SWIN-RIND: Edge Detection for Reflectance, Illumination, Normal and Depth Discontinuity with Swin Transformer

Lun Miao
miaolun@cvl.iis.u-tokyo.ac.jp
Ryoichi Ishikawa
ishikawa@cvl.iis.u-tokyo.ac.jp
Takeshi Oishi
oishi@cvl.iis.u-tokyo.ac.jp

The Institute of Industrial Science,
The University of Tokyo,
Tokyo, Japan

Abstract

Edges are caused by the discontinuities in *surface-**r**eflectance, **i**llumination, surface-**n**ormal, and **d**epth* (RIND). However, extensive research into the detection of specific edge types has not been conducted. Thus, in this paper, we propose a Swin Transformer-based method (referred to as SWIN-RIND) to detect these four edge types from a single input image. Attention-based approaches have performed well in general edge detection and are expected to work effectively for RIND edges. The proposed method utilizes the Swin Transformer as the encoder and a top-down and bottom-up multilevel feature aggregation block as the decoder. The encoder extracts cues at different levels, and the decoder integrates these cues into shared features containing rich contextual information. Then, each specific edge type is predicted through independent decision heads. To train and evaluate the proposed model, we used the public BSDS-RIND benchmark, which is based on the Berkeley Segmentation Dataset and contains annotations for the four RIND-edge types. The proposed method was evaluated experimentally, and the results demonstrate that the proposed SWIN-RIND method outperforms several state-of-the-art methods.

1 Introduction

General edge detection is a fundamental problem in computer vision that has been studied widely [6, 9, 20, 37, 40], and, in recent years, increasing attention has been paid to specific edge detection. As shown in Fig. 1 [23], edges are caused by four factors, i.e., (1) surface-reflectance discontinuity, (2) illumination discontinuity, (3) surface-normal discontinuity, and (4) depth discontinuity (RIND). More detailed categories clarify the essence and benefits of edges. For example, depth edges improve the accuracy of depth estimation [30, 35], high-quality illumination edge detection is required for contrast enhancement and shadow removal [39], and reflectance edges are necessary for road crack detection in intelligent transportation systems [19, 42]. Treating edges without distinguishing types can lead to the

(a) (b) (c) (d) (e) (f)

Figure 1: Examples of specific edges in the BSDS-RIND [28] dataset: (a) reference image; (b) general edges; (c) reflectance edges; (d) illumination edges; (e) normal edges; and (f) depth edges. These examples show that different factors cause edges. Due to the unique nature of these examples, there are no edges in (d) caused by illumination discontinuities, which indicates that the ratios of the RIND edges in images differ and can fluctuate significantly.

loss of important information or increased computational costs for downstream tasks; thus, previous studies have focused on the detection of specific edge types [7, 13, 14, 32].

Pu et al. [28] presented the BSDS-RIND, which is the first public benchmark with RIND-edge labels based on the Berkeley Segmentation Dataset (BSDS) benchmark [3]. In the same study, they also proposed a convolutional neural network (CNN)-based framework to predict RIND edges simultaneously; however, this approach struggles to extract fine edges since the emphasis on its loss design is more inclined towards the local context, and the estimation accuracy was not satisfactory.

Therefore, in this paper, we propose an end-to-end transformer-based RIND-edge detection method that takes a single image as input and predicts the four types of edges simultaneously. The proposed method utilizes the Swin Transformer [21] to encode multilevel cues and construct a top-down and bottom-up decoder for integration. The integrated feature is then transferred to independent decision heads to predict the corresponding specific edge maps. The primary contributions of this study are summarized as follows. (1) We propose an end-to-end network architecture using the Swin Transformer for RIND-edge detection. (2) We propose a combination of dice and attention losses with a self-weighted strategy to realize effective fine edge detection. (3) We demonstrate that the proposed SWIN-RIND detection method outperforms state-of-the-art methods and exhibits significant advantages in accuracy and visual performance.

2 Related Work

Early studies into edge detection [5, 18, 24, 38] focused on general edge detection and primarily utilized image gradients to find the edges. These studies extracted elementary low-level cues, e.g., color, brightness, and texture. These model-based methods were simple and tractable; however, they suffered from inferior performance compared to contemporary learning-based approaches.

With the rapid development of deep learning technologies, the CNN has demonstrated effectiveness in edge detection tasks. Conventional learning-based methods [9, 18] used

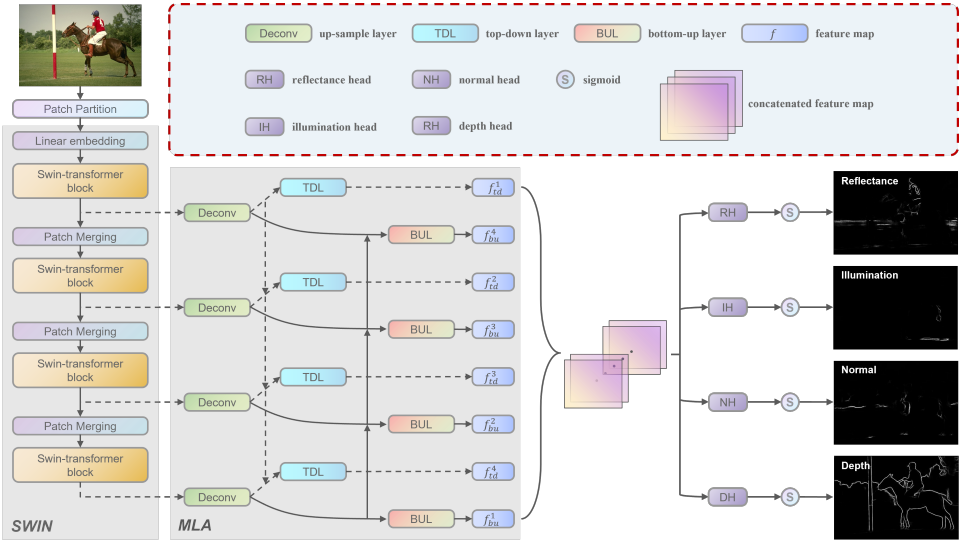


Figure 2: Overall architecture of SWIN-RIND framework. First, the image is input to the Swin Transformer to calculate multilevel semantic information. Second, a top-down and bottom-up multilevel aggregation decoder integrates cues and generates a high-dimensional feature. Finally, each decision head predicts a corresponding specific edge map.

low-level features, e.g., handcrafted features, to train a classifier to predict general edges. In contrast, CNN-based methods [3, 4, 17, 31] integrate multiscale features to achieve outstanding general edge detection performance. Recent CNN-based methods have focused on the specific edge detection task [1, 44, 45]. For example, Fu et al. [11] proposed a CNN model that generates depth edges by parsing optical flow features and small image patches, and Pu et al. [28] proposed a framework that combines multilevel features to generate specific edge maps.

In addition, the Transformer model [33] has been introduced successfully in edge detection research [29]. For example, the Vision Transformer (ViT) [10] divides an image into 16 patches to simulate input via natural language processing, and this method has achieved outstanding performance in various computer vision tasks [29, 36, 41]. A previous study[29] proposed a two-stage general edge detection framework that utilizes the ViT as a backbone network. However, the computational costs incurred by high-resolution images are such that use of the ViT method is not practical, and the global modeling capacity of ViT is limited by its lack of attention interaction between patches. In contrast, the Swin Transformer [21] is a hierarchical transformer that employs a shifted window to enhance interaction between patches.

Building on the success of the Swin Transformer, we propose the SWIN-RIND-edge detection framework to detect all four types of edges simultaneously. The proposed method leverages the hierarchical feature extraction capabilities of the Swin Transformer to learn multiscale and multilevel cues from a single input image. In addition, the dice coefficient is utilized in the loss calculation to enhance global predictions, which results in significant visual and accuracy improvements.

3 Method

An overview of SWIN-RIND is shown in Fig. 2. First, each Swin Transformer block in the SWIN module extracts different levels of information from the input image, and then the MLA decoder integrates multiscale cues and generates a concatenated feature. Finally, four independent decision heads are used to predict each specific edge type.

3.1 SWIN-RIND

3.1.1 Feature map extraction

The edges in an image have rich semantic meanings; thus, it is important to integrate information from both multilevel features and global contexts. A conventional Transformer model might fall short in terms of handling downsampled data and providing attention interaction due to its restricted attention calculation in a small image patch. To address this issue, the proposed method employs the Swin Transformer, which utilizes a shifted window attention calculation process and has a hierarchical architecture.

In the Swin Transformer, the patch partition block first splits an $H \times W$ RGB image into nonoverlapping 4×4 patches. Then, the feature dimension of the input is $3 \times 4 \times 4 = 48$, and the resolution is $\frac{H}{4} \times \frac{W}{4}$. As shown in Fig. 2, four repeated stages are followed by the patch partition module. There are three main modules in the Swin Transformer backbone. The linear embedding module projects the features to an arbitrary dimension C. The Swin Transformer block computes self-attention using a shifted window, which enhances the attention interaction. The patch merging module then downsamples the patches to enhance the feature dimension, which merges 2×2 adjacent patches inside each patch, thereby resulting in a fourfold increase. Then, a built-in linear layer projection process reduces the number of patches by half. We extract the feature maps ω generated by each stage of the Swin Transformer to form group Ω as follows:

$$\Omega = \{\omega_1, \omega_2, \omega_3, \omega_4\}, \tag{1}$$

where $\omega_1, \omega_2, \omega_3, \omega_4$ correspond to the outputs of the four stages, respectively. As a result, the features become more semantically meaningful in the higher stages, which realizes better global modeling capabilities.

3.1.2 Multilevel feature aggregation

The architecture of the decoder has a significant impact on the capability of the encoder. To identify edges, it is important to represent edge pixels in sufficient detail. Thus, inspired by a previous study[29, 46], we designed the MLA decoder with a top-down and bottom-up structure. As shown in Fig. 2, *Deconv* comprises a single deconvolution layer, and *TDL* and *BUL* comprise a 3×3 convolution layer and a deconvolution layer. The top-down and bottom-up paths combine the multilevel features effectively and accelerate the encoder's learning process. In the MLA decoder, the module first utilizes the *Deconv* layers to reshape the feed feature into the same size. Then, it integrates them from top-down and bottom-up paths and concatenates all cues to form a higher-dimension feature. This process is formulated as follows:

$$\mathcal{F} = \Phi_m(\Omega), \tag{2}$$

where Φ_m denotes the MLA decoder, and $\mathcal{F}$ represents the concatenated features.

3.1.3 Decision heads

Four independent decision heads are employed to jointly predict the different specific edge types. Here, each decision head comprises four consecutive 3×3 convolution layers, a single 1×1 convolution layer, a batch normalization layer, and ReLU. The model feeds the concatenated features into an independent decision head Γ to obtain the initial result $\mathcal{O}$:

$$\mathcal{O}_k = \Gamma_k(\mathcal{F}), k \in \{r,i,n,d\} \tag{3}$$

where Γ_k denotes the different decision heads in *Reflectance, Illumination, Normal, Depth*, and $\mathcal{O}_k$ represent the corresponding initial predictions. Finally, the initial prediction results are normalized by the *sigmoid* function σ to generate the final results:

$$\mathcal{Y}_k = \sigma(\mathcal{O}_k), k \in \{r,i,n,d\} \tag{4}$$

where $\mathcal{Y}_k$ denotes different edge predictions in *Reflectance, Illumination, Normal, Depth*.

3.2 Loss function

3.2.1 Attention loss

For edge detection tasks, there is an extreme imbalance between the number of edge pixels and the number of background pixels. Typically, natural images comprise less than 1% edge pixels. In this case, the imbalance makes the loss during the training process overwhelming, and inefficient training leads to model degeneration.

To mitigate the influence of this imbalance, we utilize the loss proposed in a previous study [54], which considers the ratio of edges to background as follows:

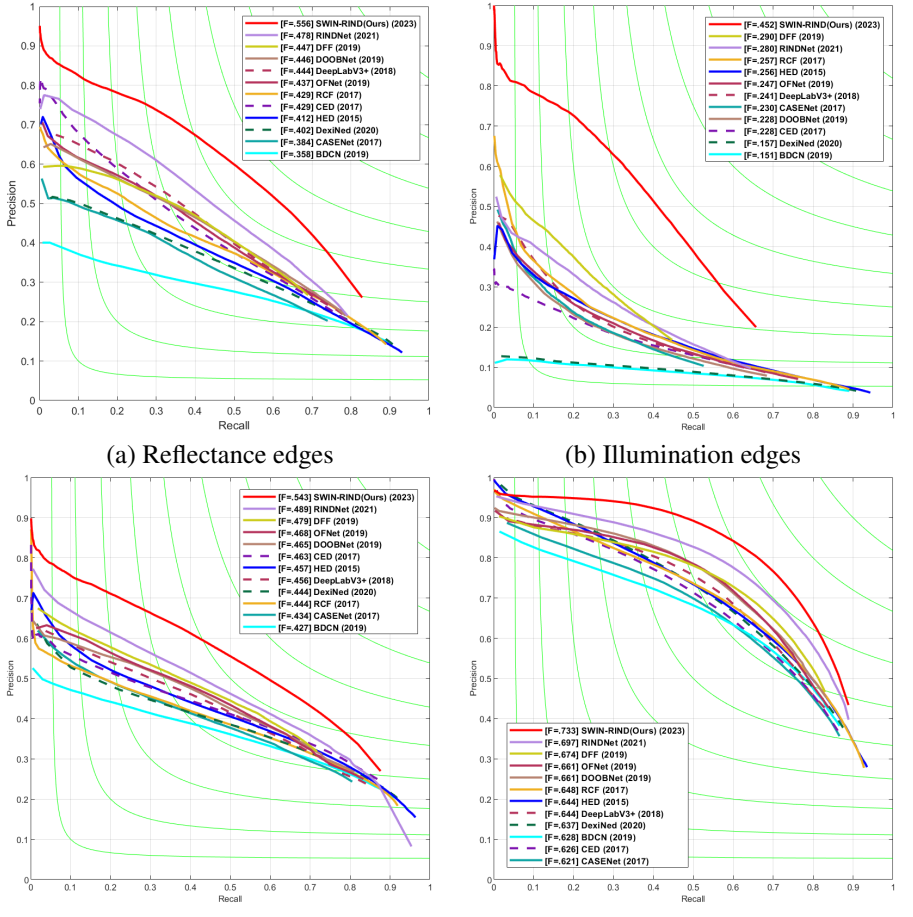
$$\mathcal{L}_a(\mathcal{Y},\mathcal{G}) = -\sum_{(i,j)} \left(\mathcal{G}_{(i,j)} \alpha \beta^{(1-\mathcal{Y}_{(i,j)})^\gamma} \cdot log(\mathcal{Y}_{(i,j)}) + (1-\mathcal{G}_{(i,j)})(1-\alpha)\beta^{\mathcal{Y}_{(i,j)}{}^\gamma} \cdot log(1-\mathcal{Y}_{(i,j)}) \right), \tag{5}$$

where $\mathcal{L}_a$ represents the attention loss, which is applied to each specific edge map, $\mathcal{Y}$ denotes the final prediction of an edge map, and $\mathcal{G}$ represents the corresponding ground truth. In addition, α is the ratio of edge pixels to background pixels, which is calculated as $\alpha = |\mathcal{G}_+|/|\mathcal{G}_-|$, where $|\mathcal{G}_+|$ and $|\mathcal{G}_-|$ correspond to the sum of the edge pixels and background pixels, respectively. Subscripts (i,j) represent the (i_{th}, j_{th}) elements of the matrix, respectively, and γ and β are hyperparameters that can be set manually. For the attention loss, the modulation factor α strongly enhances the penalization of incorrect positive classification, and $(1-\alpha)$ reduces the influence of the number of background pixels.

3.2.2 Dice loss

Edge detection is a binary segmentation task, and it is natural to utilize the binary cross-entropy loss to calculate the loss in such cases; however, the cross-entropy loss focuses more on pixel-level information, which is insufficient for global-level prediction in edge detection tasks. Inspired by a previous study [7, 8, 25], we utilize the dice loss $\mathcal{L}_d$ to allow the network to better learn fine edges as follows:

$$\mathcal{L}_d(\mathcal{Y},\mathcal{G}) = 1 - \frac{2 \cdot \sum_{(i,j)} \mathcal{Y}_{(i,j)} \mathcal{G}_{(i,j)}}{\sum_{(i,j)} \mathcal{Y}_{(i,j)}{}^2 + \sum_{(i,j)} \mathcal{G}_{(i,j)}{}^2}, \tag{6}$$



(a) Reflectance edges (b) Illumination edges

(c) Normal edges (d) Depth edges

Figure 3: Evaluation results obtained on BSDS-RIND dataset for (a) reflectance edges, (b) illumination edges, (c) normal edges, and (d) depth edges. It has been shown that SWIN-RIND has a certain advantage and performs best in all specific edge detection tasks.

where $\mathcal{L}_d$ denotes the dice loss of specific edges. In the dice loss calculation, the value of the dice loss is not greater than one. If the value approaches zero, this indicates that the overall prediction is more accurate (rather than relying on a single pixel).

3.2.3 Self-weighted total loss

Many previous studies [28, 29] manually set parameters to balance multiple losses. However, separate losses typically work against each other during network training. In this case, the most appropriate weights may change during the training process; thus, a combination of losses can be taken as an uncertainty problem [16]. Here, we utilize self-updating parameters $\rho, \tau, \varepsilon, \mu$ to find the most suitable values during the training process, and the total loss $\mathcal{L}_t$ is

Figure 4: **Representative detection results for four edge types.** From left top to right bottom, each column corresponds to: (a) the reference image, (b) RINDNet, (c) SWIN-RIND (the proposed method), and (d) the ground truth. In each column, there are different edge types present in each patch (top left: reflectance edges; top right: illumination edges; bottom left: normal edges; and bottom right: depth edges).

defined as follows:

$$\mathcal{L}_t = \frac{1}{\rho^2}\mathcal{L}_{a,r} + \frac{1}{\tau^2}\mathcal{L}_{a,i} + \frac{1}{\varepsilon^2}\mathcal{L}_{a,n} + \frac{1}{\mu^2}\mathcal{L}_{a,d} + \eta \cdot \sum_k \mathcal{L}_{d,k} + log(\rho\tau\varepsilon\mu), \tag{7}$$

where $\rho, \tau, \varepsilon, \mu$ are the self-updating parameters, and η is an amplification factor that can be set manually. In addition, $\{\mathcal{L}_{a,r}, \mathcal{L}_{a,i}, \mathcal{L}_{a,n}, \mathcal{L}_{a,d}\}$ are the attention losses of the different edge types. $\mathcal{L}_{d,k}, k \in \{r,i,n,d\}$ is the dice loss of the corresponding edges, and $log(\rho\tau\varepsilon\mu)$ is a constraint term to keep the loss factors from overinflating.

4 Experiment

4.1 Dataset

The proposed model was trained and evaluated using the BSDS-RIND [28] dataset. The resolution of the images in this dataset is 321×481 or 481×321 pixels. In addition to general edge annotation, the BSDS-RIND dataset appends RIND-edge labels and augments the

Table 1: Metrics value comparison ("**red**" for the best and "**blue**" for the second best.)

Method	Reflectance			Illumination			Normal			Depth			Average		
	ODS	OIS	AP	ODS	OIS	AP	ODS	OIS	AP	ODS	OIS	AP	ODS	OIS	AP
HED [40]	0.412	0.466	0.343	0.256	0.290	0.167	0.457	0.505	0.395	0.644	0.679	0.667	0.442	0.485	0.393
CED [37]	0.429	0.473	0.361	0.228	0.286	0.118	0.463	0.501	0.372	0.626	0.655	0.620	0.437	0.479	0.368
RCF [20]	0.429	0.448	0.351	0.257	0.283	**0.173**	0.444	0.503	0.362	0.648	0.679	0.659	0.445	0.478	0.386
DFF [45]	0.447	0.495	0.324	**0.290**	**0.337**	0.151	0.479	0.512	0.352	0.674	0.699	0.626	0.473	0.511	0.363
BDCN [12]	0.358	0.458	0.252	0.151	0.219	0.078	0.427	0.484	0.334	0.628	0.661	0.581	0.391	0.456	0.311
OFNet [23]	0.437	0.483	0.351	0.247	0.277	0.150	0.468	0.498	0.382	0.661	0.687	0.637	0.453	0.486	0.380
DexiNed [27]	0.402	0.454	0.315	0.157	0.199	0.082	0.444	0.486	0.364	0.637	0.673	0.645	0.410	0.453	0.352
CASENet [43]	0.384	0.439	0.275	0.230	0.273	0.119	0.434	0.477	0.327	0.621	0.651	0.574	0.417	0.460	0.324
DOOBNet [34]	0.446	0.503	0.355	0.228	0.272	0.132	0.465	0.499	0.373	0.661	0.691	0.643	0.450	0.491	0.376
DeepLabV3+ [6]	0.444	0.487	0.356	0.241	0.291	0.148	0.456	0.495	0.368	0.644	0.671	0.617	0.446	0.486	0.372
RINDNET [28]	**0.478**	**0.521**	**0.414**	0.280	**0.337**	0.168	**0.489**	**0.522**	**0.440**	**0.697**	**0.724**	**0.705**	**0.486**	**0.526**	**0.432**
SWIN-RIND(Ours)	**0.556**	**0.570**	**0.518**	**0.452**	**0.412**	**0.369**	**0.543**	**0.573**	**0.501**	**0.733**	**0.749**	**0.750**	**0.571**	**0.576**	**0.534**

dataset by rotating and flipping each image. There are approximately 2400 annotated training images in the BSDS-RIND dataset.

4.2 Implementation details

We implemented the proposed method in PyTorch [26] and fine-tuned a pretrained Swin Transformer model. Here, we optimized the model in an end-to-end manner using stochastic gradient descent (momentum=0.9, initial learning rate = 10^{-4}, and weight decay=10^{-4}), and the model was trained over 30 epochs with a batch size of 16 using an Nvidia RTX A6000 GPU. In terms of the loss calculation, we set $\beta = 4$ and $\gamma = 0.5$ for the attention loss calculation[34], and $\eta = 10^3$ was used to balance the magnitude between the attention and dice losses. During training, each image is cropped randomly to a size of 320×320 pixels, and the original size is maintained during the testing phase.

4.3 Quantitative evaluation

We compared the proposed method with 11 state-of-the-art edge detection works: HED [40], CED [37], RCF [20], DFF [45], BDCN [12], OFNet [23], DexiNed [27], CASENet [43], DOOBNet [34], DeepLabV3+ [6], and RINDNet [28]. In this evaluation, the BSDS-RIND dataset was used for evaluation, and the performance of the compared methods was evaluated in terms of three metrics [3], i.e., the Optimal Dataset Scale (ODS), Optimal Image Scale (OIS) and Average Precision (AP). In addition, prior to conducting the evaluation, the predicted edge maps were subjected to non-maximum suppression [3]. Table 1 and Fig. 3 show the metrics and F-measure comparison of the four types of edges. As can be seen, the proposed SWIN-RIND method outperformed the compared methods in all metrics.

4.4 Qualitative evaluation

We compared the edge detection results obtained by the proposed SWIN-RIND method with those obtained by the RINDNet method [28], and the results are shown in Fig. 4. From the comparison, we found that thicker edges indicate a higher number of misidentified edge pixels. Thick edge prediction is a typical performance issue introduced by the cross-entropy loss calculation because it focuses more on the pixel itself rather than the entire image. The weighted cross-entropy loss can be very effective in terms of improving edge prediction

in cases where the number of edge pixels is significantly less than that of the background pixels. However, this reduces the loss of background pixels, which in turn reduces the penalty incurred due to their misidentification. In this case, methods that only use the weighted cross-entropy loss generate thicker edges. In contrast, the proposed SWIN-RIND method, which utilizes dice loss to support attention loss, generates fine edge results and achieves better accuracy. Dice loss through a global calculation to rectify the learning process. In addition, we implemented the self-updating strategy to improve the integration of both loss functions.

4.5 Ablation study

We conducted an ablation study on the BUL and TDL layers inside the decoder and loss components to demonstrate their effectiveness. The results are shown in in Table 2 and Table 3, respectively. In terms of the network architecture, we consider the encoder, *Deconv* layers, and the decision heads as indispensable components. Thus, here, the primary focus of the ablation experiments was the BUL and TDL layers. As shown in Table 2, the impact of the unidirectional decoding on boundary comprehension was rather limited, and the bidirectional decoding process provided the highest accuracy. For the experiments conducted in terms of the loss calculation, we tested different combinations of loss components and self-learning parameters. Here, L_a and L_d denote the attention loss and dice loss, respectively, and $SP1$ and $SP2$ denote the self-learning parameter set $1/\{\rho,\tau,\varepsilon,\mu\}$ and set $1/\{\rho^2,\tau^2,\varepsilon^2,\mu^2\}$. CT refers to the constraint term $log(\rho\tau\varepsilon\mu)$. We believe that the dice loss plays an important role in global control. Without global control of the dice loss, coefficient $1/\{\rho^2,\tau^2,\varepsilon^2,\mu^2\}$ reduce the loss value too fast to train. The situation remains unchanged even when attempting to decrease the impact of the coefficient to $1/\{\rho,\tau,\varepsilon,\mu\}$.

MLA	BUL	TDL	ODS	OIS	AP
-	-	-	0.461	0.427	0.418
✓	×	✓	0.552	0.521	0.515
✓	✓	×	0.535	0.500	0.505
✓	✓	✓	**0.571**	**0.576**	**0.534**

Table 2: Ablation study on network architecture

L_a	L_d	SP1	SP2	CT	ODS	OIS	AP
✓	×	×	×	×	0.492	0.476	0.439
✓	×	×	✓	✓	0.113	0.114	0.052
✓	×	✓	×	✓	0.425	0.407	0.349
✓	✓	×	×	×	0.489	0.464	0.441
✓	✓	✓	×	✓	0.573	0.561	0.557
✓	✓	×	✓	✓	**0.571**	**0.576**	**0.534**

Table 3: Ablation study on loss components

5 Conclusion

In this paper, we have proposed a Swin Transformer-based end-to-end network to detect RIND edges simultaneously. The proposed SWIN-RIND method also employs a self-balance loss calculation strategy, which results in promising accuracy and visual effects. The experimental results have shown that the proposed SWIN-RIND method outperforms several state-of-the-art edge detection methods.

However, specific edge detection remains an open challenge. For example, the number of RIND-edge datasets is extremely limited. To the best of our knowledge, BSDS-RIND is the only dataset containing RIND-edge labels. In terms of performance, developing a unified model that excels at both general and specific edge detection requires further investigation.

References

[1] David Acuna, Amlan Kar, and Sanja Fidler. Devil is in the edges: Learning semantic boundaries from noisy annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11075–11083, 2019.

[2] Nicholas Apostoloff and Andrew Fitzgibbon. Learning spatiotemporal t-junctions for occlusion detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 553–559. IEEE, 2005.

[3] 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, 2010.

[4] Gedas Bertasius, Jianbo Shi, and Lorenzo Torresani. Deepedge: A multi-scale bifurcated deep network for top-down contour detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4380–4389, 2015.

[5] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986.

[6] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.

[7] Ruoxi Deng, Chunhua Shen, Shengjun Liu, Huibing Wang, and Xinru Liu. Learning to predict crisp boundaries. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 562–578, 2018.

[8] Lee R Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945.

[9] Piotr Dollar, Zhuowen Tu, and Serge Belongie. Supervised learning of edges and object boundaries. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1964–1971. IEEE, 2006.

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[11] Huan Fu, Chaohui Wang, Dacheng Tao, and Michael J Black. Occlusion boundary detection via deep exploration of context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 241–250, 2016.

[12] Jianzhong He, Shiliang Zhang, Ming Yang, Yanhu Shan, and Tiejun Huang. Bidirectional cascade network for perceptual edge detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3828–3837, 2019.

[13] Xuming He and Alan Yuille. Occlusion boundary detection using pseudo-depth. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11*, pages 539–552. Springer, 2010.

[14] Derek Hoiem, Andrew N Stein, Alexei A Efros, and Martial Hebert. Recovering occlusion boundaries from a single image. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007.

[15] Yuan Hu, Yunpeng Chen, Xiang Li, and Jiashi Feng. Dynamic feature fusion for semantic edge detection. *arXiv preprint arXiv:1902.09104*, 2019.

[16] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7482–7491, 2018.

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

[18] Joseph Jaewhan Lim, Piotr Dollar, and Charles Lawrence Zitnick III. Learned mid-level representation for contour and object detection, September 18 2014. US Patent App. 13/794,857.

[19] Huajun Liu, Xiangyu Miao, Christoph Mertz, Chengzhong Xu, and Hui Kong. Crackformer: Transformer network for fine-grained crack detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3783–3792, 2021.

[20] Yun Liu, Ming-Ming Cheng, Xiaowei Hu, Kai Wang, and Xiang Bai. Richer convolutional features for edge detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3000–3009, 2017.

[21] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[22] Rui Lu, Feng Xue, Menghan Zhou, Anlong Ming, and Yu Zhou. Occlusion-shared and feature-separated network for occlusion relationship reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10343–10352, 2019.

[23] David Marr. *Vision: A computational investigation into the human representation and processing of visual information*. MIT press, 2010.

[24] David R Martin, Charless C Fowlkes, and Jitendra Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE transactions on pattern analysis and machine intelligence*, 26(5):530–549, 2004.

[25] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)*, pages 565–571. Ieee, 2016.

[26] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[27] Xavier Soria Poma, Edgar Riba, and Angel Sappa. Dense extreme inception network: Towards a robust cnn model for edge detection. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1923–1932, 2020.

[28] Mengyang Pu, Yaping Huang, Qingji Guan, and Haibin Ling. Rindnet: Edge detection for discontinuity in reflectance, illumination, normal and depth. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6879–6888, 2021.

[29] Mengyang Pu, Yaping Huang, Yuming Liu, Qingji Guan, and Haibin Ling. Edter: Edge detection with transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1402–1412, 2022.

[30] Michael Ramamonjisoa, Yuming Du, and Vincent Lepetit. Predicting sharp and accurate occlusion boundaries in monocular depth estimation using displacement fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14648–14657, 2020.

[31] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3982–3991, 2015.

[32] Andrew N Stein and Martial Hebert. Occlusion boundaries from motion: Low-level detection and mid-level reasoning. *International journal of computer vision*, 82:325–357, 2009.

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[34] Guoxia Wang, Xiaochuan Wang, Frederick WB Li, and Xiaohui Liang. Doobnet: Deep object occlusion boundary detection from an image. In *Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part VI 14*, pages 686–702. Springer, 2019.

[35] Xiaolong Wang, David Fouhey, and Abhinav Gupta. Designing deep networks for surface normal estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 539–547, 2015.

[36] Yikai Wang, TengQi Ye, Lele Cao, Wenbing Huang, Fuchun Sun, Fengxiang He, and Dacheng Tao. Bridged transformer for vision and point cloud 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12114–12123, 2022.

[37] Yupei Wang, Xin Zhao, and Kaiqi Huang. Deep crisp boundaries. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3892–3900, 2017.

[38] Holger Winnemöller, Jan Eric Kyprianidis, and Sven C Olsen. Xdog: An extended difference-of-gaussians compendium including advanced image stylization. *Computers & Graphics*, 36(6):740–753, 2012.

[39] Qi Wu, Wende Zhang, and BVK Vijaya Kumar. Strong shadow removal via patch-based shadow edge detection. In *2012 IEEE International Conference on Robotics and Automation*, pages 2177–2182. IEEE, 2012.

[40] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1395–1403, 2015.

[41] Peng Xu, Xiatian Zhu, and David A Clifton. Multimodal learning with transformers: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[42] Fan Yang, Lei Zhang, Sijia Yu, Danil Prokhorov, Xue Mei, and Haibin Ling. Feature pyramid and hierarchical boosting network for pavement crack detection. *IEEE Transactions on Intelligent Transportation Systems*, 21(4):1525–1535, 2019.

[43] Zhiding Yu, Chen Feng, Ming-Yu Liu, and Srikumar Ramalingam. Casenet: Deep category-aware semantic edge detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5964–5973, 2017.

[44] Zhiding Yu, Weiyang Liu, Yang Zou, Chen Feng, Srikumar Ramalingam, BVK Kumar, and Jan Kautz. Simultaneous edge alignment and learning. In *Proceedings of the European conference on computer vision (ECCV)*, pages 388–404, 2018.

[45] Mingmin Zhen, Jinglu Wang, Lei Zhou, Shiwei Li, Tianwei Shen, Jiaxiang Shang, Tian Fang, and Long Quan. Joint semantic segmentation and boundary detection using iterative pyramid contexts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13666–13675, 2020.

[46] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6881–6890, 2021.