Supplementary Material for "RankED: Addressing Imbalance and Uncertainty in Edge Detection Using Ranking-based Losses"

Contents

S1. Further Insights on Certainty Map

As we discussed in Section 3.3 in the main manuscript, the conventional method of averaging multiple annotations and thresholding the averages pixel-wise has the problem of losing information about both rarely labeled edges and the number of times edges are labeled. In this section, we provide further insight by comparing the conventional annotation method with our certainty map calculation. From the figure, we see, for example, that ∼78% of edges are labeled in only one ground-truth annotation (the lowest certainty) in the Multi-cue (edge) dataset. On the other hand, this percentage drops to ∼9% in certainty map **c**. Therefore, using certainty map **c** provides a great advantage for our sorting task which gives more importance to higher certainty edges.

Furthermore, we provide a visual comparison between conventional averaging and our certainty map Figure S2. We see that our uncertainty map c ensures that pixels within the specified distance have the same certainty value, while the conventional approach does not.

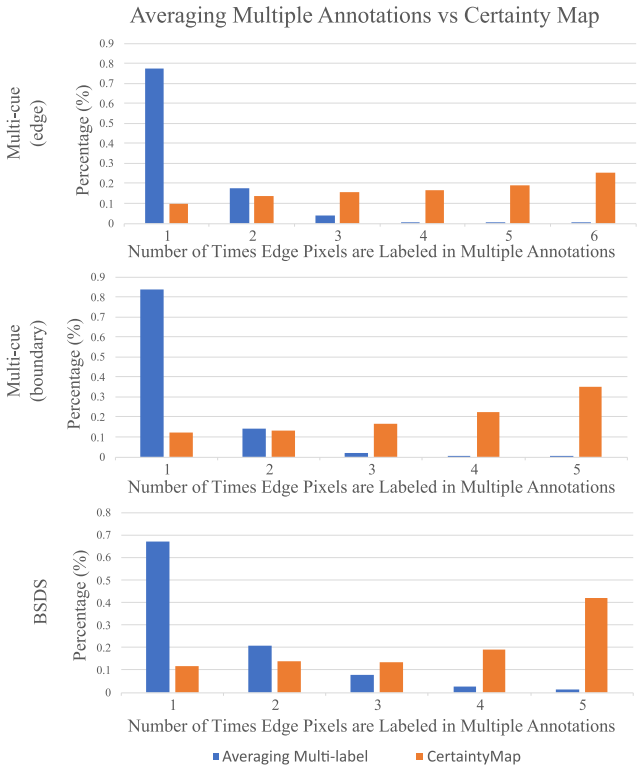
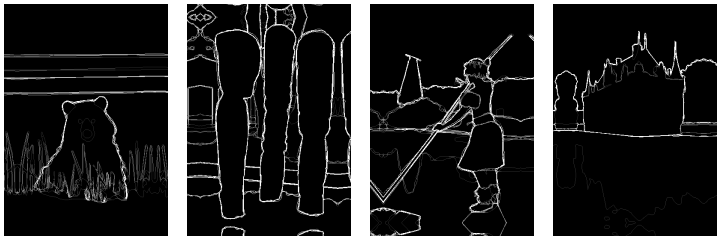


Figure S1. Comparison between Averaging Multiple Annotations (Standard Processing of Label) vs Certainty map **c** (Ours) on Multi-cue [20] and BSDS [1] datasets. While y-axes (percentage (%)) are the percentages of total edge pixels, x-axes (Number of Times Edge Pixels are Labeled in Multiple Annotations) show how many times edge pixels are labeled in multiple annotations.

S2. More Results on NYUD-v2

This section provides quantitative results for different input types (RGB, HHA, and RGB-HHA). NYUD-v2 [23] is published with both RGB and depth images. Also, using these depth images in HHA encoding [13] provides significant performance gain. By following the literature [6, 8, 15, 21, 24], we present results, shown in Table S1 for HHA and RGB-HHA (averaging of RGB and HHA results) inputs. According to Table S1, while we have the best results for AP measures in all input types, PiDiNet [24] has better performance than ours in ODS and OIS measures for HHA and RGB-HHA inputs in general. This can be explained by the usage of pre-trained weights. While PiDiNet is a lightweight edge detector that does not require any pre-trained weights, our model is initialized with weights of ImageNet [4]. Using weights trained on RGB images provides less performance gain for HHA inputs.

(a) Standard Processing (Averaging Multi-label)



(b) Ours (Certainty Map)

Figure S2. Visual comparisons between Standard label processing and our certainty map on BSDS [1] dataset.

Method	Pub.'Year	RGB			HHA			RGB-HHA		
		ODS	**OIS**	**AP**	**ODS**	**OIS**	**AP**	**ODS**	**OIS**	**AP**
gPb-ucm [1]	PAMI'11	.632	.661	.562	-	-	-	-	-	-
Silberman et al. [23]	ECCV'12	.658	.661	-	-	-	-	-	-	-
gPb+NG [12]	CVPR'13	.687	.716	.629	-	-	-	-	-	-
OEF [14]	CVPR'15	.651	.667	-	-	-	-	-	-	-
HED [27]	ICCV'15	.720	.734	.734	.682	.695	.702	.746	.761	.786
COB [19]	ECCV'16	-	-	-	-	-	-	.784	.805	.825
RCF [18]	CVPR'17	.729	.742	-	.705	.715	-	.757	.771	-
AMH-Net [28]	NeurIPS'17	.744	.758	.765	.717	.729	.734	.771	.786	.802
LPCB [6]	ECCV'18	.739	.754	-	.707	.719	-	.762	.778	-
BDCN [15]	CVPR'19	.748	.763	.770	.707	.719	.731	.765	.781	.813
PiDiNet [24]	ICCV'21	.733	.747	.715	**.728**	**.756**	-	.773	**.813**	
EDTER [21]	CVPR'22	.774	.789	.797	.703	.718	.727	.780	.797	.814
ACTD [8]	Neurocomp.'23	.762	.774	-	.723	.739	-	.783	.791	-
RANKED (R)	-	**.780**	**.793**	**.826**	.723	.735	**.760**	**.791**	.805	**.841**

Table S1. Quantitative comparisons on NYUD-v2 [23]. All results are computed with a single scale input. The best and second-best results are shown with bold and underlined texts, respectively. R: Ranking only.

S3. More Results on BSDS500

In this section, we provide Precision-Recall curves, quantitative results for different inputs (single-scale, multi-scale), and training set settings (with and without Pascal VOC).

Figure S6 shows the Precision-Recall curves for the SOTA methods and ours. Our method with multi-scale inputs has the best curve among the SOTA methods.

Moreover, we present detailed comparisons on the BSDS dataset shown in Table S2. We see that our method provides the best AP measures for all inputs and training settings except for SS+VOC (0.1 behind UAED [31]).

S4. More Results on Multi-cue

This section provides uncertainty-aware results (UaR) for the Multi-cue dataset shown in Table S3. Like UaR on the BSDS dataset in Table 6 of the main paper, our model gives a better performance for lower uncertainty (higher certainty) pixels.

S5. More Visual Results

This section presents more visual results in Figures S3, S4, and S5 on NYUD-v2, BSDS, and Multi-cue datasets, respectively. All results are obtained after the post-processing step. In general, using ranking and sorting tasks together (R+S) gives better visual results and OIS scores than using only ranking task (R) in BSDS and Multi-cue datasets.

Method		SS			MS			SS+VOC			MS+VOC		
		ODS	OIS	AP	ODS	OIS	AP	ODS	OIS	AP	ODS	OIS	AP
Canny [3]	(PAMI'86)	.611	.676	.520	-	-	-	-	-	-	-	-	-
gPb-UCM [1]	(PAMI'10)	.729	.755	.745	-	-	-	-	-	-	-	-	-
SCG [26]	(NeurIPS'12)	.739	.758	.773	-	-	-	-	-	-	-	-	-
SE [9]	(PAMI'14)	.743	.764	.800	-	-	-	-	-	-	-	-	-
OEF [14]	(CVPR'15)	.746	.770	.815	-	-	-	-	-	-	-	-	-
DeepEdge [2]	(CVPR'15)	.753	.772	.807	-	-	-	-	-	-	-	-	-
DeepContour [22]	(CVPR'15)	.757	.776	.790	-	-	-	-	-	-	-	-	-
HED [27]	(ICCV'15)	.788	.808	.840	-	-	-	-	-	-	-	-	-
DeepBoundary [16]	(ICLR'15)	.789	.811	.789	.803	.820	.848	.809	.827	.861	.813	.831	.866
CEDN [30]	(CVPR'16)	.788	.804	-	-	-	-	-	-	-	-	-	-
RDS [17]	(CVPR'16)	.792	.810	.818	-	-	-	-	-	-	-	-	-
COB [19]	(ECCV'16)	.793	.820	.859	-	-	-	-	-	-	-	-	-
AMH-Net [28]	(NeurIPS'17)	.798	.829	.869	-	-	-	-	-	-	-	-	-
RCF [18]	(CVPR'17)	.798	.815	-	-	-	-	.806	.823	-	.811	.830	.846
CED [25]	(CVPR'17)	.803	.820	.871	-	-	-	.815	.833	.889	-	-	-
LPCB [6]	(ECCV'18)	.800	.816	-	-	-	-	.808	.824	-	.815	.834	-
BDCN [15]	(CVPR'19)	.806	.826	.847	-	-	-	.820	.838	.888	.828	.844	.890
DSCD [5]	(ACMMM'20)	.802	.817	-	-	-	-	.813	.836	-	.822	.859	-
LDC [7] (	ACMMM'21)	.799	.816	.837	-	-	-	.812	.826	.857	.819	.834	.860
PiDiNet [24]	(ICCV'21)	-	-	-	-	-	-	.807	.823	-	-	-	-
FCL-Net [29]	(NN'22)	.807	.822	-	.816	.833	-	.815	.834	-	.826	.845	-
EDTER [21]	(CVPR'22)	<u>.824</u>	<u>.841</u>	.880	**.840**	**.858**	.896	.832	.847	.886	**.848**	**.865**	.903
UAED [31]	(CVPR'23)	**.829**	**.847**	<u>.892</u>	<u>.837</u>	<u>.855</u>	.897	**.838**	**.855**	**.902**	<u>.844</u>	<u>.864</u>	<u>.905</u>
CHRNet [10]	(Pat. Rec.'23)	-	-	-	-	-	-	.787	.788	.801	.830	.853	.870
ACTD [8]	(Neurocomp.'23)	.817	.836	.839	-	-	-	.821	.837	.850	.826	.842	.854
RANKED (Ranking Only)		.822	.838	.886	.829	.850	<u>.900</u>	<u>.833</u>	<u>.848</u>	<u>.901</u>	<u>.844</u>	.860	**.916**
RANKED (Ranking & Sorting)		<u>.824</u>	.840	**.895**	<u>.837</u>	<u>.855</u>	**.911**	NA	NA	NA	NA	NA	NA

Table S2. Quantitative results on BSDS dataset [1]. SS and MS represent single-scale and multi-scale, respectively. +VOC means that Pascal Context Data [11] is used as additional training data. The best and second-best results are shown with bold and underlined texts, respectively. Moreover, NA means not applicable due to a lack of uncertainty in the labels.

High uncertainty ←——————————————————————————→ Low uncertainty

		$\mathbf{c} \geq 0.2$			$\mathbf{c} \geq 0.4$			$\mathbf{c} \geq 0.6$			$\mathbf{c} \geq 0.8$			$\mathbf{c} = 1.0$		
Boundary	Method	ODS	OIS	AP	ODS	OIS	AP	ODS	OIS	AP	ODS	OIS	AP	ODS	OIS	AP
	RANKED (R)	0.954	0.959	0.992	0.959	0.964	0.994	0.965	0.970	0.995	0.971	0.974	0.996	0.976	0.979	0.998
	RANKED ($R+S$)	**0.963**	**0.967**	**0.995**	**0.969**	**0.973**	**0.996**	**0.974**	**0.978**	**0.997**	**0.979**	**0.982**	**0.998**	**0.983**	**0.986**	**0.999**
		$\mathbf{c} \geq 0.17$			$\mathbf{c} \geq 0.33$			$\mathbf{c} \geq 0.5$			$\mathbf{c} \geq 0.67$			$\mathbf{c} \geq 0.83$		
Edge	Method	ODS	OIS	AP	ODS	OIS	AP	ODS	OIS	AP	ODS	OIS	AP	ODS	OIS	AP
	RANKED (R)	0.951	0.953	0.962	0.957	0.959	0.968	0.961	0.964	0.974	0.966	0.968	0.978	0.979	0.98	0.989
	RANKED ($R+S$)	**0.962**	**0.965**	**0.973**	**0.969**	**0.972**	**0.980**	**0.974**	**0.977**	**0.985**	**0.979**	**0.981**	**0.989**	**0.989**	**0.989**	**0.996**

Table S3. Uncertainty-aware results for Multi-cue dataset. Also, c represents the certainty map mentioned in Algorithm 1. For boundary part, While case $c \geq 0.2$ contains all ground-truth edges in all labels, case c = 1.0 contains only ground-truth edges that are labeled in all labels. Similarly, $c \geq 0.17$ contains all ground-truth edges in all labels for the edge part. Thresholds are determined based on the number of labels for each RGB image. R: Ranking only. $R + S$: Ranking & Sorting.

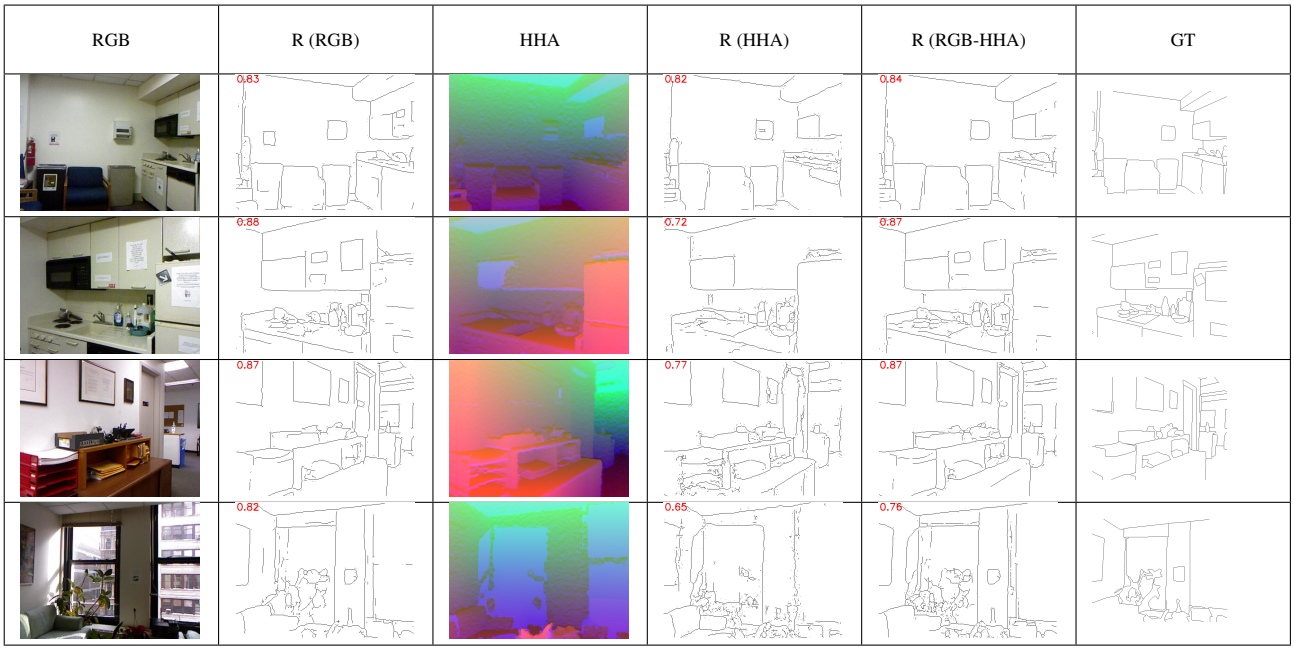


Figure S3. Visual results on NYUD-v2 dataset. All outputs are obtained after the post-processing step. R: Ranking, Red: OIS scores.

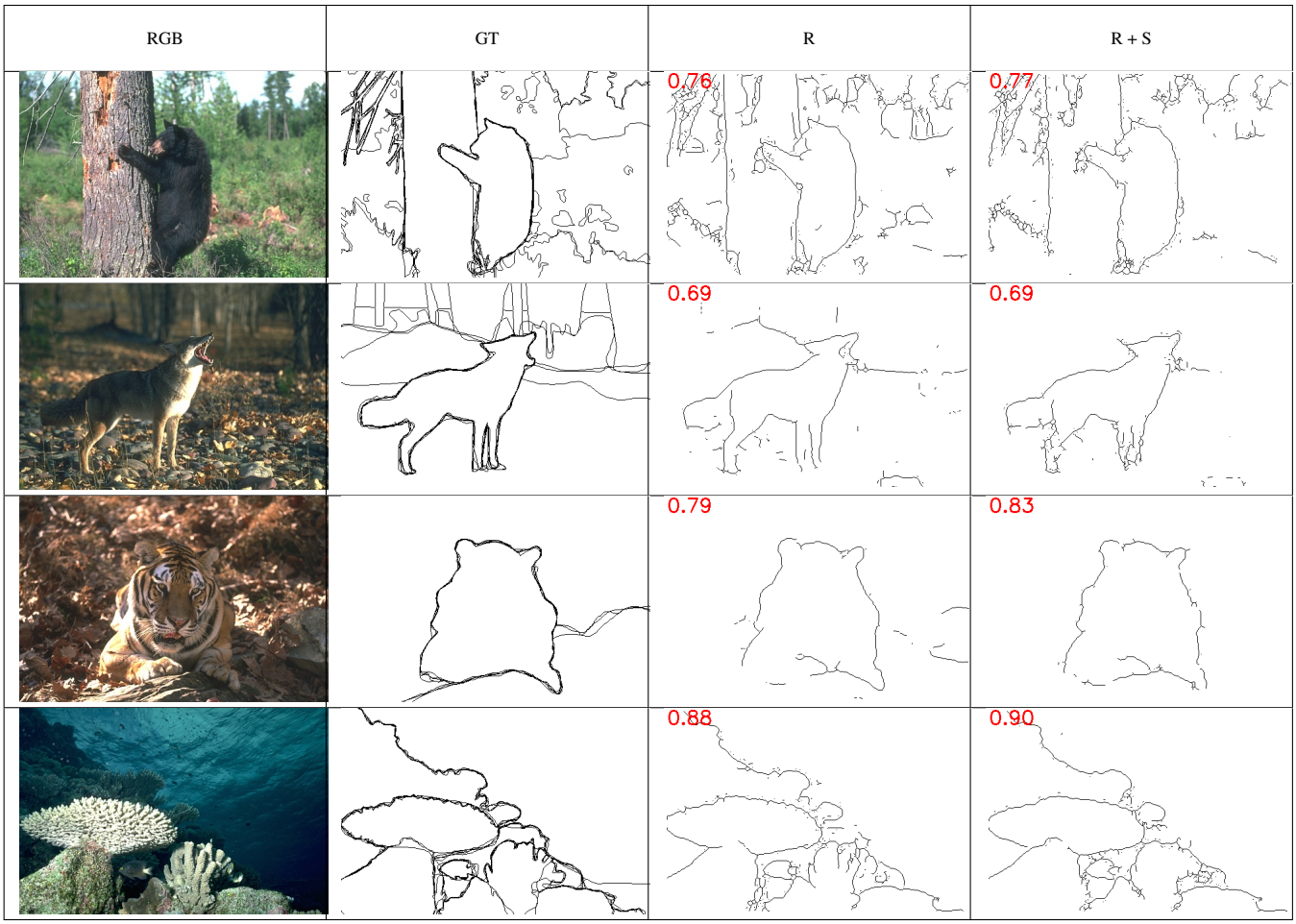
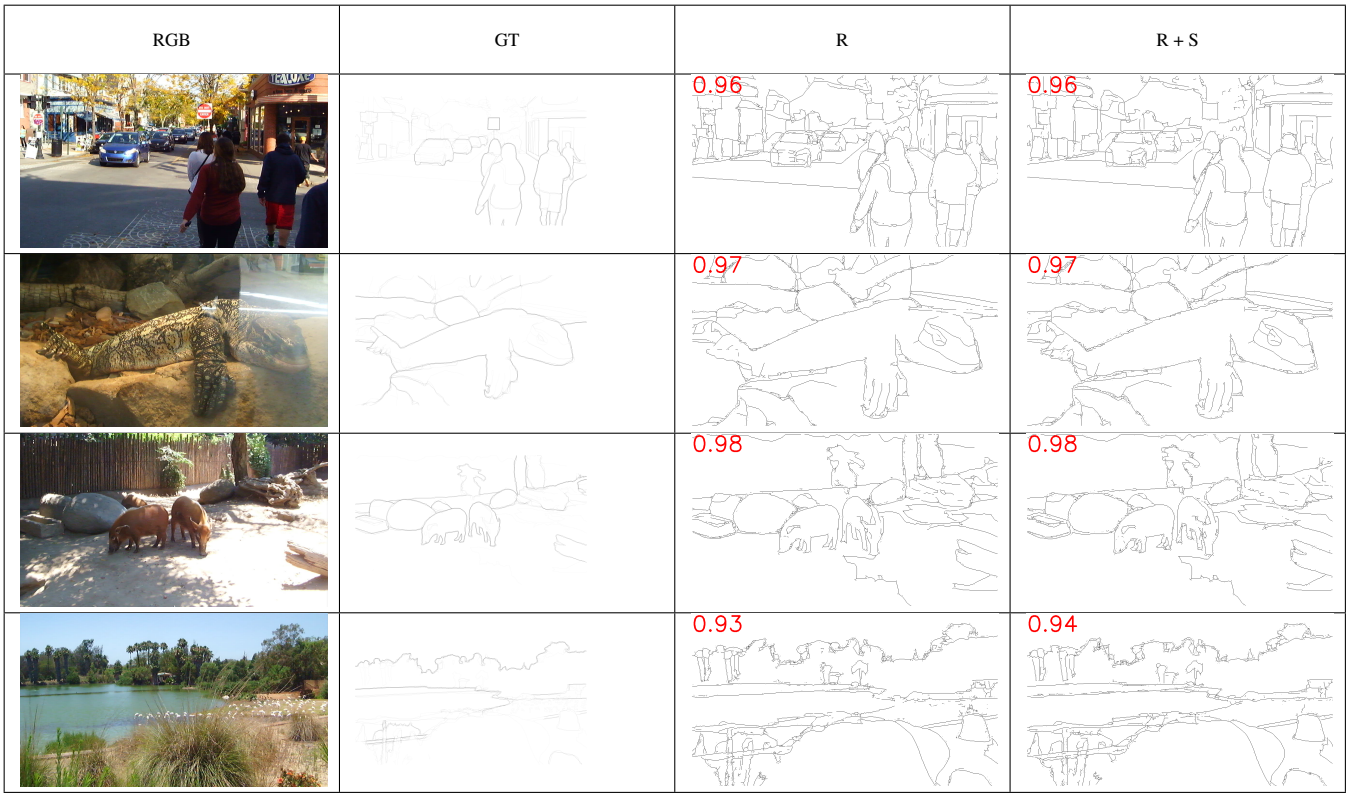
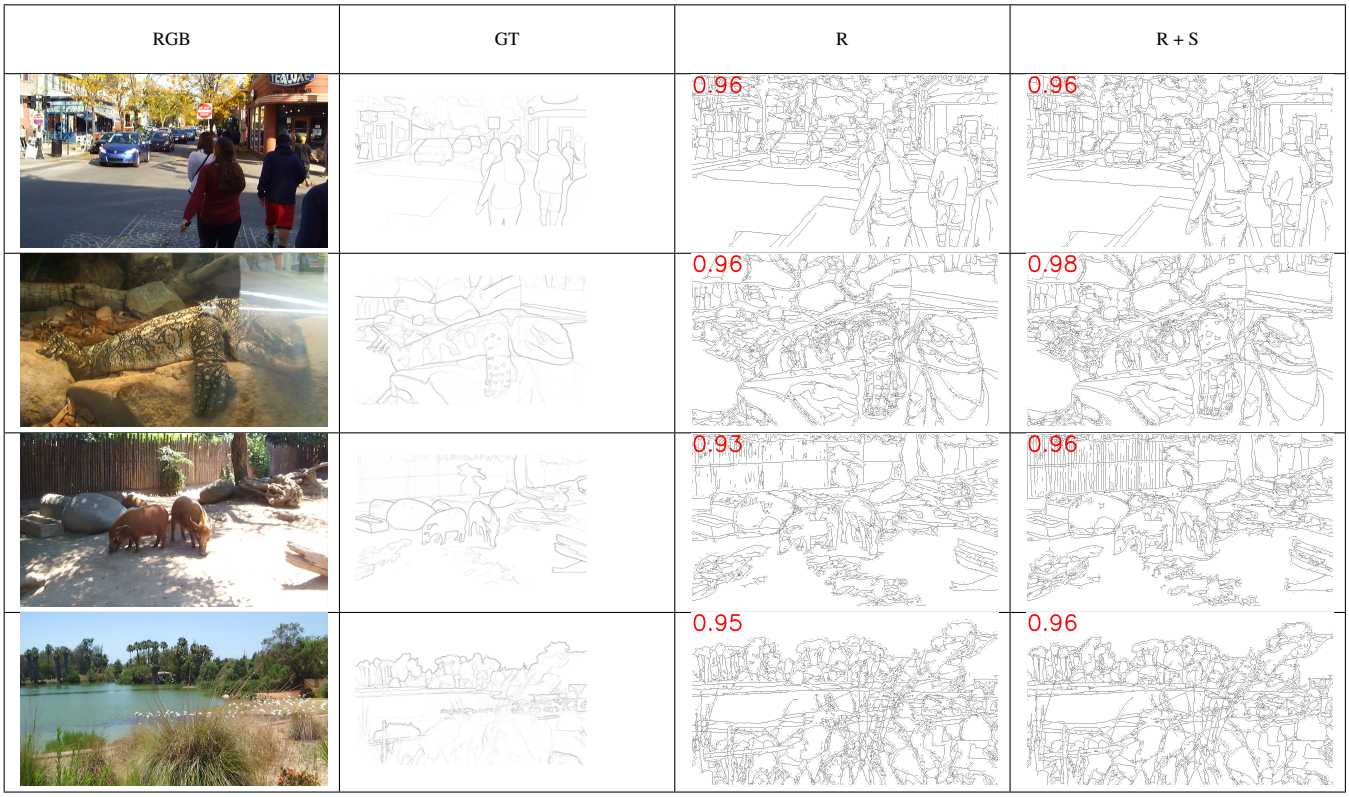


Figure S4. Visual results on BSDS dataset. All outputs are obtained after the post-processing step. R: Ranking, R + S: Ranking & Sorting, Red: OIS scores.



(a) Boundary



(b) Edge

Figure S5. Visual results on Multi-cue dataset. All outputs are obtained after the post-processing step. R: Ranking, R + S: Ranking & Sorting, Red: OIS scores.

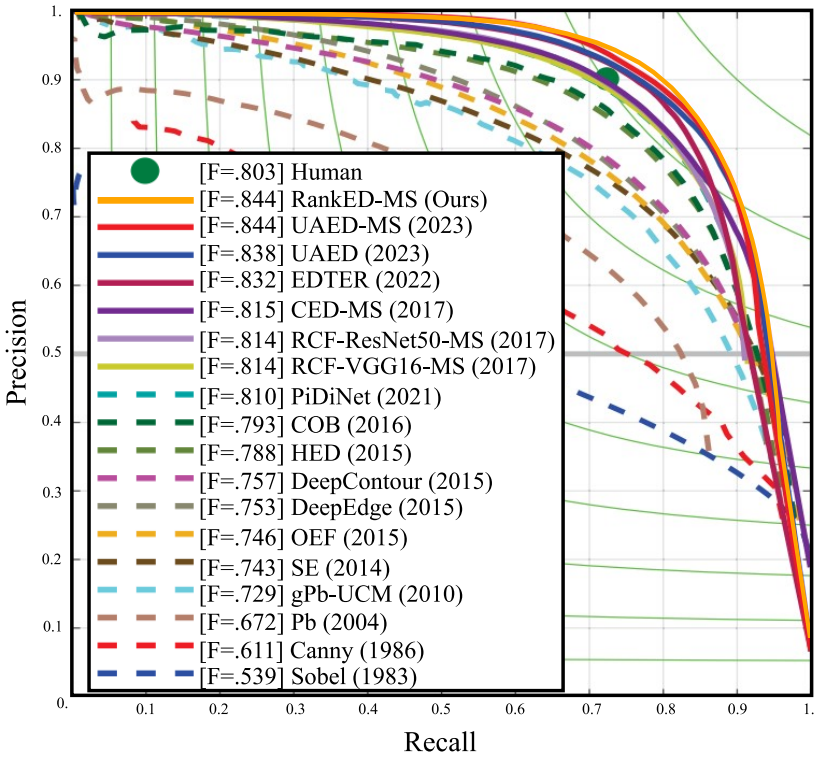


Figure S6. The Precision-Recall curve on BSDS dataset. MS represents multi-scale results.

References

[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, 2010. 1, 2, 3

[2] Gedas Bertasius, Jianbo Shi, and Lorenzo Torresani. Deepedge: A multi-scale bifurcated deep network for top-down contour detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4380–4389, 2015. 3

[3] John Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6): 679–698, 1986. 3

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 248–255. IEEE, 2009. 1

[5] Ruoxi Deng and Shengjun Liu. Deep structural contour detection. In *Proceedings of the 28th ACM international conference on multimedia*, pages 304–312, 2020. 3

[6] Ruoxi Deng, Chunhua Shen, Shengjun Liu, Huibing Wang, and Xinru Liu. Learning to predict crisp boundaries. In *Eur. Conf. Comput. Vis.*, pages 562–578, 2018. 1, 2, 3

[7] Ruoxi Deng, Shengjun Liu, Jinxin Wang, Huibing Wang, Hanli Zhao, and Xiaoqin Zhang. Learning to decode contextual information for efficient contour detection. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 4435–4443, 2021. 3

[8] Ruoxi Deng, Zhao-Min Chen, Huiling Chen, and Jie Hu. Learning to refine object boundaries. *Neurocomputing*, 557:126742, 2023. 1, 2, 3

[9] Piotr Dollár and C Lawrence Zitnick. Fast edge detection using structured forests. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(8):1558–1570, 2014. 3

[10] Omar Elharrouss, Youssef Hmamouche, Assia Kamal Idrissi, Btissam El Khamlichi, and Amal El Fallah-Seghrouchni. Refined edge detection with cascaded and high-resolution convolutional network. *Pattern Recognition*, 138:109361, 2023. 3

[11] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vis.*, 88:303–338, 2010. 3

[12] Saurabh Gupta, Pablo Arbelaez, and Jitendra Malik. Perceptual organization and recognition of indoor scenes from rgb-d images. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 564–571, 2013. 2

[13] Saurabh Gupta, Ross Girshick, Pablo Arbeláez, and Jitendra Malik. Learning rich features from rgb-d images for object detection and segmentation. In *Eur. Conf. Comput. Vis.*, pages 345–360. Springer, 2014. 1

[14] Sam Hallman and Charless C Fowlkes. Oriented edge forests for boundary detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1732–1740, 2015. 2, 3

[15] Jianzhong He, Shiliang Zhang, Ming Yang, Yanhu Shan, and Tiejun Huang. Bi-directional cascade network for perceptual edge detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3828–3837, 2019. 1, 2, 3

[16] Iasonas Kokkinos. Pushing the boundaries of boundary detection using deep learning. *arXiv preprint arXiv:1511.07386*, 2015. 3

[17] Yu Liu and Michael S Lew. Learning relaxed deep supervision for better edge detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 231–240, 2016. 3

[18] Yun Liu, Ming-Ming Cheng, Xiaowei Hu, Kai Wang, and Xiang Bai. Richer convolutional features for edge detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3000–3009, 2017. 2, 3

[19] Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Pablo Arbeláez, and Luc Van Gool. Convolutional oriented boundaries. In *Eur. Conf. Comput. Vis.*, pages 580–596, 2016. 2, 3

[20] David A Mély, Junkyung Kim, Mason McGill, Yuliang Guo, and Thomas Serre. A systematic comparison between visual cues for boundary detection. *Vision research*, 120:93–107, 2016. 1

[21] Mengyang Pu, Yaping Huang, Yuming Liu, Qingji Guan, and Haibin Ling. Edter: Edge detection with transformer. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1402–1412, 2022. 1, 2, 3

[22] Wei Shen, Xinggang Wang, Yan Wang, Xiang Bai, and Zhijiang Zhang. Deepcontour: A deep convolutional feature learned by positive-sharing loss for contour detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3982–3991, 2015. 3

[23] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *Eur. Conf. Comput. Vis.*, pages 746–760. Springer, 2012. 1, 2

[24] Zhuo Su, Wenzhe Liu, Zitong Yu, Dewen Hu, Qing Liao, Qi Tian, Matti Pietikäinen, and Li Liu. Pixel difference networks for efficient edge detection. In *Int. Conf. Comput. Vis.*, pages 5117–5127, 2021. 1, 2, 3

[25] Yupei Wang, Xin Zhao, and Kaiqi Huang. Deep crisp boundaries. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3892–3900, 2017. 3

[26] Ren Xiaofeng and Liefeng Bo. Discriminatively trained sparse code gradients for contour detection. *Adv. Neural Inform. Process. Syst.*, 25, 2012. 3

[27] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *Int. Conf. Comput. Vis.*, pages 1395–1403, 2015. 2, 3

[28] Dan Xu, Wanli Ouyang, Xavier Alameda-Pineda, Elisa Ricci, Xiaogang Wang, and Nicu Sebe. Learning deep structured multi-scale features using attention-gated crfs for contour prediction. *Adv. Neural Inform. Process. Syst.*, 30, 2017. 2, 3

[29] Wenjie Xuan, Shaoli Huang, Juhua Liu, and Bo Du. Fcl-net: Towards accurate edge detection via fine-scale corrective learning. *Neural Networks*, 145:248–259, 2022. 3

[30] Jimei Yang, Brian Price, Scott Cohen, Honglak Lee, and Ming-Hsuan Yang. Object contour detection with a fully convolutional encoder-decoder network. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 193–202, 2016. 3

[31] Caixia Zhou, Yaping Huang, Mengyang Pu, Qingji Guan, Li Huang, and Haibin Ling. The treasure beneath multiple annotations: An uncertainty-aware edge detector. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 15507–15517, 2023. 2, 3