Learning Object Detectors from Scratch

Zhiqiang (Jason) Shen

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Outline

- *DSOD (Deeply Supervised Object Detection)*

  Zhiqiang Shen*, Zhuang Liu*, Jianguo Li, Yu-Gang Jiang, Yurong Chen, and Xiangyang Xue. DSOD: Learning Deeply Supervised Object Detectors from Scratch. ***In ICCV 2017.***

- *GRP-DSOD (Gated Recurrent Feature Pyramids)*

  Zhiqiang Shen*, Honghui Shi*, Rogerio Feris, Liangliang Cao, Shuicheng Yan, Ding Liu, Xinchao Wang, Xiangyang Xue, and Thomas S. Huang. Learning Object Detection from Scratch with Gated Recurrent Feature Pyramids. ***arXiv:1712.00886.***

DSOD: Learning Deeply Supervised Object Detectors from Scratch

Presented at ICCV 2017

Zhiqiang Shen*, Zhuang Liu*, Jianguo Li, Yu-Gang Jiang, Yurong Chen, and Xiangyang Xue. DSOD: Learning Deeply Supervised Object Detectors from Scratch. *In ICCV 2017*.

This repository | Search | Pull requests | Issues | Marketplace | Explore

szq0214 / **DSOD**

Watch 36 | Unstar 385 | Fork 130

Code | Issues 20 | Pull requests 1 | Projects 0 | Wiki | Insights | Settings

DSOD: Learning Deeply Supervised Object Detectors from Scratch. In ICCV 2017.

Edit

Add topics

15 commits | **1** branch | **0** releases | **2** contributors

Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

szq0214 Update README.md — Latest commit 6a2493d on Nov 22, 2017

DSOD300_coco.py	Initial commit	5 months ago
DSOD300_detection_demo.py	add a demo script	5 months ago
DSOD300_pascal++.py	Initial commit	5 months ago
DSOD300_pascal.py	Initial commit	5 months ago
LICENSE	update LICENSE	5 months ago
README.md	Update README.md	a month ago
model_libs.py	change bottleneck to 4k channels	4 months ago
score_DSOD300_pascal.py	Initial commit	5 months ago

README.md

DSOD: Learning Deeply Supervised Object Detectors from Scratch

This repository contains the code for the following paper

Object Detection vs. Other Computer Vision Problems

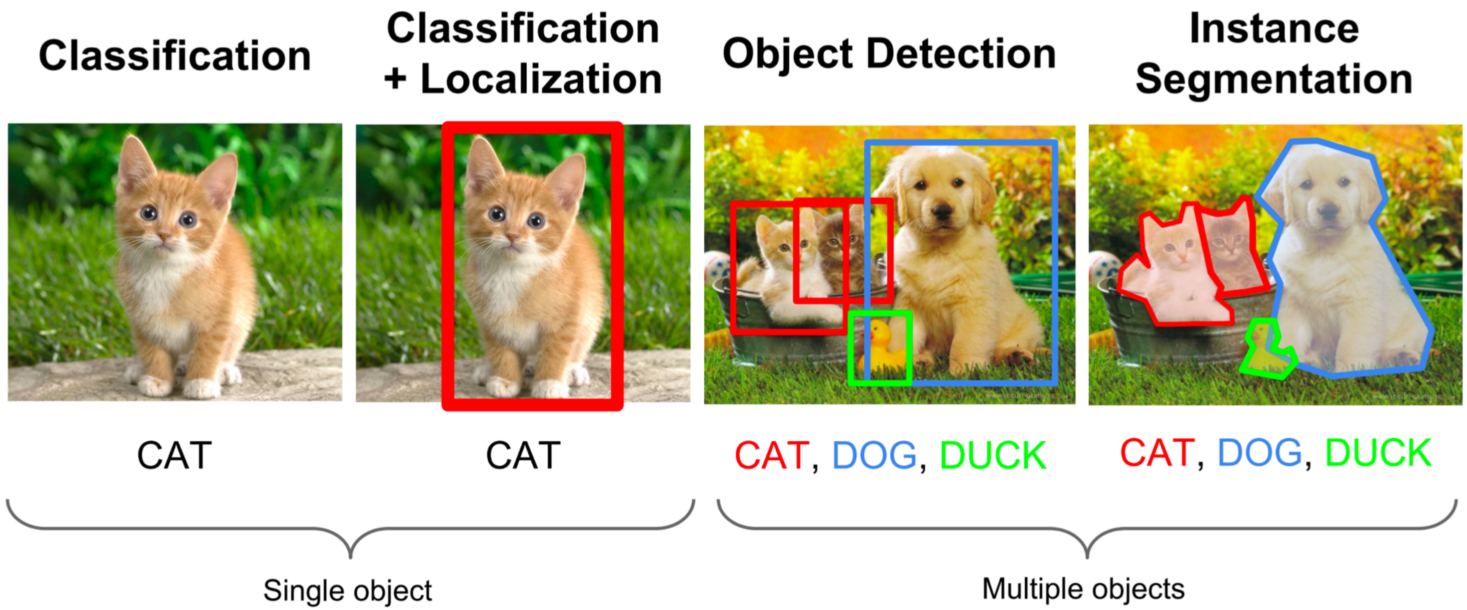
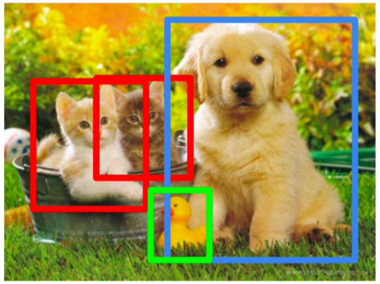


Image from CS231n

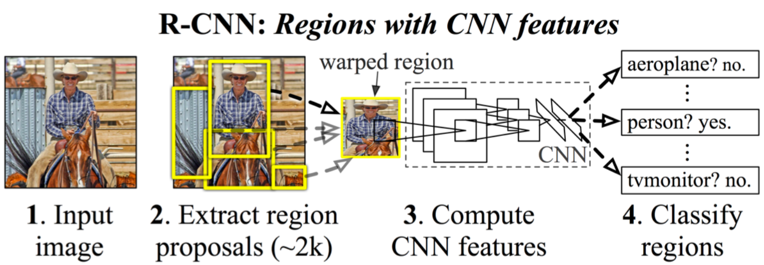
Object Detection

Object Detection

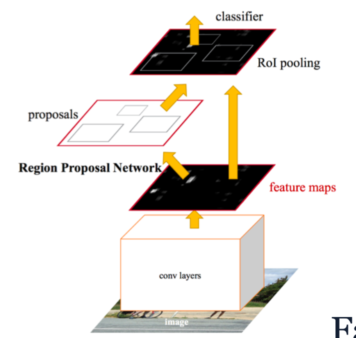


CAT, DOG, DUCK

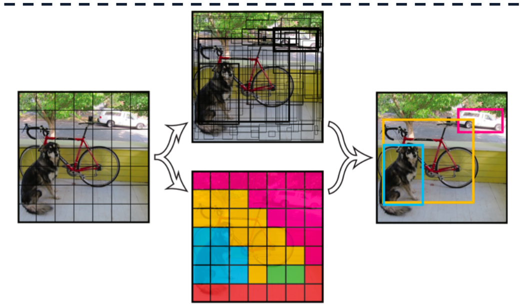
Typical Detection Methods



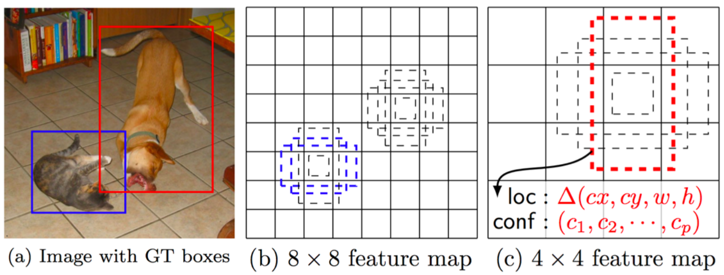
R-CNN



Faster-RCNN

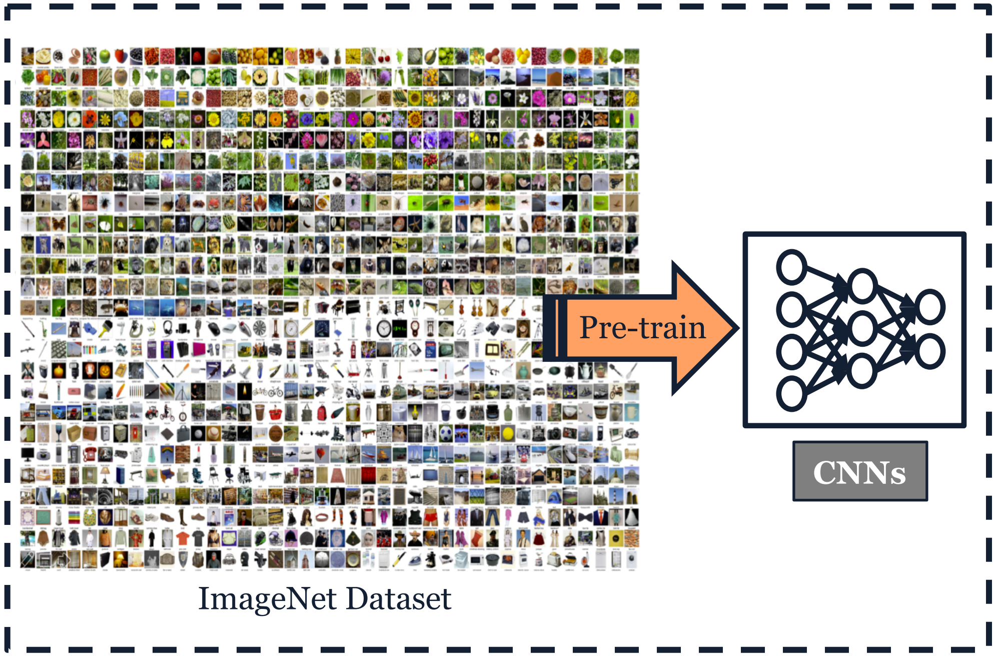


YOLO

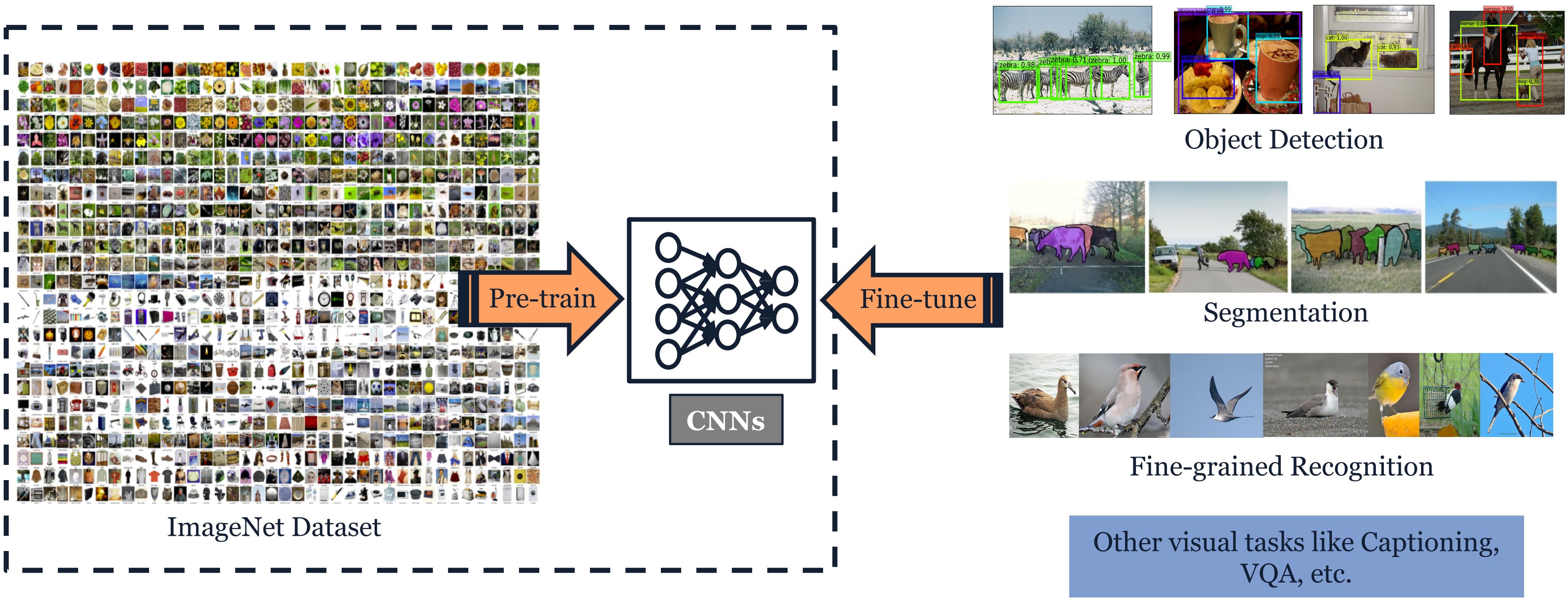


SSD

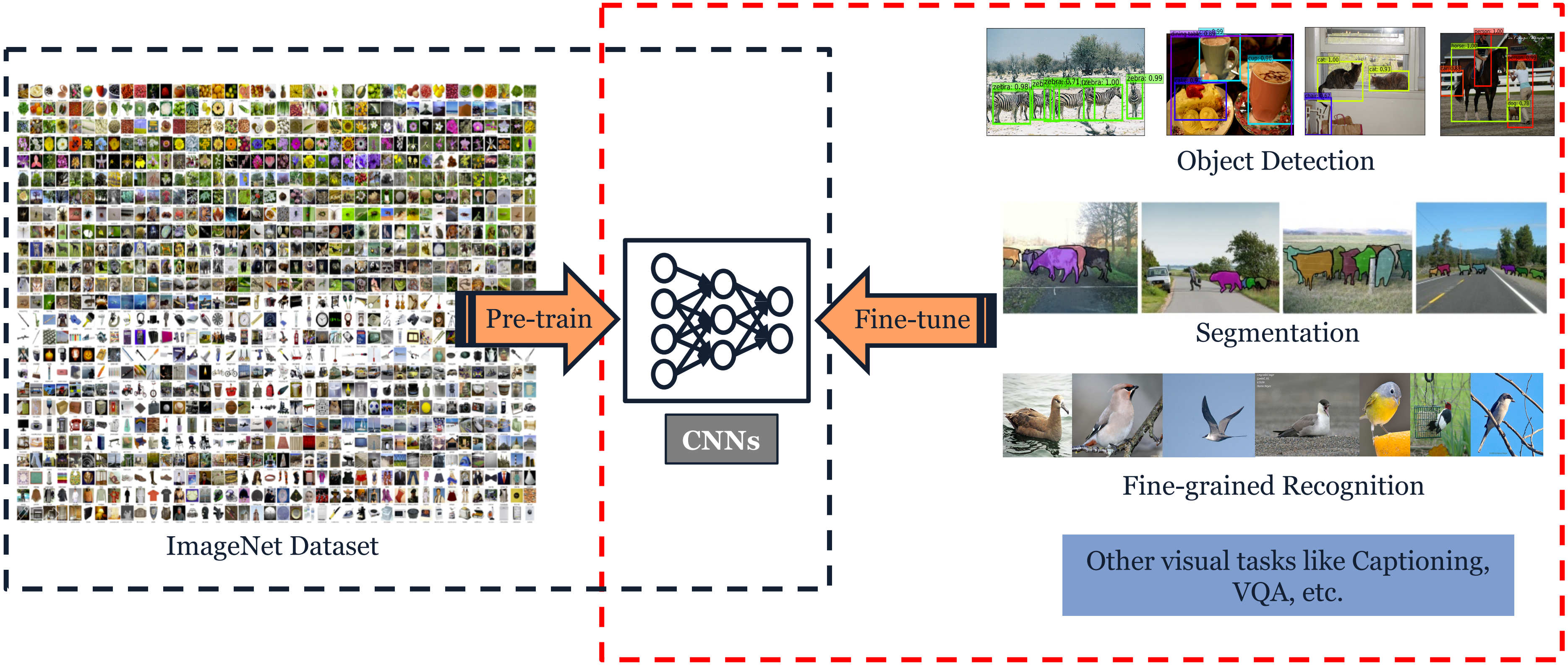
Pre-train & Fine-tune



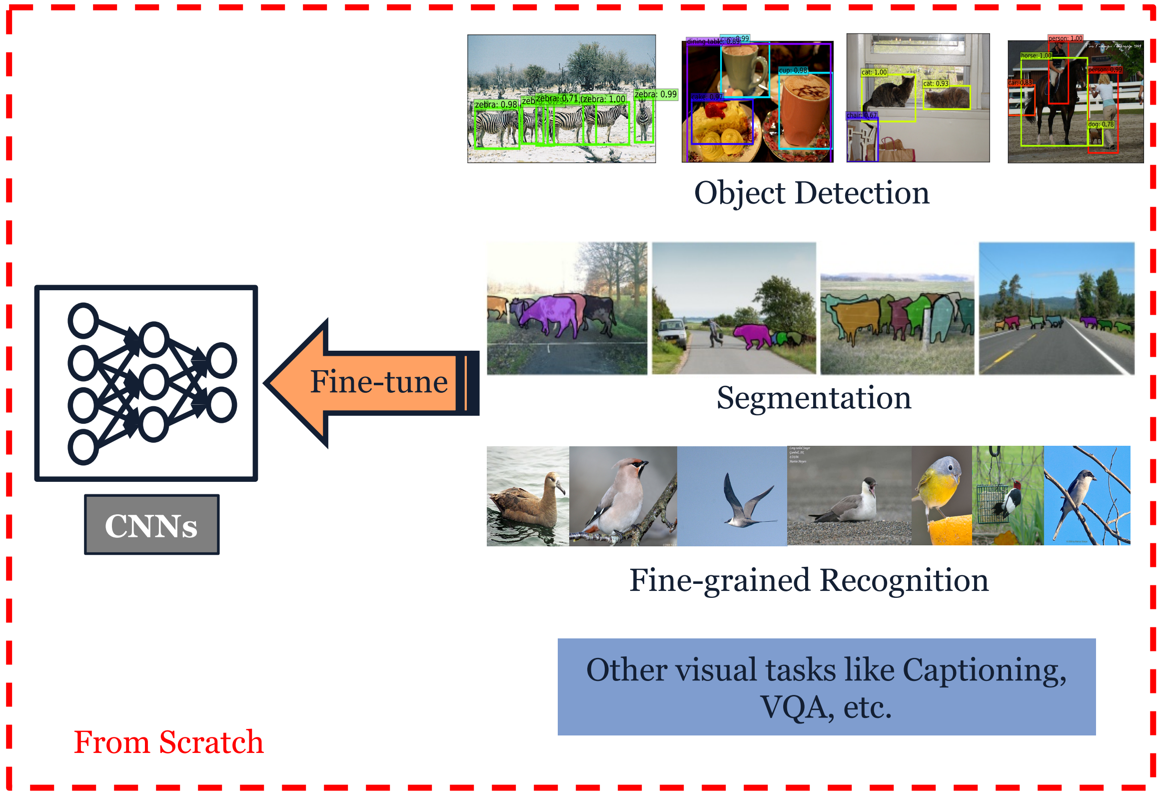
Pre-train & Fine-tune



Pre-train & Fine-tune



Pre-train & Fine-tune



Limitations

➢ ImageNet pre-trained models

- Limited structure design space.
- Learning bias.
- Domain mismatch.

Limitations

➢ ImageNet pre-trained models

- Limited structure design space.
- Learning bias.
- Domain mismatch.

Training from Scratch

Key Findings (training from scratch)

- ***Faster RCNN & R-FCN***: < 15% mAP on VOC without the pre-trained models.

- ***SSD***: 69.6% mAP on VOC.

ROI Pooling

Review: Region of Interest (RoI) pooling

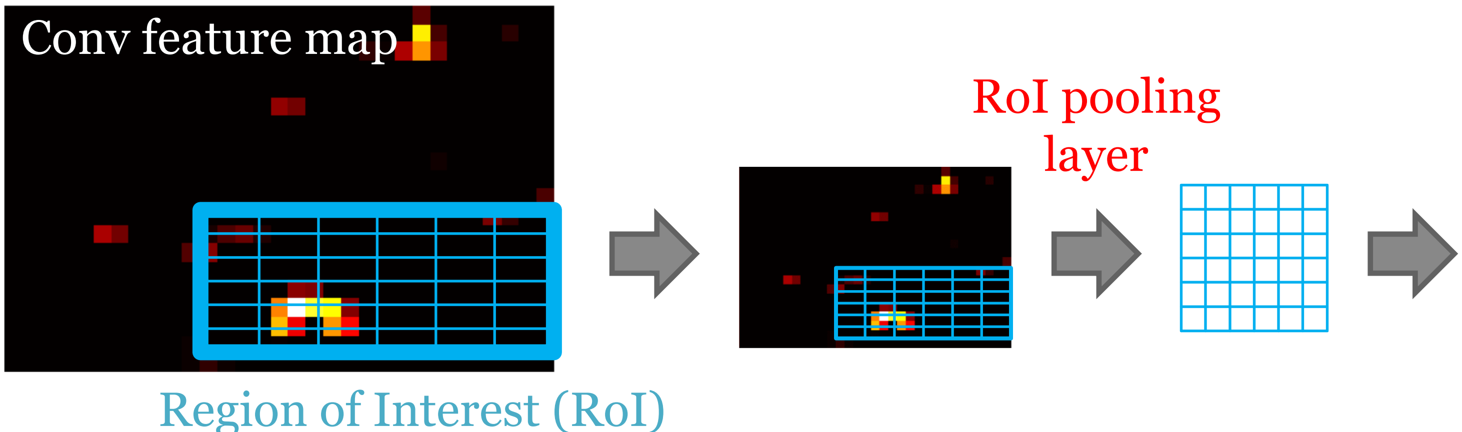


Figure from Ross Girshick

Ross Girshick. "Fast R-CNN". ICCV 2015.

Review: Region of Interest (RoI) pooling

- RoI pooling is just like max pooling

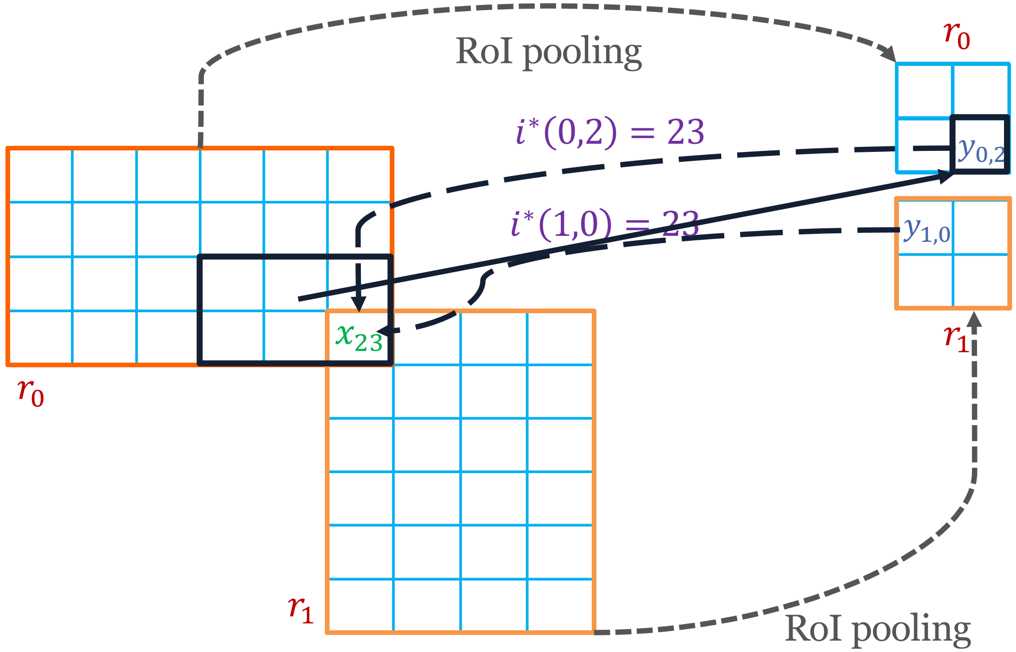


Figure from Ross Girshick

Ross Girshick. "Fast R-CNN". ICCV 2015.

Review: Region of Interest (RoI) pooling

- RoI pooling is just like max pooling
- Forward / backward

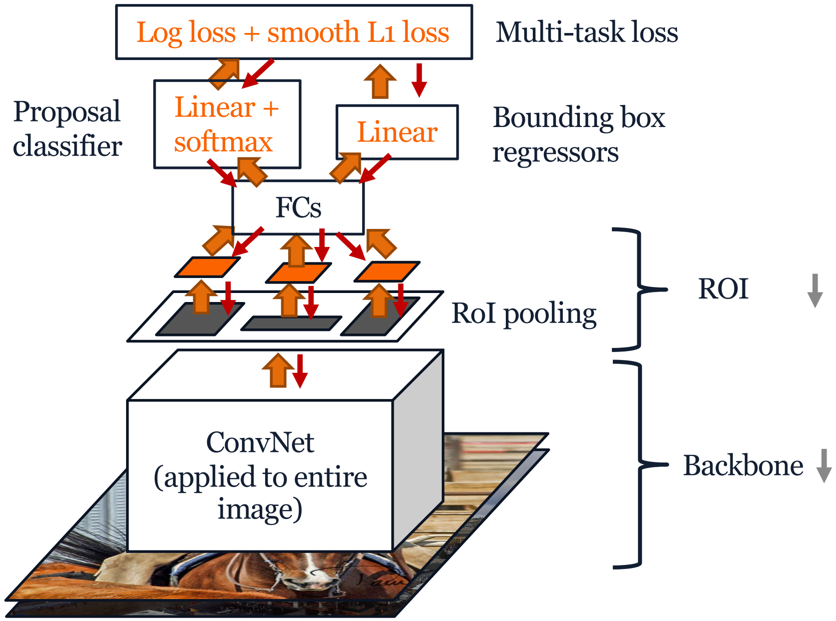


Figure from Ross Girshick

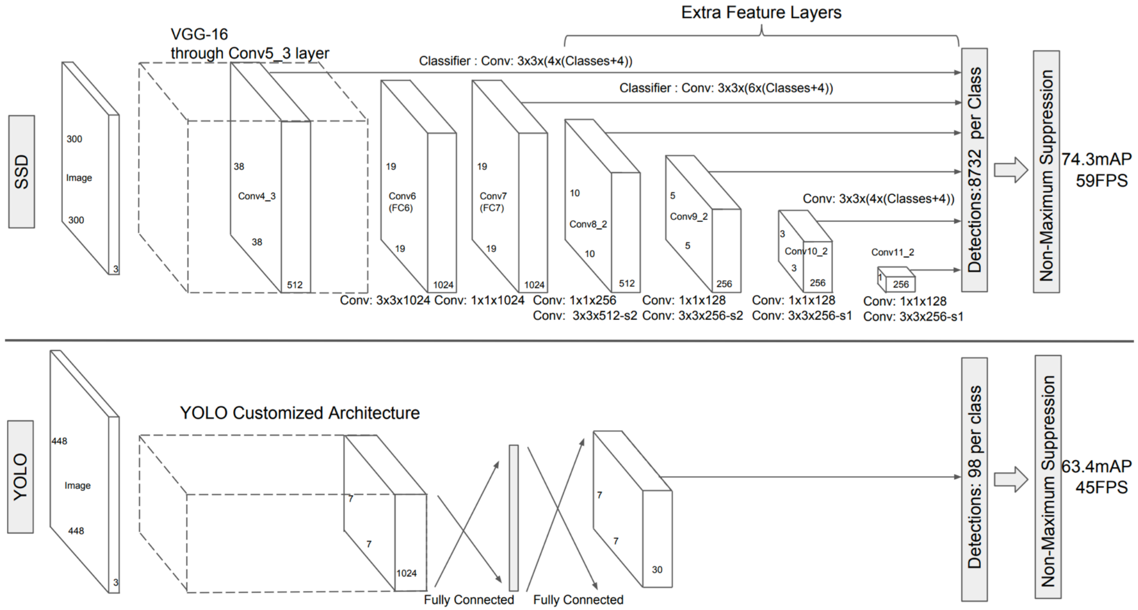
Ross Girshick. "Fast R-CNN". ICCV 2015.

Principles

- Proposal-free.

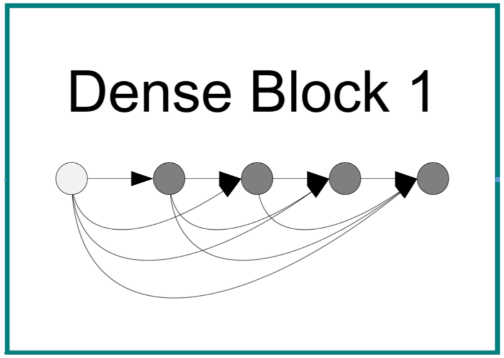
One-stage pipeline

- Proposal-free.



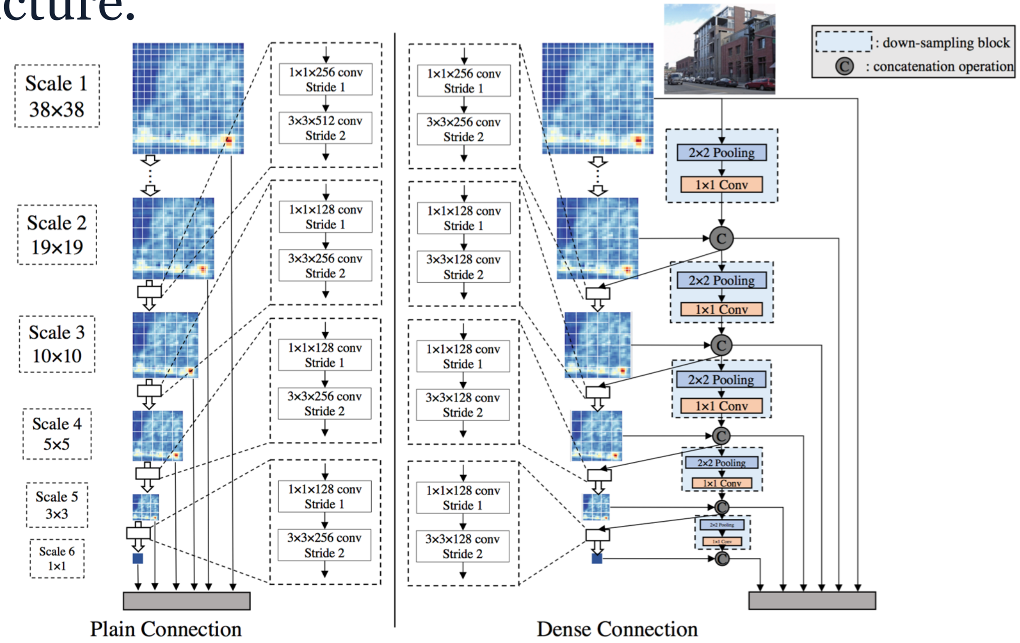
Principles

- Proposal-free.
- Deep Supervision.



Principles

- ➢ Proposal-free.
- ➢ Deep Supervision.
- ➢ Dense Prediction Structure.



Principles

- Proposal-free.
- Deep Supervision.
- Dense Prediction Structure.
- Stem Block.

Layers		Output Size (Input 3×300 × 300)	DSOD
Stem	Convolution	64×150×150	3×3 conv, stride 2
	Convolution	64×150×150	3×3 conv, stride 1
	Convolution	128×150×150	3×3 conv, stride 1
	Pooling	128×75×75	2×2 max pool, stride 2

DSOD architecture

Layers		Output Size (Input 3×300 × 300)	DSOD
Stem	Convolution	64×150×150	3×3 conv, stride 2
	Convolution	64×150×150	3×3 conv, stride 1
	Convolution	128×150×150	3×3 conv, stride 1
	Pooling	128×75×75	2×2 max pool, stride 2
Dense Block (1)		416×75×75	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)		416×75×75	1×1 conv
		416×38×38	2×2 max pool, stride 2
Dense Block (2)		800×38×38	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 8$
Transition Layer (2)		800×38×38	1×1 conv
		800×19×19	2×2 max pool, stride 2
Dense Block (3)		1184×19×19	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 8$
Transition w/o Pooling Layer (1)		1184×19×19	1×1 conv
Dense Block (4)		1568×19×19	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 8$
Transition w/o Pooling Layer (2)		1568×19×19	1×1 conv
DSOD Prediction Layers		–	Plain/Dense

Table 1: DSOD architecture (growth rate $k = 48$ in each dense block).

Experiments

- Ablation Study on PASCAL VOC2007

	DSOD300							
transition w/o pooling?		✓	✓	✓	✓	✓	✓	✓
hi-comp factor θ?			✓	✓	✓	✓	✓	✓
wide bottleneck?				✓	✓	✓	✓	✓
wide 1st conv-layer?					✓	✓	✓	✓
big growth rate?						✓	✓	✓
stem block?							✓	✓
dense pred-layers?								✓
VOC 2007 mAP	59.9	61.6	64.5	68.6	69.7	74.5	77.3	77.7

Experiments

- Ablation Study on PASCAL VOC2007
- Results on PASCAL VOC2007

Method	data	pre-train	backbone network	prediction layer	speed (*fps*)	# parameters	input size	mAP
Faster RCNN [27]	07+12	✓	VGGNet	-	7	134.7M	$\sim 600 \times 1000$	73.2
Faster RCNN [27]	07+12	✓	ResNet-101	-	2.4*	-	$\sim 600 \times 1000$	76.4
R-FCN [19]	07+12	✓	ResNet-50	-	11	31.9M	$\sim 600 \times 1000$	77.4
R-FCN [19]	07+12	✓	ResNet-101	-	9	50.9M	$\sim 600 \times 1000$	79.5
R-FCN$_{\text{multi-sc}}$ [19]	07+12	✓	ResNet-101	-	9	50.9M	$\sim 600 \times 1000$	80.5
YOLOv2 [26]	07+12	✓	Darknet-19	-	81	-	352×352	73.7
SSD300 [21]	07+12	✓	VGGNet	Plain	46	26.3M	300×300	75.8
SSD300* [21]	07+12	✓	VGGNet	Plain	46	26.3M	300×300	77.2
Faster RCNN	07+12	✗	VGGNet/ResNet-101/DenseNet	Failed				
R-FCN	07+12	✗	VGGNet/ResNet-101/DenseNet	Failed				
SSD300S†	07+12	✗	ResNet-101	Plain	12.1	52.8M	300×300	63.8*
SSD300S†	07+12	✗	VGGNet	Plain	46	26.3M	300×300	69.6
SSD300S†	07+12	✗	VGGNet	Dense	37	26.0M	300×300	70.4
DSOD300	07+12	✗	DS/64-192-48-1	Plain	20.6	18.2M	300×300	77.3
DSOD300	07+12	✗	DS/64-192-48-1	Dense	17.4	14.8M	300×300	77.7
DSOD300	07+12+COCO	✗	DS/64-192-48-1	Dense	17.4	14.8M	300×300	81.7

Table 4: **PASCAL VOC 2007 `test` detection results.** SSD300* is updated version by the authors after the paper publication. SSD300S† indicates training SSD300* from scratch with ResNet-101 or VGGNet, which serves as our baseline. Note that the speed of Faster R-CNN with ResNet-101 (2.4 *fps*) is tested on K40, while others are tested on Titan X. The result of SSD300S with ResNet-101 (63.8% mAP, without the pre-trained model) is produced with the default setting of SSD, which may not be optimal.

Experiments

- Ablation Study on PASCAL VOC2007
- Results on PASCAL VOC2007

Method	data	pre-train	backbone network	prediction layer	speed (*fps*)	# parameters	input size	mAP
Faster RCNN [27]	07+12	✓	VGGNet	-	7	134.7M	$\sim 600 \times 1000$	73.2
Faster RCNN [27]	07+12	✓	ResNet-101	-	2.4*	-	$\sim 600 \times 1000$	76.4
R-FCN [19]	07+12	✓	ResNet-50	-	11	31.9M	$\sim 600 \times 1000$	77.4
R-FCN [19]	07+12	✓	ResNet-101	-	9	50.9M	$\sim 600 \times 1000$	79.5
R-FCN$_{\text{multi-sc}}$ [19]	07+12	✓	ResNet-101	-	9	50.9M	$\sim 600 \times 1000$	80.5
YOLOv2 [26]	07+12	✓	Darknet-19	-	81	-	352×352	73.7
SSD300 [21]	07+12	✓	VGGNet	Plain	46	26.3M	300×300	75.8
SSD300* [21]	07+12	✓	VGGNet	Plain	46	26.3M	300×300	77.2
Faster RCNN	07+12	✗	VGGNet/ResNet-101/DenseNet	Failed				
R-FCN	07+12	✗	VGGNet/ResNet-101/DenseNet	Failed				
SSD300S†	07+12	✗	ResNet-101	Plain	12.1	52.8M	300×300	63.8*
SSD300S†	07+12	✗	VGGNet	Plain	46	26.3M	300×300	69.6
SSD300S†	07+12	✗	VGGNet	Dense	37	26.0M	300×300	70.4
DSOD300	07+12	✗	DS/64-192-48-1	Plain	20.6	18.2M	300×300	77.3
DSOD300	07+12	✗	DS/64-192-48-1	Dense	17.4	14.8M	300×300	77.7
DSOD300	07+12+COCO	✗	DS/64-192-48-1	Dense	17.4	14.8M	300×300	81.7

Table 4: **PASCAL VOC 2007 `test` detection results.** SSD300* is updated version by the authors after the paper publication. SSD300S† indicates training SSD300* from scratch with ResNet-101 or VGGNet, which serves as our baseline. Note that the speed of Faster R-CNN with ResNet-101 (2.4 *fps*) is tested on K40, while others are tested on Titan X. The result of SSD300S with ResNet-101 (63.8% mAP, without the pre-trained model) is produced with the default setting of SSD, which may not be optimal.

Experiments

- Ablation Study on PASCAL VOC2007
- Results on PASCAL VOC2007
- Results on PASCAL VOC2012

Method	data	backbone network	pre-train	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
ION [1]	07+12+S	VGGNet	✓	76.4	87.5	84.7	76.8	63.8	58.3	82.6	79.0	90.9	57.8	82.0	64.7	88.9	86.5	84.7	82.3	51.4	78.2	69.2	85.2	73.5
Faster RCNN [27]	07++12	ResNet-101	✓	73.8	86.5	81.6	77.2	58.0	51.0	78.6	76.6	93.2	48.6	80.4	59.0	92.1	85.3	84.8	80.7	48.1	77.3	66.5	84.7	65.6
R-FCNmulti-sc [19]	07++12	ResNet-101	✓	77.6	86.9	83.4	81.5	63.8	62.4	81.6	81.1	93.1	58.0	83.8	60.8	92.7	86.0	84.6	84.4	59.0	80.8	68.6	86.1	72.9
YOLOv2 [26]	07++12	Darknet-19	✓	73.4	86.3	82.0	74.8	59.2	51.8	79.8	76.5	90.6	52.1	78.2	58.5	89.3	82.5	83.4	81.3	49.1	77.2	62.4	83.8	68.7
SSD300* [21]	07++12	VGGNet	✓	75.8	88.1	82.9	74.4	61.9	47.6	82.7	78.8	91.5	58.1	80.0	64.1	89.4	85.7	85.5	82.6	50.2	79.8	73.6	86.6	72.1
DSOD300	07++12	DS/64-192-48-1	✗	76.3	89.4	85.3	72.9	62.7	49.5	83.6	80.6	92.1	60.8	77.9	65.6	88.9	85.5	86.8	84.6	51.1	77.7	72.3	86.0	72.2
DSOD300	07++12+COCO	DS/64-192-48-1	✗	79.3	90.5	87.4	77.5	67.4	57.7	84.7	83.6	92.6	64.8	81.3	66.4	90.1	87.8	88.1	87.3	57.9	80.3	75.6	88.1	76.7

Table 5: **PASCAL VOC 2012 `test` detection results. 07+12**: 07 `trainval` + 12 `trainval`, **07+12+S**: 07+12 plus segmentation labels, **07++12**: 07 `trainval` + 07 `test` + 12 `trainval`. Result links are DSOD300 (07+12) : http://host.robots.ox.ac.uk:8080/anonymous/PIOBKI.html; DSOD300 (07+12+COCO): http://host.robots.ox.ac.uk:8080/anonymous/I0UUHO.html.

Experiments

- Ablation Study on PASCAL VOC2007
- Results on PASCAL VOC2007
- Results on PASCAL VOC2012
- **Results on MS COCO**

Method	data	network	pre-train	Avg. Precision, IoU: 0.5:0.95	0.5	0.75	Avg. Precision, Area: S	M	L	Avg. Recall, #Dets: 1	10	100	Avg. Recall, Area: S	M	L
Faster RCNN [27]	trainval	VGGNet	✓	21.9	42.7	-	-	-	-	-	-	-	-	-	-
ION [1]	train	VGGNet	✓	23.6	43.2	23.6	6.4	24.1	38.3	23.2	32.7	33.5	10.1	37.7	53.6
R-FCN [19]	trainval	ResNet-101	✓	29.2	51.5	-	10.3	32.4	43.3	-	-	-	-	-	-
R-FCN$_{\text{multi-sc}}$ [19]	trainval	ResNet-101	✓	29.9	51.9	-	10.8	32.8	45.0	-	-	-	-	-	-
SSD300 (Huang et al.) [11]	< trainval35k	MobileNet	✓	18.8	-	-	-	-	-	-	-	-	-	-	-
SSD300 (Huang et al.) [11]	< trainval35k	Inception-v2	✓	21.6	-	-	-	-	-	-	-	-	-	-	-
YOLOv2 [26]	trainval35k	Darknet-19	✓	21.6	44.0	19.2	5.0	22.4	35.5	20.7	31.6	33.3	9.8	36.5	54.4
SSD300* [21]	trainval35k	VGGNet	✓	25.1	43.1	25.8	6.6	25.9	41.4	23.7	35.1	37.2	11.2	40.4	58.4
DSOD300	trainval	DS/64-192-48-1	✗	29.3	47.3	30.6	9.4	31.5	47.0	27.3	40.7	43.0	16.7	47.1	65.0

Table 6: **MS COCO `test-dev 2015` detection results.**

Experiments

- Ablation Study on PASCAL VOC2007
- Results on PASCAL VOC2007
- Results on PASCAL VOC2012
- Results on MS COCO

Method	data	network	pre-train	Avg. Precision, IoU: 0.5:0.95	0.5	0.75	Avg. Precision, Area: S	M	L	Avg. Recall, #Dets: 1	10	100	Avg. Recall, Area: S	M	L
Faster RCNN [27]	trainval	VGGNet	✓	21.9	42.7	-	-	-	-	-	-	-	-	-	-
ION [1]	train	VGGNet	✓	23.6	43.2	23.6	6.4	24.1	38.3	23.2	32.7	33.5	10.1	37.7	53.6
R-FCN [19]	trainval	ResNet-101	✓	29.2	51.5	-	10.3	32.4	43.3	-	-	-	-	-	-
R-FCN$_{multi-sc}$ [19]	trainval	ResNet-101	✓	29.9	51.9	-	10.8	32.8	45.0	-	-	-	-	-	-
SSD300 (Huang et al.) [11]	< trainval35k	MobileNet	✓	18.8	-	-	-	-	-	-	-	-	-	-	-
SSD300 (Huang et al.) [11]	< trainval35k	Inception-v2	✓	21.6	-	-	-	-	-	-	-	-	-	-	-
YOLOv2 [26]	trainval35k	Darknet-19	✓	21.6	44.0	19.2	5.0	22.4	35.5	20.7	31.6	33.3	9.8	36.5	54.4
SSD300* [21]	trainval35k	VGGNet	✓	25.1	43.1	25.8	6.6	25.9	41.4	23.7	35.1	37.2	11.2	40.4	58.4
DSOD300	trainval	DS/64-192-48-1	✗	29.3	47.3	30.6	9.4	31.5	47.0	27.3	40.7	43.0	16.7	47.1	65.0

Table 6: **MS COCO `test-dev 2015` detection results.**

Examples of Detection Results

- Paper: https://arxiv.org/abs/1708.01241
- Code & Models: https://github.com/szq0214/DSOD
- Network Structure:

http://ethereon.github.io/netscope/#/gist/b17d01f3131e2a60f90

57b5d3eb9e04d

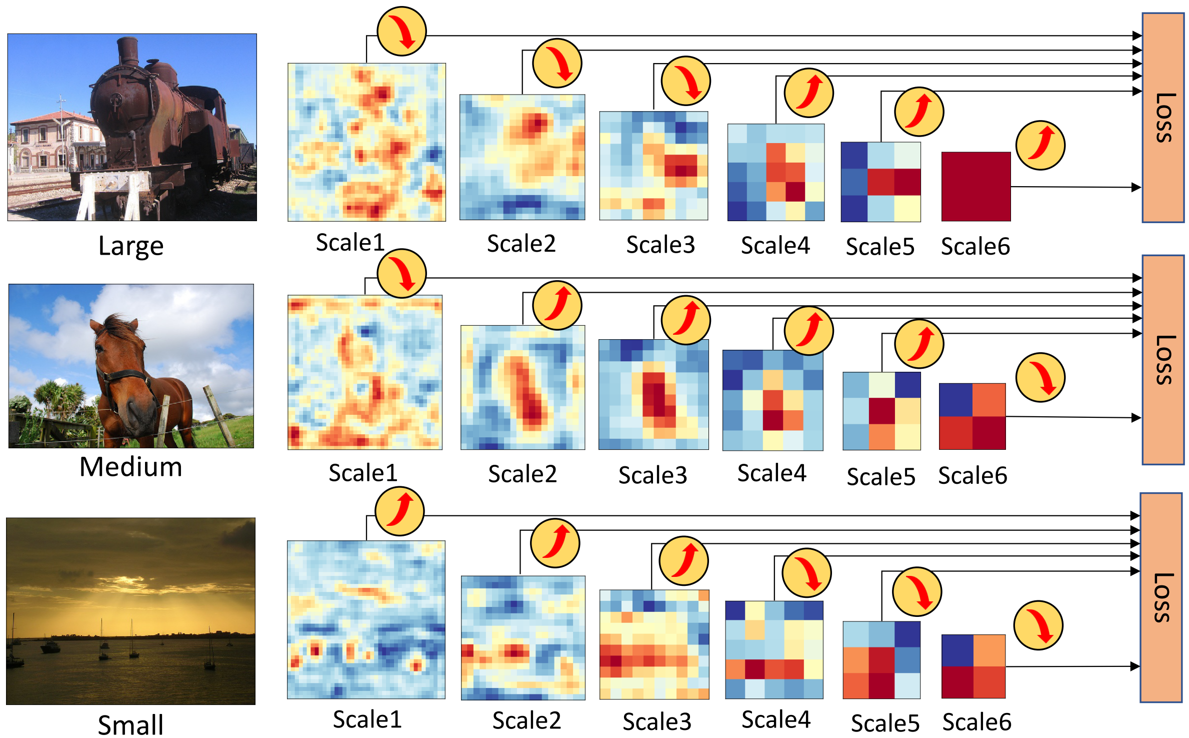
Summary of DSOD

- Learning object detectors from scratch is necessary.
- Limitations with pre-trained models.
- Principles.
- The first framework that can train object detection networks from scratch with state-of-the-art performance.

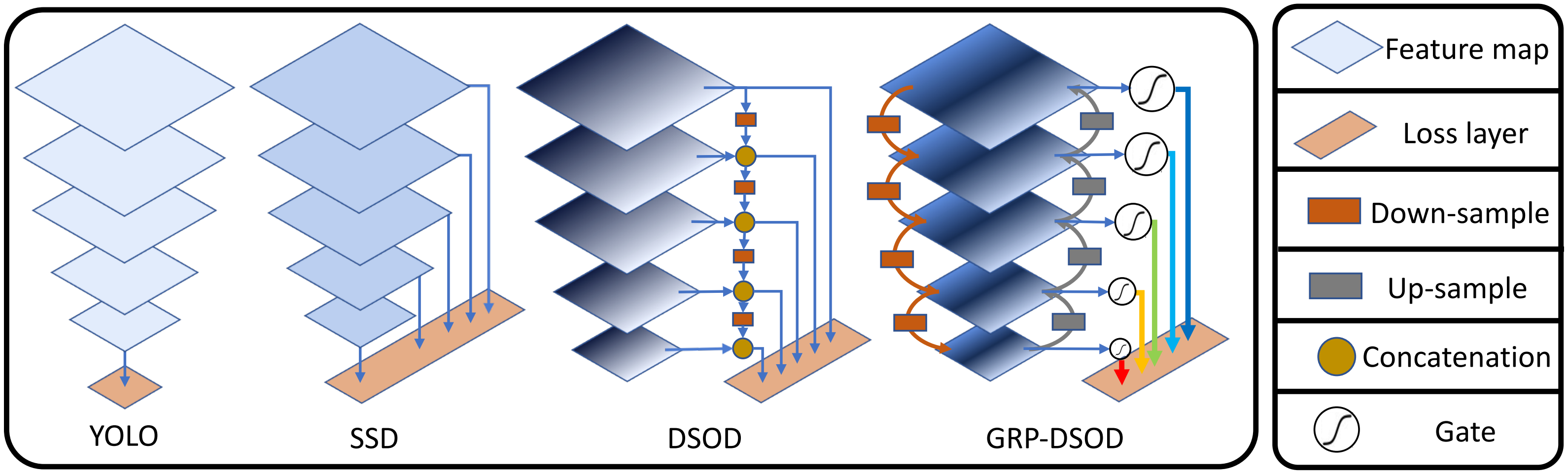
Learning Object Detectors from Scratch with Gated Recurrent Feature Pyramids

Zhiqiang Shen*, Honghui Shi*, Rogerio Feris, Liangliang Cao, Shuicheng Yan, Ding Liu, Xinchao Wang, Xiangyang Xue, and Thomas S. Huang. "Learning Object Detectors from Scratch with Gated Recurrent Feature Pyramids." *arXiv preprint arXiv:1712.00886* (2017).

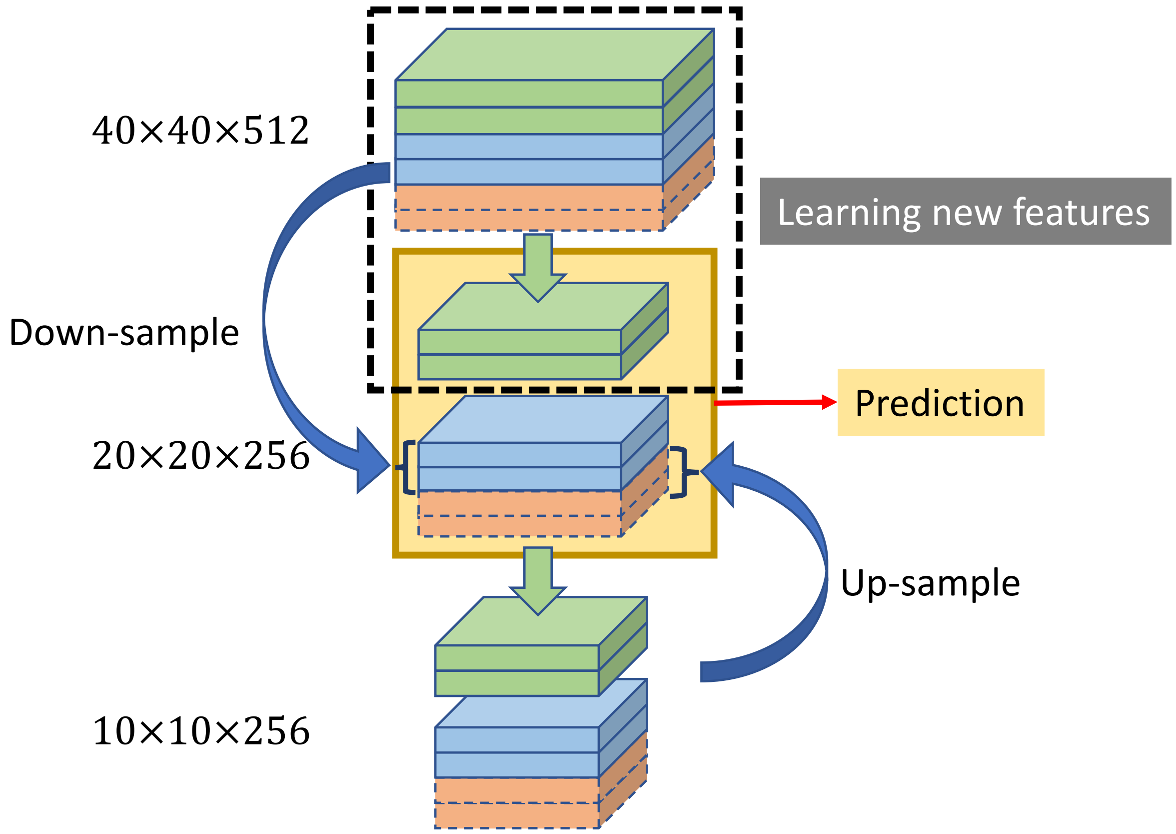
Our Main Motivation



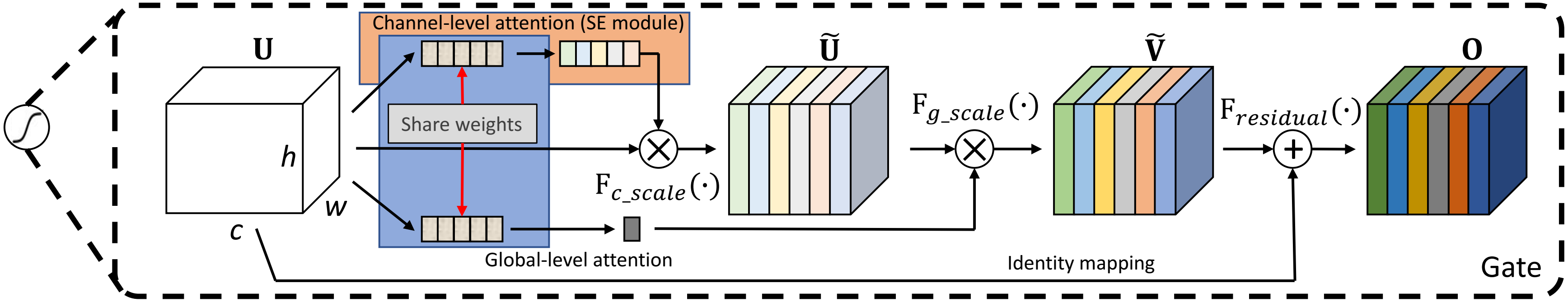
YOLO, SSD, DSOD and GRP-DSOD



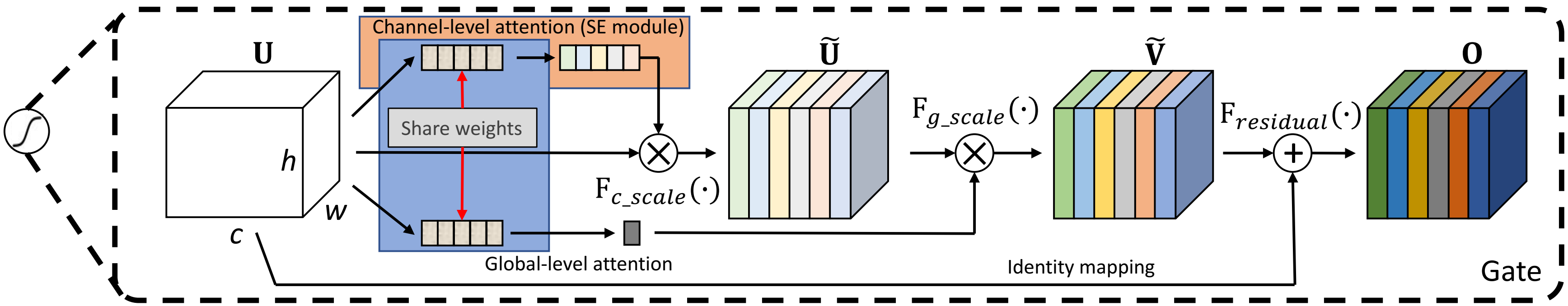
Recurrent Feature Pyramids



Gate Structure



Gate Structure



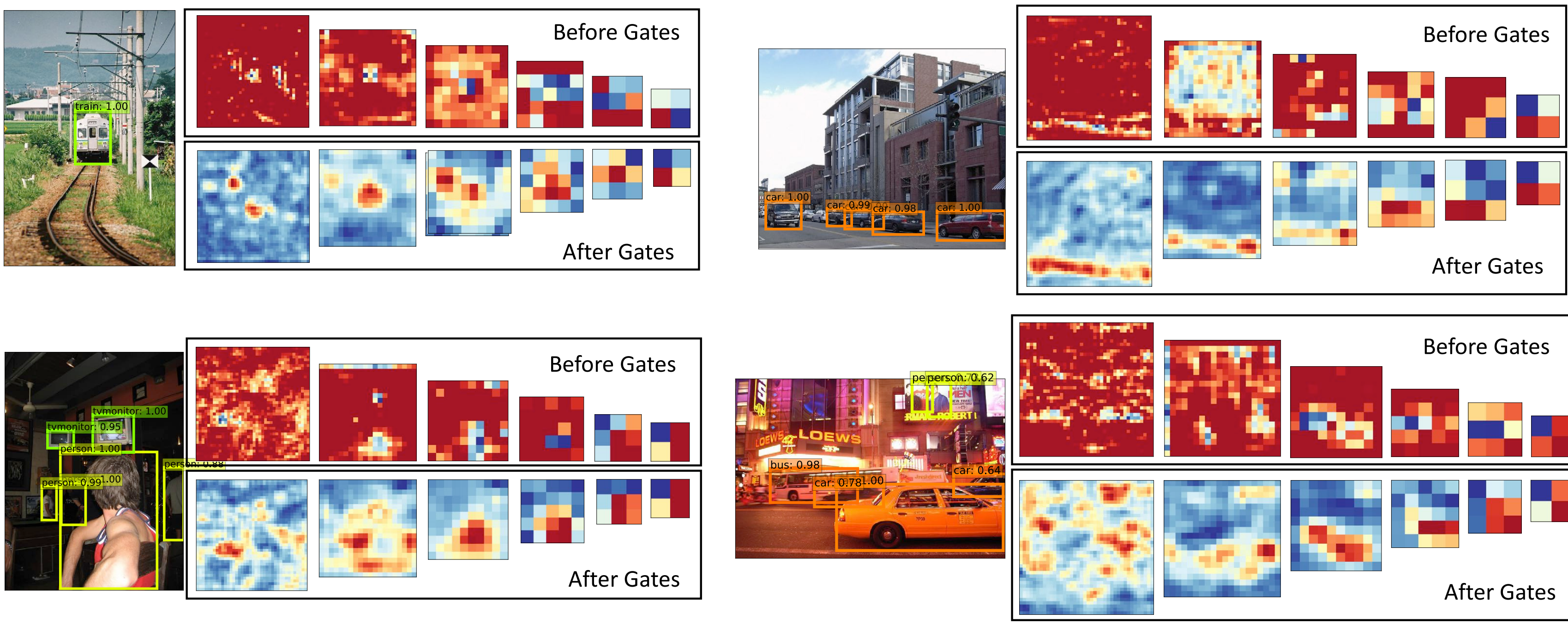
$$\mathbf{O} = \mathbf{F}_{gate}(\mathbf{U}) = \mathbf{F}_r(\mathbf{F}_g(\mathbf{F}_c(\mathbf{U})))$$

Identity Mapping

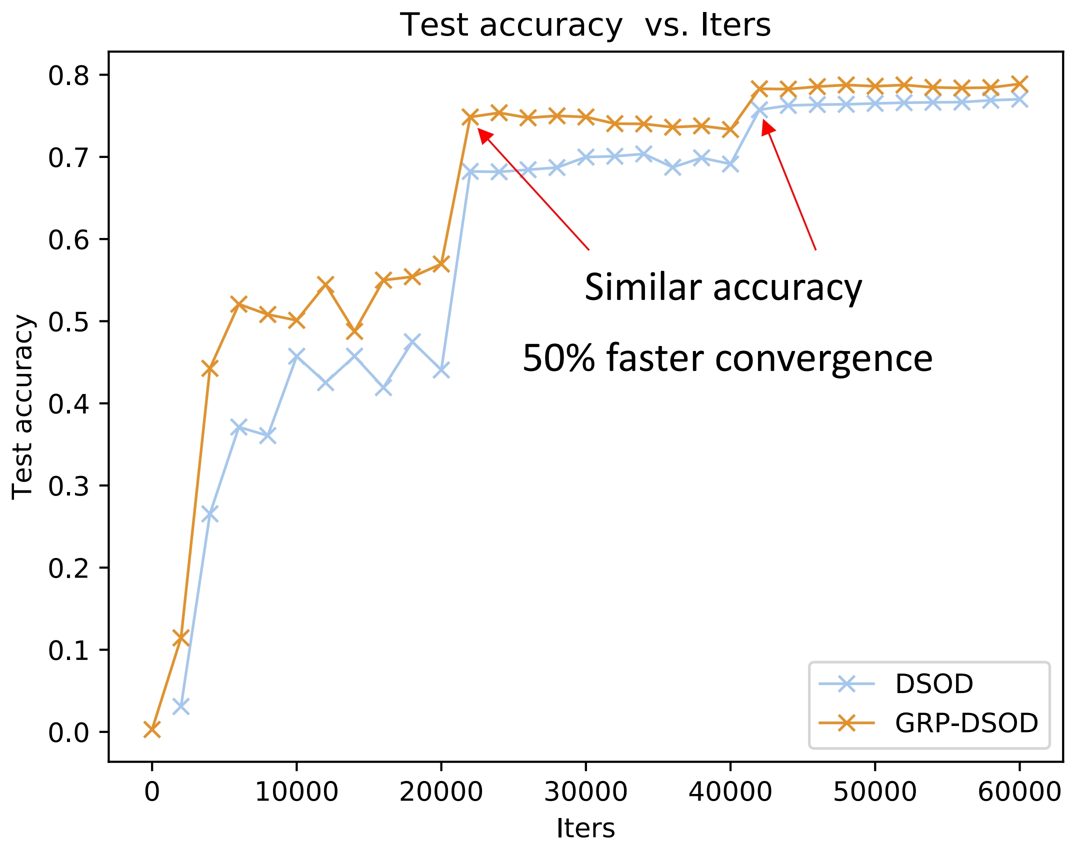
Channel-level

Global-level

Visualization of Feature Maps



High Accuracy & Faster Convergence



Ablation Study on VOC 2007

Method	mAP
DSOD300 [24]	77.7
GRP-DSOD300	78.5
GRP-DSOD320	78.7
GRP-DSOD320*	79.0
DSOD320* (using RFP only)	78.6
DSOD320* (using gates only)	78.5

Table 1: Ablation Experiments on PASCAL VOC 2007. "RFP" denotes our recurrent feature pyramid. * denotes we add one more aspect ratio 1.6 for default boxes at every prediction layer.

Results on VOC 2007

Method	data	pre-train	backbone network	prediction layer	speed (*fps*)	# parameters	input size	mAP
Faster RCNN [22]	07+12	✓	VGGNet	-	7	134.7M	$\sim 600 \times 1000$	73.2
Faster RCNN [22]	07+12	✓	ResNet-101	-	2.4*	-	$\sim 600 \times 1000$	76.4
R-FCN [3]	07+12	✓	ResNet-50	-	11	31.9M	$\sim 600 \times 1000$	77.4
R-FCN [3]	07+12	✓	ResNet-101	-	9	50.9M	$\sim 600 \times 1000$	79.5
R-FCN$_{\text{multi-sc}}$ [3]	07+12	✓	ResNet-101	-	9	50.9M	$\sim 600 \times 1000$	80.5
YOLOv2 [21]	07+12	✓	Darknet-19	-	81	-	352×352	73.7
SSD300 [19]	07+12	✓	VGGNet	Plain	46	26.3M	300×300	75.8
SSD300* [19]	07+12	✓	VGGNet	Plain	46	26.3M	300×300	77.2
SSD300S† [24]	07+12	✗	ResNet-101	Plain	12.1	52.8M	300×300	63.8*
SSD300S† [24]	07+12	✗	VGGNet	Plain	46	26.3M	300×300	69.6
SSD300S† [24]	07+12	✗	VGGNet	Dense	37	26.0M	300×300	70.4
DSOD300 [24]	07+12	✗	DS/64-192-48-1	Plain	20.6	18.2M	300×300	77.3
DSOD300 [24]	07+12	✗	DS/64-192-48-1	Dense	17.4	14.8M	300×300	77.7
GRP-DSOD300	07+12	✗	DS/64-192-48-1	Recurrent	17.5	14.1M	300×300	78.5
SSD321 [19, 6]	07+12	✓	ResNet-101	Plain	11.2	52.8M	321×321	77.1
DSSD321 [6]	07+12	✓	ResNet-101	Plain	9.5	> 52.8M	321×321	78.6
GRP-DSOD320	07+12	✗	DS/64-192-48-1	Recurrent	16.7	14.2M	320×320	78.7
GRP-DSOD320*	07+12	✗	DS/64-192-48-1	Recurrent	16.3	-	320×320	79.0

Table 2: **PASCAL VOC 2007 `test` detection results.** SSD300S† indicates training SSD300* from scratch with ResNet-101 or VGGNet. Note that the speed of Faster R-CNN with ResNet-101 (2.4 *fps*) is tested on K40, while others are tested on Titan X. For GRP-DSOD320*, we did not include the # parameters of extra default boxes and the # parameters are 14.2M. If include, the # parameters are 16M. Table adapted from [24].

Results on VOC 2007

Method	data	pre-train	backbone network	prediction layer	speed (*fps*)	# parameters	input size	mAP
Faster RCNN [22]	07+12	✓	VGGNet	-	7	134.7M	$\sim 600 \times 1000$	73.2
Faster RCNN [22]	07+12	✓	ResNet-101	-	2.4*	-	$\sim 600 \times 1000$	76.4
R-FCN [3]	07+12	✓	ResNet-50	-	11	31.9M	$\sim 600 \times 1000$	77.4
R-FCN [3]	07+12	✓	ResNet-101	-	9	50.9M	$\sim 600 \times 1000$	79.5
R-FCNmulti-sc [3]	07+12	✓	ResNet-101	-	9	50.9M	$\sim 600 \times 1000$	80.5
YOLOv2 [21]	07+12	✓	Darknet-19	-	81	-	352×352	73.7
SSD300 [19]	07+12	✓	VGGNet	Plain	46	26.3M	300×300	75.8
SSD300* [19]	07+12	✓	VGGNet	Plain	46	26.3M	300×300	77.2
SSD300S† [24]	07+12	✗	ResNet-101	Plain	12.1	52.8M	300×300	63.8*
SSD300S† [24]	07+12	✗	VGGNet	Plain	46	26.3M	300×300	69.6
SSD300S† [24]	07+12	✗	VGGNet	Dense	37	26.0M	300×300	70.4
DSOD300 [24]	07+12	✗	DS/64-192-48-1	Plain	20.6	18.2M	300×300	77.3
DSOD300 [24]	07+12	✗	DS/64-192-48-1	Dense	17.4	14.8M	300×300	77.7
GRP-DSOD300	07+12	✗	DS/64-192-48-1	Recurrent	17.5	14.1M	300×300	78.5
SSD321 [19, 6]	07+12	✓	ResNet-101	Plain	11.2	52.8M	321×321	77.1
DSSD321 [6]	07+12	✓	ResNet-101	Plain	9.5	> 52.8M	321×321	78.6
GRP-DSOD320	07+12	✗	DS/64-192-48-1	Recurrent	16.7	14.2M	320×320	78.7
GRP-DSOD320*	07+12	✗	DS/64-192-48-1	Recurrent	16.3	-	320×320	79.0

Table 2: **PASCAL VOC 2007 `test` detection results.** SSD300S† indicates training SSD300* from scratch with ResNet-101 or VGGNet. Note that the speed of Faster R-CNN with ResNet-101 (2.4 *fps*) is tested on K40, while others are tested on Titan X. For GRP-DSOD320*, we did not include the # parameters of extra default boxes and the # parameters are 14.2M. If include, the # parameters are 16M. Table adapted from [24].

Results on VOC 2012

Method	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
GRP-DSOD320*†	77.0	89.6	85.4	74.2	61.7	51.2	83.6	81.4	91.7	61.9	80.0	65.8	89.1	86.0	87.8	85.0	53.8	79.0	71.3	87.9	73.1
GRP-DSOD320*	72.5	87.1	81.9	68.6	58.3	47.0	81.5	77.3	87.7	54.9	75.5	60.7	84.5	81.3	85.1	82.2	45.1	75.4	66.6	82.5	67.0
SSD [19]	64.0	78.9	72.3	61.8	42.8	27.9	73.1	69.4	84.9	42.5	68.4	52.2	80.9	76.5	77.2	68.2	31.6	67.0	66.6	77.3	60.9
THU_ML_class	62.4	78.0	71.0	64.5	47.4	45.3	70.1	70.6	82.0	37.9	65.4	44.2	77.4	69.6	74.4	75.5	37.9	62.0	45.5	73.8	56.3
YOLOv2 [21]	48.8	69.5	61.6	37.6	28.2	18.8	63.2	53.2	65.6	27.5	44.4	35.9	61.4	57.9	66.9	63.8	16.8	52.8	39.5	65.4	46.2
DENSE_BOX	45.9	64.7	64.1	28.8	26.7	30.7	60.6	54.9	47.4	29.3	41.8	34.6	42.6	59.3	64.2	62.5	24.3	53.7	27.1	50.9	50.7
NoC	42.2	62.8	60.4	26.7	22.3	25.7	56.9	55.2	52.1	21.5	38.3	34.2	43.9	51.2	58.8	40.7	20.4	42.0	37.4	52.6	41.6

Table 3: **PASCAL VOC 2012 Competition `comp3` Leaderboard. GRP-DSOD320*†** is trained on **VOC 07++12** set and **GRP-DSOD320*** is trained on **VOC 12 trainval** set. Note that both of the two results use single model for prediction without any experimental tricks. Result links are **GRP-DSOD320*† (07++12)**: http://host.robots.ox.ac.uk:8080/anonymous/CSMRU4.html; **GRP-DSOD320* (12)**: http://host.robots.ox.ac.uk:8080/anonymous/KJSBBP.html.

Results on VOC 2012

Method	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
GRP-DSOD320*†	77.0	89.6	85.4	74.2	61.7	51.2	83.6	81.4	91.7	61.9	80.0	65.8	89.1	86.0	87.8	85.0	53.8	79.0	71.3	87.9	73.1
GRP-DSOD320*	72.5	87.1	81.9	68.6	58.3	47.0	81.5	77.3	87.7	54.9	75.5	60.7	84.5	81.3	85.1	82.2	45.1	75.4	66.6	82.5	67.0
SSD [19]	64.0	78.9	72.3	61.8	42.8	27.9	73.1	69.4	84.9	42.5	68.4	52.2	80.9	76.5	77.2	68.2	31.6	67.0	66.6	77.3	60.9
THU_ML_class	62.4	78.0	71.0	64.5	47.4	45.3	70.1	70.6	82.0	37.9	65.4	44.2	77.4	69.6	74.4	75.5	37.9	62.0	45.5	73.8	56.3
YOLOv2 [21]	48.8	69.5	61.6	37.6	28.2	18.8	63.2	53.2	65.6	27.5	44.4	35.9	61.4	57.9	66.9	63.8	16.8	52.8	39.5	65.4	46.2
DENSE_BOX	45.9	64.7	64.1	28.8	26.7	30.7	60.6	54.9	47.4	29.3	41.8	34.6	42.6	59.3	64.2	62.5	24.3	53.7	27.1	50.9	50.7
NoC	42.2	62.8	60.4	26.7	22.3	25.7	56.9	55.2	52.1	21.5	38.3	34.2	43.9	51.2	58.8	40.7	20.4	42.0	37.4	52.6	41.6

Table 3: **PASCAL VOC 2012 Competition `comp3` Leaderboard. GRP-DSOD320*†** is trained on **VOC 07++12** set and **GRP-DSOD320*** is trained on **VOC 12 trainval** set. Note that both of the two results use single model for prediction without any experimental tricks. Result links are **GRP-DSOD320*† (07++12)**: `http://host.robots.ox.ac.uk:8080/anonymous/CSMRU4.html`; **GRP-DSOD320* (12)**: `http://host.robots.ox.ac.uk:8080/anonymous/KJSBBP.html`.

Results on VOC 2012

Method	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
GRP-DSOD320*†	77.0	89.6	85.4	74.2	61.7	51.2	83.6	81.4	91.7	61.9	80.0	65.8	89.1	86.0	87.8	85.0	53.8	79.0	71.3	87.9	73.1
GRP-DSOD320*	72.5	87.1	81.9	68.6	58.3	47.0	81.5	77.3	87.7	54.9	75.5	60.7	84.5	81.3	85.1	82.2	45.1	75.4	66.6	82.5	67.0
SSD [19]	64.0	78.9	72.3	61.8	42.8	27.9	73.1	69.4	84.9	42.5	68.4	52.2	80.9	76.5	77.2	68.2	31.6	67.0	66.6	77.3	60.9
THU_ML_class	62.4	78.0	71.0	64.5	47.4	45.3	70.1	70.6	82.0	37.9	65.4	44.2	77.4	69.6	74.4	75.5	37.9	62.0	45.5	73.8	56.3
YOLOv2 [21]	48.8	69.5	61.6	37.6	28.2	18.8	63.2	53.2	65.6	27.5	44.4	35.9	61.4	57.9	66.9	63.8	16.8	52.8	39.5	65.4	46.2
DENSE_BOX	45.9	64.7	64.1	28.8	26.7	30.7	60.6	54.9	47.4	29.3	41.8	34.6	42.6	59.3	64.2	62.5	24.3	53.7	27.1	50.9	50.7
NoC	42.2	62.8	60.4	26.7	22.3	25.7	56.9	55.2	52.1	21.5	38.3	34.2	43.9	51.2	58.8	40.7	20.4	42.0	37.4	52.6	41.6

Table 3: **PASCAL VOC 2012 Competition `comp3` Leaderboard. GRP-DSOD320*†** is trained on **VOC 07++12** set and **GRP-DSOD320*** is trained on **VOC 12 trainval** set. Note that both of the two results use single model for prediction without any experimental tricks. Result links are **GRP-DSOD320*† (07++12)**: http://host.robots.ox.ac.uk:8080/anonymous/CSMRU4.html; **GRP-DSOD320* (12)**: http://host.robots.ox.ac.uk:8080/anonymous/KJSBBP.html.

Results on VOC 2012

Method	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
GRP-DSOD320*†	77.0	89.6	85.4	74.2	61.7	51.2	83.6	81.4	91.7	61.9	80.0	65.8	89.1	86.0	87.8	85.0	53.8	79.0	71.3	87.9	73.1
GRP-DSOD320*	72.5	87.1	81.9	68.6	58.3	47.0	81.5	77.3	87.7	54.9	75.5	60.7	84.5	81.3	85.1	82.2	45.1	75.4	66.6	82.5	67.0
SSD [19]	64.0	78.9	72.3	61.8	42.8	27.9	73.1	69.4	84.9	42.5	68.4	52.2	80.9	76.5	77.2	68.2	31.6	67.0	66.6	77.3	60.9
THU_ML_class	62.4	78.0	71.0	64.5	47.4	45.3	70.1	70.6	82.0	37.9	65.4	44.2	77.4	69.6	74.4	75.5	37.9	62.0	45.5	73.8	56.3
YOLOv2 [21]	48.8	69.5	61.6	37.6	28.2	18.8	63.2	53.2	65.6	27.5	44.4	35.9	61.4	57.9	66.9	63.8	16.8	52.8	39.5	65.4	46.2
DENSE_BOX	45.9	64.7	64.1	28.8	26.7	30.7	60.6	54.9	47.4	29.3	41.8	34.6	42.6	59.3	64.2	62.5	24.3	53.7	27.1	50.9	50.7
NoC	42.2	62.8	60.4	26.7	22.3	25.7	56.9	55.2	52.1	21.5	38.3	34.2	43.9	51.2	58.8	40.7	20.4	42.0	37.4	52.6	41.6

Method	data	backbone network	pre-train	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
ION [1]	07+12+S	VGGNet	✓	76.4	87.5	84.7	76.8	63.8	58.3	82.6	79.0	90.9	57.8	82.0	64.7	88.9	86.5	84.7	82.3	51.4	78.2	69.2	85.2	73.5
Faster RCNN [22]	07++12	ResNet-101	✓	73.8	86.5	81.6	77.2	58.0	51.0	78.6	76.6	93.2	48.6	80.4	59.0	92.1	85.3	84.8	80.7	48.1	77.3	66.5	84.7	65.6
R-FCNmulti-sc [3]	07++12	ResNet-101	✓	77.6	86.9	83.4	81.5	63.8	62.4	81.6	81.1	93.1	58.0	83.8	60.8	92.7	86.0	84.6	84.4	59.0	80.8	68.6	86.1	72.9
YOLOv2 [21]	07++12	Darknet-19	✓	73.4	86.3	82.0	74.8	59.2	51.8	79.8	76.5	90.6	52.1	78.2	58.5	89.3	82.5	83.4	81.3	49.1	77.2	62.4	83.8	68.7
SSD300* [19]	07++12	VGGNet	✓	75.8	88.1	82.9	74.4	61.9	47.6	82.7	78.8	91.5	58.1	80.0	64.1	89.4	85.7	85.5	82.6	50.2	79.8	73.6	86.6	72.1
DSOD300 [24]	07++12	DS/64-192-48-1	✗	76.3	89.4	85.3	72.9	62.7	49.5	83.6	80.6	92.1	60.8	77.9	65.6	88.9	85.5	86.8	84.6	51.1	77.7	72.3	86.0	72.2
SSD321 [19, 6]	07++12	ResNet-101	✓	75.4	87.9	82.9	73.7	61.5	45.3	81.4	75.6	92.6	57.4	78.3	65.0	90.8	86.8	85.8	81.5	50.3	78.1	75.3	85.2	72.5
DSSD321 [6]	07++12	ResNet-101	✓	76.3	87.3	83.3	75.4	64.6	46.8	82.7	76.5	92.9	59.5	78.3	64.3	91.5	86.6	86.6	82.1	53.3	79.6	75.7	85.2	73.9
GRP-DSOD320*	07++12	DS/64-192-48-1	✗	77.0	89.6	85.4	74.2	61.7	51.2	83.6	81.4	91.7	61.9	80.0	65.8	89.1	86.0	87.8	85.0	53.8	79.0	71.3	87.9	73.1

Table 4: **PASCAL VOC 2012 `test` detection results. 07+12**: 07 `trainval` + 12 `trainval`, **07+12+S**: 07+12 plus segmentation labels, **07++12**: 07 `trainval` + 07 `test` + 12 `trainval`. The result link for DSOD320* (07++12) is: `http://host.robots.ox.ac.uk:8080/anonymous/CSMRU4.html`.

Results on VOC 2012

Method	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
GRP-DSOD320*†	77.0	89.6	85.4	74.2	61.7	51.2	83.6	81.4	91.7	61.9	80.0	65.8	89.1	86.0	87.8	85.0	53.8	79.0	71.3	87.9	73.1
GRP-DSOD320*	72.5	87.1	81.9	68.6	58.3	47.0	81.5	77.3	87.7	54.9	75.5	60.7	84.5	81.3	85.1	82.2	45.1	75.4	66.6	82.5	67.0
SSD [19]	64.0	78.9	72.3	61.8	42.8	27.9	73.1	69.4	84.9	42.5	68.4	52.2	80.9	76.5	77.2	68.2	31.6	67.0	66.6	77.3	60.9
THU_ML_class	62.4	78.0	71.0	64.5	47.4	45.3	70.1	70.6	82.0	37.9	65.4	44.2	77.4	69.6	74.4	75.5	37.9	62.0	45.5	73.8	56.3
YOLOv2 [21]	48.8	69.5	61.6	37.6	28.2	18.8	63.2	53.2	65.6	27.5	44.4	35.9	61.4	57.9	66.9	63.8	16.8	52.8	39.5	65.4	46.2
DENSE_BOX	45.9	64.7	64.1	28.8	26.7	30.7	60.6	54.9	47.4	29.3	41.8	34.6	42.6	59.3	64.2	62.5	24.3	53.7	27.1	50.9	50.7
NoC	42.2	62.8	60.4	26.7	22.3	25.7	56.9	55.2	52.1	21.5	38.3	34.2	43.9	51.2	58.8	40.7	20.4	42.0	37.4	52.6	41.6

Method	data	backbone network	pre-train	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
ION [1]	07+12+S	VGGNet	✓	76.4	87.5	84.7	76.8	63.8	58.3	82.6	79.0	90.9	57.8	82.0	64.7	88.9	86.5	84.7	82.3	51.4	78.2	69.2	85.2	73.5
Faster RCNN [22]	07++12	ResNet-101	✓	73.8	86.5	81.6	77.2	58.0	51.0	78.6	76.6	93.2	48.6	80.4	59.0	92.1	85.3	84.8	80.7	48.1	77.3	66.5	84.7	65.6
R-FCNmulti-sc [3]	07++12	ResNet-101	✓	77.6	86.9	83.4	81.5	63.8	62.4	81.6	81.1	93.1	58.0	83.8	60.8	92.7	86.0	84.6	84.4	59.0	80.8	68.6	86.1	72.9
YOLOv2 [21]	07++12	Darknet-19	✓	73.4	86.3	82.0	74.8	59.2	51.8	79.8	76.5	90.6	52.1	78.2	58.5	89.3	82.5	83.4	81.3	49.1	77.2	62.4	83.8	68.7
SSD300* [19]	07++12	VGGNet	✓	75.8	88.1	82.9	74.4	61.9	47.6	82.7	78.8	91.5	58.1	80.0	64.1	89.4	85.7	85.5	82.6	50.2	79.8	73.6	86.6	72.1
DSOD300 [24]	07++12	DS/64-192-48-1	✗	76.3	89.4	85.3	72.9	62.7	49.5	83.6	80.6	92.1	60.8	77.9	65.6	88.9	85.5	86.8	84.6	51.1	77.7	72.3	86.0	72.2
SSD321 [19, 6]	07++12	ResNet-101	✓	75.4	87.9	82.9	73.7	61.5	45.3	81.4	75.6	92.6	57.4	78.3	65.0	90.8	86.8	85.8	81.5	50.3	78.1	75.3	85.2	72.5
DSSD321 [6]	07++12	ResNet-101	✓	76.3	87.3	83.3	75.4	64.6	46.8	82.7	76.5	92.9	59.5	78.3	64.3	91.5	86.6	86.6	82.1	53.3	79.6	75.7	85.2	73.9
GRP-DSOD320*	07++12	DS/64-192-48-1	✗	77.0	89.6	85.4	74.2	61.7	51.2	83.6	81.4	91.7	61.9	80.0	65.8	89.1	86.0	87.8	85.0	53.8	79.0	71.3	87.9	73.1

Table 4: **PASCAL VOC 2012 test detection results. 07+12**: 07 trainval + 12 trainval, **07+12+S**: 07+12 plus segmentation labels, **07++12**: 07 trainval + 07 test + 12 trainval. The result link for DSOD320* (07++12) is: http://host.robots.ox.ac.uk:8080/anonymous/CSMRU4.html.

Results on MS COCO

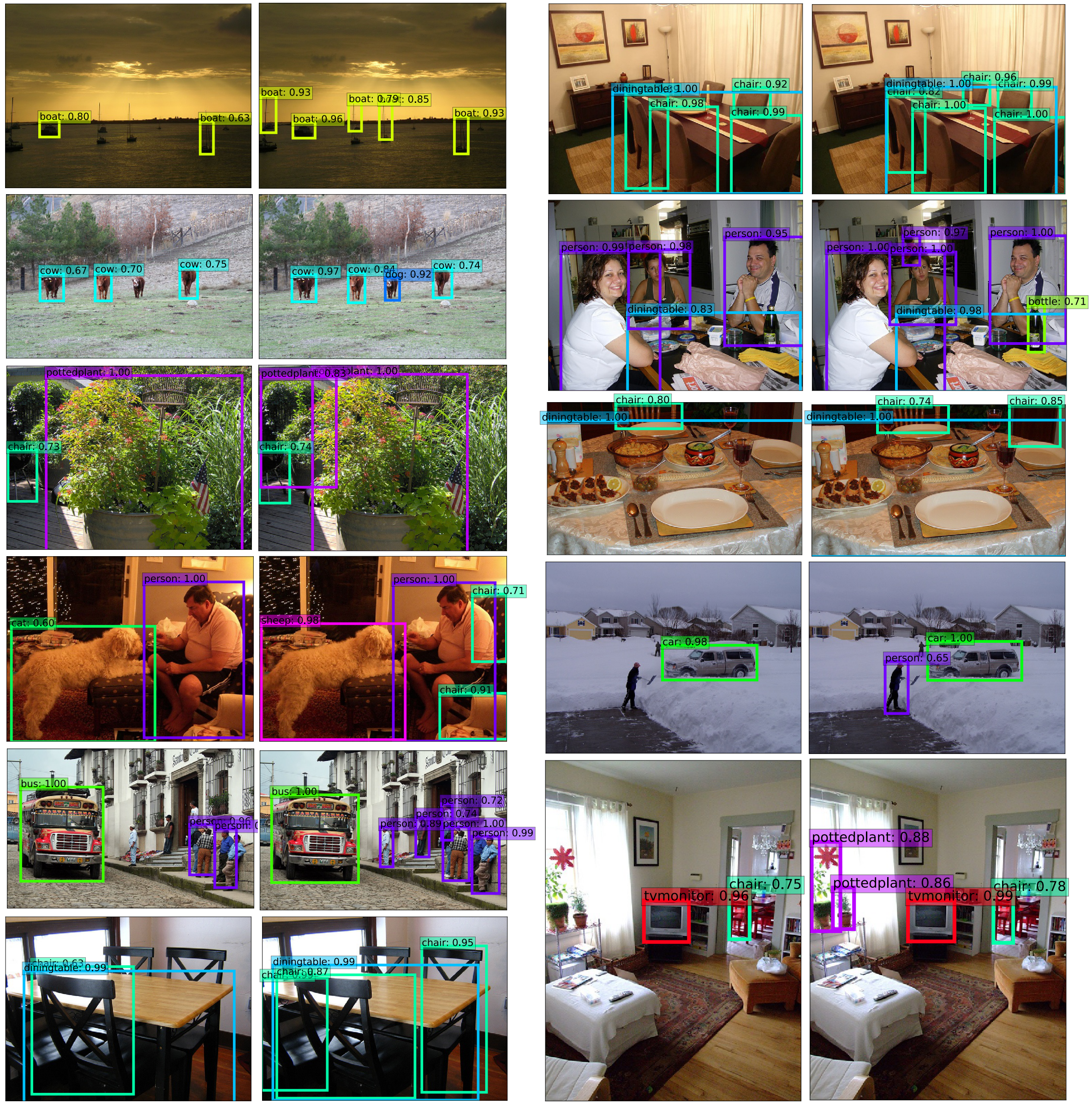
Method	data	backbone network	pre-train	Avg. Precision, IoU:			Avg. Precision, Area:			Avg. Recall, #Dets:			Avg. Recall, Area:		
				0.5:0.95	0.5	0.75	S	M	L	1	10	100	S	M	L
Faster RCNN [22]	trainval	VGGNet	✓	21.9	42.7	-	-	-	-	-	-	-	-	-	-
ION [1]	train	VGGNet	✓	23.6	43.2	23.6	6.4	24.1	38.3	23.2	32.7	33.5	10.1	37.7	53.6
R-FCN [3]	trainval	ResNet-101	✓	29.2	51.5	-	10.3	32.4	43.3	-	-	-	-	-	-
R-FCN$_{\text{multi-sc}}$ [3]	trainval	ResNet-101	✓	29.9	**51.9**	-	10.8	32.8	45.0	-	-	-	-	-	-
SSD300 (Huang et al.) [14]	< trainval35k	MobileNet	✓	18.8	-	-	-	-	-	-	-	-	-	-	-
SSD300 (Huang et al.) [14]	< trainval35k	Inception-v2	✓	21.6	-	-	-	-	-	-	-	-	-	-	-
YOLOv2 [21]	trainval35k	Darknet-19	✓	21.6	44.0	19.2	5.0	22.4	35.5	20.7	31.6	33.3	9.8	36.5	54.4
SSD300* [19]	trainval35k	VGGNet	✓	25.1	43.1	25.8	6.6	25.9	41.4	23.7	35.1	37.2	11.2	40.4	58.4
DSOD300 [24]	trainval	DS/64-192-48-1	✗	29.3	47.3	30.6	9.4	31.5	47.0	27.3	40.7	43.0	16.7	47.1	**65.0**
SSD321 [19, 6]	trainval35k	ResNet-101	✓	28.0	45.4	29.3	6.2	28.3	**49.3**	25.9	37.8	39.9	11.5	43.3	64.9
DSSD321 [6]	trainval35k	ResNet-101	✓	28.0	46.1	29.2	7.4	28.1	47.6	25.5	37.1	39.4	12.7	42.0	62.6
GRP-DSOD320	trainval	DS/64-192-48-1	✗	**30.0**	47.9	**31.8**	**10.9**	**33.6**	46.3	**28.0**	**42.1**	**44.5**	**18.8**	**49.1**	**65.0**

Table 5: **MS COCO `test-dev 2015` detection results.**

Results on MS COCO

Method	data	backbone network	pre-train	Avg. Precision, IoU:			Avg. Precision, Area:			Avg. Recall, #Dets:			Avg. Recall, Area:		
				0.5:0.95	0.5	0.75	S	M	L	1	10	100	S	M	L
Faster RCNN [22]	trainval	VGGNet	✓	21.9	42.7	-	-	-	-	-	-	-	-	-	-
ION [1]	train	VGGNet	✓	23.6	43.2	23.6	6.4	24.1	38.3	23.2	32.7	33.5	10.1	37.7	53.6
R-FCN [3]	trainval	ResNet-101	✓	29.2	51.5	-	10.3	32.4	43.3	-	-	-	-	-	-
R-FCN$_{multi-sc}$ [3]	trainval	ResNet-101	✓	29.9	**51.9**	-	10.8	32.8	45.0	-	-	-	-	-	-
SSD300 (Huang et al.) [14]	< trainval35k	MobileNet	✓	18.8	-	-	-	-	-	-	-	-	-	-	-
SSD300 (Huang et al.) [14]	< trainval35k	Inception-v2	✓	21.6	-	-	-	-	-	-	-	-	-	-	-
YOLOv2 [21]	trainval35k	Darknet-19	✓	21.6	44.0	19.2	5.0	22.4	35.5	20.7	31.6	33.3	9.8	36.5	54.4
SSD300* [19]	trainval35k	VGGNet	✓	25.1	43.1	25.8	6.6	25.9	41.4	23.7	35.1	37.2	11.2	40.4	58.4
DSOD300 [24]	trainval	DS/64-192-48-1	✗	29.3	47.3	30.6	9.4	31.5	47.0	27.3	40.7	43.0	16.7	47.1	**65.0**
SSD321 [19, 6]	trainval35k	ResNet-101	✓	28.0	45.4	29.3	6.2	28.3	**49.3**	25.9	37.8	39.9	11.5	43.3	64.9
DSSD321 [6]	trainval35k	ResNet-101	✓	28.0	46.1	29.2	7.4	28.1	47.6	25.5	37.1	39.4	12.7	42.0	62.6
GRP-DSOD320	trainval	DS/64-192-48-1	✗	**30.0**	47.9	**31.8**	**10.9**	**33.6**	46.3	**28.0**	**42.1**	**44.5**	**18.8**	**49.1**	**65.0**

Table 5: **MS COCO `test-dev 2015` detection results.**



DSOD GRP-DSOD DSOD GRP-DSOD

Summary of GRP-DSOD

- Best performance on PASCAL VOC comp3 challenge.
- Recurrent feature pyramids for enhancing the feature representation.
- Recalibrating feature activations with gating mechanism.
- *Gated Recurrent Feature Pyramid* is an independent module that can be applied to DSOD, FPN, etc.

Thanks & Questions