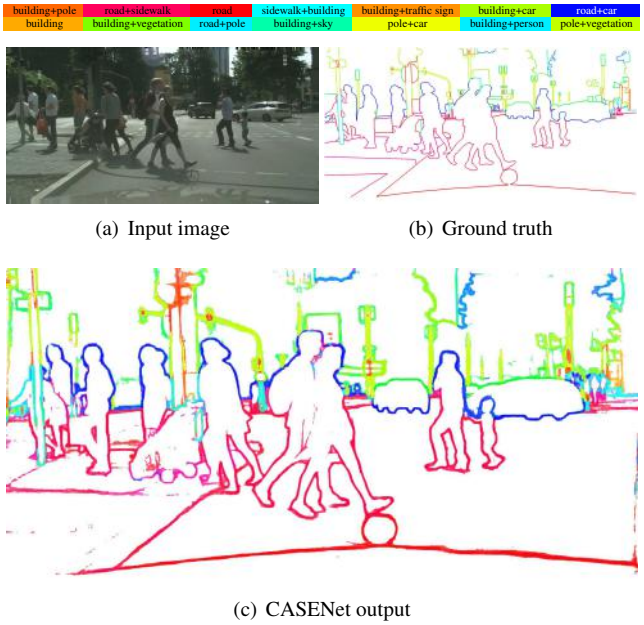
CASENet: Deep Category-Aware Semantic Edge Detection

Zhiding Yu*
Carnegie Mellon University
yzhiding@andrew.cmu.edu

Chen Feng* Ming-Yu Liu† Srikumar Ramalingam†
Mitsubishi Electric Research Laboratories (MERL)
cfeng@merl.com, mingyul@nvidia.com, srikumar@cs.utah.edu

Abstract

Boundary and edge cues are highly beneficial in improving a wide variety of vision tasks such as semantic segmentation, object recognition, stereo, and object proposal generation. Recently, the problem of edge detection has been revisited and significant progress has been made with deep learning. While classical edge detection is a challenging binary problem in itself, the category-aware semantic edge detection by nature is an even more challenging multi-label problem. We model the problem such that each edge pixel can be associated with more than one class as they appear in contours or junctions belonging to two or more semantic classes. To this end, we propose a novel end-to-end deep semantic edge learning architecture based on ResNet and a new skip-layer architecture where category-wise edge activations at the top convolution layer share and are fused with the same set of bottom layer features. We then propose a multi-label loss function to supervise the fused activations. We show that our proposed architecture benefits this problem with better performance, and we outperform the current state-of-the-art semantic edge detection methods by a large margin on standard data sets such as SBD and Cityscapes.



(a) Input image (b) Ground truth

(c) CASENet output

Figure 1. Edge detection and categorization with our approach. Given a street view image, our goal is to simultaneously detect the boundaries and assign each edge pixel with one or more semantic categories. (b) and (c) are color coded by HSV where hue and saturation together represent the composition and associated strengths of categories. Best viewed in color.

1. Introduction

Figure 1 shows an image of a road scene from Cityscapes dataset [8] with several object categories such as building, ground, sky, and car. In particular, we study the problem of simultaneously detecting edge pixels and classifying them based on association to one or more of the object categories [18, 42]. For example, an edge pixel lying on the contour separating building and pole can be associated with both of these object categories. In Figure 1, we visualize the boundaries and list the colors of typical category combinations such as "building+pole" and "road+sidewalk". In our problem, every edge pixel is denoted by a vector whose individual elements denote the strength of pixel's association with different semantic classes. While most edge pixels will be associated with only two object categories, in the case of junctions [37] one may expect the edge pixel to be associated with three or even more. We therefore do not restrict the number of object categories a pixel can be associated with, and formulate our task as a multi-label learning problem. In this paper, we propose CASENet, a deep network able to detect category-aware semantic edges. Given K defined semantic categories, the network essentially produces K separate edge maps where each map indicates the edge probability of a certain category. An example of separately visualized edge maps on a test image is given in Figure 2.

The problem of edge detection has been shown to be useful for a number of computer vision tasks such as segmen-

*The authors contributed equally.
†This work was done during the affiliation with MERL.

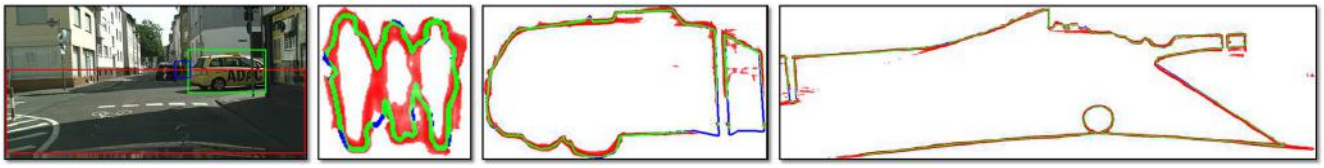


Figure 2. An example of a test image and zoomed edge maps corresponding to bounding box regions. The visualized edge maps belong to the categories of person, car and road, respectively. Green and blue denote correctly detected and missed edge pixels.

tation [1, 3, 4, 6, 52], object proposal [3], 3d shape recovery [27], and 3d reconstruction [44]. By getting a better understanding of the edge classes and using them as prior or constraints, it is reasonable to expect some improvement in these tasks. With a little extrapolation, it is not difficult to see that a near-perfect semantic edge, without any additional information, can solve semantic segmentation, depth estimation [21, 38], image-based localization [24], and object detection [13]. We believe that it is important to improve the accuracy of semantic edge detection to a certain level for moving towards a holistic scene interpretation.

Early work tends to treat edge information as low-level cues to enhance other applications. However, the availability of large training data and the progress in deep learning methods have allowed one to make significant progress for the edge detection problem in the last few years. In particular, there have been newer data sets [18]. The availability of large-scale semantic segmentation data sets [8] can also be easily processed to obtain semantic edge data set as these two problems can be seen as dual problems.

1.1. Related works

The definition of boundary or edge detection has evolved over time from low-level to high-level features: simple edge filters [5], depth edges [17], object boundaries [40], and semantic contours [18]. In some sense, the evolution of edge detection algorithms captures the progress in computer vision from simple convolutional filters such as Sobel [29] or Canny [5] to fully developed deep neural networks.

Low-level edges Early edge detection methods used simple convolutional filters such as Sobel [29] or Canny [5].

Depth edges Some previous work focuses on labeling contours into convex, concave, and occluding ones from synthetic line drawings [38] and real world images under restricted settings [14, 17]. Indoor layout estimation can also be seen as the identification of concave boundaries (lines folding walls, ceilings, and ground) [20]. By recovering occluding boundaries [22], it was shown that the depth ordering of different layers in the scene can be obtained.

Perceptual edges A wide variety of methods are driven towards the extraction of perceptual boundaries [40]. Dollar *et al.* [9] used boosted decision trees on different patches to extract edge maps. Lim *et al.* [33] computed sketch tokens which are object boundary patches using random forests. Several other edge detection methods include statistical edges [31], multi-scale boundary detection [43], and point-wise mutual information (PMI) detector [25]. More recently, Dollar and Zitnick [10] proposed a realtime fast edge detection method using structured random forests. Latest methods [3, 30, 50, 51] using deep neural networks have pushed the detection performance to state-of-the-art.

Semantic edges The origin of semantic edge detection can be possibly pinpointed to [42]. As a high level task, it has also been used implicitly or explicitly in many problems related to segmentation [49] and reconstruction [21]. In some sense, all semantic segmentation methods [7, 8, 12, 16, 35, 36, 41, 45, 48] can be loosely seen as semantic edge detection since one can easily obtain edges, although not necessarily an accurate one, from the segmentation results. There are papers that specifically formulate the problem statement as binary or category-aware semantic edge detection [3, 4, 13, 18, 28, 39, 42, 51]. Hariharan *et al.* [18] introduced the Semantic Boundaries Dataset (SBD) and proposed inverse detector which combines both bottom-up edge and top-down detector information to detect category-aware semantic edges. HFL [3] first uses VGG [47] to locate binary semantic edges and then uses deep semantic segmentation networks such as FCN [36] and DeepLab [7] to obtain category labels. The framework, however, is not end-to-end trainable due to the separated prediction process.

DNNs for edge detection Deep neural networks recently became popular for edge detection. Related work includes SCT based on sparse coding [37], N^4 fields [15], deep contour [46], deep edge [2], and CSCNN [23]. One notable method is the holistically-nested edge detection (HED) [50] which trains and predicts edges in an image-to-image fashion and performs end-to-end training.

1.2. Contributions

Our work is related to HED in adopting a nested architecture but we extend the work to the more difficult category-aware semantic edge detection problem. Our main contributions in this paper are summarized below:

- To address edge categorization, we propose a multi-label learning framework which allows improved edge learning than traditional multi-class framework.
- We propose a novel nested architecture without deep supervision on ResNet [19], where bottom features are only used to augment top classifications. We show that deep supervision may not be beneficial in our problem.
- We outperform previous state-of-the-art methods by significant margins on SBD and Cityscapes datasets.

2. Problem Formulation

Given an input image, our goal is to compute the semantic edge maps corresponding to pre-defined categories. More formally, for an input image $\mathbf{I}$ and K defined semantic categories, we are interested in obtaining K edge maps $\{\mathbf{Y}_1, \cdots, \mathbf{Y}_K\}$, each having the same size as $\mathbf{I}$. With a network having the parameters $\mathbf{W}$, we denote $\mathbf{Y}_k(\mathbf{p}|\mathbf{I}, \mathbf{W}) \in [0, 1]$ as the network output indicating the computed edge probability on the k-th semantic category at pixel $\mathbf{p}$.

2.1. Multi-label loss function

Possibly driven by the multi-class nature of semantic segmentation, several related works on category-aware semantic edge detection have more or less looked into the problem from the multi-class learning perspective. Our intuition is that this problem by nature should allow one pixel belonging to multiple categories simultaneously, and should be addressed by a multi-label learning framework.

We therefore propose a multi-label loss. Suppose each image $\mathbf{I}$ has a set of label images $\{\bar{\mathbf{Y}}_1, \cdots, \bar{\mathbf{Y}}_K\}$, where $\bar{\mathbf{Y}}_k$ is a binary image indicating the ground truth of the k-th class semantic edge. The multi-label loss is formulated as:

$$\begin{aligned}\mathcal{L}(\mathbf{W}) &= \sum_k \mathcal{L}_k(\mathbf{W}) \\ &= \sum_k \sum_{\mathbf{p}} \{-\beta \bar{\mathbf{Y}}_k(\mathbf{p}) \log \mathbf{Y}_k(\mathbf{p}|\mathbf{I}; \mathbf{W}) \\ &\quad - (1-\beta)(1-\bar{\mathbf{Y}}_k(\mathbf{p})) \log(1-\mathbf{Y}_k(\mathbf{p}|\mathbf{I}; \mathbf{W}))\},\end{aligned} \tag{1}$$

where β is the percentage of non-edge pixels in the image to account for skewness of sample numbers, similar to [50].

3. Network Architecture

We propose CASENet, an end-to-end trainable convolutional neural network (CNN) architecture (shown in Fig. 3(c)) to address category-aware semantic edge detection. Before describing CASENet, we first propose two alternative network architectures which one may come up with straightforwardly given the abundant previous literature on edge detection and semantic segmentation. Although both architectures can also address our task, we will analyze issues associated with them, and address these issues by proposing the CASENet architecture.

3.1. Base network

We address the edge detection problem under the fully convolutional network framework. We adopt ResNet-101 by removing the original average pooling and fully connected layer, and keep the bottom convolution blocks. We further modify the base network in order to better preserve low-level edge information. We change the stride of the first and fifth convolution blocks ("res1" and "res5" in Fig. 3) in ResNet-101 from 2 to 1. We also introduce dilation factors to subsequent convolution layers to maintain the same receptive field sizes as the original ResNet, similar to [19].

3.2. Basic architecture

A very natural architecture one may come up with is the Basic architecture shown in Fig. 3(a). On top of the base network, we add a classification module (Fig. 3(d)) as a 1×1 convolution layer, followed by bilinear up-sampling (implemented by a K-grouped deconvolution layer) to produce a set of K activation maps $\{\mathbf{A}_1, \cdots, \mathbf{A}_K\}$, each having the same size as the image. We then model the probability of a pixel belonging to the k-th class edge using the sigmoid unit given by $\mathbf{Y}_k(\mathbf{p}) = \sigma(\mathbf{A}_k(\mathbf{p}))$, which is presented in the Eq. (1). Note that $\mathbf{Y}_k(\mathbf{p})$ is not mutually exclusive.

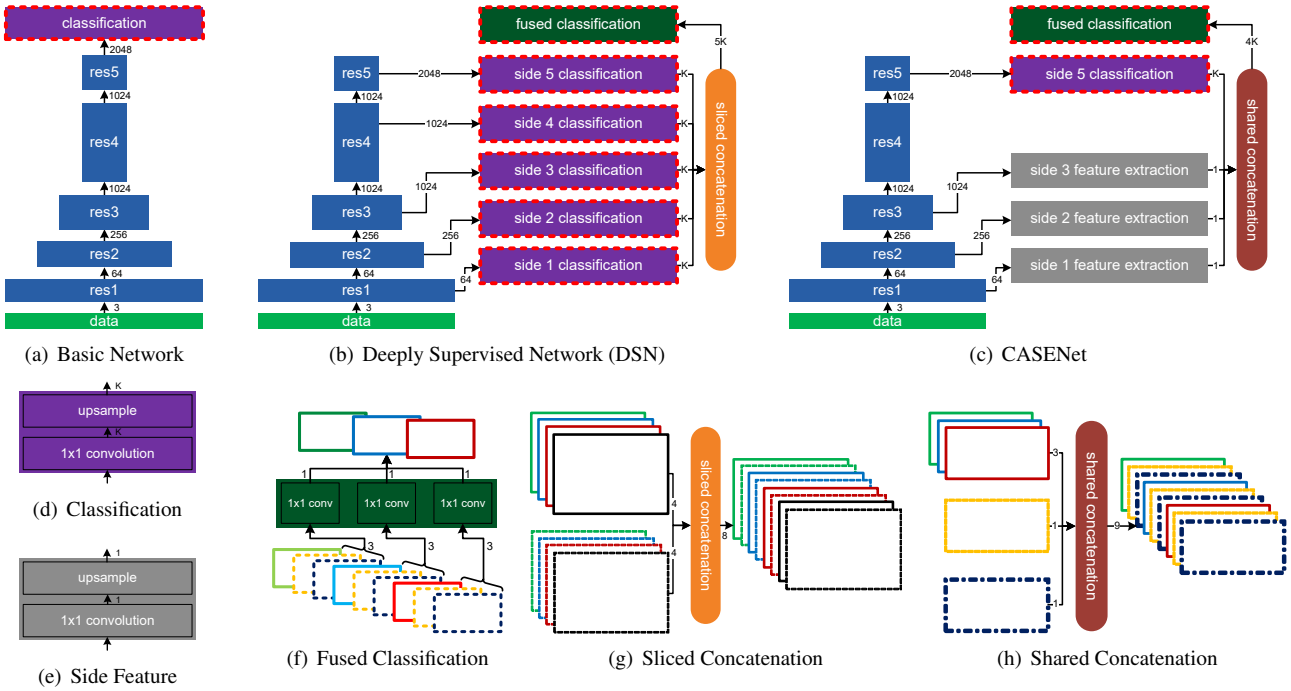
3.3. Deeply supervised architecture

One of the distinguishing features of the holistically-nested edge detection (HED) network [50] is the nested architecture with deep supervision [32]. The basic idea is to impose losses to bottom convolution sides besides the top network loss. In addition, a fused edge map is obtained by supervising the linear combination of side activations.

Note that HED only performs binary edge detection. We extended this architecture to handle K channels for side outputs and K channels for the final output. We refer to this as deeply supervised network (DSN), as depicted in Fig. 3(b). In this network, we connect an above-mentioned classification module to the output of each stack of residual blocks, producing 5 side classification activation maps $\{\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(5)}\}$, where each of them has K-channels. We then fuse these 5 activation maps through a sliced concatenation layer (the color denotes the channel index in Fig. 3(g)) to produce a $5K$-channel activation map:

$$\mathbf{A}^f = \{\mathbf{A}_1^{(1)}, \ldots, \mathbf{A}_1^{(5)}, \mathbf{A}_2^{(1)}, \ldots, \mathbf{A}_2^{(5)}, \ldots, \mathbf{A}_K^{(5)}\} \tag{2}$$

$\mathbf{A}^f$ is fed into our fused classification layer which performs K-grouped 1×1 convolution (Fig. 3(f)) to produce a K-channel activation map $\mathbf{A}^{(6)}$. Finally, 6 loss functions are computed on $\{\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(6)}\}$ using the Equation 1 to provide deep supervision to this network.



(a) Basic Network (b) Deeply Supervised Network (DSN) (c) CASENet

(d) Classification

(e) Side Feature

(f) Fused Classification (g) Sliced Concatenation (h) Shared Concatenation

Figure 3. Three CNN architectures designed in this paper are shown in (a)-(c). A solid rectangle represents a composite block of CNN layers. Any decrease of its width indicates a drop of spatial resolution of this block's output feature map by a factor of 2. A number besides an arrow indicates the number of channels of the block's output features. A blue solid rectangle is a stack of ResNet blocks. A purple solid rectangle is our classification module. A dotted red outline indicates that block's output is supervised by our loss function in equation 1. A gray solid rectangle is our side feature extraction module. A dark green solid rectangle is our fused classification module performing K-grouped 1×1 convolution. (d)-(h) depicts more details of various modules used in (a)-(c), where outlined rectangles illustrate input and output feature maps. Best viewed in color.

Note that the reason we perform sliced concatenation in conjunction with grouped convolution instead of the corresponding conventional operations is as follows. Since the 5 side activations are supervised, we implicitly constrain each channel of those side activations to carry information that is most relevant to the corresponding class.

With sliced concatenation and grouped convolution, the fused activation for a pixel $\mathbf{p}$ is given by:

$$\mathbf{A}_k^{(6)}(\mathbf{p}) = W_k^T[\mathbf{A}_k^{(1)}(\mathbf{p})^T, \cdots, \mathbf{A}_k^{(5)}(\mathbf{p})^T] \qquad (3)$$

This essentially integrates corresponding class-specific activations from different scales as the finally fused activations. Our experiments empirically support this design choice.

3.4. CASENet architecture

Upon reviewing the Basic and DSN architectures, we notice several potential associated issues in the category-aware semantic edge detection task:

First, the receptive field of the bottom side is limited. As a result it may be unreasonable to require the network to perform semantic classification at an early stage, given that context information plays an important role in semantic classification. We believe that semantic classification should rather happen on top where features are encoded with high-level information.

Second, bottom side features are helpful in augmenting top classifications, suppressing non-edge pixels and providing detailed edge localization and structure information. Hence, they should be taken into account in edge detection.

Our proposed CASENet architecture (Fig. 3(c)) is motivated by addressing the above issues. The network adopts a nested architecture which to some extent shares similarity to DSN but also contains several key improvements. We summarize these improvements below:

1. Replace the classification modules at bottom sides to the feature extraction modules.

2. Put the classification module and impose supervision only at the top of the network.

3. Perform shared concatenation (Fig. 3(h)) instead of sliced concatenation.

The difference between side feature extraction and side classification is that the former only outputs a single channel feature map $\mathbf{F}^{(j)}$ rather than K class activations. The

shared concatenation replicates the bottom features $\mathbf{F} = \{\mathbf{F}^{(1)}, \mathbf{F}^{(2)}, \mathbf{F}^{(3)}\}$ from Side-1-3 to separately concatenate with each of the K top activations:

$$\mathbf{A}^f = \{\mathbf{F}, \mathbf{A}_1^{(5)}, \mathbf{F}, \mathbf{A}_2^{(5)}, \mathbf{F}, \mathbf{A}_3^{(5)}, \ldots, \mathbf{F}, \mathbf{A}_K^{(5)}\}. \quad (4)$$

The resulting concatenated activation map is again fed into the fused classification layer with K-grouped convolution to produce a K-channel activation map $\mathbf{A}^{(6)}$.

In general, CASENet can be thought of as a joint edge detection and classification network by letting lower level features participating and augmenting higher level semantic classification through a skip-layer architecture.

4. Experiments

In this paper, we compare CASENet[1] with previous state-of-the-art methods, including InvDet [18], HFL [3], weakly supervised object boundaries [28], as well as several baseline network architectures.

4.1. Datasets

We evaluate the methods on SBD [18], a standard dataset for benchmarking semantic edge detection. Besides SBD, we also extend our evaluation to Cityscapes [8], a popular semantic segmentation dataset with pixel-level high quality annotations and challenging street view scenarios. To the best of our knowledge, our paper is the first work to formally report semantic edge detection results on this dataset.

SBD The dataset consists of 11355 images from the PASCAL VOC2011 [11] trainval set, divided into 8498 training and 2857 test images[2]. This dataset has semantic boundaries labeled with one of 20 Pascal VOC classes.

Cityscapes The dataset contains 5000 images divided into 2975 training, 500 validation and 1525 test images. Since the labels of test images are currently not available, we treat the validation images as test set in our experiment.

4.2. Evaluation protocol

On both SBD and Cityscapes, the edge detection accuracy for each class is evaluated using the official benchmark code and ground truth from [18]. We keep all settings and parameters as default, and report the maximum F-measure (MF) at optimal dataset scale (ODS), and average precision (AP) for each class. Note that for Citiscapes, we follow [18] exactly to generate ground truth boundaries with single pixel width for evaluation, and reduce the sizes of both ground truth and predicted edge maps to half along each dimension considering the speed of evaluation.

[1]Source code available at: http://www.merl.com/research/license#CASENet.

[2]There has been a clean up of the dataset with a slightly changed image number. We also report the accordingly updated InvDet results.

4.3. Implementation details

We trained and tested CASENet, HED [50], and the proposed baseline architectures using the *Caffe* library [26].

Training labels Considering the misalignment between human annotations and true edges, and the label ambiguity of pixels near boundaries, we generate slightly thicker ground truth edges for network training. This can be done by looking into neighbors of a pixel and seeking any difference in segmentation labels. The pixel is regarded as an edge pixel if such difference exists. In our paper, we set the maximum range of neighborhood to be 2. Under the multi-label framework, edges from different classes may overlap.

Baselines Since several main comparing methods such as HFL and HED use VGG or VGG based architectures for edge detection and categorization, we also adopt the CASENet and other baseline architectures on VGG (denoted as CASENet-VGG). In particular, we remove the max pooling layers after conv4, and keep the resolutions of conv5, fc6 and fc7 the same as conv4 (1/8 of input). Similar to [7], both fc6 and fc7 are treated as convolution layers with 3×3 and 1×1 convolution and dimensions set to 1024. Dilation factors of 2 and 4 are applied to conv5 and fc6.

To compare our multi-label framework with multi-class, we generate ground truth with non-overlapping edges of each class, reweight the softmax loss similar to our paper, and replace the top with a 21-class reweighted softmax loss.

Initialization In our experiment, we initialize the convolution blocks of ResNet/VGG in CASENet and all comparing baselines with models pre-trained on MS COCO [34].

Hyper-parameters We unify the hyper-parameters for all comparing methods with the same base network, and set most of them following HED. In particular, we perform SGD with iteration size of 10, and fix loss weight to be 1, momentum 0.9, and weight decay 0.0005. For methods with ResNet, we set the learning rate, step size, gamma and crop size to $1e-7$ / $5e-8$, 10000 / 20000, 0.1 / 0.2 and 352×352 / 472×472 respectively for SBD and Cityscapes. For VGG, the learning rate is set to $1e-8$ while others remain the same as ResNet on SBD. For baselines with softmax loss, the learning rate is set to 0.01 while other parameters remain the same. The iteration numbers on SBD and Cityscapes are empirically set to 22000 and 40000.

Data augmentation During training, we enable random mirroring and cropping on both SBD and Cityscapes. We additionally augment the SBD data by resizing each image with scaling factors $\{0.5, 0.75, 1.0, 1.25, 1.5\}$, while no such augmentation is performed on Cityscapes.

4.4. Results on SBD

Table 1 shows the MF scores of different methods performing category-wise edge detection on SBD, where CASENet outperforms previous methods. Upon using the benchmark code from [18], one thing we notice is that the recall scores of the curves are not monotonically increasing, mainly due to the fact that post-processing is taken after thresholding in measuring the precision and recall rates. This is reasonable since we have not taken any postprocessing operations on the obtained raw edge maps. We only report the MF on SBD since AP is not well defined under such situation. The readers may kindly refer to supplementary materials for class-wise precision recall curves.

Multi-label or multi-class? We compare the proposed multi-label loss with the reweighted softmax loss under the Basic architecture. One could see that using softmax leads to significant performance degradation on both VGG and ResNet, supporting our motivation in formulating the task as a multi-label learning problem, in contrast to the well accepted concept which addresses it in a multi-class way.

Is deep supervision necessary? We compare CASENet with baselines network architectures including Basic and DSN depicted in Fig. 3. The result empirically supports our intuition that deep supervision on bottom sides may not be necessary. In particular, CASENet wins frequently on per-class MF as well as the final mean MF score. Our observation is that the annotation quality to some extent influenced the network learning behavior and evaluation, leading to less performance distinctions across different methods. Such distinction becomes more obvious on Cityscapes.

Is top supervision necessary? One might further question the necessity of imposing supervision on Side-5 activation in CASENet. We use CASENet$^-$ to denote the same CASENet architecture without Side-5 supervision during training. The improvement upon adding Side-5 supervision indicates that a supervision on higher level side activation is helpful. Our intuition is that Side-5 supervision helps Side-5 focusing more on the classification of semantic classes with less influence from interacting with bottom layers.

Visualizing side activations We visualize the results of CASENet, CASENet$^-$ and DSN on a test image in Fig. 5. Overall, CASENet achieves better detection compared to the other two. We further show the side activations of this testing example in Fig. 6, from which one can see that the activations of DSN on Side-1, Side-2 and Side-3 are more blurry than CASENet features. This may be caused by imposing classification requirements on those layers, which seems a bit aggressive given limited receptive field

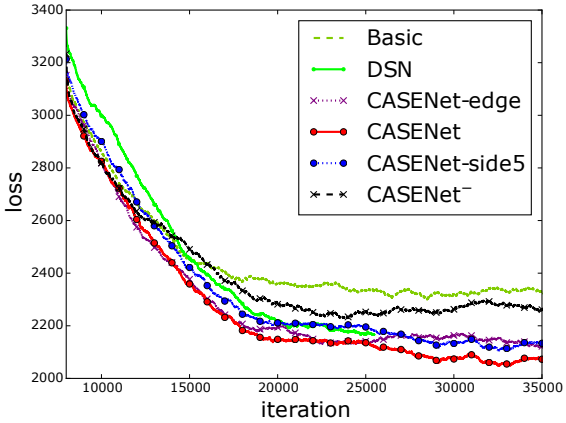


Figure 4. Training losses of different variants of CASENet on the SBD dataset. The losses are respectively moving averaged by a kernel length of 8000. All curves means the final fused losses, except for CASENet-side5 which indicates the loss of Side-5's output. Note that CASENet loss is consistently the smallest.

and information and may caused performance degradation. Also one may notice the differences in "Side5-Person" and "Side5-Boat" between CASENet$^-$ and CASENet, where CASENet's activations overall contain sharper edges, again showing the benefit of Side-5 supervision.

From ResNet to VGG CASENet-VGG in Table 1 shows comparable performance to HFL-FC8. HFL-CRF performs slightly better with the help of CRF postprocessing. The results to some extent shows the effectiveness our learning framework, given HFL uses two VGG networks separately for edge localization and classification. Our method also significantly outperforms the HED baselines from [28], which gives 44 / 41 on MF / AP, and 49 / 45 with detection.

Other variants We also investigated several other architectures. For example, we kept the stride of 2 in "res1". This downgrades the performance for lower input resolution. Another variant is to use the same CASENet architecture but impose binary edge losses (where a pixel is considered lying on an edge as long as it belongs to the edge of at least one class) on Side-1-3 (denoted as CASENet-edge in Fig. 4). However we found that such supervision seems to be a divergence to the semantic classification at Side-5.

4.5. Results on Cityscapes

We also train and test both DSN and CASENet with ResNet as base network on the Cityscapes. Compared to SBD, Cityscapes has relatively higher annotation quality but contains more challenging scenarios. The dataset contains more overlapping objects, which leads to more cases of multi-label semantic boundary pixels and thus may be better to test the proposed method. In Table 1, we provide

Metric	Category	Method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mean
MF (ODS)	Baseline	InvDet	41.5	46.7	15.6	17.1	36.5	42.6	40.3	22.7	18.9	26.9	12.5	18.2	35.4	29.4	48.2	13.9	26.9	11.1	21.9	31.4	27.9
		HFL-FC8	71.6	59.6	68.0	54.1	57.2	68.0	58.8	69.3	43.3	65.8	33.3	67.9	67.5	62.2	69.0	43.8	68.5	33.9	57.7	54.8	58.7
		HFL-CRF	73.9	61.4	74.6	57.2	58.8	70.4	61.6	71.9	46.5	72.3	36.2	71.1	73.0	68.1	70.3	44.4	73.2	42.6	62.4	60.1	62.5
	VGG	Basic-Softmax	67.6	55.3	50.4	44.9	42.3	64.6	61.0	63.9	37.4	43.1	25.3	57.9	57.1	60.0	72.0	33.0	53.5	30.9	54.4	47.7	51.1
		Basic	70.0	58.6	62.5	50.2	51.2	65.4	60.6	66.9	39.7	47.3	31.0	60.1	59.4	60.2	74.4	38.0	56.0	35.9	60.0	53.8	55.1
		CASENet	72.5	61.5	63.8	54.5	52.3	65.4	62.6	67.2	42.6	51.8	31.4	62.0	61.9	62.8	75.4	41.7	59.8	35.8	59.7	50.7	56.8
	ResNet	Basic-Softmax	74.0	64.1	64.8	52.5	52.1	73.2	68.1	73.2	43.1	56.2	37.3	67.4	68.4	67.6	76.7	42.7	64.3	37.5	64.6	56.3	60.2
		Basic	82.5	74.2	80.2	62.3	68.0	80.8	74.3	82.9	52.9	73.1	46.1	79.6	78.9	76.0	80.4	52.4	75.4	48.6	75.8	68.0	70.6
		DSN	81.6	75.6	78.4	61.3	67.6	**82.3**	74.6	82.6	52.4	71.9	45.9	79.2	78.3	76.2	80.1	51.9	74.9	48.0	76.5	66.8	70.3
		CASENet$^-$	83.0	74.7	79.6	61.5	67.7	80.7	74.1	82.8	53.3	**75.0**	44.5	79.8	**80.4**	76.2	80.2	53.2	**77.3**	47.7	75.6	66.3	70.7
		CASENet	**83.3**	**76.0**	**80.7**	**63.4**	**69.2**	81.3	**74.9**	**83.2**	**54.3**	74.8	**46.4**	**80.3**	80.2	**76.6**	**80.8**	**53.3**	77.2	**50.1**	**75.9**	**66.8**	**71.4**

Table 1. Results on the SBD benchmark. All MF scores are measured by %.

Metric	Method	road	sidewalk	building	wall	fence	pole	traffic lgt	traffic sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motorcycle	bike	mean
MF (ODS)	DSN	85.4	76.4	82.6	**51.8**	56.5	66.5	62.6	72.1	80.6	61.1	76.0	77.5	66.3	84.5	**52.3**	67.3	49.4	56.0	76.0	68.5
	CASENet	**86.6**	**78.8**	**85.1**	51.5	**58.9**	**70.1**	**70.8**	**74.6**	**83.5**	**62.9**	**79.4**	**81.5**	**71.3**	**86.9**	50.4	**69.5**	**52.0**	**61.3**	**80.2**	**71.3**
AP	DSN	**78.0**	76.0	83.9	47.9	53.1	67.9	57.9	75.9	79.9	60.2	75.0	75.4	61.0	85.8	**50.6**	67.8	42.5	51.4	72.0	66.4
	CASENet	77.7	**78.6**	**87.6**	**49.0**	**56.9**	**72.8**	**70.3**	**78.9**	**85.1**	**63.1**	**78.4**	**83.0**	**70.1**	**89.5**	46.9	**70.0**	**48.8**	**59.6**	**78.9**	**70.8**

Table 2. Results on the Cityscapes dataset. All MF and AP scores are measured by %.

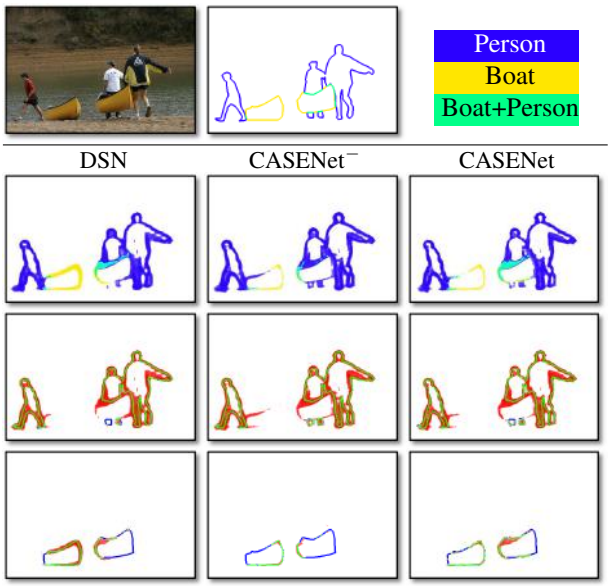


Figure 5. Example results on the SBD dataset. First row: Input and ground truth image and color codes of categories. Second row: Results of different edge classes, where the same color code is used as in Fig. 1. Third row: Results of person edge only. Last row: Results of boat edge only. Green, blue, red and white respectively denote true positive, false negative, false positive and true negative pixels, at the threshold of 0.5. Best viewed in color.

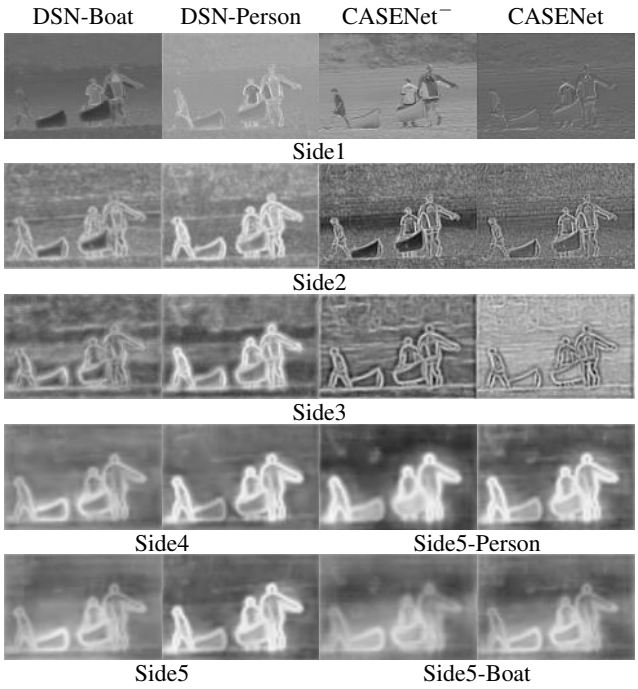


Figure 6. Side activations on the input image of Fig. 5. The first two columns show the DSN's side classification activations corresponding to the class of Boat and Person, respectively. The last two columns show the side features and classification activations for CASENet$^-$ and CASENet, respectively. Note that the pixel value range of each image is normalized to [0,255] individually inside its corresponding side activation outputs for visualization.

both MF and AP of the comparing methods. To the best of our knowledge, this is the first paper quantitatively reporting the detection performance of category-wise semantic edges on Cityscapes. One could see CASENet consistently outperforms DSN in all classes with a significant margin. Besides quantitative results, we also visualize some results in Fig. 7 for qualitative comparisons.

5. Concluding Remarks

In this paper, we proposed an end-to-end deep network for category-aware semantic edge detection. We show that the proposed nested architecture, CASENet, shows improvements over some existing architectures popular in edge detection and segmentation. We also show that the proposed multi-label learning framework leads to better learning behaviors on edge detection. Our proposed method improves over previous state-of-the-art methods with significant margins. In the future, we plan to apply our method to other tasks such as stereo and semantic segmentation.

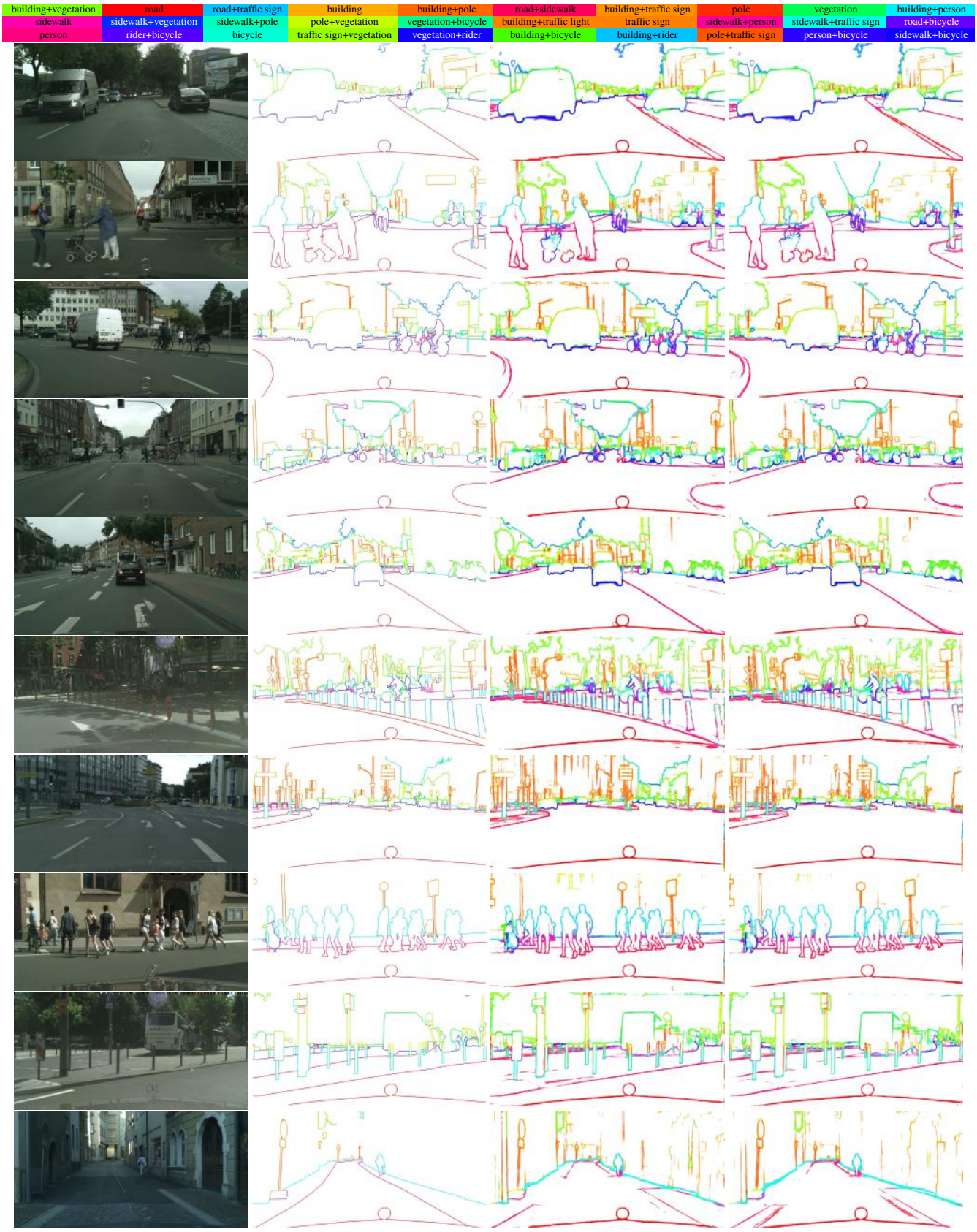


Figure 7. Example results on Cityscapes. Columns from left to right: Input, Ground Truth, DSN and CASENet. CASENet shows better detection qualities on challenging objects, while DSN shows slightly more false positives on non-edge pixels. Best viewed in color.

References

[1] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Trans. PAMI*, 33(5):898–916, 2011. 2

[2] G. Bertasius, J. Shi, and L. Torresani. Deepedge: A multi-scale bifurcated deep network for top-down contour detection. In *CVPR*, 2015. 2

[3] G. Bertasius, J. Shi, and L. Torresani. High-for-low, low-for-high: Efficient boundary detection from deep object features and its applications to high-level vision. In *ICCV*, 2015. 2, 5

[4] G. Bertasius, J. Shi, and L. Torresani. Semantic segmentation with boundary neural fields. In *CVPR*, 2016. 2

[5] J. Canny. A computational approach to edge detection. *IEEE Trans. PAMI*, (6):679–698, 1986. 2

[6] L.-C. Chen, J. T. Barron, K. M. G. Papandreou, and A. L. Yuille. Semantic image segmentation with task-specific edge detection using cnns and a discriminatively trained domain transform. In *CVPR*, 2016. 2

[7] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *ICLR*, 2015. 2, 5

[8] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The Cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 1, 2, 5

[9] P. Dollar, Z. Tu, and S. Belongie. Supervised learning of edges and object boundaries. In *CVPR*, 2006. 2

[10] P. Dollar and C. Zitnick. Structured forests for fast edge detection. In *ICCV*, 2013. 2

[11] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2011 (VOC2011) Results. http://www.pascal-network.org/challenges/VOC/voc2011/workshop/index.html. 5

[12] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *IEEE Trans. PAMI*, 35(8):1915–1929, 2013. 2

[13] V. Ferrari, F. Jurie, and C. Schmid. From images to shape models for object detection. *Int. Journal of Computer Vision*, 87(3):284–303, 2010. 2

[14] D. F. Fouhey, A. Gupta, and M. Hebert. Unfolding an indoor origami world. In *ECCV*, 2014. 2

[15] Y. Ganin and V. Lempitsky. Nˆ 4-fields: Neural network nearest neighbor fields for image transforms. In *ACCV*, 2014. 2

[16] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 2

[17] S. Gupta, P. Arbeláez, and J. Malik. Perceptual organization and recognition of indoor scenes from rgb-d images. In *CVPR*, 2013. 2

[18] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik. Semantic contours from inverse detectors. In *ICCV*, 2011. 1, 2, 5, 6

[19] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3

[20] V. Hedau, D. Hoiem, and D. Forsyth. Recovering the spatial layout of cluttered rooms. In *ICCV*, 2009. 2

[21] D. Hoiem, A. A. Efros, and M. Hebert. Geometric context from a single image. In *ICCV*, 2005. 2

[22] D. Hoiem, A. A. Efros, and M. Hebert. Recovering occlusion boundaries from an image. *Int. Journal of Computer Vision*, 91(3):328–346, 2011. 2

[23] J. Hwang and T.-L. Liu. Pixel-wise deep learning for contour detection. In *ICLR*, 2015. 2

[24] S. L. in Urban Canyons using Omni-Images. S. ramalingam and s. bouaziz and p. sturm and m. brand. In *IROS*, 2010. 2

[25] P. Isola, D. Zoran, D. Krishnan, and E. H. Adelson. Crisp boundary detection using pointwise mutual information. In *ECCV*, 2014. 2

[26] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM Multimedia*, 2014. 5

[27] K. Karsch, Z. Liao, J. Rock, J. T. Barron, and D. Hoiem. Boundary cues for 3d object shape recovery. In *CVPR*, 2013. 2

[28] A. Khoreva, R. Benenson, M. Omran, M. Hein, and B. Schiele. Weakly supervised object boundaries. In *CVPR*, 2016. 2, 5, 6

[29] J. Kittler. On the accuracy of the sobel edge detector. *Image and Vision Computing*, 1(1):37–42, 1983. 2

[30] I. Kokkinos. Pushing the boundaries of boundary detection using deep learning. 2016. 2

[31] S. Konishi, A. L. Yuille, J. M. Coughlan, and S. C. Zhu. Statistical edge detection: Learning and evaluating edge cues. *IEEE Trans. PAMI*, 25(1):57–74, 2003. 2

[32] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu. Deeply supervised nets. In *AISTATS*, 2015. 3

[33] J. Lim, C. Zitnick, and P. Dollar. Sketch tokens: A learned mid-level representation for contour and object detection. In *CVPR*, 2013. 2

[34] T.-Y. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. 5

[35] M. Y. Liu, S. Lin, S. Ramalingam, and O. Tuzel. Layered interpretation of street view images. In *RSS*, 2015. 2

[36] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 2

[37] M. Maire, S. X. Yu, and P. Perona. Reconstructive sparse code transfer for contour detection and semantic labeling. In *ACCV*, 2014. 1, 2

[38] J. Malik. Interpreting line drawings of curved objects. *Int. Journal of Computer Vision*, 1(1):73–103, 1987. 2

[39] K. K. Maninis, J. Pont-Tuset, P. Arbeláez, and L. V. Gool. Convolutional oriented boundaries. In *ECCV*, 2016. 2

[40] D. R. Martin, C. C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Trans. PAMI*, 26(5):530–549, 2004. 2

[41] P. Pinheiro and R. Collobert. Recurrent convolutional neural networks for scene labeling. In *ICML*, 2014. 2

[42] M. Prasad, A. Zisserman, A. Fitzgibbon, M. P. Kumar, and P. H. Torr. Learning class-specific edges for object detection and segmentation. In *Computer Vision, Graphics and Image Processing*. 2006. 1, 2

[43] X. Ren. Multi-scale improves boundary detection in natural images. In *ECCV*, 2008. 2

[44] Q. Shan, B. Curless, Y. Furukawa, C. Hernandez, and S. Seitz. Occluding contours for multi-view stereo. In *CVPR*, 2014. 2

[45] A. Sharma, O. Tuzel, and M. Y. Liu. Recursive context propagation network for semantic scene labeling. In *NIPS*, 2014. 2

[46] W. Shen, X. Wang, Y. Wang, X. Bai, and Z. Zhang. Deepcontour: A deep convolutional feature learned by positive-sharing loss for contour detection draft version. In *CVPR*, 2015. 2

[47] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 2

[48] R. Vemulapalli, O. Tuzel, M. Y. Liu, and R. Chellapa. Gaussian conditional random field network for semantic segmentation. In *CVPR*, 2016. 2

[49] L. Wang, J. Shi, G. Song, and I. Shen. Object detection combining recognition and segmentation. In *ACCV*, 2007. 2

[50] S. Xie and Z. Tu. Holistically-nested edge detection. In *ICCV*, 2015. 2, 3, 5

[51] J. Yang, B. Price, S. Cohen, H. Lee, and M. H. Yang. Object contour detection with a fully convolutional encoder-decoder network. In *CVPR*, 2016. 2

[52] Z. Yu, W. Liu, W. Liu, X. Peng, Z. Hui, and B. V. Kumar. Generalized transitive distance with minimum spanning random forest. In *IJCAI*, 2015. 2