Richer Convolutional Features for Edge Detection

Hongzhi Liu

Jun 4, 2018

Abstract

It is known to all that edge detection, which aims to extract visually salient edges and object boundaries from natural images, has remained as one of the main challenges in computer vision for several decades. Today, I read a thesis written by Yun Liu, who is from Nankai University. His team propose an accurate edge detector using richer convolutional features. Besides, the proposed network fully exploits multiscale and multilevel information of objects to perform the image-to-image prediction by combining all the meaningful convolutional features in a holistic manner.

1. Overview of RCF

Since objects in natural images possess various scales and aspect ratios, learning the rich hierarchical representations is very critical for edge detection. Typically, traditional methods first extract local cues of brightness, colors or other manually designed features and then sophisticated learning paradigms [2] are used to classify edge and non-edge pixels. Although edge detection approaches using low-level features have made great improvement in these years [3], their limitations are also obvious. Edges and boundaries are often defined to be semantically meaningful, however, it is difficult to use low-level cues to represent object-level information.

The aforementioned CNN-based models have advanced the state-of-the-art significantly but all of them lost some useful hierarchical CNN features when classifying pixels to edge or non-edge class. These methods usually only adopt CNN features from the last layer of each conv stage. In the paper, Yun Liu and his team present an accurate edge detector using richer convolutional features (RCF) [4] to fix this case.

2. Richer Convolutional Features

Inspired by previous literature in deep learning, the team design their network by modifying VGG16 network [6].

VGG16 network that composes of 13 *conv* layers and 3 fully connected layers has achieved state-of-the-art in a va-

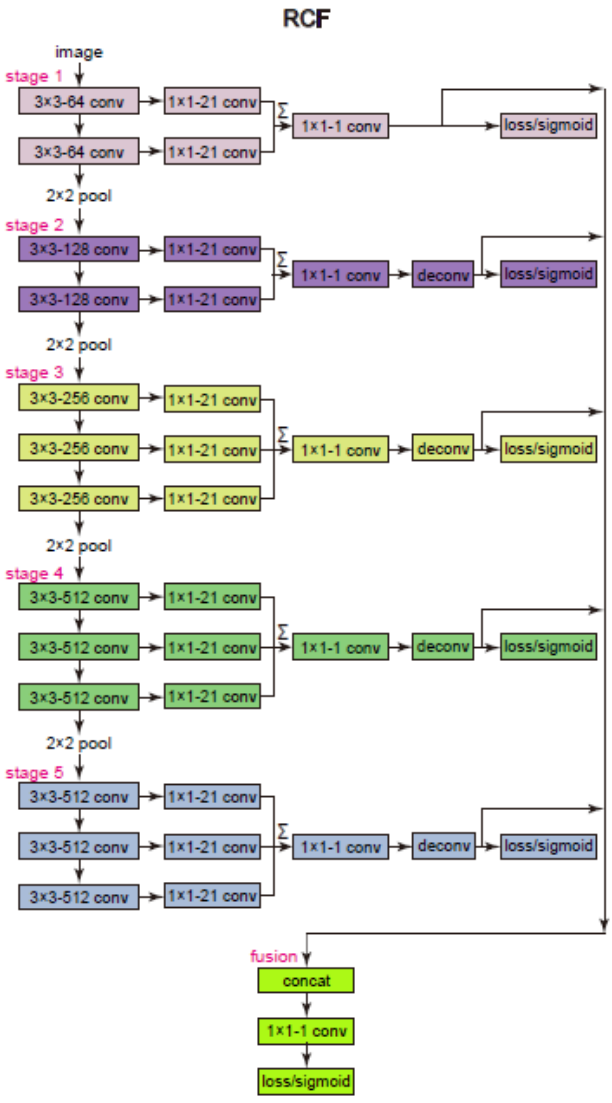

Figure 1. The RCF network architecture. The input is an image with arbitrary sizes and their network outputs an edge possibility map in the same size.

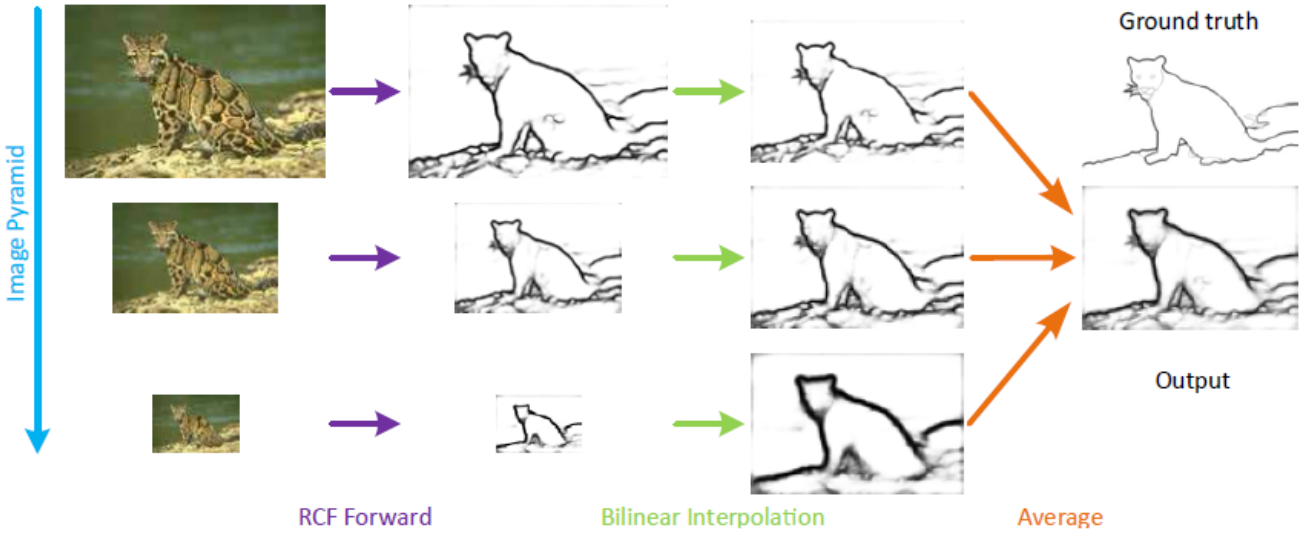

Figure 2. The pipeline of their multiscale algorithm. The original image is resized to construct an image pyramid. And these multiscale images are input to RCF network for a forward pass. Then, the team use bilinear interpolation to restore resulting edge response maps to original sizes. A simple average of these edge maps will output high-quality edges.

riety of tasks such as image classification, object detection and *etc*. Its *conv* layers are divided into five stages, in which a pooling layer is connected after each stage. The useful information captured by each *conv* layer becomes coarser with its receptive field size increasing. The use of this rich hierarchical information is hypothesized to help a lot. The starting point of their network design lies here. The novel network proposed by Yun Liu is shown in Fig. 1.

Compared with VGG16, theirs modifications can be described as following. First of all, the team cut all the fully connected layers and the *pool5* layer. On the one side, they remove the fully connected layers due to the fact that they do not align with our design of fully convolutional network. On the other hand, adding *pool5* layer will increase the stride by two times, and it's harmful for edge localization. Besides, all the up-sampling layers are concatenated. Then an 1×1 *conv* layer is used to fuse feature maps from each stage. At last, a cross-entropy loss / sigmoid layer is followed to get the fusion loss / output.

2.1. Annotatorrobust Loss Function

Edge datasets in this community are usually labeled by several annotators using their knowledge about the presences of objects and object parts. Though humans vary in cognition, these human-labeled edges for the same image share high consistency. For each image, they average all the ground truth to generate an edge probability map, which ranges from 0 to 1. Here, 0 means no annotator labeled at this pixel, and 1 means all annotators have labeled at this pixel. They consider the pixels with edge probability higher than η as positive samples and the pixels with edge probability equal to 0 as negative samples. Otherwise, if a pixel is marked by fewer than η of the annotators, this pixel may be semantically controversial to be an edge point. Thus, whether regarding it as positive or negative samples may confuse networks. So Liu ignores pixels in this category.

They compute the loss at every pixel with respect to pixel label as Equation 1:

$$l(X_i; W) = \begin{cases} \alpha \cdot \log(1 - P(X_i; W)), & \text{if } y_i = 0, \\ 0, & \text{if } 0 < y_i \leq \eta, \\ \beta \cdot \log P(X_i; W), & \text{otherwise}, \end{cases} \tag{1}$$

in which

$$\alpha = \lambda \frac{|Y^+|}{|Y^+| + |Y^-|} \tag{2}$$

$$\beta = \frac{|Y^-|}{|Y^+| + |Y^-|}. \tag{3}$$

When Y^+ and Y^- denote positive sample set and negative sample set respectively. The hyper-parameter λ is to balance positive and negative samples. The activation value and ground truth edge probability at pixel i are presented by X_i and y_i, respectively. $P(X)$ is the standard sigmoid function, and W denotes all the parameters that will be learned in their architecture. Therefore, the improved loss function can be formulated as Equation 4:

$$L(W) = \sum_{i=1}^{|I|} (\sum_{k=1}^{K} l(X_i^{(k)}; W) + l(X_i^{fuse}; W)) \tag{4}$$

Table 1. Comparison with state-of-the-art on Cityscapes dataset.

Method	ODS	OIS
Human-Boundary [5]	.760 (.017)	-
Multicue-Boundary [5]	.720 (.014)	-
HED-Boundary	.814 (.011)	.822 (.008)
RCF-Boundary	.817 (.004)	.825 (.005)
RCF-MS-Boundary	.825 (.008)	.836 (.007)
Human-Edge [5]	.750 (.024)	-
Multicue-Edge [5]	.830 (.002)	-
HED-Edge	.851 (.014)	.864 (.011)
RCF-Edge	.857 (.004)	.862 (.004)
RCF-MS-Edge	.860 (.005)	.864 (.004)

where X_i^k is the activation value from stage k while X_i^{fuse} is from fusion layer. $|I|$ is the number of pixels in image I, and K is the number of stages.

2.2. Multiscale Hierarchical Edge Detection

In single scale edge detection, the team input an original image into their fine-tuned RCF network then the output is an edge probability map. To further improve the quality of edges, they use image pyramids during testing. Specifically, they resize an image to construct an image pyramid and each of these images is input to their single-scale detector separately. Then all resulting edge probability maps are resized to original image size using bilinear interpolation. At last, these maps are averaged to get a final prediction map. Fig. 2 shows a visualized pipeline of their multiscale algorithm. The team also try to use weighted sum, but they find the simple average works very well. Considering the trade-off between accuracy and speed, they use three scales 0.5, 1.0, and 1.5 in this paper. When evaluating on BSDS500 [1] dataset, this simple multiscale strategy improves the ODS F-measure from 0.806 to 0.811, though the speed drops from 30 FPS to 8 FPS.

3. Experimental Results of the Method

Liu shows evaluation results in Tab. 1. He proposed RCF achieve substantially higher results than HED. For boundary task, RCF-MS is 1.1% ODS F-measure higher and 1.4% OIS F-measure higher than HED. For edge task, RCF-MS is 0.9% ODS F-measure higher than HED. Note that the fluctuation of RCF is also smaller than HED, which suggests RCF is more robust over different kinds of images.

References

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

[2] P. Dollár and C. L. Zitnick. Fast edge detection using structured forests. *IEEE TPAMI*, 2015. 1

[3] M. Leordeanu, R. Sukthankar, and C. Sminchisescu. Generalized boundaries from multiple image interpretations. *IEEE TPAMI*, 2014. 1

[4] Y. Liu, M.-M. Cheng, X. Hu, K. Wang, and X. Bai. Richer convolutional features for edge detection. In *CVPR*, 2017. 1

[5] D. A. Mély, J. Kim, M. McGill, Y. Guo, and T. Serre. A systematic comparison between visual cues for boundary detection. *Vision research*, 2016. 3

[6] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1