# Edge Detection: Deep Contour

Wei Shen's work – uses clustering and patches.

# Our Work



## DeepContour: A Deep Convolutional Feature Learned by Positive-sharing Loss for Contour Detection

Wei Shen, Xinggang Wang, Yan Wang, Xiang Bai, Zhijiang Zhang

**IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2015**

# Motivation



- Why to apply CNN for contour detection?
  - Contour is hard to define


  - Contour data are sufficient for CNN training (millions of local contour patches)

# Problem Formulation



- Given a color image patch $x \in \mathbb{R}^{n \times n \times 3}$, our goal is to determine whether its center pixel is passed through by contours or not.

$$x \in \mathbb{R}^{n \times n \times 3} \rightarrow z \in \{0,1\}$$

Non-contour Contour

Q: Is a good idea to directly use CNN as a blackbox to address this binary classification problem?

# Obstacle



- The large variations in the contour shapes

Ref: J. J. Lim, C. L. Zitnick, and P. Dollár. Sketch tokens: A learned mid-level representation for contour and object detection. CVPR, 2013.

Solution: Partitioning contour patches into compact clusters to convert the binary classification problem to a multi-class classification problem

# Obstacle



- How to define the loss function?

  Q: Is softmax a good choice?

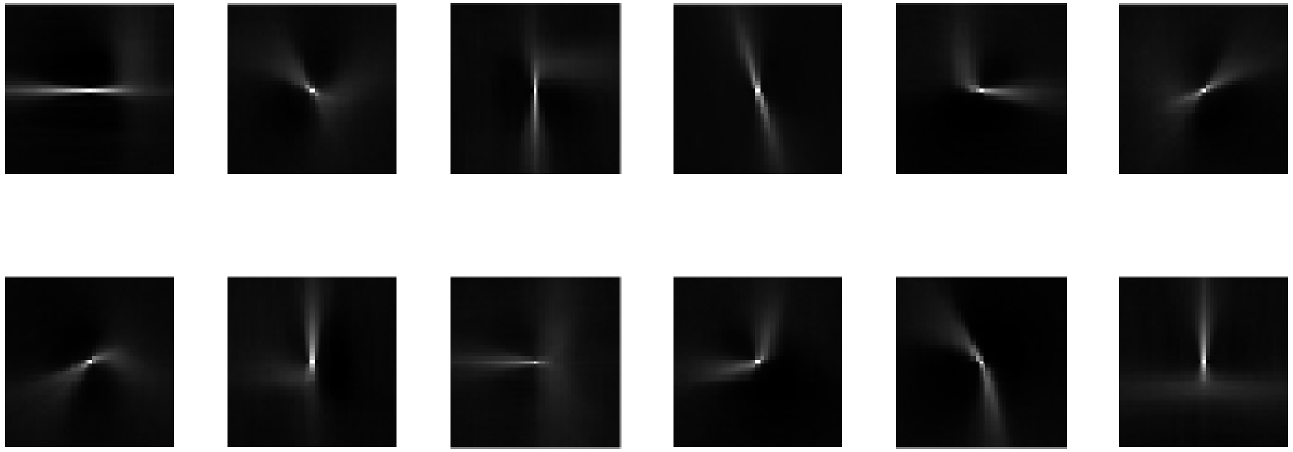
- Softmax function penalizes the loss of each class equally
- The losses for contour versus non-contour should be emphasized

Solution: Adding a regularized term to focus on the end goal of binary classification

# Data Preparation



- Pre-cluster contour patches according to their contour shapes.



- Assign a label $y$ to each contour patch $x$ according to the pre-cluster index $\{1, \ldots K\}$.

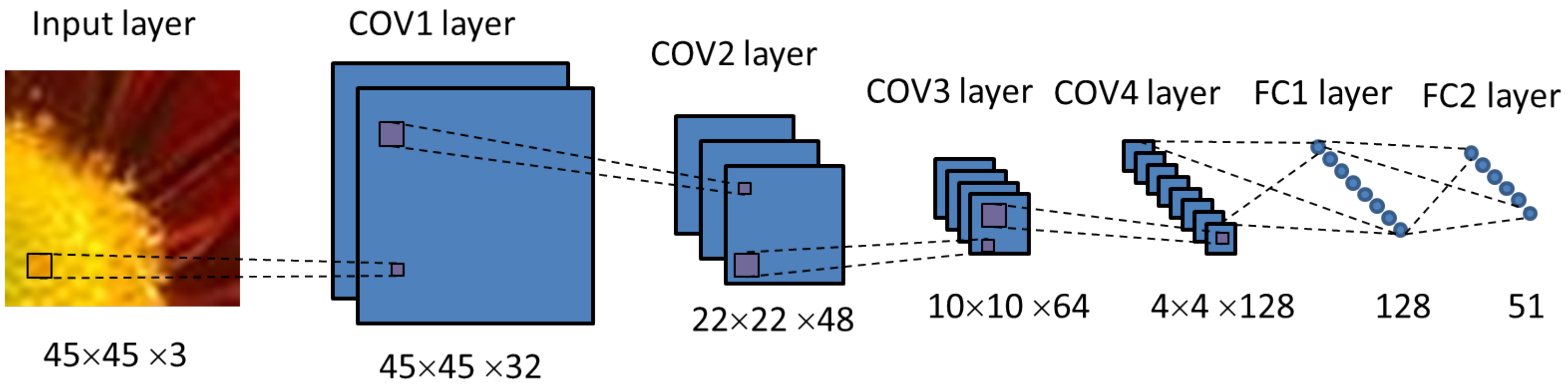
$$x \in \mathbb{R}^{n \times n \times 3} \rightarrow y \in \{0, 1, \ldots K\}$$

Negative Positive

# Method



- CNN Architecture




- Loss Function

Let $(a_j^{(i)}; j = 1, \ldots K)$ be the output of unit $j$ in FC2 for a image patch $x_j^{(i)}$, the probability that the label is $j$ is

$$p_j^{(i)} = \frac{\exp(a_l^{(i)})}{\sum_{l=0}^{K} \exp(a_l^{(i)})}$$

# Method



$$J = -\frac{1}{m}\sum_{i=1}^{m}\left(\sum_{j=0}^{K}\mathbf{1}\left(y^{(i)}=j\right)\log p_j^{(i)}\right) - \frac{1}{m}\left[\sum_{i=1}^{m}\lambda\left(\mathbf{1}\left(y^{(i)}=0\right)\log p_0^{(i)} + \sum_{j=1}^{K}\mathbf{1}\left(y^{(i)}=j\right)\log(1-p_0^{(i)})\right)\right]$$

Softmax loss

Positive-sharing loss, the loss for positive class is shared among each pre-clustered contour classes

# Method



- To apply standard back-propagation to optimize the parameters of the network

$$\frac{\partial J}{\partial a_0^{(i)}} = \frac{1}{m}\left[(\lambda+1)\mathbf{1}\left(y^{(i)}=0\right)\left(p_0^{(i)}-1\right)+(\lambda+1)\sum_{j=1}^{K}\mathbf{1}\left(y^{(i)}=j\right)p_0^{(i)}\right]$$

$$\frac{\partial J}{\partial a_l^{(i)}} = \frac{1}{m}\left[\left(\lambda\mathbf{1}\left(y^{(i)}=0\right)+1\right)p_l^{(i)}-\mathbf{1}\left(y^{(i)}=l\right)-\lambda\sum_{j=1}^{K}\mathbf{1}\left(y^{(i)}=j\right)\frac{p_0^{(i)}p_l^{(i)}}{1-p_0^{(i)}}\right]$$

# Method



- CNN model validation

$$\gamma = \frac{1}{m}\sum_{i=1}^{m}\left[\left(\mathbf{1}\left(y^{(i)} = 0\right) - \mathbf{1}\left(y^{(i)} > 0\right)\right)(p_0^{(i)} - (1 - p_0^{(i)}))\right]$$

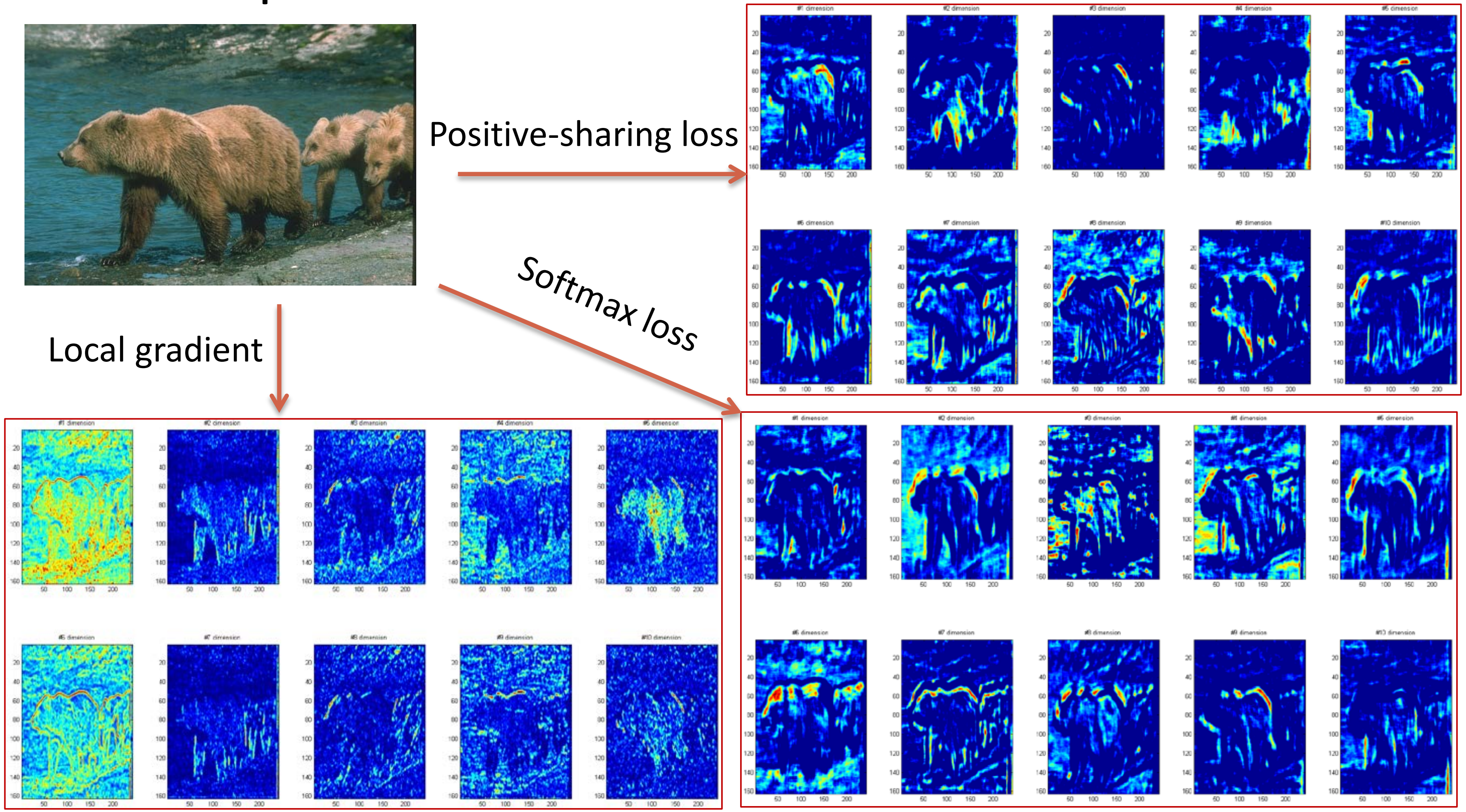
$\gamma \in [-1,1]$, measuring the discrimination of the learned model between positive and negative samples.

- The learned features of FC1 will be fed into structured forest to perform contour detection.

# Experimental results

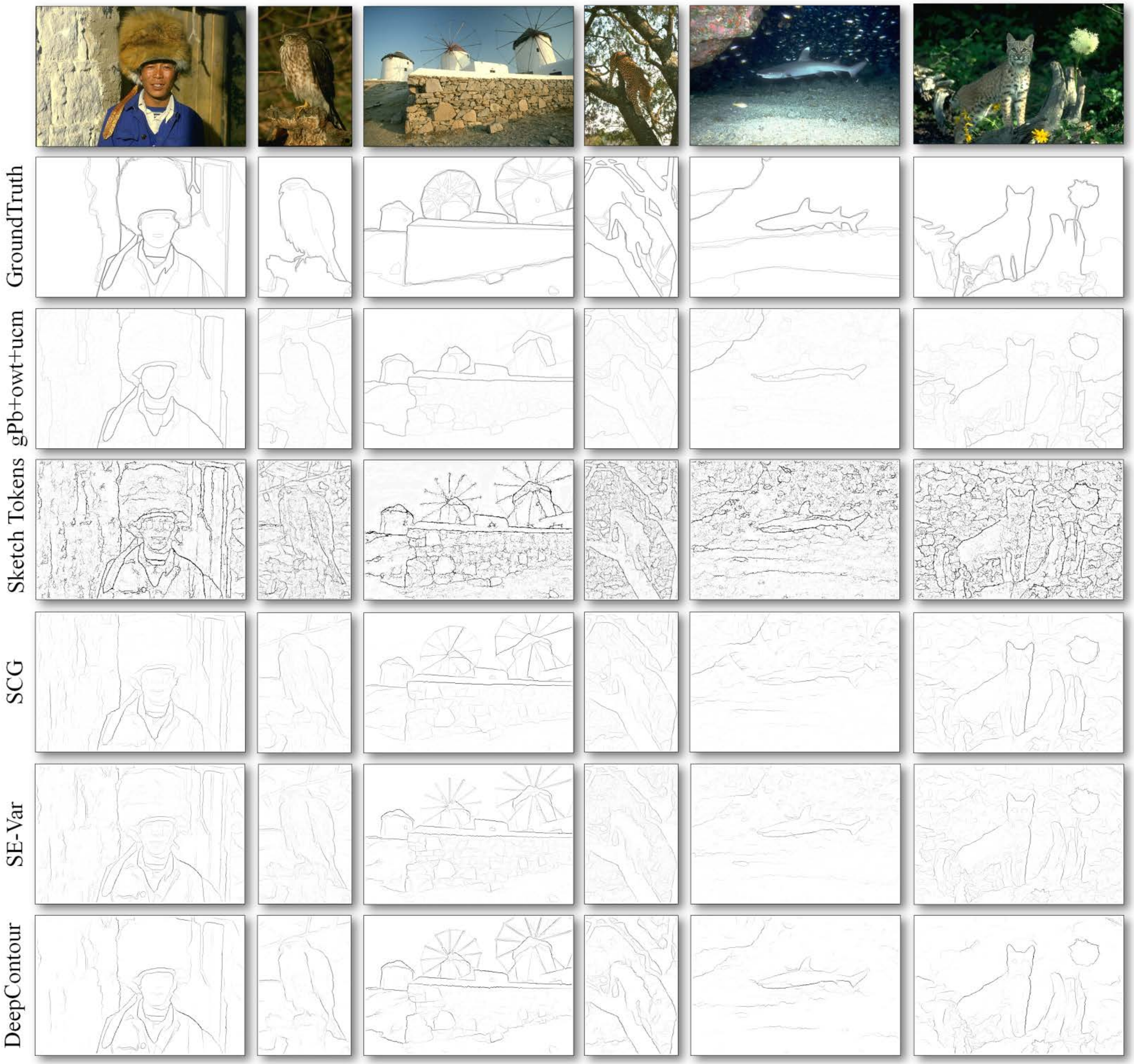


- Deep Feature Visualization



Experimental results

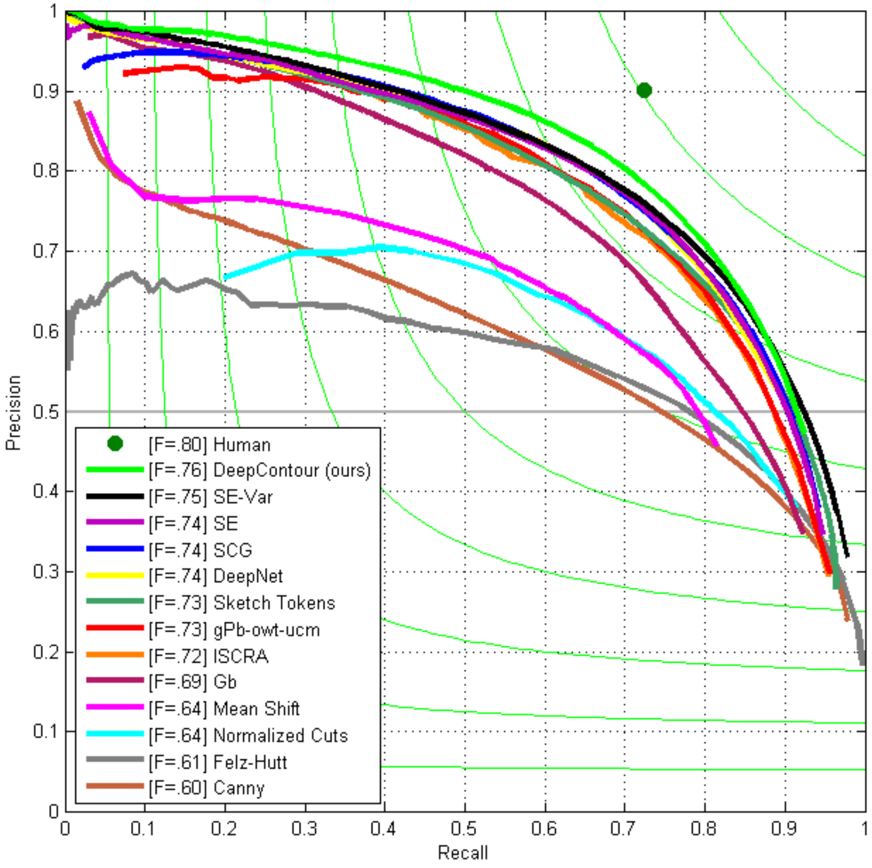
# Results on BSDS500



# Experimental results



- Results on BSDS500




| | ODS | OIS | AP |
|---|---|---|---|
| Human | .80 | .80 | - |
| Canny [6] | .60 | .63 | .58 |
| Felz-Hutt [16] | .61 | .64 | .56 |
| Normalized Cuts [8] | .64 | .68 | .45 |
| Mean Shift [7] | .64 | .68 | .56 |
| Gb [28] | .69 | .72 | .72 |
| ISCRA [39] | .72 | .75 | .46 |
| gPb-owt-ucm [2] | .73 | .76 | .73 |
| Sketch Tokens [29] | .73 | .75 | .78 |
| DeepNet [23] | .74 | .76 | .76 |
| SCG-[38] | .74 | .76 | .77 |
| PMI+sPb [21] | .74 | .77 | .78 |
| SE [11] | .74 | .76 | .78 |
| SE-Var [12] | .75 | .77 | .80 |
| $N^4$-Fields [19] | .75 | .77 | .78 |
| DeepContour (ours) | **.76** | **.78** | .80 |

# Experimental results

- Results on NYUD

# Experimental results



- Cross Dataset Generalization

| DeepContour | ODS | OIS | AP |
|---|---|---|---|
| BSDS/BSDS | .76 | .78 | .80 |
| NYU/BSDS | .72 | .74 | .77 |
| BSDS/NYU | .59 | .60 | .53 |
| NYU/NYU | .62 | .63 | .57 |

| SE [11] | ODS | OIS | AP |
|---|---|---|---|
| BSDS/BSDS | .74 | .76 | .78 |
| NYU/BSDS | .72 | .73 | .76 |
| BSDS/NYU | .55 | .57 | .46 |
| NYU/NYU | .60 | .61 | .56 |

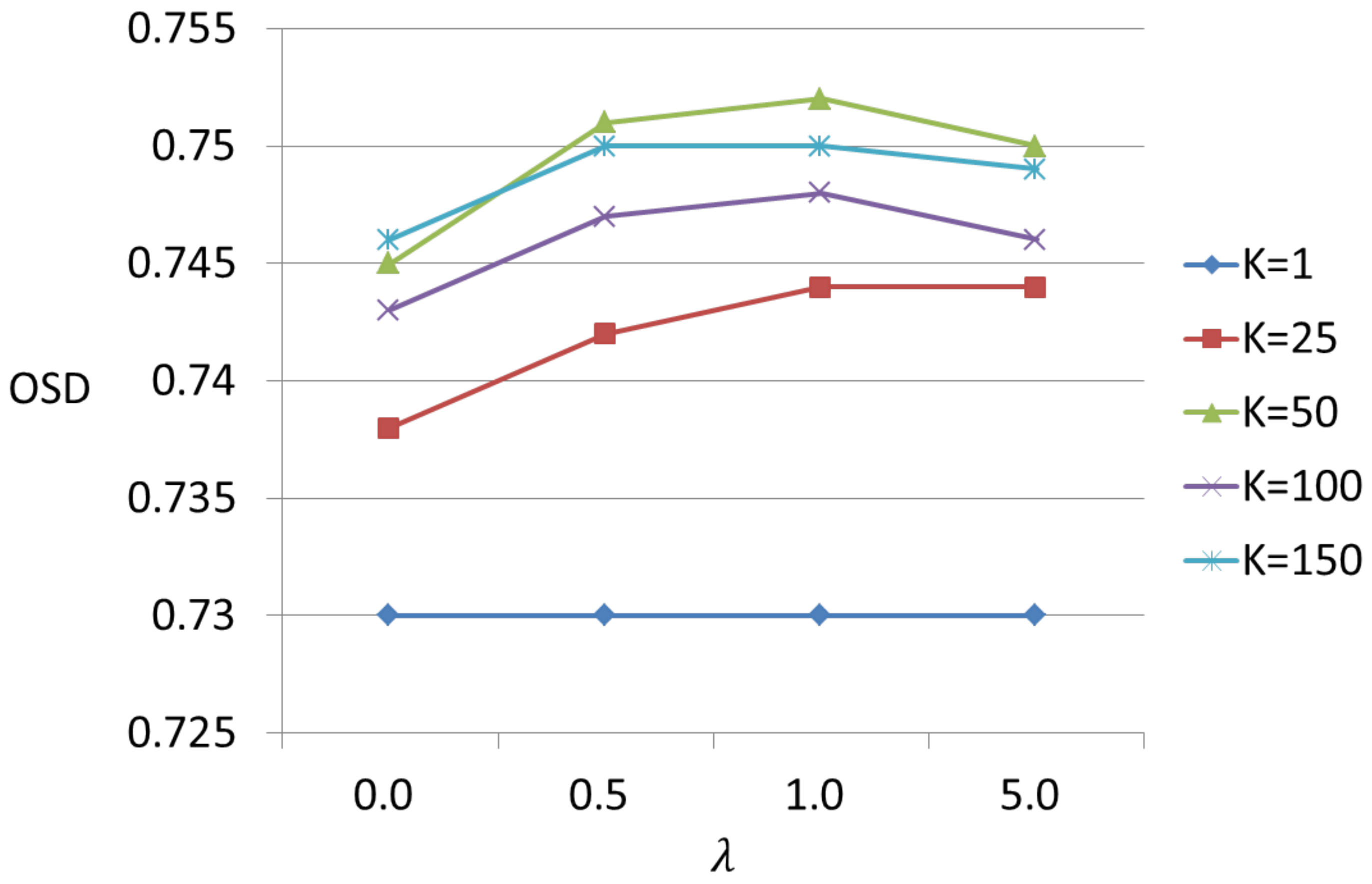
| SCG [39] | ODS | OIS | AP |
|---|---|---|---|
| NYU/NYU | .55 | .57 | .46 |

| gPb [2] | ODS | OIS | AP |
|---|---|---|---|
| NYU/NYU | .51 | .52 | .37 |

# Experimental results



- Parameter Discussion



# Outline

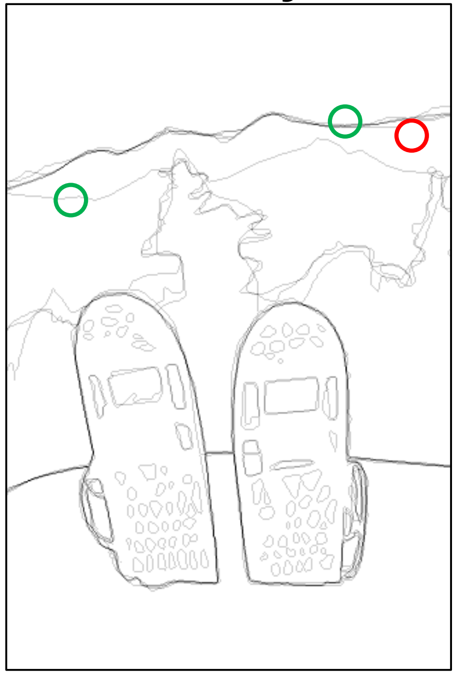


- Contour Detection
  - Overview
  - Milestones
- Our Work
- Discussions

# Discussions



- Speed
  - Caffe – per-patch mean subtraction is the bottleneck
- Accuracy
  - Limitation may caused by the confusing labels



# Discussions

上海大学
Shanghai University



Thank you!
Q&A