

Supplementary material of 'Pixel Difference Networks for Efficient Edge Detection'

Zhuo Su^{1,*} Wenzhe Liu^{2,*} Zitong Yu¹ Dewen Hu² Qing Liao³ Qi Tian⁴
Matti Pietikäinen¹ Li Liu^{2,1,†}

¹Center for Machine Vision and Signal Analysis, University of Oulu, Finland

²National University of Defense Technology, China

³Harbin Institute of Technology (Shenzhen), China ⁴Xidian University, China

{zhuo.su, zitong.yu, matti.pietikainen, li.liu}@oulu.fi

{liuwenzhe15, dwhu}@nudt.edu.cn, liaoqing@hit.edu.cn, wywqtian@gmail.com

1. Converting Pixel Difference Convolution (PDC) to Vanilla Convolution

The main goal of the conversion is to make PDC as fast and memory efficient as the vanilla convolution. As introduced in the main paper, the formulations of vanilla convolution and PDC can be written as:

$$y = f(\mathbf{x}, \boldsymbol{\theta}) = \sum_{i=1}^{k \times k} w_i \cdot x_i, \quad (\text{vanilla convolution}) \quad (1)$$

$$y = f(\nabla \mathbf{x}, \boldsymbol{\theta}) = \sum_{(x_i, x'_i) \in \mathcal{P}} w_i \cdot (x_i - x'_i), \quad (\text{PDC}) \quad (2)$$

where, x_i and x'_i are the pixels in the current input local patch, w_i is the weight in the $k \times k$ convolution kernel. $\mathcal{P} = \{(x_1, x'_1), (x_2, x'_2), \dots, (x_m, x'_m)\}$ is the set of pixel pairs picked from the local patch, and $m \leq k \times k$.

The conversion from PDC to vanilla convolution can be done in both the training and inference phases.

Conversion in the Training Phase. Eq. 2 can be transformed to fit the form of Eq. 1, according to the selection strategies of the pixel pairs. Correspondingly, PDC can be converted to vanilla convolution by firstly transforming the kernel weights to a new set of kernel weights, followed by a vanilla convolutional operation. We will discuss Central PDC (CPDC), Angular PDC (APDC) and Radial PDC (RPDC) respectively. The selection strategies of pixel pairs in the three PDC instances are shown in Fig. 1, Fig. 2 and Fig. 3. The transformations of the equations are as follows.

For CPDC (Fig. 1):

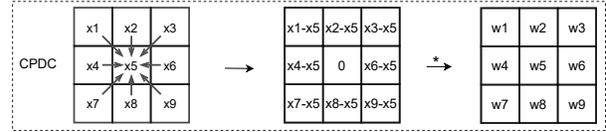


Figure 1. Selection of pixel pairs and convolution in CPDC.

$$\begin{aligned} y &= w_1 \cdot (x_1 - x_5) + w_2 \cdot (x_2 - x_5) + w_3 \cdot (x_3 - x_5) \\ &\quad + w_4 \cdot (x_4 - x_5) + w_6 \cdot (x_6 - x_5) + w_7 \cdot (x_7 - x_5) \\ &\quad + w_8 \cdot (x_8 - x_5) + w_9 \cdot (x_9 - x_5) \\ &= w_1 \cdot x_1 + w_2 \cdot x_2 + w_3 \cdot x_3 + \\ &\quad + w_4 \cdot x_4 + w_6 \cdot x_6 + w_7 \cdot x_7 + \\ &\quad + w_8 \cdot x_8 + w_9 \cdot x_9 \\ &\quad + \left(- \sum_{i=\{1,2,3,4,6,7,8,9\}} w_i \right) \cdot x_5 \\ &= \hat{w}_1 \cdot x_1 + \hat{w}_2 \cdot x_2 + \hat{w}_3 \cdot x_3 + \dots = \sum \hat{w}_i \cdot x_i \end{aligned} \quad (3)$$

For APDC (Fig. 2):

$$\begin{aligned} y &= w_1 \cdot (x_1 - x_2) + w_2 \cdot (x_2 - x_3) + w_3 \cdot (x_3 - x_6) \\ &\quad + w_4 \cdot (x_4 - x_1) + w_6 \cdot (x_6 - x_9) + w_7 \cdot (x_7 - x_4) \\ &\quad + w_8 \cdot (x_8 - x_7) + w_9 \cdot (x_9 - x_8) \\ &= (w_1 - w_4) \cdot x_1 + (w_2 - w_1) \cdot x_2 + (w_3 - w_2) \cdot x_3 \\ &\quad + (w_4 - w_7) \cdot x_4 + (w_6 - w_3) \cdot x_6 + (w_7 - w_8) \cdot x_7 \\ &\quad + (w_8 - w_9) \cdot x_8 + (w_9 - w_6) \cdot x_9 \\ &\quad + 0 \cdot x_5 \\ &= \hat{w}_1 \cdot x_1 + \hat{w}_2 \cdot x_2 + \hat{w}_3 \cdot x_3 + \dots = \sum \hat{w}_i \cdot x_i \end{aligned} \quad (4)$$

For RPDC (Fig. 3):

*Equal contributions. † Corresponding author: <http://lilyliu.com>

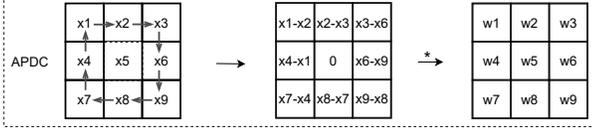


Figure 2. Selection of pixel pairs and convolution in APDC.

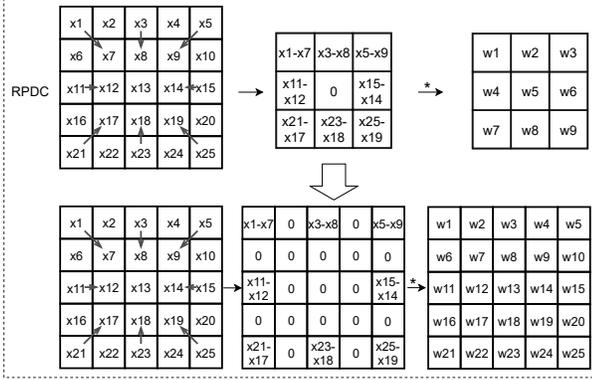


Figure 3. Selection of pixel pairs and convolution in RPDC.

$$\begin{aligned}
y &= w_1 \cdot (x_1 - x_7) + w_3 \cdot (x_3 - x_8) + w_5 \cdot (x_5 - x_9) \\
&\quad + w_{11} \cdot (x_{11} - x_{12}) + w_{15} \cdot (x_{15} - x_{14}) \\
&\quad + w_{21} \cdot (x_{21} - x_{17}) + w_{23} \cdot (x_{23} - x_{18}) \\
&\quad + w_{25} \cdot (x_{25} - x_{19}) \\
&= w_1 \cdot x_1 + w_3 \cdot x_3 + w_5 \cdot x_5 \\
&\quad + (-w_1) \cdot x_7 + (-w_3) \cdot x_8 + (-w_5) \cdot x_9 + \\
&\quad + w_{11} \cdot x_{11} + (-w_{11}) \cdot x_{12} + (-w_{15}) \cdot x_{14} \\
&\quad + w_{15} \cdot x_{15} + (-w_{21}) \cdot x_{17} + (-w_{23}) \cdot x_{18} \\
&\quad + (-w_{25}) \cdot x_{19} + w_{21} \cdot x_{21} + w_{23} \cdot x_{23} \\
&\quad + w_{25} \cdot x_{25} + \sum_{i=\{2,4,6,10,13,16,20,22,24\}} 0 \cdot x_i \\
&= \hat{w}_1 \cdot x_1 + \hat{w}_2 \cdot x_2 + \hat{w}_3 \cdot x_3 + \dots = \sum \hat{w}_i \cdot x_i \quad (5)
\end{aligned}$$

The RPDC is converted to a vanilla convolution with kernel size 5×5 .

Conversion in the Inference Phase. After training, instead of saving the original weights w_i , we directly save the new set of weights \hat{w}_i . Therefore, during inference, all the convolutional operations are vanilla convolutions.

2. Precision-Recall Curves on NYUD Dataset

The Precision-Recall curves of our methods and other approaches on NYUD dataset [8] are shown in Fig. 4. The compared methods include RCF [6], HED [9], SE+NG+ [4], SE [2], gPb+NG [3], gPb-UCM [1] and OEF [5].

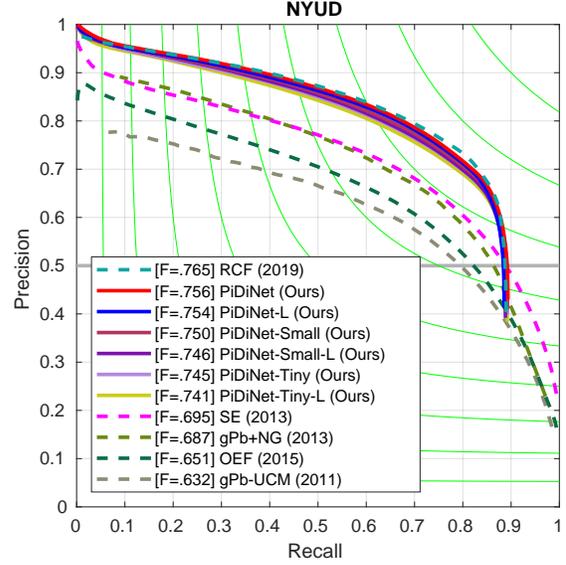


Figure 4. Precision-Recall curves of our models and some competitors on NYUD dataset.

3. Visualization

Edge Maps. The edge maps generated from the baseline architecture and PiDiNet are shown in Fig. 5. Both models were trained using only the BSDS500 dataset without the mixed VOC dataset [7]. From the figure, it is proved that PDC can help PiDiNet effectively capture more useful boundaries, with the ability to extract rich gradient information that facilitates edge detection.

Intermediate Feature Maps. We also visualize the intermediate feature maps extracted from PiDiNet, to qualitatively demonstrate the effectiveness of the compact dilation convolution based module (CDCM) and the compact spatial attention module (CSAM), which are shown in Fig. 6. It is concluded that both CDCM and CSAM take a positive role in PiDiNet on the edge detection task.

References

- [1] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):898–916, 2011. 2, 3
- [2] Piotr Dollár and C Lawrence Zitnick. Fast edge detection using structured forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(8):1558–1570, 2015. 2
- [3] Saurabh Gupta, Pablo Arbelaez, and Jitendra Malik. Perceptual organization and recognition of indoor scenes from rgb-d images. In *CVPR*, pages 564–571, 2013. 2
- [4] Saurabh Gupta, Ross Girshick, Pablo Arbelaez, and Jitendra Malik. Learning rich features from rgb-d images for object detection and segmentation. In *ECCV*, pages 345–360. Springer, 2014. 2

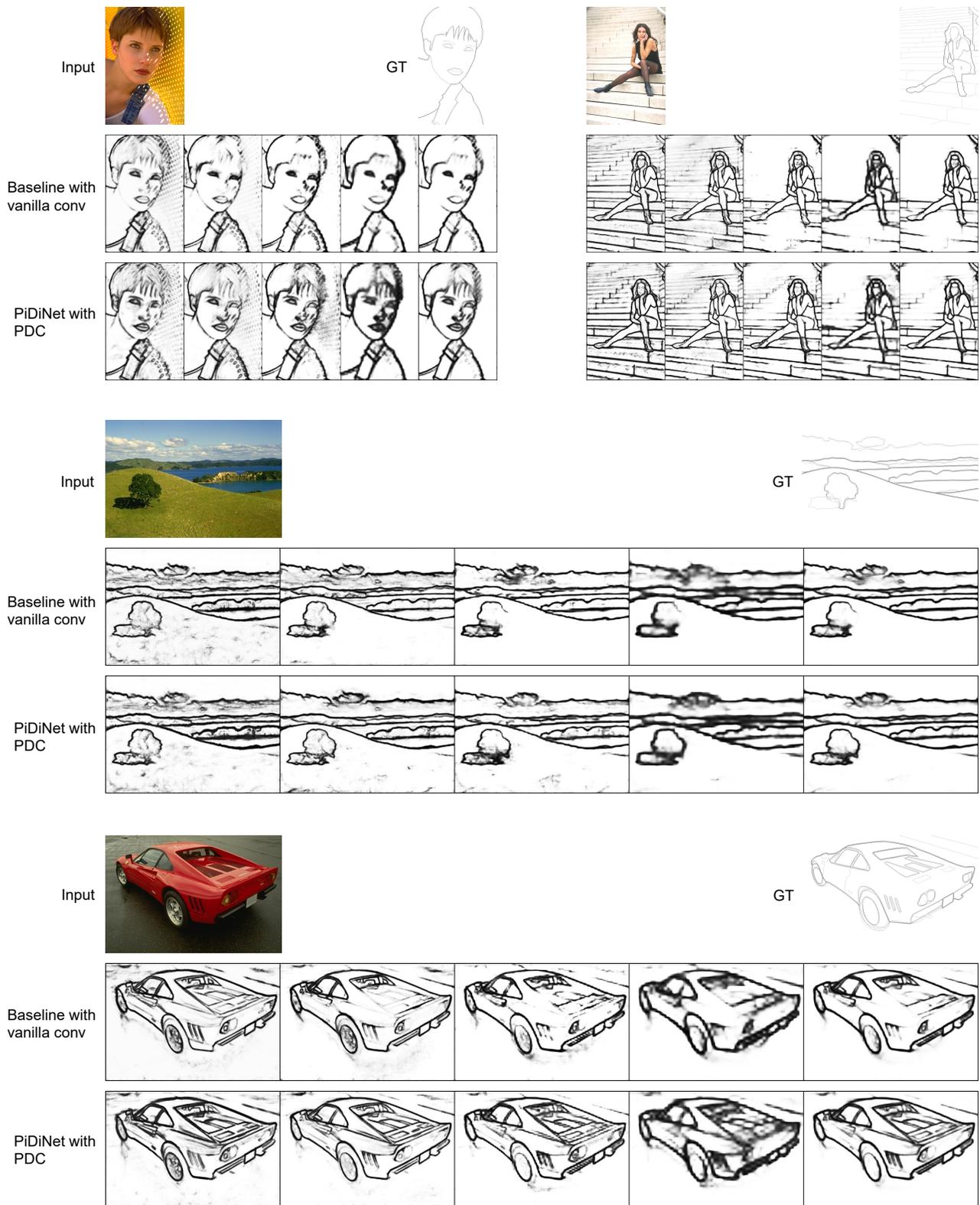


Figure 5. For each case, Top: input and ground truth image; Middle: edge maps from stage 1, 2, 3, 4 respectively and the final edge map, generated from the baseline architecture, Bottom: Corresponding edge maps generated from PiDiNet. Both the baseline architecture and PiDiNet were trained only using the BSDS500 dataset [1]. Compared with the baseline, we can see that PiDiNet can detect more useful boundaries (e.g., bangs, stairs, the contour of the tree, the characteristic textures of the car).

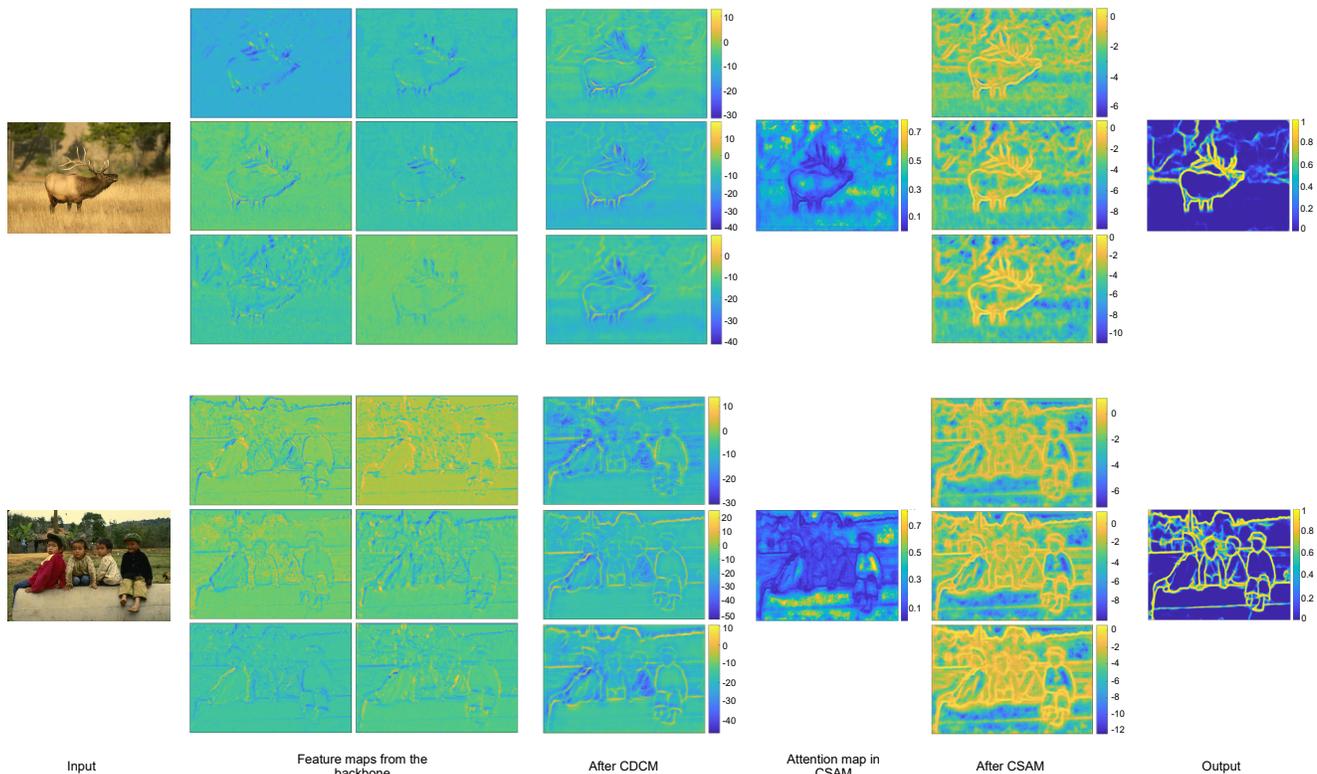


Figure 6. CDCM and CSAM can further refine the feature maps with multi-scale feature extraction and the sample adaptive spatial attention mechanism. Note that in the attention maps generated by CSAM, pixels in the background show higher intensities. This makes sense as the background pixels after CDCM have negative values, hence they will be additionally suppressed through CSAM.

- [5] Sam Hallman and Charless C Fowlkes. Oriented edge forests for boundary detection. In *CVPR*, pages 1732–1740, 2015. 2
- [6] Yun Liu, Ming-Ming Cheng, Xiaowei Hu, Jia-Wang Bian, Le Zhang, Xiang Bai, and Jinhui Tang. Richer convolutional features for edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):1939–1946, 2019. 2
- [7] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *CVPR*, pages 891–898, 2014. 2
- [8] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000. 2
- [9] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. *International Journal of Computer Vision*, 125(1-3):3–18, 2017. 2