EAGER: Edge-Aided imaGe undERstanding System

Jianzhong He, Xiaobin Liu, Shiliang Zhang
School of Electronic Engineering and Computer Science, Peking University, Beijing 100871, China
{jianzhonghe,xbliu.vmc,slzhang.jdl}@pku.edu.cn

ABSTRACT

Image understanding is a fundamental task for many multimedia and computer vision applications, such as self-driving, multimedia retrieval, and augmented reality, *etc.* In this paper, we demonstrate that edge detection could aid image understanding tasks such as semantic segmentation, optical flow estimation, and object proposal generation. Based on our recent research efforts on edge detection, we develop a robust and efficient Edge-Aided imaGe undERstanding system named as EAGER. EAGER is built on a compact and efficient edge detection module, which is constructed with a bi-directional cascade network, multi-scale feature enhancement, and layer-specific training supervision, respectively. Based on detected edges, EAGER achieves accurate semantic segment, optical flow estimation, as well as object bounding-box proposal generation for user-uploaded images and videos.

KEYWORDS

Edge Detection, Semantic Segmentation, Optical Flow, Object Proposal, Convolutional Neural Network

1 INTRODUCTION

Edge detection [15] is an important and fundamental task for image understanding. It targets to detect object boundaries and perceptual salient edges from images, which preserve the gist of an image and ignore unintended details. Edge detection often serves as an initial step of many mid-and high-level multimedia and vision tasks, such as image segmentation [1], object detection [7], optical flow estimation [17], and *etc.*

Edge detection faces many challenges. Firstly, edges of same object may vary considerably in scale. For example, the scale of edges on human face may vary considerably as the face goes from far to close to the camera. Secondly, edges in one image may contain both object-level boundaries and meaningful local details, hence requiring the edge detector to present strong discriminative power for both high-level semantics and low-level visual details. Recent years, we have witnessed impressive progresses on deep learning based edge detection. For example, Bertasius *et al.* [3] employ CNN to generate features of candidate contour points. Xie *et al.* [22] propose an end-to-end detection model that leverages the outputs from different intermediate layers with skip-connections. Liu *et al.* [12] further learn richer deep representations by concatenating features derived from all convolutional layers. Xu *et al.* [23] introduce a hierarchical deep model to extract multi-scale features and a gated conditional random field to fuse them.

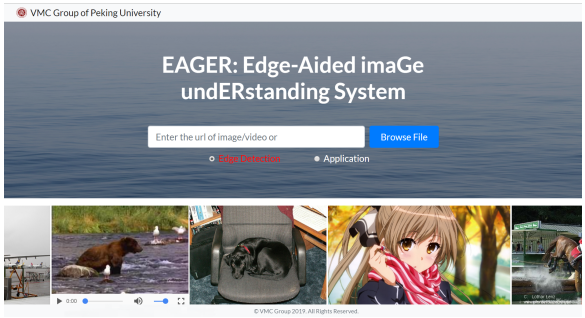

Figure 1: The main interface of EAGER.

Although recent works have significantly boosted the edge detection accuracy, they generally introduce deeper networks and increased computational complexities. Recently, we conducted several works towards efficient edge detection with compact neural network. Specifically, we propose a bi-directional network to train each intermediate network layer with layer specific supervision, leading to a more efficient edge detector training strategy. We also use the multi-scale feature enhancement to encourage the learning of multi-scale representations in a compact neural network. Those efforts have significantly boosted the accuracy and efficiency of edge detection.

Based on our edge detection works, we build the Edge-Aided imaGe undERstanding system (EAGER) and use detected edges to facilitate vision tasks including semantic segmentation, optical flow estimation, and object proposal generation. As shown in Fig. 1, EAGER is a web-based demo system composed of two modules, *i.e.*, edge detection module, and application module, respectively. It allows users to upload images or videos as input. The edge detection module detects edges from an input image or video and compares our results with the ones from several recent algorithms. Based on the edge detection module, the application module conducts and demonstrates the results of 1) semantic segmentation for images and videos, 2) bounding box proposal generation for images, and 3) optical flow estimation for videos.

ICMR '19, June 10–13, 2019, Ottawa, Canada

ACM ISBN 978-1-4503-5046-4/18/06…$15.00
https://doi.org/10.1145/3206025.3206086

Table 1: Comparison with other edge detection methods on *BSDS500* test set.

Method	ODS	OIS	AP
Human	.803	.803	–
DCD [10]	.799	.817	.849
ResNet50-AMHNet [23]	.798	.829	.869
RCF [12]	.811	.830	–
Deep Boundary [9]	.813	.831	.866
CED [20]	.815	.833	.889
ours	**.828**	**.844**	**.890**

Table 2: Comparison of semantic segmentation on the Pascal Context validation set.

methods	PA	MPA	mIOU
FCN-8s [13]	67.0	50.7	37.8
UoA-Context+CRF [11]	71.5	53.9	43.3
IFCN-8s [19]	74.5	57.7	45.0
DeepLab [6]	70.5	54.6	42.5
HED [22]-BNF	71.1	54.8	42.8
CED [20]-BNF	71.4	55.2	43.1
ours-BNF	**71.6**	**55.3**	**43.4**

2 ALGORITHMS

2.1 Edge Detection

We briefly introduce our algorithms for edge detection, which mainly consists of two steps, *i.e.*, edge prediction and none-maximum suppression. Edge prediction is conducted with our proposed bi-directional network architecture. By introducing a bi-directional structure to enforce each layer to focus on a specific scale, our network trains each intermediate layer with a layer-specific supervision. To enrich the multi-scale representations learned with a shallow network, we further introduce a multi-scale feature enhancement block implemented with dilated convolution. The final edge prediction is the fusion of all the outputs of intermediate network layers. After that, none-maximum suppression is conducted to generate the final edges.

Performance of edge detection: We use *BSDS500* [1] dataset to evaluate our edge detection module. *BSDS500* contains 200 images for training, 100 images for validation, and 200 images for testing. Each image is manually annotated by multiple annotators. The final groundtruth is the averaged annotations by the annotators. We also utilize the strategies in [22] to augment training and validation sets by randomly flipping, scaling and rotating images. We also adopt the PASCAL Context dataset [16] as our training set. We report the performance comparison in Table 1 and show the detection results in Fig. 2. As show in Table 1, our method achieves the best performance compared with others.

2.2 Applications

Semantic segmentation targets to label the semantic category of each pixel in an image. Since edge prediction keeps accurate localisation cues, we use edge detection results to facilitate semantic segmentation. This module consists of three parts, *i.e*, initial segmentation, edge detection, and refinement module, respectively. We apply deeplab [6] implemented with ResNet101 [8] as the initial segmentation model, and adopt the boundary neural field [4] to refine the initial segmentation. We use *Pascal Context* [16] dataset to evaluate our semantic segmentation module. *Pascal Context* includes 10103 images and is split into 4998 images for training, 5015 images for validation. We evaluate on the most frequent 60 classes and report the Pixel Accuracy (PA), Mean Pixel Accuracy (MPA), Mean Intersection Over Union (Mean IOU). We summarize the comparison in Table 2, where our method achieves the best results compared with others.

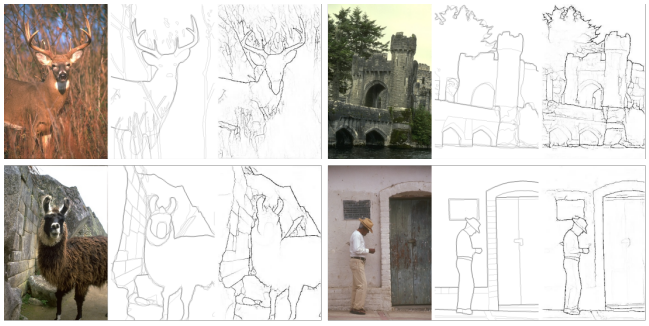

Figure 2: Examples of edge detection on *BSDS500* test set. Images from left to right in each example are the input images, ground truth, and edge predictions by EAGER.

Object proposal targets to generate object proposals which refer the object localization in image. This module mainly include two parts, *i.e.*, edge detection and proposal generation. Ultrametric Contour Map (UCM) is first computed according to the edge prediction. Object proposals are then generated using multi-scale combinatorial grouping strategy [2]. We use Pascal Context dataset [16] to evaluate our methods following the settings in previous works [14, 21]. The proposed method achieves competitive performance, *e.g.*, it achieves Average Recall of 0.73 with ∼2620 proposals per image, better than CED which achieves 0.73 with ∼2750 proposals per image.

Optical flow estimation aims to estimate the motion cues between adjacent video frames. We also use edge detection results to facilitate this task. This module consists of three parts, *i.e.*, deepmatching [18], which matches adjacent frames, edge detection for the affinity computation, and the final optimal flow refinement, respectively. We use *MPI Sintel* [5] dataset to evaluate our optical flow estimation methods. *MPI Sintel* has 23 and 12 naturalistic video sequences of animation for training and testing, respectively. Our edge detection algorithm achieves Average Endpoint Error 3.546 on *Sintel* training set, better than the 3.588 of HED [22].

3 THE EAGER SYSTEM

EAGER is a web-based demo system built with the edge detection and application models. The EAGER would run

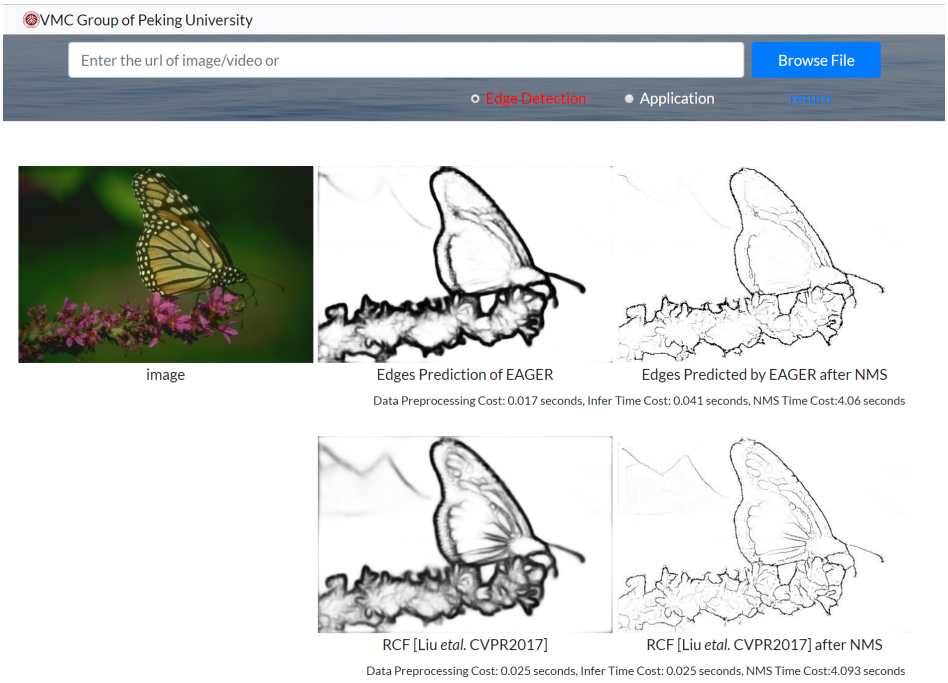

Figure 3: Interface of edge detection.

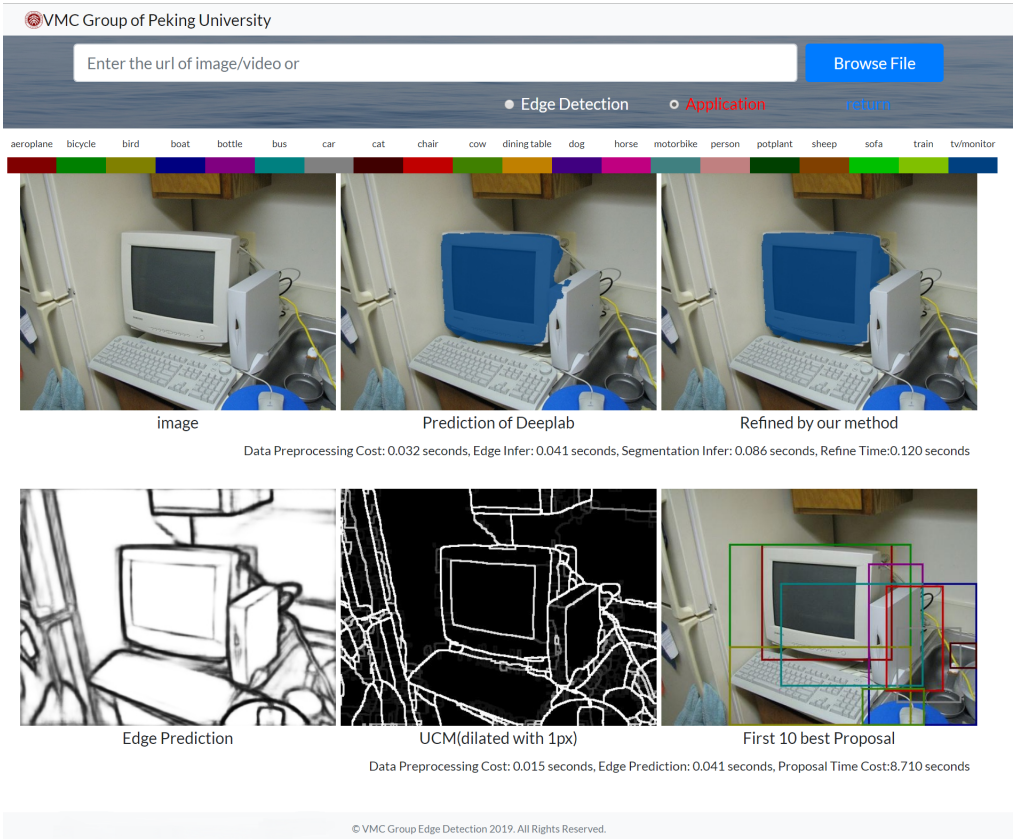

Figure 4: Interface of semantic segmentation and object proposal generation.

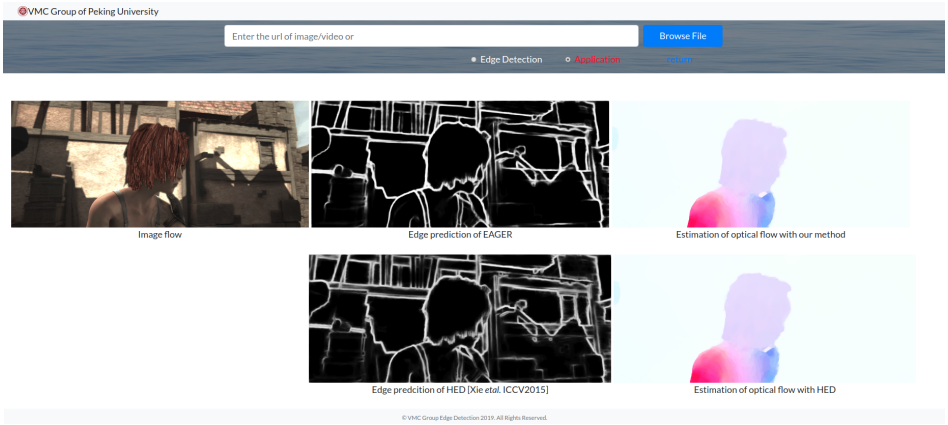

Figure 5: Interface of optical flow estimation.

on a server equipped with GTX-1080 GPU, intel i7 CPU, and 128GB memory. As shown in Fig. 1, users can select a video or image as input or upload their own images and videos. EAGER provides different interfaces to demonstrate the results of edge detection, semantic segmentation, optical flow estimation, and object proposal, respectively.

Fig. 3 shows the interface for edge detection, which demonstrates the input image, our edge prediction, our detected edges after NMS, as well as the results obtained by RCF [12]. The interface also shows the computational time.

Fig. 4 shows the interface for semantic segmentation and object proposal generation. The first row shows the input image, the semantic segmentation of Deeplab [6], as well as the final semantic segmentation refined by our edge detection result. The second row shows our edge prediction, the UCM for object proposal generation, as well as the finally generated object proposals.

Fig. 5 shows the interface for optical flow estimation. This module requires a video as input, and estimate the optical flow in the input video. The interface shows and plays the original video, the predicted edges on each video frame and the estimated optical flows between adjacent frames.

REFERENCES

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

[2] Pablo Arbeláez, Jordi Pont-Tuset, Jonathan T Barron, Ferran Marques, and Jitendra Malik. 2014. Multiscale combinatorial grouping. In *CVPR*.

[3] Gedas Bertasius, Jianbo Shi, and Lorenzo Torresani. 2015. DeepEdge: A multi-scale bifurcated deep network for top-down contour detection. In *CVPR*. 4380–4389.

[4] Gedas Bertasius, Jianbo Shi, and Lorenzo Torresani. 2016. Semantic segmentation with boundary neural fields. In *CVPR*. 3602–3610.

[5] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. 2012. A naturalistic open source movie for optical flow evaluation. In *ECCV*. Springer-Verlag.

[6] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. 2016. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv preprint arXiv:1606.00915* (2016).

[7] Vittorio Ferrari, Loic Fevrier, Frederic Jurie, and Cordelia Schmid. 2008. Groups of adjacent contour segments for object detection. *IEEE Trans. Pattern Anal. Mach. Intell.* 30(1) (2008), 36–51.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*. 770–778.

[9] Iasonas Kokkinos. 2016. Pushing the boundaries of boundary detection using deep learning. *ICLR* (2016).

[10] Yuan Liao, Songping Fu, Xiaoqing Lu, Chengcui Zhang, and Zhi Tang. 2017. Deep-learning-based object-level contour detection with CCG and CRF optimization. In *ICME*.

[11] Guosheng Lin, Chunhua Shen, Anton Van Den Hengel, and Ian Reid. 2016. Efficient piecewise training of deep structured models for semantic segmentation. In *CVPR*. 3194–3203.

[12] Yun Liu, Ming-Ming Cheng, Xiaowei Hu, Kai Wang, and Xiang Bai. 2017. Richer Convolutional Features for Edge Detection. In *CVPR*.

[13] Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully convolutional networks for semantic segmentation. In *CVPR*. 3431–3440.

[14] Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Pablo Arbeláez, and Luc Van Gool. 2018. Convolutional oriented boundaries: From image segmentation to high-level tasks. *IEEE Trans. Pattern Anal. Mach. Intell.* 40, 4 (2018), 819–833.

[15] David R Martin, Charless C Fowlkes, and Jitendra Malik. 2004. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Trans. Pattern Anal. Mach. Intell.* 26(5) (2004), 530–549.

[16] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. 2014. The Role of Context for Object Detection and Semantic Segmentation in the Wild. In *CVPR*.

[17] Jerome Revaud, Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. 2015. Epicflow: Edge-preserving interpolation of correspondences for optical flow. In *CVPR*. 1164–1172.

[18] Jerome Revaud, Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. 2016. Deepmatching: Hierarchical deformable dense matching. *International Journal of Computer Vision* 120, 3 (2016), 300–323.

[19] Bing Shuai, Ting Liu, and Gang Wang. 2016. Improving fully convolution network for semantic segmentation. *arXiv preprint arXiv:1611.08986* (2016).

[20] Yupei Wang, Xin Zhao, and Kaiqi Huang. 2017. Deep Crisp Boundaries. In *CVPR*.

[21] Yupei Wang, Xin Zhao, Yin Li, and Kaiqi Huang. 2019. Deep crisp boundaries: From boundaries to higher-level tasks. *IEEE Transactions on Image Processing* 28, 3 (2019), 1285–1298.

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

[23] Dan Xu, Wanli Ouyang, Xavier Alameda-Pineda, Elisa Ricci, Xiaogang Wang, and Nicu Sebe. 2017. Learning Deep Structured Multi-Scale Features using Attention-Gated CRFs for Contour Prediction. In *NIPS*. 3964–3973.