applied sciences

MDPI

Article

A Lightweight Residual Model for Corrosion Segmentation with Local Contextual Information

Jingxu Huang, Qiong Liu *, Lang Xiang, Guangrui Li, Yiqing Zhang and Wenbai Chen

School of Automation, Beijing Information Science and Technology University (BISTU), Beijing 100192, China
* Correspondence: liuqionglq@126.com

Abstract: Metal corrosion in high-risk areas, such as high-altitude cables and chemical factories, is very complex and inaccessible to people, which can be a hazard and compromise people's safety. Embedding deep learning models into edge computing devices is urgently needed to conduct corrosion inspections. However, the parameters of current state-of-the-art models are too large to meet the computation and storage requirements of mobile devices, while lightweight models perform poorly in complex corrosion environments. To address these issues, a lightweight residual deep-learning model based on an encoder–decoder structure is proposed in this paper. We designed small and large kernels to extract local detailed information and capture distant dependencies at all stages of the encoder. A sequential operation consisting of a channel split, depthwise separable convolution, and channel shuffling were implemented to reduce the size of the model. We proposed a simple, efficient decoder structure by fusing multi-scale features to augment feature representation. In extensive experiments, our proposed model, with only 2.41 MB of parameters, demonstrated superior performance over state-of-the-art segmentation methods: 75.64% mean intersection over union (IoU), 86.07% mean pixel accuracy and a 0.838 F1-score. Moreover, a larger version was designed by increasing the number of output channels, and model accuracy improved further: 79.06% mean IoU, 88.07% mean pixel accuracy, and 0.891 F1-score. The size of the model remained competitive at 8.25 MB. Comparison work with other networks and visualized results were used for validation and to determine the accuracy of metal corrosion surface segmentation with limited resources.

Keywords: corrosion segmentation; lightweight residual model; large convolution kernels; contextual feature

Citation: Huang, J.; Liu, Q.; Xiang, L.; Li, G.; Zhang, Y.; Chen, W. A Lightweight Residual Model for Corrosion Segmentation with Local Contextual Information. *Appl. Sci.* **2022**, *12*, 9095. https://doi.org/10.3390/app12189095

Academic Editors: Jiaqi Li, Božidar Šarler, Haiping Liu and Jian Zhang

Received: 17 August 2022
Accepted: 7 September 2022
Published: 9 September 2022

1. Introduction

Research has shown that losses due to metal corrosion account for approximately 3.4% of the world's annual Gross Domestic Product (GDP) [1], which causes huge economic losses and severely threatens people's safety. Solving this problem is considered to be one of the major challenges of modern industrialized countries. Therefore, timely determination of the extent of corrosion is significant for equipment maintenance and prevention of property damage.

The vast majority of corrosion occurs on high-altitude cables and in chemical areas that are inaccessible. Therefore, embedding efficient image processing into mobile devices will significantly improve detection efficiency and range [2]. In recent years, image processing based on deep-learning has clearly improved detection accuracy and has wide application in industrial inspections [3].

Because corrosion occurs randomly and irregularly on metal surfaces, deep semantic segmentation is better suited to detecting corrosion compared to target detection and classification. However, deep-learning algorithms tend to have a large number of parameters and require huge computational resources, especially for semantic segmentation. Currently, there in some research into lightweight deep-learning models [4–6] and real-time semantic segmentation [7], the algorithms of which follow the framework of fully convolutional

networks (FCNs) for the most part [8]. However, the traditional FCN [8] framework follows the idea of classification and ignores contextual features. Some central moment (CM) [9–11] models (CM–FCN) capture the long-range dependencies of semantic features well by performing a contextual module after the encoding stage but overlooking the detail dependencies among pixels. Based on the above analysis, a new, light, deep-learning model is proposed in this paper that captures contextual information at all stages for corrosion segmentation as shown in Figure 1. The novelty of this study and its superiority over other studies comes from three aspects:

(1) We present a mixture of large and small kernels to acquire spatial and semantic contextual information and perform superior corrosion segmentation.
(2) We follow the ShuffleNetv2 to alleviate the computational overhead caused by large kernels and to embed high-precision models into mobile devices more appropriately.
(3) The creation of a fused multi-scale feature promotes information acquisition under limited resources.

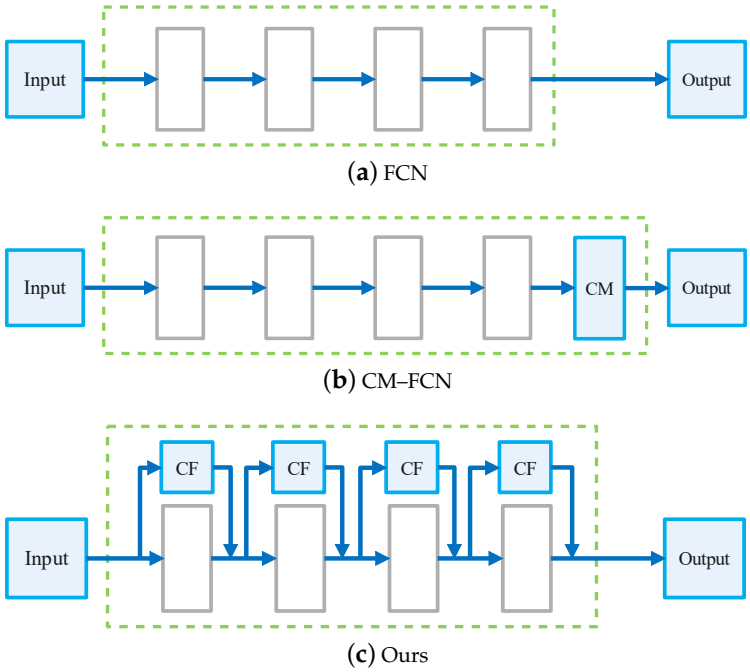

Figure 1. Common architectures for semantic segmentation. CM: contextual modules, CF: contextual features. (**a**) The FCN framework follows the guidelines of classification, which ignore contextual information. (**b**) CM–FCN model uses CM to extract contextual features of semantic information only in the last stage of the network. (**c**) The proposed model extracts the contextual features at all stages of the network.

To summarize, our main contributions are

- The design of large and small convolution kernels at all stages in the encoder to capture the long-distance dependencies between pixels and local detailed information and the design of a novel, simple decoder structure to fuse multi-scale information to improve model accuracy in an end-to-end residual deep-learning framework.
- The reduction of the model size by borrowing the core idea of ShuffleNetv2. The depthwise separable convolution and channel split operations reduce the computational overhead from large kernels, and the channel shuffle further improves feature representation capability.
- The design of two differently sized models to accommodate a variety of application scenarios. Extensive experimental results on a benchmark dataset showed that both models outperformed the state-of-the-art methods in model size and in accuracy trade-off for corrosion segmentation and degree evaluation.

2. Related Works

In this section, we introduce related works including corrosion segmentation methods as well as typical and popular semantic segmentation deep models. Some corrosion area detection methods include segmentation post-processing, but these methods don't distinguish between detection and segmentation strictly, so they are discussed in the corrosion segmentation section.

2.1. Corrosion Segmentation

To detect corrosion and segmentation, traditional image-processing and machine-learning methods are widely used: wavelet domain analysis combined with support vector machine (SVM) [12], threshold segmentation methods [13], color space-based analysis [14] and texture analysis [15]. These are simple and customized for specific applications, but they rely heavily on manual feature extraction and are difficult to generalize.

Because of its rapid development in recent years, deep learning performs prominently in many classical vision tasks. Its great advantage is that it learns features automatically.

Ortiz et al. [16] used a three-layered feedforward neural network for corrosion detection on ships. Zhang et al. [17] proposed a channel attention-based metallic corrosion detection (CAMCD) method. Squeeze-and-Excitation (SE) [18] attention improved corrosion detection performance in ResNet [19] networks. Xu et al. [20] proposed a method based on Faster R-CNN, which accurately found the corrosion area on a coated metal plate. Hou et al. [21] used a cascading Mask R-CNN network combined with transfer learning and a cable inspection robot to solve automatically the accuracy and location problems in a stay-cable surface inspection. The proposed method reached the best IoU (0.743) and F1-Score (85.1%) among the classic canny algorithm and mainstream segmentation networks. Fondevik et al. [22] built a corrosion dataset for the segmentation and performance evaluation of the pyramid scene parsing network (PSPNet) [11] and Mask R-CNN [23] for semantic segmentation and instant segmentation on this dataset, respectively. The authors also developed a two-stage data augmentation scheme that has been empirically shown to reduce overfitting significantly and improve, for instance, segmentation performance. The above two papers both used the Mask R-CNN framework [23], which combines Faster R-CNN [24] with FCN [8] to be an effective framework for corrosion segmentation. These methods confirmed the great potential of deep-learning models for corrosion segmentation. However, most of them are focused on specific applications and ignore the essential characteristics of general corrosion images, which makes implementing an algorithm for all corrosion image segmentation difficult. Because a corrosion image has no fixed pattern due to its randomness, its main features are distributed at different depths of the network. Thus, both spatial and semantic levels of contextual information need to be learned to reflect better the corrosion and multi-scale features that need to be incorporated so that the feature representation capability can be augmented.

2.2. Image Semantic Segmentation

As we know, large models perform better according to common sense, such as Deeplabv3+ [10] using Xception [25] as the backbone of 208.70 MB, and DenseASPP [26] using ResNet50 [19] as backbone of 103.79 MB. However, these high-accuracy large models are unsuitable for mobile devices. Meanwhile, some real-time semantic segmentation approaches are proposed to solve the trade-off between model volume and accuracy. ENet [27] removes the last layer of pooling indexes so that the final number of output feature maps is equal to the number of categories. ICNet [28] uses a cascade of feature fusion units with a low-resolution image to capture semantic information, and medium-resolution and high-resolution images to capture details. ESPNet [29] is an efficient semantic segmentation network for high-resolution images under limited computational resources. LinkNet [30] adds high-resolution residuals in the decoder to recover lost details. ERFNet [31] uses asymmetric convolution to further reduce the number of parameters. These networks have

fewer parameters and seek better real-time performance, but the limited receptive fields restrict them to discover rich contextual information for high precise segmentation.

To solve the above problems, CGNet [32] proposed that a CG block extract local and surrounding contextual features. ContextNet [33] proposed a two-branch network to extract global contextual information and retain detailed information. EDANet [34] uses dense connected asymmetric convolution to obtain information at different scales. Most of these networks carry out extensive dilation convolution to expand the receptive field and aggregate contextual information. However, discontinuous convolution kernels may lead to raster-like segmentation regions. Furthermore, it was the first time that RepLKNet [35] used very large convolution kernels to capture contextual information.

To apply a combination of traditional machine-learning methods and deep-learning models to segmentation is another way to establish long-range dependencies. Condition Random Field (CRF) as Recurrent Neural Network (RNN), CRFasRNN [36], combines CNN and CRF to build an end-to-end network. EMANet [37] uses the classical EM algorithm to maintain interclass differences and reduce intraclass differences, effectively reducing the complexity of non-local blocks [9,38–40]. Furthermore, DANet [9] proposes dual self-attention to model long-range dependencies. CCNet [38] captures long-range dependencies only in the horizontal and vertical directions, effectively reducing the amount of self-attentive computation. There are also some semantic segmentation studies based on ViT [39,41,42]. These networks are effective in obtaining global contextual information, but they introduce a large computational overhead. Based on the above analysis, contextual information is very important for improving segmentation accuracy. Current semantic segmentation methods, which use dilated convolution or a global block to extract contextual information, are unable to balance accuracy and computational overhead for corrosion segmentation. To solve this problem, a new lightweight model has been proposed that uses large convolution kernels inspired by RepLKNet together with ShuffleNetv2 [6] to satisfy the high precision and light-weight requirements simultaneously.

3. Method

In this section, we introduce the most significant module in our network architecture, the local contextual block, and then present the details of the proposed model.

3.1. Local Contextual Block

Long-distance dependency is very important for image semantic segmentation, and it is always a topical issue. Here, the local contextual block was proposed for expanding the receptive field and building a long-distance attachment. The overall architecture of the local contextual block is shown in Figure 2.

The local contextual block mainly consists of convolution kernels of different size and point convolution. Large convolution kernels are used to capture a larger area of corrosion and to establish long-range dependencies between pixels. The small convolution kernels are used to capture detailed local information. Depthwise separable convolution is used to mitigate the problem of huge parameters when using large convolution kernels. Inspired by ShuffleNetv2 [6], only half of the channels were involved in convolution at a time; the others are directly connected to the output. A channel shuffle operation is used to facilitate channel communication [6]. In this way, our approach not only reduced the number of parameters, but also improved the generalization performance.

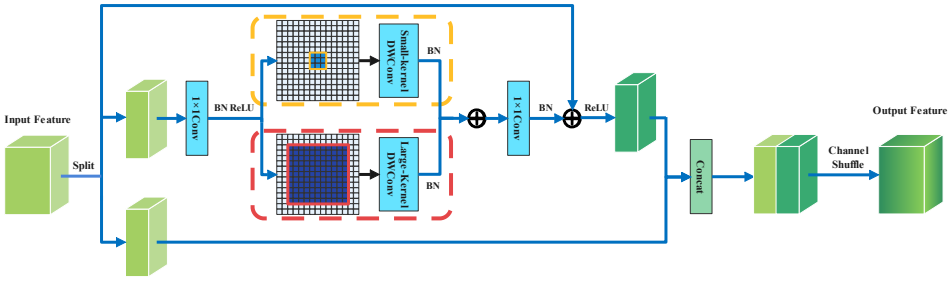

Figure 2. Details of the local contextual block. DWConv stands for DepthWise Convolution.

To improve the real-time performance of the algorithm, the input and output channels of the local contextual block were the same. Suppose the input channel is C. The sizes of the large and small convolution kernels are K_{large} ($K_{large} > 9$) and K_{small} ($K_{small} < 5$), respectively. The model parameters with a normal large kernel and DepthWise Convolution (DWConv) are as follows:

$$Normal : C^2({K_{large}}^2 + {K_{small}}^2) \tag{1}$$

and

$$DWConv : \frac{C^2}{2} + \frac{C}{2}({K_{large}}^2 + {K_{small}}^2). \tag{2}$$

Supposing $C = 256$, $K_{large} = 31$ and $K_{small} = 3$, our parameter volume is greatly reduced to 0.247% of that of the original, traditional convolution model, which also provides the possibility of applying large convolution kernels in a lightweight network. Meanwhile, the residual structure [19] is a good way to solve the problem of gradient disappearance when the network goes deeper. The residual allows the input to be directly connected to the output, thus forming a constant mapping that facilitates fast forward propagation of the signal. It prevents network degradation problems well and accelerates the network convergence. We followed this design and introduced the residual structure to the left part of the local contextual block as shown in (3).

$$Out = CS(Concat(right, \mathcal{F}(left) + left)), \tag{3}$$

where $left$ and $right$ mean the two branches after splitting; $\mathcal{F}(x)$ represents the convolution operation for the target; CS represents the channel shuffle operation; and Out represents the final output.

3.2. Encoder–Decoder Architecture

To improve preservation of original pixel position information, we removed all pooling layers and changed the downsampling factor to 8× in the encoder. The image was downsampled at the beginning to filter out irrelevant information to alleviate the computational effort of the subsequent process. In stage 1, small convolution kernels were used to extract local information, and then a downsampling module from ShuffleNetv2 was employed, as shown in Figure 3. After each downsampling step, feature maps at different scales were extracted by stacking the local contextual block. The detailed parameter settings are shown in Table 1.

We implemented a simple decoder architecture to fuse multi-scale features. The four feature scales are first compressed to the same number of channels C, and then upsampled to recover to the same resolution $H \times W$. Subsequently, concatenation was performed in the channel dimension, and then the final result was obtained through two layers of point convolution as shown in (4). The overall architecture of the proposed network is shown in Figure 3.

$$\begin{aligned} \hat{F}_i &= Conv(C_i, C)(F_i), \forall_i \\ \hat{F}_i &= Upsample(H \times W)(\hat{F}_i), \forall_i \\ F &= Conv(4C, C)(Concat(\hat{F}_i)), \forall_i \\ M &= Conv(C, nclass)(F) \end{aligned} \tag{4}$$

where M refers to the predicted output; $nclass$ represents the number of classes; F_i stands for the feature; $Conv(C_{in}, C_{out})$ refers to the input channel C_{in} and the output channel C_{out} after point convolution; and $Upsample(H \times W)$ represents up-sampling to the size of $H \times W$. Unlike the huge decoder structure of Unet [43] and SegNet [44], our decoder achieved high accuracy with a small number of parameters.

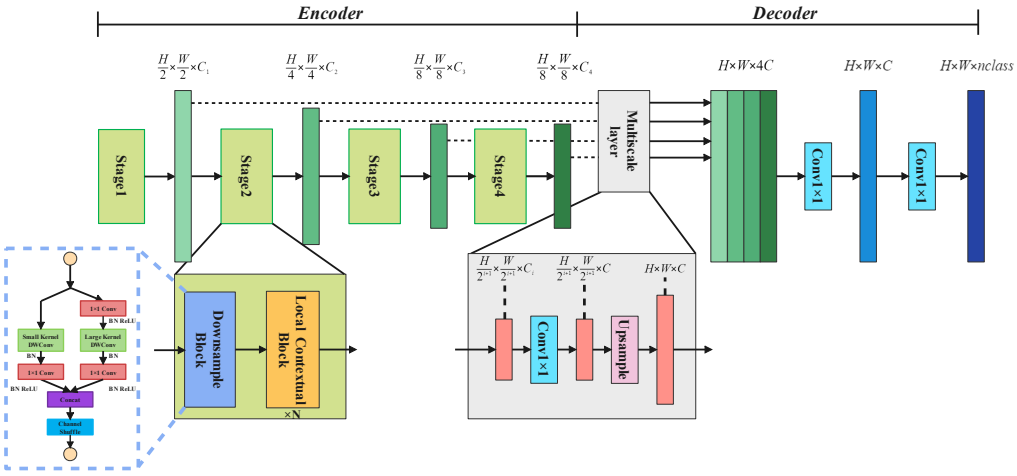

Figure 3. The overall architecture of the proposed network, which consists of two main modules. The CNN encoder extractd details and semantic features, and the decoder fuses the multi-scale features. The specific module parameters of the encoder are shown in Table 1.

To explore the most suitable kernel size, we tested the changing accuracy with the kernel size variation as shown in Figure 4. The corresponding model parameters can also be seen. We finally choose 17 × 17 for the large kernal size to make a good trade-off between model size and segmentation accuracy.

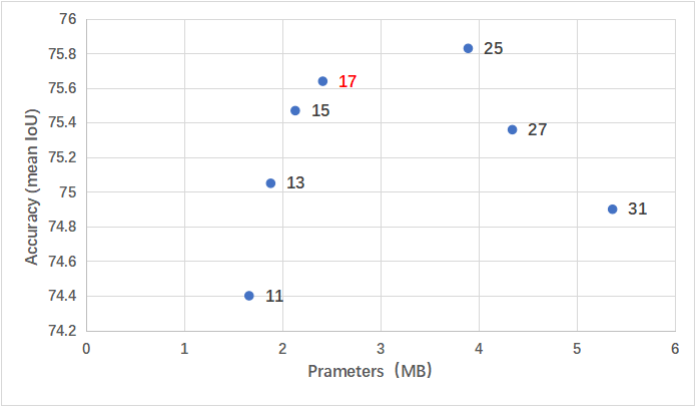

Figure 4. The accuracy and model size vs. the kernel size. A good trade-off was obtained when the kernel size was 17 × 17.

We designed two different sizes of models, small and large, for devices that had different computing power. Large models had relatively more channels compared to small models and thus a larger model size. The large model was more suitable for sensitive equipment or applications that require high accuracy in corrosion segmentation of metal

surfaces, such as aerospace equipment, hydride tanks, automotive wheels, or the metal links of bridges. The small model requires daily inspections. By embedding the corrosion segmentation model in drones, mobile robots and edge computation chips, the corrosion area and degree of corrosion were obtained online and transmitted to the upper computer to help engineers make further decisions.

Table 1. The detailed parameter settings of the encoder. The large and small models had different output channels and model sizes.

Layer	Kernel Size	Stride	Repeat	Output Channels	
				Small	Large
Image				3	3
Stage 1	3 × 3	2	1	32	72
	3 × 3 (DW)	1	1	32	72
	1 × 1	1	1	32	72
Stage 2		2	1	32	72
		1	3	64	144
Stage 3		2	1	64	144
		1	7	128	288
Stage 4		1	1	128	288
		1	3	256	576
Params				2.41 MB	8.25 MB

4. Experiment

In this section, we evaluated the performance of our algorithm on a public corrosion image dataset [45] and then compared it with existing semantic segmentation algorithms and conducted ablation studies to demonstrate the effectiveness of our approach.

4.1. Experiment Settings

Data Acquirement and Augmentation. We used a corrosion image dataset produced by University Libraries Virginia Tech [45], which was collected from Virginia Department of Transportation bridge inspection reports. These were annotated semantically according to the corrosion condition status guidelines reported by the American Association of State Highway and Transportation Officials (AASHTO) and the Bridge Inspector's Reference Manual (BIRM). The entire dataset was divided into four categories: Background, Fair, Poor, and Severe. We expanded the number of images from 440 to 3850 through data enhancement, such as random cropping and contrast or saturation enhancement. The size of the image was adjusted to 512 × 512.

Implementation Protocol. All experiments were performed on the PyTorch 1.10.1 with 1× V100GPU and 4× 2080Ti. The network was trained using the SGD optimizer with an initial learning rate of 0.007, which declined to 0.00007 by the cosine curve method. The loss function as shown in (7) was defined as the sum of Dice loss (5) and Focal loss (6). The α_t was used to regulate the ratio between the two losses. The Dice loss [46] is a metric function used to calculate the similarity between set X and set Y. $|X|$ and $|Y|$ stand for the number of pixels in X and Y, respectively, and $|X \bigcap Y|$ represents the intersection pixel numbers between X and Y. The Focal loss [47] is an improved cross-entropy loss function to solve the sample imbalance, where γ is the modulation factor guiding the algorithm focus on the difficult samples and p_t is the proportion of pixel numbers of four categories to the total numbers in image. Based on experience, we set $\gamma = 2$ and $\alpha_t = 0.5$:

$$DL = 1 - \frac{2|X \bigcap Y|}{|X| + |Y|} \tag{5}$$

$$FL(p_t) = -(1 - p_t)^{\gamma} log(p_t) \tag{6}$$

$$Totalloss = (1 - \alpha_t)DL + \alpha_t FL(p_t) \tag{7}$$

Evaluation Metrics. Five indicators were employed to evaluate performance through all experiments: mean Pixel Accuracy (mPA), mean Intersection over Union (mIoU), F1-score, Frames Per Second (FPS) and the volume of the model. We used 2080Ti uniformly to test the FPS of all models with the same image. The model with high evaluation metrics and a small number of parameters is what we expected.

$$mIoU = \frac{1}{k+1}\sum_{i=0}^{k}\frac{p_{ii}}{\sum_{j=0}^{k}p_{ij} + \sum_{j=0}^{k}p_{ji} - p_{ii}} \tag{8}$$

$$mPA = \frac{1}{k+1}\sum_{i=0}^{k}\frac{p_{ii}}{\sum_{j=0}^{k}p_{ij}} \tag{9}$$

where p_{ij} denotes the number of true values of i that are predicted to be j; $K+1$ is the number of categories (including the background); p_{ii} is the True Positive(TP); and p_{ij} and p_{ji} denote false positive (FP) and false negative (FN), respectively.

4.2. Comparison with Similar Methods

We mainly compared the proposed method with mainstream lightweight and classical semantic segmentation models. The results of the experiment are shown in Table 2. As it shows, our proposed small model performed the best, achieving the highest scores on mPA, mIoU, F1-score among the lightweight models. The volume of the model came third behind ENet [27] and CGNet [32]. The large model achieved an mIoU of 79.06 and a volume of the model of only 8.25 MB. Metal corrosion images were characterized by both continuity and localization. Continuity was reflected in the fact that corrosion areas tended to appear in patches; therefore, we directly used a large convolution kernel with higher accuracy than that used for dilated convolution. The local nature reflected in the corrosion image did not take global constraints as corrosion occurred locally and randomly. Therefore, when we kept increasing its size, the larger convolution kernel lost the localization feature, leading to a dramatic drop in segmentation accuracy. Therefore, designing a large convolution kernel that did not lose localization features is key to improving the accuracy of the metal corrosion image segmentation method. This was also demonstrated in the ablation study section, where there was an approximate parabolic relationship between algorithm accuracy and convolution kernel size.

Table 2. Comparison between our model and other SOTA image segmentation methods.

Model	Params (MB)	mIoU(%)	mPA(%)	F1-Score	FPS
ENet [27]	**1.36**	67.79	80.64	0.766	43.55
CGNet [32]	1.88	73.32	85.09	0.820	12.77
EDANet [34]	2.60	67.49	80.49	0.761	21.79
DABNet [48]	2.87	68.89	79.95	0.779	46.30
ContextNet [33]	3.34	69.17	81.35	0.828	**104.78**
ESPNetv2 [29]	4.75	72.13	83.34	0.798	38.61
ERFNet [31]	7.87	69.10	81.12	0.794	41.43
Ours–small	2.41	**75.64**	**86.07**	**0.838**	16.87
Deeplabv3+ [10]	22.18	77.67	**88.96**	0.879	25.58
LinkNet [30]	44.00	67.58	77.82	0.820	**48.22**
SegNet [44]	112.32	58.37	70.89	0.778	12.89
CCNet [38]	200.69	78.47	86.58	**0.899**	7.42
Ours–large	**8.25**	**79.06**	88.07	0.891	11.60

As for training, all the models converged within 300 epochs without pre-training in cases where other configurations were all the same. We found that CCNet [38] and Deeplabv3+ [10], also performed better, which proved the importance of contextual features for corrosion segmentation. SegNet [44] had a large number of parameters, but the final performance was the worst. The reason may be that SegNet [44] performed 32-fold

downsampling, resulting in a large information loss. In contrast, models like CGNet [32], DABNet [48], ContextNet [33], ENet [27], ERFNet [31] performed 8-fold downsampling. Their final performances were also better compared to SegNet [44], which confirmed our conjecture. EDANet [34] performed poorly, probably because the dense, connected approach introduced redundant information. Regarding the FPS evaluation, our small model scored 16.87 while the large model scored 11.60, which met the real-time requirements of daily testing. The model capacity of both the large and small models did not exceed 10 MB, which can easily be embedded on mobile devices for corrosion segmentation. Compared with the small model, the large model required more computing power and higher accuracy.

After that, we used the above models to make predictions, and the visualization results are shown in Figure 5. CGNet [32] and ContextNet [33] appeared to have discontinuous segmentation regions due to the use of dilated convolution. EDANet [34] and ENet [27] could not accurately identify the degree of corrosion area because of their weak fitting ability. The final segmentation effect of Deeplabv3+ [10] was better than that of the lightweight network, but it was not accurate enough to grasp the boundary of the corrosion region. Our method was more precise and closer to the ground truth from the detected corrosion areas and their corrosion degree. However, the effect of our model was not very good for part of the small area corrosion segmentation. It will be the focus of our subsequent research.

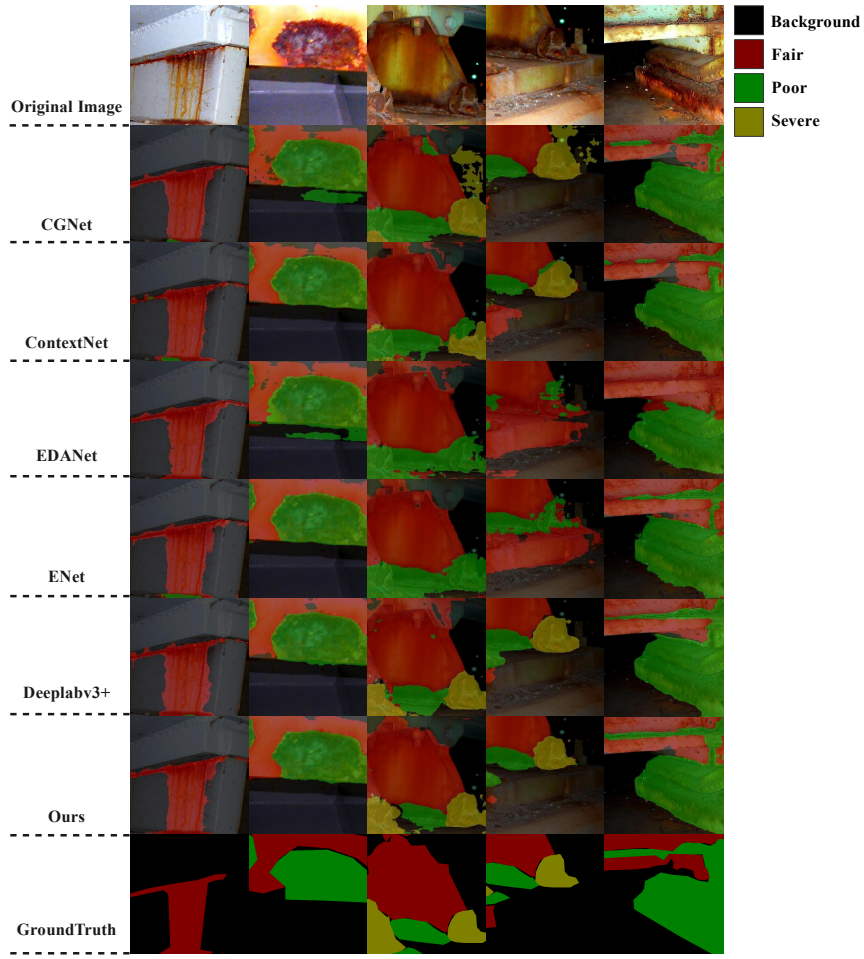

Figure 5. Visualization comparison results between our model and other SOTA methods, where Background, Fair, Poor and Severe represented the four degrees of corrosion. Best viewed in color.

4.3. Ablation Study

We evaluated the various design decisions for our method as well as its components.

Ablation Study for the Mixture of Small and Large Kernel. Since both the small and large kernels are meant to capture different types of corrosion features, we ran an ablation

by removing each of these kernels. The results are presented in Table 3. We found that both large and small kernels had improved accuracy, which confirmed that precise segmentation results need not only a larger range of receptive fields but also local detailed information. It should be noted that the large kernel brings a larger computational overhead but improves accuracy over the small kernel. The best performance was achieved when a mixture of small and large kernels was used.

Table 3. Ablation study results for the mixture of small and large kernels.

Small Kernel	Large Kernel	Params	mIoU	mPA	F1-Score
w/o	w	2.36 MB	75.30%	85.64%	0.824
w	w/o	**1.38 MB**	71.38%	82.95%	0.793
w	w	2.41 MB	**75.64**%	**86.07**%	**0.838**

Ablation Study for Residual Connection. Inspired by ResNet [19], we used residual learning in the local contextual blocks to improve feature representation ability (Table 4). We found that the residual connection improved the mIoU from 74.67 to 75.61% without adding extra parameters. One possible reason is that the residual connection had a stronger ability to facilitate the flow of information in the network.

Table 4. Ablation study results for residual connection.

Residual	Params	mIoU	mPA	F1-Score
w/o	2.41 MB	74.67%	85.55%	0.821
w	2.41 MB	**75.64**%	**86.07**%	**0.838**

Ablation Study for Local Contextual Block. To verify the effect of local contextual information, We replace the local contextual block with an inverted residual block, and the ablation study result is shown in Table 5. For convenience, we referred to the inverted residual and local contextual blocks as IRB and LCB, respectively. The IRB was a classical lightweight block from MobileNetV2 [4]. It first used 1×1 convolution to expand the dimension of the input feature map, then performed the convolution operation with 3×3 depthwise convolution. Finally, it used 1×1 convolution to reduce its dimension. We set the expansion rate to 1 to ensure our model had the same settings. We found that the parameters of the model with the IRB was 2.04 MB, which was slightly lighter than our model caused by the large kernel, but the other evaluation metrics were inferior to ours. The results indicated that the local contextual feature was efficient for corrosion segmentation.

Table 5. The ablation study results for local contextual block.

IRB	LCB	Params	mIoU	mPA	F1-Score
w	w/o	**2.04 MB**	74.53%	85.31%	0.813
w/o	w	2.41 MB	**75.64**%	**86.07**%	**0.838**

5. Conclusions

In this work, we proposed a lightweight deep encoder–decoder network to learn contextual features at all stages to solve the corrosion segmentation task. Our proposed encoder part consists of a local contextual block and a downsample block that used a mixture of large and small convolution kernels to establish long-range dependencies between pixels and local detailed information. Moreover, we used the core idea of ShuffleNetv2 and a residual connection to reduce the model size and further improve model accuracy. An efficient and simple decoder was proposed to fuse multi-scale features at different stages to

augment feature representation capability. With the proposed architecture, we designed two differently sized models and consistently showed excellent corrosion segmentation with a better trade-off between segmentation accuracy and model size. Our proposed small model achieved the best performance for mIoU (75.61%), mPA (86.07%) and F1-score (0.838) with parameters of only 2.41 MB, while the large model remained competitive with 8.25 MB and achieved the best mIoU (79.06%). In future work, we may focus on dataset collection. Through training with larger and more complex datasets, the performance of our model is expected to improve, and a wide range of applications will be covered. Meanwhile, embedding the proposed model into the edge computing chip on site and further optimizing the model constitutes our current work.

Author Contributions: Conceptualization, J.H., Q.L., L.X., G.L. and Y.Z.; Methodology, J.H., Q.L. and L.X.; Funding acquisition, Q.L. and W.C.; Investigation, L.X., G.L. and Y.Z.; Writing code and performing computational experiment, J.H., Q.L. and L.X.; Writing—original draft, J.H., Q.L. and L.X.; Writing—review & editing, J.H., Q.L., L.X., Y.Z. and W.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the Beijing Natural Science Foundation (4202026), Qin Xin Talents Cultivation Program of Beijing Information Science and Technology University (QXTCP A202102).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The experimental data that was used in this work can be accessed at https://github.com/One-LL/A-Lightweight-Residual-Model-for-Corrosion-Segmentation-with-Local-Contextual-Information, accessed on 10 August 2021.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Mazumder, M.A.J. Global impact of corrosion: Occurrence, cost and mitigation. *Glob. J. Eng. Sci* **2020**, *5*, 4. [CrossRef]
2. Zhang, C.; Patras, P.; Haddadi, H. Deep learning in mobile and wireless networking: A survey. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 2224–2287. [CrossRef]
3. Luo, D.; Cai, Y.; Yang, Z.; Zhang, Z.; Zhou, Y.; Bai, X. Survey on industrial defect detection with deep learning. *J. Sci. Sin. Inf.* **2022**, *52*, 1002–1039. [CrossRef]
4. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
5. Howard, A.; Sandler, M.; Chu, G.; Chen, L.C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V. Searching for mobilenetv3. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 1314–1324.
6. Ma, N.; Zhang, X.; Zheng, H.T.; Sun, J. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 116–131.
7. Takos, G. A survey on deep learning methods for semantic image segmentation in real-time. *arXiv* **2020**, arXiv:2009.12942.
8. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
9. Fu, J.; Liu, J.; Tian, H.; Li, Y.; Bao, Y.; Fang, Z.; Lu, H. Dual attention network for scene segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 3146–3154.
10. 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 European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 801–818.
11. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2881–2890.
12. Morizet, N.; Godin, N.; Tang, J.; Maillet, E.; Fregonese, M.; Normand, B. Classification of acoustic emission signals using wavelets and Random Forests: Application to localized corrosion. *Mech. Syst. Signal Process.* **2016**, *70*, 1026–1037. [CrossRef]
13. Hoang, N.D. Image processing-based pitting corrosion detection using metaheuristic optimized multilevel image thresholding and machine-learning approaches. *Math. Probl. Eng.* **2020**, *2020*, 6765274. [CrossRef]

14. Zou, Z.; Ma, L.; Fan, Q.; Gan, X.; Qiao, L. Feature recognition of metal salt spray corrosion based on color spaces statistics analysis. In Proceedings of the SPIE Optical Engineering + Applications, San Diego, CA, USA, 6–10 August 2017; Volume 10396; pp. 562–569.
15. Atha, D.J.; Jahanshahi, M.R. Evaluation of deep learning approaches based on convolutional neural networks for corrosion detection. *Struct. Health Monit.* **2018**, *17*, 1110–1128. [CrossRef]
16. Ortiz, A.; Bonnin-Pascual, F.; Garcia-Fidalgo, E.; Company-Corcoles, J.P. Vision-based corrosion detection assisted by a micro-aerial vehicle in a vessel inspection application. *Sensors* **2016**, *16*, 2118. [CrossRef] [PubMed]
17. Zhang, S.; Deng, X.; Lu, Y.; Hong, S.; Kong, Z.; Peng, Y.; Luo, Y. A channel attention based deep neural network for automatic metallic corrosion detection. *J. Build. Eng.* **2021**, *42*, 103046. [CrossRef]
18. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
19. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
20. Xu, X.; Wang, Y.; Wu, J.; Wang, Y. Intelligent corrosion detection and rating based on faster region-based convolutional neural network. In Proceedings of the 2020 Global Reliability and Prognostics and Health Management (PHM-Shanghai), Shanghai, China, 16–18 October 2020; pp. 1–5.
21. Hou, S.; Dong, B.; Wang, H.; Wu, G. Inspection of surface defects on stay cables using a robot and transfer learning. *Autom. Constr.* **2020**, *119*, 103382. [CrossRef]
22. Fondevik, S.K.; Stahl, A.; Transeth, A.A.; Knudsen, O. Image Segmentation of Corrosion Damages in Industrial Inspections. In Proceedings of the 2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI), Baltimore, MD, USA, 9–11 November 2020; pp. 787–792.
23. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
24. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Washington, DC, USA, 7–13 December 2015; pp. 1440–1448.
25. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1251–1258.
26. Yang, M.; Yu, K.; Zhang, C.; Li, Z.; Yang, K. Denseaspp for semantic segmentation in street scenes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 3684–3692.
27. Paszke, A.; Chaurasia, A.; Kim, S.; Culurciello, E. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv* **2016**, arXiv:1606.02147.
28. Zhao, H.; Qi, X.; Shen, X.; Shi, J.; Jia, J. Icnet for real-time semantic segmentation on high-resolution images. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 405–420.
29. Mehta, S.; Rastegari, M.; Caspi, A.; Shapiro, L.; Hajishirzi, H. Espnet: Efficient spatial pyramid of dilated convolutions for semantic segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 552–568.
30. Chaurasia, A.; Culurciello, E. Linknet: Exploiting encoder representations for efficient semantic segmentation. In Proceedings of the 2017 IEEE Visual Communications and Image Processing (VCIP), St. Petersburg, FL, USA, 10–13 December 2017; pp. 1–4.
31. Romera, E.; Alvarez, J.M.; Bergasa, L.M.; Arroyo, R. Erfnet: Efficient residual factorized convnet for real-time semantic segmentation. *IEEE Trans. Intell. Transp. Syst.* **2017**, *19*, 263–272. [CrossRef]
32. Wu, T.; Tang, S.; Zhang, R.; Cao, J.; Zhang, Y. Cgnet: A light-weight context guided network for semantic segmentation. *IEEE Trans. Image Process.* **2020**, *30*, 1169–1179. [CrossRef] [PubMed]
33. Han, W.; Zhang, Z.; Zhang, Y.; Yu, J.; Chiu, C.C.; Qin, J.; Gulati, A.; Pang, R.; Wu, Y. Contextnet: Improving convolutional neural networks for automatic speech recognition with global context. *arXiv* **2020**, arXiv:2005.03191.
34. Lo, S.Y.; Hang, H.M.; Chan, S.W.; Lin, J.J. Efficient dense modules of asymmetric convolution for real-time semantic segmentation. In Proceedings of the ACM Multimedia Asia, Beijing, China, 16–18 December 2019; pp. 1–6.
35. Ding, X.; Zhang, X.; Han, J.; Ding, G. Scaling up your kernels to 31 $\times$ 31: Revisiting large kernel design in cnns. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–20 June 2022; pp. 11963–11975.
36. Zheng, S.; Jayasumana, S.; Romera-Paredes, B.; Vineet, V.; Su, Z.; Du, D.; Huang, C.; Torr, P.H. Conditional random fields as recurrent neural networks. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1529–1537.
37. Li, X.; Zhong, Z.; Wu, J.; Yang, Y.; Lin, Z.; Liu, H. Expectation-maximization attention networks for semantic segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019; pp. 9167–9176.
38. Huang, Z.; Wang, X.; Huang, L.; Huang, C.; Wei, Y.; Liu, W. Ccnet: Criss-cross attention for semantic segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019; pp. 603–612.
39. Zheng, S.; Lu, J.; Zhao, H.; Zhu, X.; Luo, Z.; Wang, Y.; Fu, Y.; Feng, J.; Xiang, T.; Torr, P.H. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 6881–6890.

40. Cao, Y.; Xu, J.; Lin, S.; Wei, F.; Hu, H. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, Seoul, Korea, 27–28 October 2019.
41. Chen, J.; Lu, Y.; Yu, Q.; Luo, X.; Adeli, E.; Wang, Y.; Lu, L.; Yuille, A.L.; Zhou, Y. Transunet: Transformers make strong encoders for medical image segmentation. *arXiv* **2021**, arXiv:2102.04306.
42. Xie, E.; Wang, W.; Yu, Z.; Anandkumar, A.; Alvarez, J.M.; Luo, P. SegFormer: Simple and efficient design for semantic segmentation with transformers. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 12077–12090.
43. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; pp. 234–241.
44. Badrinarayanan, V.; Kendall, A.; Cipolla, R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [CrossRef] [PubMed]
45. Bianchi, E.; Hebdon, M. *Corrosion Condition State Semantic Segmentation Dataset*; University Libraries, Virginia Tech: Blacksburg, VA, USA, 2021. [CrossRef]
46. Milletari, F.; Navab, N.; Ahmadi, S.A. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In Proceedings of the 2016 Fourth International Conference on 3D Vision (3DV), Stanford, CA, USA, 25–28 October 2016; pp. 565–571.
47. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
48. Li, G.; Yun, I.; Kim, J.; Kim, J. Dabnet: Depth-wise asymmetric bottleneck for real-time semantic segmentation. *arXiv* **2019**, arXiv:1907.11357.