P-2B-30

# Distinct Class-specific Saliency Maps for Weakly-supervised Semantic Segmentation

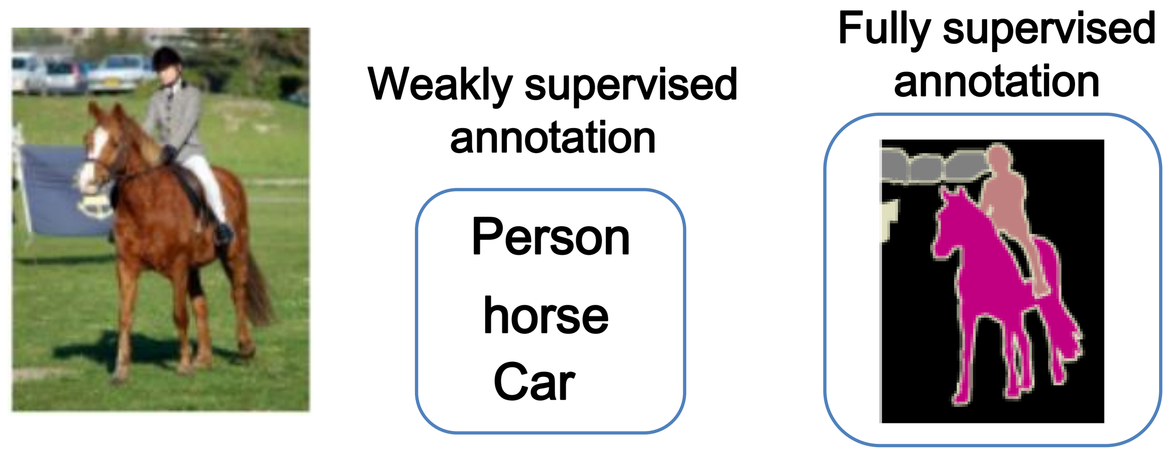
Wataru Shimoda Keiji Yanai

The University of Electro-Communications, Tokyo, Japan

## Objective

*Weakly supervised segmentation*

- Use only image-level annotation



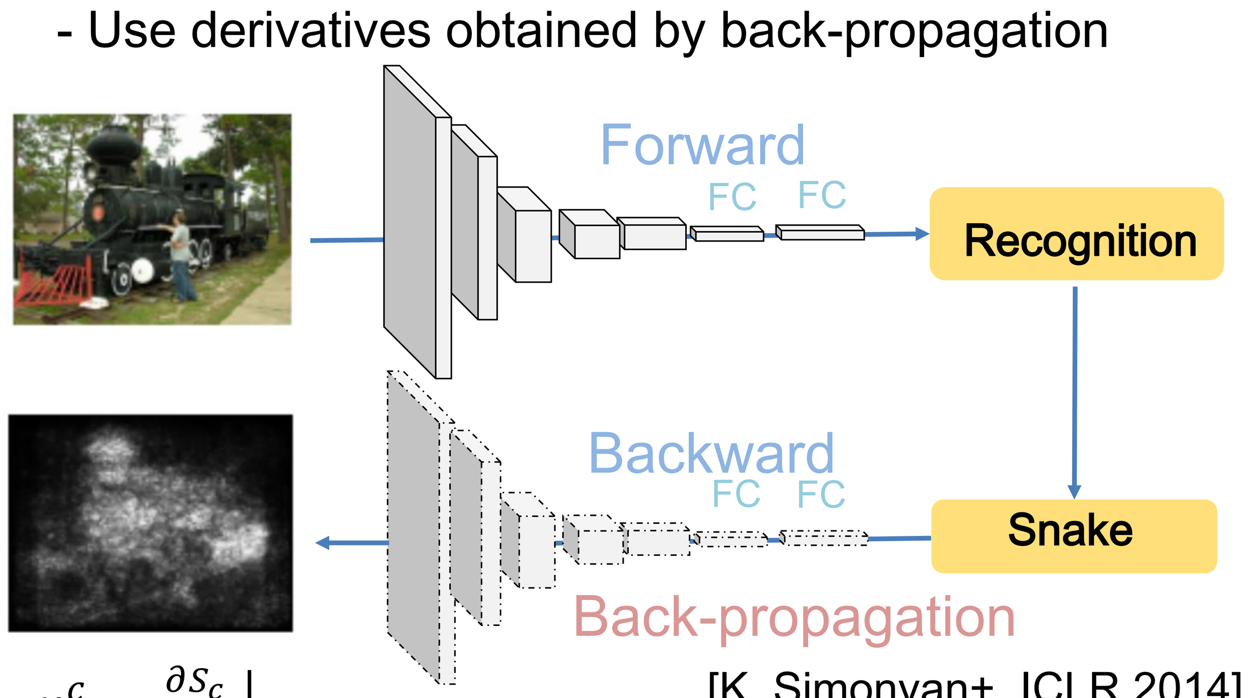

*Contributions*

- Improved the method by Simonyan et al. [1] greatly
- Achieved state-of-the-art in weakly-supervised segmentation with PASCAL VOC 2012

## BP-based Visualization

Visualize class-specific saliency maps based on the derivatives of the class scores with respect to the input image

- proposed by K. Simonyan et al. at ICLR 2014 [1]
- Visualize contributed pixels on CNN classification
- Use derivatives obtained by back-propagation



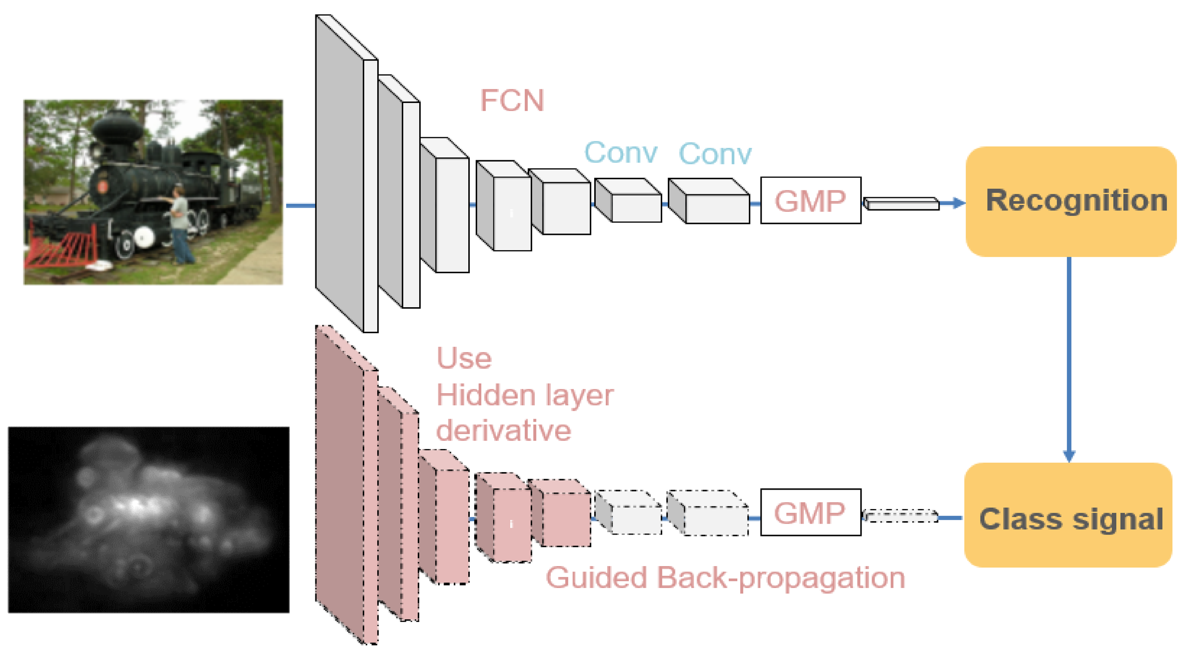

$v_i^c = \frac{\partial S_c}{\partial L_i}\Big|_{L_i^0}$ [K. Simonyan+, ICLR 2014]

White region means high derivative values which corresponds to the important pixels to enhance the given class score. (In the above fig. "Snake")

## CNN Architecture

*Improved points* (each point contributes 1~3pt improvement)

- Fully Convolutional Net - Guided back propagation [2]
- Use the derivatives of multiple intermediate layers
- Aggregate multi-scale class saliency maps (3 scales)

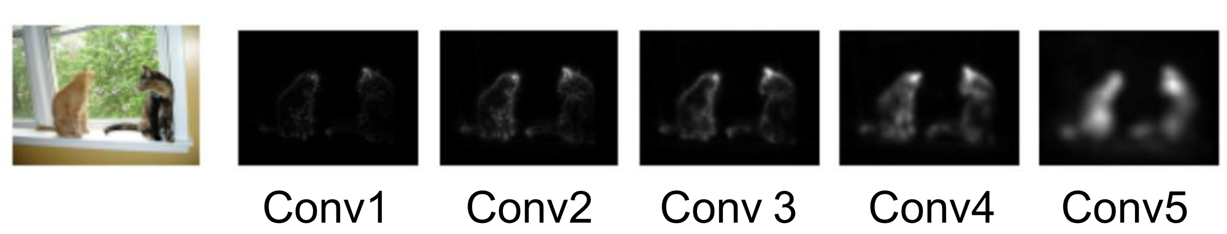



We back-propagate expected class scores generated by setting 1 for one of the top N-classes and 0 for the others. $w_i^c$ represents up-sampled i-th layer derivative which is obtained by propagating class scores from the top layer. Each class saliency maps $M_i^c \in \mathbb{R}^{m \times n}$ is calculated by:

$$M^c_{i,x,y} = \max_{k_i} |w^c_{i,h_i(x,y,k)}| \quad ,$$

where $h_i(x,y,k)$ is the index of the element of $w_i^c$.

- Fine-tune full-conv VGG-16 network with Sigmoid cross entropy loss with random-resized images (300~700px)
- Sum up the derivatives of Conv3, Conv4, and Conv5.
- Aggregate the class maps of 400*400, 500*500, and 600*600.



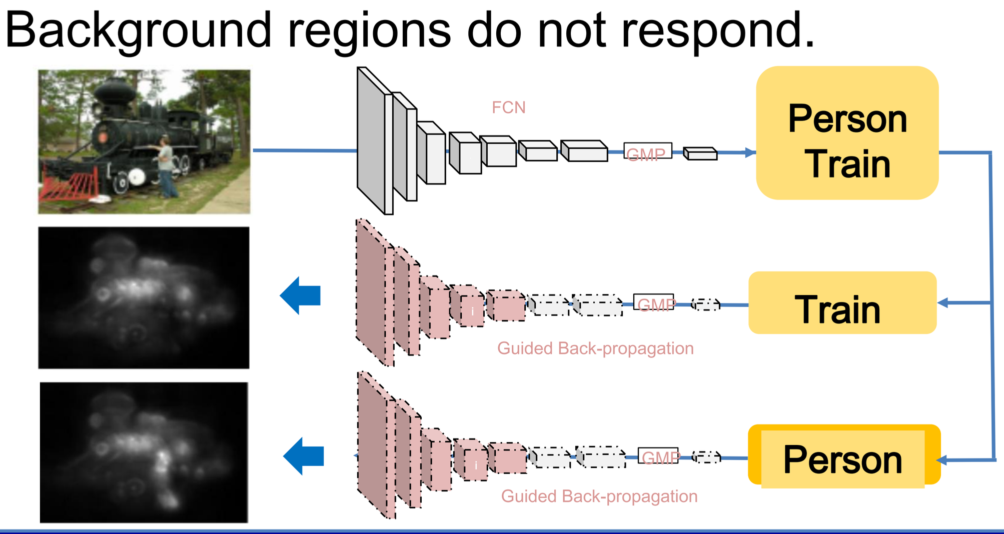

## Subtraction of class-specific derivatives

*For multi-class images*

- Only small differences were observed among the derivatives of the different classes

*Assumption*

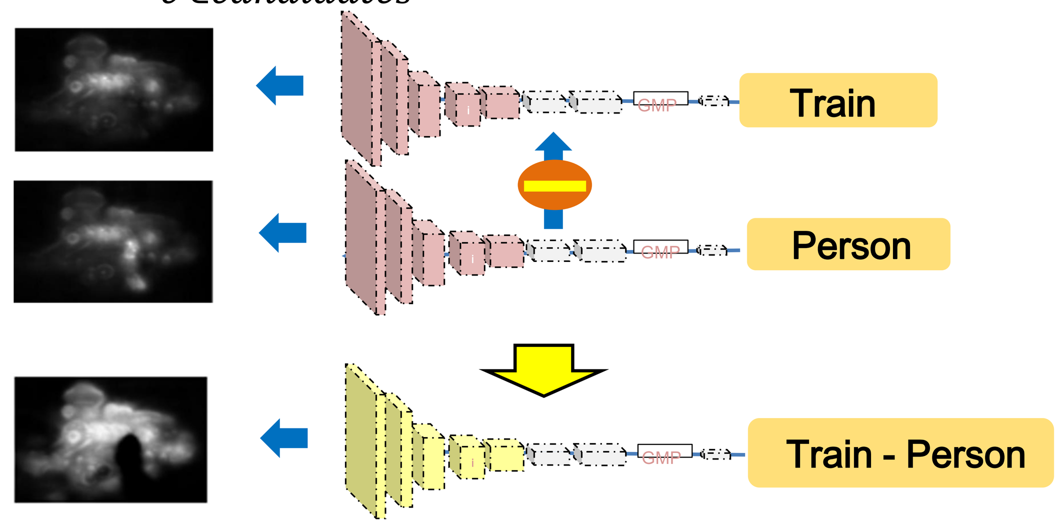
- Raw saliency maps are affected by both class-specific saliency and generic object-ness
- The degree of class saliency factors should be larger than the generic object-ness factor.
- Background regions do not respond.




*Subtraction*

- Subtract the derivatives among the different classes.
- Interestingly, in most of the cases, we obtained much clearer class maps than raw maps.
- The improved class saliency maps $\hat{M}_i^c$ with respect to class $c$ is computed by

$$\hat{M}^c_{i,x,y} = \sum_{c \in candidates} \left(M^c_{i,x,y} - M^{c'}_{i,x,y},\ 0\right) \ [c \neq c'],$$

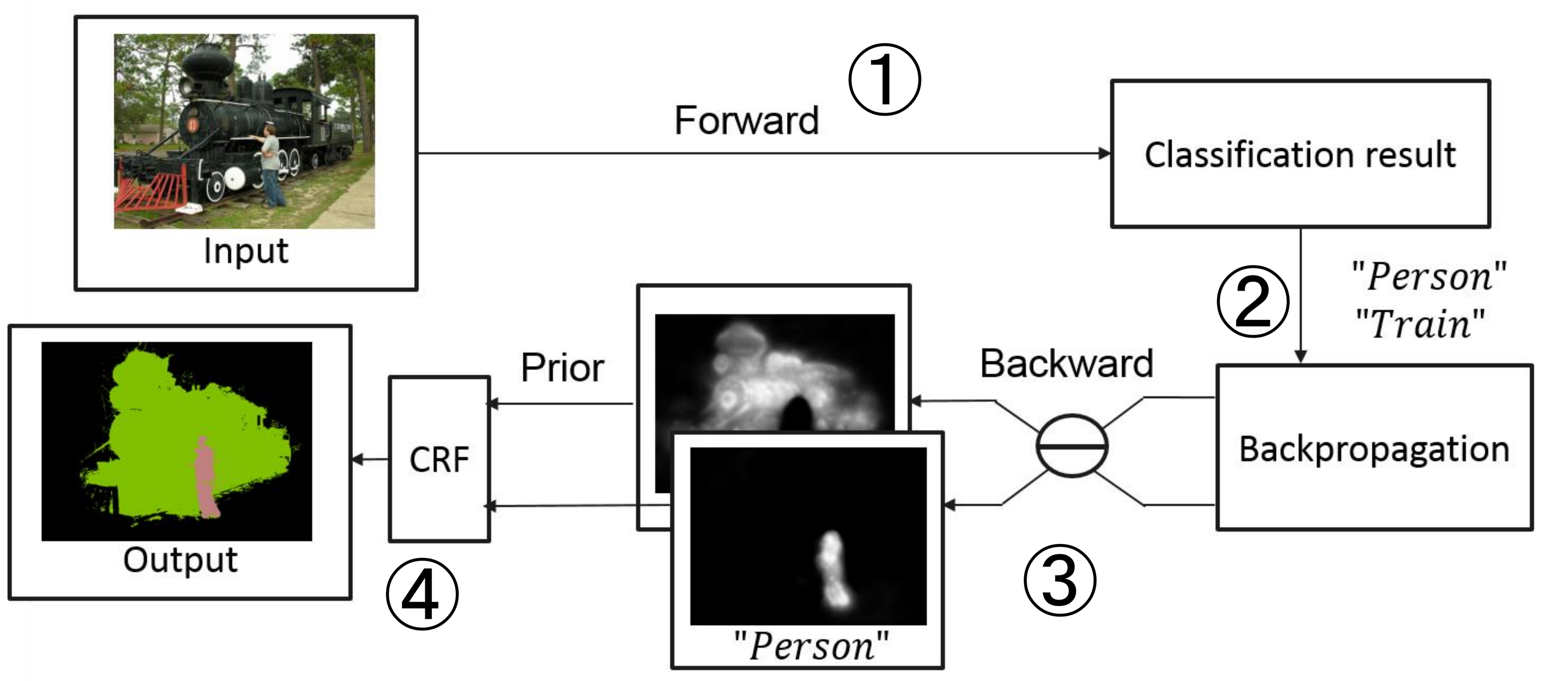



## Proposed Method

*Steps*

1. Multi-class recognition
2. Back-propagation for each of the detected classes
3. Subtracting the class maps among the Top-N classes
4. Unify the class maps by FC-CRF (dense CRF)

We regarded the classes the output scores from the multi-class CNN are more that 0.5 as the candidate classes.



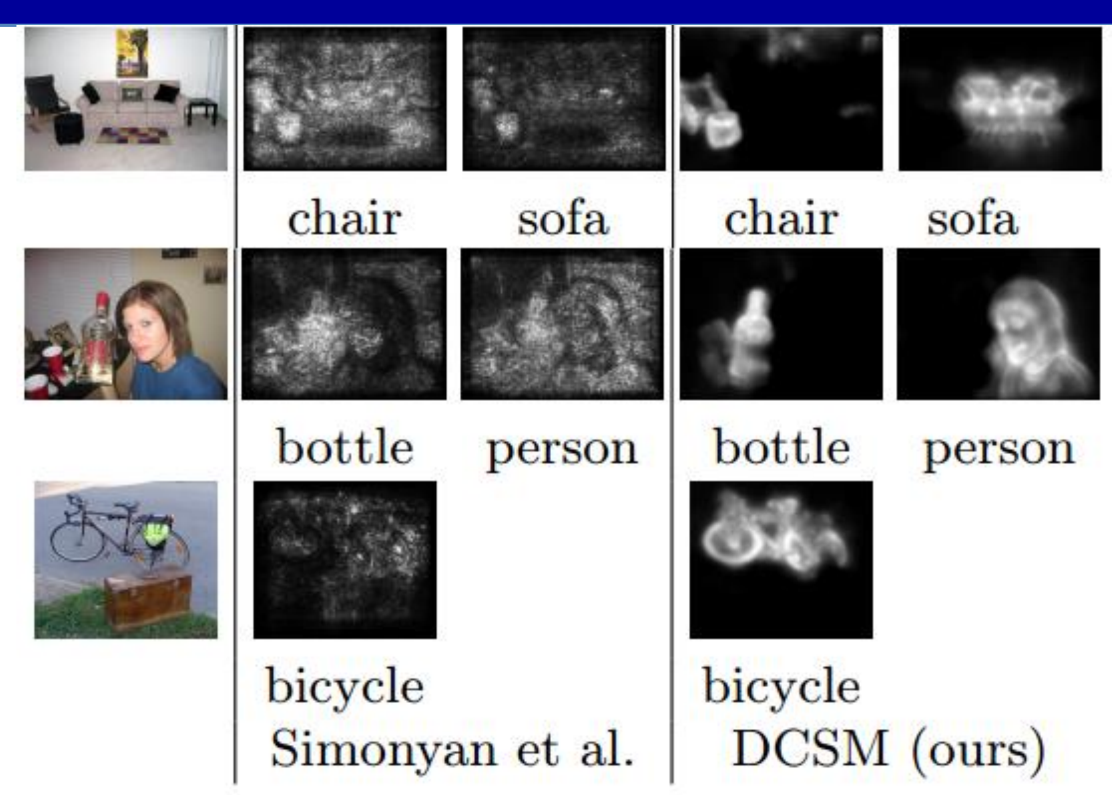

## Experiments

**Dataset : Pascal VOC 2012 + trainaug [3]**

*Comparison with Simonayn et al. [1]*

- Our gradient maps visualize class regions clearly.
- We applied FC-CRF to saliency maps obtained by Simonyan et al.[1] in the same way.
- The margin was more than 10 %.

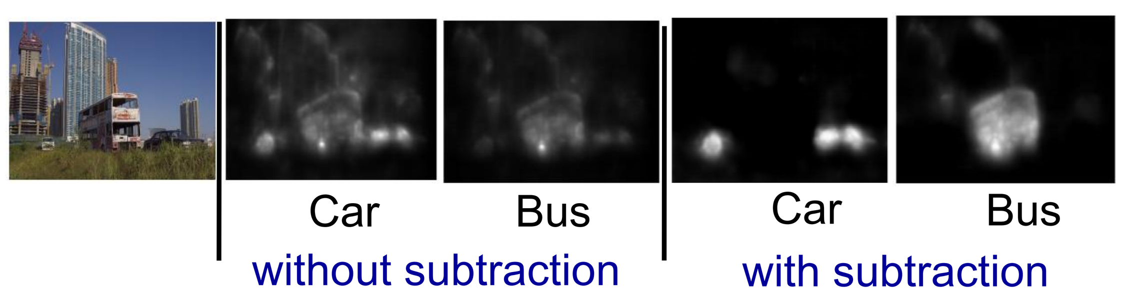
| Method | Mean IOU |
|---|---|
| Sim et al. + CRF | 33.8 |
| Ours | 44.2 |




*Effect of subtraction*

- Subtracting among the top-N classes
- N=0 means no subtraction.
- N=4 achieved the best score.

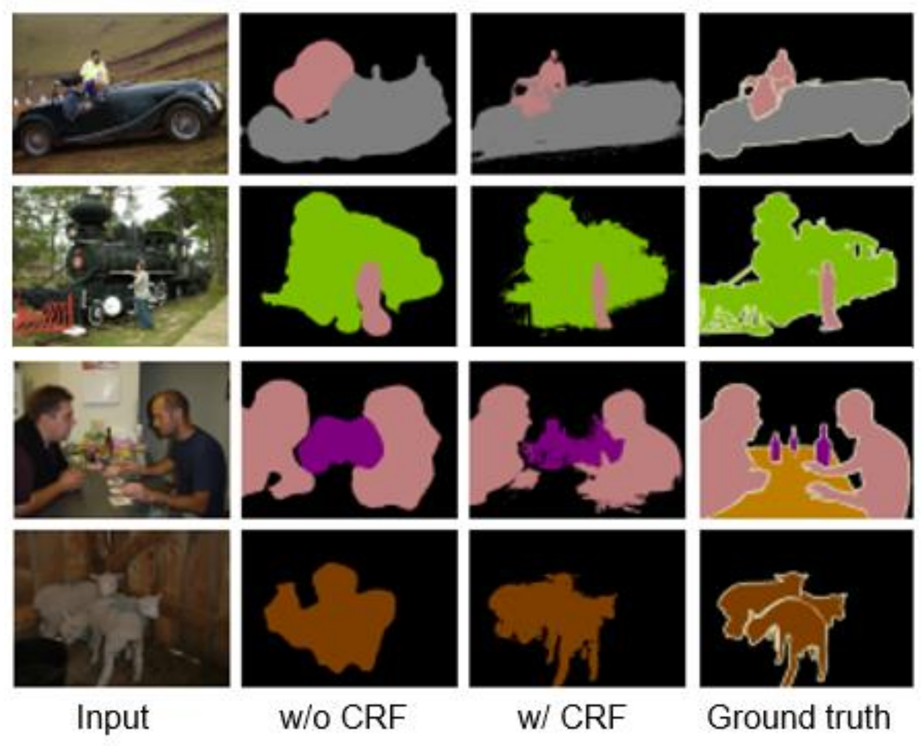
| Class N | 0 | 1 | 2 | 3 | 4 | 5 | 10 |
|---|---|---|---|---|---|---|---|
| Mean IU | 38.2 | 42.2 | 43.5 | 44.1 | 44.2 | 44.0 | 43.7 |




*Comparison with state-of-the-arts*

- A means using additional images.
- B means using additional supervision.

| Method | A | B | Mean IOU |
|---|---|---|---|
| One point (ECCV 2016) | - | ✓ | 46.1 |
| Check Mask (ECCV2016) | - | ✓ | 51.5 |
| MIL-FCN (ICLR 2015) | - | - | 25.7 |
| EM-Adapt(ICCV 2015) | - | - | 38.2 |
| CCNN (ICCV 2015) | - | - | 34.5 |
| MIL-seg (CVPR2015) | ✓ | - | 42.0 |
| STC (arXiv:1509.03150) | ✓ | - | 49.8 |
| SEC (ECCV 2016) | - | - | 50.7 |
| Ours w/o CRF | - | - | 40.5 |
| Ours w/ CRF | - | - | 44.2 |




*Project page*

http://mm.cs.uec.ac.jp/shimoda-k/space/dcsm/

*Source code*

https://github.com/shimoda-uec/dcsm

Caffe-based implementation which takes 0.3 [s] with GPU for an one-time forward-backward pass.

## References

[1] K. Simonyan et al. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. ICLR, 2014.

[2] J. Springenberg et al. Striving for Simplicity: The All Convolutional Net. ICLR, 2015.

[3] B. Hariharan et al. Semantic Contours from Inverse Detectors. ICCV, 2011.