DeepContour: A Deep Convolutional Feature Learned by Positive-sharing Loss for Contour Detection

Wei Shen[1], Xinggang Wang[2], Yan Wang[3], Xiang Bai[2], Zhijiang Zhang[1]
[1] Key Lab of Specialty Fiber Optics and Optical Access Networks, Shanghai University.
[2] School of Electronic Information and Communications, Huazhong University of Science and Technology.
[3] Rapid-Rich Object Search Lab, Nanyang Technological University.

Contour detection is a fundamental problem in computer vision, which serves as the basis of a variety of tasks such as image segmentation [1] and object recognition [6]. Traditionally, the framework for contour detection designs a variety of gradient features for each image pixel, followed by learning a binary classifier to determine whether an image pixel is passed through by contours or not [1, 2, 7].

In this paper, we investigate how to automatically learn contour features to replace the hand-designed gradient features for contour detection by the powerful deep convolutional neural networks (CNNs) [5]. Different from other deep learning based contour detection approaches [3, 4], which uses the deep networks as blackbox models, we customize the learning process by considering the intrinsic properties of contours. We first partition contour patches into compact clusters according to their inherent structures and assign each of them a shape label. Such a process converts the binary classification problem (i.e. predicting whether an image patch belongs to contour or non-contour) to a multi-class problem (i.e. predicting whether an image patch belongs to each shape class or the negative class). Fitting contour data of different shapes by different model parameters is followed by the spirit of divide-and-conquer strategy, which can ease the training difficulty due to the diversity of the contour data. Then, we define a new loss function, named positive-sharing loss, in which the loss for the positive class is shared among each shape class. It brings better regularization to the pre-defined clusters and move focus back the end goal of binary classification.

Following [6], we obtain the shape classes by clustering patches extracted from the binary images representing the hand labeled ground truth contours. This clustering process leads to K shape classes, with which we can assign a label y to each color image patch x extracted from the images in the dataset. If x is a contour patch, its class label is supplied by the shape clustering results, i.e. its class label $y = k(k \in \{1,\dots,K\})$ if its corresponding patch extracted from the hand labeled ground truth binary image belongs to k-th shape class. Otherwise, if x is a background patch, its label is assigned by $y = 0$.

Fig. 1 depicts the overall architecture of our CNN, which contains six layers with parameters to be learned; the first four are convolutional and the remaining two are fully-connected. Only convolutional and fully-connected layers contain learnable parameters, while the others are parameter free.

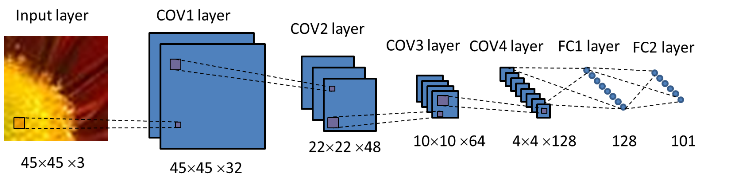

Figure 1: An illustration of the architecture of our CNN, explicitly visualizing the dimensions of each network layers. Due to the limited space, we only show the layers with learnable parameters. The convolutional layers are represented by blue squares, while fully-connected ones are marked by blue dots. The big and small light red blocks depict the convolutional kernels and results, respectively.

Given a training set which contains m image patches: $\left\{x^{(i)}, y^{(i)}\right\}_{i=1}^{m}$, where $x^{(i)}$ is the i-th image patch and $y^{(i)} \in \{0,1,\dots,K\}$ is its class label. If $y^{(i)} = 0$, then $x^{(i)}$ is a negative patch; If $y^{(i)} = k > 0$, then $x^{(i)}$ is a positive patch and belongs to the k-th shape class. Let $(a_j^{(i)}; j = 0,1,\dots K)$ be the output of unit j in the second fully-connected layer (FC2) for $x^{(i)}$, then the probability that the label of $x^{(i)}$ is j is given by

$$p_j^{(i)} = \frac{\exp(a_j^{(i)})}{\sum_{l=0}^{K} \exp(a_l^{(i)})}. \tag{1}$$

Our loss function is defined by

$$J = -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{j=0}^{K} \mathbf{1}(y^{(i)} = j)\log p_j^{(i)} + \sum_{i=1}^{m} \lambda\left(\left(\mathbf{1}(y^{(i)} = 0)\log p_0^{(i)} + \sum_{j=1}^{K}\mathbf{1}(y^{(i)} = j)\log(1 - p_0^{(i)})\right)\right)\right], \tag{2}$$

where $\mathbf{1}(\cdot)$ is the indicator function and λ is a controlling parameter. The first term in Eq. 1 is the a $(K+1)$-way softmax loss, and the second one is the loss to emphasize the incorrect estimation between the zero label and nonzero labels. The partial derivatives of the new loss are obtained by

$$\frac{\partial J}{\partial a_0^{(i)}} = \frac{1}{m}\left[(\lambda+1)\mathbf{1}(y^{(i)} = 0)(p_0^{(i)} - 1) + (\lambda+1)\sum_{j=1}^{K}\mathbf{1}(y^{(i)} = j)p_0^{(i)}\right], \tag{3}$$

and

$$\frac{\partial J}{\partial a_l^{(i)}} = \frac{1}{m}\left[(\lambda \mathbf{1}(y^{(i)} = 0) + 1)p_l^{(i)} - \mathbf{1}(y^{(i)} = l) - \lambda\sum_{j=1}^{K}\mathbf{1}(y^{(i)} = j)\left(\frac{p_0^{(i)} p_l^{(i)}}{1 - p_0^{(i)}}\right)\right]. \tag{4}$$

Our conclusion can be summarized into two potins: one is partition contour (positive) data subclasses is necessary for training an effective CNN model, the other is the proposed positive-sharing loss function, which emphasizes the losses for the contour and non-contour rather than the loss for each subclass, facilitating to explore more discriminative features than softmax loss function.

[1] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5):898–916, 2011.

[2] Piotr Dollár and C. Lawrence Zitnick. Structured forests for fast edge detection. In *Proc. ICCV*, pages 1841–1848, 2013.

[3] Yaroslav Ganin and Victor S. Lempitsky. N^4-fields: Neural network nearest neighbor fields for image transforms. In *Proc. ACCV*, 2014.

[4] Jyri J. Kivinen, Christopher K. I. Williams, and Nicolas Heess. Visual boundary prediction: A deep neural prediction network and quality dissection. In *Proc. AISTATS*, pages 512–521, 2014.

[5] Y. LeCun, B. Boser, J.S. Denker, D.Henderson, R.E. Howard, W. Hubbard, and L.D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4), 1989.

[6] Joseph J. Lim, C. Lawrence Zitnick, and Piotr Dollár. Sketch tokens: A learned mid-level representation for contour and object detection. In *Proc. CVPR*, pages 3158–3165, 2013.

[7] David R. Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. ICCV*, pages 416–425, 2001.

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