Semantic Boundary Refinement by Joint Inference from Edges and Regions

Chao Yang
University of Southern California
chaoy@usc.edu

Abstract

We study the problem of detecting boundaries for specific classes of objects. Our approach leverages recent advances in semantic segmentation and bottom-up boundary detection. We propose a mechanism for combining multiple sources of information: predicted segmentation masks, bottom-up contours, and a novel local class-specific boundary detector. These are jointly mapped to final category-specific boundary strength estimated by a trained classifier. In experiments on VOC2012 [7] and Microsoft COCO [15], our method dramatically outperforms recent prior work, for some classes doubling the accuracy of boundary prediction.

1. Introduction

Recognition of semantic categories in images has been one of the core tasks in computer vision since the beginning of the field [22, 4, 9, 18]. The details of tasks associated with recognition have evolved along with the complexity of available computation vision tools, from image classification (does the image contain instances of the category of interest) through bounding-box detection, to category- and instance-level segmentation, where the goal is to label every pixel that belongs to categories of interest. In this latter task, one could devise different measures of accuracy. Recent work has largely focused on measures based on *region* accuracy, measured ultimately as a function of area of overlap. The accuracy of contours or boundaries, on the other hand, has received less attention.

The focus of our work presented in this paper is on *semantic boundary detection*: given an image, the goal is to predict where are the boundaries that separate objects in a category of interest from everything else. This objective combines elements of semantic segmentation (since it is ultimately about category-aware partition of the image) and edge detection (since it deals with finding thin boundary elements rather than labeling regions). Both of these tasks have seen significant advances in recent years [9, 23, 21], and one could consider directly combining predictions from

state of the art semantic segmentation and state of the art non-semantic segmentation to yield semantic boundaries.

We show that such a combination could indeed produce a strong baseline, which for many classes far outperforms the best reported results in the literature. However, we can do better, by refining the information that these two sources (regions and edges) give about category-specific boundaries. The mechanism by which the information is fused is based on joint reasoning over semantic segmentation, bottom-up contours, and local semantic boundary classification; the parameters of this mechanism are learned from the data. A schematic overview of our approach, which we describe in detail in Section 2, is shown in Figure 1.

1.1. Related Work

Non-specific boundary detection There is a rich tradition of bottom-up, non-category specific edge or boundary detection in computer vision. Some recently proposed methods [6, 19] achieve results that approach human level performance, with the standard evaluation protocol measuring precision-recall behavior (like F-measure and average precision). We choose to work with MCG [2, 19] the authors of which made pre-computed boundary maps available for all the data sets we worked with. This approach achieves F-measure of 0.74 on the BSDS test set, compared to F=0.81 for inter-human agreement. However, this is a *non-specific* boundary detector; it is not aware of object categories, and lumps all predicted boundaries together. Consequently, its performance as a category-specific, semantic boundary detector is abysmal: in [9] it is reported to achieve only 4% mean average precision on VOC data. More recently deep learning also significantly advances the state of the art boundary detection performance [23].

Semantic segmentation For many years semantic segmentation was dominated by variations on conditional random fields (CRFs), with increasingly complex high-order potentials [14, 8] and complex pre-processing and multi-stage procedures [3, 24]. Subsequent progress in the field has been associated with approaches using deep convolutional neural network (convnets) in conjunction with region proposals [10]. Even more recently, state of the art has been
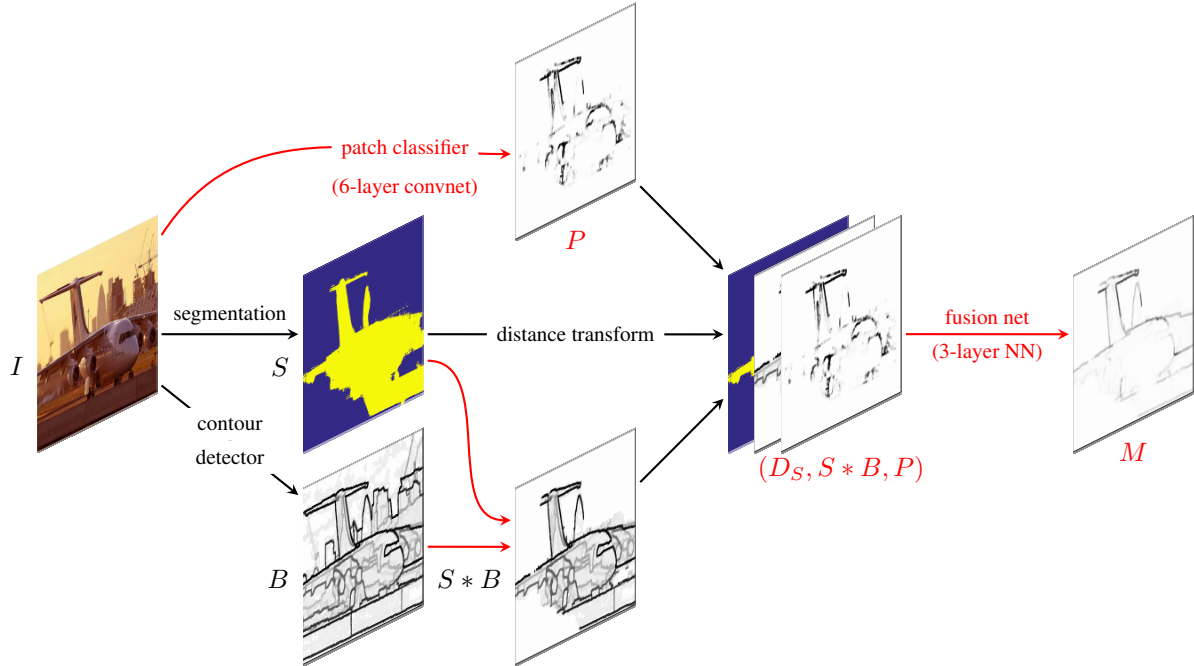
Figure 1. Overview of our semantic boundary detection pipeline. I: input image; S: class-specific figure/ground segmentation; B: non-semantic boundary map; $B * S$: masked boundary baseline; D_S: distance transform computed on edges of S; P: class-specific boundary map computed by a patch classifier (6-layer fully convolutional convnet). The final class-specific boundary map M is obtained by applying a fusion classifier (2-layer neural network) on every pixel of a 3-channel map consisting of D_s, $S * B$ and P.

driven by feed-forward architectures that use the entire hierarchy of features computed in convnets to classify image elements [16, 17, 11, 12], with further improvement sometimes obtained by re-introducing CRFs on top of the convnet-based predictions [25, 4]. In experiments reported in this paper we have worked with the multi-scale version of DeepLab-CRF architecture [5]. For each location in the image, it estimates category scores from features combined from multiple layers of a deep convnet.

Semantic boundary detection There has been relatively little work directed at semantic boundary detection per se. The most notable efforts are the Inverse Detectors method [9], the more recent Situational Object Boundary Detection method [21]. In [9], bottom-up, non-specific boundary predictions are combined with coarse object detections in the form of bounding boxes. Our approach is similar in its use of bottom-up (non-semantic) and top-down (semantic) information, but we use semantic segmentation, rather than bounding box detectors, for the latter. Another difference is that we learn an expressive fusion classifier (neural network) that combines strength of bottom-up contours, distance to segmentation mask and a novel local semantic boundary detector, whereas in [9] the bottom-up contours are simply modulated by the weights predicted from class-specific detectors. In Section 3 we show that our method outperforms both [21] and [9] by a wide margin.

2. Boundary refinement from regions and contours

We build the boundary detector for each class separately. Ultimately, our predictions are computed from three sources of information: semantic segmentation, non-specific (bottom-up) boundary detection, and class-specific local boundary detector. The first two components are based on existing tools developed by others, and the third one is learned. We describe the details for each of these components below.

Figure 2. Segmentation and boundary baselines shown on *Chair* category (Pascal VOC). Left: image I, middle: segmentation mask from DeepLab-CRF B, right: masked boundary $B * S$ with B dilated by 5 pixels.

Segmentation baseline A semantic segmentation system produces hypothesized masks for the category, and we could treat the contours of these masks as semantic boundary prediction. Note that these are hard 0/1 values, so

in evaluation we obtain a single precision/recall point that can be converted to average precision somewhat generously by extrapolating a piece-wise linear graph connecting this point to the zero recall/full precision point.

Masked boundary baseline Non-category-specific, bottom-up boundary detectors produce strong responses for boundaries of many classes (and some interior boundaries), and because of their very low precision for any specific class they can not be used for category-specific detection. This was confirmed empirically in [9] for the detector in [1], and again in [21] for the detector in [6]; we observe this to be the case for our detector of choice [19] as well. However, we can attempt to boost the precision by masking the boundary map with class-specific segmentation mask, possibly dilated and/or with interior portion removed, to allow some "slack" in boundary localization. As we report in Section 3, this indeed produces a much improved semantic boundary detector. Figure 2 illustrates these baselines for a few images.

Local class-specific boundary detector Although as we show in Section 3 the masked boundary baseline achieves results that on some classes outperform previous state of the art, it leaves much to be desired. Instead, we can train a class-specific boundary predictor from scratch. We implement it as a multi-layer convnet, trained to classify image patches as boundary or non-boundary for the class at hand.

The input to the patch classifier consists of a 35×35 region with three color channels. Details of the network architecture are summarized in Table 1. Training examples are sampled from regions retained by the $S * B$ masked boundary. The network is trained separately for each class.

	layer type	RF size	#units	stride
1	conv	4	96	1
2	max-pool	2		2
3	conv	3	256	1
4	max-pool	2		2
5	conv	4	64	1
7	ip		256	
8	ip		256	
9	ip		2	

Table 1. Architecture of the local boundary detector; "ip" stands for "inner product" (fully connected layers). The last ip layer is the classifier.

Information fusion The three components described above each captures a different aspect of visual information. Instead of choosing one of these components, we can learn a fusion mechanism that will combine their prediction. The simplest form of such a mechanism would be a linear classifier, which assigns a weight to each channel. We found that (modestly, but consistently) better results are obtained by a non-linear classifier. Let $\mathbf{x} = (d, b, c)$ be the triplet representing the input to the classifier at a pixel, where d is the

distance to nearest point on the boundary of segmentation mask for the category of interest (or a large default value if no such segmentation is present); b is the value of non-class-specific boundary map; and c is the boundary probability estimated by the local boundary detector. Then, our classifier predicts

$$\log \Pr(m = 1|\mathbf{x}) \propto \mathbf{w}_3^T \sigma(\mathbf{w}_2^T \sigma(\mathbf{w}_1^T \mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2) + b_3,$$

that is, it is a three-layer feed-forward network with fully connected layers and sigmoid activation functions. We use 64 units in the first hidden layer (i.e., $\mathbf{w}_1 \in \mathbb{R}^{64}$) and 16 units in the second layer. We can also express it as a convnet, with filters of size 1×1 throughout, and carry out the feedforward computation efficiently using standard tools for fully convolutional networks.

The entire pipeline for our semantic boundary detector is illustrated in Figure 1.

3. Experiments

3.1. Results on VOC 2012

Following the SBD (semantic boundary detection) evaluation protocol in [9, 21] we used the 8,498 images as SBD training set, and 2,820 as SBD test set. The remaining 676 images in VOC2012 were used as a validation/tuning set on which we evaluated models and tuned parameters.

The bottom-up MCG boundary detector was trained on BSDS, and used by us without modification. For the semantic segmentation model we initialized with VGG16 [20] and trained DeepLab-CRF on the SBD training set. After evaluating different parameter settings on the tuning set, we set the dilation radius for the segmentation mask to 5 pixels, and suppressed bottom-up contours with values below 0.05. Finally, we used Caffe [13] to implement and apply the local patch-classifier and the fusion network. The patch-classifier is trained per-class, taking approximately 5 hours until convergence on a Titan X GPU.

Method	mean average precision
Inverse detectors [9]	19.9
Situational detectors [21]	31.6
Segmentation only S	45.8
Masked boundary $S * B$	46.0
Local patch classifier P	26.9
Fusion $(D_s, S * B, P)$	51.8

Table 2. Mean average precision for VOC2012 data set.

We evaluated average precision of different semantic boundary detectors per class, using the methodology and code from [9]. To make results comparable, we enabled the default behavior which ignores interior contours (inside the ground truth masks for each class); this is somewhat
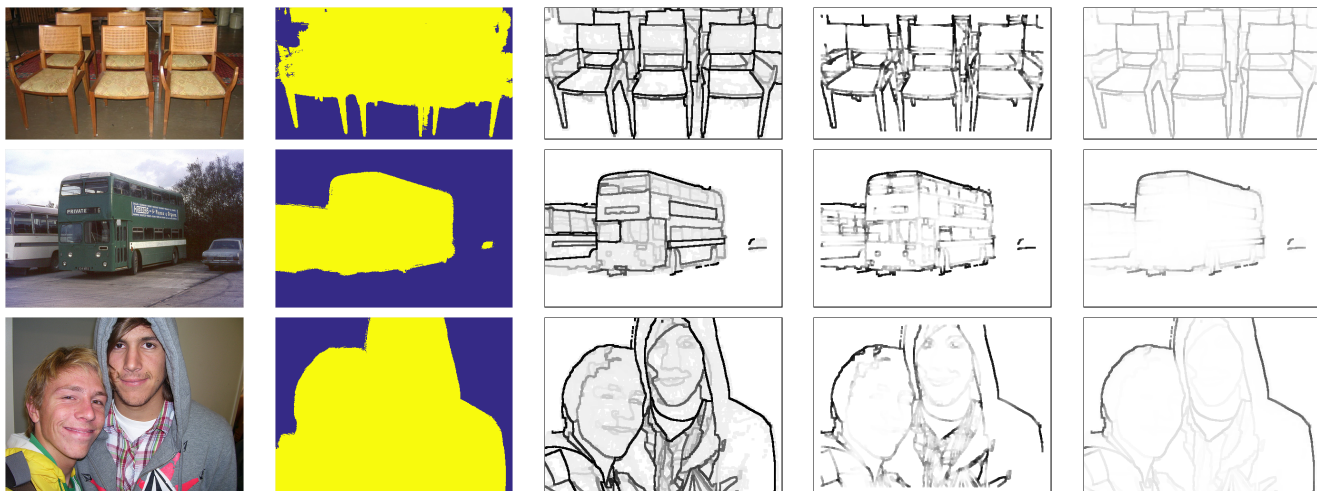
Figure 3. Examples of results with our pipeline on images in SBD test set. From left: input image, category-specific segmentation mask B, masked boundary $B * S$, patch classifier output P, and our final result with the fusion network. Target categories, from top: chair, bus, person, dog.

unfavorable to our method which, as the examples in Figures 3 demonstrate, is particularly good at suppressing interior contours. The main result we report is the average precision (AP).

The AP values averaged over 20 object categories are reported in Table 2.In all classes except bicycle, our results are significantly better, sometimes almost doubling the AP. We hypothesize that bicycle includes many cases of thin structures, which are hard to capture by semantic segmentation and consequently by our fusion technique.

3.2. Results on Microsoft COCO

Following the setup in [21] for COCO, we used the first 5,000 images from the `val` set as the test set for our experiments. We used the same bottom-up MCG contour model as in the VOC experiments; the 80-category DeepLab-CRF segmentation model trained on COCO `train` was obtained from the authors of [4]. In Table 3 we report the results of evaluating components of our system in this setup. The relative standing reflects that in Table 2, but all the numbers are lower, reflecting the increased difficulty of COCO compared to VOC. Another factor that may influence the precision is the quality of the ground truth – the COCO segmentation labeling is rough comparing with VOC.

Method	mean average precision
Segmentation only S	36.9
Masked boundary $S * B$	38.5
Local patch classifier P	22.4
Fusion $(D_s, S * B, P)$	45.0

Table 3. Mean average precision for Microsoft COCO data set.

Neither [9] nor [21] includes per-class AP results on COCO. However, [21] includes results of experiments (both on COCO and on VOC2012) in which ground truth

boundaries for all categories are pooled, effectively defining a class-agnostic, foreground (any class) vs. background boundary detection tasks. These are evaluated against "class-agnostic" situational detectors that are trained to predict such boundaries. We can emulate this experiment by max-pooling the predictions of our category-specific detectors for each pixel, forming a class-agnostic boundary detection map. A comparison of AP figures with our method and that of [21] is shown in Table 4. It is clear that on this task, too, our fusion method performs significantly better than the prior work.

Method	VOC2012	COCO
Situational detectors [21]	42.6	43.4
Fusion $(D_s, S * B, P)$	61.2	58.9

Table 4. Average precision on class-agnostic object boundary detection task (all object boundaries, and predictors for all categories, pooled together).

4. Conclusions

We have proposed an architecture that fuses region- and edge-based information for category-specific semantic boundary detection which significantly improves boundary prediction results. It is important to note that we can plug in any boundary detection and semantic segmentation method into our fusion framework. Given recent advances in both venues [23, 22, 5], we can potentially improve the results further. Another intriguing question that arises from our findings is whether better category-specific boundary predictions can help improve region prediction (semantic segmentation). We plan to explore these aspects in the future.

References

[1] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE TPAMI*, 33(5), 2011. 3

[2] P. Arbeláez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. In *CVPR*, 2014. 1

[3] J. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu. Semantic segmentation with second-order pooling. In *ECCV*. 2012. 1

[4] 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*, 2014. 1, 2, 4

[5] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv preprint arXiv:1606.00915*, 2016. 2, 4

[6] P. Dollár and C. L. Zitnick. Structured forests for fast edge detection. In *ICCV*, 2013. 1, 3

[7] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, 2015. 1

[8] J. M. Gonfaus, X. Boix, J. Van de Weijer, A. D. Bagdanov, J. Serrat, and J. Gonzalez. Harmony potentials for joint classification and segmentation. In *CVPR*, 2010. 1

[9] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik. Semantic contours from inverse detectors. In *ICCV*, 2011. 1, 2, 3, 4

[10] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In *ECCV 2014*, 2014. 1

[11] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *CVPR*, 2015. 2

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 2

[13] 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 *Proceedings of the ACM International Conference on Multimedia*, 2014. 3

[14] L. Ladický, C. Russell, P. Kohli, and P. H. S. Torr. Associative hierarchical CRFs for object class image segmentation. In *ICCV*, 2009. 1

[15] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*. 2014. 1

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

[17] M. Mostajabi, P. Yadollahpour, and G. Shakhnarovich. Feedforward semantic segmentation with zoom-out features. In *CVPR*, 2015. 2

[18] G. Papandreou, L.-C. Chen, K. Murphy, and A. L. Yuille. Weakly- and semi-supervised learning of a dcnn for semantic image segmentation. *arXiv preprint arXiv:1502.02734*, 2015. 1

[19] J. Pont-Tuset, P. Arbeláez, J. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping for image segmentation and object proposal generation. In *arXiv:1503.00848*, March 2015. 1, 3

[20] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint http://arxiv.org/abs/1409.1556*, 2014. 3

[21] J. Uijlings and V. Ferrari. Situational object boundary detection. *arXiv preprint arXiv:1504.06434*, 2015. 1, 2, 3, 4

[22] Z. Wu, C. Shen, and A. v. d. Hengel. High-performance semantic segmentation using very deep fully convolutional networks. *arXiv preprint arXiv:1604.04339*, 2016. 1, 4

[23] S. Xie and Z. Tu. Holistically-nested edge detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1395–1403, 2015. 1, 4

[24] P. Yadollahpour, D. Batra, and G. Shakhnarovich. Discriminative re-ranking of diverse segmentations. In *CVPR*, 2013. 1

[25] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. Torr. Conditional random fields as recurrent neural networks. *arXiv preprint arXiv:1502.03240*, 2015. 2