DBRNet:Dual-Branch Real-Time Segmentation NetWork For Metal Defect Detection

Abstract. Metal surface defect detection is an important task for quality control in industrial production processes, and the requirements for accuracy, and running speed are becoming increasingly high. However, maintaining the realization of real-time surface defect segmentation remains a challenge due to the complex edge details of metal defects, inter-class similarity, and intra-class differences. For this reason, we propose Dual-branch Real-time Segmentation NetWork (DBRNet) for pixel-level defect classification on metal surfaces. First, we propose the Low-params Feature Enhancement Module (LFEM), which improves the feature extraction capability of the model with fewer parameters and does not significantly reduce the inference speed. Then, to solve the problem of inter-class similarity, we design the Attention Flow-semantic Fusion Module (AFFM) to effectively integrate the high-dimensional semantic information into the low-dimensional detail feature map by generating flow-semantic offset positions and using global attention. Finally, the Deep Connection Pyramid Pooling Module (DCPPM) is proposed to aggregate multi-scale context information to realize the overall perception of the defect. Experiments on NEU-Seg, MT, and Severstal Steel Defect Dataset show that the DBRNet outperforms the other state-of-the-art approaches in balance accuracy, speed, and params. The code is publicly available at `https://github.com/fffcompu/DBRNet-Defect`

Keywords: Metal surface defect detection · real-time · semantic segmentation.

1 Introduction

In the industrial production process, many types of defects inevitably appear on the surface of metal materials, such as spots, scratches, and surface inclusions, which will have a negative impact on the performance of the product or even pose a safety hazard, so surface metal defect detection methods become critical. However, metal defect detection is still a challenge due to the complex edge details, large differences in the same defects class (Intra-class differences), and the local similarity problem of different defects class (Inter-class similarity).

With the development of deep learning. Semantic segmentation is proposed, which is different from Faster RCNN [15] and YOLO [2,9] series of target detection methods, which can achieve pixel-level classification and more accurate prediction of boundaries. Therefore, generic segmentation methods [17,1,25,3] are widely used in metal defect segmentation. Zhang et al. [13], used a dual parallel attention module added to the encoder of DeepLabV3+ to improve local

representation and enrich context dependence achieved 89.95 mIoU on the steel defect dataset. Zhang et al. [22] proposed MCNet, using a dense block pyramid pool to make full use of context information to accomplish surface defect segmentation of tracks. Zhan et al. [20] proposed the BSU-Net, which combines the U-Net with a feature expansion network and demonstrated its effectiveness in steel defect segmentation. Dong et al. [5] proposed PGA-Net with pyramidal feature fusion blocks and global context attention blocks, which effectively connect various feature maps extracted from the backbone network, establishe a deep supervision mechanism and obtains precise bounds for segmenting various defects. Damacharla [4] proposes the TL-UNet approach, which uses a migration learning approach to accomplish automatic surface segmentation. However, although the above network structure can segment the defects well, it does not achieve real-time results due to the complexity of its structure.

In recent years, many scholars are researching fast segmentation methods for road scenes. BiseNetV2 [21] proposes a bilateral structure with two branches to extract semantic features and detailed features respectively, which ensures real-time performance while extracting spatial information efficiently. ShuffleNetV2 [12] uses channel shuffle and group convolution to reduce computational cost. STDCNet [6] designs a short-term dense cascade module to extract multi-scale features and proposes a detail aggregation module to learn the decoder, which can preserve the underlying spatial details more accurately without increasing the inference time. DDRNet [7] introduces bilateral connections to enhance the information exchange between context and detail branches for precise segmentation. However, metal defects differ significantly from road scenes, thus affecting the accuracy and robustness of these algorithms when processing images containing metal defects. To achieve high-precision defect segmentation while maintaining real-time speed, this paper proposes DBRNet with pixel-level metal defect prediction, and the contributions of this paper are summarized as follows:

(1) We propose a Low-params Feature Enhancement Module (LFEM), which uses depth-wise convolution with different kernel sizes to reduce the number of parameters and enhance feature extraction without significantly slowing down the model's inference speed.

(2) To deal with the problem of inter-class similarity, Attention Flow-semantic Fusion Module(AFFM) is proposed, using global attention and semantic flow makes it effectively fuse high-level semantic features while preserving detailed features.

(3) The Deep Connection Pyramid Pooling Module (DCPPM) is designed to be added to the semantic branch to improve the overall perception of defects by using dense connections to fuse different pooling features.

(4) The experimental results on three publicly available defect datasets demonstrate the effectiveness of the proposed DBRNet for metal surface defect segmentation.

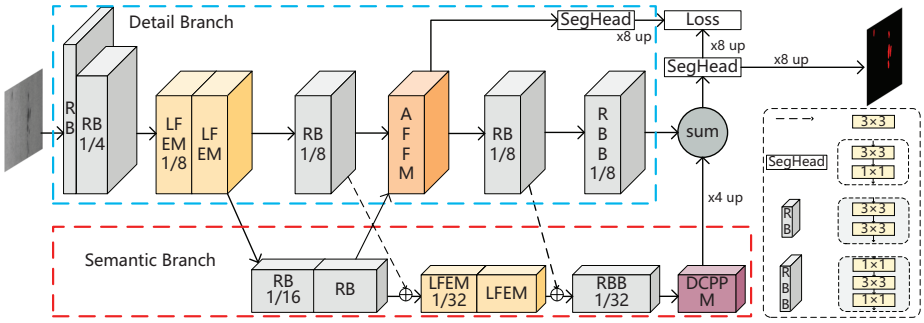

Fig. 1. The overall architecture diagram of the network. 3×3, 1×1 represents the standard convolution using the corresponding kernel size. ⊕ and "sum" indicated element-wise addition. "up" represents upsampling operation.

2 Method

The overall structure of the network is shown in Fig. 1. The model is based on dual-branch architecture. Where the semantic branch is concerned with the category information of defects and the detail branch is concerned with the location details of defects. The whole network consists of the LFEM, AFFM, DCPPM, and ResNet's Residual Bottleneck Block (RBB) and Residual Basic Block (RB) modules. Firstly, two LFEM are used to replace the original RB blocks in the detail branch and semantic branch to generate 1/8 and 1/32 feature maps of the input image resolution, respectively, to enhance the feature extraction capability of the model. Afterwards, AFFM is added before the fourth RB of the detail branch to efficiently fuse semantic features into the high-resolution image. The feature maps of the detail branch are fused into the semantic branch by 3×3 convolution downsampling before LFEM and RBB of the semantic branch to improve edge recognition ability. Finally, the "sum" element-wise addition combines the detail branch and semantic branch information. And the output is obtained by SegHead operation. It is important to note that the processing of SegHead and Auxiliary loss is kept consistent with that of DDRNet [7].

2.1 Low-params Feature Enhancement Module

The number of parameters and the model inference speed are not a single linear relationship, as most devices are optimized for 3×3 standard convolution, making RB block composed of 3×3 convolution to run faster. However, simply stacking the RB block makes the number of parameters of the model rise dramatically and the feature extraction method is homogeneous. To balance params, speed, and accuracy, We propose Low-params Feature Enhancement Module (LFEM) inspired by ShuffleNetV2 [12] and LAANet [23] for replacing the original RB block which generates feature maps of 1/8 input resolution size at the detail branch and 1/32 input resolution size at the semantic branch.

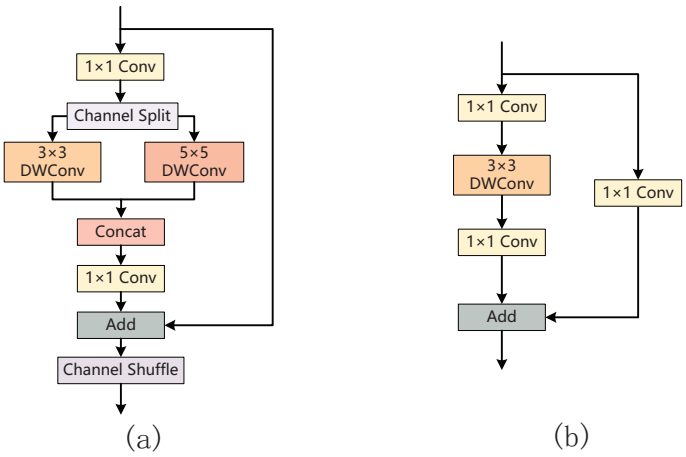

Fig. 2. The details of LFEM. (a) is utilized when the stride=1. (b) is based the mobile inverted bottleneck, which is used when the stride=2. Conv represents standard convolution. DWConv is the depth-wise convolution.

LFEM is shown in Fig. 2. It can be divided into two situations. When the stride is equal to 1, at the beginning of LFEM, the channel number of input is expanded 2× to obtain an upgrade feature map through 1×1 convolution to enhance the ability of feature information extraction. To maintain computing efficiency, we will then divide the upgraded feature maps into two branches. Each branch obtains a half-channel upgrade feature map by channel split, and then uses 3×3 depth-wise convolution on the first branch to extract local information, use 5×5 depth-wise convolution on the second branch to obtain more complex feature information. The outputs of two branches are connected through channel connection. Afterward, 1×1 convolution is used to increase information sharing between branches, followed by performing residual add connections to prevent gradient disappearance and explosion. Finally, channel shuffle is used to promote information fusion between channels, thereby achieving higher segmentation accuracy. When the stride is equal to 2, it is similar to the MobileNet structure. However, in order to improve the information retention ability during the downsampling process, 1×1 standard convolution is used for downsampling the residual part.

2.2 Attention Flow-Semantic Fusion Module

In defect segmentation, the local similarity of different defects makes it difficult to identify the class information of defects. Integrating semantic information into detail feature maps can enable segmentation results to retain detailed information while containing rich category information. However, directly using element-wise operations to fuse semantic features into detail feature maps leads to misalignment and loss of feature information. We are inspired by SFNet [11].

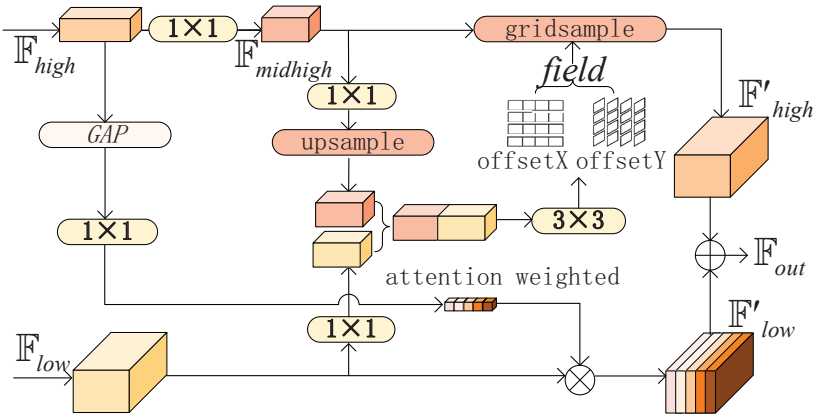

Fig. 3. The architecture of AFFM. ⊗ indicated element-wise multiplication operation. *GAP* indicated global average pooling operation. gridsample represents the grid_sample function.

We propose AFFM that uses global attention mechanisms and semantic offset position of adjacent features to adaptively align high-level semantic feature into low-level detail feature to improve the ability to identify defects class.

As Shown in Fig. 3. The input of AFFM is the $\mathbb{F}_{high} \in \mathbb{R}^{H/2\times W/2\times 2C}$ represents the high-level feature map from the semantic branch and the $\mathbb{F}_{low} \in \mathbb{R}^{H\times W\times C}$ represents the low-level feature map from the detail branch, with the output being $\mathbb{F}_{out} \in \mathbb{R}^{H\times W\times C}$. We first use global average pooling for the high-level feature to quickly extract global context information, followed immediately by using 1×1 convolution to adjust to the same number of channels as low-level feature and multiplying the obtained attention weighted $\in \mathbb{R}^{1\times 1\times C}$ with low-level feature to achieve a simple way to weight low-level feature to highlight detailed information. To cope with the inefficient feature fusion between high-level and low-level features caused by the difference in resolution and information contained, we resample the high-level feature to achieve feature alignment by setting a learnable semantic offset position $field \in \mathbb{R}^{H\times W\times 2}$. Specifically, we use 1×1 convolution to perform channel compression on the high-level feature to get the $\mathbb{F}_{midhigh} \in \mathbb{R}^{H/2\times W/2\times C}$, and then use 1×1 convolution and upsampling to generate a contrast feature map. The low-level feature map is only processed with 1×1 convolution operation to generate a contrast feature map with the same shape $H \times W \times C/2$. These two comparison feature maps are fused in a channel-connected manner and fed into 3×3 convolution to explore the semantic offset positions between features at different levels. The semantic offset position $field$ consists of offsetX and offsetY, which represent the offset position on the horizontal axis and the vertical axis, respectively, of the sampling coordinates of each pixel point during the sampling process. The operation of gridsample is used to resample the $\mathbb{F}_{midhigh}$ to generate the $\mathbb{F}'_{high}$ embedded in the correct position in the low-level feature. Finally, the element-wise addition operation

combines $\mathbb{F}^{'}_{low}$ and $\mathbb{F}^{'}_{high}$ to obtain the final output $\mathbb{F}_{out}$. The entire process can be formulated as:

$$\begin{aligned}
\mathbb{F}^{'}_{low} &= \mathrm{Conv}_{1\times1}\left(GAP\left(\mathbb{F}_{high}\right)\right) \otimes \mathbb{F}_{low} \\
\mathbb{F}_{midhigh} &= \mathrm{Conv}_{1\times1}\left(\mathbb{F}_{high}\right) \\
field &= \mathrm{Conv}_{3\times3}\left(\mathrm{Concat}\left(\mathrm{Conv}_{1\times1}\left(\mathbb{F}_{low}\right), \mathrm{up}\left(\mathrm{Conv}_{1\times1}\left(\mathbb{F}_{midhigh}\right)\right)\right)\right) \\
\mathbb{F}^{'}_{high} &= \mathrm{gridsample}\left(\mathbb{F}_{midhigh}, field\right) \\
\mathbb{F}_{out} &= \mathbb{F}^{'}_{low} \oplus \mathbb{F}^{'}_{high}
\end{aligned} \tag{1}$$

Where $\mathrm{Conv}_{i\times i}$ indicated standard convolution with i × i kernel size. Concat indicated channel connection operation.

2.3 Deep Connection Pyramid Pooling Module

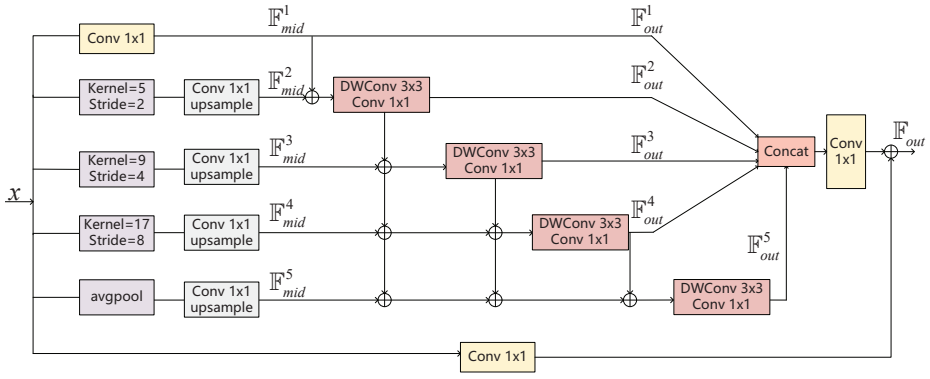

Fig. 4. The details of DCPPM.

For defect segmentation, some defects are intra-class differences, which require context information to achieve the overall perception of metal defects. Pooling operations can be used to get multi-scale context information quickly. However, traditional pooling modules lack information sharing after the pooling operation. For this purpose, we designed DCPPM, which uses pooling operations with different kernel sizes, and dense join to obtain more fine-grained multi-scale context information in the semantic branch.

As shown in Fig. 4. The input $x \in \mathbb{R}^{H\times W\times C}$ is fed to five parallel branches to perform multi-scale feature fusion operation. The 5th branch uses the global average pooling operation $\mathrm{P}_g\left(.\right)$. The 2nd branch to 4th branch utilize P_i, $i \in \{2,3,4\}$ with kernel size$\{5,9,17\}$ pooling operations. Then Except for the first branch, the other four branches were subjected to pooling operation, after which $\mathbb{F}^i_{mid} \in \mathbb{R}^{H\times W\times C/4}$ was generated by 1×1 standard convolution and upsampling

operation for subsequent fusion. The process of obtaining $\mathbb{F}^i_{mid}$ is as follows:

$$\mathbb{F}^i_{mid}\begin{cases}\mathrm{Conv}_{1\times1}(x) & i=1;\\ \mathrm{up}\left(\mathrm{Conv}_{1\times1}\left(\mathrm{P}_i(x)\right)\right) & 1<i<5\\ \mathrm{up}\left(\mathrm{Conv}_{1\times1}\left(\mathrm{P}_g(x)\right)\right) & i=5.\end{cases} \tag{2}$$

The branches of larger pooling kernel are fused with deeper feature information as a way to obtain multi-scale information at different fine-grained levels. Specifically, to get the output $\mathbb{F}^i_{out}$ of each branch, the 1st branch does not have any operation, $\mathbb{F}^0_{mid}$ is directly used as output $\mathbb{F}^0_{out}$. The output of 2nd branch and 3rd branch is the result of summing $\mathbb{F}^i_{mid}$ and $\mathbb{F}^{i-1}_{out}$ and then using the 3×3 depth-wise convolution and 1×1 standard convolution operations. The 4th and 5th branches are summed by $\mathbb{F}^i_{mid}$ and $\mathbb{F}^{i-1}_{out}$, $\mathbb{Q}_{i-1}$, immediately also followed by depth-wise convolution and standard convolution operation to obtain the output. $\mathbb{Q}_i$ represents the output of the intermediate addition operation of branch i. f stands for 3×3 depth-wise convolution and 1×1 standard convolution.

$$\mathbb{F}^i_{out}\begin{cases}\mathbb{F}^i_{mid} & i=1;\\ f\left(\mathbb{F}^i_{mid}\oplus\mathbb{F}^{i-1}_{out}\right) & 1<i\leq3\\ f\left(\mathbb{F}^i_{mid}\oplus\mathbb{F}^{i-1}_{out}\oplus\mathbb{Q}_{i-1}\right) & 4\leq i\leq5.\end{cases} \tag{3}$$

The $\mathbb{F}^i_{out}$ of all paths are stacked together and fed into 1×1 convolution, and 1×1 convolution is also used for the input x. Then, the two outputs are summed to obtain $\mathbb{F}_{out}$.

3 Experiments

3.1 Datasets and Evaluation Metrics

Datasets. The NEU-Seg [5] Dataset consists of 3600 hot-rolled steel strip images with a resolution of 200×200. It contains three types of defects, Inclusion, Patches and Scratches. Each type of defect contains 1200 samples. To evaluate the FPS, we resize the image's resolution to 512×512. Each image was annotated at the semantic level.

MT Dataset [8], it contains 5 types of magnetic-tile defects, porosity, cracks, wear, fracture, unevenness. It contains 392 defect images and 952 images without defects. All images are not the same resolution, so the image size resize to 512×512, and each image is made defective segmentation of the label.

The Severstal Steel Defect Dataset [18] contains 12568 steel images with a resolution of 1600×256 and pixel-level annotations of the four defect categories, and the sample distribution in the dataset contains 5902 defect-free samples and 6666 defect samples.

Mertics. In this paper, we use the evaluation metrics widely used in the field of Real-Time Segmentation. mean Intersection over Union (mIoU), Frames Per Second (FPS), Params, and Floating Point operations (FLOPs) to measure the quality and efficiency of the model.

3.2 Implementation Details

Using the SGD optimizer with momentum and linear learning rate strategy. The SGD momentum value was set to 0.9, the initial learning rate was set to 1e-2, the weight decay factor was set to 5e-4. For data augmentation, the NEU-Seg and MT datasets, we used random augmentation to 0.5 to 2.5 followed by random cropping to 512×512, and the Severstal Steel Defect Dataset was randomly cropped to 512×256. The batch size during training was set to 8, all datasets were divided into train:val:test=6:2:2, and the loss function was OHEM. All experiments were performed on NVIDIA RTX3080 with PyTorch 1.11.0 and cuda11.3 version.

3.3 Comparison Experiments

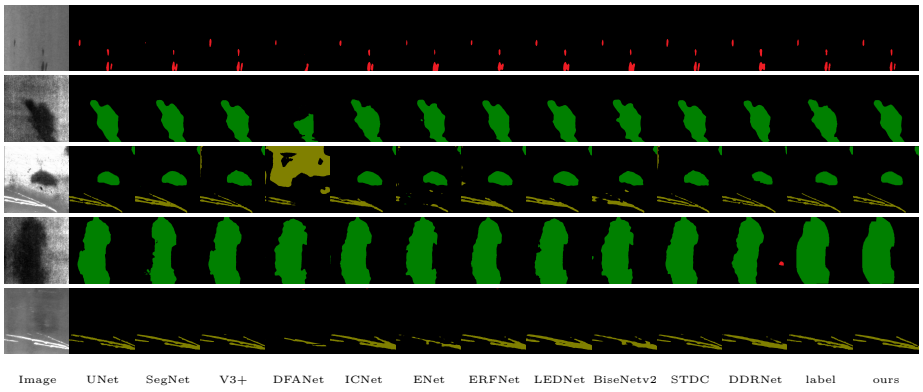

Fig. 5. Visualisation of results on NEU-Seg dataset.

We compared our DBRNet with 11 representative current segmentation methods on three publicly available typical defect segmentation datasets. Table 1 and Table 2 show the experimental results.

Results on NEU-Seg Dataset. On this dataset, DBRNet achieves 123.40 FPS at 512×512 of the image's resolution and achieves 83.14% mIoU. DBRNet's mIoU and FPS is the best compared to other methods. Specifically, compared with DDRNet, our method's Params, and FLOPs decreased by 2.35MB and 1.45G, respectively, with a small increase in FPS. In terms of accuracy, DBRNet's mIoU is 1.89% higher than DDRNet. DBRNet is also competitive among DeepLabV3+, U-Net methods in accuracy. The comparison visualization on this dataset is shown in Fig. 5. It can be seen that the results predicted by DBRNet are close to the ground truth (label) in the detail part, which demonstrates its effectiveness in metal surface defect segmentation.

Table 1. Comparative experiments on NEU-Seg and MT datasets

Method	Resolution	Params(M)	FLOPs(G)	FPS	mIoU	
					NEU-Seg	MT
U-Net [17]	512×512	28.95	361.23	26.24	82.65	31.52
SegNet [1]	512×512	29.44	160.02	53.67	76.70	15.30
DeepLabV3+ [3]	512×512	54.70	83.20	47.24	82.79	30.50
DFANet [10]	512×512	2.15	**1.79**	39.26	45.02	20.07
ICNet [24]	512×512	26.70	5.58	56.00	79.13	61.35
ENet [14]	512×512	**0.35**	2.42	83.48	78.79	38.18
ERFNet [16]	512×512	2.06	12.89	84.87	79.09	35.13
LEDNet [19]	512×512	0.91	5.72	69.80	78.99	48.09
BiseNetV2 [21]	512×512	5.19	17.75	117.90	81.38	62.05
STDCNet [6]	512×512	9.34	11.37	121.81	82.09	62.58
DDRNet [7]	512×512	5.69	4.89	121.70	81.25	67.18
ours	512×512	3.34	3.44	**123.40**	**83.14**	**70.51**

Table 2. Comparative experiments on Severstal Steel Defect Dataset

Method	Resolution	Params(M)	FLOPs(G)	FPS	mIoU
U-Net [17]	1600×256	28.95	564.42	17.05	54.65
SegNet [1]	1600×256	29.44	251.53	33.43	44.20
DeepLabV3+ [3]	1600×256	54.70	130.00	36.40	60.07
DFANet [10]	1600×256	2.15	**2.79**	38.00	35.83
ICNet [24]	1600×256	26.70	8.73	55.00	60.02
ENet [14]	1600×256	**0.35**	7.04	72.31	47.89
ERFNet [16]	1600×256	2.06	20.14	83.87	55.30
LEDNet [19]	1600×256	0.91	8.94	70.50	45.79
BiseNetV2 [21]	1600×256	5.19	27.75	112.00	59.08
STDCNet [6]	1600×256	9.34	7.27	116.12	57.64
DDRNet [7]	1600×256	5.69	7.63	**122.30**	59.54
ours	1600×256	3.34	5.41	120.06	**60.61**

Results on MT Dataset. On this dataset, the FPS, Params, and FLOPs of DBRNet are the same as in NEU-Seg, because the resolution of the images is the same. The difference is that the accuracy of our method on this dataset is significantly higher than the other comparison methods, with mIoU of 70.51%, which proves that DBRNet is equally effective in segmenting metal defects with a more number of types.

Results on Severstal Steel Defect Dataset. DBRNet's mIoU is also the best on this dataset. Although the FPS metric is lower than DDRNet 2.24, the mIoU of DBRNet is 1.07% higher than DDRNet. The mIoU of ICNet is similar to that of DBRNet, but our method is faster and has smaller number of parameters. Compared to other methods, our method achieves an excellent balance between speed, params, and accuracy on this dataset.

3.4 Ablation Study

To verify the effectiveness of the proposed method, ablation experiments were performed on three datasets. It should be noted that our baseline is modified DDRNet and ablation method is a replacement of the original module. Table 3 and Table 4 show the ablation experimental results.

Table 3. Ablation in NEU-Seg and MT datasets

Baseline	LFEM	AFFM	DCPPM	Params(M)	FLOPs(G)	FPS	mIoU	
							NEU-Seg	MT
✓				5.53	4.27	**137.01**	79.55	66.38
✓	✓			3.84	3.58	130.71	80.57	68.26
✓		✓		5.54	4.29	129.64	80.46	67.88
✓			✓	5.01	4.14	133.33	79.87	67.09
✓	✓		✓	**3.32**	**3.45**	126.59	81.40	68.87
✓	✓	✓		3.85	3.59	124.29	82.38	69.70
✓	✓	✓	✓	3.34	3.74	123.40	**83.14**	**70.51**

Table 4. Ablation in Severstal Steel Defect Dataset

Baseline	LFEM	AFFM	DCPPM	Params(M)	FLOPs(G)	FPS	mIoU
✓				5.53	6.68	**133.60**	58.64
✓	✓			3.84	5.59	125.31	60.02
✓		✓		5.54	6.70	123.40	59.83
✓			✓	5.01	6.47	132.00	59.31
✓	✓		✓	**3.32**	**5.39**	123.30	60.37
✓	✓	✓		3.85	5.62	122.74	60.54
✓	✓	✓	✓	3.34	5.41	120.06	**60.61**

Effectiveness of single module. Compared to the baseline. After adding LFEM to replace corresponding RB blocks, FPS only decreased by approximately 7 but the number of parameters decreased by 1.69MB and mIoU increased by 1.02% , 1.88% , 1.38% in NEU-Seg, MT, Severstal Steel Defect Dataset respectively. This suggests that the addition of LFEM is more effective in extracting feature capability than using RB blocks alone. The number of parameters increased by just 0.01M after adding AFFM, and mIoU increased by 0.91% , 1.5% , and 1.19% on the three datasets. This demonstrates that AFFM can effectively fuse semantic information and solve the similarity problem of different defects without adding significant parameters. The addition of DCPPM increased miou by 0.32% , 0.71% , and 0.67% on the three datasets, demonstrating that it can aggregate multi-scale information to achieve an overall perception of defects.

Effectiveness of Different module combinations. In addition to the single analysis of the modules, some experiments were set up to evaluate the effect

of the combination of the different modules. As seen in rows 5 to 7 of Table 3 and Table 4. Each combination improved the accuracy of the model. The best results were achieved when all three modules were added, specifically, the amount of parameters of the model were reduced by 2.19MB and the mIoU reached 83.14%, 70.51% , and 60.61% on the three datasets. The results show that the addition of the three modules can effectively improve the performance of the model.

4 Conclusion

In this paper, we propose DBRNet that implements real-time metal defect segmentation. For the model to have high accuracy and efficiency. First, we design the LFEM module with fewer params to enhance the ability of feature extraction. secondly, to solve the inter-class similarity problem, we use the AFFM module, which can more effectively fuse high-dimensional semantic information into low-dimensional detail information. In addition, we added DCPPM to the semantic branch to increase the multi-scale information of the model and thus realize the overall perception of defects. Based on the experimental results on three datasets, the algorithm shows excellent performance in terms of model efficiency and segmentation results. This indicates that the model can be deployed on metal defect detectors to achieve real-time segmentation of metal defects.

References

1. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE transactions on pattern analysis and machine intelligence **39**(12), 2481–2495 (2017)
2. Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M.: Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934 (2020)
3. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the ECCV. pp. 801–818 (2018)
4. Damacharla, P., Rao, A., Ringenberg, J., Javaid, A.Y.: Tlu-net: a deep learning approach for automatic steel surface defect detection. In: 2021 International Conference on Applied Artificial Intelligence (ICAPAI). pp. 1–6. IEEE (2021)
5. Dong, H., Song, K., He, Y., Xu, J., Yan, Y., Meng, Q.: Pga-net: Pyramid feature fusion and global context attention network for automated surface defect detection. IEEE Transactions on Industrial Informatics **16**(12), 7448–7458 (2019)
6. Fan, M., Lai, S., Huang, J., Wei, X., Chai, Z., Luo, J., Wei, X.: Rethinking bisenet for real-time semantic segmentation. In: Proceedings of the IEEE/CVF conference on CVPR. pp. 9716–9725 (2021)
7. Hong, Y., Pan, H., Sun, W., Jia, Y.: Deep dual-resolution networks for real-time and accurate semantic segmentation of road scenes. arXiv preprint arXiv:2101.06085 (2021)
8. Huang, Y., Qiu, C., Yuan, K.: Surface defect saliency of magnetic tile. The Visual Computer **36**, 85–96 (2020)
9. Jocher, G.: Yolov5. `https://github.com/ultralytics/yolov5` (2021)

10. Li, H., Xiong, P., Fan, H., Sun, J.: Dfanet: Deep feature aggregation for real-time semantic segmentation. In: Proceedings of the IEEE/CVF conference on CVPR. pp. 9522–9531 (2019)
11. Li, X., You, A., Zhu, Z., Zhao, H., Yang, M., Yang, K., Tan, S., Tong, Y.: Semantic flow for fast and accurate scene parsing. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16. pp. 775–793. Springer (2020)
12. Ma, N., Zhang, X., Zheng, H.T., Sun, J.: Shufflenet v2: Practical guidelines for efficient cnn architecture design. In: Proceedings of the ECCV. pp. 116–131 (2018)
13. Pan, Y., Zhang, L.: Dual attention deep learning network for automatic steel surface defect segmentation. Computer-Aided Civil and Infrastructure Engineering **37**(11), 1468–1487 (2022)
14. Paszke, A., Chaurasia, A., Kim, S., Culurciello, E.: Enet: A deep neural network architecture for real-time semantic segmentation. arXiv preprint arXiv:1606.02147 (2016)
15. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems **28** (2015)
16. Romera, E., Alvarez, J.M., Bergasa, L.M., Arroyo, R.: Erfnet: Efficient residual factorized convnet for real-time semantic segmentation. IEEE Transactions on Intelligent Transportation Systems **19**(1), 263–272 (2017)
17. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18. pp. 234–241. Springer (2015)
18. Severstal: Steel defect detection, kaggle challange 2019. `https://www.kaggle.com/c/severstal-steel-defect-detection` (2019)
19. Wang, Y., Zhou, Q., Liu, J., Xiong, J., Gao, G., Wu, X., Latecki, L.J.: Lednet: A lightweight encoder-decoder network for real-time semantic segmentation. In: 2019 IEEE international conference on image processing (ICIP). pp. 1860–1864. IEEE (2019)
20. Xinzi, Z.: Bsu-net: A surface defect detection method based on bilaterally symmetric u-shaped network. In: 2020 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE). pp. 1771–1775. IEEE (2020)
21. Yu, C., Gao, C., Wang, J., Yu, G., Shen, C., Sang, N.: Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation. International Journal of Computer Vision **129**, 3051–3068 (2021)
22. Zhang, D., Song, K., Xu, J., He, Y., Niu, M., Yan, Y.: Mcnet: Multiple context information segmentation network of no-service rail surface defects. IEEE Transactions on Instrumentation and Measurement **70**, 1–9 (2020)
23. Zhang, X., Du, B., Wu, Z., Wan, T.: Laanet: lightweight attention-guided asymmetric network for real-time semantic segmentation. Neural Computing and Applications **34**(5), 3573–3587 (2022)
24. Zhao, H., Qi, X., Shen, X., Shi, J., Jia, J.: Icnet for real-time semantic segmentation on high-resolution images. In: Proceedings of the ECCV. pp. 405–420 (2018)
25. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: Proceedings of the IEEE CVPR. pp. 2881–2890 (2017)