Semantic Segmentation / Instance Segmentation Based on Deep learning

Yiding Liu

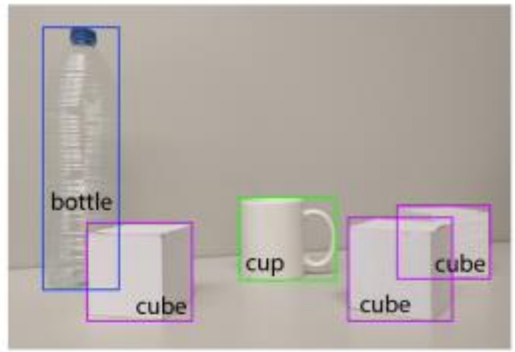
2018.12.08

Outline

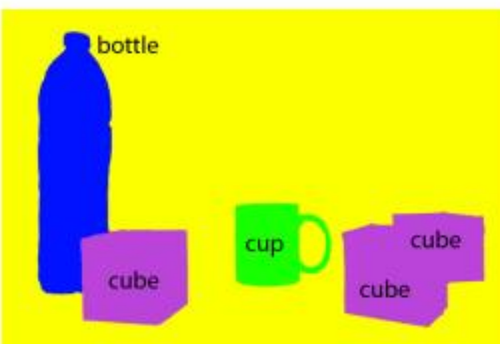
- Overview of segmentation problem
- Semantic segmentation
- Instance Segmentation
- Our work

Definition of segmentation problem

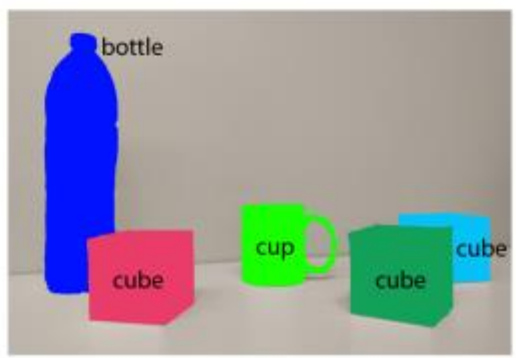
(a) Image classification



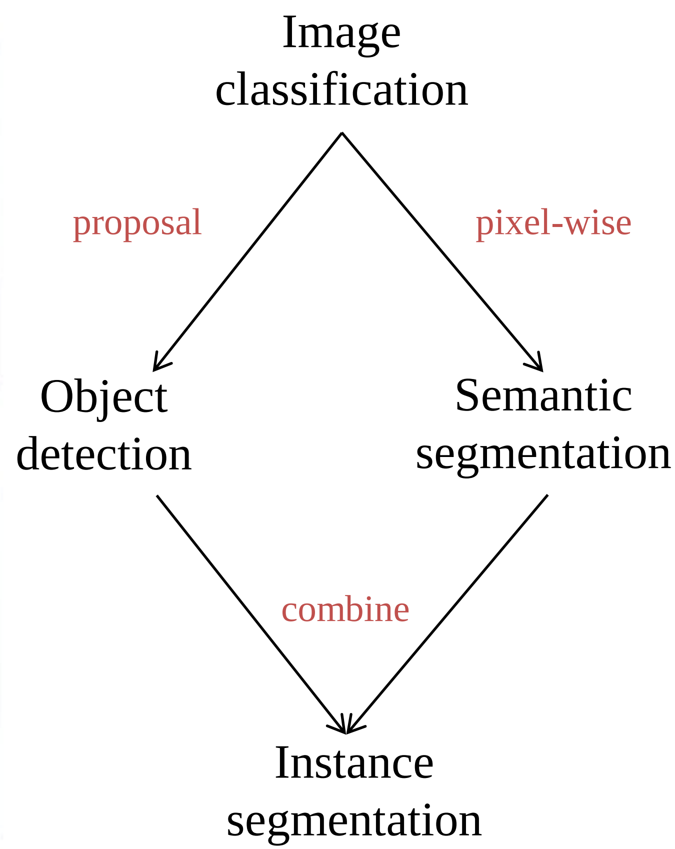
(b) Object localization



(c) Semantic segmentation



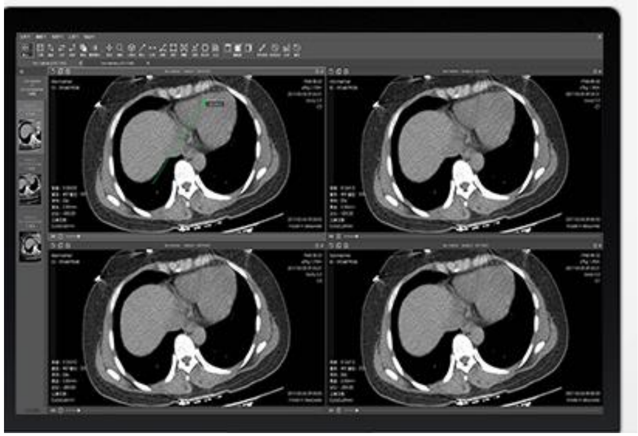
(d) Instance segmentation



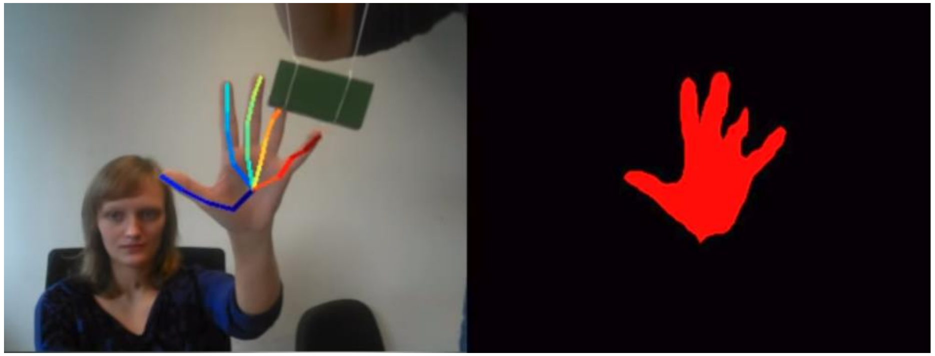
Applications

Autonomous driving

Medical treatment



Human-person interaction



…

Semantic segmentation

- make dense predictions inferring labels for every pixel

Fully Convolution Network

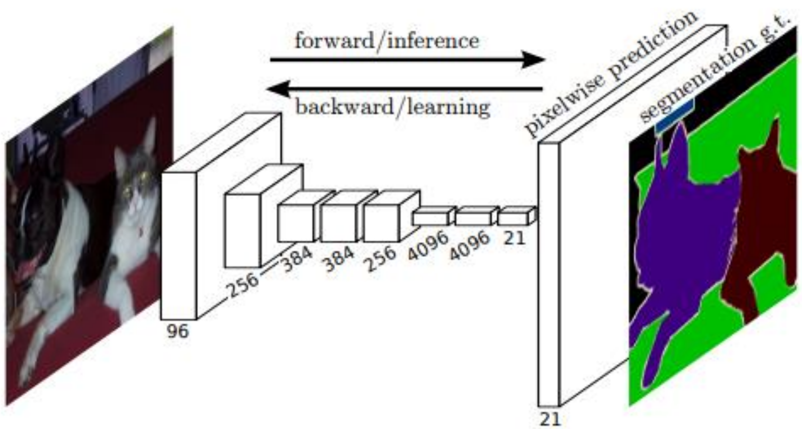


Figure 1. Fully convolutional networks can efficiently learn to make dense predictions for per-pixel tasks like semantic segmentation.

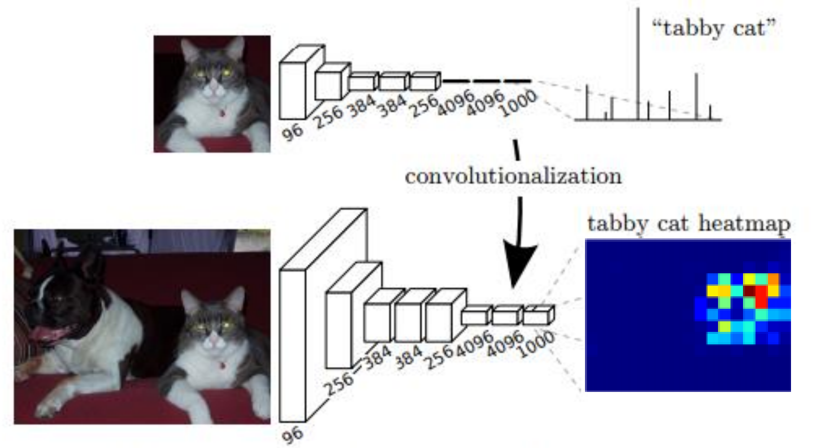


Figure 2. Transforming fully connected layers into convolution layers enables a classification net to output a heatmap. Adding layers and a spatial loss (as in Figure 1) produces an efficient machine for end-to-end dense learning.

Challenges

- Resolution
  - 32x down-sample for classic classification models at pool5

factor	mean IU
128	50.9
64	73.3
32	86.1
16	92.8
8	96.4
4	98.5

- Contexts
  - Objects may have multiple scales and it is hard for convolution kernels to handle a large variation of scales

FCN

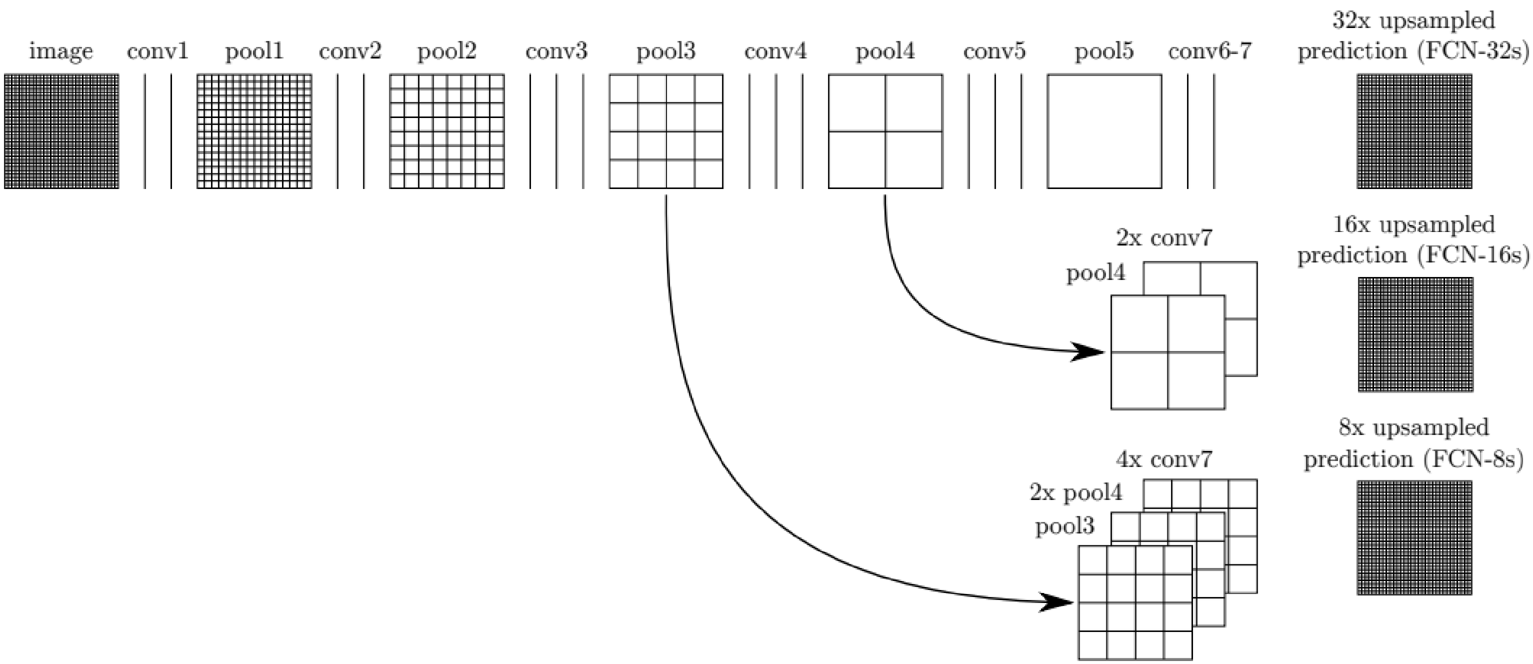


Figure 3. Our DAG nets learn to combine coarse, high layer information with fine, low layer information. Pooling and prediction layers are shown as grids that reveal relative spatial coarseness, while intermediate layers are shown as vertical lines. First row (FCN-32s): Our single-stream net, described in Section 4.1, upsamples stride 32 predictions back to pixels in a single step. Second row (FCN-16s): Combining predictions from both the final layer and the `pool4` layer, at stride 16, lets our net predict finer details, while retaining high-level semantic information. Third row (FCN-8s): Additional predictions from `pool3`, at stride 8, provide further precision.

Long J, Shelhamer E, Darrell T. **Fully convolutional networks for semantic segmentation** CVPR 2015.

SegNet

- Upsample with corresponding pooling indices

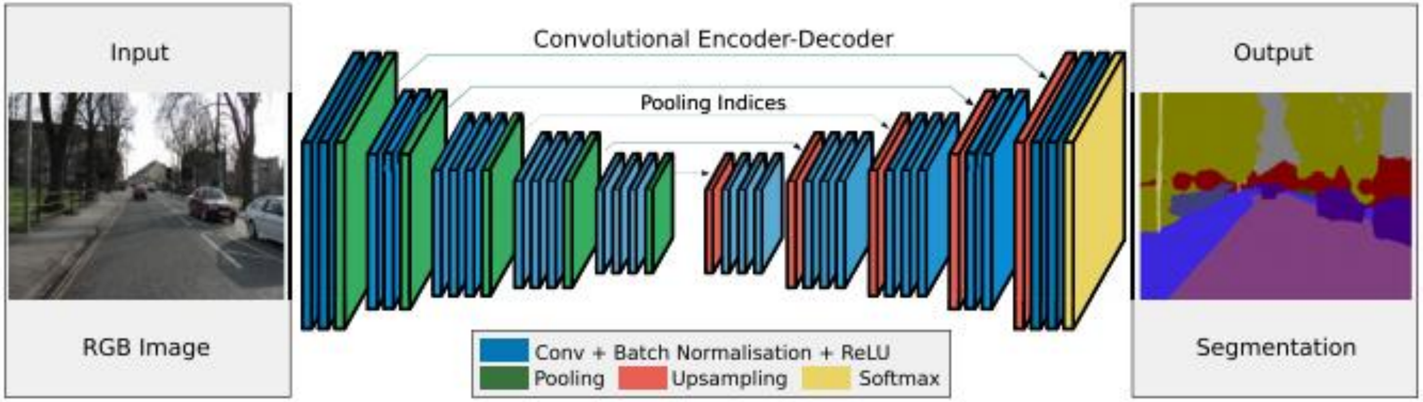
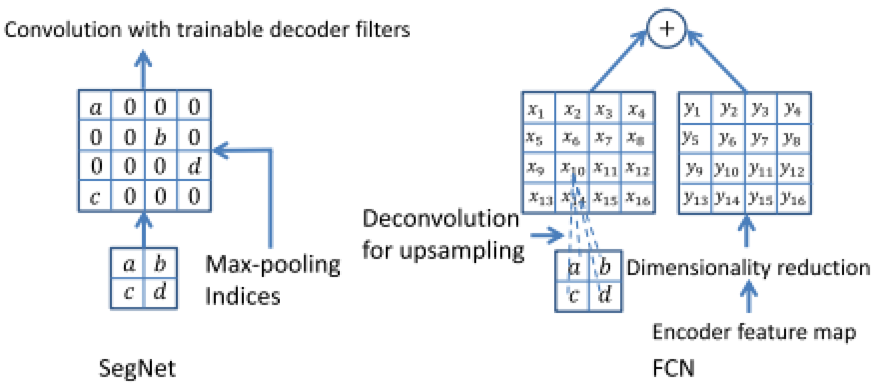


Fig. 2. An illustration of the SegNet architecture. There are no fully connected layers and hence it is only convolutional. A decoder upsamples its input using the transferred pool indices from its encoder to produce a sparse feature map(s). It then performs convolution with a trainable filter bank to densify the feature map. The final decoder output feature maps are fed to a soft-max classifier for pixel-wise classification.



Badrinarayanan V, Kendall A, Cipolla R. **SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation** TPAMI 2017

U-Net

- Dense concatenation with encoder features

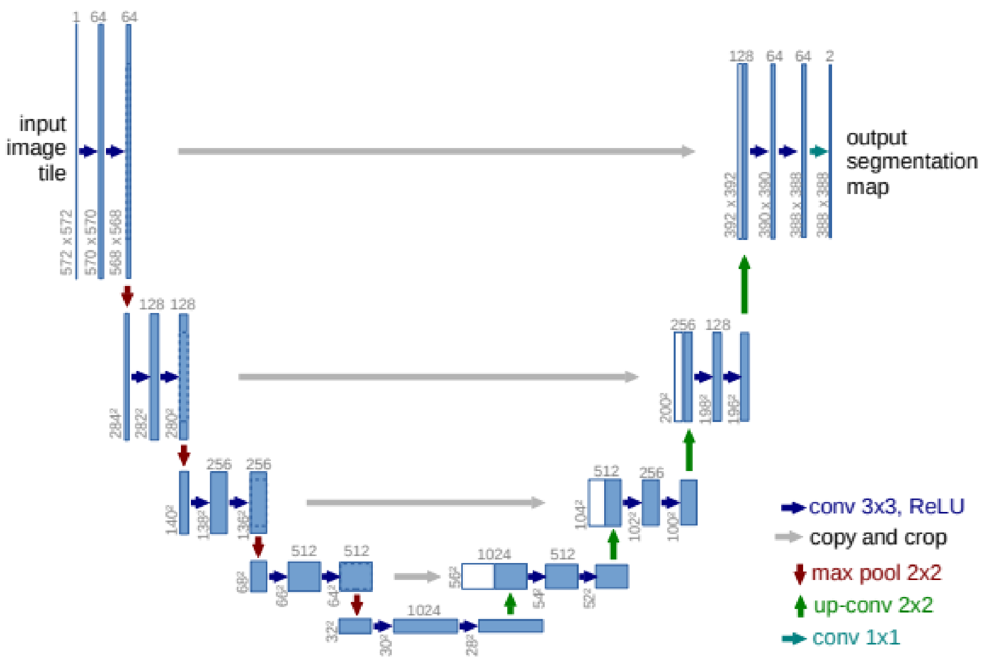
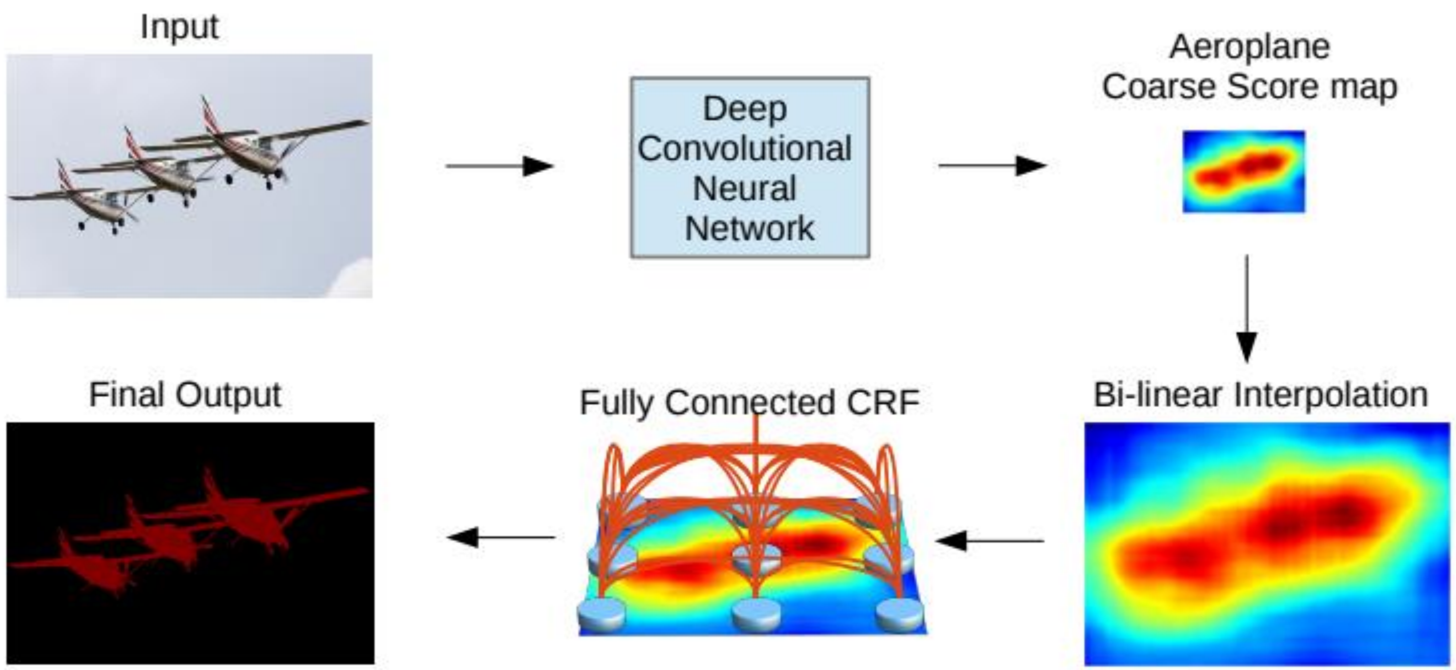


Fig. 1. U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations.

Ronneberger O, Fischer P, Brox T. **U-net: Convolutional networks for biomedical image segmentation** MICCAI 2015

Deeplab



L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. **Semantic image segmentation with deep convolutional nets and fully connected CRFs**. ICLR 2015

Deeplab

- Dilated convolution
  - Remove last few pooling operation for a dense prediction.
  - Introduce dilated convolution to utilize the ImageNet pre-trained model

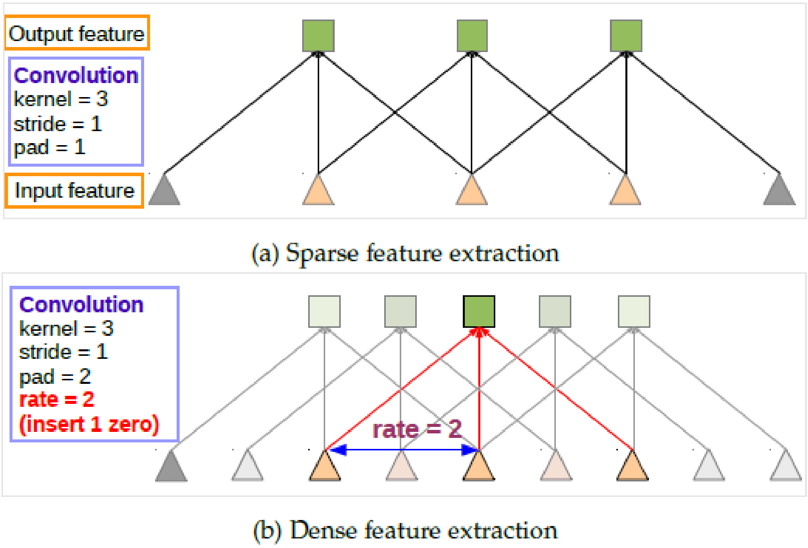


Fig. 2: Illustration of atrous convolution in 1-D. (a) Sparse feature extraction with standard convolution on a low resolution input feature map. (b) Dense feature extraction with atrous convolution with rate $r = 2$, applied on a high resolution input feature map.

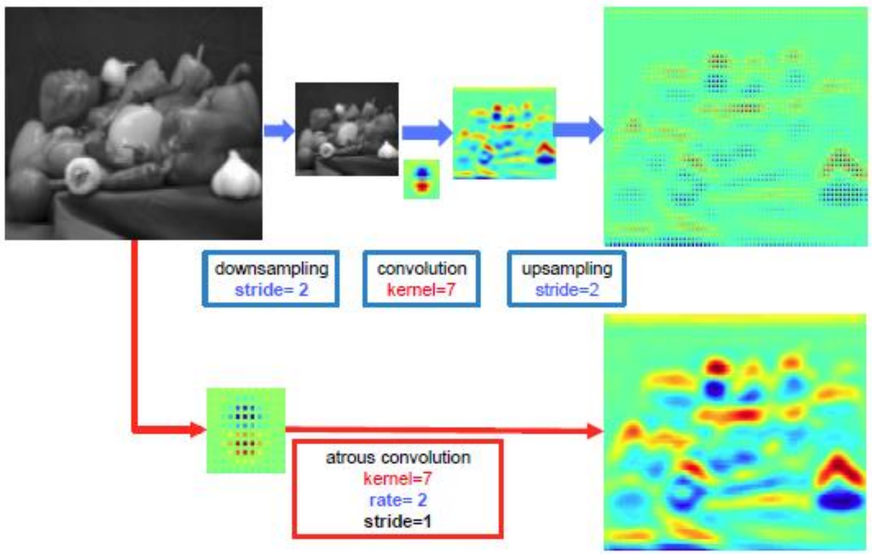
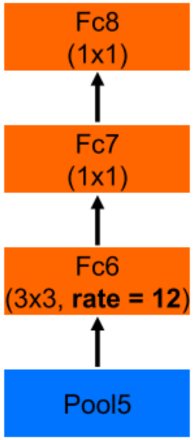


Fig. 3: Illustration of atrous convolution in 2-D. Top row: sparse feature extraction with standard convolution on a low resolution input feature map. Bottom row: Dense feature extraction with atrous convolution with rate $r = 2$, applied on a high resolution input feature map.

L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. **Semantic image segmentation with deep convolutional nets and fully connected CRFs**. ICLR 2015

Deeplab

- LargeFOV
  - Dilated convolution with large rate can capture features with a large field of view.



(a) DeepLab-LargeFOV

- Multi-scale Prediction
  - Jump connection for more precise boundaries

L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. **Semantic image segmentation with deep convolutional nets and fully connected CRFs**. ICLR 2015

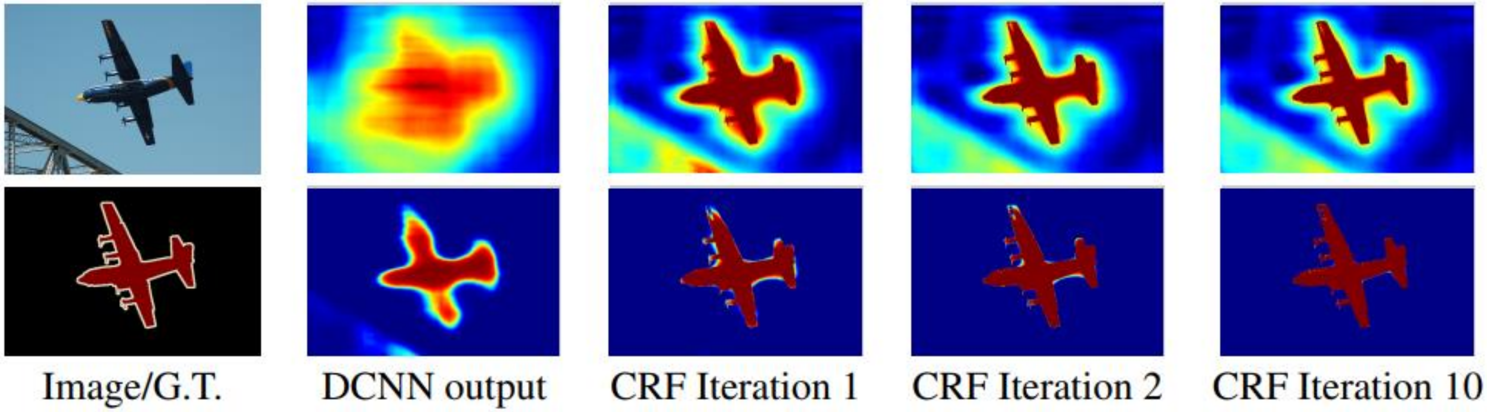
Deeplab

- Fully connected CRF
  - Refine boundaries

$$E(\boldsymbol{x}) = \sum_i \theta_i(x_i) + \sum_{ij} \theta_{ij}(x_i, x_j)$$

$$\theta_i(x_i) = -\log P(x_i)$$

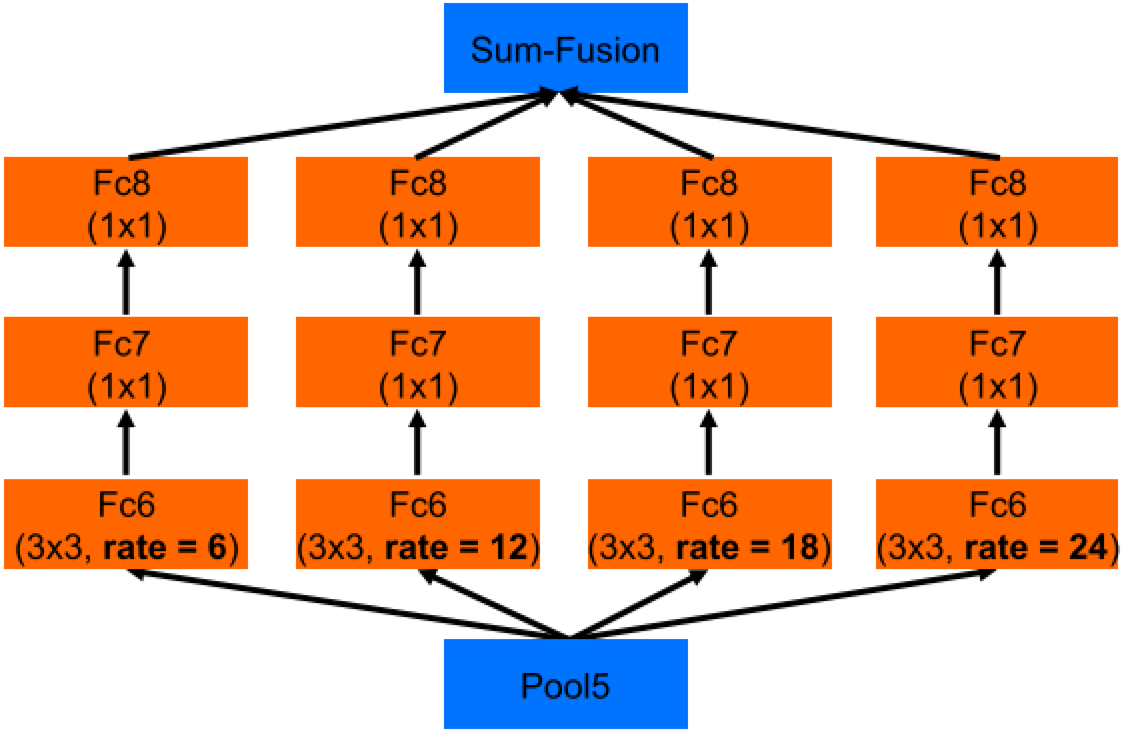
$$\theta_{ij}(x_i, x_j) \;=\; w_1 \exp\Big(-\frac{||p_i - p_j||^2}{2\sigma_\alpha^2} - \frac{||I_i - I_j||^2}{2\sigma_\beta^2}\Big) + w_2 \exp\Big(-\frac{||p_i - p_j||^2}{2\sigma_\gamma^2}\Big)$$



L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. **Semantic image segmentation with deep convolutional nets and fully connected CRFs**. ICLR 2015

Deeplab v2

- Atrous spatial pyramid pooling(ASPP)

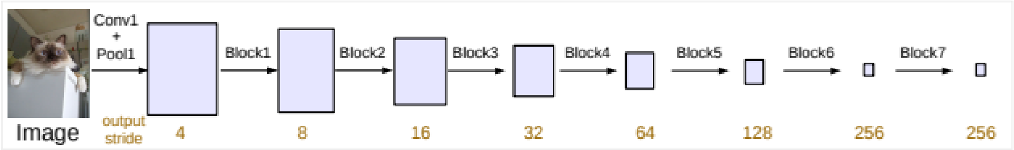


(b) DeepLab-ASPP

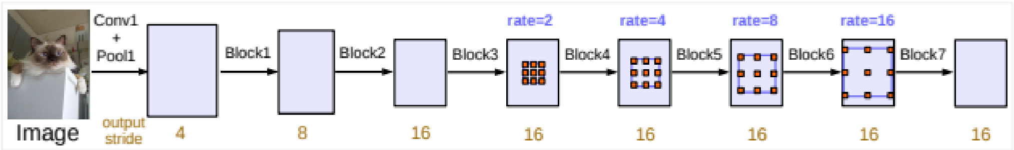
Chen L C, Papandreou G, Kokkinos I, et al. **Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs** TPAMI 2018

Deeplab v3

Deeper models



(a) Going deeper without atrous convolution.



(b) Going deeper with atrous convolution. Atrous convolution with *rate* > 1 is applied after block3 when *output_stride* $= 16$.
Figure 3. Cascaded modules without and with atrous convolution.

Parallel modules

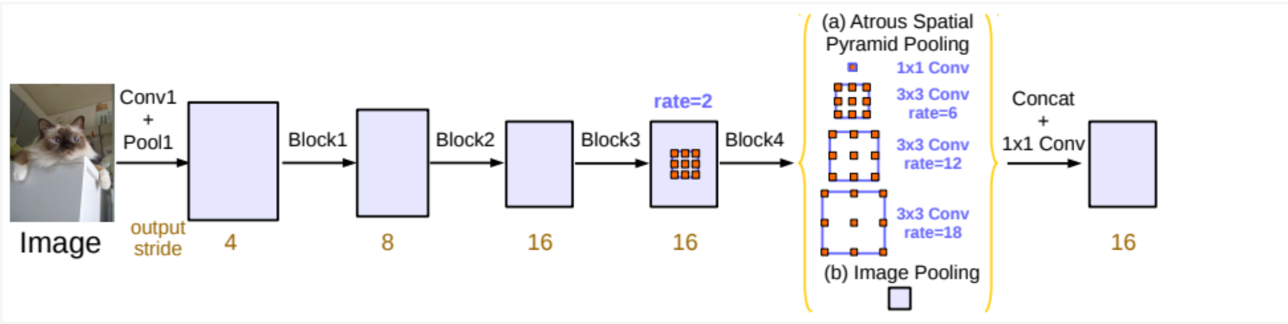


Figure 5. Parallel modules with atrous convolution (ASPP), augmented with image-level features.

Chen L C, Papandreou G, Schroff F, et al. **Rethinking atrous convolution for semantic image segmentation** arXiv 2017

Deeplab v3+

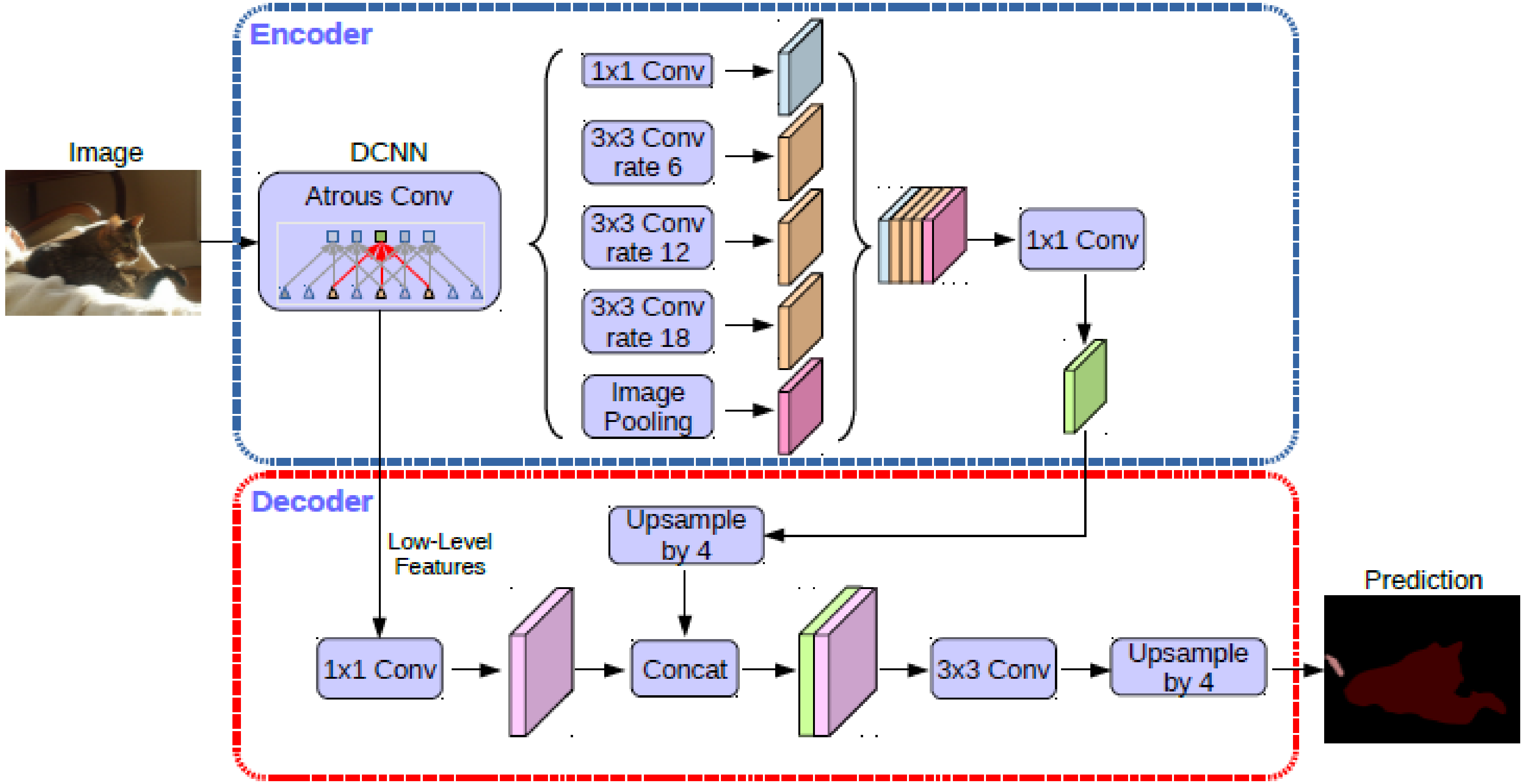
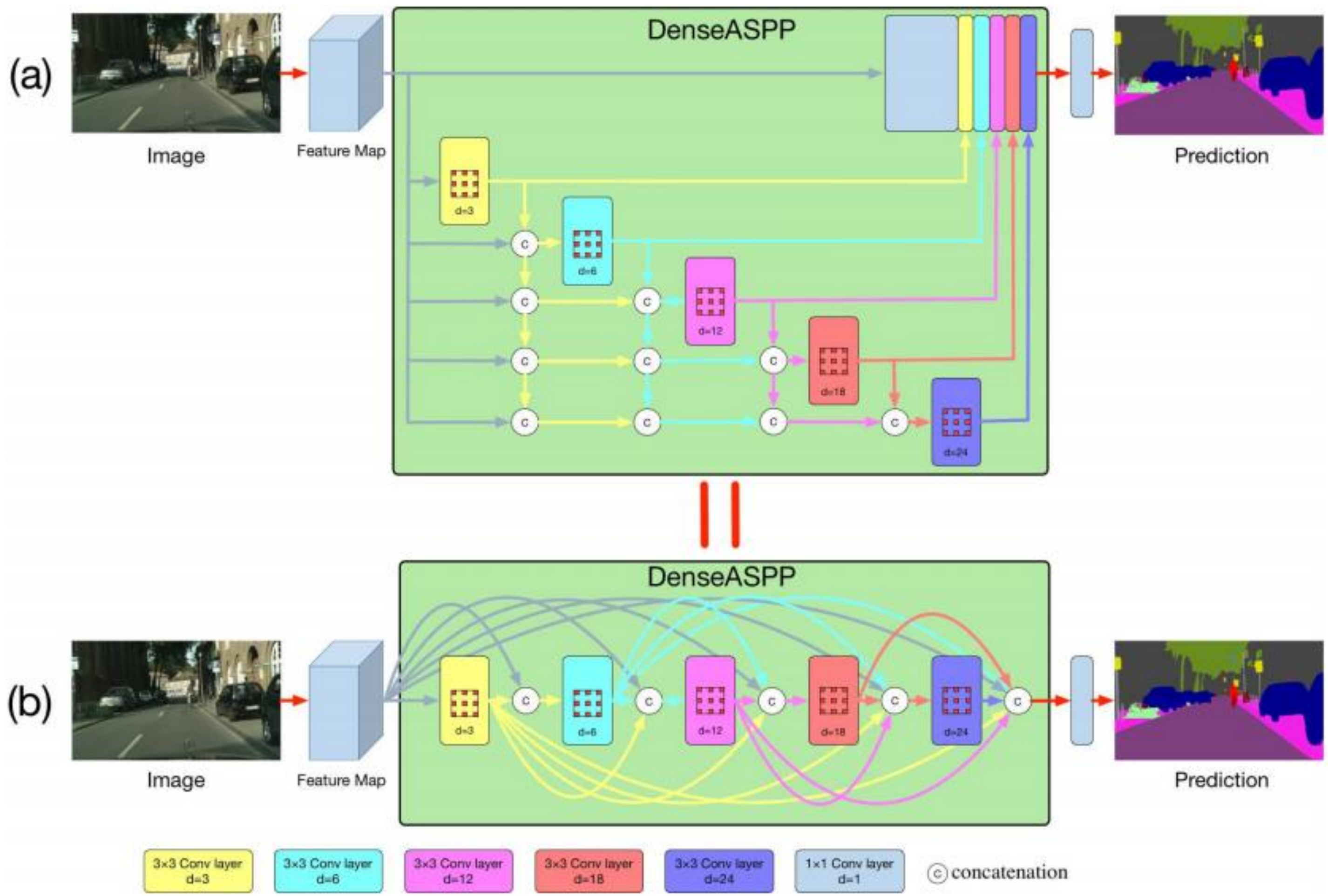


Figure 2. Our proposed DeepLabv3+ extends DeepLabv3 by employing a encoder-decoder structure. The encoder module encodes multi-scale contextual information by applying atrous convolution at multiple scales, while the simple yet effective decoder module refines the segmentation results along object boundaries.

Chen, Liang-Chieh,Zhu, Yukun et al**. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation** ECCV 2018

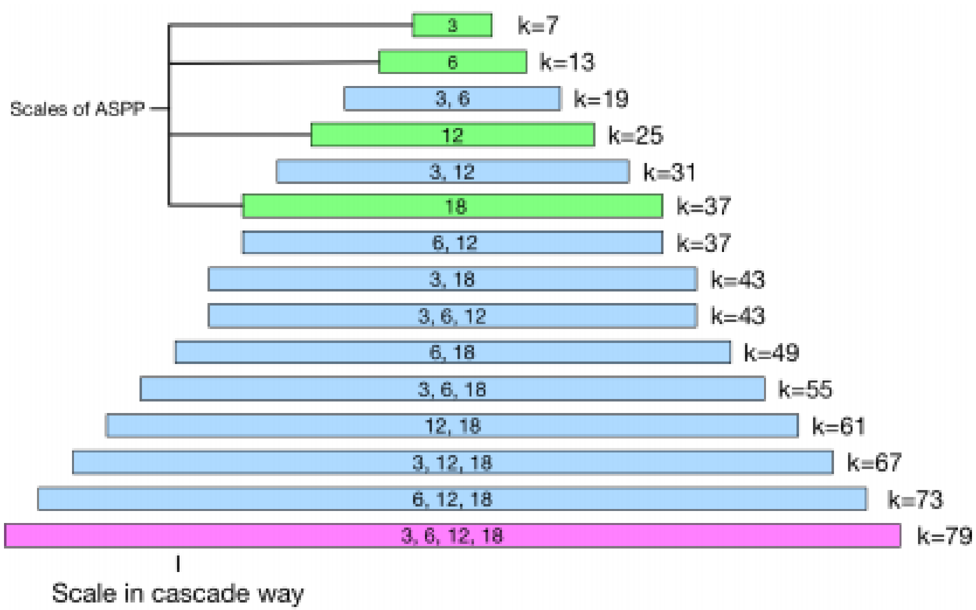
DenseASPP



Maoke Yang, Kun Yu, Chi Zhang, Zhiwei Li, Kuiyuan Yang **DenseASPP for Semantic Segmentation in Street Scenes** CVPR 2018

DenseASPP

- Scale diversity



Maoke Yang, Kun Yu, Chi Zhang, Zhiwei Li, Kuiyuan Yang **DenseASPP for Semantic Segmentation in Street Scenes** CVPR 2018

PSPNet

- Pyramid pooling / deep supervision

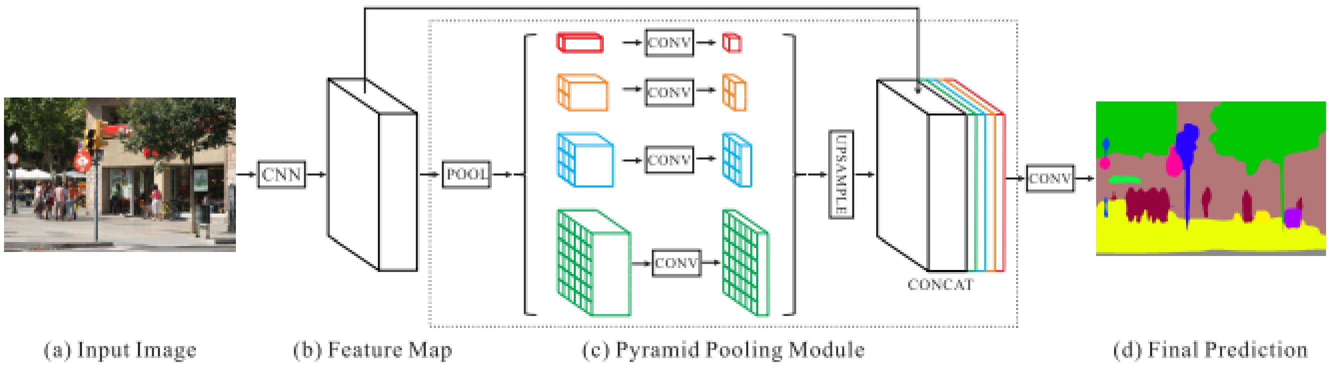


Figure 3. Overview of our proposed PSPNet. Given an input image (a), we first use CNN to get the feature map of the last convolutional layer (b), then a pyramid parsing module is applied to harvest different sub-region representations, followed by upsampling and concatenation layers to form the final feature representation, which carries both local and global context information in (c). Finally, the representation is fed into a convolution layer to get the final per-pixel prediction (d).

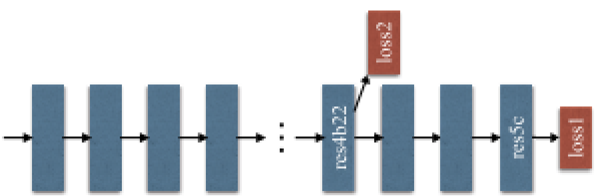
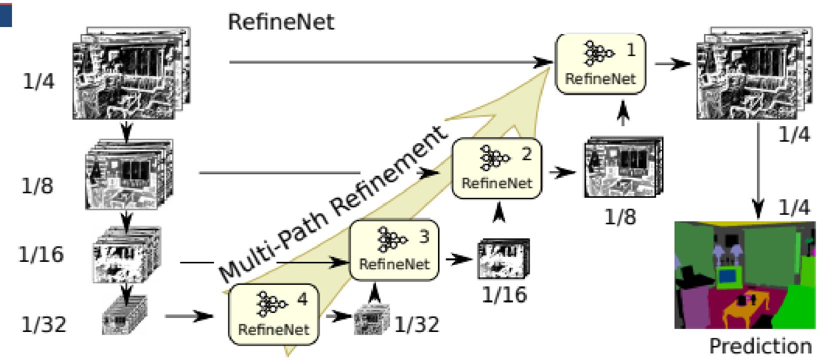


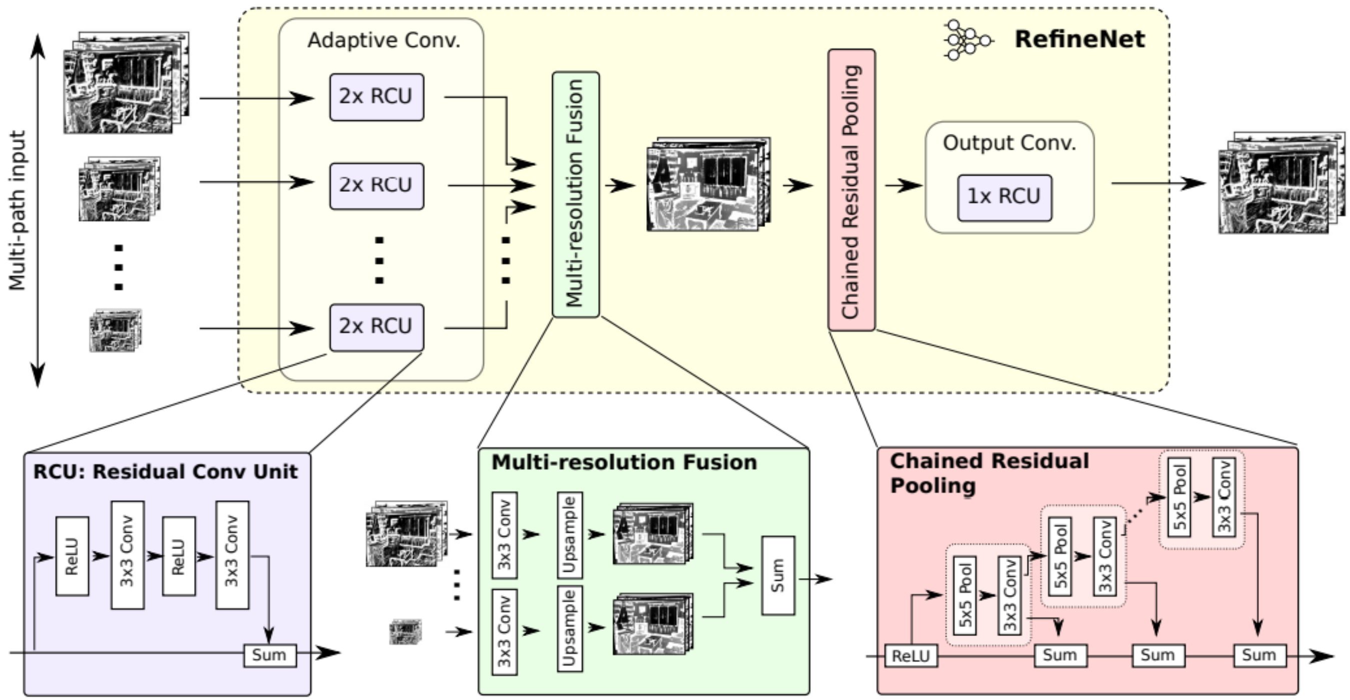
Figure 4. Illustration of auxiliary loss in ResNet101. Each blue box denotes a residue block. The auxiliary loss is added after the res4b22 residue block.

Zhao H, Shi J, Qi X, et al. **Pyramid scene parsing network** CVPR 2017

RefineNet

- Fuse multiple strides
- Residual pooling



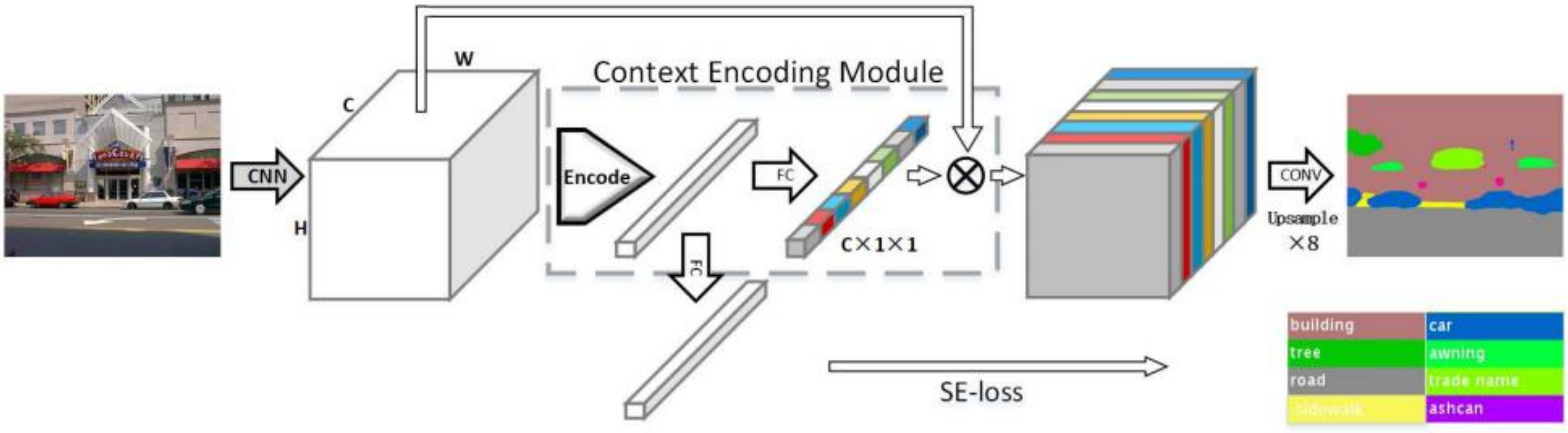


(b) (c) (d)

Lin G, Milan A, Shen C, et al. **RefineNet: Multi-path Refinement Networks for High-Resolution Semantic Segmentation** CVPR 2017

EncNet

- Channel-wise attention with dictionary
- Add another semantic-encoding loss (classification loss) to balance the small objects and large objects



Zhang H, Dana K, Shi J, et al**. Context encoding for semantic segmentation** CVPR 2018.

PSANet

- Pixel-wise attention

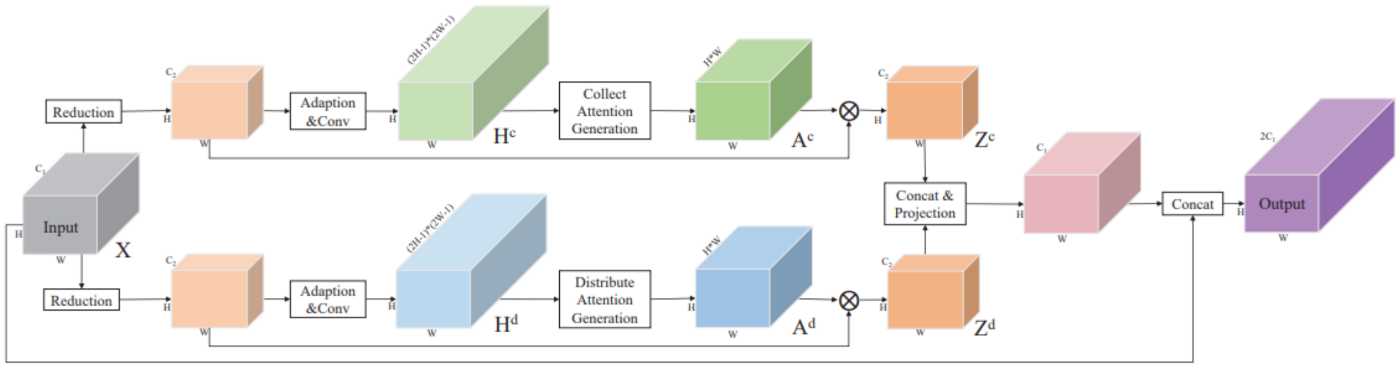


Fig. 2. Architecture of the proposed PSA module.

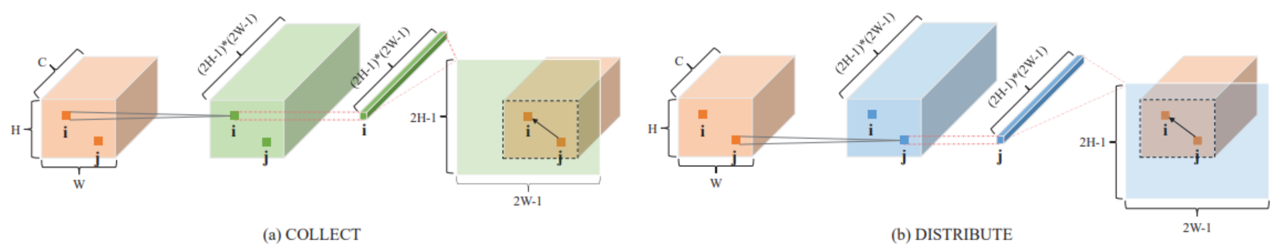
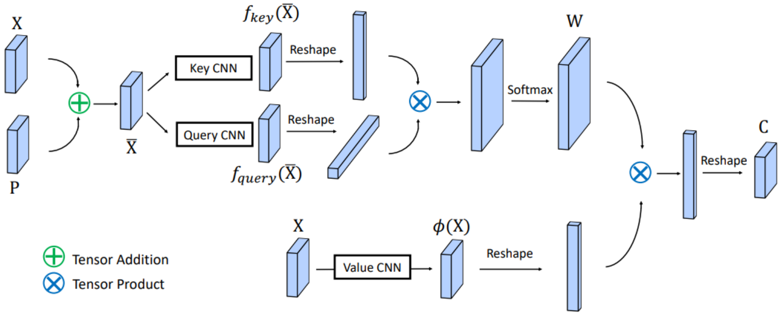


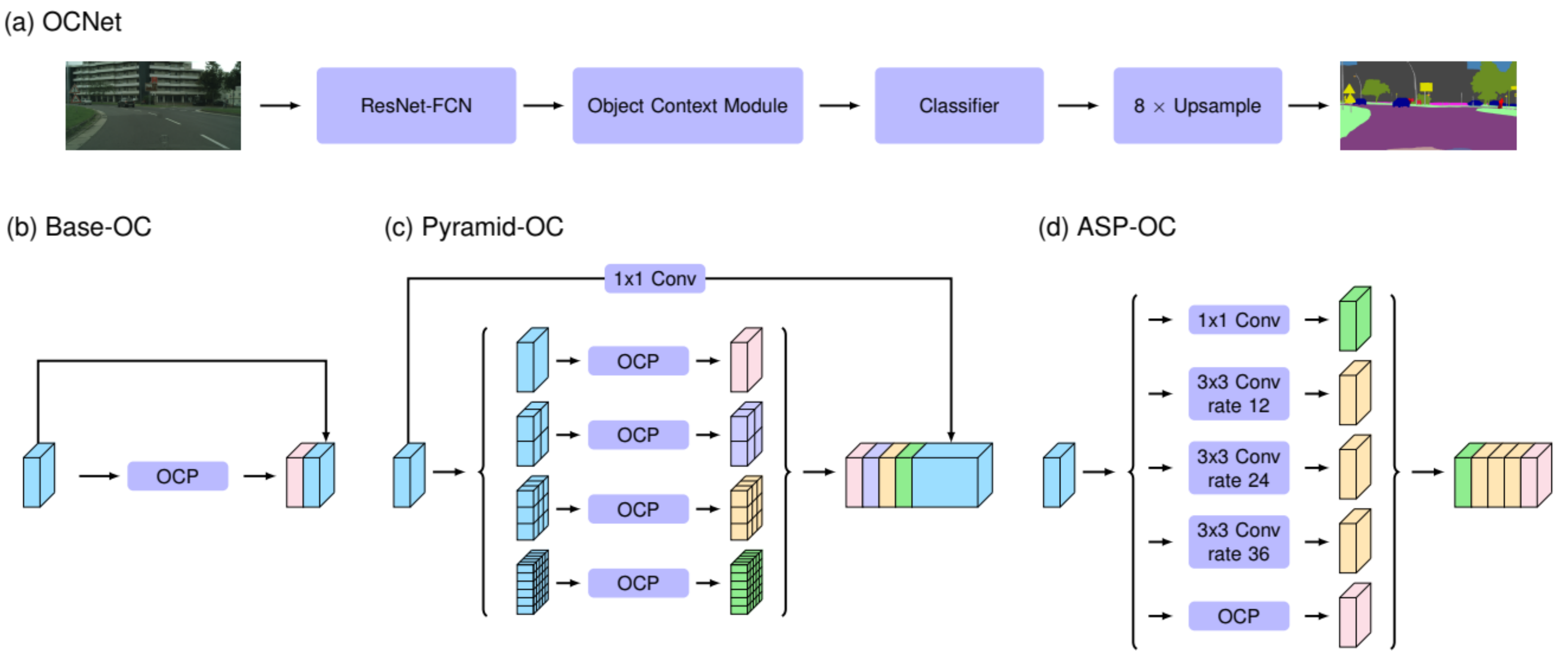
Fig. 3. Illustration of Point-wise Spatial Attention.

Zhao H, Zhang Y, Liu S, et al. **PSANet: Point-wise Spatial Attention Network for Scene Parsing** ECCV 2018

OCNet

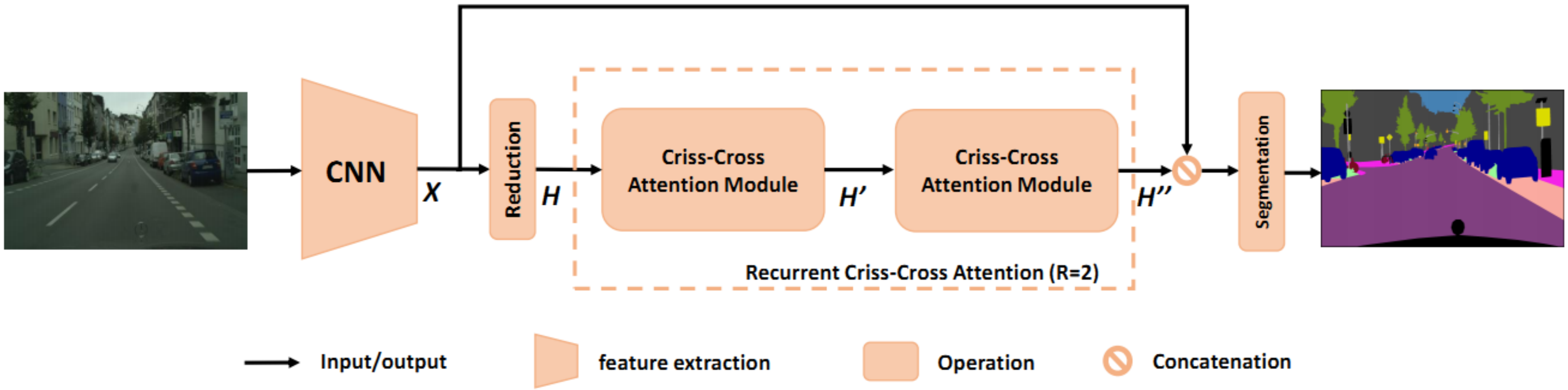
- Object context pooling (self-attention)

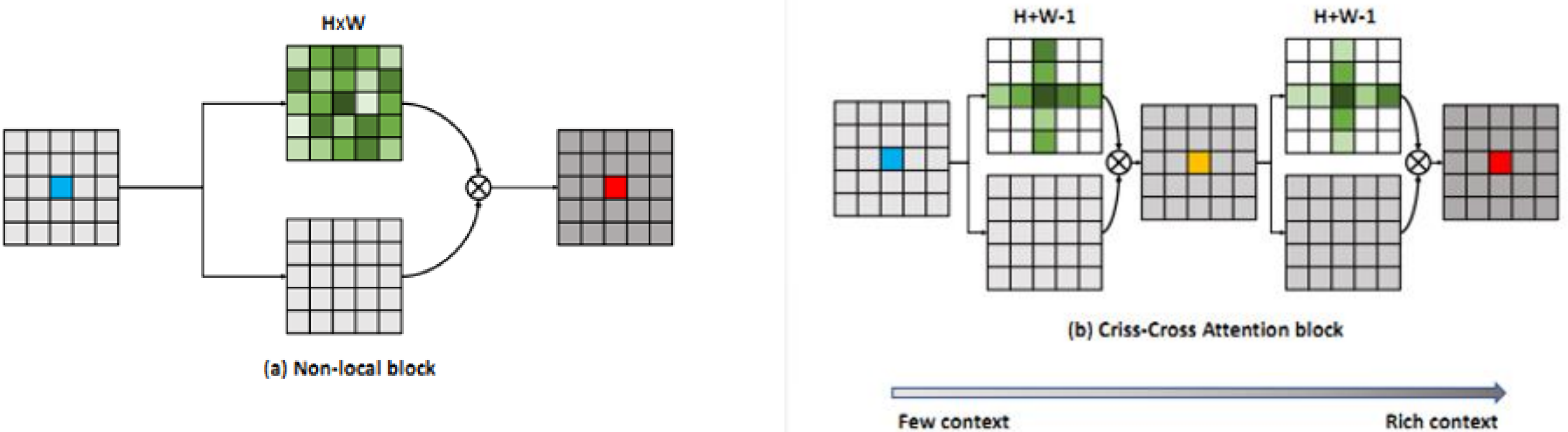




Yuan Y, Wang J. **Ocnet: Object context network for scene parsing** arXiv preprint arXiv:1809.00916, 2018.

CCNet





(a) Non-local block

(b) Criss-Cross Attention block

Huang Z, Wang X, Huang L, et al. **CCNet: Criss-Cross Attention for Semantic Segmentation** arXiv preprint arXiv:1811.11721, 2018.

Datasets

- Pascal VOC 2012
  - 20 classes
  - 10000+ training / 1449 validation

Datasets

- Cityscapes
  - 19 classes
  - 2975 train / 500 validation

Evaluation

- Pixel Acc
  - As a pixel-wise classification problem

- mIoU
  - Calculate IoU for each class among images and average by classes

Results

Table 3: Comparison to state-of-the-art on the test set of Cityscapes.

Method	Conference	Backbone	mIoU (%)
PSPNet [33]†	CVPR2017	ResNet-101	78.4
PSANet [34]†	ECCV2018	ResNet-101	78.6
OCNet†	-	ResNet-101	**80.1**
RefineNet [13]‡	CVPR2017	ResNet-101	73.6
SAC [32]‡	ICCV2017	ResNet-101	78.1
DUC-HDC [23]‡	WACV2018	ResNet-101	77.6
AAF [9]‡	ECCV2018	ResNet-101	79.1
BiSeNet [28]‡	ECCV2018	ResNet-101	78.9
PSANet [34]‡	ECCV2018	ResNet-101	80.1
DFN [29]‡	CVPR2018	ResNet-101	79.3
DSSPN [12]‡	CVPR2018	ResNet-101	77.8
DepthSeg [10]‡	CVPR2018	ResNet-101	78.2
DenseASPP [27]‡	CVPR2018	DenseNet-161	80.6
OCNet‡	-	ResNet-101	**81.7**

† Training with only the train-fine datasets.

‡ Training with both the train-fine and val-fine datasets.

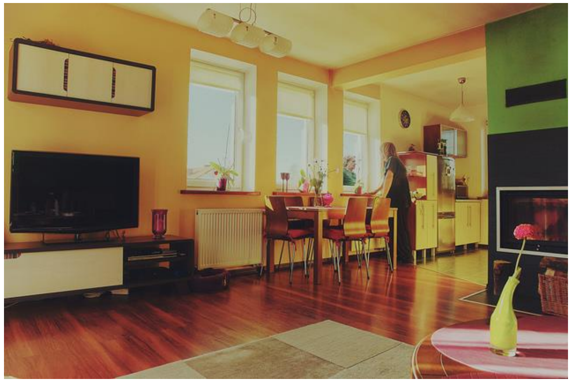
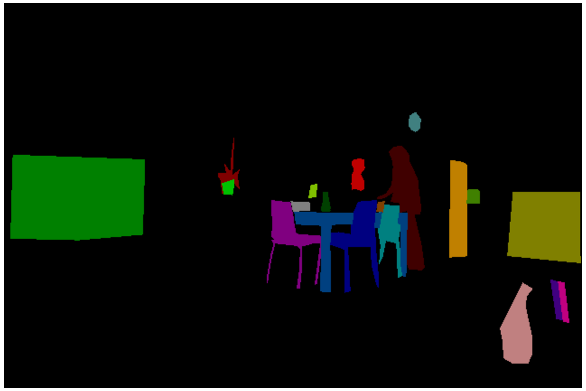
Results

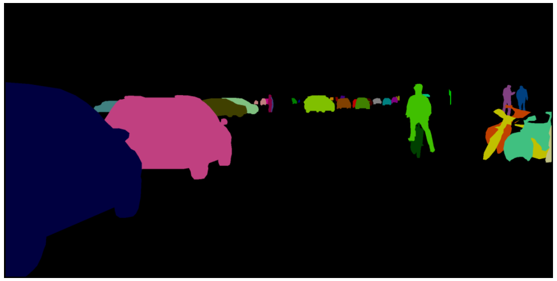
Table 4: Comparison to global pooling (GP), pyramid pooling module (PPM) in PSPNet [33], and atrous spatial pyramid pooling (ASPP) in DeepLabv3 [3] on the validation set of ADE20K.

Method	mIoU (%)	Pixel Acc (%)
ResNet-50 Baseline	34.35 ± 0.01	76.41 ± 0.01
ResNet-50 + GP [16]	41.17 ± 0.38	79.87 ± 0.04
ResNet-50 + PPM [33]	41.34 ± 0.01	79.96 ± 0.01
ResNet-50 + ASPP [3]	42.53 ± 0.03	80.44 ± 0.01
ResNet-50 + Base-OC	40.66 ± 0.26	79.77 ± 0.03
ResNet-50 + Pyramid-OC	42.28 ± 0.08	80.21 ± 0.03
ResNet-50 + ASP-OC	$\mathbf{43.06 \pm 0.01}$	$\mathbf{80.70 \pm 0.01}$

Instance Segmentation

- Detection and segmentation for individual object instances



challenges

- Small objects
  - There are many small objects which are hard to detect and segment

- Annotations are exchangeable
  - Unlike semantic segmentation problems, annotations are hard to directly be applied in the network

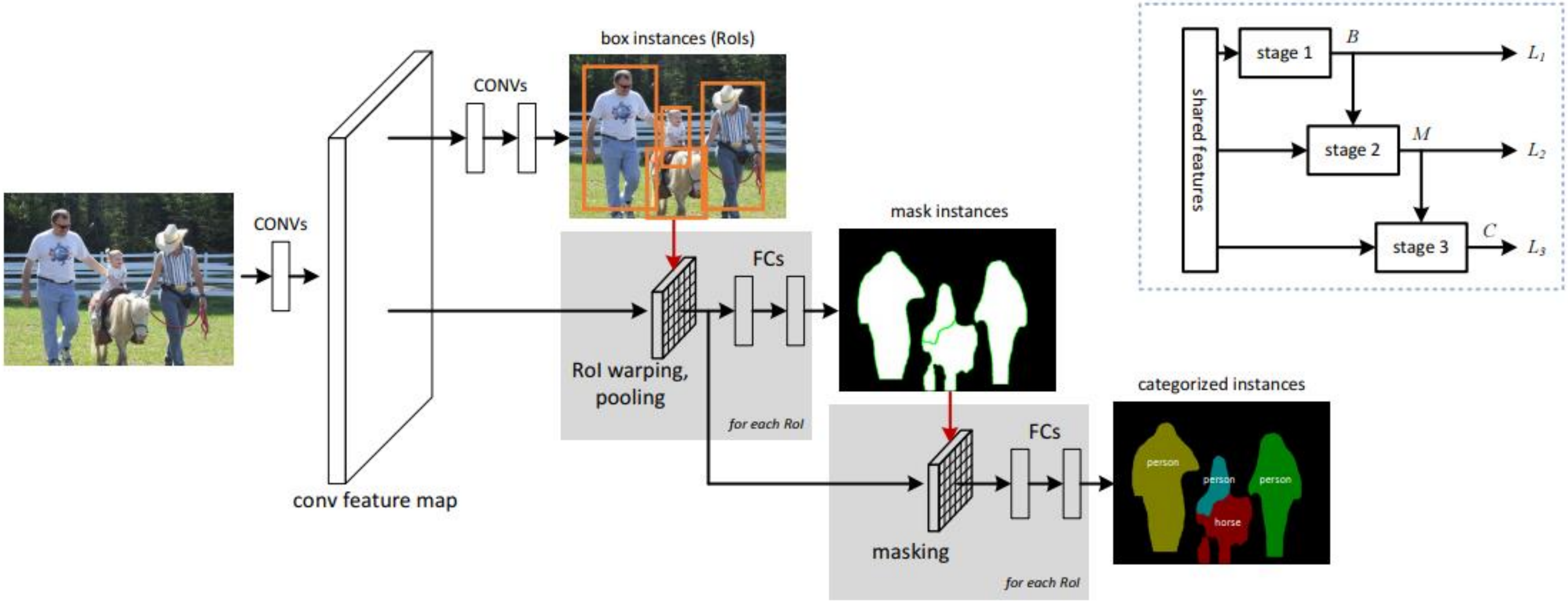
Methods

- Proposal-based: from detection to segmentation
  - Bounding boxes(proposals) from SS/RPN/Faster R-CNN
  - Try to generate mask within the proposal

- Proposal-free: learn to cluster
  - pixel-level featuers / necessary information
  - Clustering pixels

MNC

- Process every proposal



Dai J, He K, Sun J. **Instance-aware semantic segmentation via multi-task network cascades** CVPR 2016

Instance sensitive FCN

- Position sensitive maps

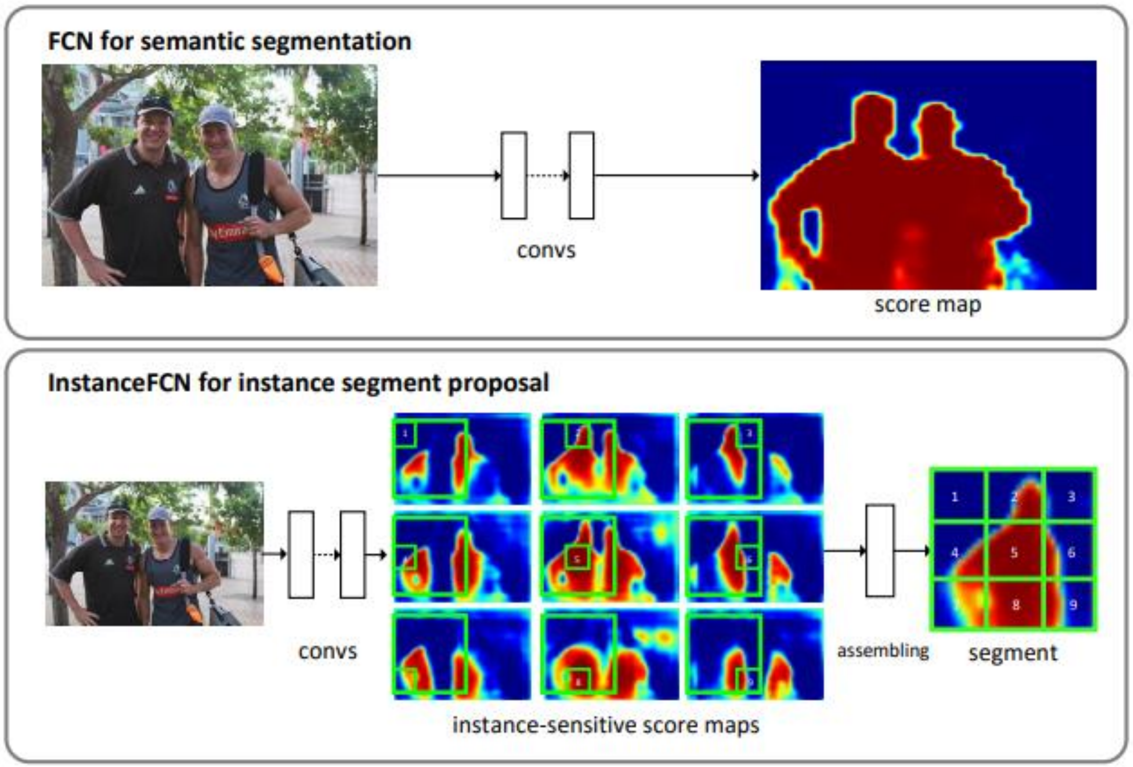
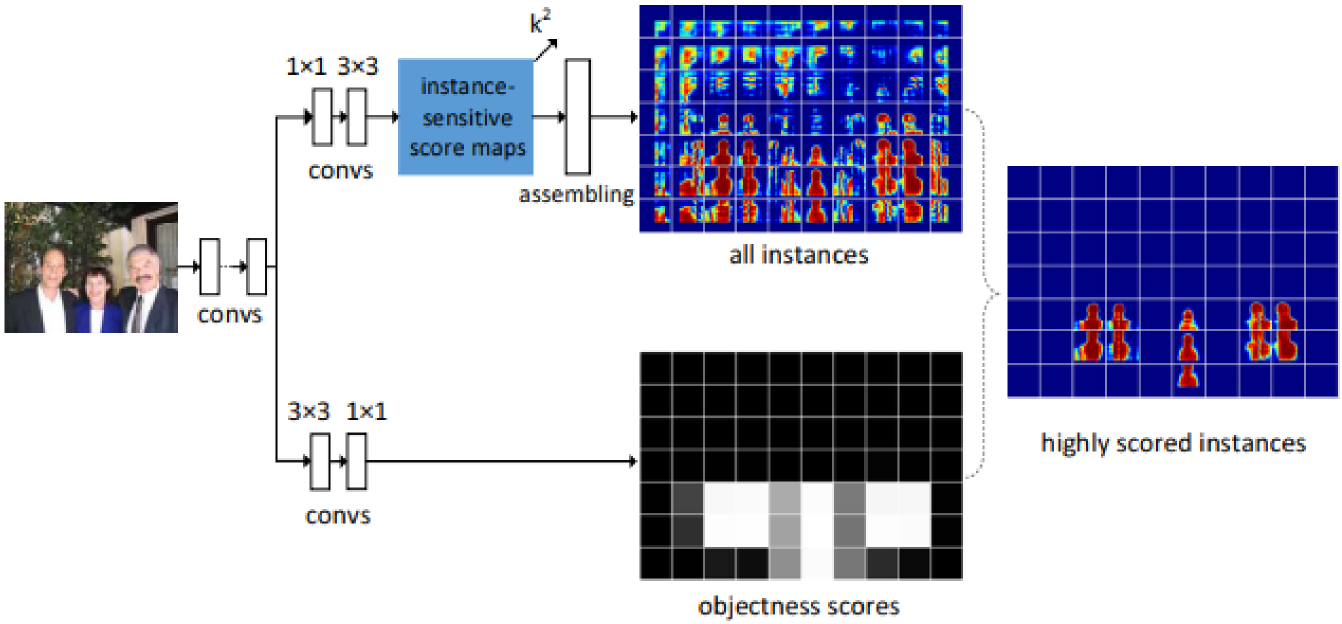


Figure 1. Methodological comparisons between: (**top**) FCN [1] for semantic segmentation; (**bottom**) our InstanceFCN for instance segment proposal.

Dai J, He K, Li Y, et al. **Instance-sensitive fully convolutional networks** ECCV 2016

Instance sensitive FCN

- Pooling within fix-size window



Dai J, He K, Li Y, et al. **Instance-sensitive fully convolutional networks** ECCV 2016

FCIS

- Enhanced position-sensitive map

Figure 2. Instance segmentation and classification results (of "person" category) of different ROIs. The score maps are shared by different ROIs and both sub-tasks. The red dot indicates one pixel having different semantics in different ROIs.

Li Y, Qi H, Dai J, et al. **Fully Convolutional Instance-Aware Semantic Segmentation** CVPR 2017

FCIS

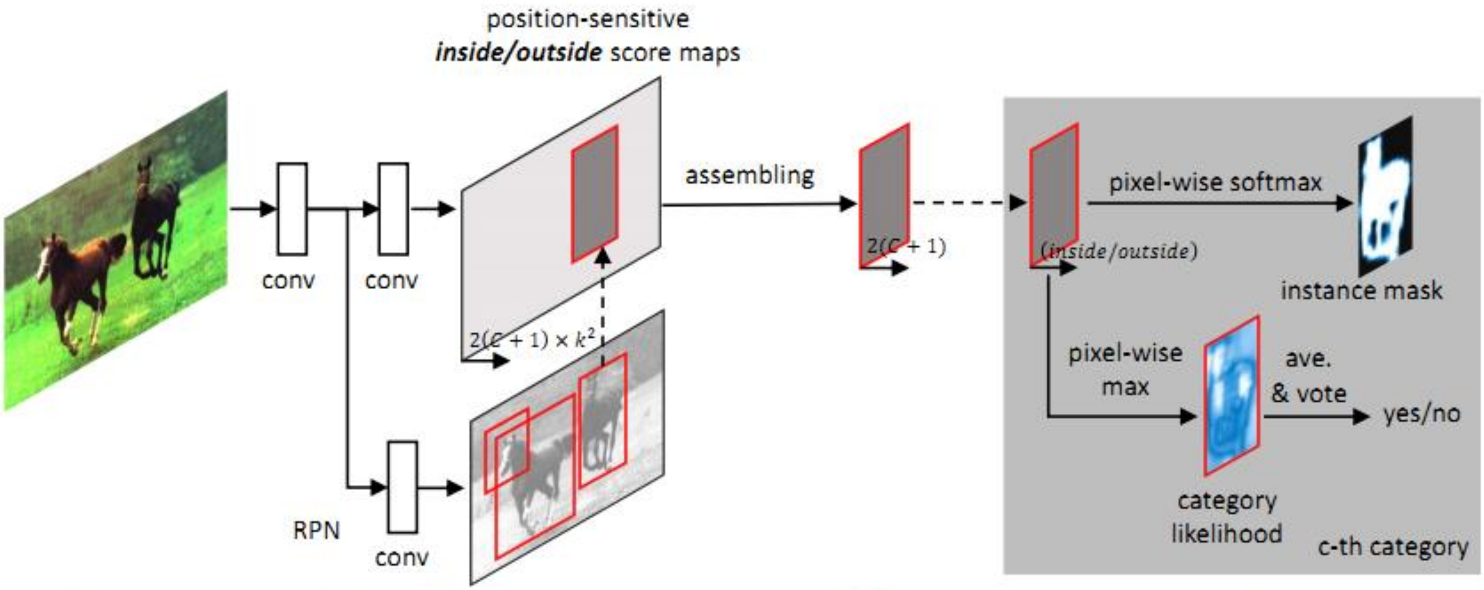


Figure 3. Overall architecture of FCIS. A region proposal network (RPN) [34] shares the convolutional feature maps with FCIS. The proposed region-of-interests (ROIs) are applied on the score maps for joint object segmentation and detection. The learnable weight layers are fully convolutional and computed on the whole image. The per-ROI computation cost is negligible.

Li Y, Qi H, Dai J, et al. **Fully Convolutional Instance-Aware Semantic Segmentation** CVPR 2017

Mask R-CNN

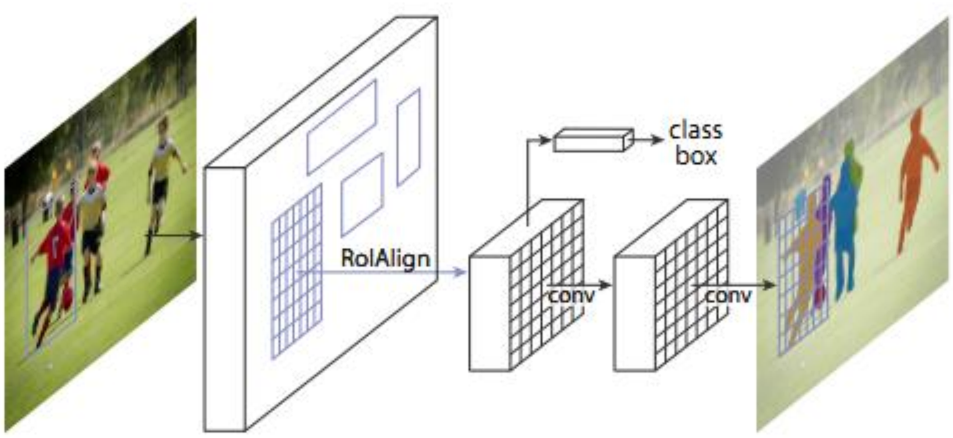
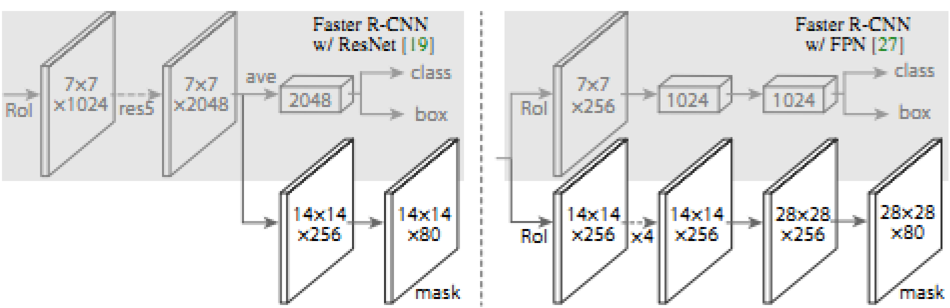


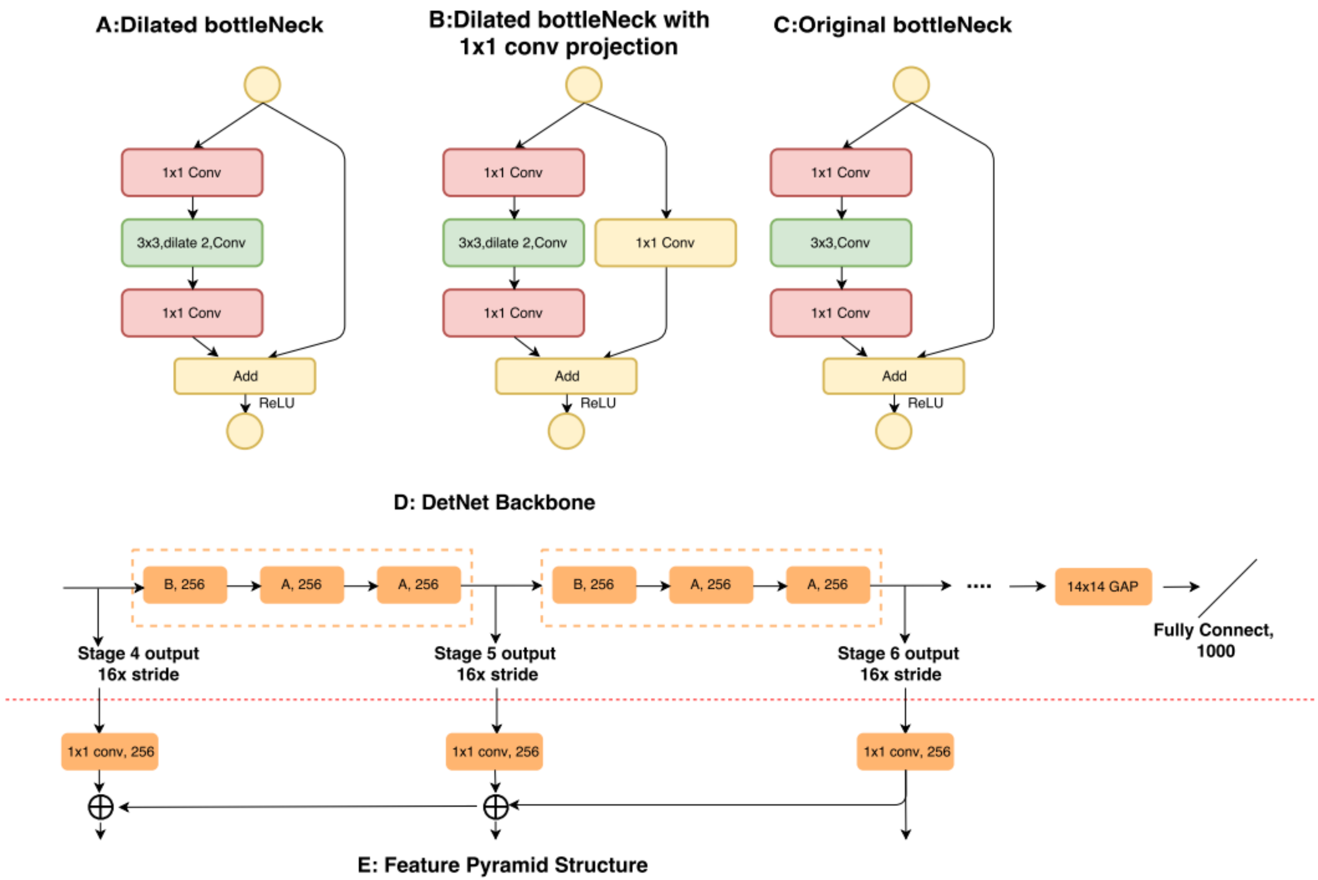
Figure 1. The **Mask R-CNN** framework for instance segmentation.



He K, Gkioxari G, Dollár P, et al. **Mask r-cnn** ICCV 2017

DetNet

- Deeper: more stages
- Keep spacial information



Li Z, Peng C, Yu G, et al. **Detnet: Design backbone for object detection** ECCV 2018

PANet

- Path augmentation
- Adaptive feature pooling
- Heavier mask head

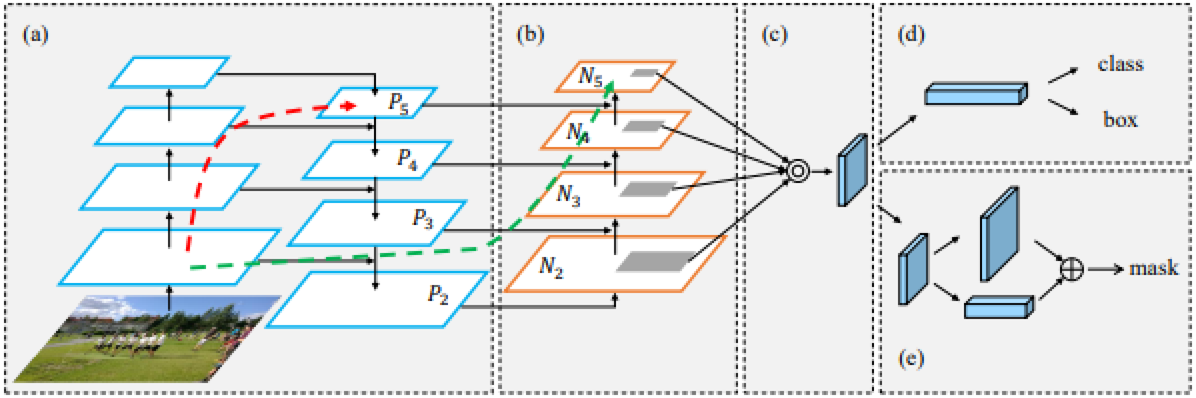


Figure 1. Illustration of our framework. (a) FPN backbone. (b) Bottom-up path augmentation. (c) Adaptive feature pooling. (d) Box branch. (e) Fully-connected fusion. Note that we omit channel dimension of feature maps in (a) and (b) for brevity.

Liu S, Qi L, Qin H, et al. **Path aggregation network for instance segmentation** CVPR 2018

Proposal-free network

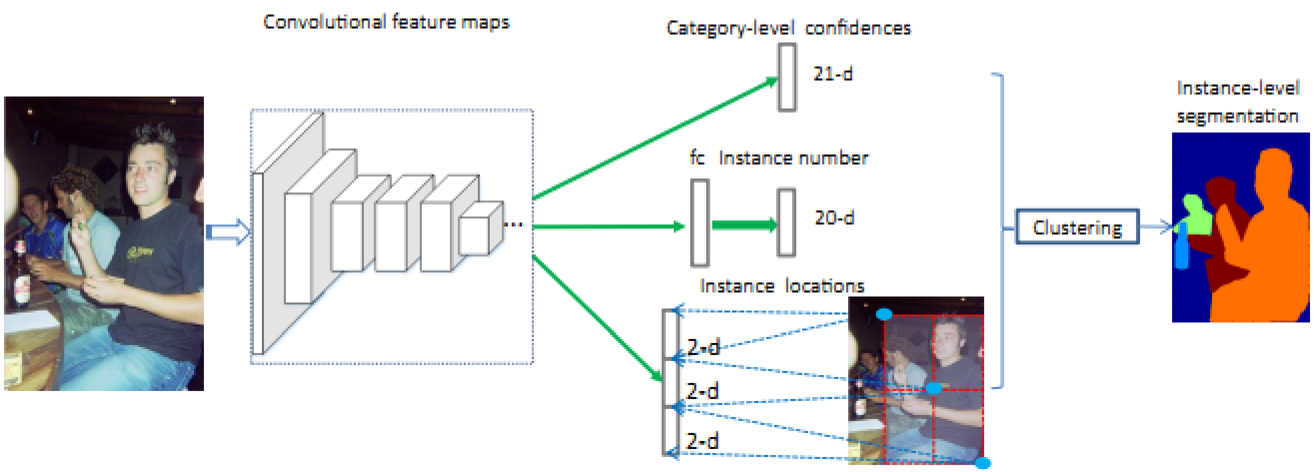
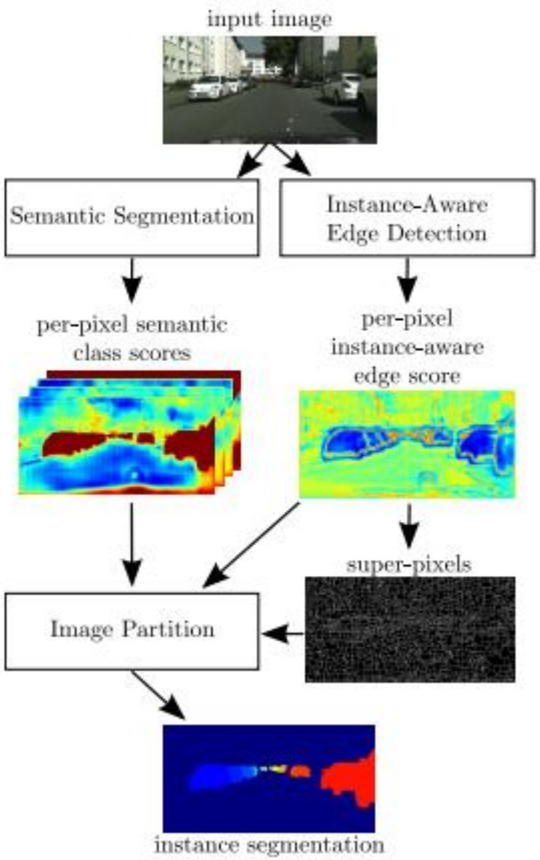


Fig. 2. The proposal-free network overview. Our network predicts the instance numbers of all categories and the pixel-level information that includes the category-level confidences for each pixel and the coordinates of the instance bounding box each pixel belongs to. The instance location prediction for each pixel involves the coordinates of center, top-left corner and bottom-right corner of the object instance that a specific pixel belongs to. Any off-the-self clustering method can be utilized to generate ultimate instance-level segmentation results.

Liang X, Wei Y, Shen X, et al. **Proposal-free network for instance-level object segmentation** arXiv preprint arXiv:1509.02636, 2015.

InstanceCut



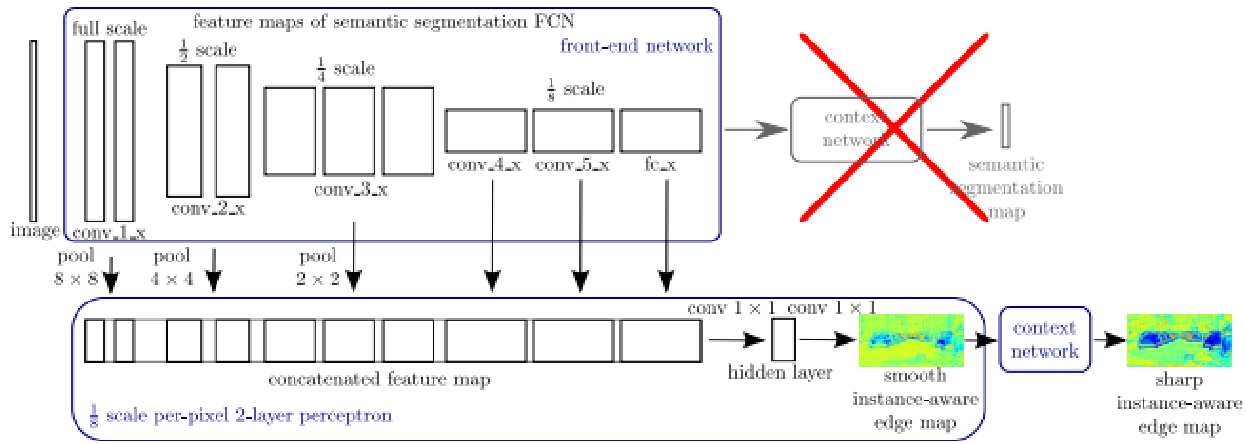


Figure 3: **Instance-aware edge detection block.** The semantic segmentation FCN is the front-end part of the network [52] trained for semantic segmentation on the same dataset. Its intermediate feature maps are downsampled, according to the size of the smallest feature map, by a max-pooling operation with an appropriate stride. The concatenation of the downsampled maps is used as a feature representation for a per-pixel 2-layer perceptron. The output of the perceptron is refined by a context network of Dilation10 [52] architecture.

Kirillov A, Levinkov E, Andres B, et al. **Instancecut: from edges to instances with multicut** CVPR. 2017

SGN

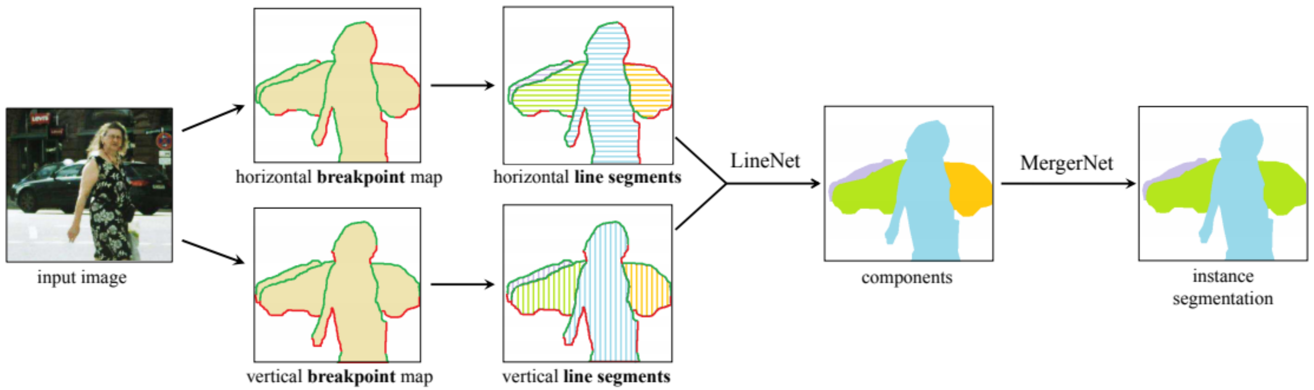
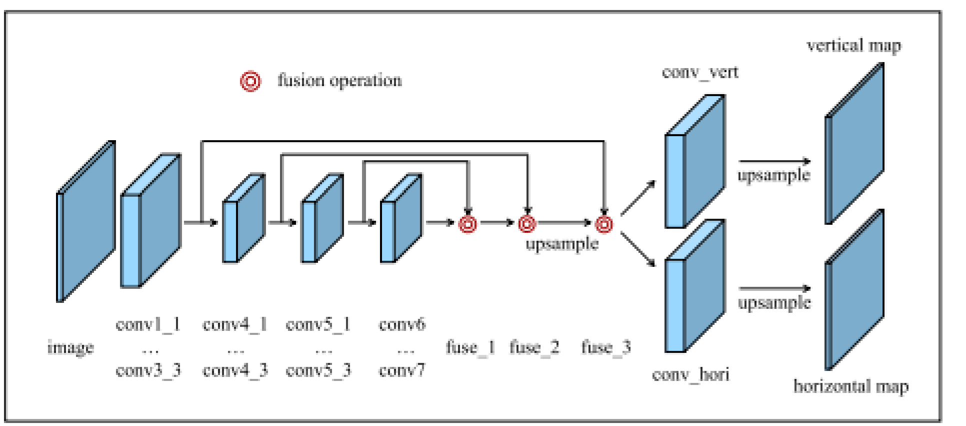


Figure 1. **Sequential Grouping Networks (SGN):** We first predict breakpoints. LineNet groups them into connected components, which are finally composed by the MergerNet to form our final instances.

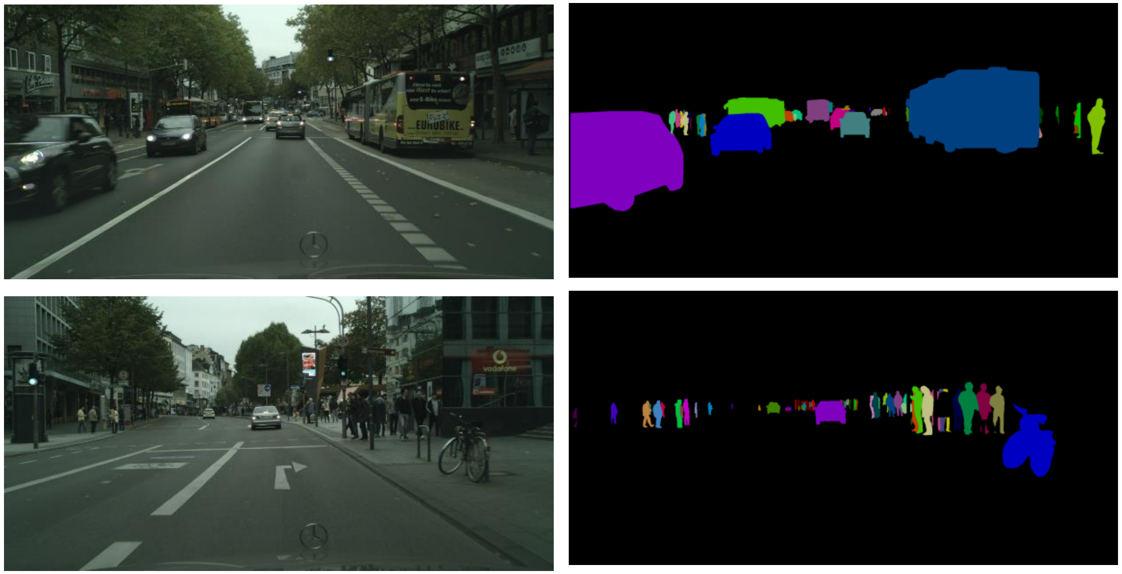


(a)

Liu S, Jia J, Fidler S, et al. **Sgn: Sequential grouping networks for instance segmentation** ICCV 2017.

dataset

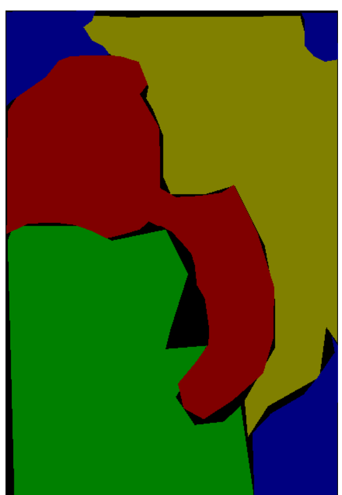
- Cityscapes
  - 9 classes with instance annotations



dataset

- COCO
  - 81 classes



Evaluation

- AP50
  - If IoU is larger than 0.5 with ground truth, we take them as positive
- mAP:
  - Same as detection

Performance

Models	Backbone	mAP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
MNC [39]	ResNet-101	24.6	44.3	24.8	4.7	25.9	43.6
FCIS [40] + OHEM [41]	ResNet-101-C5-dilated	29.2	49.5	-	7.1	31.3	50.0
FCIS+++ [40] +OHEM	ResNet-101-C5-dilated	33.6	54.5	-	-	-	-
Mask R-CNN [33]	ResNet-101	35.7	58.0	37.8	15.5	38.1	52.4
Mask R-CNN	**DetNet-59**	**37.1**	60.0	39.6	18.6	39.0	51.3

Table 8. Comparison of instance segmentation results between our approach and other state-of-the-art on MSCOCO test-dev datasets. Benefit from DetNet-59, we achieve a new state-of-the-art on instance segmentation task.

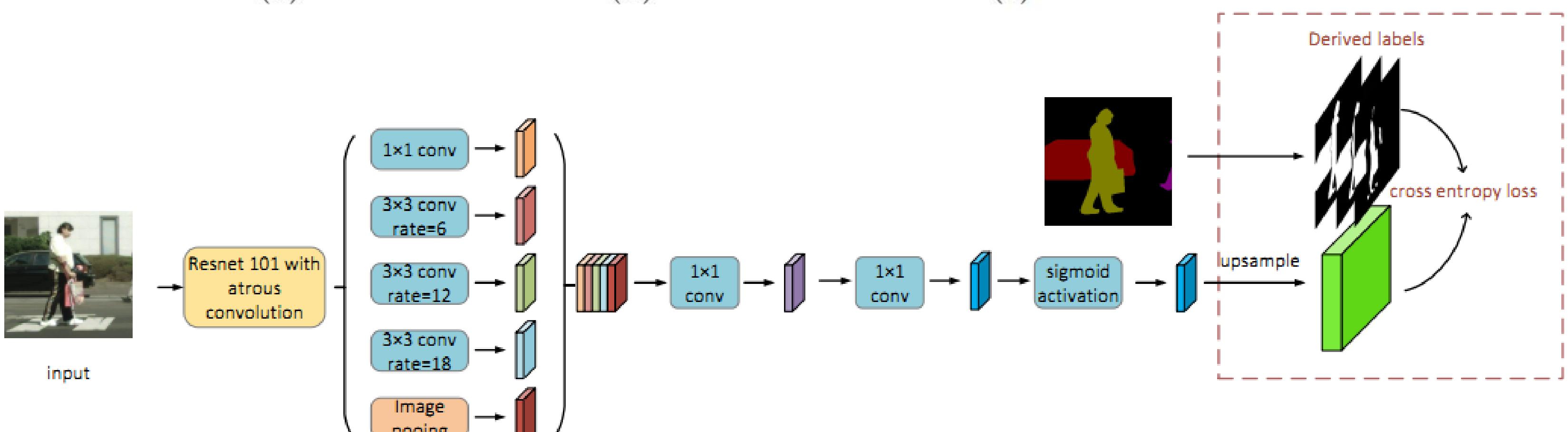
Method	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{bb}_{S}	AP^{bb}_{M}	AP^{bb}_{L}	Backbone
Champion 2016 [27]	41.6	62.3	45.6	24.0	43.9	55.2	2×ResNet-101 + 3×Inception-ResNet-v2
RentinaNet [36]	39.1	59.1	42.3	21.8	42.7	50.2	ResNet-101
Mask R-CNN [21]+FPN [35]	38.2	60.3	41.7	20.1	41.1	50.2	ResNet-101
Mask R-CNN [21]+FPN [35]	39.8	62.3	43.4	22.1	43.2	51.2	ResNeXt-101
PANet / PANet [ms-train]	41.2 / 42.5	60.4 / 62.3	44.4 / 46.4	22.7 / 26.3	44.0 / 47.0	54.6 / 52.3	ResNet-50
PANet / PANet [ms-train]	45.0 / **47.4**	65.0 / **67.2**	48.6 / **51.8**	25.4 / **30.1**	48.6 / **51.7**	59.1 / **60.0**	ResNeXt-101

Table 2. Comparison among PANet, winner of COCO 2016 object detection challenge, RentinaNet and Mask R-CNN on COCO *test-dev* subset in terms of box AP, where the latter three are baselines.

Graph merge

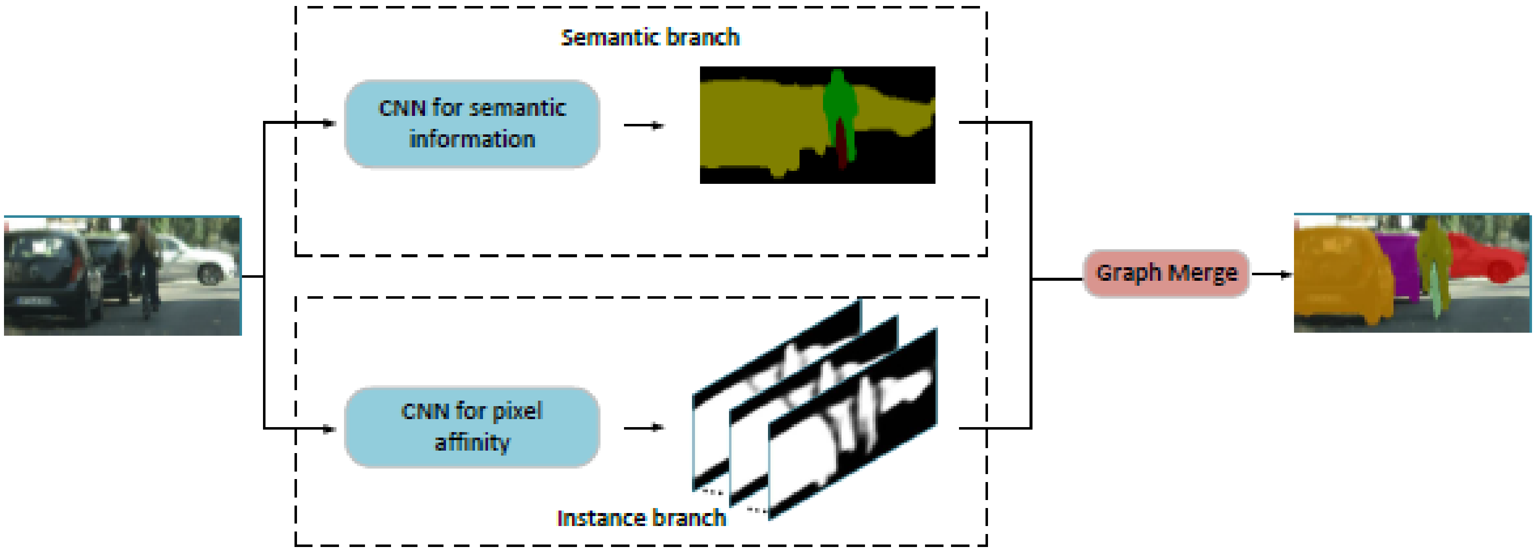
- Pixel affinity
  - If a pair of pixels belongs to a same instance
  - Predict by FCN

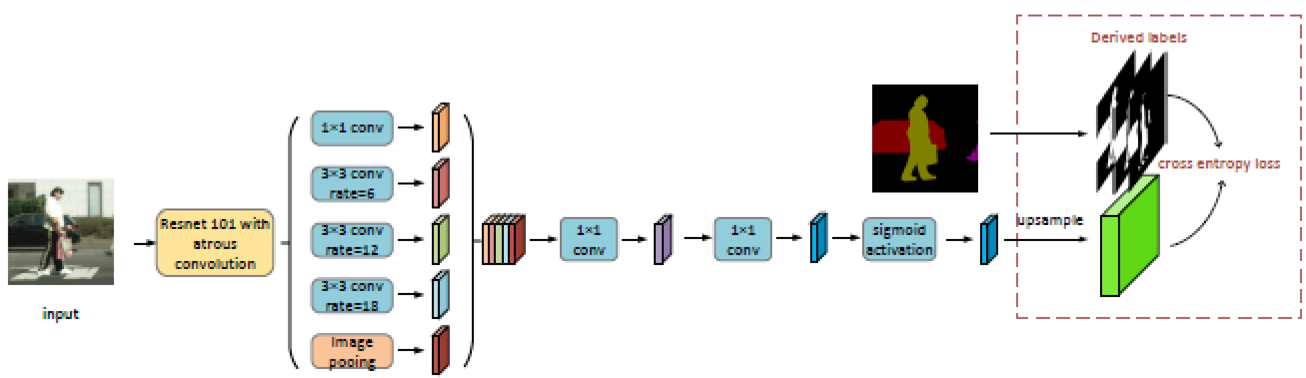
(a) (b) (c)



Liu Y, Yang S, Li B, et al. **Affinity Derivation and Graph Merge for Instance Segmentation** ECCV 2018

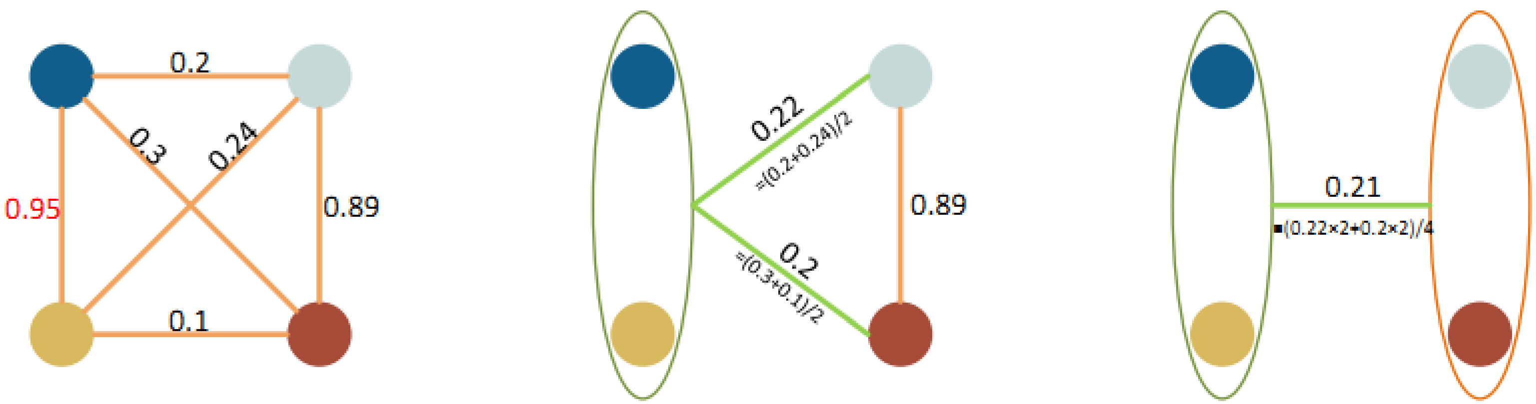
Network Structure

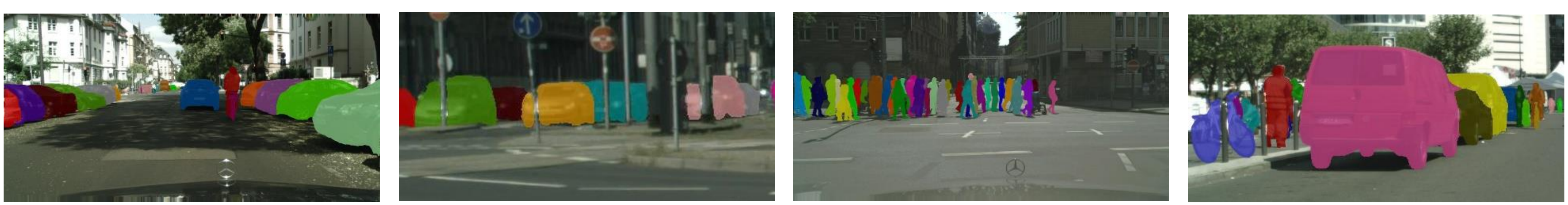




Graph merge

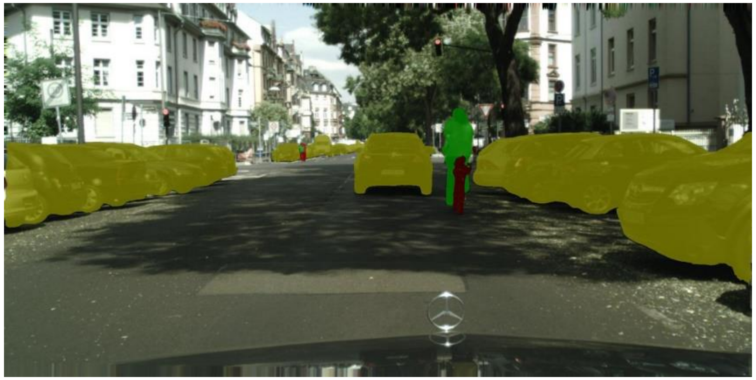
- Graph merge algorithm:
  - Regard the whole image as a graph
  - Pixels as vertexes and affinities as edges
  - Find the largest edge in the graph and merge two pixels together

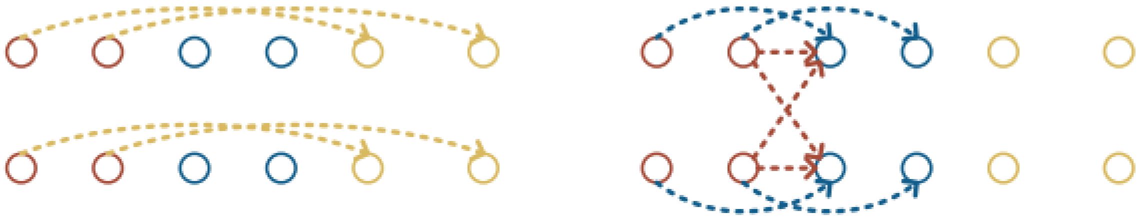




Implementation details

- Excluding Backgrounds (generating ‘rois’ and resize)
- Affinity Refinement based on Semantic class
- Forcing Local Merge
- Semantic Class Partition





Results

Table 2. Graph Merge Strategy: we test for our graph merge strategies for our algorithm including **PAR**: Pixel Affinity Refinement, **RR**: Resizing ROIs, **FLM**: Forcing Local Merge and **SCP**: Semantic Class Partition. Note that 2 and 4 in FLM represent the merge window size, default as 1.

PAR	RR	FLM	SCP	AP
				18.9
✓				22.8
✓	✓			28.7
✓	✓	2		29.2
✓	✓	4		27.5
✓	✓	2	✓	30.7

Table 3. Additional inference strategies: We test for additional inference strategies for our algorithm including **Semantic OS**: output stride for semantic branch, **Instance OS**: output stride for instance branch **SHF**: Semantic horizontal flip inference,**IHF**: Instance horizontal flip inference and **SCR**: Semantic Class Refinement. We also list several results from other methods for comparison.

Methods	Semantic OS	Instance OS	SHF	IHF	SCR	AP
DWT[3]						21.2
SGN[37]						29.2
Mask RCNN[23]						31.5
Ours	16	16				30.7
	8	16				31.2
	16	8				31.2
	8	8				32.1
	8	8	✓			32.8
	8	8		✓		32.6
	8	8	✓	✓		33.5
	8	8	✓	✓	✓	34.1

Results on Cityscapes test set

Table 1. Instance segmentation performance on the Cityscapes *test* set. All results listed are trained only with Cityscapes.

Methods	person	rider	car	trunk	bus	train	mcycle	bicycle	AP 50%	AP
InstanceCut[29]	10.0	8.0	23.7	14.0	19.5	15.2	9.3	4.7	27.9	13.0
SAIS[22]	14.6	12.9	35.7	16.0	23.2	19.0	10.3	7.8	36.7	17.4
DWT[3]	15.5	14.1	31.5	22.5	27.0	22.9	13.9	8.0	35.3	19.4
DIN[2]	16.5	16.7	25.7	20.6	30.0	23.4	17.1	10.1	38.8	20.0
SGN[37]	21.8	20.1	39.4	**24.8**	33.2	**30.8**	17.7	12.4	44.9	25.0
Mask RCNN[23]	30.5	23.7	**46.9**	22.8	32.2	18.6	**19.1**	**16.0**	**49.9**	26.2
Ours	**31.5**	**25.2**	42.3	21.8	**37.2**	28.9	18.8	12.8	45.6	**27.3**