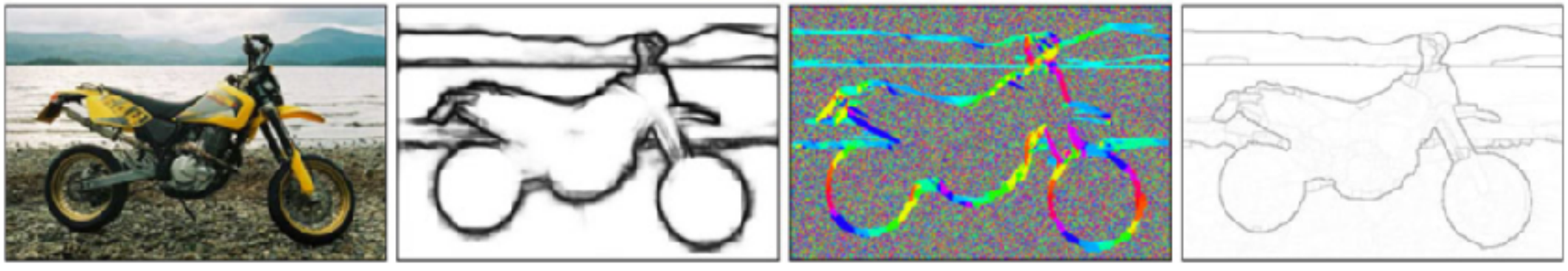

Convolutional Oriented Boundaries:
From Image Segmentation to High-Level Tasks

K.K. Maninis, J. Pont-Tuset, P. Arbela´ez, L. Van Gool

ECCV 2016 & IEEE Trans PAMI 2018

Kai Xie
kxie.cs@gmail.com

Outline

- Introduction
- Contour detection module: Convolutional Oriented Boundaries (COB)
  - Overview*
  - Network architecture
- Fusion of Semantic Boundaries and Semantic Segmentation
- Results

Introduction

Main contribution

- Proposes Convolutional Oriented Boundaries (COB), a generic CNN architecture that allows end-to-end learning of multi-scale oriented contours.
- Proposes a sparse boundary representation for efficient construction of hierarchical regions from the contour signal.

Overview

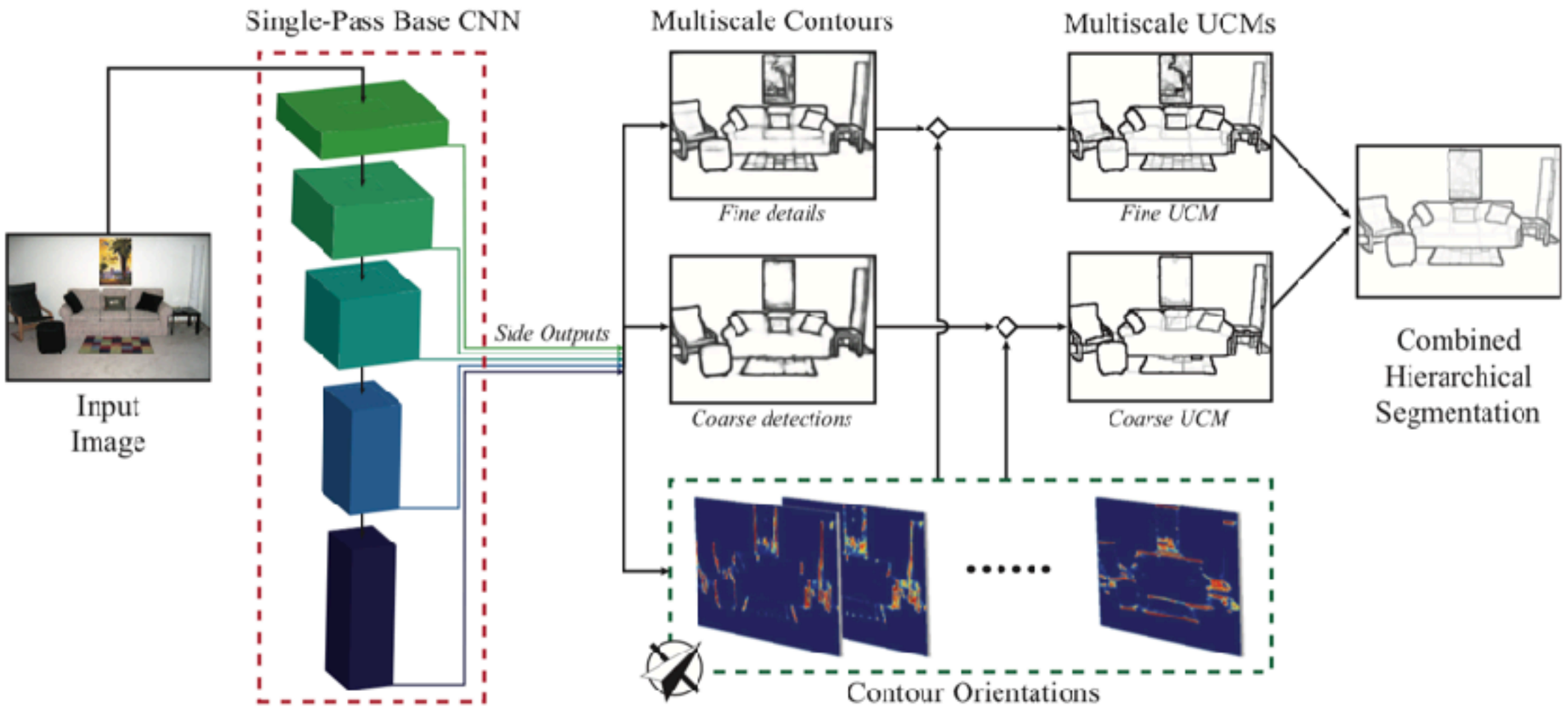

Fig. 1. Overview of COB: From a single pass of a base CNN, we obtain multiscale oriented contours. We combine them to build Ultrametric Contour Maps (UCMs) at different scales and fuse them into a single hierarchical segmentation structure.

Network architecture

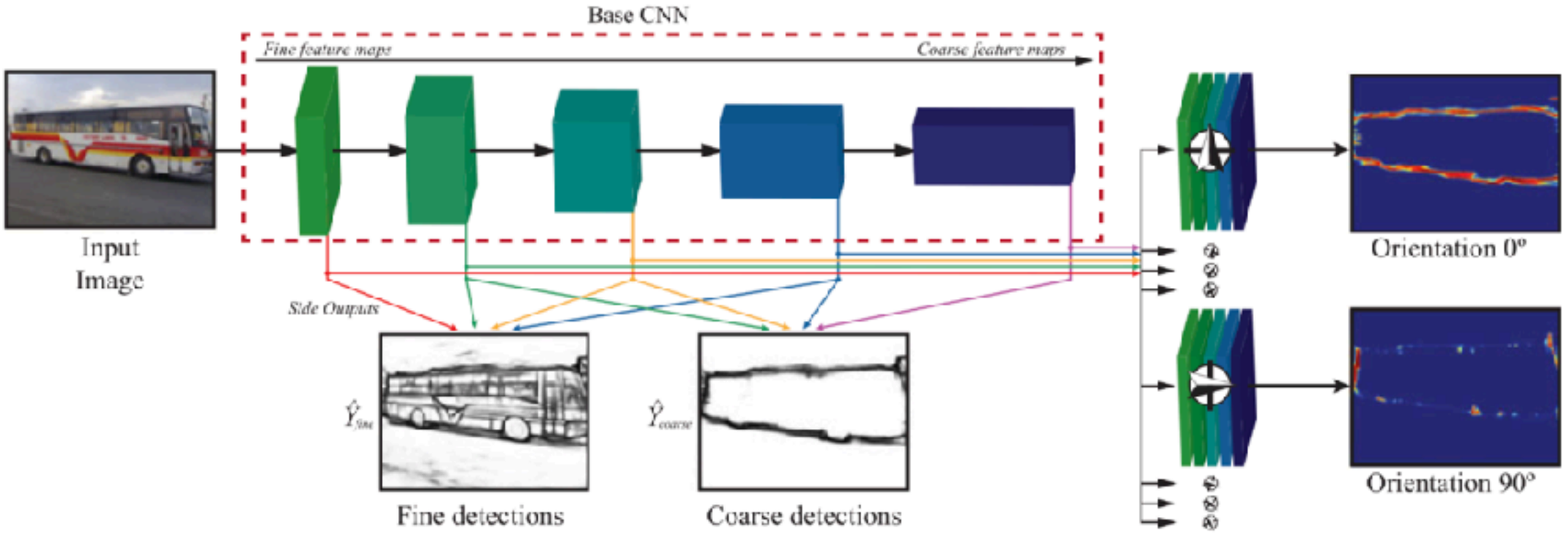

Fig. 2. Our deep learning architecture (best viewed in color). The connections show the different stages that are used to generate the multiscale contours. Orientations further require additional convolutional layers in multiple stages of the network.

- build a multi-scale oriented contour detector.
- combine the side activations of the 4 finest and 4 coarsest scales to a fine-scale and a coarse-scale output with trainable weights.

Loss

the authors, we denote the training dataset by $S = \{(X_n, Y_n), n = 1, \ldots, N\}$, with X_n being the input image and $Y_n = \{y_j^{(n)}, j = 1, \ldots, |X_n|\}, y_j^{(n)} \in \{0, 1\}$ the predicted pixelwise labels. For simplicity, we drop the subscript n. Each of the M side outputs minimizes the objective function:

$$\ell_{side}^{(m)}\left(\mathbf{W}, \mathbf{w}^{(m)}\right) = -\beta \sum_{j \in Y_+} \log P\left(y_j = 1 | X; \mathbf{W}, \mathbf{w}^{(m)}\right) - (1-\beta) \sum_{j \in Y_-} \log P\left(y_j = 0 | X; \mathbf{W}, \mathbf{w}^{(m)}\right)$$

where $\ell_{side}^{(m)}$ is the loss function for scale $m \in \{1, \ldots, M\}$, $\mathbf{W}$ denotes the standard set of parameters of the CNN, and $\{\mathbf{w}^{(m)}, m = 1, \ldots, M\}$ the corresponding weights of the the m-th side output. The multiplier β is used to handle the imbalance of the substantially greater number of background compared to contour pixels. Y_+ and Y_- denote the contour and background sets of the ground-truth Y, respectively. The probability $P(\cdot)$ is obtained by applying a sigmoid $\sigma(\cdot)$ to the activations of the side outputs $\hat{A}_{side}^{(m)} = \{a_j^{(m)}, j = 1, \ldots, |Y|\}$. The activations are finally fused linearly, as: $\hat{Y}_{fuse} = \sigma\left(\Sigma_{m=1}^{M} h_m \hat{A}_{side}^{(m)}\right)$ where $\mathbf{h} = \{h_m, m = 1, \ldots, M\}$ are the fusion weights. The fusion output is also trained

Estimation of Contour Orientations

- The orientation of each contour pixel is obtained by approximating the ground-truth boundaries with polygons, and assigning each pixel the orientation of the closest polygonal segment.
- Orientation map is obtained as:

$$O(x,y) = \mathcal{T}\left(\arg\max_k B_k(x,y)\right), k = 1,\ldots,K$$

where *B_k(x, y)* denotes the response of the k-th orientation bin of the CNN at the pixels with coordinates *(x, y)* and *T (·)* is the transformation function which associates each bin with its central angle.

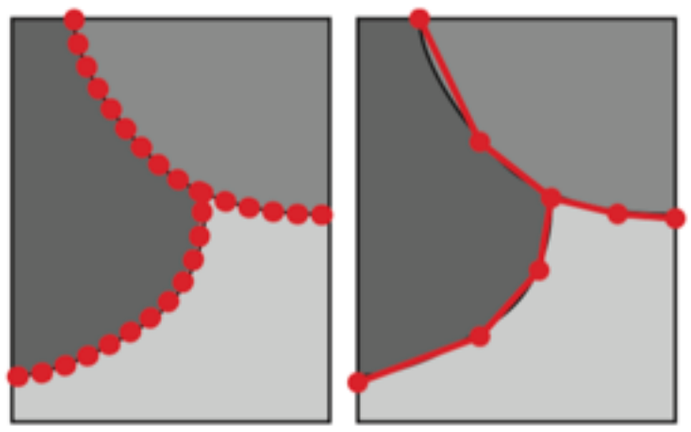

Fig. 5. Polygon simplification: From all boundary points (left) to simplified polygons (right).

Fusion

Approach

Separately approach semantic segmentation and contour detection, and fuse the results of the two tasks.

Fusion method

We couple with the COB boundaries with Semantic Segmentation results by dilated convolutions. Specifically, we mask the boundaries with Semantic Segmentation results, with a tolerance of 0.02 of the image diagonal.

Results

Object Boundary Detection

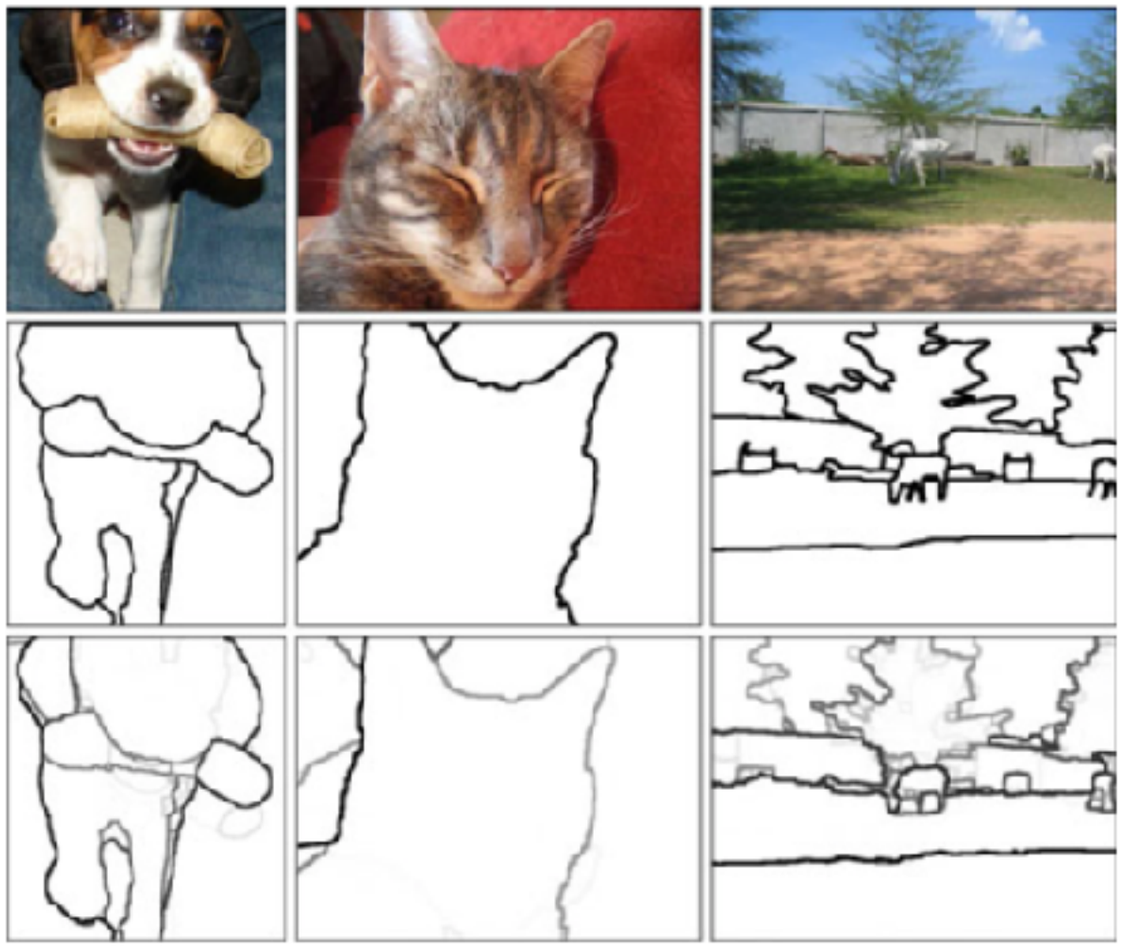

Fig. 9. *Qualitative results on PASCAL - hierarchical regions.* Row 1: Original images, Row 2: Ground-truth boundaries, Row 3: Hierarchical regions with COB.

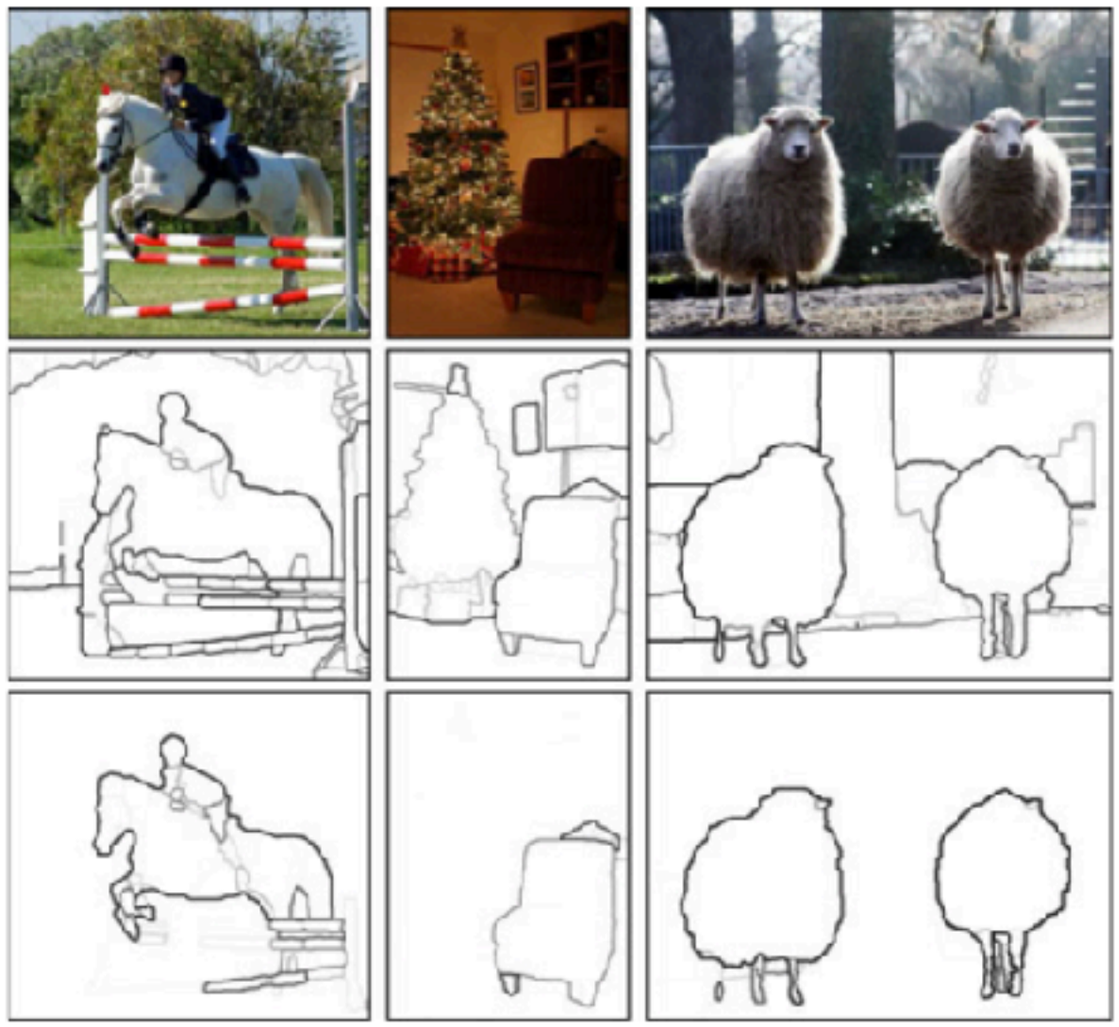

Fig. 10. *Qualitative results for object boundaries.* Row 1: Original images, Row 2: Generic image segmentation results, Row 3: Object boundary results.

Results

Semantic Segmentation

TABLE 6
PASCAL VOC Segmentation val Evaluation: Effect of COB on Semantic Segmentation

Technique	BG	Plane	Bicycle	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Table	Dog	Horse	MBike	Person	Plant	Sheep	Sofa	Train	TV	Mean
COB-dil	**93.5**	**90.3**	**39.7**	**83.2**	**66.2**	**68.9**	**92.6**	**84.6**	**89.2**	**36.9**	**84.7**	**53.1**	**82.9**	**87.0**	**83.1**	**86.3**	**54.7**	**84.8**	**45.7**	**84.6**	**68.9**	**74.3**
DilatedConv [68]	92.8	87.1	39.2	79.6	65.9	66.3	90.0	82.5	85.3	36.2	81.7	51.7	78.1	83.8	80.2	83.4	50.5	82.6	43.1	83.8	65.3	71.9
COB-PSP	**95.4**	**90.9**	**44.8**	90.2	**76.1**	**84.1**	96.1	**92.1**	**95.3**	45.6	**95.4**	**59.9**	**92.0**	**93.2**	90.8	**90.1**	**68.0**	**93.4**	**50.2**	**93.3**	79.8	**81.7**
PSPNet [69]	95.3	90.7	44.4	90.2	74.8	83.4	**96.3**	92.0	95.0	**46.4**	94.6	59.1	91.9	92.5	**91.0**	89.9	66.0	91.6	50.2	93.0	**80.0**	81.3

Per-class IoU and mean IoU are reported.

Fig. 16. *Qualitative results for semantic segmentation.* Row 1: Original images, Row 2: Dilated convolution network, Row 3: Dilated network with COB superpixels.

We treat the COB UCMs as superpixels, by applying a low value threshold (0.1) to the hierarchy, which results in high recall. We then snap the semantic segmentation results to the superpixels by majority voting of the regions, i.e., superpixels that overlap more than 50 percent with the semantic class, are assigned the corresponding label.