A-Net: An A-Shape Lightweight Neural Network for Real-Time Surface Defect Segmentation

Biao Chen, Tongzhi Niu, Wenyong Yu, *Senior Member, IEEE*, Ruoqi Zhang, Zhenrong Wang, and Bin Li

***Abstract*— Surface defect segmentation is a critical task in industrial quality control. Existing neural network architectures often face challenges in providing both real-time performance and high accuracy, limiting their practical applicability in time-sensitive, resource-constrained industrial setting. To bridge this gap, we introduce A-Net, an A-shape lightweight neural network specifically designed for real-time surface defect segmentation. Initially, A-Net introduces a pioneering A-shaped architecture tailored to efficiently handle both low-level details and high-level semantic information. Secondly, a series of lightweight feature extraction blocks are designed, explicitly engineered to meet the stringent demands of industrial defect segmentation. Finally, rigorous evaluations across multiple industry-standard benchmarks demonstrate A-Net's exceptional efficiency and high performance. Compared to the well-estabilished U-Net, A-Net achieves comparable or superior intersection over union (IoU) scores with gains of −0.21%, −0.3%, +4.7%, and +5.94% on NEU-seg, DAGM-seg, MCSD-seg, and MT dataset, respectively. Remarkably, A-Net does so with only 0.39M parameters, a 98.8% reduction, and 0.44G floating point operations (FLOPs), a 99% decrease in computational load. Besides, A-Net shows extremely fast inference speed on edge device without GPU because of its low FLOPs. A-Net contributes to the development of effective and efficient defect segmentation networks, suitable for real-world industrial applications with limited resources.**

***Index Terms*— Lightweight neural network, neural network architecture, real-time neural network, surface defect detection.**

NOMENCLATURE

Symbol	Quantity
$\text{Params}_{\text{Conv}}$	Parameter number of general convolution.
$\text{FLOPs}_{\text{Conv}}$	FLOPs of general convolution.
$\text{Params}_{\text{dwConv}}$	Parameter number of DW-Conv.
$\text{FLOPs}_{\text{dwConv}}$	FLOPs of DW-Conv.
C_i	Number of input channels for convolution.
C_o	Number of output channels for convolution.
K_h	Convolution kernel height.
K_w	Convolution kernel width.
F_h	Feature map height.
F_w	Feature map width.
TP	Positive predictions that match with the ground truth.
FP	Positive predictions that do not match with ground truth.
opr	Different operations in every stage.
S_n	nth stage in A-Net.
C_{in}	Number of input channels for different stage.
C_m	Number of intermediate channels in different stage.
C_{out}	Number of output channels in different stage.
LFEM	Different lightweight feature extraction modules.
L_{weight}	Loss function in training stage.
L_{dice}	Dice loss function.
L_{bce}	Binary cross-entropy loss function.
p_d	Predicted pixel of models.
g_d	Pixel of ground truth.
FN	Negative predictions that do not match with ground truth.

Manuscript received 26 September 2023; revised 6 November 2023; accepted 15 November 2023. Date of publication 8 December 2023; date of current version 22 December 2023. This work was supported by the National Natural Science Foundation of China under Grant 52375494. The Associate Editor coordinating the review process was Dr. Wenqiang Liu. *Biao Chen and Tongzhi Niu contributed equally to this work.) (Corresponding author: Wenyong Yu.)*

Biao Chen, Tongzhi Niu, Wenyong Yu, Zhenrong Wang, and Bin Li are the School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan 430000, China (e-mail: u202010899@hust.edu.cn; tzniu@hust.edu.cn; ywy@hust.edu.cn; zora_wang@hust.edu.cn; libin999@hust.edu.cn).

Ruoqi Zhang is with the China-EU Institute for Clean and Renewable Energy, Huazhong University of Science and Technology, Wuhan 430000, China (e-mail: m202271390@hust.edu.cn).

Digital Object Identifier 10.1109/TIM.2023.3341115

I. INTRODUCTION

IN RECENT years, the field of defect segmentation has gained significant prominence as a crucial aspect of industrial surface defect detection. The objective is to precisely locate and size defects for effective quality control [1], [2]. Advances in semantic segmentation architectures, such as fully convolutional networks (FCN) [3], SegNet [4], U-Net [5], and PGA-Net [6], have improved performance metrics. Concurrently, there is a growing requirement for efficient algorithms capable of low-latency edge deployment in computationally constrained environments. This has led to a rising interest in the development of defect segmentation networks that optimize the trade-off between computational efficiency and effectiveness.

Recently, numerous researchers have proposed the design of low-latency, high-efficiency CNN models that maintain satisfactory segmentation accuracy. We will discuss segmentation network design from two perspectives: the architectures and the lightweight approaches.

Regarding the architectures, there are three prevalent approaches: 1) encoder–decoder architectures excel at pixel-wise prediction through comprehensive feature extraction, but may lack nuanced context understanding (e.g., FCN

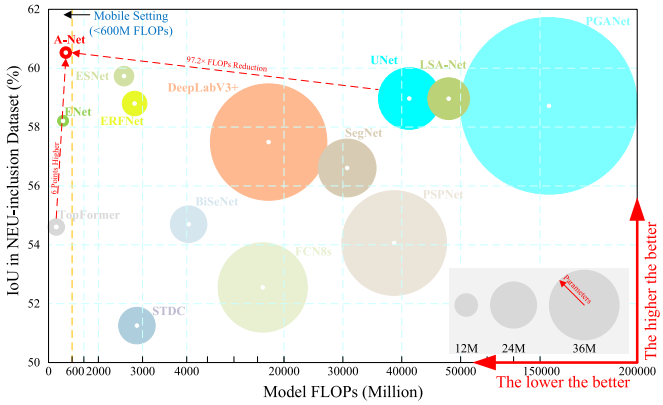

Fig. 1. Comparisons between classical and lightweight semantic segmentation networks and the A-Net on NEU-inclusion dataset.

[3], SegNet [4], U-Net [5], and PGA-Net [6]); 2) pyramid pooling architectures mitigate this by integrating multiscale features, albeit at the cost of increased computational complexity (e.g., PSPNet [7], DeepLab [8]); and 3) bilateral architectures provide a balanced approach by adopting a multipath framework to combine low-level details with high-level semantics (e.g., BiSegNet [9], BiSegNet V2 [10], STDC-MA network (STDC) [11]).

In the pursuit of lightweight approaches, several key techniques warrant thorough investigation: 1) input restriction and channel pruning serve as direct methods for reducing computational load, albeit with trade-offs in representation capacity (e.g., ENet [12], ICNet [13]); 2) weight quantization and knowledge distillation emerge as sophisticated techniques to compact the model further, targeting both the storage and computational facets (e.g., [14], [15]); and 3) the well-designed convolution blocks, such as depthwise separable convolutions, provide architectural innovations that strike a balance between efficiency and performance (e.g., ERFNet [16], ENet [12]).

Moreover, we analyze the challenges of designing a lightweight model for surface defect segmentation: 1) limited defect image availability hinders lightweight models, which are inherently limited in their feature extraction capabilities; 2) defects' varied sizes and irregular shapes have been tackled by prior methods using large-scale or dilated convolutions and pyramid structures, but these add computational complexity; and 3) the subtle differences between defective and normal areas complicate segmentation. While multiple skip connections and auxiliary training branches can improve accuracy, they increase memory overhead.

To overcome the aforementioned challenges, we extended bilateral architectures and well-designed convolution blocks to propose a lightweight network called A-Net, which demonstrates strong performance on various surface defect datasets, while maintaining exceptional lightness. Fig. 1 shows the comparisons between classical and lightweight semantic segmentation networks and the A-Net in terms of intersection over union (IoU) performance, model FLOPs, and parameters on NEU-inclusion dataset. Due to the large difference in the number of parameters between the lightweight segmentation network and the universal segmentation network, we enlarge the circle representing A-Net and ENet by a factor of 15; ESNet, ERFNet, and topformer by a factor of 6; and BiSeNet and STDC by a factor of 1.5 to make the picture more beautiful. It is obvious that the proposed A-Net is superior than all models shown in this figure, while using much fewer FLOPs and parameters.

Initially, we proposed an A-shaped structure, depicted in Fig. 2(d). A-Net retains the encoder–decoder structure and incorporates the concept of bilateral architectures to extract both low-level detailed information and high-level semantic information. Rather than adding additional branches, it is designed with different feature extraction layers within the same path: shallow layers capture detailed information, while deeper layers focus on semantic content. These features are fused through a single skip connection, giving the architecture an 'A'-like shape. As a result, A-Net not only preserves the multiscale feature extraction and fusion characteristics but also minimizes memory usage typically incurred by multiple skip connections.

Subsequently, we designed a series of lightweight convolution blocks comprising: 1) feature extraction blocks, which include a Light Block and a Wide Block corresponding to 3 $\times$ 3 and 5 $\times$ 5 receptive fields, respectively; 2) up- and down-sampling blocks, composed of 2 $\times$ 2 convolutional layers with a stride of 2 and deconvolutional layers, respectively; and 3) concatenation blocks. Within these blocks, we employed depthwise convolution, dropout layers, and residual connection structures to prevent overfitting, gradient vanishing, and gradient explosion issues, thus creating a lightweight network model adaptable to small datasets.

In summary, our main contributions are as follows.

1) We propose a novel network architecture, dubbed A-Net, which extracts information at different levels in stages during the down-sampling stage and facilitates the aggregation of information at various levels through one skip connection in the up-sampling stage.
2) A series lightweight convolution blocks are designed for A-Net. These blocks enhance the receptive field, capture rich contextual information, and prevent severe overfitting on small datasets effectively while minimizing computational costs.
3) A-Net achieves remarkable results on different datasets (NEU-seg, DAGM-seg, MCSD-seg, MT dataset). More specifically, it demonstrates competitive performance against classic large models such as U-Net (with 31.39M parameters and 42.75G FLOPs), requiring only 0.39M parameters and 0.44G FLOPs.

In order to better showcase our work, we have organized the symbols appearing in the paper and provided the corresponding meanings of every symbol in nomenclature, where the DW-Conv represents the depthwise separable convolution. Besides, the code of the work shown in the article is available on GitHub: https://github.com/Max-Chenb/A-Net.

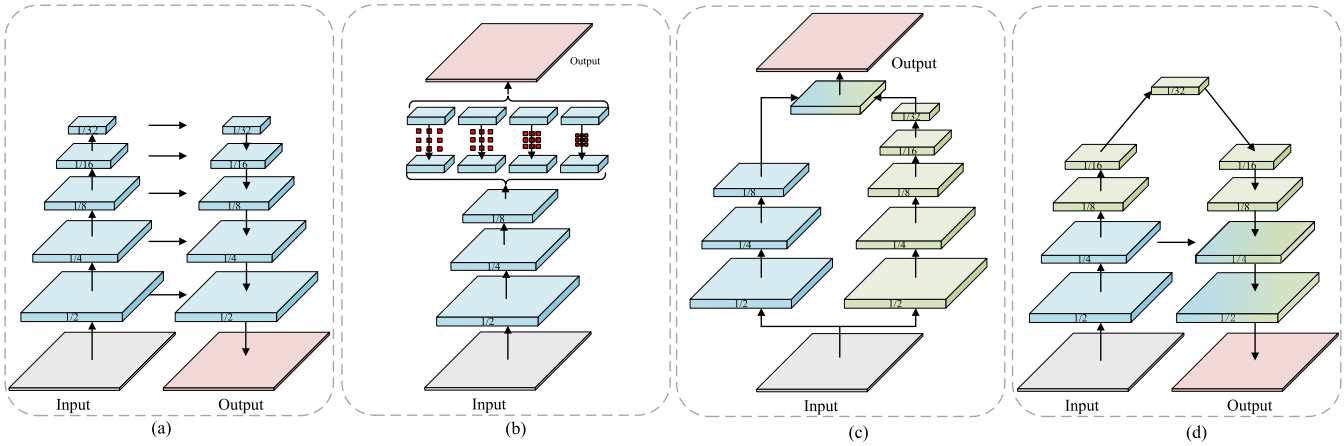

Fig. 2. Three popular semantic segmentation network architectures (a)–(c) and the novel architecture proposed in this article (d). (a) Encoder–decoder backbone. (b) Pyramid pooling backbone. (c) Bilateral segmentation backbone. (d) A-shaped lightweight backbone.

II. Related Work

In recent years, notable progress has been made in the realm of industrial surface defect segmentation. This section centers its examination on three primary categories of methodologies that are particularly germane to our work, specifically generic semantic segmentation, lightweight architectures and real-time semantic segmentation techniques, and industrial surface defect segmentation.

A. Generic Semantic Segmentation

With the introduction of the FCN [3], methods based on this framework have continuously pushed the state-of-the-art performance on various benchmarks. Currently, the mainstream FCN [3] structures are encoder–decoder structures, as depicted in Fig. 2(a). The down-sampling stage captures information of different scales in the input image, while the up-sampling stage recovers the feature map resolution and maps it into semantic segmentation output. To enhance the performance of the encoder–decoder structure, most high-performing semantic segmentation networks employ a horizontal connection structure. For example, U-Net [5] uses a concatenate operation to connect feature maps with the same resolution in the encoder and decoder and then aggregates information in different channels through convolution operation. SegNet [4] uses a method to save maximum pooled coordinates to guide up-sampling. RefineNet [17] performs up-sampling of the encoder's feature map using multipath refinement. DFN [18] employs a channel attention module to merge the backbone network and recover details.

In addition, DeepLab [8] adopts cavity convolution of different sizes at the decoder stage to up-sample the feature map obtained from the encoder stage to the same resolution and aggregate to fuse feature information of different scales, as shown in Fig. 2(b). HRNet [19] utilizes multiple branches to maintain high resolution for higher precision segmentation.

Recently, in order to pursue higher performance, some researchers have introduced transformer [20] in the field of natural language processing (NLP) into visual tasks. The original representative of vision transformer is the ViT model for image classification proposed by Dosovitskiy et al. [21] Its basic idea is to divide the image into several patches and simultaneously input it into the network and convert it into a sequence for operation, so that the perception field can be expanded into the whole image. It improves the ability of the network to extract the overall features of the image, and finally builds a network model suitable for visual tasks. Swim-transformer [22] module is proposed on the basis of ViT to further optimize the attention mechanism. After that, Zheng et al. [23] proposed the first VIT-based image segmentation representative model SERT, which realized end-to-end image segmentation by adding PUP and MLA up-sampling modules. Cao et al. [24] proposed Swim-Unet for image segmentation task and replaced the convolutional layer in U-net with swim-transformer block to further improve the performance.

However, these architectures predominantly rely on operations with a high number of parameters and computational overhead. Consequently, the majority of such networks are characterized by a considerable size and low-inference speed.

B. Lightweight and Real-Time Semantic Segmentation

With the advancement of deep learning, numerous large-scale network models have been proposed. However, due to their high parameter count and computational overhead, it has become challenging to meet the stringent requirements of real-world applications that demand prompt response times. Consequently, researchers have recently shifted their focus toward neural network algorithms that exhibit lightweight and real-time characteristics. Among these, ENet [12] stands out as the pioneer work that emphasizes convolutional neural network efficiency. This network adopts an encoder–decoder structure, employs maximum pooling coordinates to guide up-sampling, and achieves an extremely high reasoning speed. Similarly, ICNet [13] leverages image concatenation strategy to accelerate the network's reasoning speed. ERFNet [16] incorporates residual connections and factorized convolutions to ensure accuracy while improving efficiency. ESNet [25] employs the decomposition of convolutional units and other lightweight convolutional operations to construct a symmetrical structure real-time semantic segmentation network. Finally, DFANet

[26] utilizes feature repetition to decrease network complexity while preserving feature expression.

Despite the ability of above-mentioned networks to achieve a lightweight network structure or real-time inference speed, the aforementioned lightweight or real-time semantic segmentation networks often entail a trade-off between performance and segmentation capability of small-scale features. This is due to their inability to effectively attend to both low-level details and high-level semantic information simultaneously [10].

To address the aforementioned challenges, BiSeNetV2 [10] introduces a bilateral segmentation backbone as illustrated in Fig. 2(c). This architecture incorporates both detailed and semantic branches during the sampling phase to enable the simultaneous extraction of corresponding information, which is subsequently aggregated and directly up-sampled to the output resolution. Despite BiSeNetV2's ability to perform real-time semantic segmentation via GPU-accelerated computing while simultaneously extracting details and semantic information, the network's parameter count and computational requirements remain significant due to the existence of two subsampling branches. Thus, its deployment on industrial edge devices without GPU-accelerated computing is not viable.

Besides, topformer [27] up-samples tokens through pyramid structure, and then integrates tokens of different scales. This method reduces the number of parameters and computational complexity in transformer, and improves the inference speed of the network. However, due to the large amount of data required for its training, it is not applicable to the field of industrial defect detection.

To address these limitations, this paper proposes the A-Net structure illustrated in Fig. 2(d). A-Net employs a specially designed feature extraction module to realize a lightweight network structure and real-time reasoning while aggregating detailed information and semantic information through a single jump connection.

C. Industrial Surface Defect Segmentation

The segmentation of industrial surface defects based on neural networks has garnered increasing attention with the development of deep learning. In recent years, fully convolutional neural network-based methods for industrial surface defect segmentation have emerged continuously. For example, Wang et al. [28] proposed an FCN-based method for refining and segmenting defects in tire images by fusing multiscale sampling layer feature maps, while Yu et al. [29] developed a multistage FCN method to achieve more precise defect segmentation. Moreover, MCuePush U-Net [30] employs a three-channel image output of MCue module as U-Net's input to improve defect segmentation performance, while FL-SegNet [31] combines the original SegNet network with a focal loss function to segment multiple defects in tunnel lining. DeepCrack [32], based on SegNet, fuses multiscale deep convolution features learned at hierarchical convolution stages to capture fine crack structures. Finally, PGA-Net [6] introduces a pyramid feature aggregation and global context attention network to achieve better defect segmentation performance. The aforementioned networks for surface defect segmentation can effectively achieve precise segmentation of specific defects. However, their network architectures are large and require high computational resources, making their deployment and real-time inference at the edge costly. In contrast, the A-Net proposed in this study employs a specially designed network architecture and feature extraction module to achieve a lightweight network structure and real-time inference while maintaining sufficient defect segmentation performance.

III. A-Shaped Lightweight and Real-Time Network

A. Overview

As depicted in Fig. 3, the proposed lightweight real-time industrial defect segmentation network is of A-shaped architecture, hence named A-Net. A-Net is comprised of two distinct parts, namely feature extraction and feature fusion. During the feature extraction stage, the feature maps with darker colors correspond to higher levels of information, while in the feature fusion stage, feature maps with darker colors correspond to a greater degree of detailed information recovery. The feature extraction stage is composed of two stages: detail extraction and semantic extraction. These stages employ different stacking modes of down-sampling module (down block) and lightweight feature extraction module (Light Block and Wide Block). The aim of detail extraction is to extract low-level detailed information more effectively, whereas the goal of semantic extraction is to capture high-level semantic information more precisely. The feature fusion stage employs alternately stacked up-sampling module (up block) and lightweight feature extraction module (Light Block and Wide Block) to achieve refined feature recovery. Further, we aggregate low-level detailed information with high-level semantic information through a jump connection structure specially designed for this purpose. Finally, the segmentation output is obtained through the process of up-sampling, feature fusion, and seg head.

B. Motivation

To achieve a lightweight network structure capable of real-time inferencing on edge devices, it is necessary to minimize the number of parameters and computational complexity of the network. The computational complexity of the network is represented by floating point operations (FLOPs).

Industrial defect images present a challenge to semantic segmentation networks due to the varying sizes and shapes of defect regions. To address this challenge, we integrate detail extraction and semantic extraction in the feature extraction stage and aggregate the extracted information via a jump connection after up-sampling. This approach enables the network to focus on information of different scales in the image simultaneously while maintaining a low parameter number and FLOPs, leading to high-precision semantic segmentation of industrial surface defects.

When the dataset size is small, deep fully convolutional neural networks are susceptible to the issues of gradient disappearance and explosion, which can lead to ineffective convergence. Therefore, we designed a lightweight feature extraction convolutional operation with a residual connection structure to address these issues. Additionally, we adopted

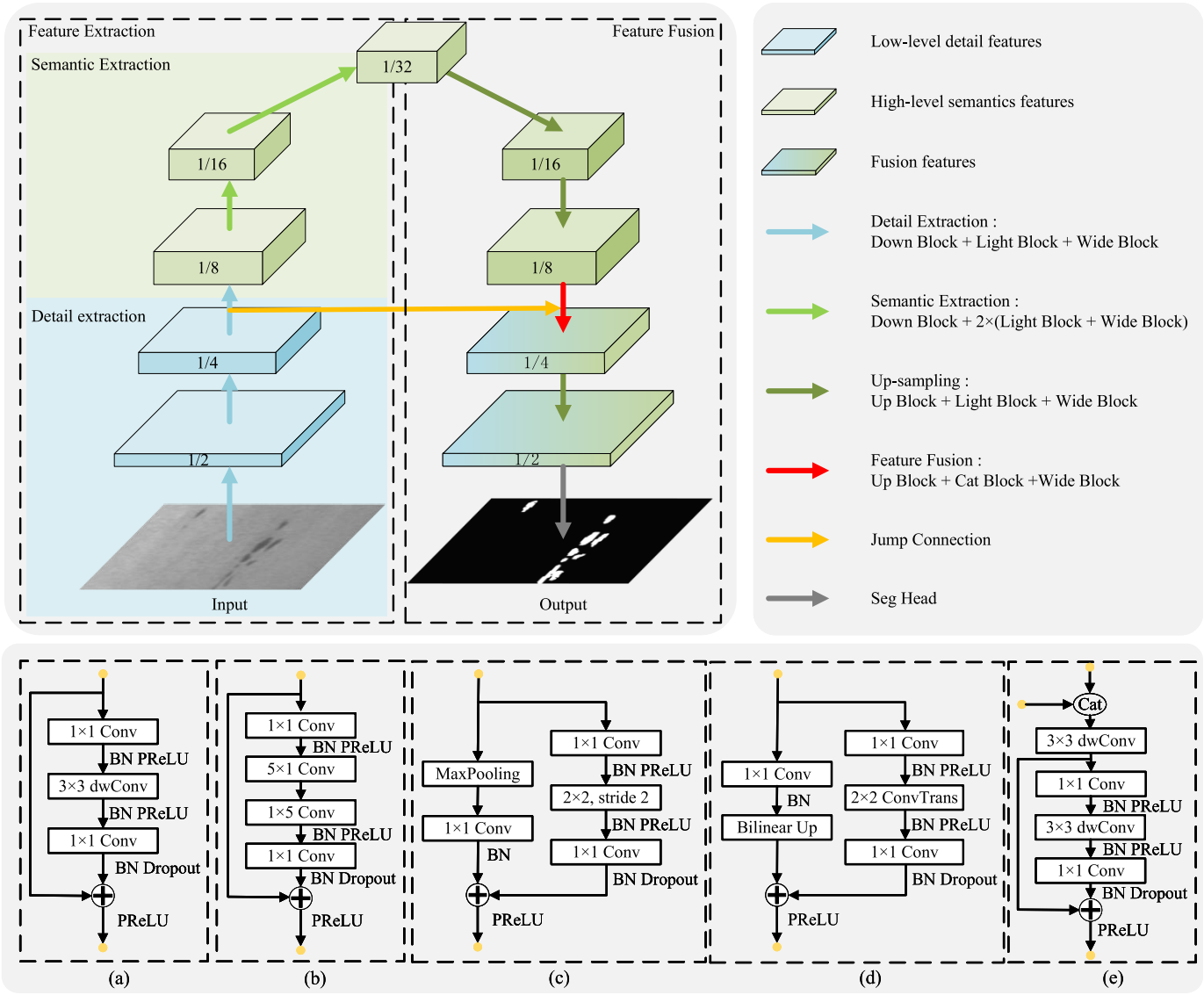

Fig. 3. Architecture of proposed network in this article. (a) Light Block. (b) Wide Block. (c) Down block. (d) Up block. (e) Cat block.

different convolutional operation block stacking modes in different feature extraction stages to further expand the receptive field of the semantic extraction stage. As a result, A-Net achieves effective extraction of low-level details and high-level semantic information with an extremely low parameter number and FLOPs.

Furthermore, to improve the performance of industrial surface defect segmentation and to address the issue of indistinct boundaries between defect and nondefect regions, we have incorporated a staggered design of up-sampling and convolution operation blocks in our feature fusion stage. Nevertheless, this design imposes additional computational overhead. Hence, we have integrated lightweight feature extraction convolution operations, namely Light Block and Wide Block, in both feature extraction and feature fusion stages to mitigate the computational complexity. This approach strikes a balance between computational efficiency and segmentation accuracy, enabling our proposed network to achieve high-precision industrial surface defect segmentation.

C. Feature Extraction

This section presents a detailed description of the down block and two lightweight feature extraction blocks (Light and Wide Block). The feature extraction stage is comprised of two stages: detail extraction and semantic extraction. For the detail extraction stage, we utilize the stacking of down block, Light Block, and Wide Block. On the other hand, to rapidly expand the receptive field in the semantic extraction stage, we use the stacking mode of down block, two Light Blocks, and two Wide Blocks. The various blocks are elaborated below.

1) Down Block: To address the issue of vanishing or exploding gradients that may arise in deep neural networks, we incorporate a residual connection architecture within the down block. As input and output sizes vary, both branches necessitate sampling during down-sampling. For a lightweight design, we apply point-wise convolution to condense the primary channel, followed by a 2×2 convolution with a stride of 2 for down-sampling the feature map, and then

another point-wise convolution to expand the channel count. Meanwhile, the residual channel utilizes max-pooling for down-sampling. To integrate the distinct information from both branches, we merge their sampled outputs and apply the PReLU activation function, yielding the final output of the sampling module. Fig. 3 illustrates the down block structure. Besides, Table I provides further details of the module setting, where the opr represents different operations at different stages; the Input represents input image; the Down and the Up represent down-sampling module and up-sampling module, respectively; the LFEM represents lightweight feature extraction module (including Light Block, Wide Block, and cat block); the C_{in} represents the number of input channels; the C_m represents the number of intermediate channels; the C_{out} represents the number of output channels; and the Output Size represents the resolution of output feature graph of each module.

2) Light Block and Wide Block: The feature extraction module is a vital element of a semantic segmentation neural network, significantly impacting its training convergence and dataset performance. However, standard convolution operations involve substantial computational demands. To maintain a lightweight structure while enabling the network to extract features from images with an extensive receptive field, we substitute traditional convolutions with lightweight convolution operations, such as depthwise separable convolution, point-wise convolution, and factorized convolution. This reduction in network parameters diminishes computational complexity. Although dilated convolution can expand the receptive field without increasing parameter count and computational complexity, its inferior computational efficiency results in a higher inference delay; thus, we exclude it from our network design.

In pursuit of network lightness, we devise two unique feature extraction modules with varying receptive field sizes. The first module, dubbed Light Block, consists of a depthwise separable convolution between two point-wise convolutions and employs a residual connection. This module, with a 3 $\times$ 3 receptive field, is optimized for computational efficiency. The second module, termed Wide Block, adopts factorized convolution (5 $\times$ 1 and 1 $\times$ 5) instead of the traditional 5 $\times$ 5 convolution, enabling a larger 5 $\times$ 5 receptive field. Analogous to the Light Block, the Wide Block is flanked by two point-wise convolutions and incorporates a residual connection. Fig. 3 showcases the specific architectures of these feature extraction modules.

Our proposed feature extraction module exhibits a symmetric channel structure, maintaining an equal number of input and output channels. The initial point-wise convolution reduces the channel count to 1/4 of the output channels, followed by depthwise separable convolution or factorized convolution with an equal number of input and output channels to expand the receptive field. Subsequently, the latter point-wise convolution increases the channel count to achieve the desired output channel dimension. This channel design effectively mitigates the computational complexity arising from large convolution kernels. Table I provides more detailed channel configurations.

Using an input size of 32 $\times$ 112 $\times$ 112 and an output size of 32 $\times$ 112 $\times$ 112 as an example, with the intermediate channel count set to 1/4 of the output channel count, we compute the parameter quantity and FLOPs of the feature extraction module and compare them to those of a standard convolution operation. The specific formulas for calculating the parameter quantity and FLOPs of common convolution operations are as follows (bias is not considered):

$$\text{Params}_{\text{Conv}} = K_h \times K_w \times C_i \times C_o \tag{1}$$

$$\text{FLOPs}_{\text{Conv}} = \frac{2K_h \times K_w - 1}{g} \times C_i \times F_h \times F_w \times C_o \tag{2}$$

where C_i and C_o represent the number of input and output channels for the convolution, respectively. K_h and K_w denote the height and width of the convolution kernel, while F_h and F_w represent the height and width of the feature map. k corresponds to the size of the convolution kernel, and g stands for the number of convolution groups.

For depthwise separable convolution, it can be considered as a standard convolution with the number of groups $g = K_h \times K_w$, and the number of input and output channels being C_i. Additionally, it includes the standard 1 $\times$ 1 convolution. Thus, the specific formula for calculating the parameter quantity and FLOPs is as follows (excluding bias consideration):

$$\text{Params}_{\text{dwConv}} = C_i \times (K_h \times K_w) + C_i \times C_o \tag{3}$$

$$\text{FLOPs}_{\text{dwConv}} = (2K_h \times K_w - 1) \times F_h \times F_w \times C_o + C_i \times F_h \times F_w \times C_o. \tag{4}$$

Upon calculating the above parameters, we observe that the 3 $\times$ 3 standard convolution operation contains 9.22k parameters and 115.61M FLOPs, while the Light Block only has 0.75k parameters and 9.93M FLOPs. Similarly, the 5 $\times$ 5 standard convolution operation has 25.6k parameters and 309.76M FLOPs, compared to the Wide Block, which only has 1.25k parameters and 16.26M FLOPs. Consequently, our designed feature extraction module significantly reduces the parameter count and FLOPs while retaining the same receptive field size as the standard convolution.

Considering feature extraction at multiple scales, our module is designed to accommodate receptive fields of 3 $\times$ 3 and 5 $\times$ 5. By utilizing various stacking configurations of feature extraction modules during different stages of down-sampling (detail extraction and semantic extraction), we can effectively control the receptive field size for each pixel in the feature map at different stages. This approach enables efficient extraction of both low-level details and high-level semantics according to our requirements.

Additionally, we employ several strategies to improve the performance of our module. In particular, we incorporate the residual connection approach, embed the dropout layer, and implement the PReLU (parametric rectified linear unit) function for activation before combining the input and output of the feature extraction module. The PReLU activation function

TABLE I
NUMBER OF CHANNELS IN EACH STAGE OF A-NET

Stage	Downsampling				Upsampling				Output Size
	opr	C_{in}	C_m	C_{out}	*opr*	C_{in}	C_m	C_{out}	
S_0	Input			3	Seg Head	32	16	1	224×224
S_1	Down	3	8	32	LFEM	32	8	32	112×112
	LFEM	32	8	32	Up	64	8	32	112×112
S_2	Down	32	16	64	LFEM	64	16	64	64×64
	LFEM	64	16	64	Up	128	16	64	64×64
S_3	Down	64	32	128	LFEM	128	32	128	32×32
	LFEM	128	32	128	Up	128	32	128	32×32
S_4	Down	128	32	128	LFEM	128	32	128	16×16
	LFEM	128	32	128	Up	128	32	128	16×16
S_5	Down	128	32	128					8×8
	LFEM	128	32	128					8×8

is expressed as follows:

$$\mathrm{PReLU}(x_i) = \begin{cases} x_i, & \text{if } x_i > 0 \\ a_i x_i, & \text{if } x_i \le 0 \end{cases} \tag{5}$$

where a is the parameter obtained through training.

The residual connection effectively tackles the issues of gradient explosion or vanishing gradients that can occur in deep networks, facilitating efficient convergence of the network on small datasets. Incorporating the dropout layer within the feature extraction module also helps prevent overfitting on small datasets. Moreover, the PReLU activation function introduces increased flexibility to the network without substantially augmenting the parameter count or computational overhead, thus further optimizing the performance of the feature extraction module.

As mentioned earlier, our two lightweight feature extraction modules are capable of effectively extracting features from images at different stages of the network using specific combinations.

D. Feature Fusion

During the feature fusion stage, we utilize a stacking configuration consisting of up blocks, Light Blocks (or cat blocks), and Wide Blocks in an interleaved manner to accomplish fine feature recovery and feature fusion. In the final up-sampling step, we develop a simple seg head to map the up-sampled features to segmentation output.

1) Up Block: Two prevalent methods for up-sampling are interpolation up-sampling and deconvolution. To address the issues of vanishing or exploding gradients in deep networks, we adopt the residual connection structure in our up block, as detailed in Section III-C1. This approach involves creating two branches using deconvolution and bilinear up-sampling operations, and implementing channel compression through point-wise convolution before deconvolution. By expanding the number of channels after deconvolution, a lightweight sampling module is constructed. The outputs of the two branches are then summed and activated by PReLU. During the up-sampling process, the number of output channels gradually decreases, with the number of intermediate channels set at 1/4 of the output channel count. Table I presents the channel settings.

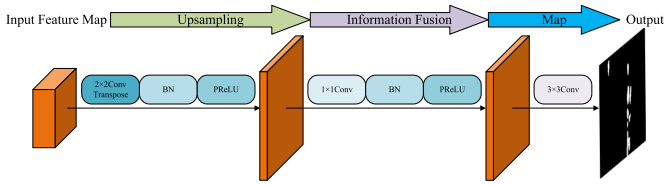

Fig. 4. Seg head architecture.

Moreover, in the up-sampling process, we merge low-level details with high-level semantics after up-sampling through a jump connection at 1/4 size of input, as it is the boundary between the detail extraction stage and the semantic extraction stage. First, we concatenate the feature map obtained from up-sampling high-level semantics with the details extracted during the detail extraction stage. Subsequently, we utilize lightweight depthwise separable convolution to compress the channel count and integrate the spatial information across different channels. This combined feature map is then input into the Light Block for further feature fusion and extraction operations. Fig. 3 illustrates the specific architecture of this process.

2) Seg Head: In the final up-sampling stage of our network, we have designed a straightforward segmentation head. This segmentation head consists of a deconvolution layer, a point-wise convolution layer, and a 3 × 3 standard convolution layer, as depicted in Fig. 4. The deconvolution layer is responsible for up-sampling the feature map, initially half the size of the input image, while simultaneously reducing the number of channels. The point-wise convolution layer serves to integrate spatial information from various channels of the up-sampled feature map. Lastly, the 3 × 3 standard convolution layer maps the feature map into the desired segmentation output, thereby completing the entire network computation process.

E. Loss Function and Training

To further enhance network performance, the loss function formula used in the training process is as follows:

$$L_{\text{weight}}(p_d, g_d) = L_{\text{dice}}(p_d, g_d) + 0.5 \times L_{\text{bce}}(p_d, g_d) \tag{6}$$

where $p_d \in \mathbb{R}^{H\times W}$ denotes the predicted pixel and $g_d \in \mathbb{R}^{H\times W}$ denotes the corresponding pixel of ground-truth. Additionally, L_{bce} represents the binary cross-entropy loss, while L_{dice} represents the dice loss, which is given as follows:

$$L_{\text{dice}}(p_d, g_d) = 1 - \frac{2\sum_i^{H\times W} p_d^i g_d^i + \varepsilon}{\sum_i^{H\times W} \left(p_d^i\right)^2 + \sum_i^{H\times W} \left(g_d^i\right)^2 + \varepsilon}. \tag{7}$$

Moreover, we have not employed a complex training method to train A-Net. Instead, we have utilized a simple gradient descent method to train A-Net without incorporating any auxiliary training strategies.

IV. EXPERIMENTS

In this section, we begin by introducing the industrial surface defect dataset, our experimental setup, and the evaluation

metrics employed. Next, we carry out ablative experiments to examine the impact of our designed components on network performance. We then perform a comparative analysis of the performance and network structure lightness of our proposed method relative to other state-of-the-art algorithms on different datasets. Lastly, we assess the computational efficiency of our proposed lightweight networks on the CPU platform, followed by a comprehensive analysis and comparison of the results.

A. Datasets, Settings, and Evaluation Metrics

1) Datasets: In this article, we have selected two distinct surface defect datasets, namely the NEU-DET defect dataset and the DAGM defect dataset, to substantiate and evaluate the applicability and generality of our proposed method.

a) NEU-seg dataset: The NEU dataset is a standard dataset collected by [33] to address the problem of automatic recognition for hot-rolled steel strips. The dataset includes six types of strip steel plates, comprising patch, crazing, pitted-surface, inclusion, scratches, and rolled-in scale, with each surface defect containing 300 images. The original resolution of the images in the dataset is 200 × 200, and all have corresponding defect type labels. We selected three surface defects (inclusion, patches, and scratches) for pixel-level marking. We then adjusted their resolution to 224 × 224 and divided them into training set and test set, containing 250 and 50 images, respectively, to enable their application to our industrial defect image segmentation.

b) DAGM-seg dataset: The DAGM dataset [34] is manually generated and contains multiple types of industrial surface defect images with an original resolution of 512 × 512. We chose categories 7 through 10, encompassing a total of 4 datasets, and then divided them into training set and test set, containing 250 and 50 images, respectively.

c) MCSD dataset: The main challenge in the MCSD dataset is the complex and changing background. The resolution of images in this dataset is 512 × 512, and the training set and test set contain 886 and 222 images, respectively.

d) MT dataset: The MT dataset mainly verifies the detection effect of the network for different defects under the conditions of uneven illumination, complex background, and large shape differences. The resolution of images is 512 × 512, and the training set and test set contain 341 and 51 images, respectively.

2) Setting:

a) Training: To ensure fairness, all models are trained from scratch. We employ the stochastic gradient descent (SGD) algorithm with a learning rate of 0.0003 and a momentum of 0.9 to train all models. For the NEU-seg datasets, we adopt a batch size of 16, while for the DAGM-seg, MCSD, and MT datasets, we use a batch size of 4. The weight decay is set at 0.0001. Moreover, we divide 15% of the training set into validation and train all networks for 2000 epochs during training stage.

b) Data augmentation: Images are randomly rotated by 90° and randomly flipped during training to expand the training set and prevent severe overfitting.

c) Evaluation: When testing network performance, we employ the simplest and fastest method, which involves directly loading the test data to assess the performance of every model after training.

d) Setup: We conduct experiments using PyTorch 1.9.0, and all models are evaluated on a single NVIDIA GeForce GTX 1080Ti with CUDA 11.7, CUDNN 8.5, and TensorRT 8.5.3.

3) Evaluation Metrics: In order to evaluate the model performance and complexity more comprehensively, we use the IoU index of segmentation results to assess the model performance and the number of model parameters and FLOPs to evaluate model complexity and computational consumption. The IoU is represented as a percentage, with higher IoU values indicating better model performance. The calculation formula is as follows:

$$\mathrm{IoU} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN} + \mathrm{FP}}. \tag{8}$$

True positives (TP) refer to positive predictions that match the ground-truth. False negatives (FN) represent negative predictions that do not match the ground-truth. False positives (FP) denote positive predictions that do not match the ground-truth.

Additionally, the number of model parameters is the sum of the number of parameters for all operations in the model, and its unit is typically expressed in megaParams (M). The calculation formula for the number of parameters of a single convolution operation is shown in (1). The fewer the number of model parameters, the lower the model complexity. The model FLOPs parameter is the sum of FLOPs of all operations in the model, with the unit generally being gigaFLOPS (G). The calculation formula for FLOPs of a single convolution operation is shown in (2). The lower the FLOPs of the model, the lower the computational consumption. Therefore, a lightweight model requires that the number of network parameters and FLOPs be maintained at a low level.

B. Ablative Experiments

In this section, a comprehensive analysis of the lightweight nature and feature extraction capability of the proposed Light Block and Wide Block architectures is conducted by replacing them with 3 × 3 convolution and 5 × 5 convolution layers. Subsequently, the jump connection aggregation structure and the final split header structure are incorporated into the network architecture in a step-by-step manner. By systematically examining the network's performance with varying degrees of ablation and conducting a thorough evaluation of the number of network parameters and FLOPs, the efficacy and lightweight advantages of the proposed components are effectively demonstrated. The outcomes of the ablation experiments are presented in Table II, where the number under inclusion, patches, and scratches represents IoU (%) of models on corresponding dataset.

The results presented in Table II demonstrate that the proposed A-Net backbone yields commendable segmentation performance and maintains a low parameter count and FLOPs simultaneously, even in the absence of specifically designed lightweight feature extraction modules, jump connections, and segmentation headers. Upon incorporating the proposed

TABLE II
ABLATIVE EXPERIMENTS ON THE NEU-SEG DATASET

Light Block	Wide Block	Jump Connection	Seg Head	Inclusion	Patches	Scratches	Parameters	FLOPs
				47.13	75.70	55.44	4.41M	3.12G
√				46.43	76.08	56.10	3.36M	2.39G
	√			52.71	76.08	55.79	1.43M	1.02G
√	√			51.28	77.65	54.94	0.38M	0.28G
√	√	√		56.77	77.04	59.30	0.39M	0.31G
√	√	√	√	**60.53**	**78.76**	**59.51**	0.39M	0.44G

lightweight feature extraction structure, the model's parameter count and FLOPs are reduced by more than 90%, compared to the ordinary convolutional model, with a slight increase in performance. This outcome validates the effectiveness of the lightweight feature extraction module proposed in this study, which uses special convolutions, dropout, and residual connection rationally to adapt the network to different industrial surface defect detection datasets. Upon adopting the jump connection structure, the model's performance is significantly improved while only adding a few parameters and FLOPs.

Following the integration of the seg head structure proposed in this study, the network's performance on the NEU-inclusion dataset is notably enhanced, while a slight performance improvement is observed on other datasets. This observation substantiates the efficacy of the seg head structure proposed in enhancing the model's generalization ability across various datasets. Finally, from the perspective of model lightness, the A-Net model structure's parameter count determined in this study is only 0.39M, and FLOPs are only 0.44G, thereby satisfying the deployment requirements of edge devices (FLOPs lower than 0.6G).

C. Comparative Experiments

We ended up choosing ten classical segmentation networks [FCN [3], SegNet [4], PSPNet [7], DeeplabV3+ [35], RefineNet [17], U-Net [5], Swin-Unet [24], CCNet [36], and two networks designed for industrial image segmentation (PGA-Net [6], LSA-Net [37])], and seven light networks (BiSeNet [9], BiSeNetV2 [10], STDC [38], ERFNet [16], ESNet [25], ENet [25], and topformer [27]) that performs well in natural images as the baseline network to compare with our network.

1) NEU-Seg Dataset: Table III presents the performance of each baseline network and the A-Net proposed in this paper on the NEU-seg dataset. The number under inclusion, patches, and scratches represents IoU (%) of models on corresponding dataset, while "-" represents that the model cannot converge effectively on the corresponding dataset. Besides, the wave line under the number shows that the corresponding model ranks second in this category, while the underline indicates ranking third.

The analysis of various segmentation network performances in the table reveals that larger models generally achieve higher IoU scores than smaller models. In comparison with larger models, the A-Net proposed in this paper achieves the highest IoU on the NEU-inclusion dataset and is only 1.15% away from the highest IoU on the NEU-patches dataset. Besides, the performance of A-Net on the scratches dataset ranks third among all methods in the table. However, the Swin-Unet is unable to effectively converge, because of the small datasets. These results demonstrate that the A-Net proposed in this paper exhibits excellent performance on the industrial surface defect dataset.

TABLE III
PERFORMANCE OF DIFFERENT METHODS AND OUR METHOD ON THE NEU-SEG DATASET

Methods	Inclusion	Patches	Scratches	Parameters	FLOPs
Classical					
FCN	52.55	78.78	55.39	45.47M	16.00G
SegNet	56.61	79.84	58.05	29.44M	30.73G
PSPNet	54.06	79.80	57.78	53.32M	38.71G
DeepLabV3+	57.49	78.34	55.07	59.34M	16.97G
RefineNet	58.72	**79.91**	60.02	80.22M	161.30G
U-Net	58.97	79.55	**60.92**	31.39M	42.75G
Swin-Unet	-	-	-	27.17M	5.88G
CCNet	42.79	75.18	50.27	67.69M	14.88G
PGA-Net	31.12	66.89	24.91	51.41M	315.69G
LSA-Net	58.94	79.68	59.20	21.60M	47.16G
Lightweight					
BiSeNet	54.70	79.13	56.71	12.40M	4.14G
BiSeNetV2	11.23	57.83	20.16	4.95M	1.91G
STDC	50.43	76.84	54.74	12.04M	2.97G
ERFNet	58.80	77.77	33.61	2.08M	2.82G
ESNet	59.73	78.84	58.71	1.66M	2.58G
ENet	58.21	78.57	59.43	**0.35M**	0.37G
TopFormer	54.56	76.13	53.24	3.00M	**0.24G**
A-Net	**60.53**	78.76	59.51	0.39M	0.44G

In terms of network lightweightness, the A-Net proposed in this paper achieves a remarkable advantage over large models concerning the number of parameters and FLOPs. Specifically, A-Net's parameter quantity is only 1.32% of SegNet, the network with the minimum parameters among the large models, and its FLOPs are only 2.75% of the FLOPs of FCN, the network with the lowest FLOPs among the large networks. Compared to small models, A-Net's number of parameters and FLOPs are only slightly higher than those of ENet and lower than other small models. Furthermore, it is evident that the segmentation performance of A-Net surpasses that of other small models. The A-Net architecture successfully achieves the design goal of a lightweight network structure, thereby attaining the best precision-lightweightness balance on the NEU-seg dataset.

Fig. 5 displays the visual segmentation outputs of each comparative network on the NEU-seg dataset. The results demonstrate that A-Net not only accomplishes efficient defect segmentation but also exhibits noteworthy proficiency in detecting defects of diverse scales. Furthermore, A-Net man-

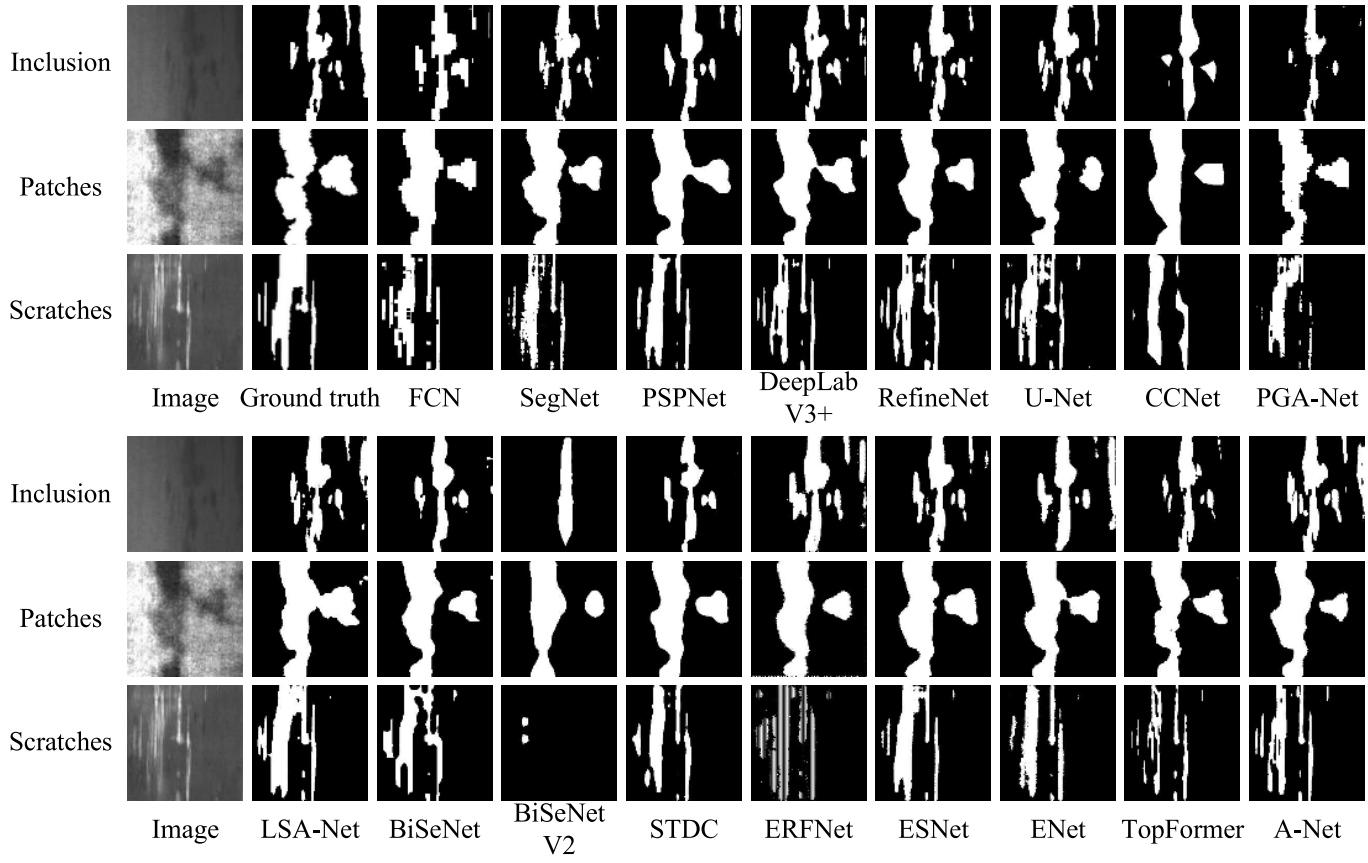

Fig. 5. Visual display of the results of every network on the NEU-Seg Dataset.

ifests impressive boundary segmentation capabilities. These accomplishments can primarily be attributed to the network backbone and the lightweight feature extraction module devised by the authors. This module comprises phased feature extraction stages and feature fusion stages, which enables the network to effectively extract and resolve features of varying scales.

2) DAGM-Seg Dataset: Table IV presents the performance of each baseline network and the A-Net model proposed in this study, on the DAGM-seg dataset. In this table, the number under Class7, Class8, Class9, and Class10 represents IoU (%) of models on corresponding dataset, while "-" represents that the model cannot converge effectively on the corresponding dataset. In addition, the wave line under the number indicates that the corresponding model ranks second in this category, while the underline indicates ranking third.

The convergence performance of the comparison networks indicates that A-Net exhibits commendable convergence performance similar to large models and can effectively converge on the DAGM-seg dataset even with a limited number of images (i.e., 250 images). In contrast, most of the small comparative models and the Swin-Unet are unable to converge effectively on DAGM-class8, DAGM-class9, and DAGM-class10. Hence, the A-Net model surpasses its smaller counterparts by demonstrating superior convergence capabilities for datasets of smaller magnitudes.

Analysis of the performance of various networks, as presented in the table, reveals that larger models generally achieve higher IoU values than smaller models. However, among the models compared, the A-Net proposed in this study outperforms all others by achieving the highest IoU on the DAGM-class7 dataset. Additionally, A-Net's performance on the DAGM-class8 ranks second among all models in the table, trailing only the U-Net in the classical model category. Besides, A-Net's performance on the DAGM-class8 ranks third among all models in the table, trailing only the U-Net and LSA-Net in the classical model category. Finally, the IoU attained by A-Net on the DAGM-class9 dataset is only 0.34% lower than the highest IoU recorded. These results attest to the exceptional performance of the A-Net model on industrial surface defect datasets.

Fig. 6 displays the visual segmentation outputs of each comparative network on the DAGM-seg dataset, except for networks that cannot converge effectively. It is obvious that the A-Net also has powerful ability on segmenting small objects.

Considering the extremely low parameter quantity and FLOPs of A-Net, the proposed A-Net segmentation network achieves the best precision-lightweight trade-off on the DAGM-seg dataset. This demonstrates the effectiveness of the A-Net model in addressing the challenges posed by industrial surface defect segmentation tasks while maintaining a lightweight architecture suitable for deployment on edge devices.

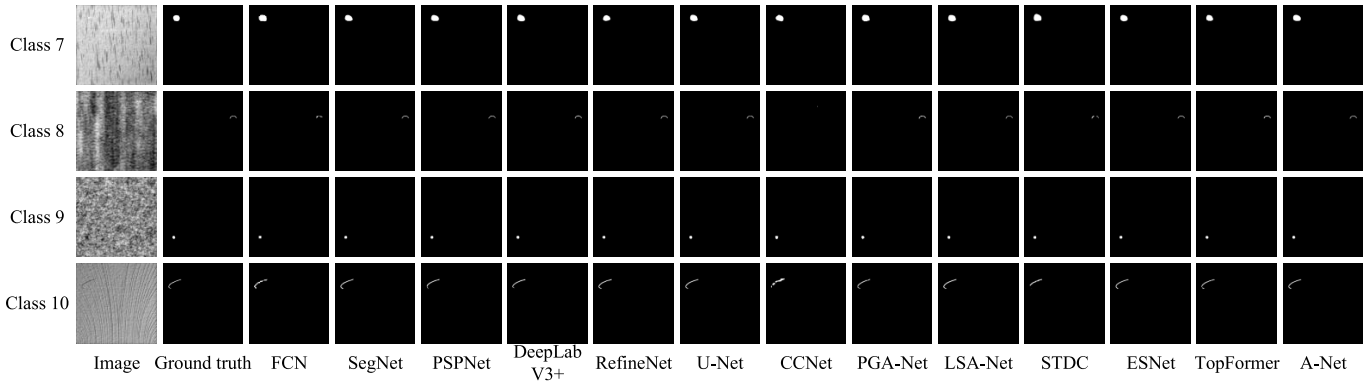

Fig. 6. Visual display of the results of every network on the DAGM-seg Dataset.

TABLE IV
PERFORMANCE OF DIFFERENT METHODS AND OUR METHOD ON THE DAGM-SEG DATASET

Methods	Class7	Class8	Class9	Class10	Parameters	FLOPs
Classical						
FCN	79.16	43.54	74.89	52.70	45.47M	83.61G
SegNet	80.62	69.40	86.75	73.04	29.44M	160.52G
PSPNet	81.21	71.00	86.99	72.34	53.32M	201.35G
DeepLabV3+	81.09	70.73	88.22	73.89	59.34M	88.63G
RefineNet	81.06	70.70	87.93	73.73	80.22M	842.74G
U-Net	82.74	**76.16**	**88.37**	**77.79**	31.39M	223.34G
Swin-Unet	-	-	-	-	27.17M	26.41G
CCNet	80.75	-	74.68	36.48	67.69M	77.74G
PGA-Net	81.07	61.76	82.55	65.65	51.41M	1649.33G
LSA-Net	81.73	75.36	88.00	77.10	21.60M	246.40G
Lightweight						
BiSeNet	80.64	-	-	-	12.40M	21.62G
BiSeNetV2	62.82	-	-	-	4.95M	9.96G
STDC	78.93	48.84	84.68	59.15	12.04M	15.54G
ERFNet	49.87	-	-	-	2.08M	14.74G
ESNet	82.02	74.51	88.12	75.36	1.66M	13.48G
ENet	79.47	-	-	-	**0.35M**	1.94G
TopFormer	80.73	68.37	85.76	72.17	3.00M	**1.22G**
A-Net	**82.86**	75.99	88.03	77.01	0.39M	2.30G

TABLE V
PERFORMANCE OF DIFFERENT METHODS AND OUR METHOD ON THE MCSD DATASET

Methods	MCSD	Parameters	FLOPs
Classical			
FCN	69.51	45.47M	83.61G
SegNet	74.36	29.44M	160.52G
PSPNet	72.77	53.32M	201.35G
DeepLabV3+	76.00	59.34M	88.63G
RefineNet	72.88	80.22M	842.74G
U-Net	73.98	31.39M	223.34G
Swin-Unet	45.50	27.17M	26.41G
CCNet	74.53	67.69M	77.74G
PGA-Net	77.23	51.41M	1649.33G
LSA-Net	77.69	21.60M	246.40G
Lightweight			
BiSeNet	68.33	12.40M	21.62G
BiSeNetV2	72.45	4.95M	9.96G
STDC	68.30	12.04M	15.54G
ERFNet	-	2.08M	14.74G
ESNet	71.99	1.66M	13.48G
ENet	75.54	**0.35M**	1.94G
TopFormer	67.27	3.00M	**1.22G**
A-Net	**78.68**	0.39M	2.30G

3) MCSD Dataset: Table V presents the performance of each baseline network, as well as the A-Net model proposed in this study, on the MCSD dataset. In this table, the number under MCSD represents IoU (%) of models on corresponding dataset, while "-" represents that the model cannot converge effectively on the corresponding dataset. Moreover, the wave line under the number indicates that the corresponding model ranks second in this category, while the underline indicates ranking third.

From Table V, compared to the competitive classical network and the network designed for industrial surface defect detection, our model achieves the best performance with extremely low parameter quantity. Besides, compared to the lightweight network model, our model performance is much higher and the parameter quantity is also relatively low. These results demonstrate the effectiveness of the A-Net model in addressing the challenges posed by industrial surface defect segmentation tasks.

Fig. 7 displays the visual segmentation outputs of each comparative network on the MCSD dataset, except for networks that cannot converge effectively. From Fig. 7, our method shows the best segmentation performance and the highest recognition ability.

4) MT Dataset: Table VI presents the performance of each baseline network, as well as the A-Net model proposed in this study, on the MT dataset. The number under MT represents IoU (%) of models on corresponding dataset, while "-" represents that the model cannot converge effectively on the corresponding dataset. Furthermore, the wave line under the

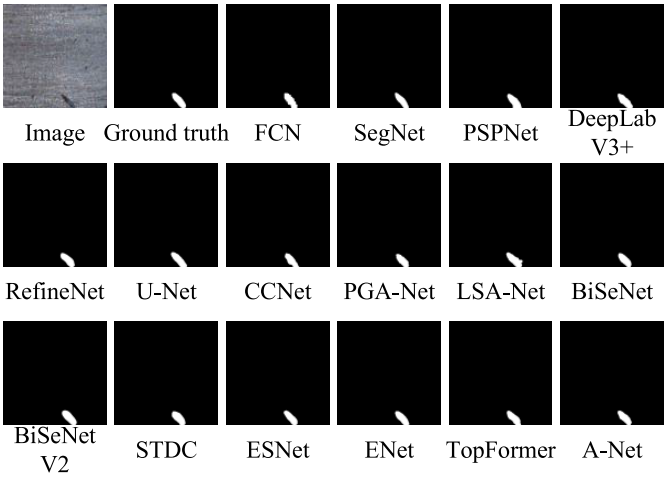

Fig. 7. Visual display of the results of every network on the MCSD Dataset.

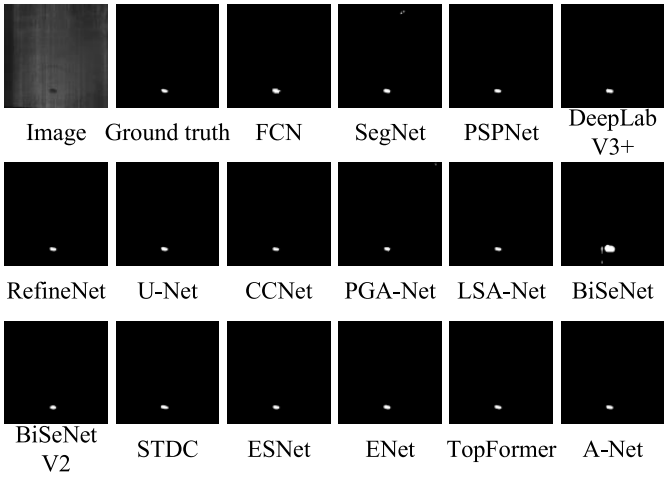

Fig. 8. Visual display of the results of every network on the MT Dataset.

TABLE VI
PERFORMANCE OF DIFFERENT METHODS AND OUR METHOD ON THE MT DATASET

Methods	MT	Parameters	FLOPs
Classical			
FCN	47.66	45.47M	83.61G
SegNet	65.45	29.44M	160.52G
PSPNet	69.45	53.32M	201.35G
DeepLabV3+	62.20	59.34M	88.63G
RefineNet	62.59	80.22M	842.74G
U-Net	63.24	31.39M	223.34G
Swin-Unet	-	27.17M	26.41G
CCNet	62.61	67.69M	77.74G
PGA-Net	61.70	51.41M	1649.33G
LSA-Net	68.46	21.60M	246.40G
Lightweight			
BiSeNet	44.56	12.40M	21.62G
BiSeNetV2	48.08	4.95M	9.96G
STDC	57.73	12.04M	15.54G
ERFNet	-	2.08M	14.74G
ESNet	**69.54**	1.66M	13.48G
ENet	65.05	**0.35M**	1.94G
TopFormer	64.94	3.00M	**1.22G**
A-Net	69.18	0.39M	2.30G

number indicates ranking second and the underline indicates ranking third.

From the table, it can be seen that on the MT dataset, our lightweight model A-Net can compete with the best performing model, and has a gap of only 0.36% compared to the highest performing IoU, which shows that the A-Net performs better than many classical models in the table. Considering the extremely low parameter quantity and FLOPs of A-Net, the proposed A-Net segmentation network achieves the best precision-lightweight balance on the MT dataset.

Fig. 8 displays the visual segmentation outputs of each comparative network on the MT dataset, except for networks that cannot converge effectively. From Fig. 8, it can be seen

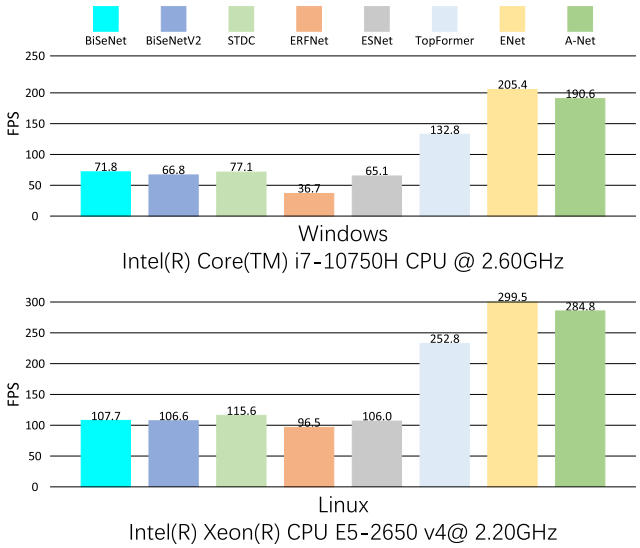

Fig. 9. Results of inference speed test on edge devices.

that A-Net has excellent segmentation ability for small defects, which is sufficient to compete with classical network models and networks designed for industrial surface defect detection.

D. Inference Speed Test on CPU

To better simulate model deployment at the industrial edge and explore the inference speed of the model without GPU acceleration, we use the Benchmark Python Tool in OpenVINO [39] to test the inference speed of BiSeNet, BiSeNetV2, STDC, ERFNet, ESNet, TopFormer, ENet, and A-Net on two edge devices, which are CPU-based platforms (Intel(R) Core(TM) i7-10750H CPU @ 2.60 GHz in windows and Intel Xeon(R) CPU E5-2650 v4 @ 2.20 GHz in Ubuntu 18.04). The test is set as follows: The input image size is $3 \times 224 \times 224$, the batch size is 1, and the test epoch number is 5000.

The results obtained from the test, as shown in Fig. 9, demonstrate that the A-Net model proposed in this study outperforms other models in terms of inference speed on both Windows and Linux systems. The slower inference speed on

Windows systems can be attributed to the greater number of irrelevant processes competing for system resources.

Despite this, the proposed A-Net architecture achieves inference speeds that are several times faster than those of real-time or lightweight semantic segmentation networks, such as BiSeNet, BiSeNetV2, ERFNet, and ESNet, when running on a CPU. Additionally, A-Net approaches the inference speeds of the lightest network, ENet, on CPU, thereby establishing its superiority over competing models.

These results confirm the effectiveness of the A-Net model for deployment on industrial edge devices, where high-speed inference and lightweight architecture are crucial for real-time processing and analysis of industrial surface defects. By outperforming other state-of-the-art models, A-Net proves to be a suitable solution for addressing the challenges associated with industrial edge computing.

E. Experiment Conclusion

Based on the performance and inference speed tests conducted in previous sections, along with the analyses of parameter numbers and FLOPs, and the inference frames per second (FPS) tests performed on CPU, it is evident that the A-Net network structure proposed in this study demonstrates competitive performance on various industrial surface defect segmentation datasets when compared to classical semantic segmentation network models.

In addition, A-Net boasts an impressively low parameter count and FLOPs, while also achieving high inference speeds on CPU platforms. These attributes contribute to the lightweight nature and computational efficiency of the A-Net model, making it particularly well-suited for deployment on edge devices in industrial settings.

In conclusion, the A-Net network structure achieves an optimal balance between precision and speed compared to the other networks examined in this study. This balance makes it a promising solution for real-time detection and analysis of industrial surface defects, thereby addressing the challenges associated with industrial edge computing.

V. Conclusion

In this article, we have presented A-Net, a lightweight and real-time network for industrial surface defect segmentation, specifically designed to address the challenges arising from limited data, varying defect sizes, irregular outlines, and subtle differences between defect and normal areas. The proposed A-shaped network structure consists of two main components, feature extraction and feature fusion, efficiently extracting low-level detail and high-level semantic information while facilitating the aggregation of information at different levels.

Through the design of lightweight convolution blocks, we have managed to prevent overfitting, gradient disappearance, and gradient explosion, making the network suitable for small datasets. Moreover, A-Net demonstrates competitive performance compared to classic large models, such as U-Net, while significantly reducing the number of parameters and computational costs and shows high inference speed without GPU acceleration.

However, in these comparative experiments, though the A-Net can achieve performance that competes with other networks, it cannot achieve the highest performance among all the models on all defect categories, which is the flaw of our method. In the future, we are going to further improve the performance of lightweight neural network designed for the surface defect detection field.

Our work contributes to the ongoing development of effective and efficient defect segmentation networks, paving the way for real-world industrial applications with limited resources. Future research directions include further optimization of the network architecture, exploring additional lightweight approaches, and investigating the applicability of A-Net to other domains and tasks that require low-latency and computationally efficient models.

References

[1] W. Wang et al., "A real-time steel surface defect detection approach with high accuracy," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–10, 2022.

[2] H. Chen, Y. Du, Y. Fu, J. Zhu, and H. Zeng, "DCAM-Net: A rapid detection network for strip steel surface defects based on deformable convolution and attention mechanism," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–12, 2023.

[3] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[4] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder–decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[5] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. 18th Int. Conf. Med. Image Comput. Comput.-Assist. Intervent. (MICCAI)*, Munich, Germany. Cham, Switzerland: Springer, 2015, pp. 234–241.

[6] H. Dong, K. Song, Y. He, J. Xu, Y. Yan, and Q. Meng, "PGA-Net: Pyramid feature fusion and global context attention network for automated surface defect detection," *IEEE Trans. Ind. Informat.*, vol. 16, no. 12, pp. 7448–7458, Dec. 2020.

[7] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6230–6239.

[8] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

[9] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Bisenet: Bilateral segmentation network for real-time semantic segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 325–341.

[10] C. Yu, C. Gao, J. Wang, G. Yu, C. Shen, and N. Sang, "BiSeNet V2: Bilateral network with guided aggregation for real-time semantic segmentation," *Int. J. Comput. Vis.*, vol. 129, no. 11, pp. 3051–3068, Nov. 2021.

[11] X. Lei et al., "STDC-MA network for semantic segmentation," *IET Image Process.*, vol. 16, no. 14, pp. 3758–3767, Dec. 2022.

[12] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "ENet: A deep neural network architecture for real-time semantic segmentation," 2016, *arXiv:1606.02147*.

[13] H. Zhao, X. Qi, X. Shen, J. Shi, and J. Jia, "Icnet for real-time semantic segmentation on high-resolution images," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 405–420.

[14] X. Xu et al., "Quantization of fully convolutional networks for accurate biomedical image segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8300–8308.

[15] Y. Liu, K. Chen, C. Liu, Z. Qin, Z. Luo, and J. Wang, "Structured knowledge distillation for semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2604–2613.

[16] E. Romera, J. M. Álvarez, L. M. Bergasa, and R. Arroyo, "ERFNet: Efficient residual factorized ConvNet for real-time semantic segmentation," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 1, pp. 263–272, Jan. 2018.

[17] G. Lin, A. Milan, C. Shen, and I. Reid, "RefineNet: Multi-path refinement networks for high-resolution semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1925–1934.

[18] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Learning a discriminative feature network for semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1857–1866.

[19] J. Wang et al., "Deep high-resolution representation learning for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3349–3364, Oct. 2021.

[20] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5999–6009.

[21] A. Dosovitskiy et al., "An image is worth 16 × 16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[22] Z. Liu et al., "Swin Transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9992–10002.

[23] S. Zheng et al., "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 6877–6886.

[24] H. Cao et al., "Swin-Unet: Unet-like pure transformer for medical image segmentation," in *Proc. Eur. Conf. Comput. Vis. Workshops (ECCV)*, Tel Aviv, Israel. Cham, Switzerland: Springer, 2023, pp. 205–218.

[25] Y. Wang, Q. Zhou, J. Xiong, X. Wu, and X. Jin, "ESNet: An efficient symmetric network for real-time semantic segmentation," in *Proc. 2nd Chin. Conf. Pattern Recognit. Comput. Vis.*, Xi'an, China. Cham, Switzerland: Springer, 2019, pp. 41–52.

[26] H. Li, P. Xiong, H. Fan, and J. Sun, "DFANet: Deep feature aggregation for real-time semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9514–9523.

[27] W. Zhang et al., "TopFormer: Token Pyramid transformer for mobile semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 12083–12093.

[28] R. Wang, Q. Guo, S. Lu, and C. Zhang, "Tire defect detection using fully convolutional network," *IEEE Access*, vol. 7, pp. 43502–43510, 2019.

[29] Z. Yu, X. Wu, and X. Gu, "Fully convolutional networks for surface defect inspection in industrial environment," in *Proc. Int. Conf. Comput. Vis. Syst.*, Shenzhen, China. Cham, Switzerland: Springer, Jul. 2017, pp. 417–426.

[30] Y. Huang, C. Qiu, and K. Yuan, "Surface defect saliency of magnetic tile," *Vis. Comput.*, vol. 36, no. 1, pp. 85–96, Jan. 2020.

[31] Y. Dong et al., "A deep-learning-based multiple defect detection method for tunnel lining damages," *IEEE Access*, vol. 7, pp. 182643–182657, 2019.

[32] Q. Zou, Z. Zhang, Q. Li, X. Qi, Q. Wang, and S. Wang, "DeepCrack: Learning hierarchical convolutional features for crack detection," *IEEE Trans. Image Process.*, vol. 28, no. 3, pp. 1498–1512, Mar. 2019.

[33] K. Song and Y. Yan, "A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects," *Appl. Surf. Sci.*, vol. 285, no. 21, pp. 858–864, Nov. 2013.

[34] M. Wieler and T. Hahn, "Weakly supervised learning for industrial optical inspection," in *Proc. DAGM Symp.*, 2007. [Online]. Available: https://conferences.mpi-inf.mpg.de/dagm/2007/index.html

[35] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder–decoder with atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 801–818.

[36] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, "CCNet: Criss-cross attention for semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 603–612.

[37] W. Li, B. Li, S. Niu, Z. Wang, M. Wang, and T. Niu, "LSA-Net: Location and shape attention network for automatic surface defect segmentation," *J. Manuf. Processes*, vol. 99, pp. 65–77, Aug. 2023.

[38] M. Fan et al., "Rethinking BiSeNet for real-time semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 9716–9725.

[39] A. Demidovskij et al., "OpenVINO deep learning workbench: Comprehensive analysis and tuning of neural networks inference," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 783–787.

Biao Chen is currently pursuing the B.Eng. degree with the School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan, China.

His research interests include intelligent manufacturing, surface defect detection, and deep learning.

Tongzhi Niu received the B.Eng. degree from the Wuhan University of Technology, Wuhan, China, in 2018. He is currently pursuing the Ph.D. degree with the School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan.

From 2023 to 2024, he participated in a joint training program at the National University of Singapore, Singapore. His research interests include intelligent manufacturing, surface defect detection, and deep learning.

Wenyong Yu (Senior Member, IEEE) received the M.S. and Ph.D. degrees from the Huazhong University of Science and Technology, Wuhan, China, in 1999 and 2004, respectively.

He is currently an Associate Professor with the School of Mechanical Science and Engineering, Huazhong University of Science and Technology. His research interests include machine vision, intelligent control, and image processing.

Ruoqi Zhang received the B.Eng. degree from the Qingdao University of Science and Technology, Qingdao, China, in 2021. She is currently pursuing the master's degree with the Huazhong University of Science and Technology, Wuhan, China.

Her research interests include power prediction of distributed photovoltaics in different scenarios.

Zhenrong Wang received the master's degree in control engineering from the Harbin Institute of Technology, Harbin, China, in 2020. She is currently pursuing the Ph.D. degree with the Huazhong University of Science and Technology, Wuhan, China.

Her research interests include intelligent manufacturing, defect detection, image processing, and neural architecture search.

Bin Li received the B.S., M.S., and Ph.D. degrees from the Huazhong University of Science and Technology, Wuhan, China, in 1982, 1989, and 2006, respectively, all in mechanical engineering.

He is currently a Professor with the School of Mechanical Science and Engineering, Huazhong University of Science and Technology. His current research interests include intelligent manufacturing and computer numerical control machine tools.