The progress of the reading plan:

Index	Semantic Seg	All
15	14	50

Paper Information

Paper Title :

Not All Pixels Are Equal: Difficulty-Aware Semantic Segmentation via Deep Layer Cascade

Conference :

CVPR 2017

Authors and Institutions

Authors

- Xiaoxiao Li 1
- Ziwei Liu 1
- Ping Luo 2;1
- Chen Change Loy 1;2
- Xiaoou Tang 1;2

Institutions

- 1 Department of Information Engineering, The Chinese University of Hong Kong
- 2 Shenzhen Key Lab of Comp. Vis. & Pat. Rec., Shenzhen Institutes of Advanced Technology, CAS, China

Official Codes

https://github.com/liuziwei7/region-conv

Some articles to comprehend this paper

- 图像分割“Not All Pixels Are Equal: Difficulty-Aware Semantic Segmentation via Deep Layer Cascade”

Network Structure

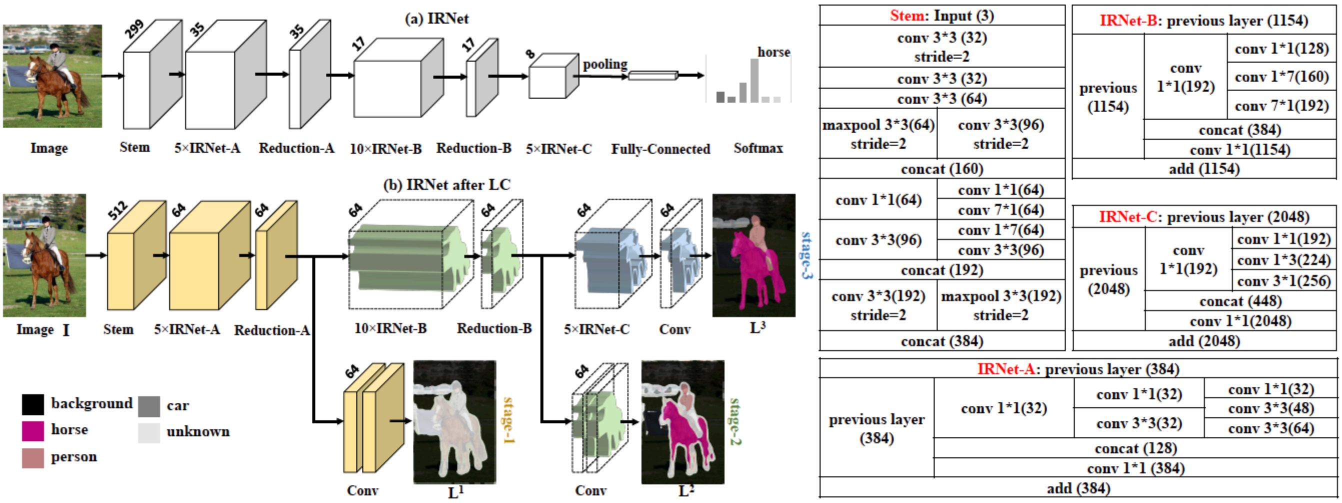


Figure 2: (a) depicts the Inception-ResNet-v2 (IRNet) for classification task. (b) is the architecture of Layer Cascade IRNet (IRNet-LC). The tables at the right show the structure of IRNet.

Note

How to divide different difficulties?

- Easy Set(ES): correct classified confidence > 95%
- Hard Set(HS): misclassified confidence > 95%
- Moderate Set(MS): covers pixels that have classification scores smaller than 0.95

Network

IRNet is turned into LC by dividing its different components as different stages.

In addition, we append two convolutional layers and a softmax loss at the end of each stage.

In this case, the original IRNet with one loss function develops into multiple stages, where each stage has its own loss function. So multi-stage training is applied.

Details about stage-1:

In the first stage, given a 3x512x512 image I, stage-1 predicts a 21x64x64 segmentation label map L1, where each 21x1 column vector, denoted as $L_i^1 \in R^{21x1}$, indicates the probabilities (confidence scores) of the i-th pixel belonging to 21 object categories in VOC respectively.

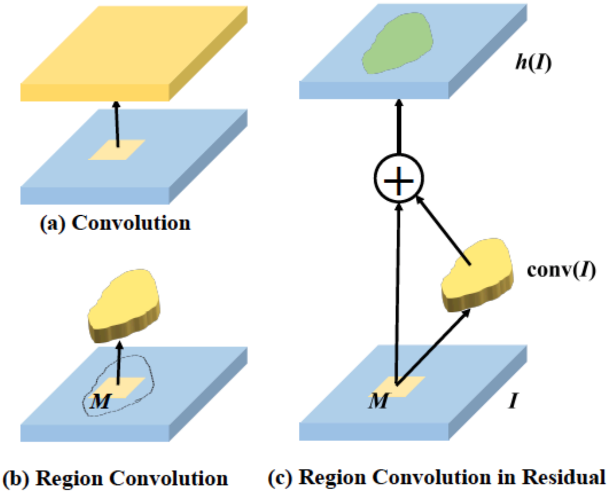
Using the softmax function to let $\sum_{j=1}^{21} L_{ij}^1 = 1$.

If the maximum score of the i-th pixel, $l_i^1 = max(L_i^1)$ and $l_i^1 \in \{L_{ij}^1 | j = 1 \dots 21\}$, is larger than a threshold

ρ, we accept its prediction and do not propagate it forward to stage-2.

Region Convolution

As presented above, stage-2 and -3 only calculate convolutions on those pixels that have been propagated forward.



(a) Convolution

(b) Region Convolution

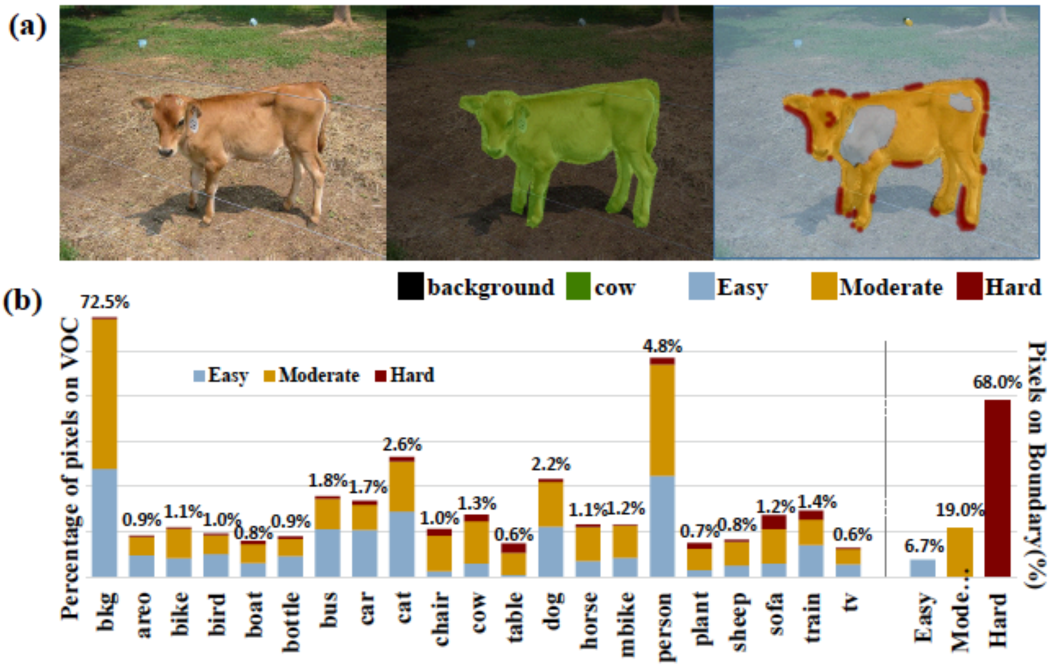
(c) Region Convolution in Residual

Figure 3: (a) shows the conventional convolution that operates on an entire image. (b) is region convolution (RC) where filters only convolve irregular region of interest denoted as M. Values of the other region are set as zeros. (c) illustrates RC in a residual module. **Best viewed in color.**

So region convolution is used here.

Five questions about this paper:

1. [Problem Definition / Motivation] What problem is this paper trying to solve?



Pixels in each classes are divided by three level of difficulties. And ES (easy set) occupies at least 30% pixels of most objects. The right one reveals that 70% pixels in HS are located at object boundaries, which have large ambiguity.

So we should focus on the hard set more.

2. [Contribution / Method] What’s new in this paper? / How does this paper solve the above problems?

Contribution:

- Layer Cascade (LC) approach is proposed to significantly **reduce computations** while improving the segmentation accuracies.
- LC’s properties can be easily applied to many recent advanced network structures.

3. Details about the experiment

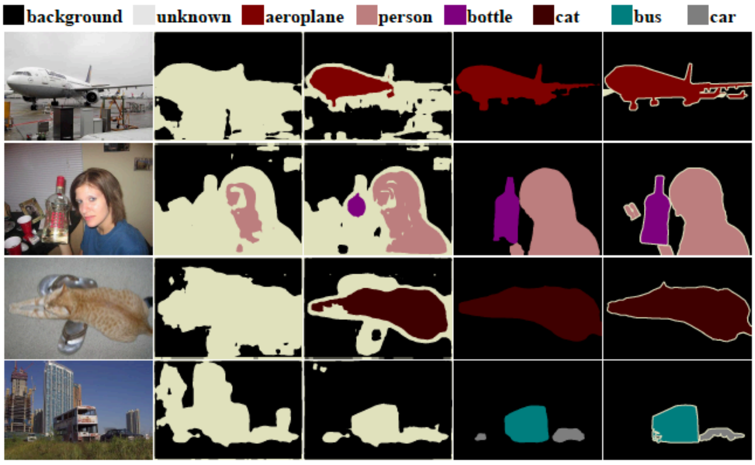
3.1 Which Datasets are used?

- PASCAL VOC 2012
- Cityscapes

3.2 How is the experiment set up?

Cascade Training:

Similar to the previous step, all stages are trained jointly, but different stages minimize their pixel-wise softmax losses with respect to different regions.



(a) input image (b) stage-1 (c) stage-2 (d) stage-3 (e) ground truth

Figure 4: Visualization of different stages' outputs in VOC12 dataset. **Best viewed in color.**

3.3 What's the evaluation metric?

mIoU

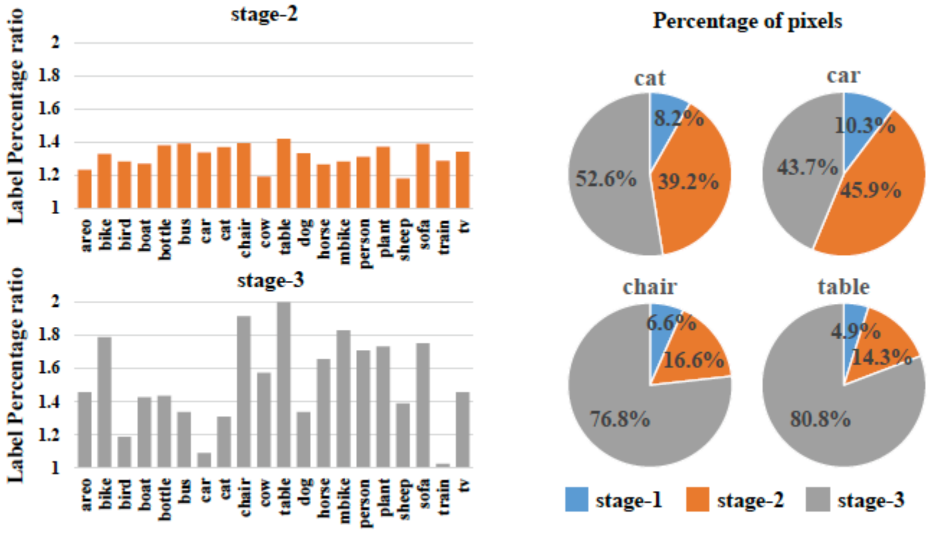
3.4 Ablation Study and Results

Table 1: Ablation study on probability thresholds ρ.

ρ	1	0.995	0.985	0.970	0.950	0.930	0.900	0.800
stage-1 (%)	0	15	23	30	35	35	44	56
stage-2 (%)	0	14	29	31	30	41	31	29
mIoU (%)	72.70	73.56	**73.91**	73.63	73.03	72.53	71.20	66.95

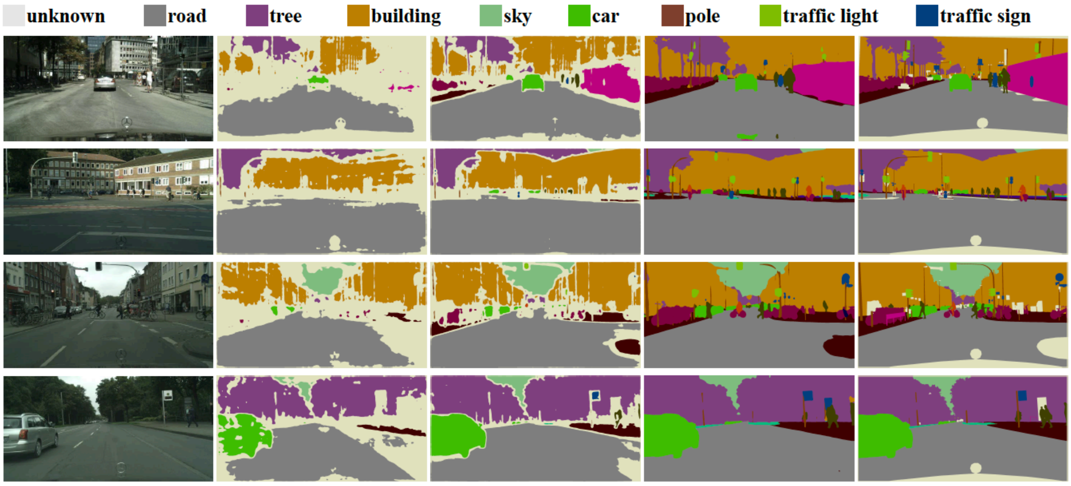
Table 2: Comparisons with related methods.

	mIoU(%)
IRNet [32]	72.22
DSN [17]	72.70
DSN [17] + Dropout [30]	72.63
Model Cascade (MC)	44.20
Layer Cascade (LC)	**73.91**



(a) (b)

Figure 5: (a) is the change of label distribution in stage-2 and -3. (b) shows the percentage of pixels that are classified in different stages.



(a) input image (b) stage-1 (c) stage-2 (d) stage-3 (e) ground truth

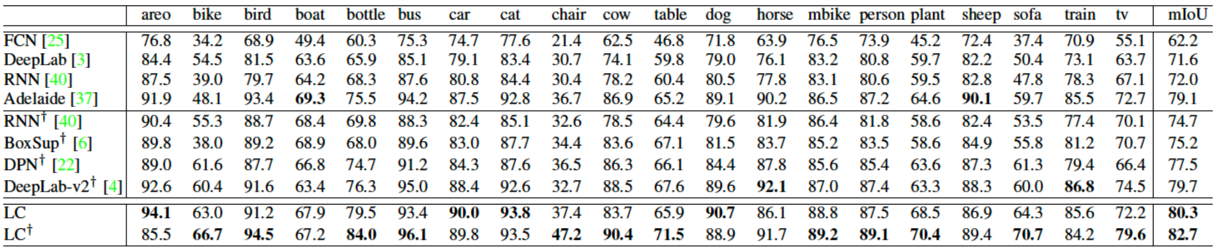
Figure 6: Visualization of different stages' outputs in Cityscapes dataset. **Best viewed in color.**

Table 3: A comparison of performance and speed of Layer Cascade (LC) against existing methods.

	mIoU	ms	FPS
DeepLab-v2 [4]	70.42	140.0	7.1
SegNet [1]	59.90	69.0	14.6
LC	**73.91**	65.1	14.7
LC (fast)	66.95	42.5	**23.6**

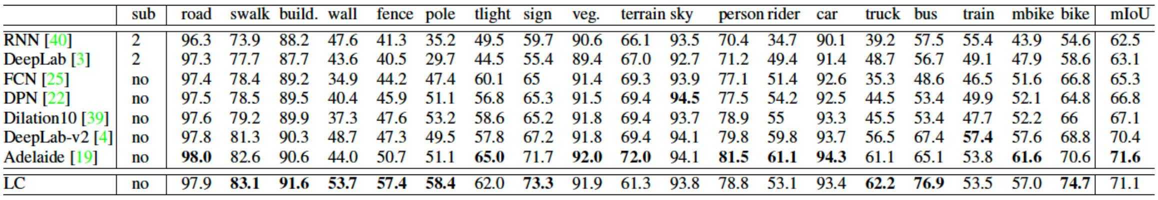
3.5 What is the ranking of the experiment results?

Table 4: Per-class results on VOC12 *test set*. Approaches pre-trained on COCO [20] are marked with †.



	areo	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mIoU
FCN [25]	76.8	34.2	68.9	49.4	60.3	75.3	74.7	77.6	21.4	62.5	46.8	71.8	63.9	76.5	73.9	45.2	72.4	37.4	70.9	55.1	62.2
DeepLab [3]	84.4	54.5	81.5	63.6	65.9	85.1	79.1	83.4	30.7	74.1	59.8	79.0	76.1	83.2	80.8	59.7	82.2	50.4	73.1	63.7	71.6
RNN [40]	87.5	39.0	79.7	64.2	68.3	87.6	80.8	84.4	30.4	78.2	60.4	80.5	77.8	83.1	80.6	59.5	82.8	47.8	78.3	67.1	72.0
Adelaide [37]	91.9	48.1	93.4	**69.3**	75.5	94.2	87.5	92.8	36.7	86.9	65.2	89.1	90.2	86.5	87.2	64.6	**90.1**	59.7	85.5	72.7	79.1
RNN† [40]	90.4	55.3	88.7	68.4	69.8	88.3	82.4	85.1	32.6	78.5	64.4	79.6	81.9	86.4	81.8	58.6	82.4	53.5	77.4	70.1	74.7
BoxSup† [6]	89.8	38.0	89.2	68.9	68.0	89.6	83.0	87.7	34.4	83.6	67.1	81.5	83.7	85.2	83.5	58.6	84.9	55.8	81.2	70.7	75.2
DPN† [22]	89.0	61.6	87.7	66.8	74.7	91.2	84.3	87.6	36.5	86.3	66.1	84.4	87.8	85.6	85.4	63.6	87.3	61.3	79.4	66.4	77.5
DeepLab-v2† [4]	92.6	60.4	91.6	63.4	76.3	95.0	88.4	92.6	32.7	88.5	67.6	89.6	**92.1**	87.0	87.4	63.3	88.3	60.0	**86.8**	74.5	79.7
LC	**94.1**	63.0	91.2	67.9	79.5	93.4	**90.0**	**93.8**	37.4	83.7	65.9	**90.7**	86.1	88.8	87.5	68.5	86.9	64.3	85.6	72.2	**80.3**
LC†	85.5	**66.7**	**94.5**	67.2	**84.0**	**96.1**	89.8	93.5	**47.2**	**90.4**	**71.5**	88.9	91.7	**89.2**	**89.1**	**70.4**	89.4	**70.7**	84.2	**79.6**	**82.7**

Table 5: Per-class results on Cityscapes *test set*. "sub" denotes whether the method used subsampling images for training.



	sub	road	swalk	build.	wall	fence	pole	tlight	sign	veg.	terrain	sky	person	rider	car	truck	bus	train	mbike	bike	mIoU
RNN [40]	2	96.3	73.9	88.2	47.6	41.3	35.2	49.5	59.7	90.6	66.1	93.5	70.4	34.7	90.1	39.2	57.5	55.4	43.9	54.6	62.5
DeepLab [3]	2	97.3	77.7	87.7	43.6	40.5	29.7	44.5	55.4	89.4	67.0	92.7	71.2	49.4	91.4	48.7	56.7	49.1	47.9	58.6	63.1
FCN [25]	no	97.4	78.4	89.2	34.9	44.2	47.4	60.1	65	91.4	69.3	93.9	77.1	51.4	92.6	35.3	48.6	46.5	51.6	66.8	65.3
DPN [22]	no	97.5	78.5	89.5	40.4	45.9	51.1	56.8	65.3	91.5	69.4	**94.5**	77.5	54.2	92.5	44.5	53.4	49.9	52.1	64.8	66.8
Dilation10 [39]	no	97.6	79.2	89.9	37.3	47.6	53.2	58.6	65.2	91.8	69.4	93.7	78.9	55	93.3	45.5	53.4	47.7	52.2	66	67.1
DeepLab-v2 [4]	no	97.8	81.3	90.3	48.7	47.3	49.5	57.8	67.2	91.8	69.4	94.1	79.8	59.8	93.7	56.5	67.4	**57.4**	57.6	68.8	70.4
Adelaide [19]	no	**98.0**	82.6	90.6	44.0	50.7	51.1	**65.0**	71.7	**92.0**	**72.0**	94.1	**81.5**	**61.1**	**94.3**	61.1	65.1	53.8	**61.6**	70.6	**71.6**
LC	no	97.9	**83.1**	**91.6**	**53.7**	**57.4**	**58.4**	62.0	**73.3**	91.9	61.3	93.8	78.8	53.1	93.4	**62.2**	**76.9**	53.5	57.0	**74.7**	71.1

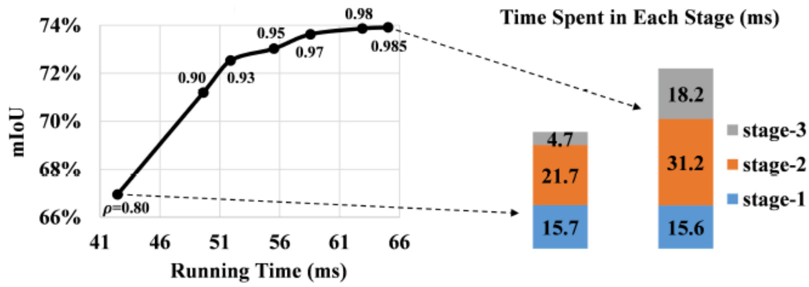
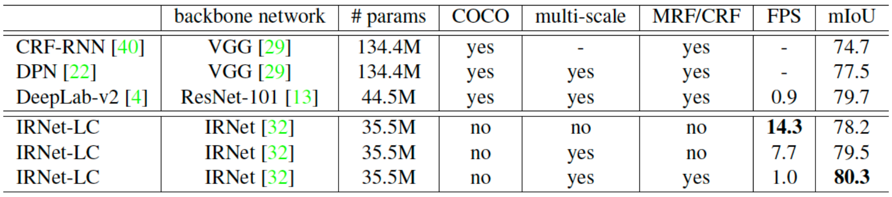


Figure 7: (a) shows the performance and speed trade-off in Layer Cascade (LC) by adjusting ρ. (b) is the time spent in each stage.

Table 6: Comparisons with state-of-the-art methods on VOC12 *test set*. '-' indicates the corresponding information was not disclosed in the previous papers.



	backbone network	# params	COCO	multi-scale	MRF/CRF	FPS	mIoU
CRF-RNN [40]	VGG [29]	134.4M	yes	-	yes	-	74.7
DPN [22]	VGG [29]	134.4M	yes	yes	yes	-	77.5
DeepLab-v2 [4]	ResNet-101 [13]	44.5M	yes	yes	yes	0.9	79.7
IRNet-LC	IRNet [32]	35.5M	no	no	no	**14.3**	78.2
IRNet-LC	IRNet [32]	35.5M	no	yes	no	7.7	79.5
IRNet-LC	IRNet [32]	35.5M	no	yes	yes	1.0	**80.3**

4. Advantages (self-summary rather than the author's)

It reveals that some pixels are harder to be classified. But with the net going deeper, the recognition ablity also is increasing. It's hard to say the performance advance comes from the multi-stage method.

5. Disadvantages (self-summary rather than the author's)

- After applying LC on Inception-ResNet-v2 (IRNet) [32], its speed and accuracy are improved by 42.8% and 1.7%, respectively. So the main contribution is on the speed, not the accuracy. But this idea is really interesting.

But I really believe the hard classes's samples are the bottleneck of semantic segmentation, not the edges of each object. So maybe a better method should be proposed.

6. Don’t understand

Secondly, as feature maps with high resolution consume a large amount of GPU memory in the learning process, they limit the size of minibatch (e.g. 8), making the batch normalization (BN) layers [14] unstable (as which need to estimate sample mean and variance from the data in a mini-batch). We cope with this issue by simply fixing the values of all parameters in BNs. This strategy works well in practice.