Situational Object Boundary Detection

Jasper Uijlings, Vittorio Ferrari
CALVIN group - University of Edinburgh

Most methods for object boundary detection are monolithic and use a single predictor to predict all object boundaries in an image [1, 2] regardless of the image content. But intuitively, the appearance of object boundaries is dependent on what is depicted in the image. For example, black-white transitions are often good indicators of object boundaries, unless the image depicts a zebra as in Fig 1. Outdoors, the sun may cast shadows which create strong contrasts that are not object boundaries, while similar colour contrasts in an indoor environment with diffuse lighting may be caused by object boundaries. Furthermore, not all objects are equally important in all circumstances: one may want to detect the boundary between a snowy mountain and the sky in images of winter holidays, while ignoring sky-cloud transitions in images depicting air balloons, even though such boundaries may be visually very similar. These examples show that one cannot expect a monolithic predictor to accurately predict object boundaries in all situations.

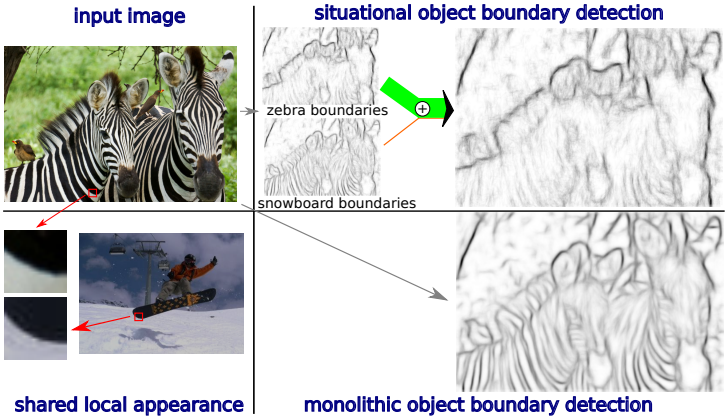

Figure 1: *Monolithic vs situational object boundary detection. Black-white transitions indicate an object boundary for the snowboard, but are false object boundaries for a zebra. This ambiguity cannot be resolved by a monolithic detector. In contrast, by training class specific object boundary detectors and classifying the image as a zebra, we correctly ignore most stripes.*

In this work we recognize the need for different object boundary detectors in different situations: first we define a set of situations and pre-train object boundary detectors for each of them using structured edge forests by Dollár and Zitnick [2]. For a test image, we classify which situations the image depicts based on its context. We model context by global image appearance through SIFT + Fisher vectors or CNN features. Having determined the situation, we apply the appropriate set of object boundary detectors. Hence conditioned on the situation of an image we choose which object boundary detectors to run. We call this Situational Object Boundary Detection. Our process is visualised in Figure 2.

More specifically, we perform situational object boundary prediction by

$$\mathbf{D}(I) = \frac{1}{Z}\sum_{j=1}^{n} P(\hat{S}_j|I)\cdot\hat{D}_j(I) \qquad (1)$$

which is the sum over the object boundary predictions $\hat{D}_j(I)$ of the most likely n situations, weighted by the probabilities $P(\hat{S}_j|I)$ that situation $\hat{S}_j$ occurs in image I. Here $Z = \sum_{j=1}^{n} P(\hat{S}_j|I)$ is a normalizing factor and n is small for computational efficiency.

Now one important question is how to define such situations. Since the appearance of object boundaries are for a large part dependent on the object class, one natural choice is to use each object class as a single situation. This results in *class specific* object boundary detectors, which can deal for example with the zebra in Figure 1. However, object boundaries are also determined by the object pose and the background or context of the image. Since this can vary within a single object class, we propose to cluster images of a single class into subclasses based on global image appearance. This leads to *subclass specific* object boundary detectors. Finally, one can imagine that the context of the image itself determines what kind of object boundaries to expect. For example, in the countryside one can expect cow/grass boundaries, while in the city one can expect street/car boundaries. Therefore we cluster images based on their global image appearance which results in *class agnostic* object boundary detectors. Hence we experiment with three types of situations: *class specific*, *subclass specific*, and *class agnostic*.

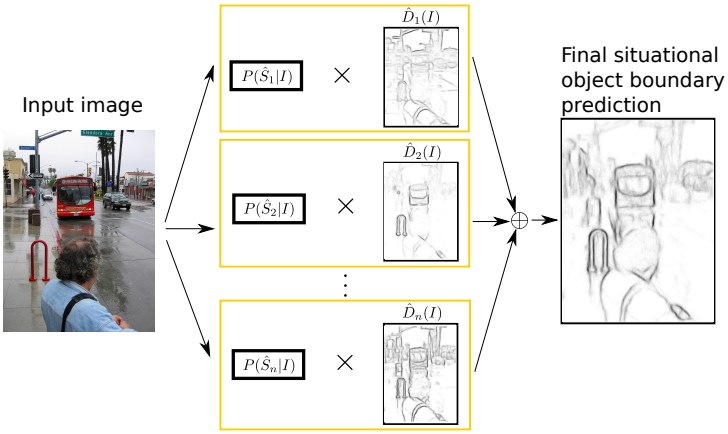

Figure 2: *Overview of situational object boundary detection. For each situation there is a specialised boundary detector $\hat{D}_j$ which we apply by $\hat{D}_j(I)$. The resulting specialised predictions vary greatly and are combined into a final prediction using Equation 1.*

We evaluate our situational object boundary detection method on three large datasets: Pascal VOC 2012 segmentation [3], Microsoft COCO [6], and part of ImageNet [7]. We note that Microsoft COCO is two orders of magnitude larger than the classical BSD500 [1]. For ImageNet we train from segments which are created in a semi-supervised fashion by Guillaumin et al. [4]. Our experiments show that our situational object boundary detection gives significant improvements over a monolithic approach.

Additionally, we apply our class-specific object boundary detection method to the task of *semantic contour detection* [5], i.e. *class-specific* object boundary detection. Using the their SDB dataset and evaluation software, we show substantial improvements over [5].

[1] P Arbeláez, M Maire, C Fowlkes, and J Malik. Contour Detection and Hierarchical Image Segmentation. *TPAMI*, 2011.

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

[3] M. Everingham, S. Eslami, L. van Gool, C. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge - a retrospective. *IJCV*, 2014.

[4] M. Guillaumin, D. Küttel, and V Ferrari. ImageNet auto-annotation with segmentation propagation. *IJCV*, 2014.

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

[6] 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.

[7] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A.C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *IJCV*, 2015.

This is an extended abstract. The full paper is available at the Computer Vision Foundation webpage.