

MSCOCO & PLACES Challenges 2017

Megvii (Face++) Team

Results

Track	Rank	Ensemble	Single
COCO (<i>Detection</i>)	1 st	52.8	50.5
PLACES (<i>InstanceSeg</i>)	1 st	30.7	28.7
COCO (<i>Keypoint</i>)	1 st	72.6	70.9
COCO (<i>InstanceSeg</i>)	2 nd	46.4	45.0

I. COCO'17 Detection



Chao PENG*



Tete XIAO*



Zeming LI*



Yuning JIANG



Xiangyu ZHANG



Kai JIA



Gang YU



Jian SUN

Batchsize in Training

Classification (ImageNet)

[A. Krizhevsky, NIPS'12]

[K. Simonyan, ICLR'15]

[K. He, CVPR'16]

[P. Goyal, Arxiv'17]

[Y. You, Arxiv'17]

Detection (MSCOCO)

[S. Ren, NIPS'15]

[S. Ren, TPAMI'16]

[J. Dai, NIPS'16]

[T. Lin, CVPR'17]

[K. He, ICCV'17]

Batchsize in Training

	Classification (ImageNet)	Detection (MSCOCO)
128	[A. Krizhevsky, NIPS'12]	[S. Ren, NIPS'15]
256	[K. Simonyan, ICLR'15]	[S. Ren, TPAMI'16]
	[K. He, CVPR'16]	[J. Dai, NIPS'16]
8K	[P. Goyal, Arxiv'17]	[T. Lin, CVPR'17]
32K	[Y. You, Arxiv'17]	[K. He, ICCV'17]

Batchsize in Training

	Classification (ImageNet)	Detection (MSCOCO)	
128	[A. Krizhevsky, NIPS'12]	[S. Ren, NIPS'15]	2
256	[K. Simonyan, ICLR'15]	[S. Ren, TPAMI'16]	8
	[K. He, CVPR'16]	[J. Dai, NIPS'16]	
8K	[P. Goyal, Arxiv'17]	[T. Lin, CVPR'17]	16
32K	[Y. You, Arxiv'17]	[K. He, ICCV'17]	

Batchsize in Training

	Classification (ImageNet)	Detection (MSCOCO)	
128	[A. Krizhevsky, NIPS'12]	[S. Ren, NIPS'15]	2
256	[K. Simonyan, ICLR'15]	[S. Ren, TPAMI'16]	8
	[K. He, CVPR'16]	[J. Dai, NIPS'16]	
8K	[P. Goyal, Arxiv'17]	[T. Lin, CVPR'17]	
32K	[Y. You, Arxiv'17]	[K. He, ICCV'17]	16

Why batchsize is so **small** in detection?

Batchsize Constraints



↑ ↓

A green double-headed vertical arrow indicating the height of the small image.

224 v.s. 800



Detection eats much more GPU memory!

Small Batchsize Leads to

- Unstable gradient



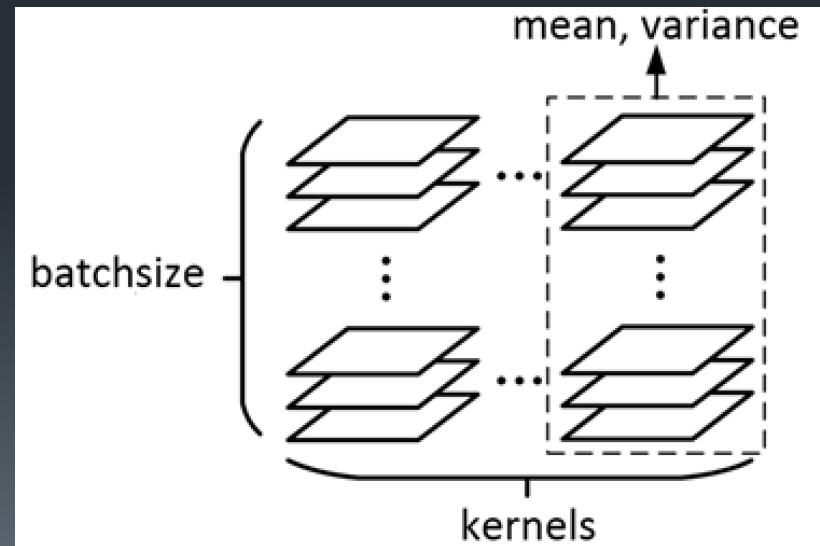
Small Batchsize Leads to

- Unstable gradient



Small Batchsize Leads to

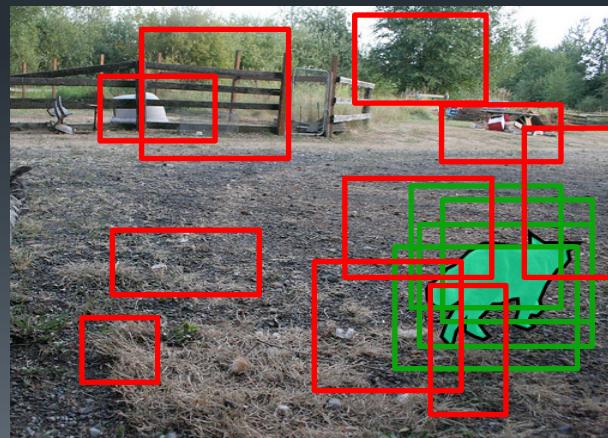
- Unstable gradient
- Inaccurate BN statistics



Small Batchsize Leads to

- Unstable gradient
- Inaccurate BN statistics
- Extremely imbalanced data

Non-objects >> Objects



Small Batchsize Leads to

- Unstable gradient
- Inaccurate BN statistics
- Extremely imbalanced data
- Very long training period
-



MegDet -

The First Large-batch Detector

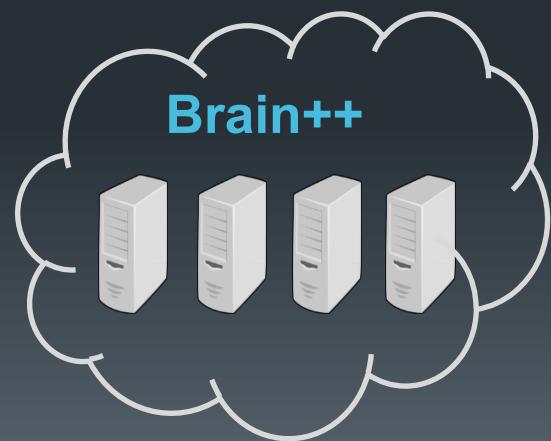
MegDet -

The First Large-batch Detector

- MegBrain & Brain++ empowered

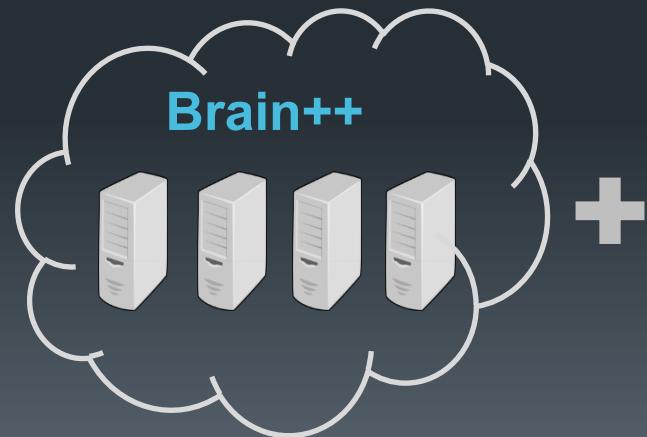
MegBrain & Brain++

Platform

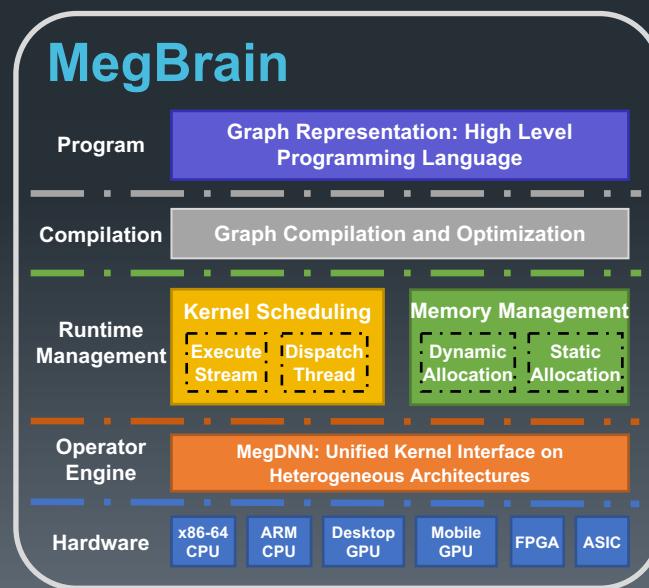


MegBrain & Brain++

Platform

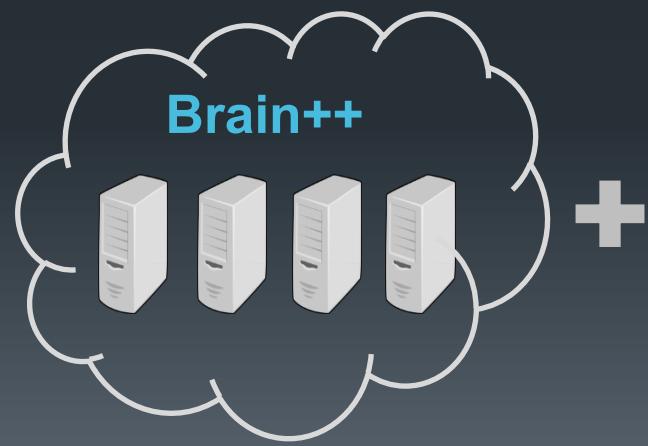


Framework

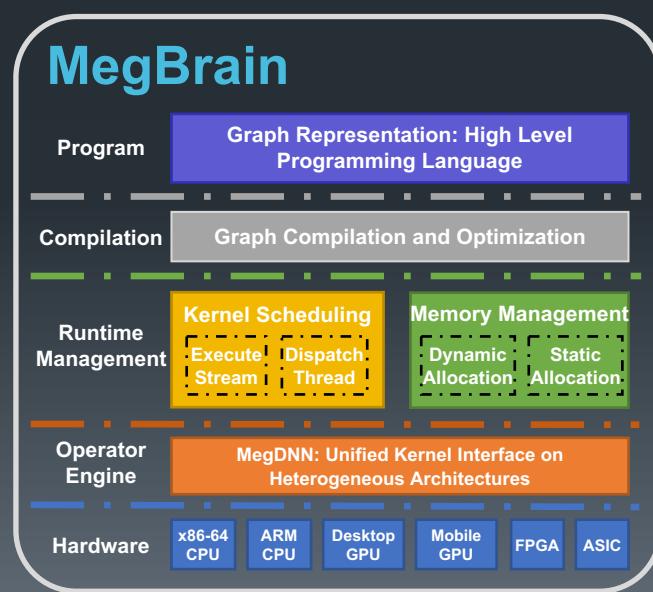


MegBrain & Brain++

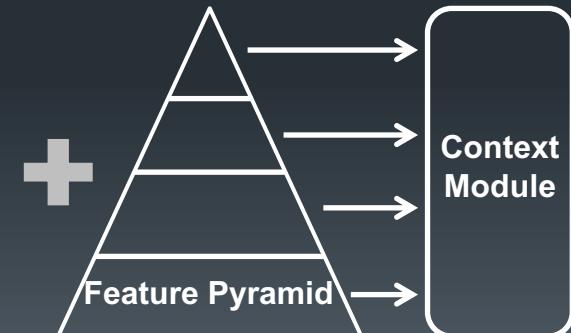
Platform



Framework



Algorithm



MegDet -

The First Large-batch Detector

- MegBrain & Brain++ empowered
- Multi-device batch normalization

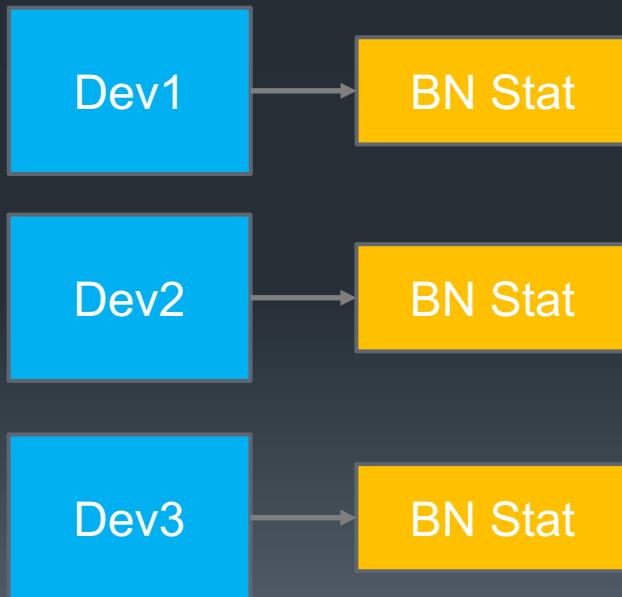
Multi-device BatchNorm

Dev1

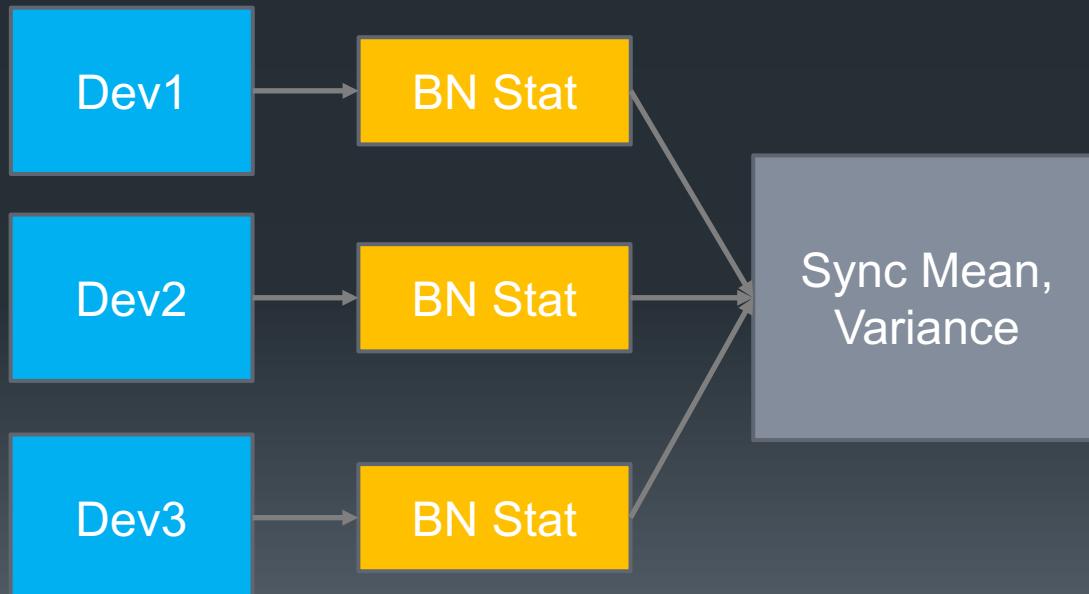
Dev2

Dev3

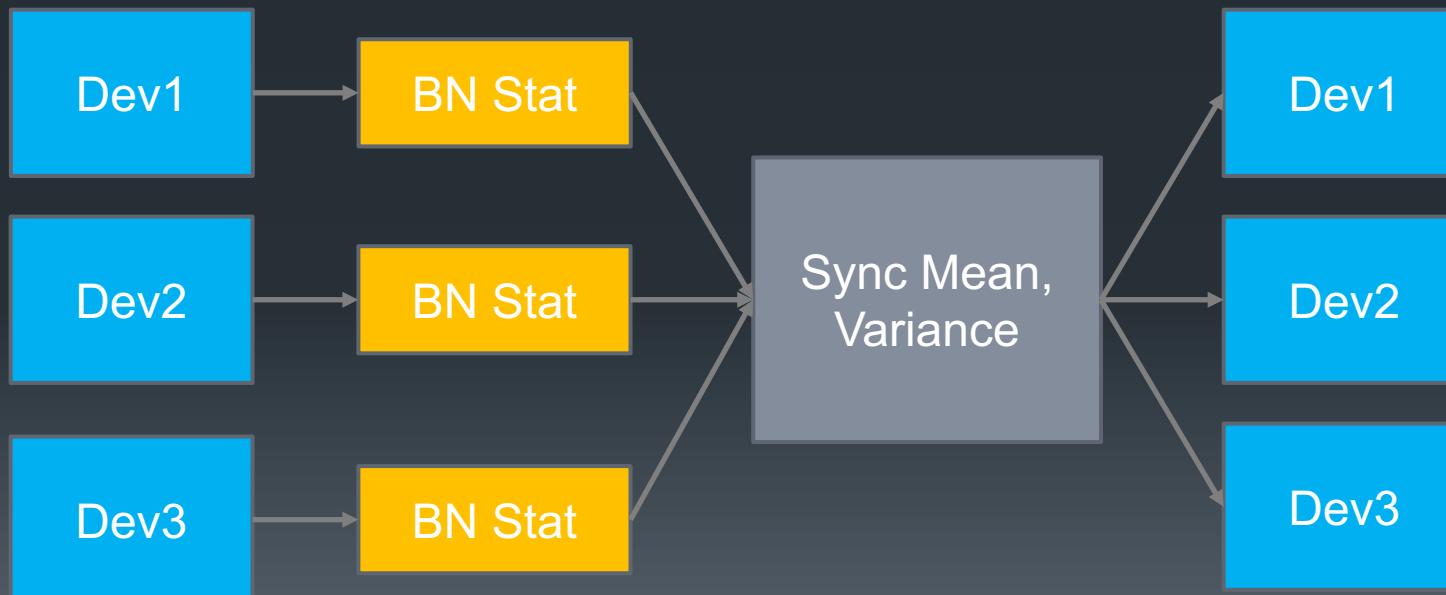
Multi-device BatchNorm



Multi-device BatchNorm



Multi-device BatchNorm

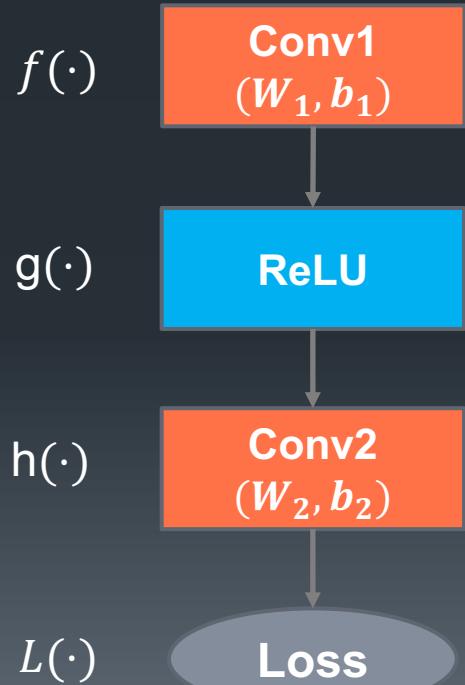


MegDet -

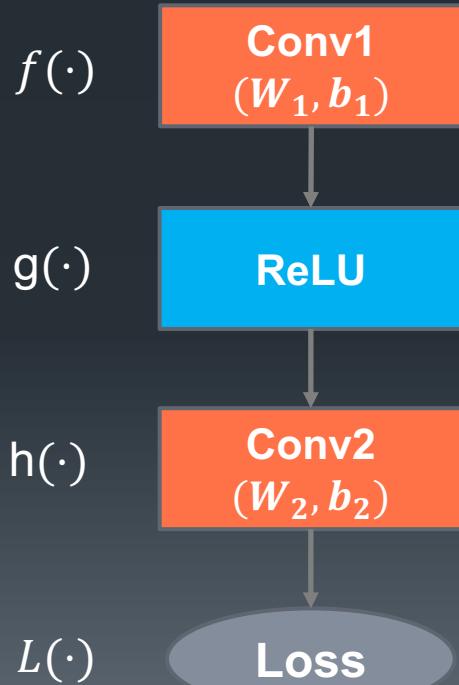
The First Large-batch Detector

- MegBrain & Brain++ empowered
- Multi-device batch normalization
- Sublinear memory

Sublinear Memory

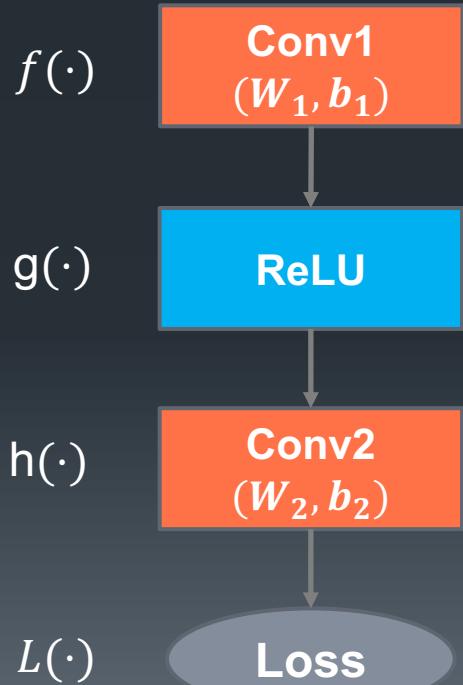


Sublinear Memory



$$\frac{\partial L}{\partial W_2} = \frac{\partial L}{\partial h} \frac{\partial h}{\partial W_2}$$

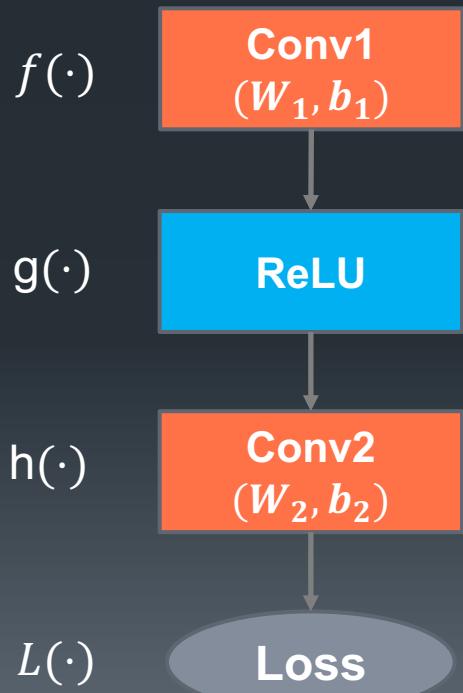
Sublinear Memory



$$\frac{\partial L}{\partial W_2} = \frac{\partial L}{\partial h} \frac{\partial h}{\partial W_2}$$

We need store results of **Conv2!**

Sublinear Memory



$$h(\cdot) = g(f(x)) \quad \frac{\partial L}{\partial W_2} = \frac{\partial L}{\partial h} \frac{\partial h}{\partial W_2}$$

$$\frac{\partial L}{\partial W_2} = \frac{\partial L}{\partial g(f(x))} \frac{\partial g(f(x))}{\partial W_2}$$

We can compute the $h(\cdot)$ in BP and thus don't need to store the **Conv2!**

[T. Chen et al, Arxiv'16]

MegDet -

The First Large-batch Detector

- MegBrain & Brain++ empowered
- Multi-device batch normalization
- Sublinear memory
- Large learning rate policy

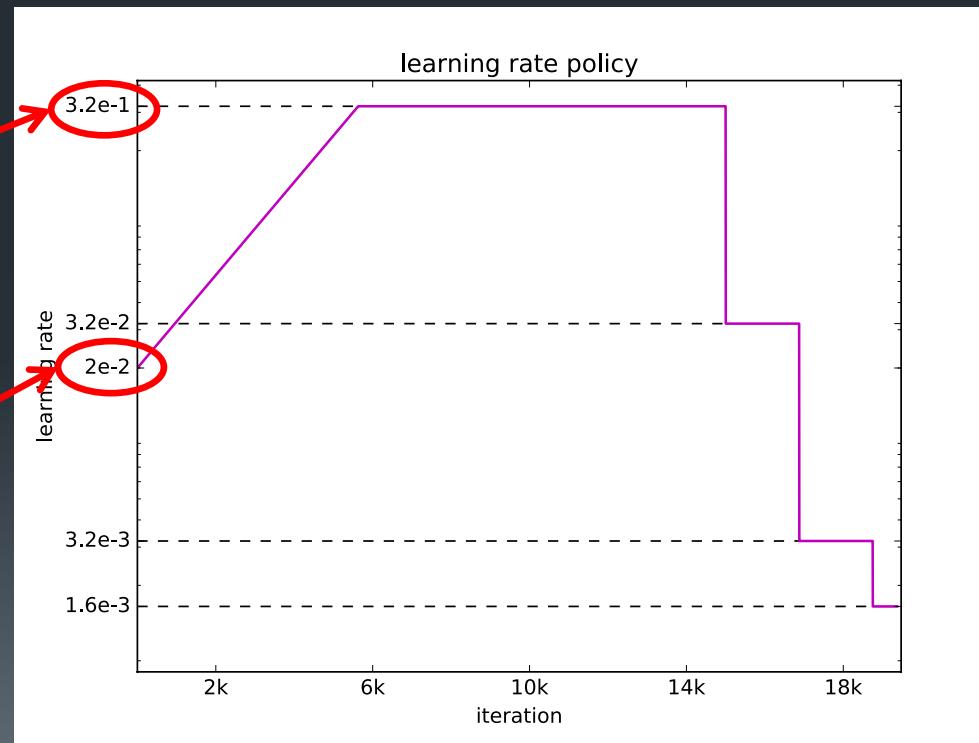
Large Learning Rate

in MegDet: **0.32**

||

x 16

in FPN: **0.02**



Large Learning Rate

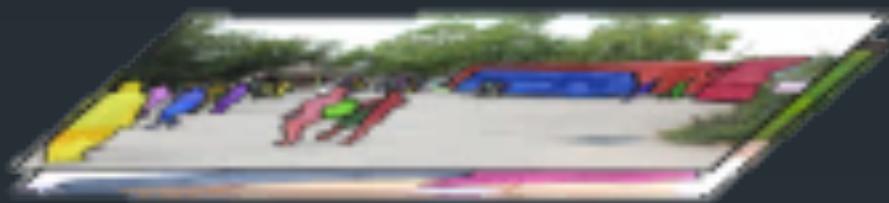
in MegDet: **0.32**

||

x 16

in FPN: **0.02**





Faster RCNN, 2015
batchsize = 2

[S. Ren et al, Faster R-CNN, NIPS'15]



Faster RCNN, 2015
batchsize = 2



FPN, 2016
batchsize = 16



Faster RCNN, 2015
batchsize = 2



FPN, 2016
batchsize = 16



MegDet, 2017
batchsize = 256

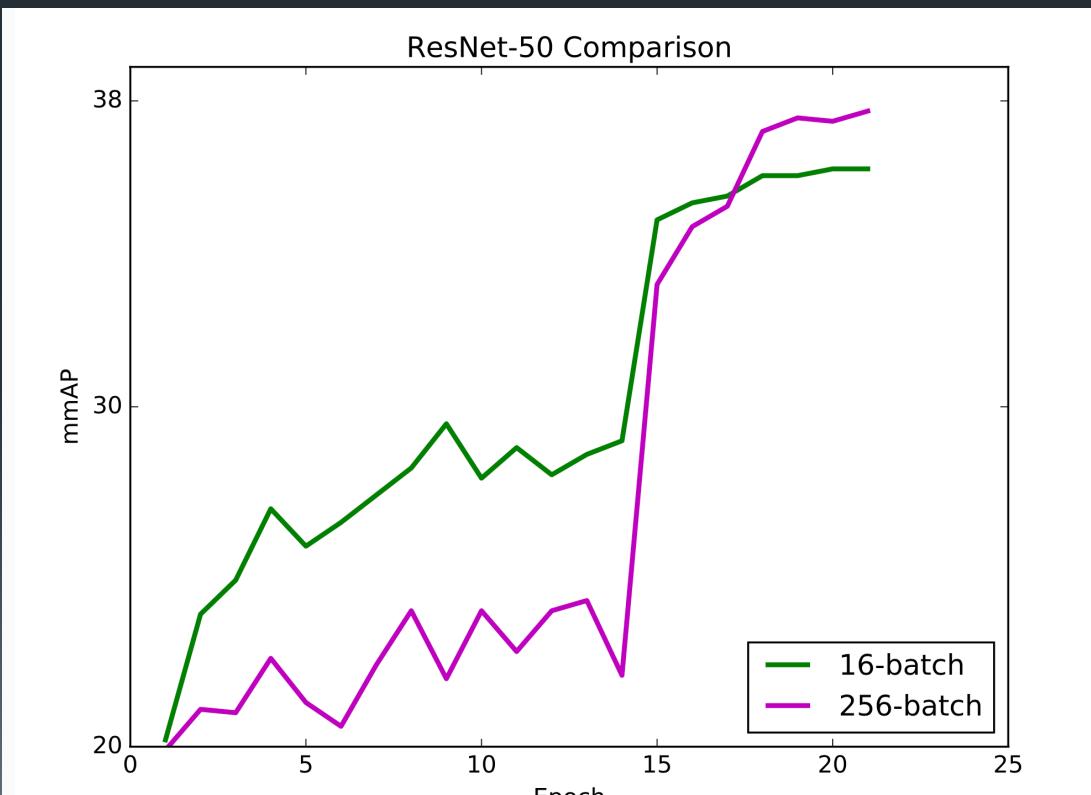
With Large Batch, We Have

- More stable gradient

With Large Batch, We Have

- More stable gradient
- More accurate BN statistics

Large Batch v.s. Small Batch



With Large Batch, We Have

- More stable gradient
- More accurate BN statistics
- Faster training (train COCO in hours)

Speed-up Training



Train detector on COCO in **Four** hour!

Face++ 旷视

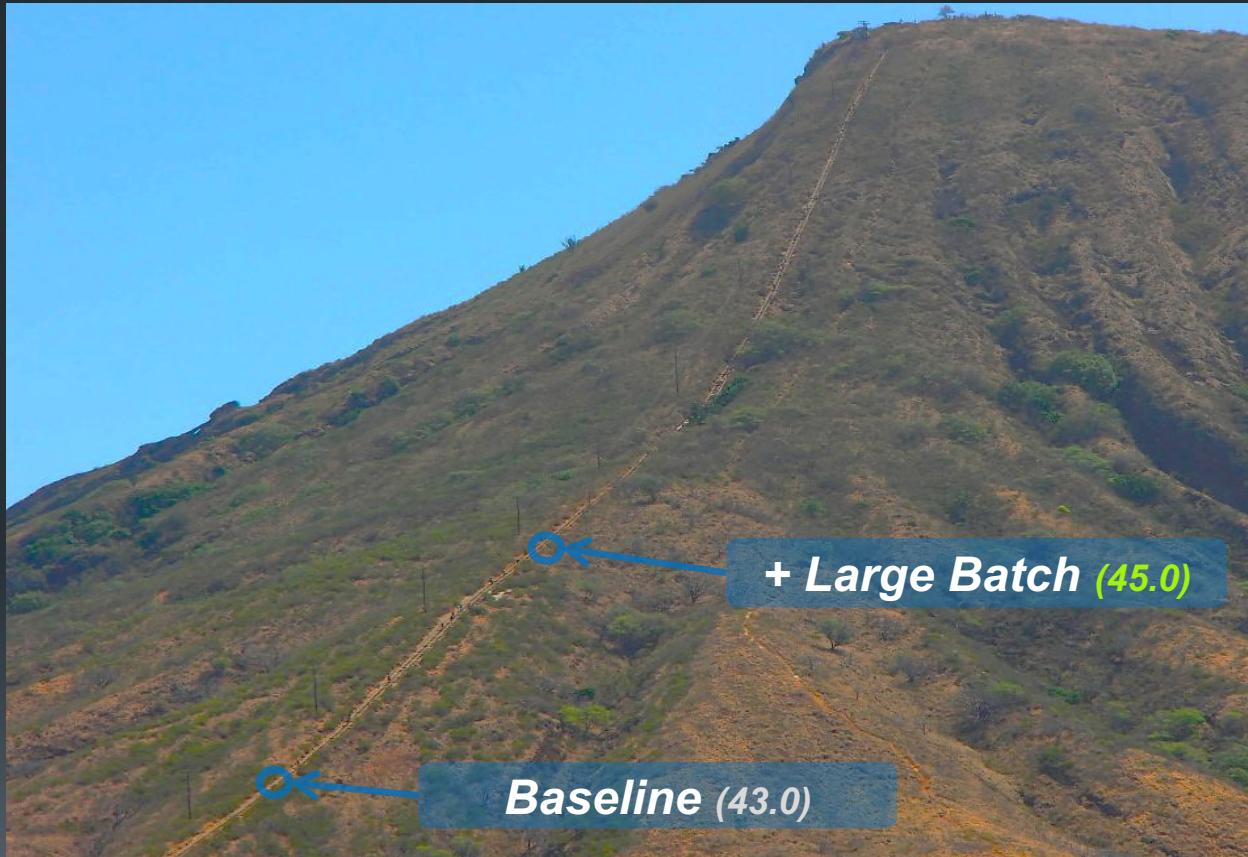


“KOKO Head” in Honolulu, Hawaii

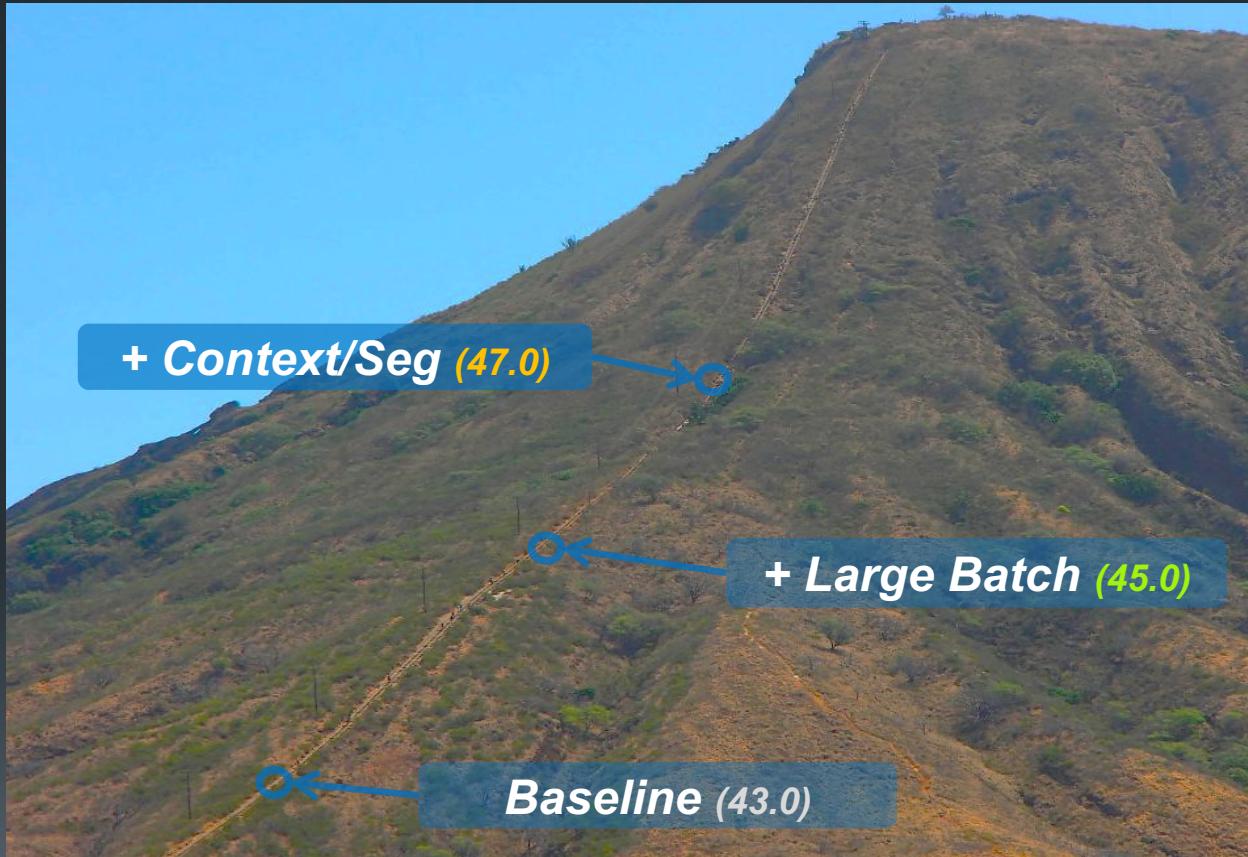
Face++ 旷视



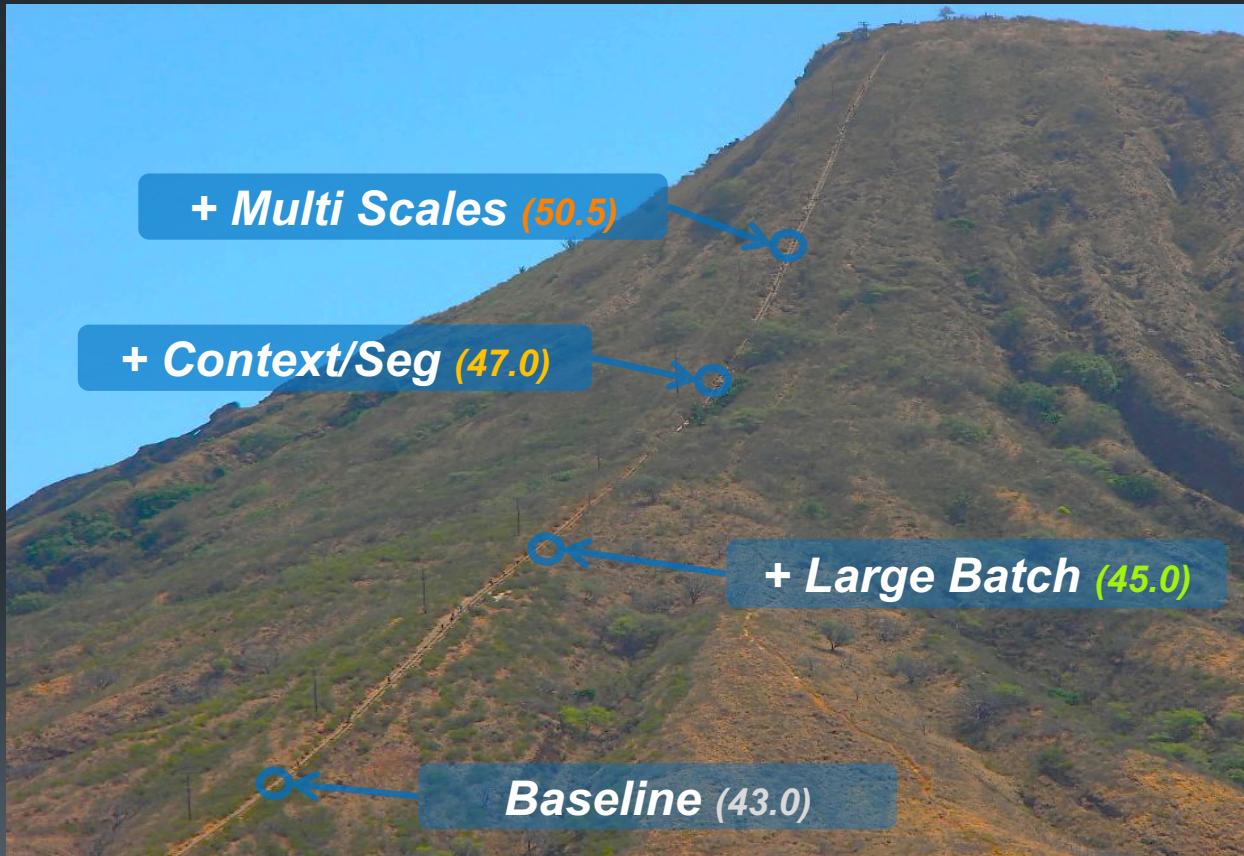
“KOKO Head” in Honolulu, Hawaii



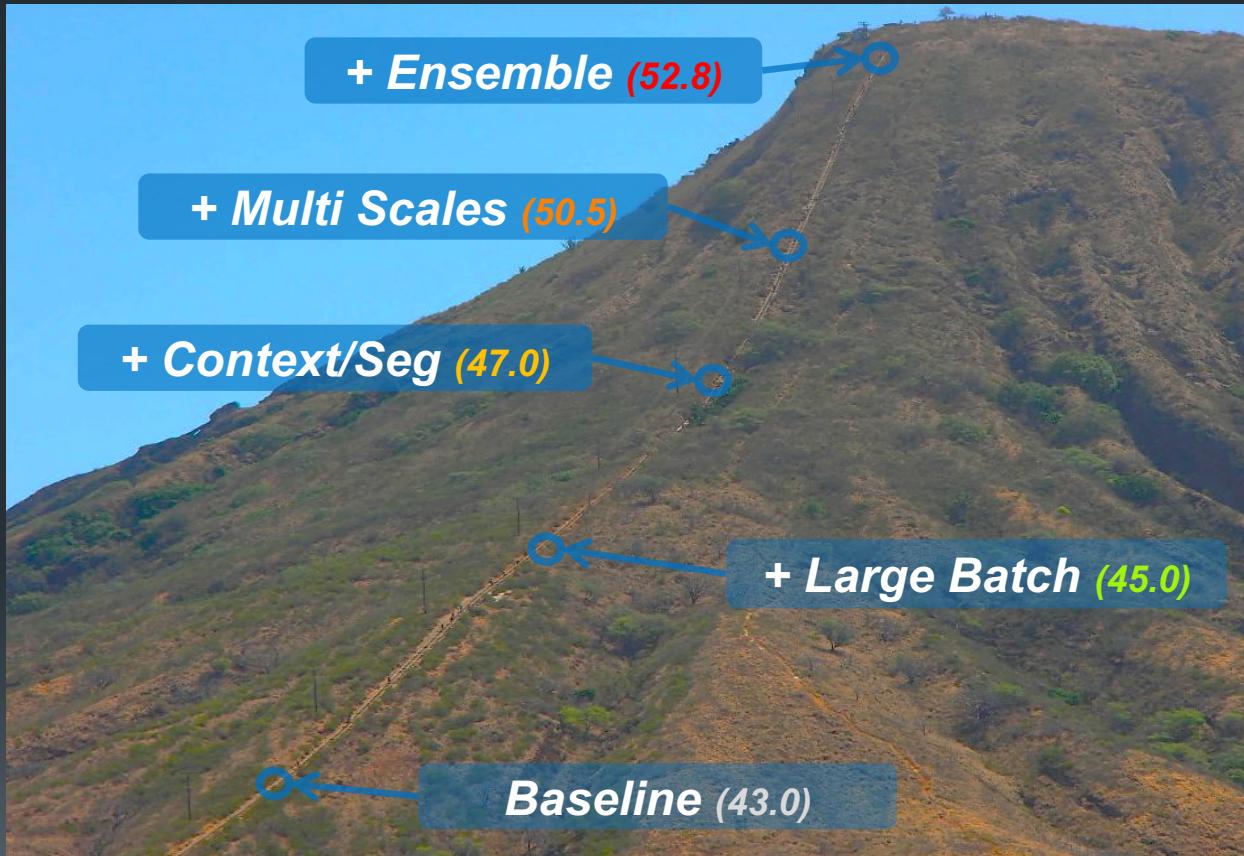
“KOKO Head” in Honolulu, Hawaii



“KOKO Head” in Honolulu, Hawaii



“KOKO Head” in Honolulu, Hawaii



“KOKO Head” in Honolulu, Hawaii

II. COCO & PLACES'17 Instance Segmentation



Ruixuan LUO*



Borui JIANG*



Tete XIAO*



Chao PENG*



Yuning JIANG



Zeming LI



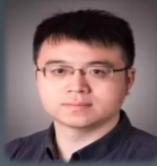
Xiangyu ZHANG



Gang YU

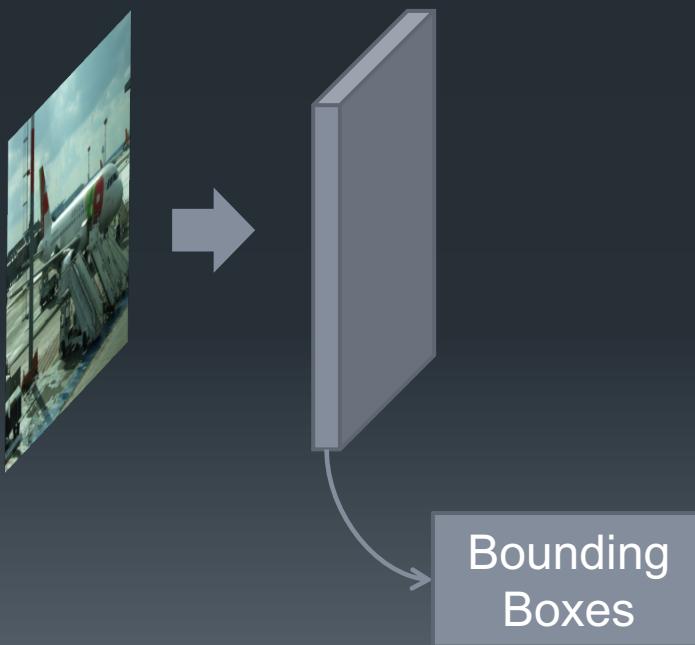


Yadong MU

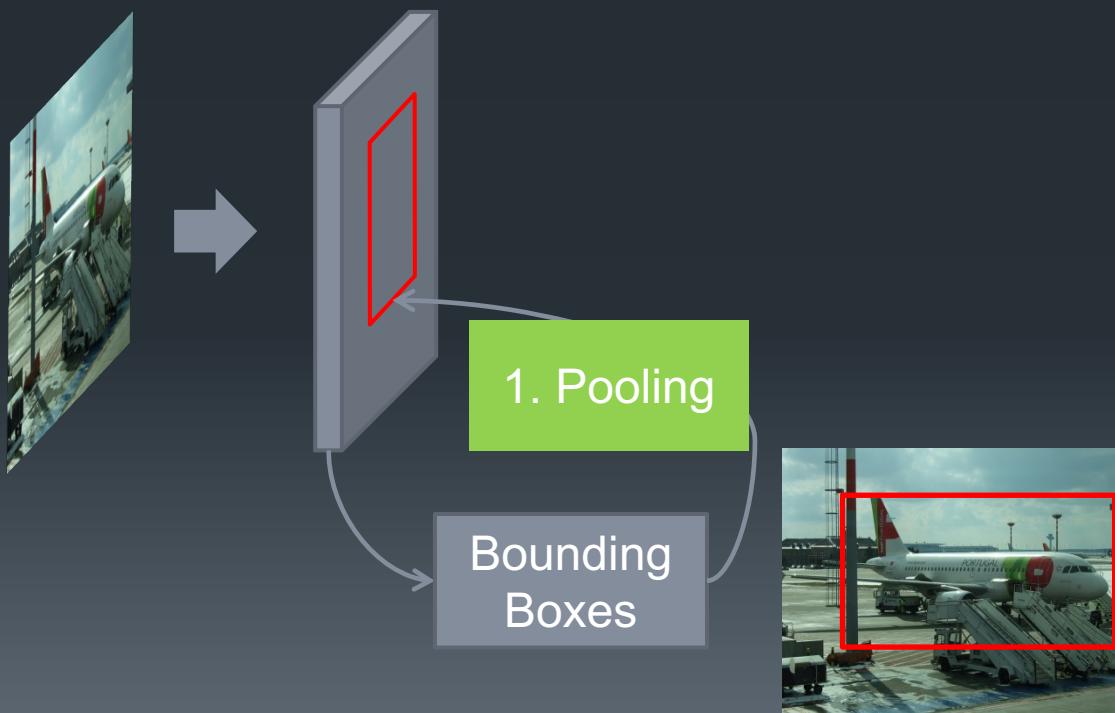


Jian SUN

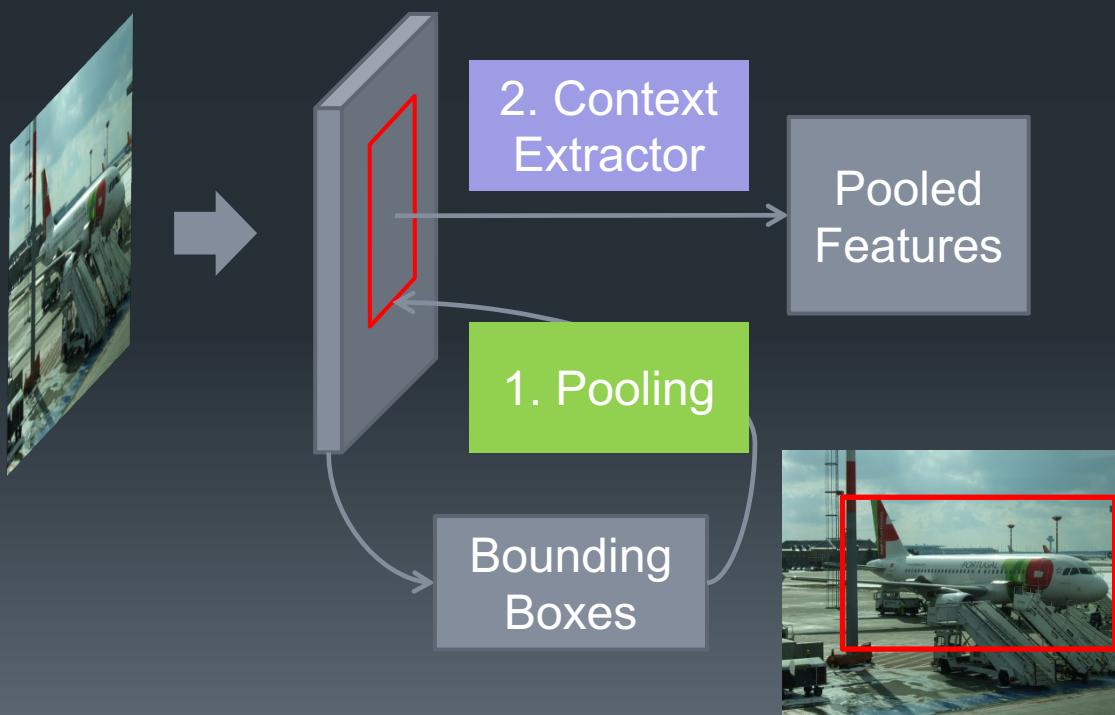
Overview



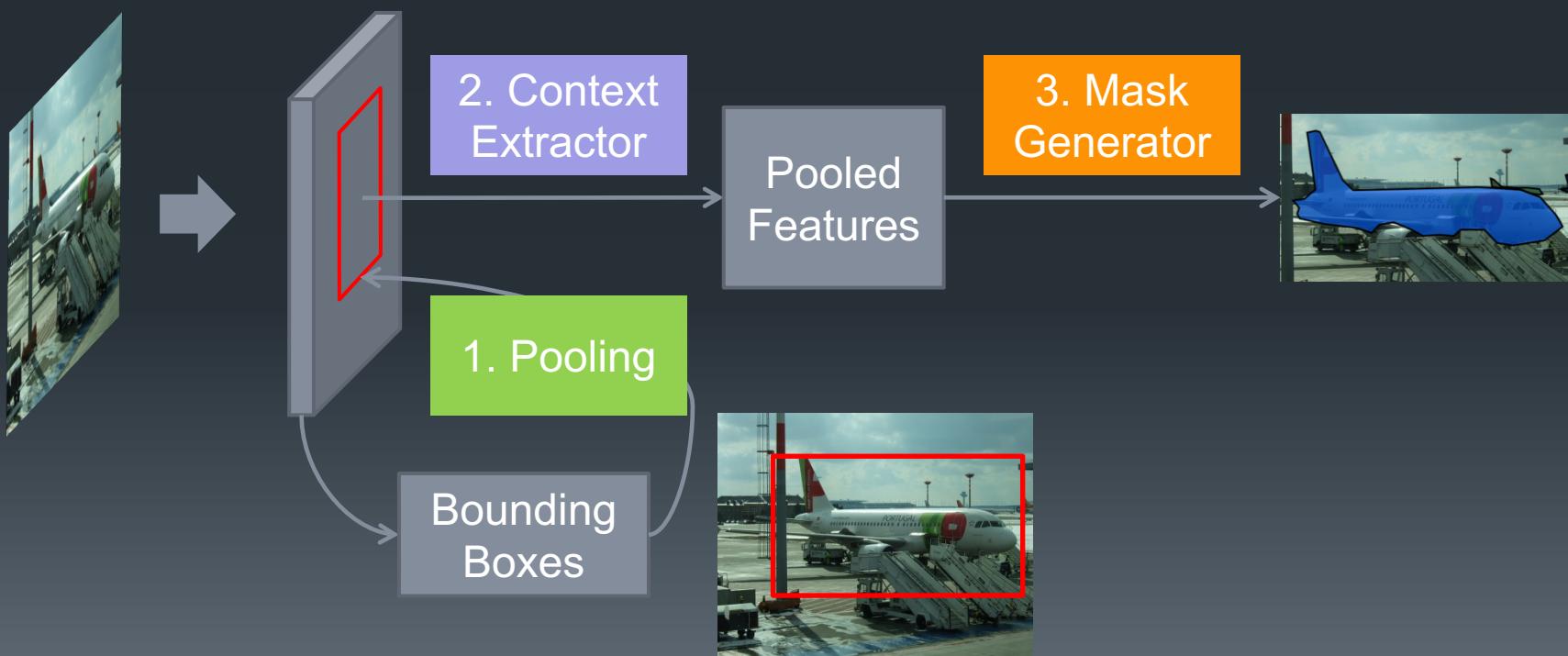
Overview



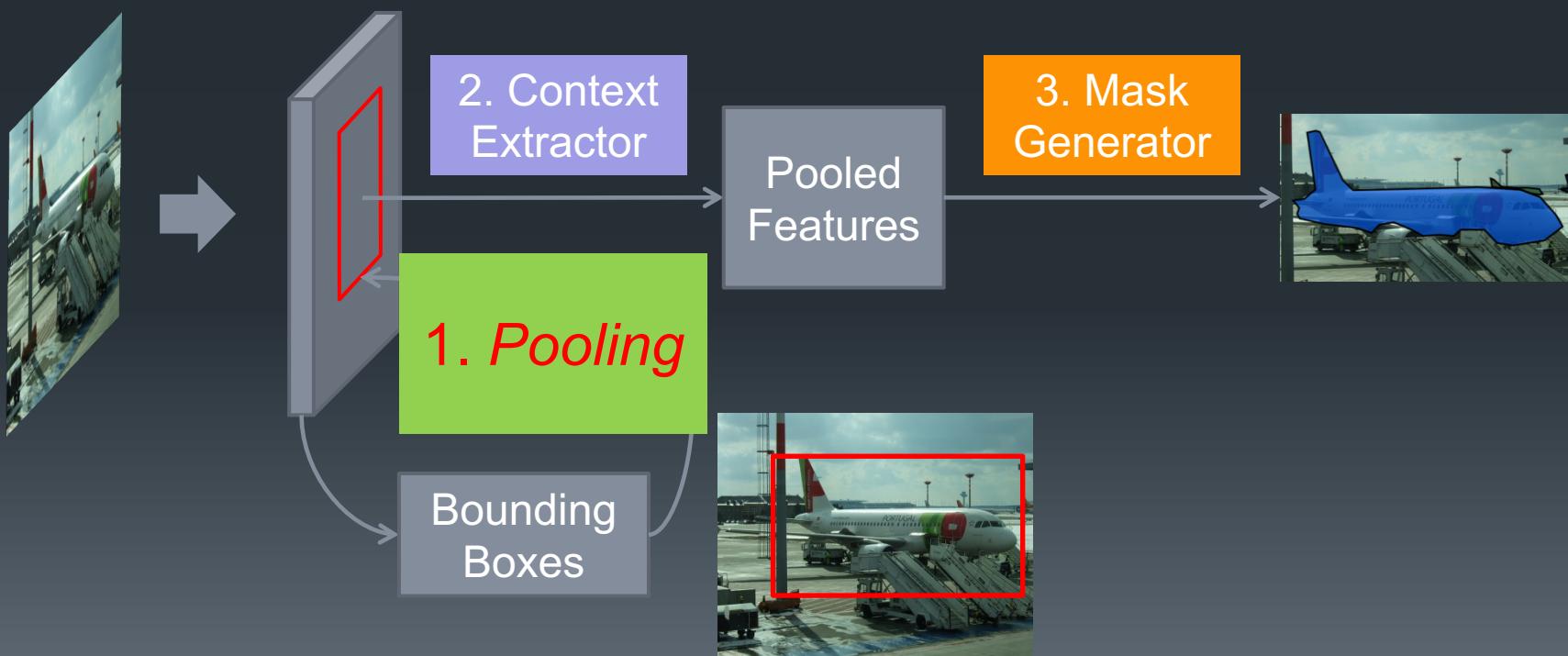
Overview



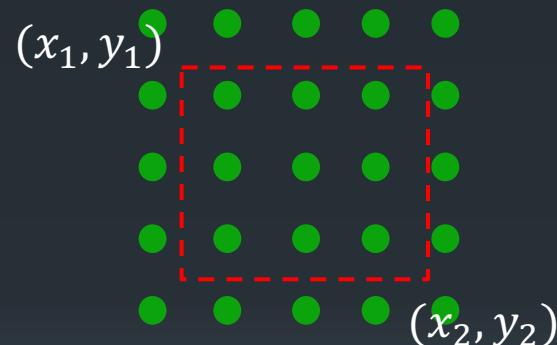
Overview



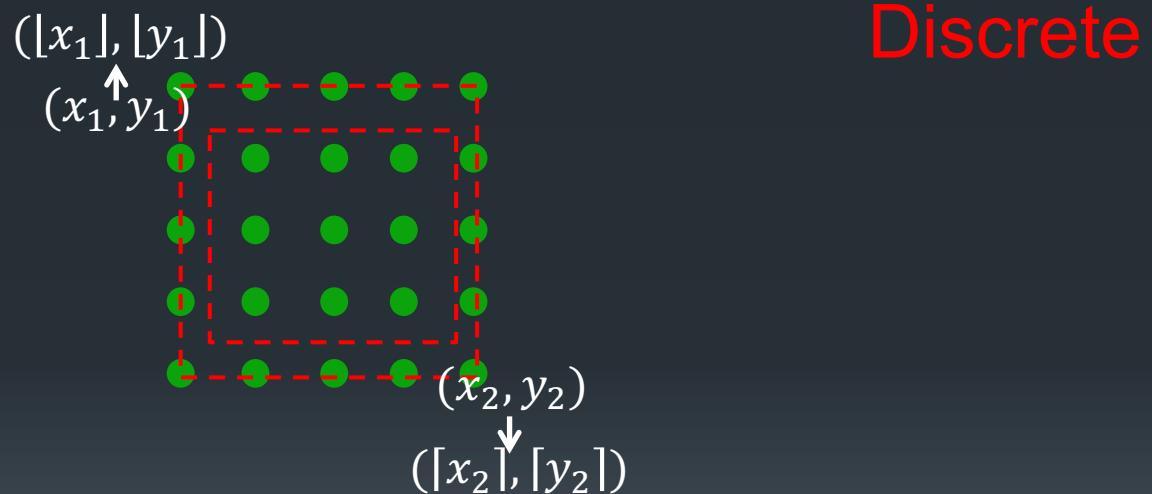
Overview



1. Pooling - Vanilla Rol Pooling



1. Pooling - Vanilla Rol Pooling



$$\frac{\sum_{i=\lfloor x_1 \rfloor}^{\lfloor x_2 \rfloor} \sum_{j=\lfloor y_1 \rfloor}^{\lfloor y_2 \rfloor} w_{i,j}}{(\lfloor x_2 \rfloor - \lfloor x_1 \rfloor + 1) * (\lfloor y_2 \rfloor - \lfloor y_1 \rfloor + 1)}$$

1. Pooling - Precise ROI Pooling

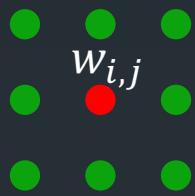


$$f(x, y) = \sum_{i,j} g(x, y, i, j) * w_{i,j}$$

where

$$g(x, y, i, j) = \max(0, 1 - |x - i|) * \max(0, 1 - |y - j|)$$

1. Pooling - Precise ROI Pooling



Discrete



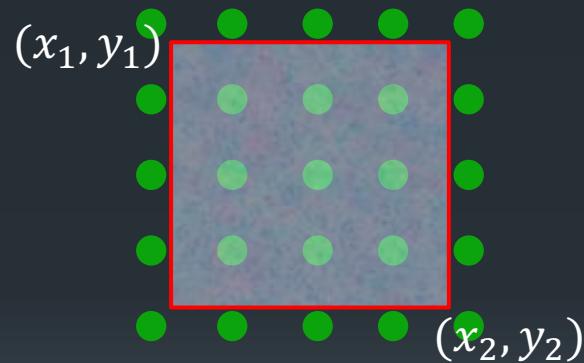
Continuous

$$f(x, y) = \sum_{i,j} g(x, y, i, j) * w_{i,j}$$

where

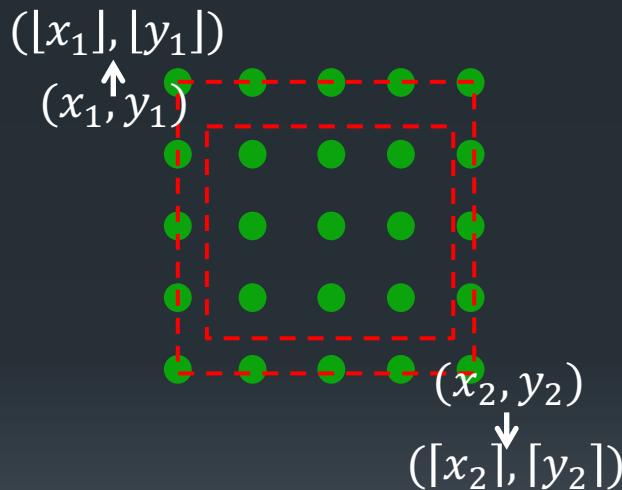
$$g(x, y, i, j) = \max(0, 1 - |x - i|) * \max(0, 1 - |y - j|)$$

1. Pooling - Precise ROI Pooling



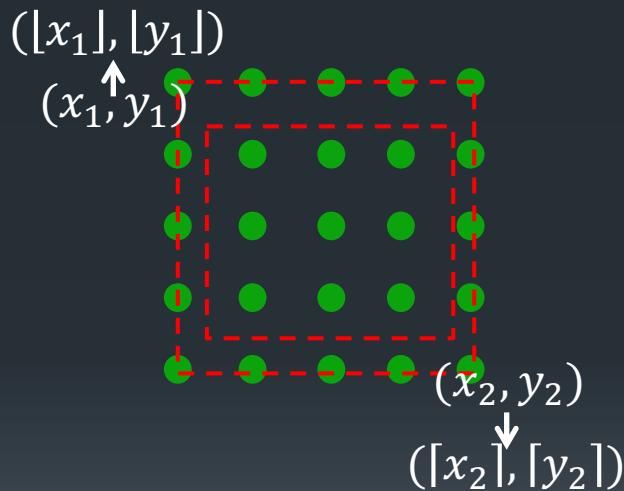
$$pr_roi(x_1, y_1, x_2, y_2) = \iint_{\substack{x_1 \leq x \leq x_2, \\ y_1 \leq y \leq y_2}} f(x, y) dx dy / ((x_2 - x_1) * (y_2 - y_1))$$

RoI Pooling



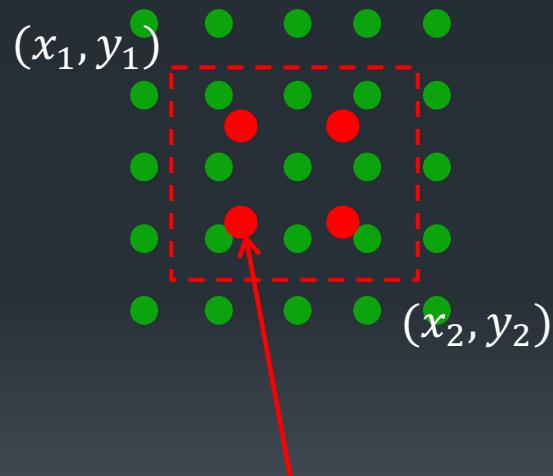
$$\frac{\sum_{i=\lfloor x_1 \rfloor}^{\lfloor x_2 \rfloor} \sum_{j=\lfloor y_1 \rfloor}^{\lfloor y_2 \rfloor} w_{i,j}}{(\lfloor x_2 \rfloor - \lfloor x_1 \rfloor + 1) * (\lfloor y_2 \rfloor - \lfloor y_1 \rfloor + 1)}$$

RoI Pooling



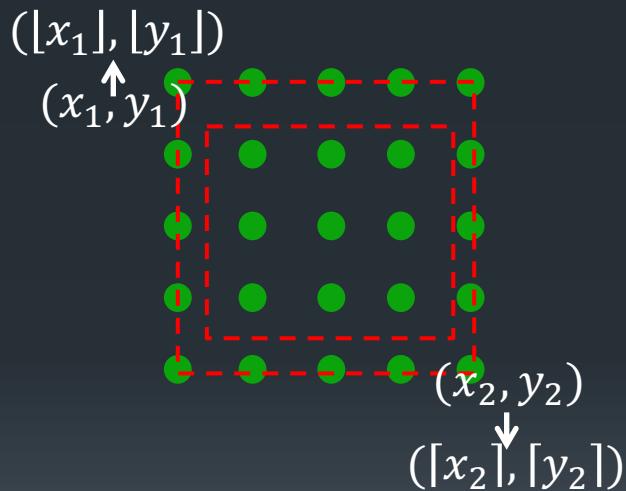
$$\frac{\sum_{i=[x_1]}^{[x_2]} \sum_{j=[y_1]}^{[y_2]} w_{i,j}}{([x_2] - [x_1] + 1) * ([y_2] - [y_1] + 1)}$$

RoI Align



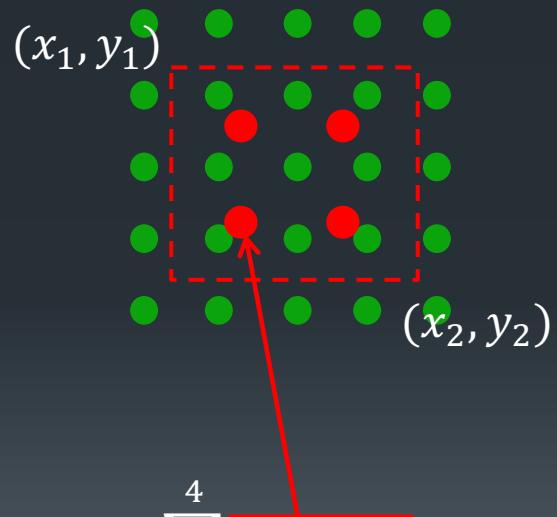
$$\sum_{i=1}^4 f(a_i, b_i) / 4$$

RoI Pooling



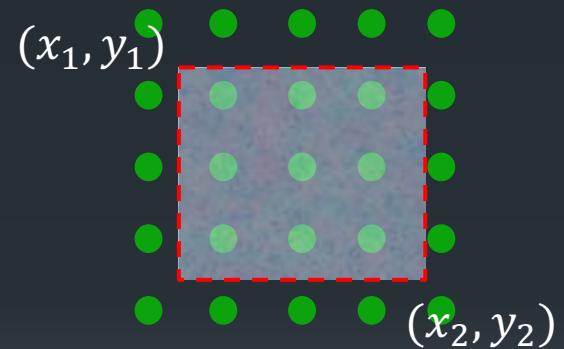
$$\frac{\sum_{i=\lfloor x_1 \rfloor}^{\lfloor x_2 \rfloor} \sum_{j=\lfloor y_1 \rfloor}^{\lfloor y_2 \rfloor} w_{i,j}}{(\lfloor x_2 \rfloor - \lfloor x_1 \rfloor + 1) * (\lfloor y_2 \rfloor - \lfloor y_1 \rfloor + 1)}$$

RoI Align



$$\sum_{i=1}^4 f(a_i, b_i) / 4$$

Precise RoI Pooling



$$\frac{\iint_{\substack{x_1 \leq x \leq x_2, \\ y_1 \leq y \leq y_2}} f(x, y) dx dy}{(x_2 - x_1) * (y_2 - y_1)}$$

2. Context Extractor

layer $x+1$



layer x

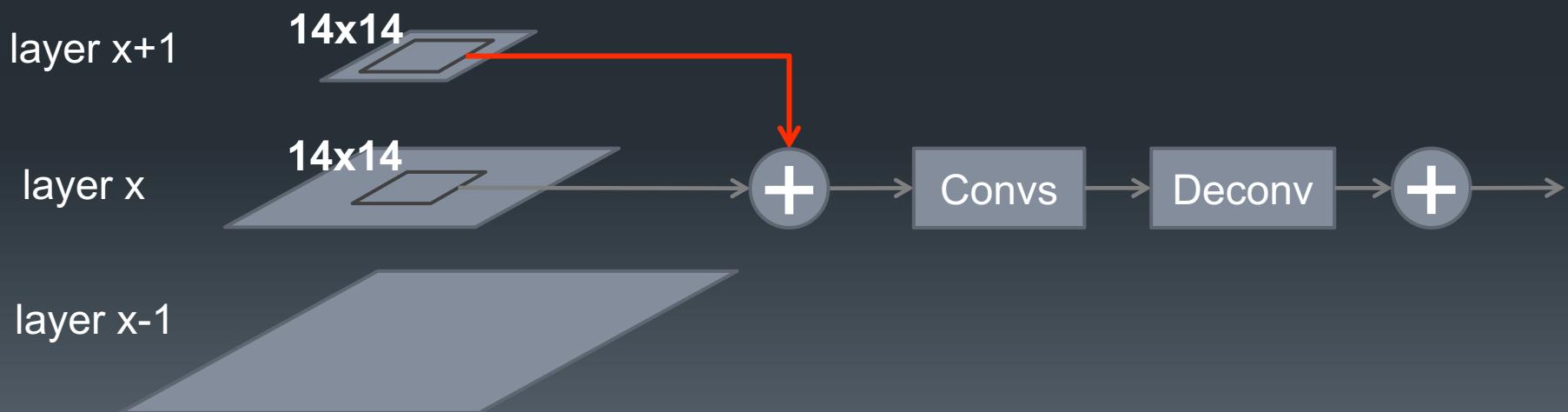
14x14



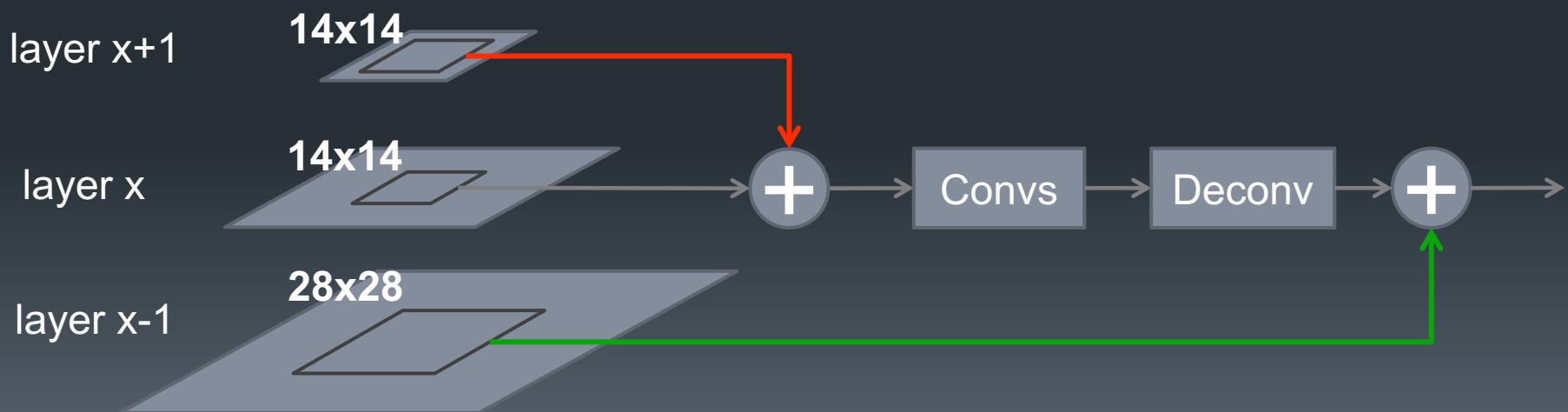
layer $x-1$



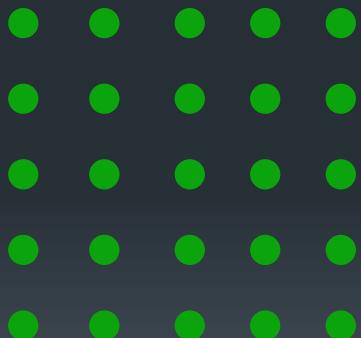
2. Context Extractor



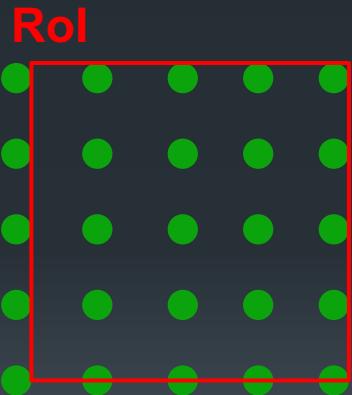
2. Context Extractor



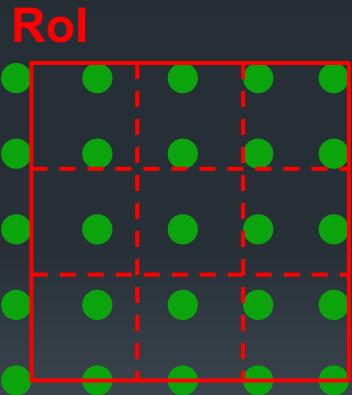
3. Mask Generator



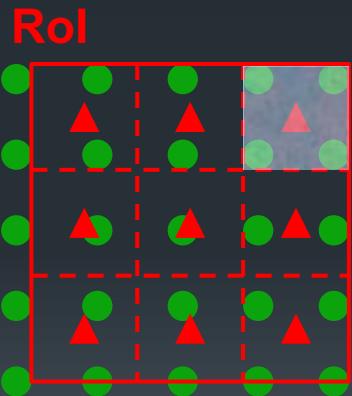
3. Mask Generator



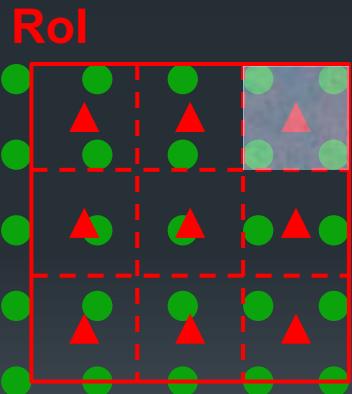
3. Mask Generator



3. Mask Generator

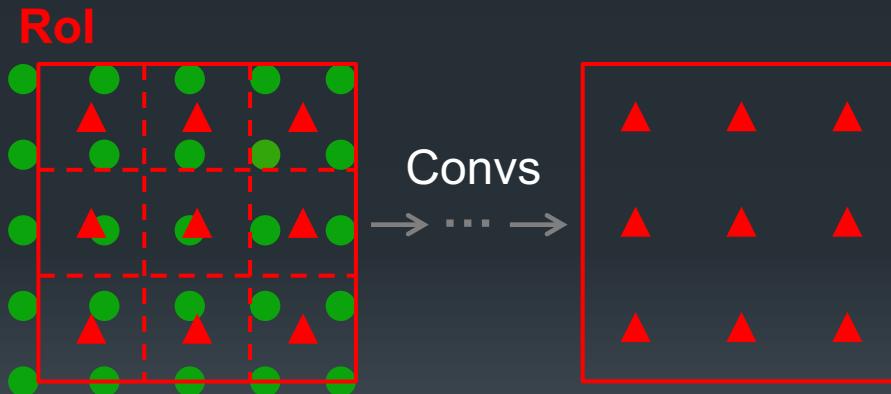


3. Mask Generator



Pixel ● → Grid ▲

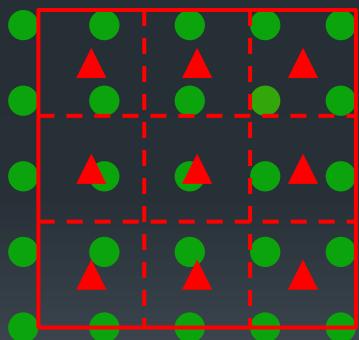
3. Mask Generator



Pixel ● → *Grid* ▲

3. Mask Generator

RoI



Convs
→ ... →



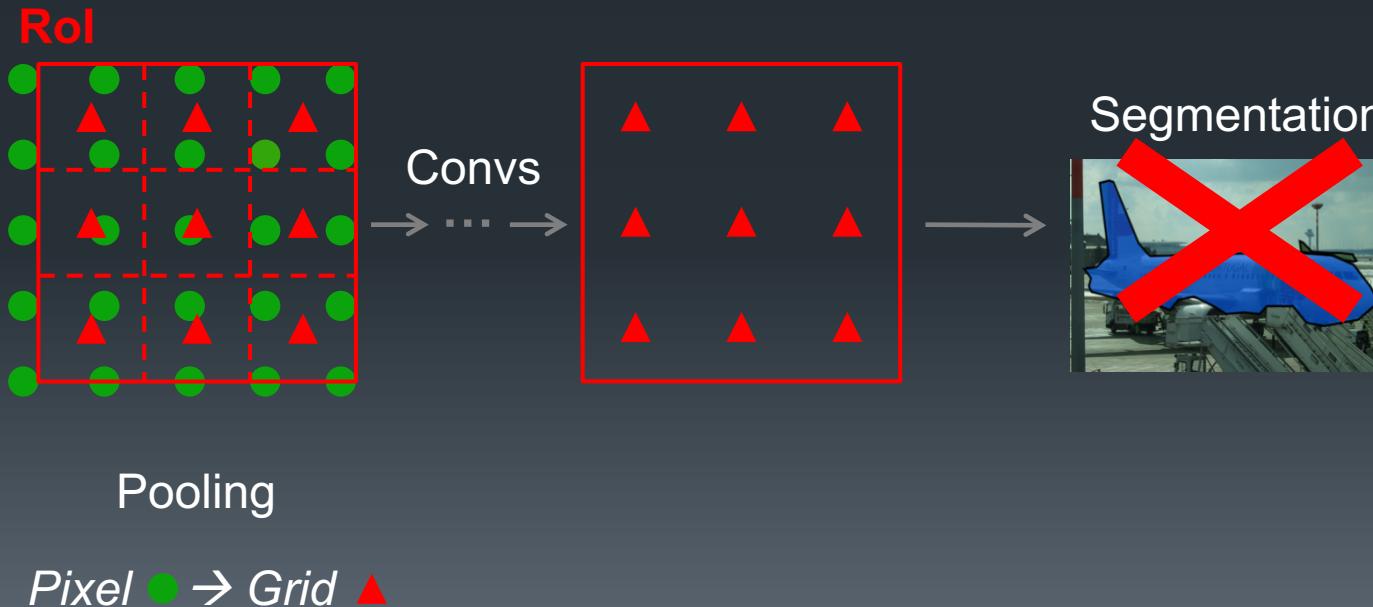
Segmentation



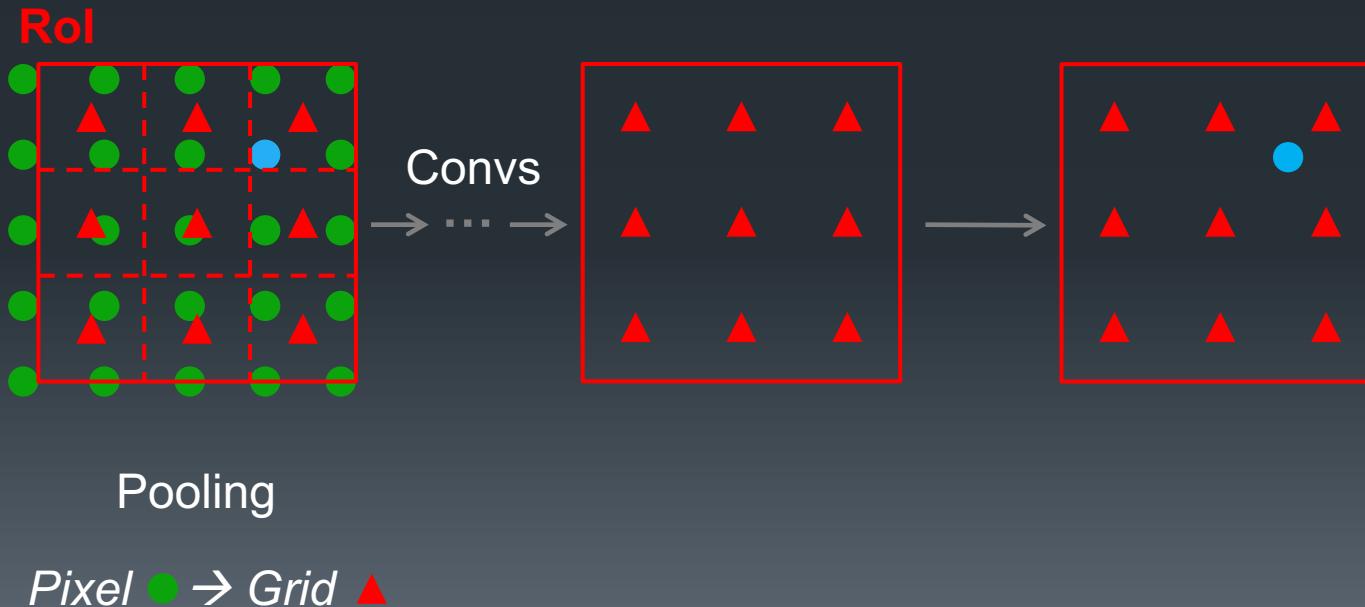
Pooling

Pixel ● → Grid ▲

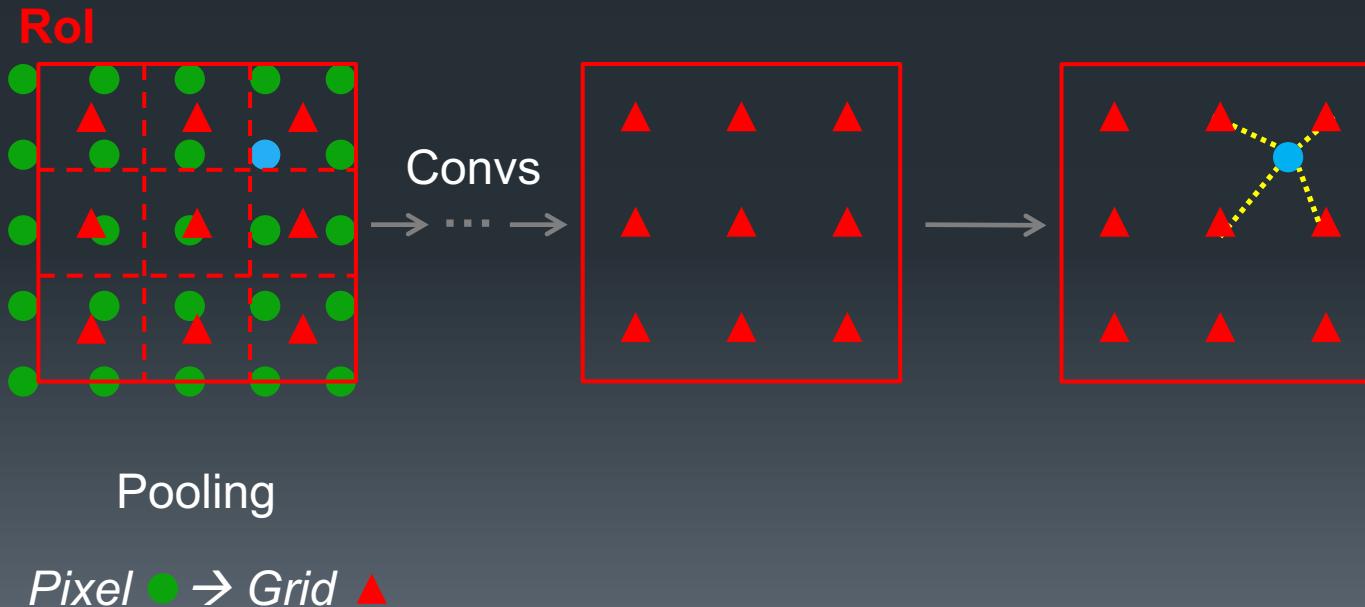
3. Mask Generator



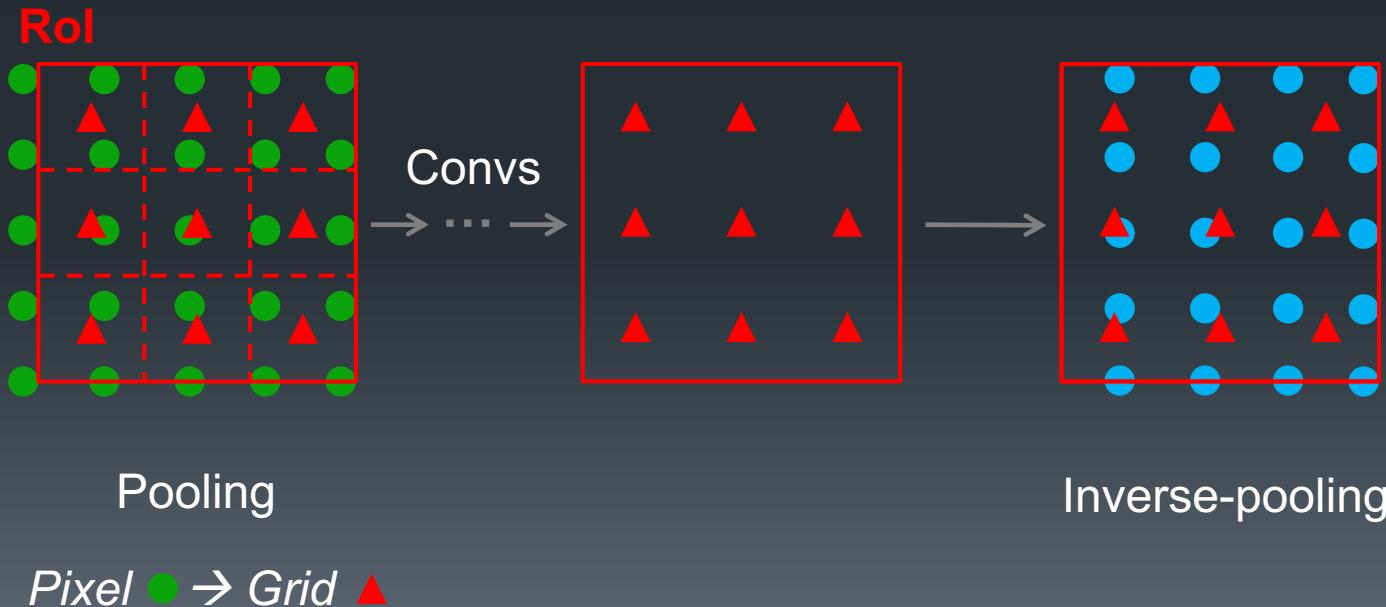
3. Mask Generator



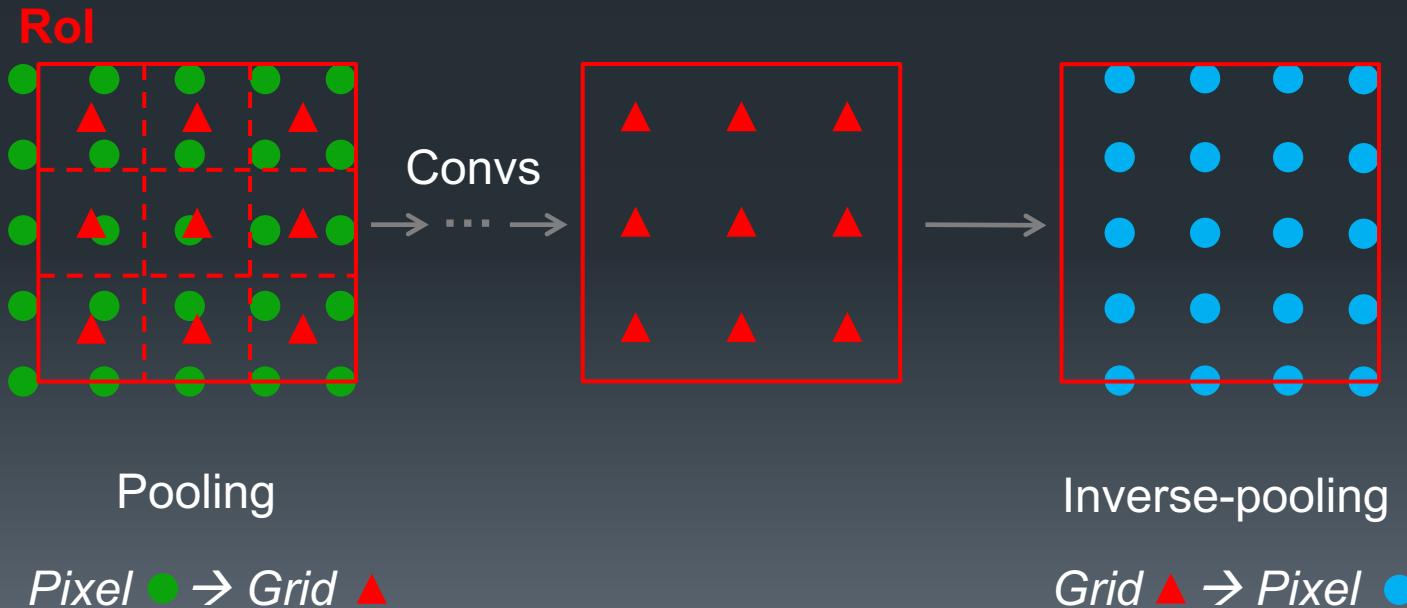
3. Mask Generator



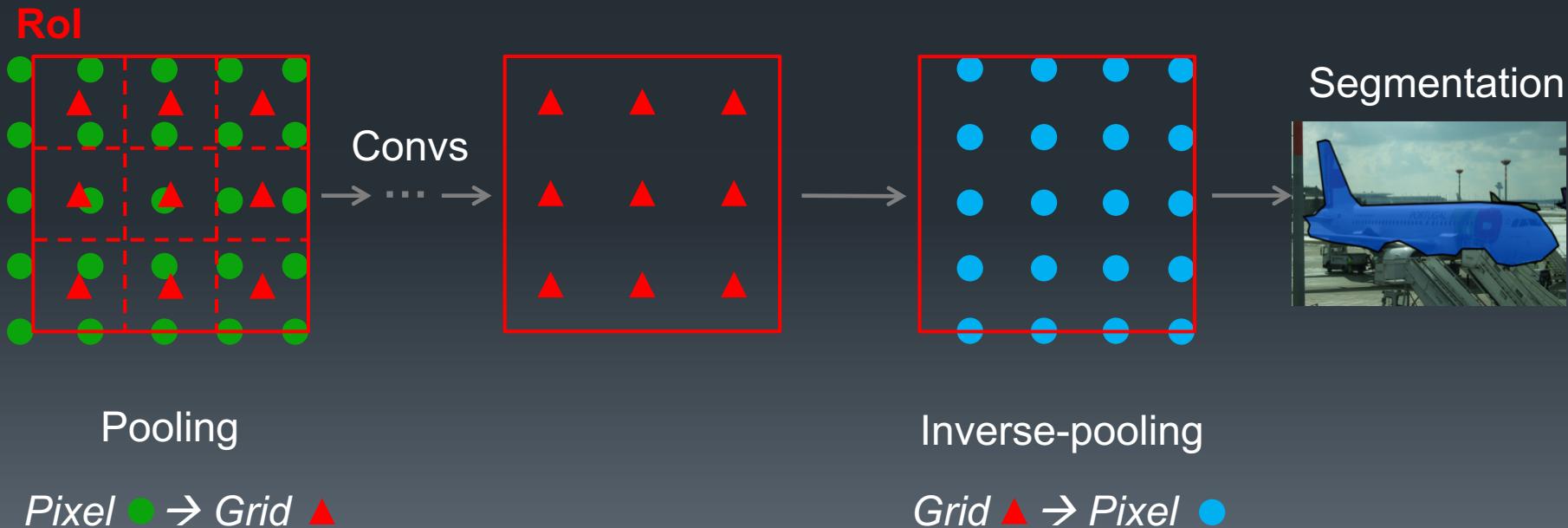
3. Mask Generator



3. Mask Generator

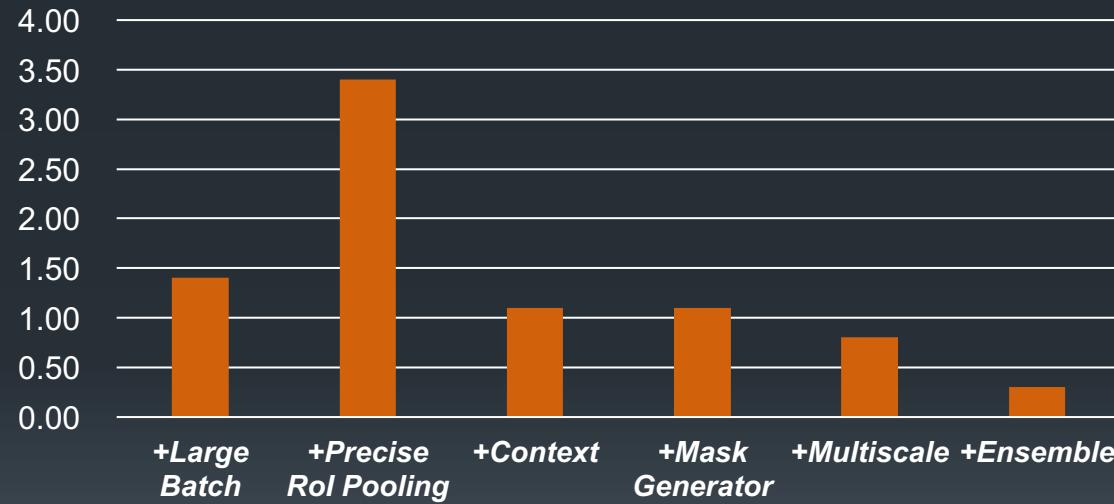


3. Mask Generator



Results

Improvement on COCO



Track	Ensemble	Single	Bbox
COCO	46.4 (2 models)	46.1	52.8
PLACES	30.7 (3 models)	29.8	35.1

Conclusion

- MegDet, the first large-batch detector proves successful in detection task.

Conclusion

- MegDet, the first large-batch detector proves successful in detection task.
- For segmentation, the devil lies in pixels.

Conclusion

- MegDet, the first large-batch detector proves successful in detection task.
- For segmentation, the devil lies in pixels.
- *Training fast is the key for rapid innovation cycle.*



We are hiring!

@Beijing, @Nanjing, @Seattle

career@megvii.com

Thanks & Questions



COCO team on “KOKO Head” :)