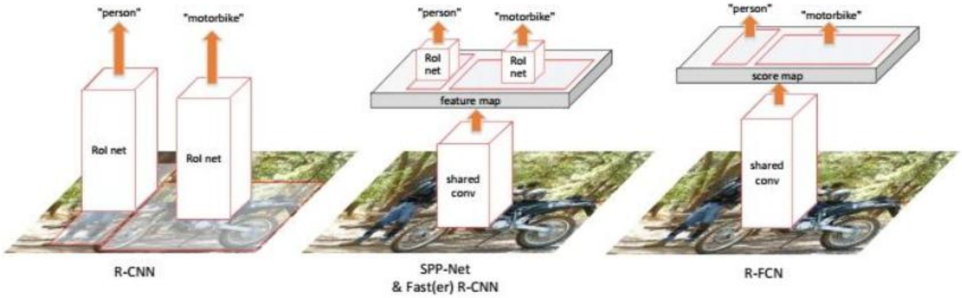


Table 1: Methodologies of *region-based* detectors using **ResNet-101** [10].

	R-CNN [8]	Faster R-CNN [20, 10]	R-FCN [ours]
depth of shared convolutional subnetwork	0	91	101
depth of RoI-wise subnetwork	101	10	**0**

R-FCN

- Contribution:
  - Competitive with Faster R-CNN
  - Speed up for 2.5-20 times than Faster R-CNN

R-FCN

- Implementation:
  - Compute feature map for an image
  - RPN (Region Proposals Network):
    - Compute RoI with the feature map
  - Object Classification:
    - **Position-sensitive score maps : $k^2(C+1)$-d**
    - Vote for each classification
  - Bound Box Regression:
    - $4k^2$-d vector

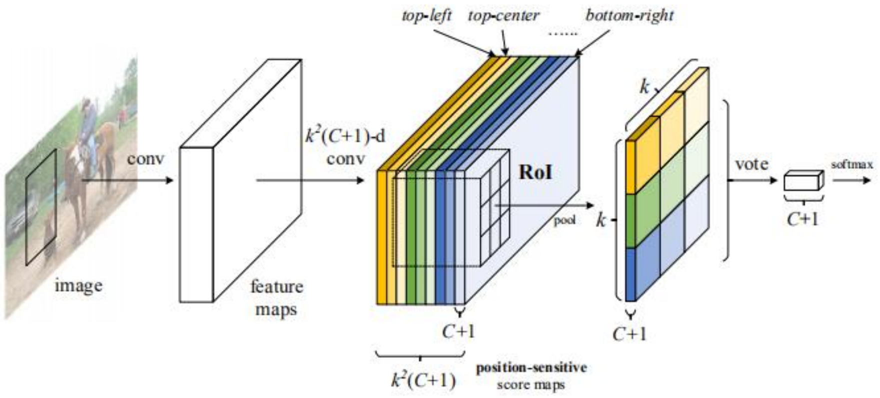


Figure 1: Key idea of **R-FCN** for object detection. In this illustration, there are $k \times k = 3 \times 3$ position-sensitive score maps generated by a fully convolutional network. For each of the $k \times k$ bins in an RoI, pooling is only performed on one of the k^2 maps (marked by different colors).

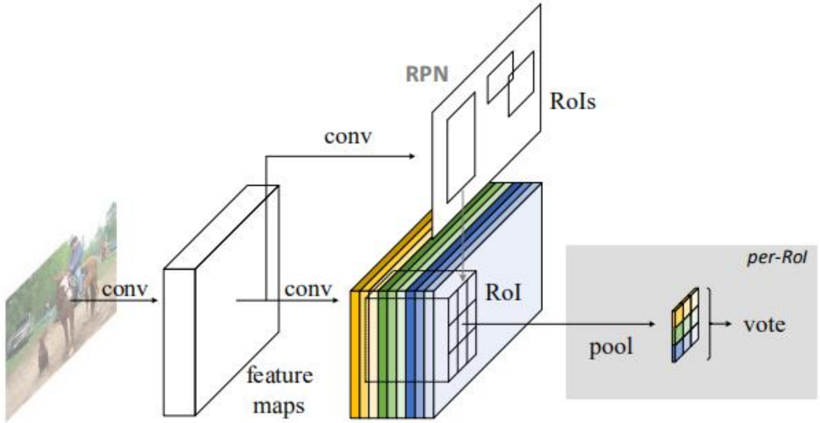


Figure 2: Overall architecture of R-FCN. A Region Proposal Network (RPN) [19] proposes candidate RoIs, which are then applied on the score maps. All learnable weight layers are convolutional and are computed on the entire image; the per-RoI computational cost is negligible.

R-FCN

- Implementation:
  - For category **person** (an example)

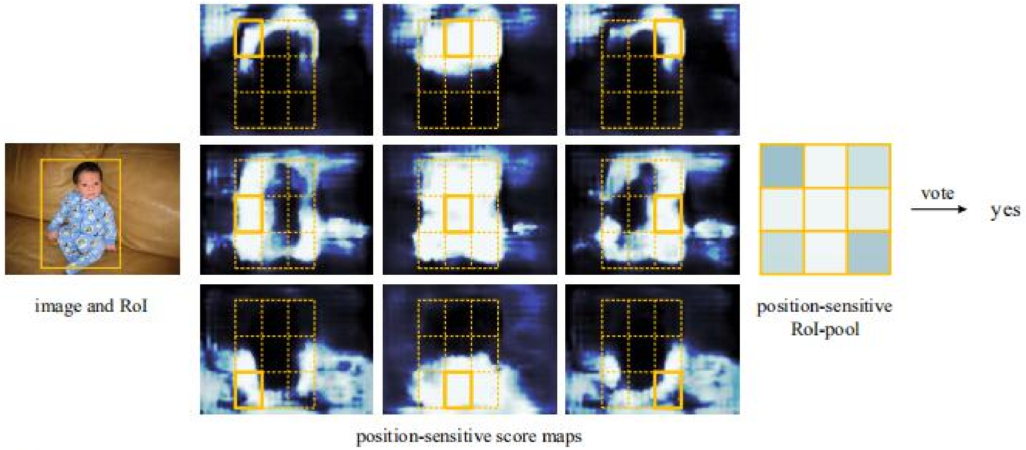


Figure 3: Visualization of R-FCN ($k \times k = 3 \times 3$) for the *person* category.

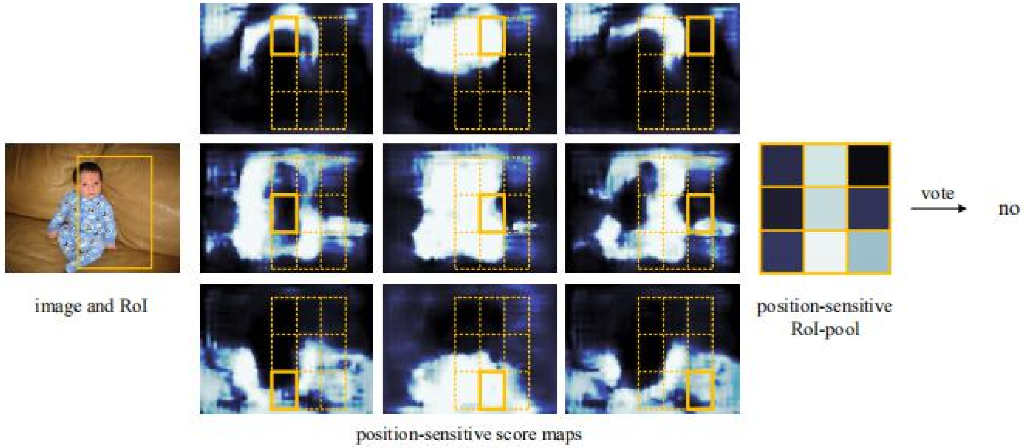


Figure 4: Visualization when an RoI does not correctly overlap the object.

Feature pyramid networks for object detection

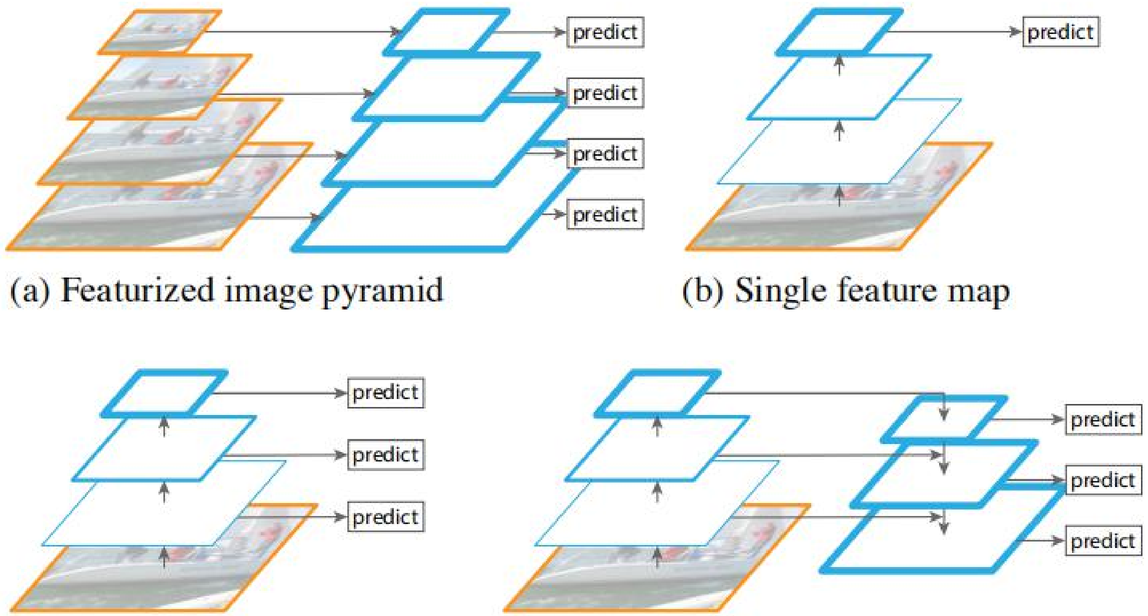
Author: Tsung-Yi Lin, Piotr Dollar´ , Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie

FPN

- Basic Info.:
  - 2017 CVPR
- Team intro.:
  - Facebook AI Research
  - Cornell University and Cornell Tech
- BG & Motivation:
  - Low-level feature - Multi-scale; High-level feature - Strong semantic info.
  - Feature pyramid could help detect objects with different scales
  - Deep learning based detectors avoid using pyramid representations due to compute and memory intensive (impractical for real applications)

FPN

- A simple compare:



(a) Featurized image pyramid (b) Single feature map

(c) Pyramidal feature hierarchy (d) Feature Pyramid Network

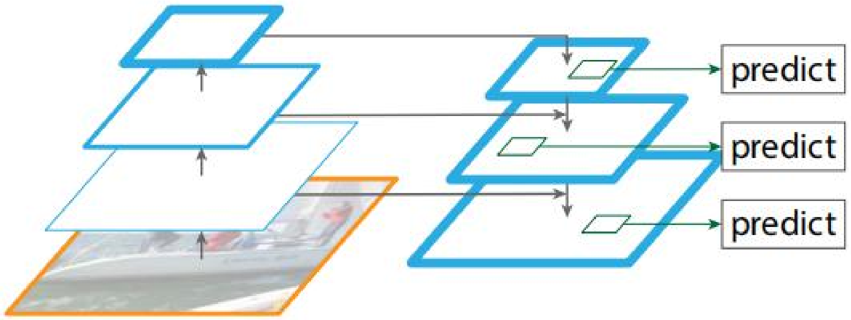
(a) Using IMAGE pyramid to build FEATURE pyramid : slow

(b) Do prediction at the fina feature level: like SPP Net and RCNNs

(c) No upsampling pathway : perform poorly when detecting small scale objects

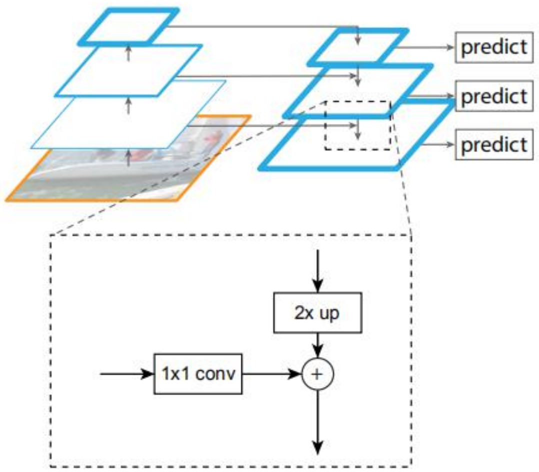
FPN

- Contribution:
  - Proposing bottom-up pathway, top-down pathway and lateral connections (横向连接) structure
  - Creating a feature pyramid owning strong semantics at all scales
  - Do not increase testing times
  - A module which could be used in detection network



FPN

- Implementation:
  - Backbone: ResNet (use the output of Conv2:5, exclude Conv1: large memory)
  - Bottom-up pathway: 2x downsampling
  - Lateral connection: 1*1 conv. (**reduce channel dimensions**)
  - Top-down pathway: 2x upsampling (finally 3*3 conv to generate feature map)

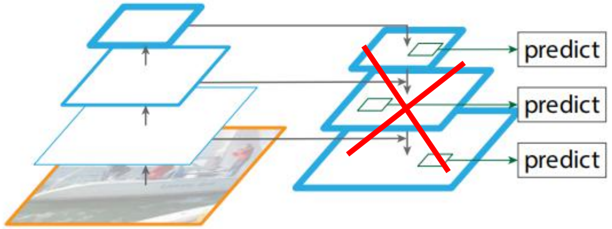


layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		[3×3, 64 3×3, 64] ×2	[3×3, 64 3×3, 64] ×3	[1×1, 64 3×3, 64 1×1, 256] ×3	[1×1, 64 3×3, 64 1×1, 256] ×3	[1×1, 64 3×3, 64 1×1, 256] ×3
conv3_x	28×28	[3×3, 128 3×3, 128] ×2	[3×3, 128 3×3, 128] ×4	[1×1, 128 3×3, 128 1×1, 512] ×4	[1×1, 128 3×3, 128 1×1, 512] ×4	[1×1, 128 3×3, 128 1×1, 512] ×8
conv4_x	14×14	[3×3, 256 3×3, 256] ×2	[3×3, 256 3×3, 256] ×6	[1×1, 256 3×3, 256 1×1, 1024] ×6	[1×1, 256 3×3, 256 1×1, 1024] ×23	[1×1, 256 3×3, 256 1×1, 1024] ×36
conv5_x	7×7	[3×3, 512 3×3, 512] ×2	[3×3, 512 3×3, 512] ×3	[1×1, 512 3×3, 512 1×1, 2048] ×3	[1×1, 512 3×3, 512 1×1, 2048] ×3	[1×1, 512 3×3, 512 1×1, 2048] ×3
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

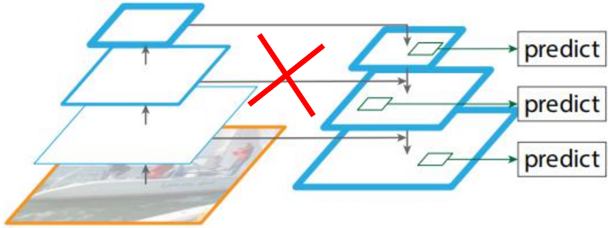
FPN

- Experiments:
  - Work with RPN (Region Proposals Network): To produce suitable anchor boxes
  - **Anchor boxes**: A set of bounding boxes whose aspect ratios and areas (manually)

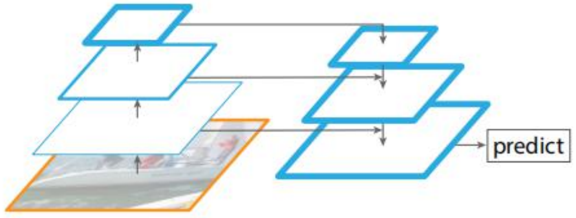
RPN	feature	# anchors	lateral?	top-down?	AR^{100}	AR^{1k}	AR_s^{1k}	AR_m^{1k}	AR_l^{1k}
(a) baseline on conv4	C_4	47k			36.1	48.3	32.0	58.7	62.2
(b) baseline on conv5	C_5	12k			36.3	44.9	25.3	55.5	64.2
(c) **FPN**	$\{P_k\}$	200k	✓	✓	**44.0**	**56.3**	**44.9**	**63.4**	66.2
Ablation experiments follow:									
(d) bottom-up pyramid	$\{P_k\}$	200k	✓		37.4	49.5	30.5	59.9	**68.0**
(e) top-down pyramid, w/o lateral	$\{P_k\}$	200k		✓	34.5	46.1	26.5	57.4	64.7
(f) only finest level	P_2	750k	✓	✓	38.4	51.3	35.1	59.7	67.6



(d)



(e)



(f)

FPN

- Experiments:
  - Work with FCNs

Fast R-CNN	proposals	feature	head	lateral?	top-down?	AP@0.5	AP	AP_s	AP_m	AP_l
(a) baseline on conv4	RPN, $\{P_k\}$	C_4	conv5			54.7	31.9	15.7	36.5	45.5
(b) baseline on conv5	RPN, $\{P_k\}$	C_5	$2fc$			52.9	28.8	11.9	32.4	43.4
(c) **FPN**	RPN, $\{P_k\}$	$\{P_k\}$	$2fc$	✓	✓	**56.9**	**33.9**	**17.8**	**37.7**	**45.8**
Ablation experiments follow:										
(d) bottom-up pyramid	RPN, $\{P_k\}$	$\{P_k\}$	$2fc$	✓		44.9	24.9	10.9	24.4	38.5
(e) top-down pyramid, w/o lateral	RPN, $\{P_k\}$	$\{P_k\}$	$2fc$		✓	54.0	31.3	13.3	35.2	45.3
(f) only finest level	RPN, $\{P_k\}$	P_2	$2fc$	✓	✓	56.3	33.4	17.3	37.3	45.6

Faster R-CNN	proposals	feature	head	lateral?	top-down?	AP@0.5	AP	AP_s	AP_m	AP_l
(*) baseline from He *et al.* [16]†	RPN, C_4	C_4	conv5			47.3	26.3	-	-	-
(a) baseline on conv4	RPN, C_4	C_4	conv5			53.1	31.6	13.2	35.6	**47.1**
(b) baseline on conv5	RPN, C_5	C_5	$2fc$			51.7	28.0	9.6	31.9	43.1
(c) **FPN**	RPN, $\{P_k\}$	$\{P_k\}$	$2fc$	✓	✓	**56.9**	**33.9**	**17.8**	**37.7**	45.8

  - Do well when detecting SMALL objects
  - Competitive when detecting LARGE objects

Cascade R-CNN: Delving into High Quality Object Detection

Author: Zhaowei Cai, Nuno Vasconcelos

Cascade R-CNN

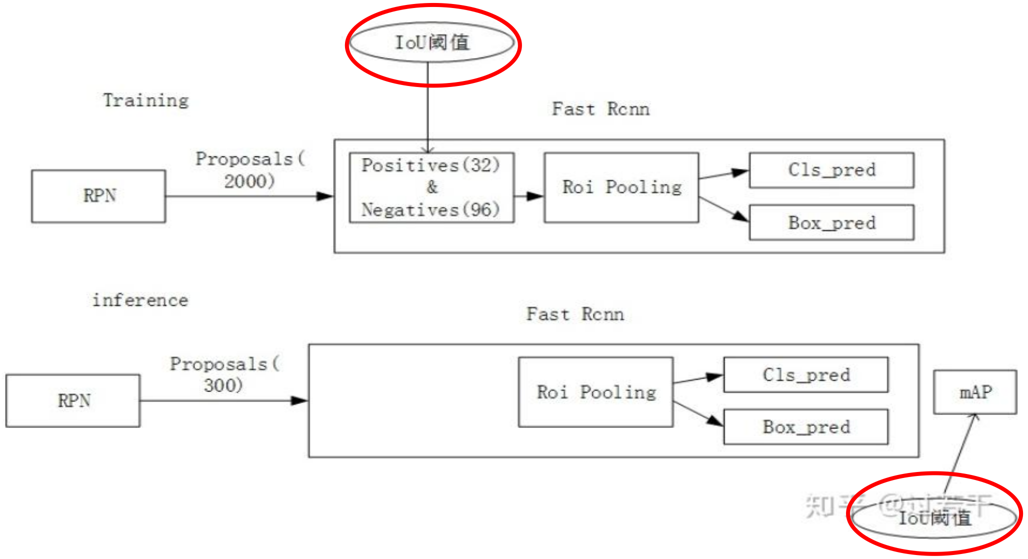
- Basic Info.:
  - 2018 CVPR
- Team intro.:
  - UC San Diego
- BG & Motivation:
  - IoU threshold u is required to define positive/negative examples (**training**)
  - Trade-off: a) low u produces noisy detections; b) high u decreases performance
  - The output of a detector is a good distribution for training the next better detector

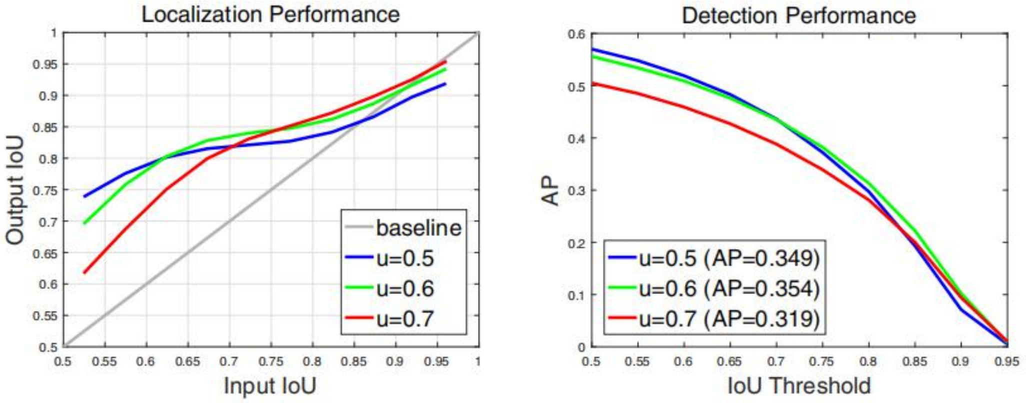
Cascade R-CNN

- Basic Info.:
  - 2018 CVPR
- Team intro.:
  - UC San Diego
- BG & Motivation:
  - IoU threshold u is required to define positive/negative examples (**training**)
  - Trade-off: a) low u produces noisy detections; b) high u decreases performance
  - The output of a detector is a good distribution for training the next better detector

Cascade R-CNN

- Introduction:





(c) Regressor (d) Detector

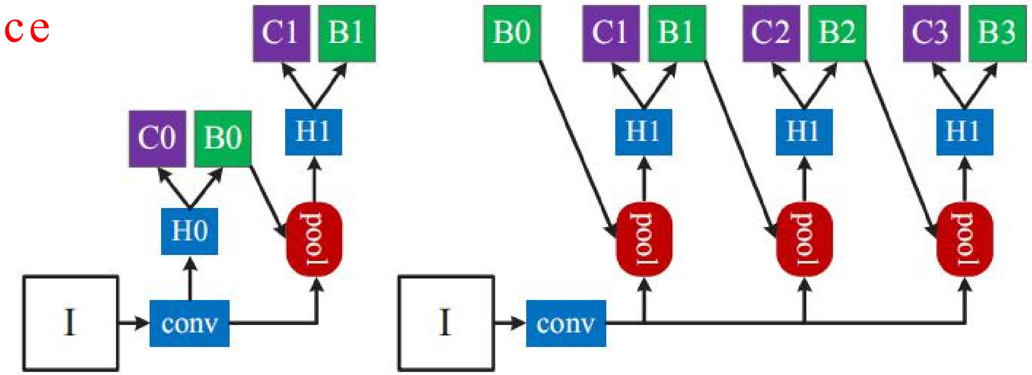
- Some **“close” false positives (close but not correct bboxes)** will be produced with IoU threshold (typical u=0.5) in **training** stage

- (c): A detector optimized at a single IoU level is not necessarily optimal at others（我理解的是两个红圈里的IoU的值要接近，mismatch）

- (d): Simply increase IoU threshold could degrade detection performance

Cascade R-CNN

- Some other work:
  - Iterative Bounding-box Regression(迭代回归):
    - Represent a candidate b-box **b** of image patch x as $f(x, \mathbf{b})$
    - A single regression step of f is suboptimal
    - Use a iterative process to replace it: $f'(x, \mathbf{b}) = f \circ f \circ \cdots \circ f(x, \mathbf{b})$,
    - Problem: No benifit beyond applying f twice



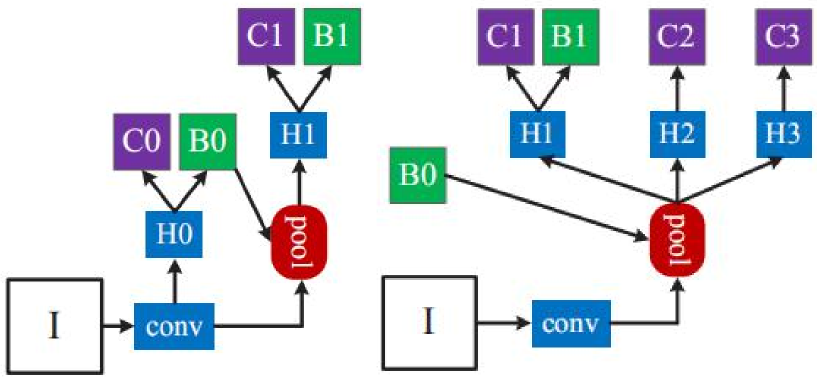
(a) Faster R-CNN (b) Iterative BBox at inference

Cascade R-CNN

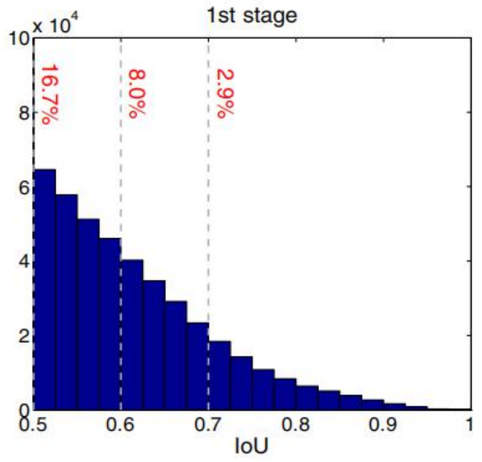
- Some other work:
  - Integral Loss:
    - Calculate loss with different IoU threshold: (分类Loss)

$$L_{cls}(h(x), y) = \sum_{u \in U} L_{cls}(h_u(x), y_u), \quad \text{where } U = \{0.5, 0.55, \cdots, 0.75\}$$

    - Problem: Quick decrease of positive samples with *u* (figure on Right)



(a) Faster R-CNN (c) Integral Loss

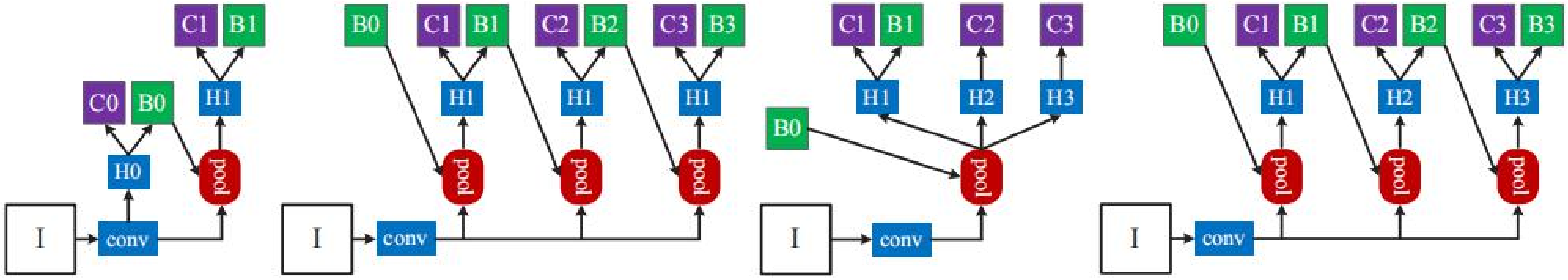


Cascade R-CNN

- Implementataion of Cascade R-CNN:

$$f(x,\mathbf{b}) = f_T \circ f_{T-1} \circ \cdots \circ f_1(x,\mathbf{b}),$$

  - Different from iterative regression: a) f is a resampling procedure which changes distribution; b) used at both training and inference stage; (c) f_t is optimized for all stages
  - Loss: $L(x^t, g) = L_{cls}(h_t(x^t), y^t) + \lambda[y^t \geq 1] L_{loc}(f_t(x^t, \mathbf{b}^t), \mathbf{g}),$ where $\mathbf{b}^t = f_{t-1}(x^{t-1}, \mathbf{b}^{t-1})$ (t is the stage)



(a) Faster R-CNN (b) Iterative BBox at inference (c) Integral Loss (d) Cascade R-CNN

Thanks for Listening~

Gong Qiqi

Reference List

- 1. *https://zhuanlan.zhihu.com/p/59002127*, 深度学习不可忽略之OHEM:Online Hard Example Mining
- 2. *https://blog.csdn.net/wfei101/article/details/79284512*, 目标检测：RFCN算法原理<—>
- 3. *https://www.zhihu.com/search?type=content&q=R-FCN*, 目标检测：R-FCN（NIPS 2016）
- 4. *https://blog.csdn.net/baidu_30594023/article/details/82623623*, FPN全解-最全最详细
- 5. *https://zhuanlan.zhihu.com/p/42553957*, Cascade R-CNN 详细解读