Pushing the Boundaries of Boundary Detection using Deep Learning

Iasonas Kokkinos, CentraleSupelec & INRIA.
http://cvn.ecp.fr/iasonas/deepboundaries.html

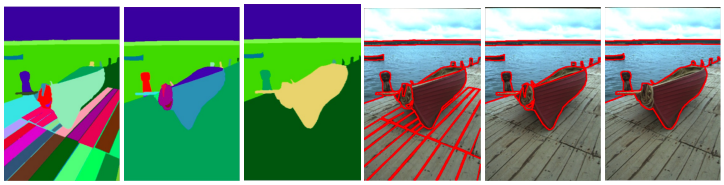

Figure 1: Ground-truth segmentations provided by different annotators for an image from the BSD dataset, and associated boundary maps. The evident lack of agreement among humans is reflected in a low F-measure of human annotators on the task, $F = 0.803$. Our system delivers $F = 0.813$.

Method	Baseline	MIL	G-DSN	M-Scale	VOC	Grouping
ODS	0.778	0.786	0.789	0.803	0.809	0.813
OIS	0.796	0.808	0.811	0.820	0.827	0.831
AP	0.804	0.802	0.789	0.848	0.861	0.866

Table 1: Improvements obtained in this work over our own reproduction of a HED-type baseline; each column builds on the previous improvements.

Image segmentation is an ill-posed problem, since multiple solutions can be considered plausible depending on the task at hand. This is reflected in the inconsistency of human segmentations, illustrated in Fig. 1. When evaluated on the test set of Berkeley Segmentation Dataset (BSD) humans have an F-measure of 0.803 Arbelaez et al. [1], indicating the task's difficulty.

Progress in boundary detection has been consistently narrowing the gap between human and machine performance. The Holistic Edge Detection approach of Xie and Tu [9] dramatically improved the F-measure of boundary detection from 0.75 to 0.78, while requiring only 0.4 seconds on the GPU; additional dataset augmentation yielded an F-measure of 0.79.

Our system, originally introduced in [6] is the first to yield an F-measure on the BSD dataset that is higher than that of humans: when using a common threshold for the whole dataset (Optimal Dataset Scale -ODS) our system's F-measure equals $F = 0.813$, while when an oracle sets the threshold per image (Optimal Image Scale -OIS) we obtain $F = 0.831$. Our detector is fully integrated in Caffe and processes a 321x481 image in less than a second. A preliminary version of our boundary detection system has been made publicly available from the author's website.

Our starting point is the 'Holistic Edge Detection' (HED) work of Xie and Tu [9], which uses 'Deep Supervised Network' (DSN) [7] training to fine-tune the VGG network for the task of boundary detection. Using the notation of [9], we have a training set $S = (X_n, Y_n), n = 1,\dots,N$ with X_n being the input image and $Y_n = \{y_j^{(n)}, j = 1,\dots,|X_n|\}, y_j^{(n)} \in \{0,1\}$ being the predicted labels. We drop the n subscript for brevity.

We consider a multi-layer network, represented in terms of the union of its individual layer parameters, $\mathbf{W}$, to which we append a set of per-layer 'side' parameters $\mathbf{w}^{(1)},\dots\mathbf{w}^{(5)}$, treating the first five convolutional layers of VGG, and fusion weights $\mathbf{h}$. The training objective of HED is:

$$\mathcal{L}_{HED}(\mathbf{W},\mathbf{w},\mathbf{h}) = \mathcal{L}_{side}(\mathbf{W},\mathbf{w}) + \mathcal{L}_{fuse}(\mathbf{W},\mathbf{w},\mathbf{h}), \quad \text{where} \qquad (1)$$

$$\mathcal{L}_{side}(\mathbf{W},\mathbf{w}) = \sum_{m=1}^{M}\sum_{j\in Y} w_{\hat{y}_j} S(\hat{y}_j, s_j^m),\ \mathcal{L}_{fuse}(\mathbf{W},\mathbf{w},\mathbf{h}) = \sum_{j\in Y} w_{\hat{y}_j} S(\hat{y}_j, \sum_{m=1}^{M} h_m s_j^m).$$

In the expressions above j ranges over the image domain, y_j is the ground-truth label, $w_{\hat{y}_j}$ is a class-dependent weight and S is the cross-entropy loss. The side layers provide $M = 5$ complementary estimates for the presence of the boundary $s_j^1,\dots,s_j^m$, computed as inner products between neuron activations and discriminatively trained weights, $\mathbf{W}^m$; the fusion layer learns to combine these estimates into a global decision $\sum_{m=1}^{M} h_m s_j^m$.

Having outlined the HED framework, we now turn to our contributions, consisting in (i) Multiple Instance Learning for boundary detection (ii)

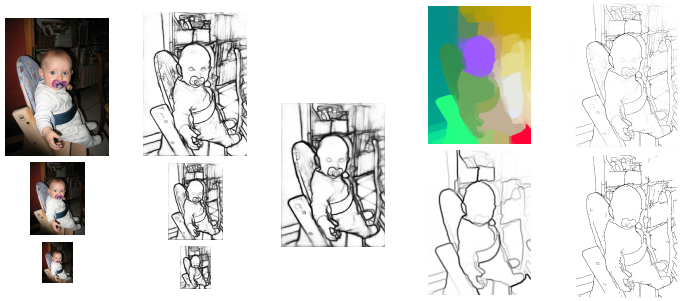

Figure 2: Overview of the main computation stages in our system: an input image is processed at three different scales in order to obtain multi-scale information. The the three scales are fused and sent as input to the Normalized Cuts algorithm, that delivers eigenvectors (we show the first three of eight dimensions as an RGB image) and the resulting 'Spectral Boundaries'. The latter are fused with the original boundary map, nonmaximum suppressed, and optionally thresholded (bottom row). All stages are implemented in Caffe, requiring less than a second on an Nvidia Titan GPU.

Graduated Deep Supervision (iii) Multi-Scale training, (iv) introducing external data, (v) combining CNNs with Normalized Cuts. The improvements due to these contributions are summarized in Table 1, where we report our ODS- and OIS-based F-measures on the BSD test set, alongside with the average precision (AP). We compare to our home-trained HED baseline that yields a performance just slightly below that of [9].

Dealing with annotation inconsistencies using MIL: The first of our contributions aims at dealing with the inconsistency of human annotations in the BSD, illustrated in Fig. 3. As can be seen, even if the two annotators agree about the semantics (a tiger in water), they may not place the boundaries at a common location. This makes it challenging to define 'positive' and 'negative' training samples in the vincinity of boundaries.

We therefore adopt a Mutliple Instance Learning (MIL) approach, as in [5], and associate every ground-truth boundary position j with a set of N_j positions and an associated feature 'bag', $\mathcal{X}_j = \{X_{j,1},\dots,X_{j,N_j}\}$. These positions are estimated by identifying the image positions that (i) lie closer to i than any other ground-truth pixel and (ii) have a distance below a threshold d. As before, for each feature X_j of the i-th bag our classifier provides a score s_j, but now the decision is taken by maximizing over the evidence provided by the instances belonging to a bag. Our cost function thus becomes:

$$l^m(\mathbf{W},\mathbf{w}^{(m)}) = \sum_{j\in Y_-} w_{\hat{y}_j} S(-1, s_j^m) + \sum_{j\in Y_+} w_{\hat{y}_j} S(1, \max_{j\in\mathcal{B}_i} s_j^m) \qquad (2)$$

where $\mathcal{B}_i$ is the 'bag' of pixel indices associated with sample i; this allows positive samples to select the neighbours that most support them while forcing all negatives to be negative. Even though the max operation is not differentiable, we can compute a subgradient and use it for backpropagation.

Graduated DSN Training: The side-layer terms in the objective function

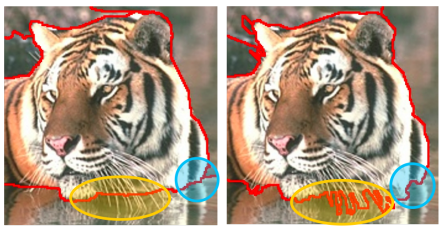

Figure 3: Location uncertainty of human annotations in the BSD dataset: even if annotators agree on the semantics, their boundaries may not coincide.

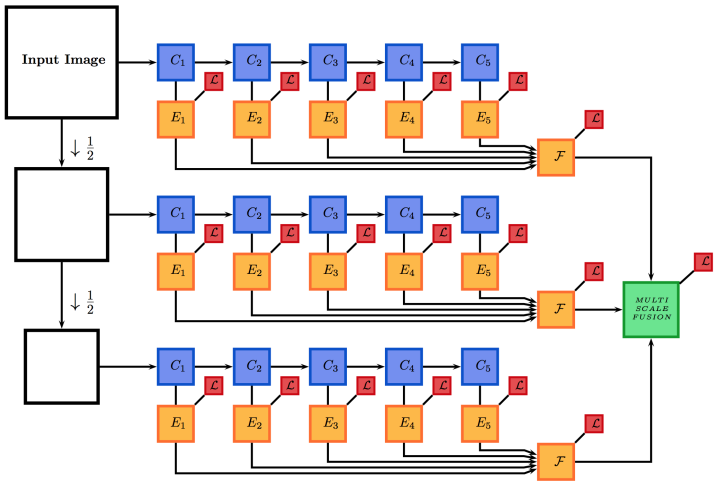

Figure 4: Network architecture used for multi-resolution HED training: as in HED, every intermediate layer of a DCNN (shown in blue) is processed by a side layer (shown in orange) which is penalized by a loss function $\mathcal{L}$. The intermediate results are combined in a late fusion stage, which is again trained with the same loss function. In our architecture three differently scaled versions of the input image are provided as inputs to three FCNN networks that share weights - their multi-resolution outputs are fused in a late fusion stage, extending DSN to multi-resolution training.

of HED, Eq. 1 can be understood as steering the optimization problem to a good solution, forcing not only the final output, but also the intermediate layers to be discriminative. But once the network parameters are in the right regime, we can discard any guidance that was required to get us there. This is a strategy used in the classical Graduated Non-Convexity algorithm [2], and here we show that it also helps improve DSN when applied to boundary detection. For this we modify the HED objective, associating the side term with a decreasing weight while fixing the fusion term's weight:

$$\mathcal{L}^{(t)}(\mathbf{W},\mathbf{w},\mathbf{h}) = (1-\frac{t}{T})\mathcal{L}_{side}(\mathbf{W},\mathbf{w}) + \mathcal{L}_{fuse}(\mathbf{W},\mathbf{w},\mathbf{h}),$$

where t is the current training epoch and T is the total number of epochs. Our 'Graduated-DSN' training criterion starts from DSN, where every intermediate layer is trained for classification, and eventually leads to a skip-layer architecture, where the early layers are handled exclusively by the final fusion criterion. By the end the fusion-layer can manipulate the side-layers at will. The improvements are reported in the G-DSN column of Table 1.

Multi-Resolution Architecture: As shown in Fig. 4, we fuse boundary information coming from multiple scales through a multi-resolution architecture with tied weights, meaning that layers that operate at different resolutions share weights with each other. Parameter sharing across layers both accelerates convergence and also avoids over-fitting. In order to capture fine-level boundaries the top-resolution image is an upsampled version of the original. The multi-resolution results are combined through an additional fusion layer that combines the fused results of the individual resolutions. The improvements are reported in the 'M-Scale' column of Table 1.

Training with external data: We use boundaries from the VOC Context dataset [8], where all objects and 'stuff' present in the scene are manually segmented. Our sole modification to those boundaries has been to label the interiors of houses as 'don't care' regions that are ignored by the loss, since all of the window, or door boundaries are missed by the annotators. We flip these images, resulting in roughly 20000 images, which are appended to the original dataset. The 'VOC' column of Table 1 indicates the improvement.

Using grouping in a deep architecture: We 'globalize' our bottom-up contour detection results through the Normalized Cut technique [1]; the directional derivatives of the resulting eigenvectors can be used for boundary detection, known as the 'spectral probability of boundary' cue [1]. One of the main impediments to the application of this technique has been computation time. For this we integrate the GPU-based Damascene system of [3]

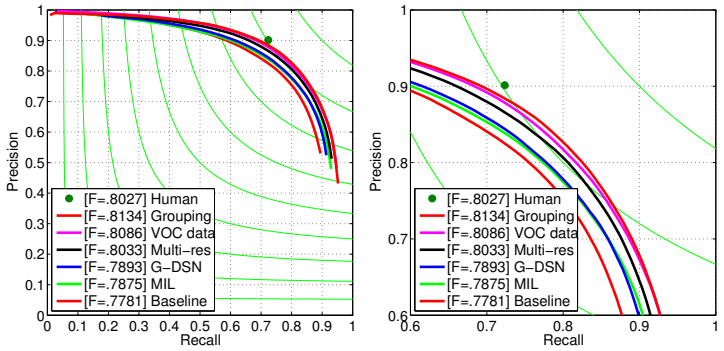

Figure 5: Impact of the different improvements described in Section 2: starting from a baseline that performs only slightly worse than the HED system of [9] we end up with a detector that largely surpasses human F-measure, illustrated in terms of green isocontours. On the right we zoom into the high-F measure regime.

Method	ODS	OIS	AP
gPb-owt-ucm [1]	0.726	0.757	0.696
SE-Var [4]	0.746	0.767	0.803
HED-fusion [9]	0.790	0.808	0.811
HED-late merging [9]	0.788	0.808	0.840
Ours (DCNN + sPb)	0.8134	0.8308	0.866

Figure 6: Comparison to the state-of-the-art in boundary detection, including the latest version of HED, trained with its most recent dataset augmentation [9]. We clearly outperform HED across all performance measures, while maintaining the computational speed above 1 frame per second.

with the Caffe deep learning framework; when integrated with our boundary detector Damascene yields 8 eigenvectors for a 577×865 image in less that 0.2 seconds. This further improves the performance of our detector, yielding an F-measure of 0.813, which is substantially better than our earlier performance of 0.809, and humans, who operate at 0.803.

Summary: We summarize the impact of the different steps described above in Fig. 5 - starting from a baseline that performs slightly worse than the HED system of Xie and Tu [9] we have introduced a series of changes that resulted in a system that performs boundary detection with an F-measure that exceeds that of humans. Our method outperforms the current state-of-the-art method of Xie and Tu [9] in terms of all typical performance measures, as shown in Table 6. A preliminary version of this boundary detector is available from our website; the full code will soon be released.

[1] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *PAMI*, 2011.

[2] Andrew Blake and Andrew Zisserman. *Visual Reconstruction*. MIT Press, 1987.

[3] Bryan C. Catanzaro, Bor-Yiing Su, Narayanan Sundaram, Yunsup Lee, Mark Murphy, and Kurt Keutzer. Efficient, high-quality image contour detection. In *Proc. ICCV*, 2009.

[4] Piotr Dollár and C. Lawrence Zitnick. Fast edge detection using structured forests. *PAMI*, 37(8):1558–1570, 2015.

[5] Iasonas Kokkinos. Boundary detection using f-measure-, filter- and feature- (f^3) boost. In *ECCV*, 2010.

[6] Iasonas Kokkinos. Pushing the boundaries of boundary detection using deep learning. *ICLR*, 2016.

[7] Chen-Yu Lee, Saining Xie, Patrick W. Gallagher, Zhengyou Zhang, and Zhuowen Tu. Deeply-supervised nets. In *Proc. AISTATS*, 2015.

[8] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam Cho, Seong Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. 2014.

[9] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *Proc. ICCV*. 2015.