remote sensing

MDPI

Article

Trans-DCN: A High-Efficiency and Adaptive Deep Network for Bridge Cable Surface Defect Segmentation

Zhihai Huang [1], Bo Guo [2,*], Xiaolong Deng [1], Wenchao Guo [1] and Xing Min [3]

[1] School of Civil and Transportation Engineering, Guangdong University of Technology, Guangzhou 510006, China; 2112209044@mail2.gdut.edu.cn (Z.H.); 2112309040@mail2.gdut.edu.cn (X.D.); 2112109051@mail2.gdut.edu.cn (W.G.)
[2] School of Architecture and Urban Planning, Shenzhen University, Shenzhen 518060, China
[3] Guangdong Metro Design Research Institute Co., Ltd., Guangzhou 510499, China; minxing@dtsjy.com
* Correspondence: guobo@gdut.edu.cn

Abstract: Cables are vital load-bearing components of cable-stayed bridges. Surface defects can lead to internal corrosion and fracturing, significantly impacting the stability of the bridge structure. The detection of surface defects from bridge cable images faces numerous challenges, including shadow disturbances due to uneven lighting and difficulties in addressing multiscale defect features. To address these challenges, this paper proposes a novel and cost-effective deep learning segmentation network, named Trans-DCN, to detect defects in the surface of the bridge cable. The network leverages an efficient Transformer-based encoder and integrates multiscale features to overcome the limitations associated with local feature inadequacy. The decoder implements an atrous Deformable Convolution (DCN) pyramid and dynamically fuses low-level feature information to perceive the complex distribution of defects. The effectiveness of Trans-DCN is evaluated by comparing it with state-of-the-art segmentation baseline models using a dataset comprising cable bridge defect images. Experimental results demonstrate that our network outperforms the state-of-the-art network SegFormer, achieving a 27.1% reduction in GFLOPs, a 1.2% increase in mean Intersection over Union, and a 1.5% increase in the F1 score. Ablation experiments confirmed the effectiveness of each module within our network, further substantiating the significant validity and advantages of Trans-DCN in the task of bridge cable defect segmentation. The network proposed in this paper provides an effective solution for downstream cable bridge image analysis.

Keywords: deep learning; defect segmentation; bridge cable surface defects; non-destructive cable health assessment

Citation: Huang, Z.; Guo, B.; Deng, X.; Guo, W.; Min, X. Trans-DCN: A High-Efficiency and Adaptive Deep Network for Bridge Cable Surface Defect Segmentation. *Remote Sens.* **2024**, *16*, 2711. https://doi.org/10.3390/rs16152711

Academic Editors: Joan Ramon Casas Rius, Massimiliano Pieraccini, Necati Catbas, Rolando A. Chacón and Daniele Zonta

Received: 22 May 2024
Revised: 12 July 2024
Accepted: 22 July 2024
Published: 24 July 2024

1. Introduction

Bridge cables have undergone significant technological advancements, evolving from iron chains to modern high-strength steel cables. These cables play a crucial role in expanding bridge spans and ensuring structural stability and safety, serving as essential load-bearing components in suspension bridges and cable-stayed bridges. However, due to exposure to the harsh natural environment, high-intensity load, and material corrosion, bridge cables are susceptible to various defects, including cracks, potholes, scratches, etc. The surface defects may result in the rapid deterioration of the bridge cables, posing significant threats to bridge safety [1]. Therefore, timely monitoring of the structural health of the bridge cables is imperative to inform appropriate maintenance interventions [2].

The traditional methods for detecting bridge cable defects relied solely on manual visual inspection and basic physical measurement tools, such as laser scanning. Manual inspection is extremely labor-intensive and inefficient, particularly when confronted with numerous extreme working environments. With the progression of science and technology, certain physical technical applications, such as ultrasonic detection methods and optical fiber sensing technology, have gradually been implemented [3,4]. However, these

instruments tend to be costly and are more suited for detecting internal defects, making it challenging to observe and evaluate surface integrity. The potential incompleteness in surface inspection poses a hazard, as these surface defects can rapidly lead to internal corrosion and fractures. In recent years, computer vision technology and artificial intelligence algorithms have demonstrated their efficacy in various engineering domains and have been extensively employed in bridge cable defect detection.

Cable-climbing robots have become prevalent in inspecting bridge cable defects due to their efficient data acquisition capabilities and stability. Currently, numerous studies have explored the integration of cable-climbing robots with computer vision systems for bridge defect detection tasks [5–7]. Leveraging the widespread availability and convenience of these cable-climbing robots, we employed an embedded system-based vision acquisition device for cable defect detection. This device comprises four RGB cameras arranged around a central cable, a pulley power system, an image acquisition system, and a wireless control system, as shown in Figure 1. The robot can adapt to cables of various sizes and capture exterior images from four perspectives.

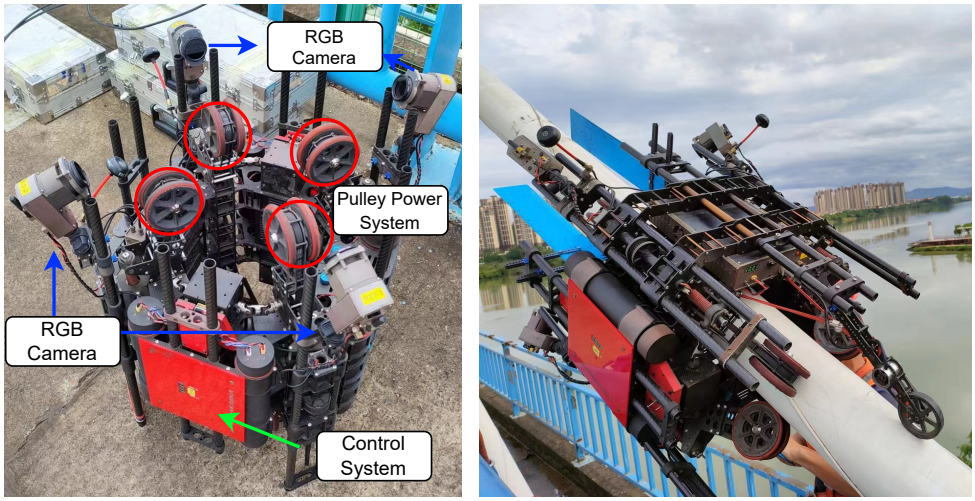

Figure 1. Cable-climbing robot. (**a**) Basic components of our cable-climbing robot. (**b**) Cable-climbing robot is deployed on the bridge cable, ready for work.

One of the primary challenges in bridge cable surface inspection lies in the effectively utilizing collected imagery to rapidly identify the precise locations of defects and damage details. Current research has integrated cable-climbing robots with computer vision for application in bridge cable defect detection tasks, encompassing both classical computer vision algorithms and advanced deep learning methodologies for detection purposes. The former often require significant manual intervention for preprocessing and struggle to cope with the challenges posed by lighting variations. The latter approach reduces the preprocessing stage but demands a large volume of image samples [8]. Moreover, the conventional network framework lacks generalization ability and robustness when confronted with changes in lighting and complex distributions of defects [9].

We identify two primary challenges in enhancing the generalization capability and robustness of deep neural networks for bridge cable defect extraction. The first challenge arises from constantly changing collection environments, such as insufficient lighting or angles obstructed by shadows, as well as complex background texture interference other than the cable itself. This can lead to the omission of local features and incorrect associations. It is crucial to fully extract effective feature information from the images. Transformers, based on the self-attention mechanism, establish direct global relationships among all pixels in the image, effectively capturing and utilizing long-range dependency information [10–12], thus effectively addressing this challenge. Therefore, in our proposed deep neural network, we introduce a novel Transformer as the encoder to comprehensively extract global contextual

feature relationships, compensating for the lack or absence of local features, a capability not inherent in convolutional methods based on local spatial aggregation [13]. Additionally, we integrate feature information from all stages, retaining potential defect-related features to precisely delineate defect areas according to the requirements in the decoder.

Another challenge arises from the intricate and varied genre of bridge cable defects, mainly reflected in the fact that the same defect may appear in multiple sizes and the defects exhibit a broad spectrum of changes and diverse textures. Particularly, cracks typically appear as thin dark lines or strips with constantly varying angle and direction, such as circumferential cracks, longitudinal cracks, and minor cracks. Standard convolutions struggle to adequately capture such elongated features, as different positions may correspond to defects of different scales or deformations [14]. Fixed spatial aggregation methods may confuse the encoding of positional information for deep networks. Therefore, it is imperative to comprehensively extract these defect regions. We proposed a Serial-Parallel pyramid module based on atrous Deformable Convolution (SPP-DCN) in the decoder, employing multiple progressively dilated Deformable Convolutions to adaptively capture the distribution of defect features. This is followed by the utilization of the Squeeze-and-Excitation (SE) mechanism to accentuate effective boundary information. Meanwhile, within the low-level layers of the encoder, which capture rich edge information at the local level, we introduce several shortcuts between the encoder and the decoder. An efficient Adaptive Spatial Feature Fusion (ASFF) module is employed to comprehensively integrate and summarize this edge information in the decoder, ensuring the integrity of defect extraction.

RGB natural images are a common data format in the field of defect detection. Consequently, our study focuses on developing a deep learning network for bridge cable defect detection based on RGB data input. Traditional segmentation frameworks often employ a simple encoder–decoder structure to extract features, followed by direct deconvolution or upsampling to obtain segmentation results. However, this approach is not well suited for the specific characteristics of cable defect detection. To achieve effective, rapid, and robust detection of surface defects in bridge cables and address the aforementioned challenges, we propose an end-to-end efficient semantic segmentation deep neural network named Trans-DCN. The Trans-DCN model features an efficient feature extraction encoder and a fully refined feature-aware decoder, which are crucial for bridge cable defect segmentation. The code is publicly available at https://github.com/hzh1231/Trans_dcn (accessed on 21 July 2024).

Thus, the primary contributions of this paper are as follows:

1. The introduction of an efficient backbone based on Transformer, complemented by a multi-layer feature aggregation module in the encoder, facilitating the comprehensive utilization of global contextual defect features.
2. The establishment of a serial–parallel structure featuring multiple atrous Deformable Convolutions, which dynamically adjust the receptive field according to the defect distribution, ensuring a comprehensive detection with multiple granularities.

In this paper, we specifically illustrated the technical details of the network and conducted experiments comparing it with state-of-the-art models. We also analyzed and discussed the effectiveness of each module in our work to demonstrate the new baseline performance in bridge cable defect detection.

2. Related Work

2.1. Bridge Cable Surface Defect Detection

Polyethylene (PE) pipes serve as the protective layer for bridge cables. Once the PE pipe ages or ruptures, corrosive substances can infiltrate the internal steel wires through the defect in the protective layer [15]. Therefore, it is crucial to assess the surface defect of the cables non-destructively [16]. Figure 2 illustrates some common surface defects found in bridge cables.

Traditional manual inspection methods are inefficient and subjective, making it difficult to meet the maintenance needs of modern bridges. Numerous researchers have

explored the utilization of cameras mounted on Unmanned Aerial Vehicles (UAVs) or cable-climbing robots to capture images of surface defects on bridges. These images are then subjected to intelligent analysis through computer vision techniques [7,17–19]. Threshold-based methods, which rely on the analysis of grayscale and gradient features of images, can achieve basic defect extraction [20,21]. However, directly applying image processing methods for defect detection encounters challenges regarding insufficient robustness, often necessitating extensive preprocessing to attain satisfactory results. With further technological advancements, machine learning, by constructing high-level feature representations, can effectively address the challenges posed by changes in lighting conditions [22]. Li et al. achieved high accuracy in defect image classification experiments by deploying the Support Vector Machine (SVM) based on particle swarm optimization [1]. Nonetheless, feature extraction for defect images still required the use of mathematical statistical methods.

Typically, deep learning has proven effective in detecting defects in domains with similar distributions [23]; some research has already achieved basic defect detection by utilizing simple and common network architectures [24–26]. However, these generic framework methods struggle to adapt to the specific characteristics of cable images, particularly when facing predominantly strip-like cracks, which often result in missed or false detections of defects. Therefore, we have developed an advanced approach for defect detection that takes into account environmental disturbances and the distribution characteristics of defects, with the aim of fully enhancing the potential of neural network applications.

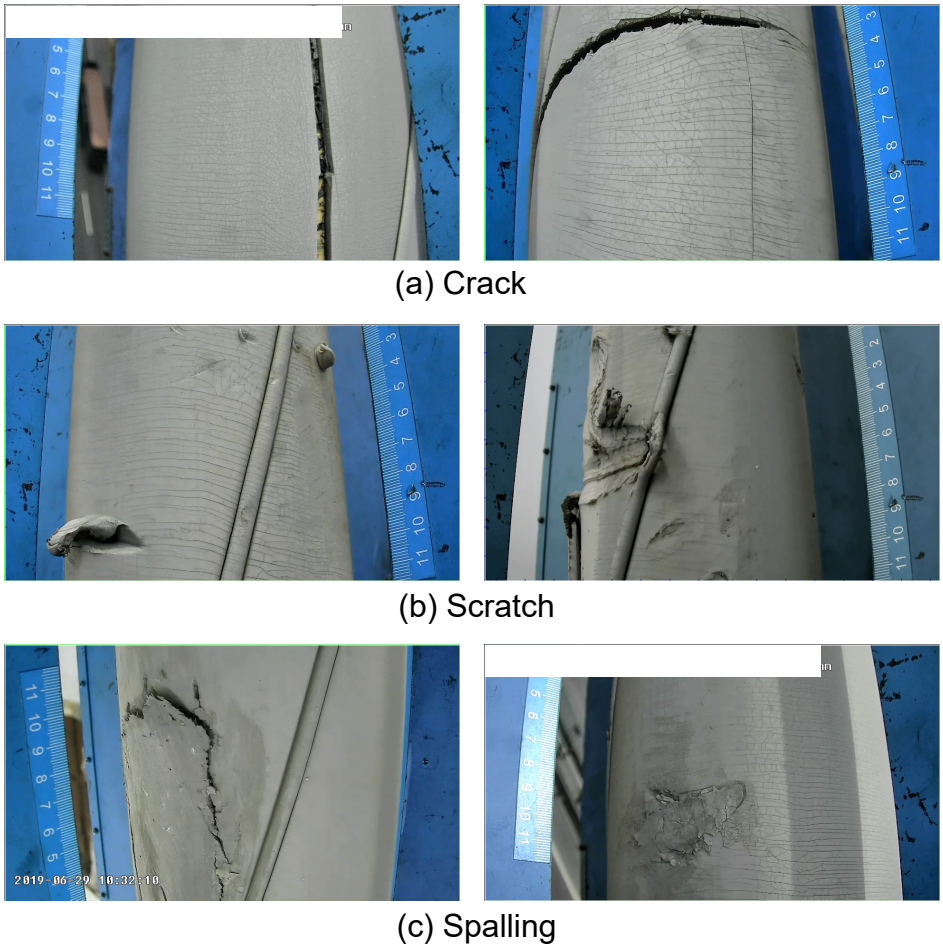

Figure 2. Some examples of surface defects on cables, where helical fillets are designed to suppress rain–wind-induced vibrations.

2.2. Image Segmentation Based on Deep Neural Network

Most previous studies have proposed methods based on image classification or object detection using bounding boxes [27–29]. However, these methods cannot provide precise information regarding defect paths and densities. Therefore, there is a pressing demand for a pixel-based segmentation approach to distinguish defective pixels.

With sufficient training data, supervised learning strategies significantly enhance the capabilities of segmentation models. The Convolutional Neural Network (CNN) framework is the most commonly used approach, as CNNs demonstrate powerful feature extraction capabilities, especially in capturing local features within images [13,30–32]. In particular, Long et al. initially introduced the Fully Convolutional Network (FCN), which can be trained end-to-end, greatly improving accuracy and paving the way for deep learning-based pixel-level semantic segmentation [33]. This advancement assists intelligent systems in grasping spatial relationships or making critical judgments [34]. Shi et al. segmented cracks in steel and rubber bearing corrosion images using VGG-Unet [35]. Despite being affected by the imbalance between foreground and background in the dataset, this encoder–decoder framework was considered one of the most well-designed architectures for segmentation, as it is well suited for recovering multiscale details of target objects [36]. Deng et al. incorporated the Atrous Spatial Pyramid Pooling (ASPP) module into LinkNet, achieved high recognition accuracy for bridge surface structural damage based on a pre-trained encoder, even on small datasets [37]. The network proposed is based on an encoder–decoder framework and utilizes transfer learning from large datasets to quickly deploy into our small domain of defect segmentation. Additionally, composite loss functions were employed to better learn foreground defect features.

Although convolutional networks demonstrate powerful capabilities in extracting local features, their fixed receptive fields limit the ability to correlate features at large scales. In contrast, self-attention mechanisms enrich the representation capability of features based on global texture modeling, effectively compensating for the limitations of convolutions. Self-attention has recently gained favor among many researchers after Transformers became the new state of the art in Natural Language Processing (NLP) [12]. This led to the development of Vision Transformer (ViT) [11], which has been adapted for image segmentation tasks [38–40]. The Swin Transformer confines self-attention within sliding windows, significantly reducing computational costs [41]. He et al. embedded the Swin Transformer into UNet to form a dual encoder, which is used in remote sensing image segmentation, enhancing the accuracy of small-scale ground objects through a spatial interaction module and a feature compression module [42]. However, the increased computational cost is not conducive.

Our proposed model effectively integrates Transformers and convolutional networks. The efficient encoder, developed based on the Transformer, comprehensively extracts potential defect features, while Spatial Reduction Attention (SRA) is introduced to reduce complexity [43]. Subsequently, in the convolution-based decoder, features extracted by the encoder are further refined, ultimately yielding critical bridge cable defect predictions.

3. Method

We propose an encoder–decoder defect image segmentation architecture, Trans-DCN, designed to efficiently achieve comprehensive detection of the defect areas of bridge cables. In this section, we will outline our overall network architecture. The flowchart of Trans-DCN is shown in Figure 3. Before that, it is necessary to provide a brief overview of some relevant techniques used in the network, including Dot Product Self-Attention (DPSA), Deformable Convolution (DCN), Depthwise Separable Convolution (DSC), and SE.

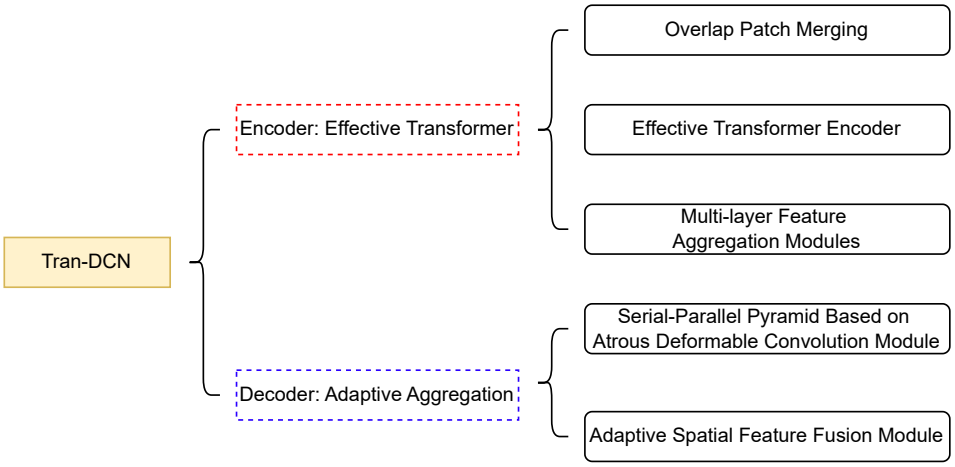

Figure 3. The flowchart of our proposed network Trans-DCN, detailed in Section 3.2.

3.1. Overview of Related Methods

Self-attention mechanism. In a self-attention [12], given an input sequence $s \in \mathbb{R}^{N\times d}$, we obtain $\boldsymbol{Q}$(query), $\boldsymbol{K}$(key) and $\boldsymbol{V}$(value) $\in \mathbb{R}^{N\times D}$ matrices with different spatial relationships by multiplying three learnable mapping matrices W_q, W_k, $W_v \in \mathbb{R}^{d\times D}$ with s. The Dot Product Self-Attention is defined by scaling the dot product of $\boldsymbol{Q}$ and $\boldsymbol{K}$ and passing it through a softmax activation to produce attention weights, which are then applied to the $\boldsymbol{V}$, represented as follows:

$$Attention(Q, K, V) = softmax\left(\frac{Q \cdot K^T}{\sqrt{D}}\right)V \tag{1}$$

where $\sqrt{D}$ represents the scaling parameter, and N is the length of the embedding sequence. Self-attention can be regarded as the weighted average of the input sequence with respect to itself and all other sequences. It aggregates information from all sequences, thereby extracting rich contextual information from the image. Meanwhile, the computational complexity of self-attention is quadratic, $\mathcal{O}(N^2D)$, primarily dependent on the number of input sequences.

The self-attention mechanism has been widely utilized in the field of NLP. ViT is one of the earliest works to apply a pure Transformer to computer vision and demonstrates a powerful large-scale correlation capability of image features, a degree of correlation that is challenging for convolution-based networks to achieve. Given the large-scale variations observed in defect areas of bridge cables, the broad-contextual feature correlation enabled by self-attention can enhance the network's capability to extract relevant information. Therefore, we proposed a multiscale feature alignment-utilizing backbone to serve as the encoder of the model.

Deformable Convolution. For regular 2D convolution in the field of image processing, we first define a receptive field grid R, which represents the deviation from the center of the convolution kernel. Taking a convolutional kernel 3×3 as an example, R with one dilation is as follows: $R = \{(-1,-1),(-1,0),\cdots,(1,1)\}$. For a given input feature map x, after convolutional computation, each element p_o on the output feature map y can be obtained through the following calculation:

$$y(p_o) = \sum_{p_n \in R} w(p_n) \cdot x(p_o + p_n) \tag{2}$$

where q_n enumerates the elements of R, and w is the learnable weight. It is worth noting that bias has not been considered in this formulation.

Compared to attention mechanisms, regular convolution exhibits lower computational complexity but is constrained by certain spatial aggregation capabilities, thereby sacrificing the ability to aggregate broad-contextual information. Deformable Convolution seeks to strike a balance between self-attention and regular convolution by introducing additional learnable offsets [14], as shown in Figure 4a. In Deformable Convolution, the external offset Δp_n is added to the receptive field grid R. Therefore, Deformable Convolution is represented in Equation (2) as follows:

$$y(p_o) = \sum_{p_n \in R} w(p_n) \cdot x(p_o + p_n + \Delta p_n) \tag{3}$$

where Δp_n is learnable from the input feature map and maintains the same number of R. Particularly, the learned Δp_n are floating-point numbers, while the feature map is integer. Bilinear interpolation sampling is required to obtain $x(p_o + p_n + \Delta p_n)$.

Deformable Convolution inherits the efficient capability of regular convolution while also possessing adaptive spatial aggregation capability and dynamic adjustment of the range of perceived features, which is similar to self-attention. We propose a serial–parallel Deformable Convolution module in the decoding stage, which achieves adaptive multiscale spatial feature aggregation and ensures a comprehensive perception of the defects.

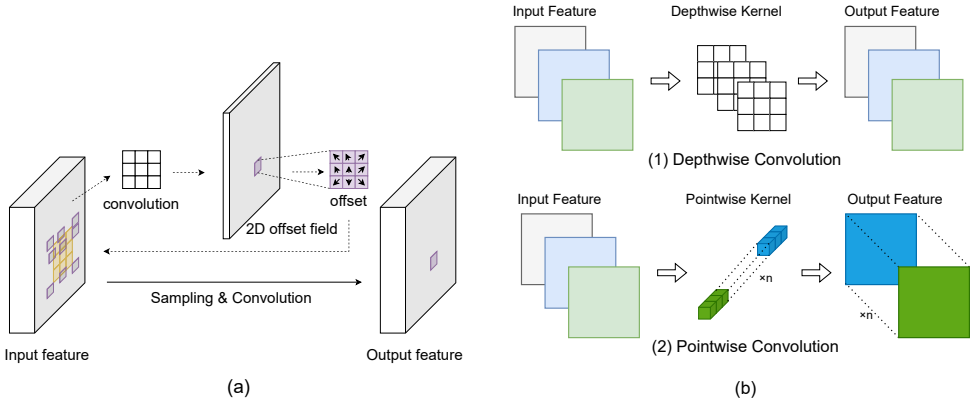

Figure 4. (**a**) Deformable Convolution. The offset field is learned from the input feature. Taking a 3×3 convolution as an example, each element of the offset field contains a dimension of 9×2 elements because the offset is decomposed into a two-dimensional vector. The dotted arrows indicate the calculation of the offset, and the solid arrows indicate the final convolution calculation. (**b**) Depthwise Separable Convolution.

Depthwise Separable Convolution. Depthwise Separable Convolution was first introduced in MobileNet [44]. It factorizes a standard convolution into a depthwise convolution and a pointwise convolution. Specifically, each input feature channel corresponds to a unique kernel in depthwise convolution, ensuring that the convolution operates independently on each channel. Subsequently, information from different channels is linearly combined into a fixed number of channels in 1×1 pointwise convolution, facilitating efficient information exchange and integration between channels, as shown in Figure 4b.

In addition, we also incorporate batch normalization and ReLU activation after each layer, enhancing the convolutional operations' fitting capabilities [45]. Depthwise Separable Convolution is more efficient than standard convolution. Given the kernel size K and output channel size N, we can obtain the ratio of computational complexity to standard convolution as $\frac{1}{N} + \frac{1}{K^2}$. Therefore, we utilize DSC as a substitute for standard convolution to reduce computational costs in our model.

Squeeze-and-Excitation. The SE introduces compression (Squeeze) and excitation (Excitation) operations to establish relationships between channels, enabling the model to adaptively learn the importance of each channel and adjust its contribution in the feature map [46]. Therefore, the model can focus more on important channels, enhancing

the discriminative capability of target feature. The SE mechanism can be represented as follows:

$$SE(F) = F \bigotimes sigmod(FC(ReLU(FC(GP(F))))) \quad (4)$$

where F is the input feature, $\otimes$ denotes the element-wise multiplication operation, FC represents the fully connected layer, GP stands for the global pooling layer, and $sigmod$ and $ReLU$ are non-linear activation layers.

3.2. Proposed Network Architecture

In this paper, we propose a state-of-the-art segmentation model Trans-DCN (Figure 5) that combines the effective feature extraction capability of Transformers as the encoder (Section 3.3), and integrates the adaptive aggregation structure as the decoder (Section 3.4). By establishing an encoder–decoder framework that combines the advantages of self-attention and convolution, we aim to establish a baseline that could accurately identify defect locations on bridge cables.

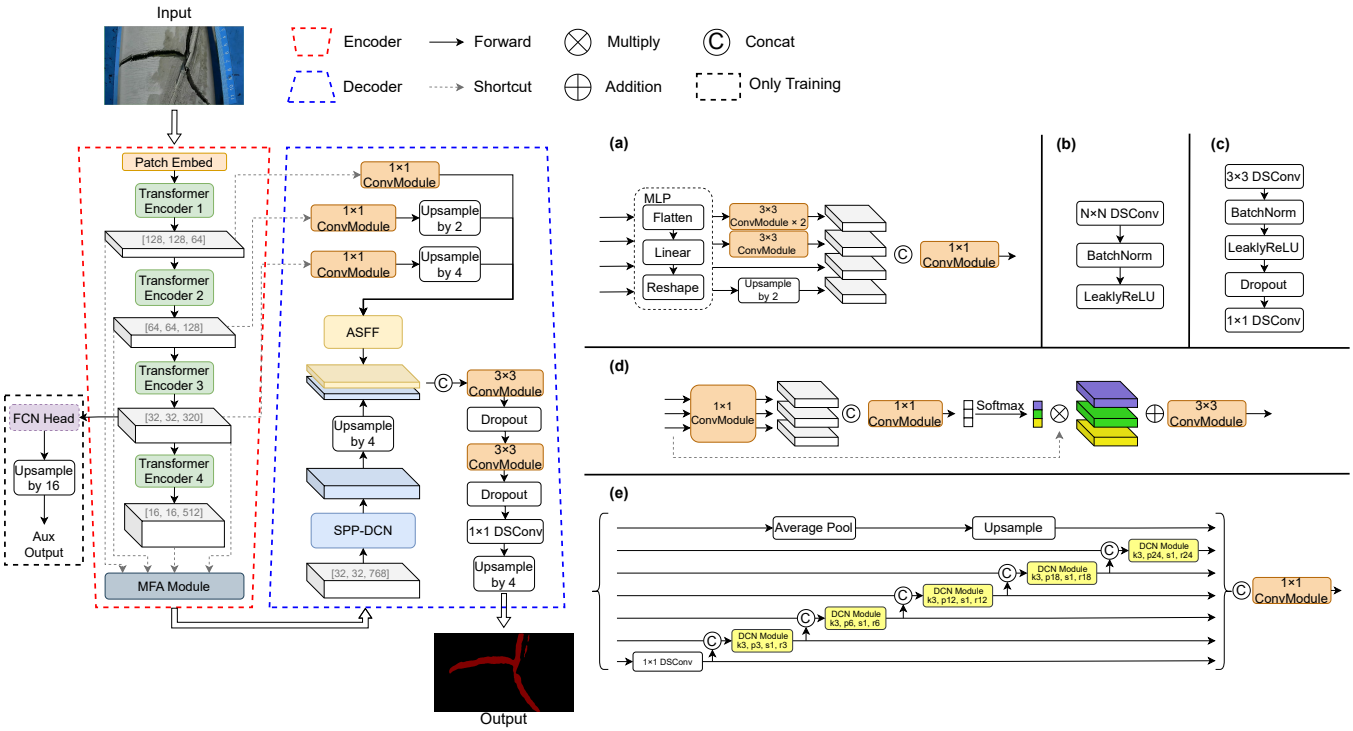

Figure 5. Our proposed network Trans-DCN. (**a**) is the Multi-layer Feature Aggregation (MFA) module, detailed in Section 3.3.3. (**b**) is the basic convolution module. (**c**) is the Fully Convolutional Network (FCN) head. (**d**) is the Adaptive Spatial Feature Fusion (ASFF) module, detailed in Section 3.4.2. (**e**) is the Serial-Parallel pyramid module based on atrous Deformable Convolution (SPP-DCN), detailed in Section 3.4.1. Best viewed with zoom in.

In the encoder, we adopt the fundamental Transformer encoding paradigm [11,41,43] to capture global contextual relationships. Our proposed Transformer encoder comprises four stages without positional encoding (The positional information is contained in the FNN module subsequently). Preceding each encoder, we include a patch merging layer for patch fusion and downsampling (Section 3.3.1). Notably, recognizing the significance of retaining all potentially relevant target features comprehensively to furnish the decoder with ample target feature information, we preserve the output of each Transformer encoder layer and forward them into subsequent Multi-layer Feature Aggregation (MFA) modules through shortcuts (Section 3.3.3).

In the decoder, we SPP-DCN module to receive feature information from the MFA module. This module dynamically adjusts the perception range, enabling comprehensive attention to incorporate local fine-grained details across different scales. Additionally, we employ ASFF module to adaptively fuse the feature maps originating from the first three layers of the Transformer encoder, which maximizes the exploitation of low-level feature information in the encoder (Section 3.4.2).

3.3. Encoder Based on Effective Transformer

The Transformer has significantly advanced the evolution of deep network models in various fields and tasks. Presently, a majority of large models are derived from the evolution of the Transformer. Likewise, we inherit the advantages of self-attention and employ efficient Transformers in the encoding stage.

Overall, the Transformer backbone we use can be divided into four stages (Figure 6a). Each stage includes a patch merging module and multiple stacked Transformer encoders. The patch merging module is responsible for image patch merging and downsampling, while the Transformer encoder module computes self-attention to obtain global contextual information.

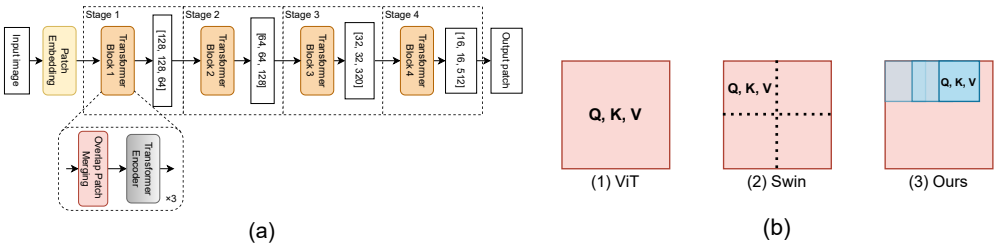

Figure 6. (**a**) Effective Transformer backbone in the encoder. (**b**) Comparing our self-attention relationships with ViT and Swin.

3.3.1. Overlap Patch Merging

ViT divides the input image into patches using fixed divided windows, which poses the issue of no overlap information exchange between windows. Conversely, Swin Transformer utilizes sliding windows to carry contextual window information. Unlike both approaches, we ingeniously utilize convolution to construct patches with overlap relationships. This approach simplified patch construction between overlapping windows while ensuring efficient contextual correlation with local information, as shown in Figure 6b. Specifically, we use stride and kernel size to control the overlap range and also map the patch embedding dimension at the same time.

We use OS to represent the downsampling scale. Taking a stage i ($i = 1, 2, 3, 4$) as an example, we use convolutional kernels with a stride s_i ($s_i > 1$) to establish patches with overlapping relationships on the input feature map $\frac{H}{OS_{i-1}} \times \frac{W}{OS_{i-1}} \times ED_{i-1}$. We design the kernel size k_i and padding p_i to downsample the feature map. Therefore, the size of the output feature map becomes $\frac{H}{OS_i} \times \frac{W}{OS_i} \times ED_i$. The specific parameters for each stage are elaborated on in Table 1. It is worth noting that we use larger convolutional kernels in the first stage to achieve greater downsampling, thereby reducing the computational costs of the backbone.

Table 1. Parameters of the Transformer backbone in the encoder. We got inspiration from SwinTransformer [41] and ResNet [47].

Stage i	Overlap Patch Merging			Efficient Multi-Head Attention		
	Overlap Convolution (k, s, p)	Embedding Dimension (ED)	Output Stride (OS)	Multi-Head Number	Depth	Reduction Ratio
Stage 1	(7, 4, 3)	64	4	1	3	8
Stage 2	(3, 2, 1)	128	8	2	4	4
Stage 3	(3, 2, 1)	320	16	5	6	2
Stage 4	(3, 2, 1)	512	32	8	3	1

3.3.2. Effective Transformer Encoder

The Transformer encoder is tasked with establishing contextual relationships among input patch embeddings. A fully complete Transformer encoder includes the following components: layer normalization, self-attention, feedforward neural network (FNN), and two residual connections, as shown in Figure 7.

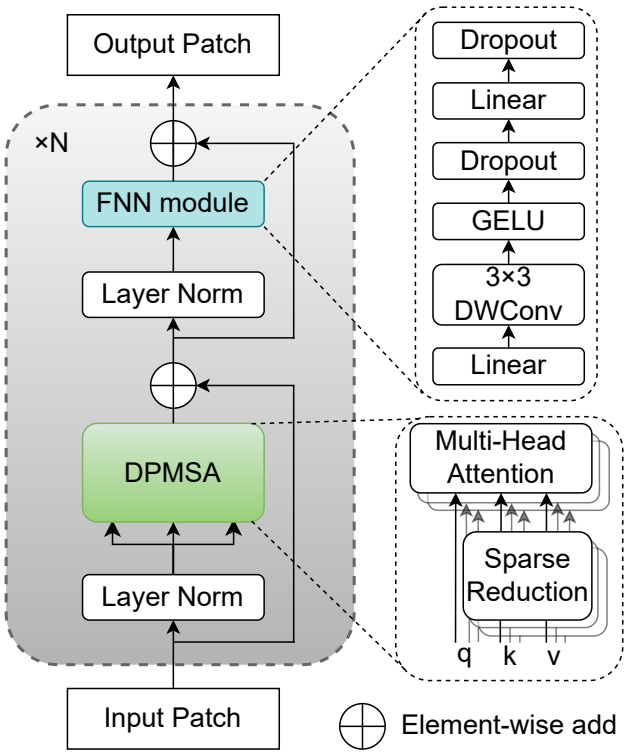

Figure 7. Effective Transformer encoder.

Effective Multi-Head Attention. Due to the quadratic computational complexity of self-attention, it poses a significant computational burden, particularly with larger input image sizes. Inspired by the Pyramid Vision Transformer (PVT) [43], we introduce SRA to alleviate computational costs. Given the matrices K, V, we reduce their dimensions through *Reshape* and *Linear* transformations, represented as follows:

$$\begin{aligned} K', V' &= Reshape\left(\frac{N}{R},\ D \cdot R\right)(K, V) \\ \hat{K}, \hat{V} &= Linear(D \cdot R,\ D)\left(K', V'\right) \end{aligned} \tag{5}$$

where R denotes the reduction ratio. Notably, reduction solely applied to K, V will not change the output dimension of the self-attention, and it decreases the computational complexity of the self-attention from $\mathcal{O}(N^2D)$ to $\mathcal{O}\left(\frac{N^2D}{R}\right)$.

Moreover, to comprehensively capture spatial information with varying emphases and bolster the backbone's generalization, we employ multiple mapping matrices to transform the input sequence into $Q, K, V \in \mathbb{R}^{N \times D_h}$ located in distinct spatial domains, and then compute multi-head self-attention parallelly using Equation (1). Subsequently, we combine the results of multi-head self-attention through mapping, which fuses the contextual relationships from multiple spatial domains, represented as follows:

$$DPMSA(s) = [DPSA_1(s); DPSA_2(s); \cdots; DPSA_h(s)] \cdot W \tag{6}$$

where $W \in \mathbb{R}^{h \cdot D_h \times D}$ is the mapping matrix, and h is the multi-head number. The effective multi-head attention is shown in Figure 7.

FNN module. The feedforward neural network plays a pivotal role in introducing non-linearity to the outputs post self-attention. Given the absence of positional encoding in the Transformer encoder, it can be particularly important to compensate for positional information in FNN. Specifically, we employ an added 3×3 depthwise convolution with zero padding in FNN, which incorporates positional information efficiently [48]. The FNN module could be represented as follows:

$$FNN(s) = MLP(GeLU(DWConv_{3\times3}(MLP(s)))) \tag{7}$$

Overall, the Transformer encoder can be represented as follows:

$$\begin{aligned} s'_l &= DPMSA(LN(s_{l-1})) + s_{l-1} \\ s_l &= FNN\left(LN\left(s'_l\right)\right) + s'_l \end{aligned} \tag{8}$$

where s_l represents the sequence of the l-th layer of the Transformer encoder, and LN represents layer normalization.

3.3.3. MFA Module for Feature Fusion

We employed a multi-hierarchical feature extraction architecture based on the Transformer, where low-level features emphasize fine-grained relationships, while high-level features capture global contextual relationships. To fully utilize both global and local multiscale information, we deploy an MFA module to aggregate results from all Transformer stages, as shown in Figure 5. Specifically, we first map all feature maps through an MLP layer to a unified channel dimension to balance the contribution level of each stage. Subsequently, we use the feature map with an OS of 16 as the reference and align and unify the other feature maps accordingly. This process involves bilinear upsampling the feature map with an OS of 32, and downsampling the feature maps with an OS of 4 and 16. Finally, we merge these four feature maps through concatenation. This module effectively balances efficiency and memory consumption. The details of the structure are shown is Figure 5c.

3.4. Decoder Based on Adaptive Aggregation

In the decoding stage, we aim to focus attention on the target areas by fully utilizing encoding feature information, refining target boundaries, and sharpening segmentation results. To comprehensively correlate the multiscale range in the defect areas, we propose the SPP-DCN module. Additionally, to effectively reuse the information extracted by the encoder, we employ the ASFF module.

3.4.1. Serial0-Parallel Pyramid Based on Atrous Deformable Convolution Module

Inspired by DeepLabv3 [49] and DenseASPP [50], we have designed a multi-branch pyramid structure with a Deformable Convolution module called SPP-DCN, as shown in

Figure 5e. This module connects the MFA module from encoder and dynamically adjusts the perception field of defects efficiently.

In our approach, we use 7 parallel branches in the pyramid. Among them, two are designed to retain the essential original feature information, and one branch initializes the receptive field using a 1×1 Depthwise Separable Convolution. Furthermore, we employ another branch with a global average pooling layer to maximize the receptive field. The remaining 5 branches utilize Deformable Convolutions with varying dilation rates. Inspired by DenseASPP [50], we set the dilation rates to $\{3, 6, 12, 18, 24\}$. It is worth noting that in addition to parallel branches, we also incorporate a serial structure. Specifically, the outputs of branches with Deformable Convolution are stacked with the input of the subsequent branch through concatenation operation. This strategy is aimed to establish cascading dilated convolution relationships, compensating for the loss of local perceptual details with larger dilation rates.

In this module, another improvement is the redesign of the Deformable Convolution module with dilation rates, as depicted in Figure 5a. Firstly, we followed the method proposed by Huang et al. [51], which involves incorporating modulation scalar in Deformable Convolutions to distinguish whether the areas we offset are regions of interest. Secondly, we introduce group convolution in Deformable Convolutions. Specifically, each group has its own sampling offsets, enabling different groups within a convolutional layer to exhibit distinct spatial aggregation patterns, which enriches features for decoding. Therefore, the Deformable Convolutions (Equation (3)) can be enhanced as follows:

$$y(p_o) = \sum_{g \in G} \sum_{p_n \in R} m_{gk} \cdot w_g(p_n) \cdot x_g(p_o + p_n + \Delta p_n) \tag{9}$$

where G is the total number of groups. m_{gk} represents the modulation scalar of the k-th sampling point in the g-th group, and it undergoes softmax normalization along the k dimension. w represents the convolutional learning weights along the g-th group. Both m_{gk} and Δp_n are learned from the input feature map. We introduce dilation rates in Deformable Convolutions to allow p_n to span larger regions while Δp_n achieves minor adjustments, allowing the convolution to to dynamically focus more effectively on distant feature relationships.

Subsequently, we employ channel-wise feature weighting using the SE module, followed by channel mapping through an MLP module, as shown in Figure 8. Furthermore, we integrate a shortcut path to preserve the original features, designed to prevent overfitting.

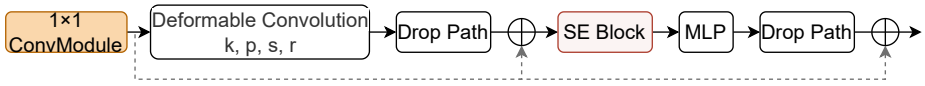

Figure 8. Atrous Deformable Convolution Module, where k is kernel size, p is padding, s is stride, and r is dilation rate.

3.4.2. Adaptive Spatial Feature Fusion Module

To fully capitalize on the defect feature information extracted by the encoder, we establish shortcut connections from the first three stages between the encoder and decoder. These features are particularly beneficial for locating defect areas and refining edge details. The ASFF module is introduced to receive feature maps that have been aligned in channels and scales through separate 1×1 convolutions and upsampling, as shown in Figure 5.

In the ASFF module, we uniformly compress and concatenate the three input features, then obtain the weight parameters for each feature through convolution and softmax.

Finally, we perform weighted summation on the original input features, which can be represented as follows:

$$\begin{aligned} W_1, W_2, W_3 &= softmax(f^1_{64\to1}, f^2_{128\to1}, f^3_{320\to1}) \\ ASFF(f^1, f^2, f^3) &= W_1 \cdot f^{1\to1} + W_2 \cdot f^{2\to1} + W_3 \cdot f^{3\to1} \end{aligned} \tag{10}$$

where f^i represents the feature from the i-th stage, $f_{\cdot\to\cdot}$ represents the feature dimension transformation, and $f^{\cdot\to\cdot}$ represents the feature scale alignment. As described, we dynamically adjust the contributions of the feature maps through the ASFF module, adapting the low-level information required for the decoding stage.

3.5. Loss Function

We train the segmentation network by minimizing a multi-loss function:

$$L_{train} = L_{last\_out} + \gamma L_{aux\_out} \tag{11}$$

where L_{last_out} represents the final output of the decoder, while L_{aux_out} represents the prediction loss comes from the FCN head, as shown in Figure 5. γ represents the balancing weight, which was set at 0.5 in our experiment (the experiment is shown in Section 5.5).

The FCN head is exclusively utilized during training, with its specific structure outlined in Figure 5a. Our purpose is to establish a shortcut to directly supervise the training of the backbone, accelerating the convergence speed of the network.

For each loss, specifically, we supervise them with the linear combination of Cross-Entropy (CE) loss and Dice loss [52], aiming to balance the contributions between positive and negative samples. The composite loss function is formulated as follows:

$$L = \alpha L_{CE} + \beta L_{Dice} \tag{12}$$

where α and β are balancing weights. The CE loss is defined as follows:

$$L_{CE} = -\frac{1}{n}\sum_{i=1}^{n} y_i \cdot ln p_i + (1 - y_i) ln(1 - p_i) \tag{13}$$

where n represents the number of pixels in the image, p_i denotes the predicted classes of the pixel, and y_i denotes the label classes of the pixel. The Dice loss is introduced to balance the training loss where background learning dominates in the CE loss:

$$L_{Dice} = 1 - \frac{\sum_{i=1}^{n} p_i \cdot y_i + \varepsilon}{\sum_{i=1}^{n} p_i + y_i + \varepsilon} - \frac{\sum_{i=1}^{n} (1 - p_i)(1 - y_i) + \varepsilon}{\sum_{i=1}^{n} 2 - p_i - y_i + \varepsilon} \tag{14}$$

where ε is the smooth parameter introduced to prevent overly large changes. It was set to 1 in our experiment.

4. Experiment Results

4.1. Datasets

The RGB image dataset used in the experiments was collected by a cable climbing robot, with detected bridge locations predominantly situated at some large cable-stayed bridges in Shanxi Province, China. We manually conducted a rough screening of images with defects as the dataset and uniformly cropped them to a size of 960×540. The dataset comprises a total of 2304 images, which were divided into training, validation, and test sets according to a ratio of 7:2:1. Subsequently, we manually annotated the ground truth of the defects for supervising network training, which included but was not limited to circumferential cracks, longitudinal cracks, scratches, spalling, and attachments. To ensure the robustness of our network in complex backgrounds, we intentionally retained

elements such as the bridge-climbing robot itself and natural environmental factors, while maintaining an oblique perspective during image capture, as shown in Figure 9.

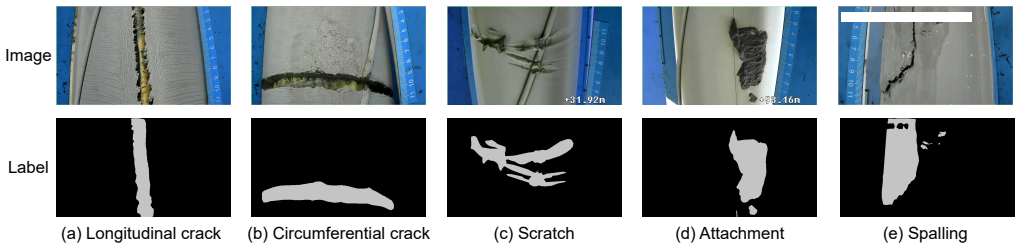

Figure 9. Bridge cable datasets. Particular emphasis was given to showcasing some types of defects along with their labels.

To avoid over-fitting, it is crucial to mention that before training the model, we randomly cropped images patch to a size of 512×512, and performed common data augmentations to enhance the diversity of input shapes, including but not limited to Random Horizontal Flip, Random Scale Crop, Random Gaussian Blur, and Random Rotate.

4.2. Evaluation Metrics

To report the performance of bridge cable defect segmentation, we employed the following evaluation metrics: accuracy (Acc), mean Intersection over Union (mIoU), and F1 score.

1. Acc indicates the agreement between the predicted class and the ground truth labels for all pixels, and it can be expressed as follows:

$$\text{Acc} = \frac{N_{TP} + N_{TN}}{N_{TP} + N_{FP} + N_{FN} + N_{TN}} \tag{15}$$

   where N_{TP} is the number of true positives, N_{FP} is the number of false positives, N_{FN} is the number of false negatives, and N_{TN} is the number of true negative.
2. mIoU indicates the mean Intersection over Union (IoU) between the predicted class and the ground truth label, which can be expressed as follows:

$$\text{mIoU} = \frac{1}{n+1} \sum_{i=0}^{n} \frac{N_{TP}}{N_{TP} + N_{FP} + N_{FN}} \tag{16}$$

   where n represents the number of classes.
3. F1 score is a harmonic mean of precision and recall, and it can be calculated as follows:

$$\text{F1} = \frac{2 \times Precision \times Recall}{Precision \times Recall} \tag{17}$$

   where

$$\begin{aligned} \text{Precision} &= \frac{N_{TP}}{N_{TP} + N_{FP}} \\ \text{Recall} &= \frac{N_{TP}}{N_{TP} + N_{FN}} \end{aligned} \tag{18}$$

The metrics mentioned range from 0 to 1, where values closer to 1 indicate higher segmentation accuracy.

4.3. Implementation Details

We implemented our methods on Pytorch 1.8.0 and conducted training on a workstation equipped with four Nvidia RTX 3090 GPUs. SyncBatchNorm was applied before each weight layer in our implementation for parallel training across multiple GPUs. For

training, we determined the optimal optimizer parameters through cross-validation. The specific experimental results are presented in Appendix A. The best parameter settings identified were to use the SGD optimizer with an initial momentum of 0.9 and a weight decay of 5×10^{-4}. The learning rate was scheduled using a polynomial exponential decay, starting with an initial value of 0.007.

We pre-trained the backbone on the ImageNet-1K dataset and randomly initialize the remaining structure. During fine-tuning with the bridge cable defect dataset, we trained the models for 100 epochs with a batch size of 15. Subsequently, we selected the model with the best mIoU for testing.

4.4. Comparison with Regard to State-of-the-Art Models

In this section, we evaluate the segmentation performance of Trans-DCN in detecting bridge cable defects through a series of experiments. We compare Trans-DCN with various baselines, such as DeepLabv3+, DANet, SegNeXt, ViT-CoMer, and others, employing different backbones. We used cross-validation experiments to determine the optimal training parameters, ensuring that these models achieve the best performance on the bridge cable defect dataset we established. The optimal training parameters identified are detailed in Appendix B. Those segmentation results are shown in Table 2. It is worth noting that, along with evaluation metrics (including Acc, mIoU, and F1 score), we also provide the parameter count and computational cost of each model, which is because segmentation accuracy is influenced by these factors. Our efficient model aims to minimize the parameter count and computational cost while ensuring excellent evaluation. We selected some challenging cases for model prediction, and the visualization of the segmented defect area is presented in Figure 10.

Table 2. Comparison with state-of-the-art methods. ↑ represents a better segmentation performance, ↓ represents lower computing requirements.

Method	Model Architecture	Params. (M) ↓	GFlops ↓	Evaluation ↑		
				Acc (%)	mIoU (%)	F1-Score (%)
Convolution-based	U-Net [53]	13.39	124.49	96.33	78.21	74.25
	FCN (VGG-16) [33]	35.31	148.53	96.90	79.12	76.31
	SegNet (VGG-16) [54]	29.44	160.68	97.34	81.41	79.25
	PSPNet (ResNet50) [55]	46.71	59.21	97.03	78.26	74.45
	DeepLabv3 (ResNet50) [49]	41.99	173.79	97.12	78.69	74.98
	DeepLabv3+ (MobileNet-V2) [56]	41.83	33.75	97.02	78.90	76.17
	DeepLabv3+ (ResNet101) [56]	74.87	82.88	97.09	80.72	78.32
	DenseASPP (DenseNet121) [50]	9.20	43.19	96.79	75.67	69.33
	DANet (ResNet101) [57]	66.56	283.44	96.86	75.61	69.06
	LR-ASPP (MobileNet-V3) [58]	30.22	20.07	96.57	76.93	72.99
	SegNeXt (Base) [59]	29.91	41.55	94.66	65.34	56.13
Transformer-based	SegFormer (B2) [60]	47.22	71.36	97.31	81.47	79.36
	Swin-Unet (Tiny) [61]	27.17	5.92	96.71	78.02	74.61
	Bi-Former (Base) [62]	56.81	91.10	95.12	68.02	58.25
	ViT-CoMer (Small) [63]	60.50	1194.16	96.70	77.32	73.49
	Trans-DCN (ours)	44.96	52.03	**97.49**	**82.63**	**80.89**

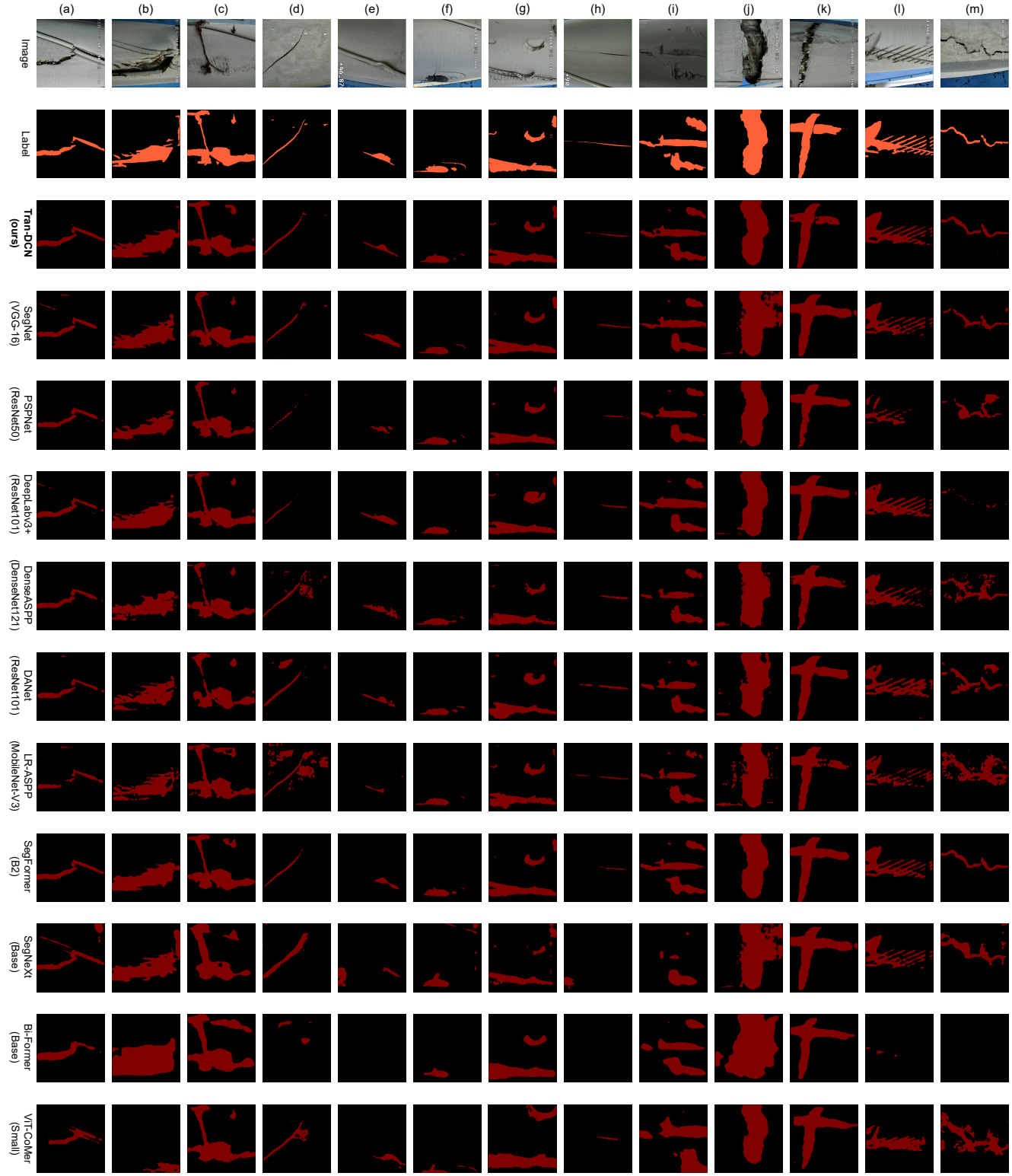

Figure 10. Segmentation results compared to state-of-the-art models. (**a**,**d**,**h**,**m**) represent strip defects, (**l**) represents low illumination conditions, (**e**) depicts minor scratches, (**b**,**j**) illustrate severe defects, while (**c**,**f**,**g**,**i**,**k**) examples show mixed defect conditions. Best viewed with zoom in.

As shown in Table 2, our model, employing with effective Transformer as the encoder and adaptive aggregation as the decoder, achieves a performance of 82.63% in mIoU, attaining state-of-the-art performance among the baseline models. Compared to the latest baseline (ViT-CoMer), our model demonstrates an accuracy improvement of 5.31%. Furthermore, compared to the state-of-the-art baseline (SegFormer), our accuracy improved by 1.16%. Our model exhibits a significant improvement and outperforms other baselines in defect extraction by dynamically adjusting the perceived field and promptly adjusting the utilization of backbone features. Notably, our model's parameter count and computational cost remain lower, indicating that our proposed method is not only high-performing but also efficient. Specifically, our parameter count and computational cost are almost as same as the PSPNet baseline, but our accuracy has been improved by 4.37% (mIoU). Moreover, compared with ViT-CoMer using the UperNet method, our proposed method achieves a parameter reduction of 15.34 M and a computation reduction of 1096 GFlops. This highlights the significant efficient improvement achieved by using an effective Transformer and Depthwise Separable Convolution fully.

Benefiting from the powerful Transformer serving as the backbone for the full contextual feature correlation mechanism, our method excels in detecting large-scale defect distributions, as shown in Figure 10b,c,g,k. Our method achieved defect predictions closest to the ground truth compared to other baselines. Taking Figure 10b,c as examples, our method has a comprehensive perception of defects in the image, effectively capturing even the smallest local damages compare to convolutional models represented by SegNet. DenseASPP and DANet also display shortcomings in predicting long cracks, resulting in fractured segmentation areas. Transformer models represented by Bi-Former tend to segment areas exceeding the boundaries, while ViT-CoMer shows incorrect segmentation. Additionally, in Figure 10j, SegNet, LR-ASPP, and Bi-Former show more false positives, mainly due to the presence of interfering textures outside the circumferential crack. However, our method, by fully considering multi-level features, was able to suppress this noise and preserve the integrity of the target.

Additionally, thanks to the well-designed adaptive Deformable Convolution and dynamic spatial aggregation module, our method accurately predicts strips defects and maintains complete edges, as shown in Figure 10a,d,h,m. Models based on fixed convolution patterns inevitably exhibit discontinuities when faced with elongated features, as seen in PSPNet and DANet (particularly highlighted in Figure 10d,h,m). Models based on ASPP, such as LR-ASPP and DenseASPP, encounter difficulties in aggregating sufficient spatial information when dealing with this type of challenge, as atrous convolutions lose their advantage in strip targets. Although our approach also involves atrous convolutions, it promptly corrects unnecessary expansions through a combination with Deformable Convolutions. Transformer-based models, such as Bi-Former and ViT-CoMer, still perform disappointingly in this scenario. The SegFormer baseline performs similarly to ours, but there is a significant difference observed in Figure 10h.

Meanwhile, we intentionally select defect images featuring shallow scratches or captured in environments with low or uneven lighting conditions to illustrate the robustness of our method under challenging conditions, as shown in Figure 10e,i. The results demonstrate that our model is capable of fully exploring potential defect information, with sharp edges observed. However, some false positives are present simultaneously. The prediction is incomplete in Figure 10e, while SegFormer performs satisfactorily in Figure 10i by exploring complete information at multiple granularities.

In most cases in Figure 10, our detection accurately captures the area of defects adequately. Specifically, Figure 10a,d,h,j,l highlight the strong segmentation integrity of our model. Therefore, our proposed network sets a new state-of-art performance in bridge defect segmentation.

5. Analysis

5.1. Ablation Study

In this section, we introduce ablation experiments, wherein MFA (Section 3.3.3), SPP-DCN (Section 3.4.1), ASFF (Section 3.4.2), and Aux-Loss (Section 3.5) are included as ablation units. The results of these experiments are shown in Table 3. Specifically, in the model without using the MFA module, only the output of the last stage is utilized in the decoder. In the model without the SPP-DCN module, a regular convolution is directly employed for the connection. In the model without the ASFF module, the shortcut is directly removed.

Table 3. Ablation study. ↑ represents a better segmentation performance.

Model	MFA	SPP-DCN	ASFF	Aux-Loss	Evaluation ↑		
					Acc (%)	mIoU (%)	F1-Score (%)
1	-	-	-	-	95.55	77.12	70.23
2	✓	-	-	-	97.13	80.29	77.89
3	✓	-	-	✓	97.14	80.48	78.03
4	✓	✓	-	✓	97.35	81.42	79.24
5	✓	-	✓	✓	97.30	81.14	78.87
6	✓	✓	✓	✓	**97.49**	**82.63**	**80.89**

From Table 3, it is evident that the accuracy of our model can be increased by up to 5.5% (mIoU) with our improvements. Specifically, solely adding the MFA module resulted in a 3.1% increase in mIoU, indicating the necessity of fully integrating multiscale feature maps in the encoder. Subsequently, incorporating auxiliary training loss led to a slight improvement in mIoU during validation. Furthermore, the cross-validation model (4–5), which integrates the SPP-DCN module or ASFF module, exhibited a 1% or 0.7% increase in mIoU, respectively, highlighting the effectiveness of each proposed module in the model. Additionally, captured from Figure 11, it can be observed that removing each effective module causes a decrease in the validation accuracy of our model. Overall, each well-designed module in our model has proven to be indispensable, resulting in a final mIoU score of 82.63%.

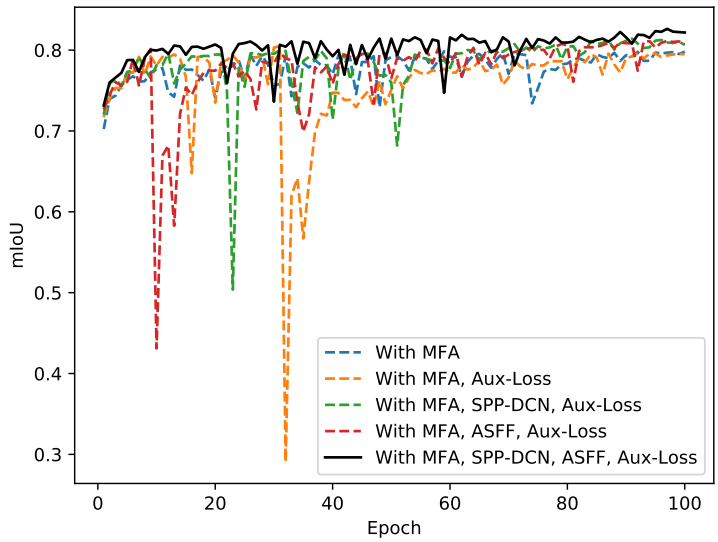

Figure 11. The training evolution curve based on mIoU, where the variables are the units of the ablation study.

5.2. Influence of Transformer Backbone

In this section, we compare and evaluate the segmentation accuracy and parameter count using different backbones to demonstrate the performance of our proposed method. We selected several convolution-based and Transformer-based backbones, using their default parameters from the officially provided codes and enabling pre-training. It is worth noting that for backbones with multiple versions, we attempted to match them with parameter counts comparable to our proposed method. Their version and the experimental results are shown in Table 4.

Table 4. Comparison with different backbones in the encoder. ↑ represents a better segmentation performance, ↓ represents lower computing requirements.

Method	Backbone	Params. (M) ↓	Evaluation ↑		
			Acc (%)	mIoU (%)	F1-Score (%)
Convolution-based	ResNet-101 [47]	42.5	96.37	78.70	73.69
	ResNeXt-101 (32 × 8d) [64]	86.74	96.60	79.93	76.72
	Xception [56]	37.87	89.32	57.16	48.66
	MobileNet-V2 [65]	15.4	96.96	75.14	66.91
	ConvNeXtV2-B [66]	88.72	93.37	65.35	55.27
Transformer-based	Swin Transformer-T [41]	96.52	93.62	76.76	72.81
	Swin TransformerV2-T [67]	27.58	95.94	69.00	59.86
	ViT-CoMer-S [63]	37.34	96.90	78.98	75.94
	Proposed method	24.2	**97.49**	**82.63**	**80.89**

Our proposed method, compared to the latest Transformer-based backbone ViT-CoMer, achieves an improvement of 3.6% in mIoU, with a parameter reduction of 13 M. Compared to the state-of-the-art convolution-based backbone ResNeXt-101, our method still achieves a 2.6% mIoU improvement, with a parameter reduction of 62.54 M, fully highlighting the advantages of our method. Additionally, our method outperforms the backbone MobileNet-V2 by 7.5%, with an increase of only 8.8 M parameters. Therefore, our method stands out as the most cost-effective choice.

Additionally, we conducted experiments on the embedding dimensions in the backbone. We compared the segmentation accuracy of the original embedding dimensions in ResNet [47] with our optimized dimensions. The experimental data are presented in Table 5. It is noteworthy that the multi-head number in the multi-head attention mechanism varies with the embedding dimensions. We ensured that every 64 dimensions correspond to one head in the self-attention calculation, thereby guaranteeing richer spatial learning patterns.

Table 5. Accuracy, parameter count and computational cost as a function of the embedding dimension of the Transformer backbone. ↑ represents a better segmentation performance, ↓ represents lower computing requirements.

Stage *i*	Embedding Dimension	Multi-Head Number	Params. (M) ↓	GFlops ↓	Evaluation ↑		
					Acc (%)	mIoU (%)	F1-Score (%)
1	64	1	92.06	69.72	97.41	81.89	79.82
2	256	4					
3	512	8					
4	1024	16					
1	64	1	24.2	39.44	**97.49**	**82.63**	**80.89**
2	128	2					
3	320	5					
4	512	8					

As shown in the Table 5, our optimized embedding dimensions not only reduce the parameter count by 73% but also improve segmentation accuracy, which indicates that the dimensions we designed basically meet the efficient requirements for cable defect segmentation.

5.3. Influence of SPP-DCN

In this section, we individually evaluate the effectiveness of the internal structure of the SPP-DCN module, as shown in Figure 5e. We conducted an ablation study specifically designed for this module, wherein the shortcut, MLP, and SE modules serve as ablation units. The experimental results are presented in Table 6.

Table 6. Ablation study based on SPP-DCN. ↑ represents a better segmentation performance.

Model	Shortcut	MLP	SE	Evaluation ↑		
				Acc (%)	mIoU (%)	F1-Score (%)
1	-	-	-	97.29	81.03	80.25
2	-	✓	✓	**97.50**	82.47	80.65
3	✓	-	-	97.38	80.10	80.22
4	✓	-	✓	97.46	81.14	80.21
5	✓	✓	-	97.37	82.16	80.31
6	✓	✓	✓	97.49	**82.63**	**80.89**

From the table, it is notable that simply adding a shortcut after the atrous Deformable Convolution (model 3) does not yield a positive effect (mIoU decreases by 1%). This is because the features do not pass through subsequent fully connected and channel-attention layers, rendering the residual connection ineffective. Conversely, adding both MLP and SE modules after the convolution simultaneously (model 2) can increase the mIoU by 1.5%, which demonstrates that adding both modules can effectively enhance the adaptively perceived defect information. Simultaneously, the cross-validation (model 4–5), which integrates the MLP module or SE module, exhibited a similar 1.1% increase in mIoU. Consequently, the conclusion can be drawn that SPP-DCN's capabilities can be fully enhanced by combining all three methods (model 6), producing the best segmentation outcomes (1.6% improvement).

Moreover, we provide the visualization of the receptive fields compared to the ASPP module, as shown in Figure 12. Our receptive fields in SPP-DCN are more uniform and flexible, which is advantageous for detecting scattered defects. The examples on the right side well demonstrate that our design can better adjust the attention area, effectively reducing the possibility of mis-segmentation.

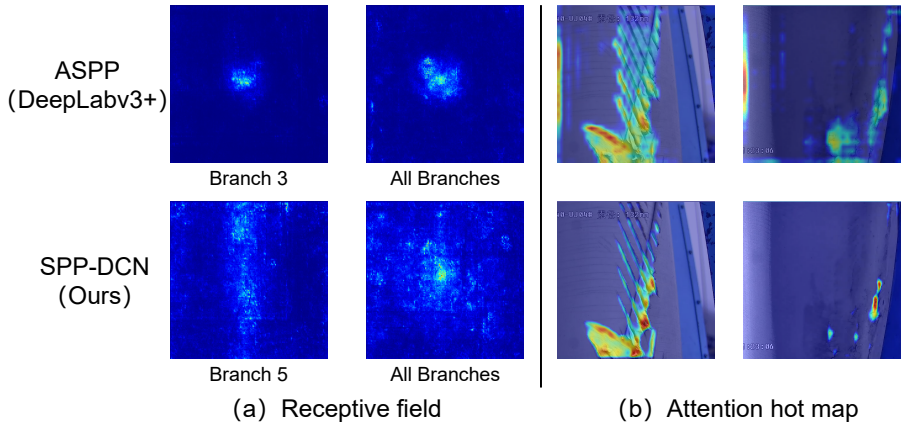

Figure 12. SPP-DCN vs. ASPP. (**a**) Visualization of the receptive field based on the pyramid module as we choose the large dilation rate branch, with yellow regions indicating higher attention. (**b**) Attention visualizations on the defect image. The brightness corresponds to the attention level, with red regions indicating higher attention.

In addition, we conducted an experiment with branches with different dilation rates, as shown in Table 7. The experimental results indicate that setting the dilation rates to $\{3, 6, 12, 18, 24\}$ achieved the highest segmentation mIoU, which is 6.7% higher than the dilation rates of $\{6, 12, 18\}$. Our branch setup provides a maximum receptive field of 128, enabling better perception of defects over a large scale while ensuring the detection of small-scale features. Consequently, this design is more effective in handling the varying feature of cable defects.

Table 7. Accuracy as a function of the number of branches in the SPP-DCN. RF represents receptive field. ↑ represents a better segmentation performance.

Branches	Max RF	Evaluation ↑		
		Acc (%)	mIoU (%)	F1-Score (%)
(6, 12, 18)	73	96.29	75.95	71.62
(3, 6, 12, 18)	79	96.44	78.21	75.01
(6, 12, 18, 24)	122	96.76	79.94	77.25
(3, 6, 12, 18, 24)	128	**97.49**	**82.63**	**80.89**

5.4. Influence of ASFF

In this section, we analyze the impact of the channel alignment scale within the ASFF on the performance. As illustrated in the network architecture (Figure 5), we fully exploit the low-level features from the encoder through shortcuts and perform adaptive aggregation in the decoder. Before softmax calculation, it is crucial to carefully determine the channel alignment of each feature map, as it influences contributions post-concatenation with the subsequent SPP-DCN module. We directly consider the channel dimension of the feature maps in the three shortcuts, and the experimental results are detailed in Table 8.

Table 8. Accuracy, parameter count and computational cost as a function of the channel alignment dimension of the ASFF. ↑ represents a better segmentation performance, ↓ represents lower computing requirements.

Model	Channel	Params. (M) ↓	GFlops ↓	Evaluation ↑		
				Acc (%)	mIoU (%)	F1-Score (%)
1	64	0.06	0.7	97.47	82.12	80.16
2	128	0.2	2.64	**97.49**	**82.63**	**80.89**
3	320	0.99	15.43	97.33	81.80	79.81

In Table 8, we show performance alongside parameter count and computational cost, varying with channel dimension. Notably, setting $C = 128$ yields commendable performance while maintaining competitive computational cost. As the dimension increases, there is a decline in mIoU, coupled with a substantial rise in computational cost. Conversely, reducing the dimension compromises performance, rendering it unsatisfactory.

5.5. Influence of Composite Loss Function

In this section, we conducted experiments to evaluate the weight of the composite loss function during model training (Section 3.5). We focused on understanding the impact of the weight ratio between the CE loss α and the Dice loss β on both training accuracy and convergence speed. Notably, we employed a step size of 0.5 for the CE loss and 0.25 for the Dice loss, as larger Dice loss values may induce training instability. Additionally, we also included γ as a variable in the experiment. The experimental results are detailed in the Table 9.

In Table 9, we can observe that the composite loss function with $\alpha = 2$ and $\beta = 0.5$ achieves the highest performance with 82.63% in mIoU (1.23% improvement). Notably, altering the values of α and β results in a decline in the validation segmentation performance, and altering the γ will have a very slightly impact on the evaluation. Furthermore,

Figure 13 illustrates that the composite loss function with $\alpha = 2$ and $\beta = 0.5$ exhibits superior convergence and optimization capabilities compared to other combinations.

Table 9. Accuracy as a function of difference ratio of composite loss functions. ↑ represents a better segmentation performance.

Ratio			Evaluation ↑		
α	β	γ	Acc (%)	mIoU (%)	F1-Score (%)
1	0	0.5	97.28	81.40	79.27
1	0.5	0.5	97.37	81.71	79.65
1.5	0.5	0.5	97.39	82.09	80.19
2	0.5	0.5	97.49	**82.63**	**80.89**
2.5	0.5	0.5	97.42	82.41	80.36
3	0.5	0.5	97.37	81.95	80.01
2	0.25	0.5	97.36	81.94	80.00
2	0.75	0.5	97.44	82.29	80.44
2	0.5	0	97.44	82.06	80.10
2	0.5	0.25	97.44	82.17	80.27
2	0.5	0.75	**97.55**	82.46	80.60
2	0.5	1	97.46	82.20	80.30

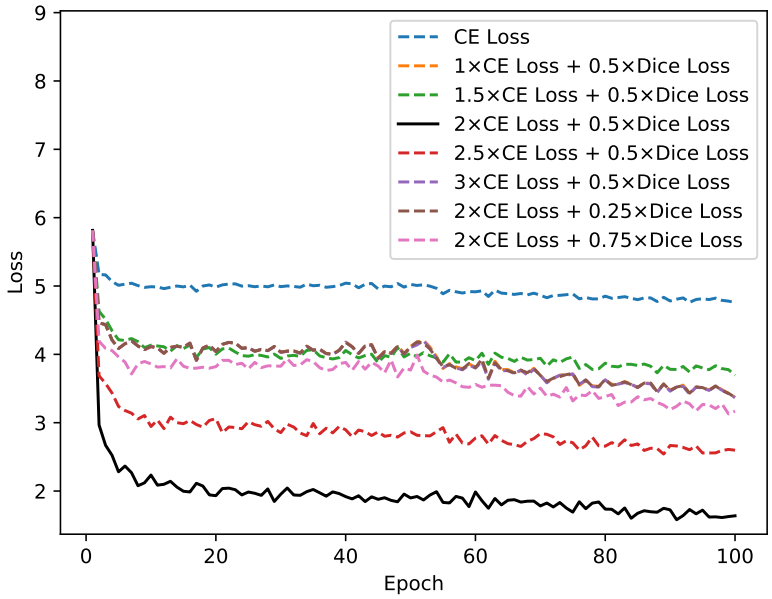

Figure 13. The training loss based on different composite loss functions. To ensure clarity, we uniformly aligned the origin despite the disparate scales. Our purpose was to observe the changes in the value range and convergence speed across different configurations.

6. Discussion

Our work serves as foundational research and opens up various possibilities for future endeavors in the field of cable defect detection such as precise defect localization and measurement. However, it is important to acknowledge the limitations of our work. For instance, the presence of extremely small cracks lacking proper annotation may compromise the network's detection capabilities in this regard. Additionally, imbalance in bridge cable defect categories within our dataset poses a challenge for multi-class semantic segmentation. To address these challenges, our future endeavors will concentrate on few-shot semantic segmentation models capable of efficiently detecting relevant defects with minimal reliance on a large number of samples for specific classes. Furthermore, we are committed to expanding our dataset to encompass a broader range of bridge cable defect scenarios, thereby providing a robust benchmark for training and evaluating our models. We are interested in determining whether the proposed network can maintain its excellent

extraction capabilities when applied to defect domains beyond bridge cables. In future work, we will evaluate our method with more diverse data resources to test its robustness. Furthermore, we will investigate the network's adaptability to other data input sources (e.g., infrared sensors), exploring its versatility and potential for broader applications.

7. Conclusions

This paper presents an efficient and practical bridge cable defect segmentation network, Trans-DCN, which fully integrates Transformers and convolutions, achieving state-of-the-art defect segmentation performance within the encoder–decoder framework. The proposed model employs an efficient Transformer as the backbone, effectively leveraging features extracted by self-attention, which contains both local fine-grained and global contextual information. Furthermore, multiple-layer features are judiciously reused in the encoder. In the decoder, SPP-DCN is designed to dynamically adjust the perception range, concentrating the attention based on the distribution characteristics of the defects. Additionally, the model incorporated several shortcuts between the encoder and decoder, and adaptive fusion of multi-layer features was used in the decoder. Moreover, the introduction of Depthwise Separable Convolutions and enhancements to the loss function serve to improve the efficiency and convergence speed of the model.

For the detection of bridge cable defects, we collected actual captured images dataset for conducting experiments. The experimental results emphasize that in comparison to DeeplabV3+, SegFormer, Swin-Unet, and other baseline models, our model achieved state-of-the-art performance in detecting bridge cable defects. Specifically, our model achieves an Acc of 97.49%, an mIoU of 82.63%, and an F1 score of 85%. Simultaneously, we conducted multiple experiments to demonstrate the effectiveness of each module within our model.

Our designed network imposes no specific requirements on the characteristics of the data, needing only RGB images, and no preprocessing is required to remove irrelevant background parts. This flexibility means the data source can be from cable-climbing robots, UAVs, or manual collection. Our proposed method is well suited for offline, fast, and automated analysis of bridge cable surface defects, providing a cost-effective solution.

Author Contributions: Conceptualization, Z.H., X.D. and B.G.; methodology, Z.H. and X.D.; formal analysis, Z.H. and B.G.; validation, Z.H. and W.G.; dataset, B.G. and Z.H.; writing—original draft, Z.H. and X.D.; writing—review and editing, B.G. and W.G.; funding acquisition, B.G.; resources, X.M.; supervision, X.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Natural Science Foundation of China (Grant No. U21A20139).

Data Availability Statement: The dataset for this article is not publicly available due to confidentiality and intellectual property rights restrictions.

Acknowledgments: The authors sincerely acknowledge the hardware technical support provided by Huazhong University of Science and Technology, China.

Conflicts of Interest: Author Xing Min was employed by the Guangdong Metro Design Research Institute Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Appendix A

In Appendix A, we introduce the optimal training parameters determined through cross-validation experiments. Specifically, we validated three parameters of the optimizer mentioned in Section 4.3, including momentum, initial learning rate, and weight decay. Since the backbone has been pre-trained, we primarily used the SGD optimizer to fine-tune the entire network. The results of the optimizer experiments are presented in Table A1.

We can clearly see that setting the momentum to 0.9, the initial learning rate to 0.007, and the weight decay to 0.005 achieves the best training results. Increasing the momentum excessively results in large gradient calculation inertia, making it difficult for the network

to converge. Similarly, increasing the weight decay too much leads to under-fitting of the model, negatively affecting the training process. The optimal initial learning rate should be around 0.005–0.007. Additionally, we conducted experiments using the Adam optimizer. The results indicated that for our transfer learning task, the SGD optimizer with momentum and the polynomial learning rate strategy achieves better network convergence.

We also plotted the changes in the loss function value during training, as shown in Figures A1–A3.

Table A1. Training parameter experiments. ↑ represents a better segmentation performance.

Optimizer	Momentum	Learning Rate	Weight Decay	Acc (%) ↑	F1-Score ↑	mIoU ↑
SGD	0.8	0.007	0.005	97.43	79.96	81.96
	0.85			97.44	80.16	82.10
	0.9			**97.49**	**80.89**	**82.63**
	0.95			97.41	79.53	81.68
	0.99			96.44	72.35	76.47
SGD	0.9	0.007	0.005	96.25	71.01	75.56
			0.001	97.03	77.88	80.33
			0.0005	**97.49**	**80.89**	**82.63**
			0.0001	97.48	80.51	82.36
SGD	0.009	0.001	0.005	96.77	77.62	80.01
		0.003		97.43	80.08	82.05
		0.005		97.45	80.11	82.25
		0.007		**97.49**	**80.89**	**82.63**
		0.01		96.12	68.95	74.28
Adam	0.9	0.007	0.0005	96.52	73.20	76.76

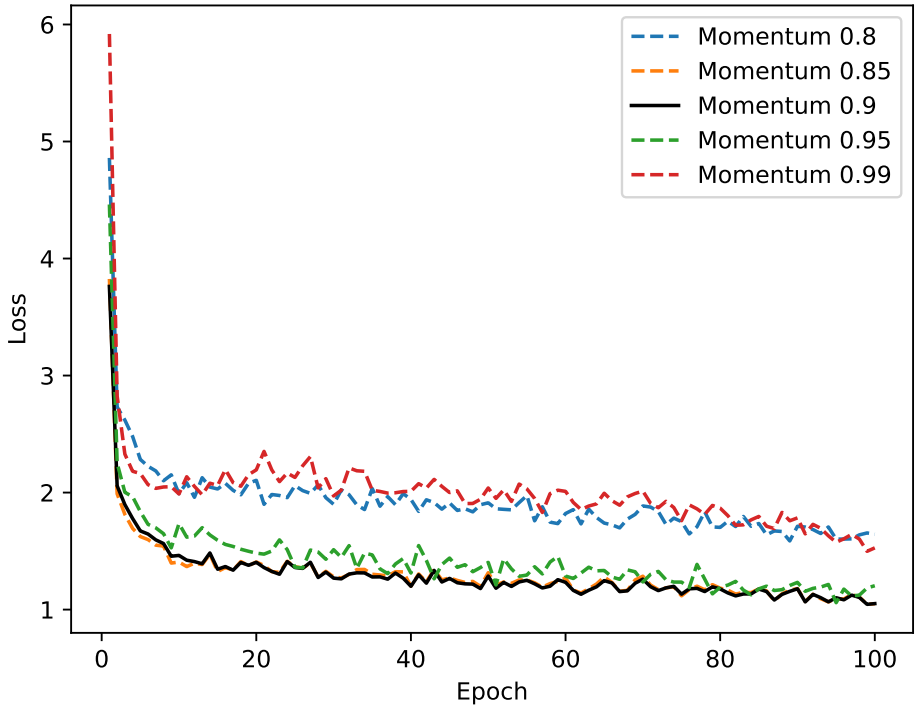

Figure A1. The line chart compares the changes in loss values with increasing epochs under different momentum conditions.

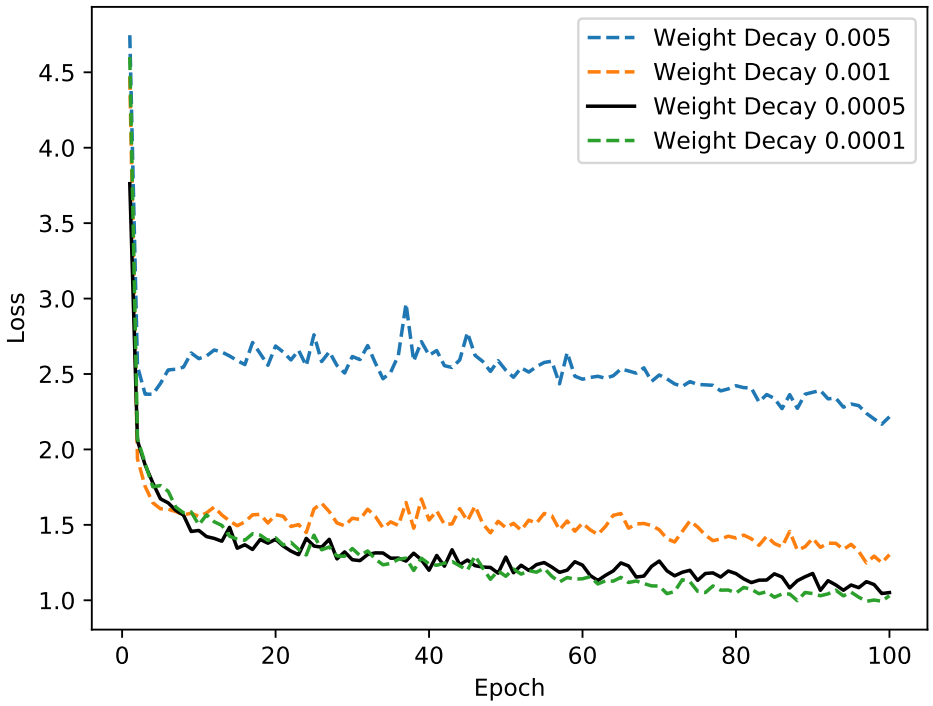

Figure A2. The line chart compares the changes in loss values with increasing epochs under different weight decay conditions.

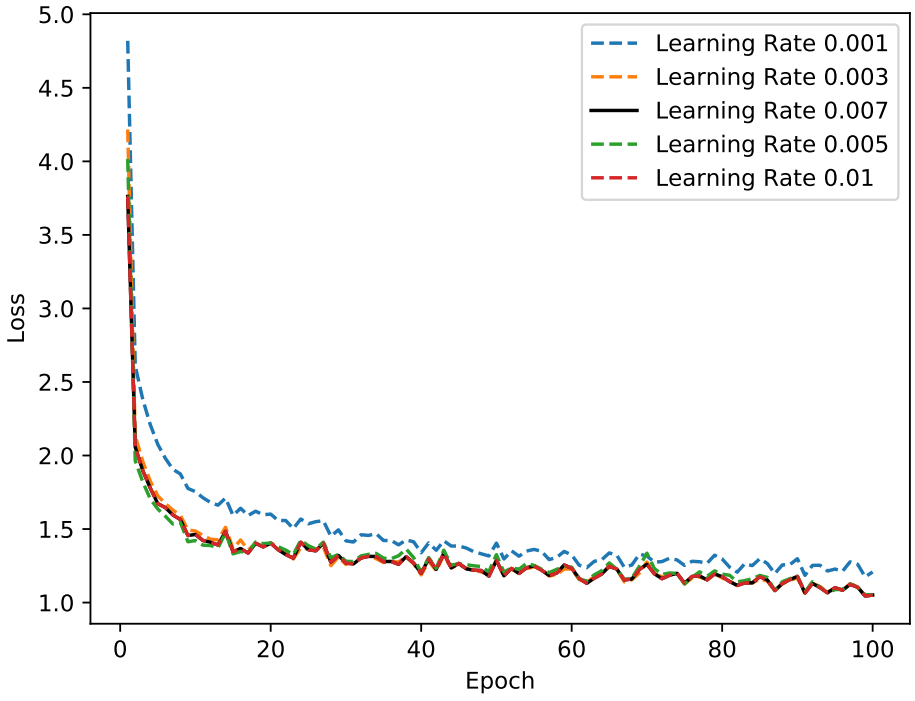

Figure A3. The line chart compares the changes in loss values with increasing epochs under different initial value of learning rate conditions.

Appendix B

In Appendix B, we determine the best segmentation performance of the state-of-the-art models we compared through cross-validation experiments, and we list their training parameter settings. The size of the input images is standardized to ensure segmentation at the same scale, and all models are transfer-learned based on the backbone network. We

retain the parameter settings that achieve the highest segmentation performance in terms of mIoU, as shown in Table A2.

Table A2. Optimal training parameters for state-of-the-art models

Model Architecture	Params. (M)	Input Size	Optimizer	Momentum	Learning Rate
U-Net [53]	13.39	512	SGD	0.9	0.005
FCN [33]	35.31	512	SGD	0.9	0.001
SegNet [54]	29.44	512	SGD	0.9	0.007
DeepLabv3 [49]	41.99	512	SGD	0.9	0.001
DeepLabv3+ [56]	74.87	512	SGD	0.9	0.007
DenseASPP [50]	9.20	512	SGD	0.9	0.001
DANet [57]	66.56	512	SGD	0.9	0.001
LR-ASPP [58]	30.22	512	SGD	0.9	0.001
SegNeXt [59]	29.91	512	SGD	0.9	0.007
SegFormer [60]	47.22	512	SGD	0.9	0.007
Swin-Unet [61]	27.17	512	SGD	0.9	0.007
Bi-Former [62]	56.81	512	SGD	0.9	0.007
ViT-CoMer [63]	60.50	512	SGD	0.9	0.007

References

1. Li, X.; Guo, Y.; Li, Y. Particle swarm optimization-based SVM for classification of cable surface defects of the cable-stayed bridges. *IEEE Access* **2019**, *8*, 44485–44492. [CrossRef]
2. Wickramasinghe, W.R.; Thambiratnam, D.P.; Chan, T.H.; Nguyen, T. Vibration characteristics and damage detection in a suspension bridge. *J. Sound Vib.* **2016**, *375*, 254–274. [CrossRef]
3. Rizzo, P.; di Scalea, F.L. Feature extraction for defect detection in strands by guided ultrasonic waves. *Struct. Health Monit.* **2006**, *5*, 297–308. [CrossRef]
4. Li, H.; Ou, J.; Zhou, Z. Applications of optical fibre Bragg gratings sensing technology-based smart stay cables. *Opt. Lasers Eng.* **2009**, *47*, 1077–1084. [CrossRef]
5. Cho, K.H.; Jin, Y.H.; Kim, H.M.; Moon, H.; Koo, J.C.; Choi, H.R. Caterpillar-based cable climbing robot for inspection of suspension bridge hanger rope. In Proceedings of the 2013 IEEE International Conference on Automation Science and Engineering (CASE), Madison, WI, USA, 17–20 August 2013 ; IEEE: Piscataway, NJ, USA, 2013; pp. 1059–1062. [CrossRef]
6. Xu, F.Y.; Wang, X.S.; Wang, L. Climbing model and obstacle-climbing performance of a cable inspection robot for a cable-stayed bridge. *Trans. Can. Soc. Mech. Eng.* **2011**, *35*, 269–289. [CrossRef]
7. Nguyen, S.T.; La, H.M. A climbing robot for steel bridge inspection. *J. Intell. Robot. Syst.* **2021**, *102*, 75. [CrossRef]
8. Sun, L.; Zhang, Y.; Wang, W.; Zhao, J. Lightweight Semantic Segmentation Network for RGB-D Image Based on Attention Mechanism. *Packag. Eng.* **2022**, *43*, 10.
9. Gong, R.; Ding, S.; Zhang, C.; Su, H. Lightweight and multi-pose face recognition method based on deep learning. *J. Comput. Appl.* **2020**, *40*, 6.
10. Shang, H.; Sun, C.; Liu, J.; Chen, X.; Yan, R. Defect-aware transformer network for intelligent visual surface defect detection. *Adv. Eng. Inform.* **2023**, *55*, 101882. [CrossRef]
11. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv* **2020**, arXiv:2010.11929. [CrossRef]
12. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. *arXiv* **2017**, arXiv:1706.03762. [CrossRef]
13. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. In Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015), San Diego, CA, USA, 7–9 May 2015; Computational and Biological Learning Society: San Diego, CA, USA, 2015. [CrossRef]
14. Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; Wei, Y. Deformable Convolutional Networks. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 764–773. [CrossRef]
15. Li, X.; Gao, C.; Guo, Y.; He, F.; Shao, Y. Cable surface damage detection in cable-stayed bridges using optical techniques and image mosaicking. *Opt. Laser Technol.* **2019**, *110*, 36–43. [CrossRef]
16. Salehi, H.; Burgueño, R. Emerging artificial intelligence methods in structural engineering. *Eng. Struct.* **2018**, *171*, 170–189. [CrossRef]

17. Luo, K.; Kong, X.; Zhang, J.; Hu, J.; Li, J.; Tang, H. Computer vision-based bridge inspection and monitoring: A review. *Sensors* **2023**, *23*, 7863. [CrossRef] [PubMed]
18. Yeum, C.M.; Dyke, S.J.; Ramirez, J. Visual data classification in post-event building reconnaissance. *Eng. Struct.* **2018**, *155*, 16–24. [CrossRef]
19. Xu, Z.; Wang, Y.; Hao, X.; Fan, J. Crack Detection of Bridge Concrete Components Based on Large-Scene Images Using an Unmanned Aerial Vehicle. *Sensors* **2023**, *23*, 6271. [CrossRef] [PubMed]
20. Chen, J.; Wang, H.; Tu, C.L.; Wang, X.S.; Li, X.D. Surface Defect Detection of Cable Based on Threshold Image Difference. In Proceedings of the 2021 IEEE Far East NDT New Technology & Application Forum (FENDT), Kunming, China, 14–17 December 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 185–190. [CrossRef]
21. Hu, J.; He, H.; Liao, G.; Hu, G. Study on Image Processing of Bridge Cable Surface Defect Detection System. In *Advances in Precision Instruments and Optical Engineering, Proceedings of the International Conference on Precision Instruments and Optical Engineering, Chengdu, China, 25–27 August 2021*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 447–456. [CrossRef]
22. Qu, Z.; Feng, H.; Zeng, Z.; Zhuge, J.; Jin, S. A SVM-based pipeline leakage detection and pre-warning system. *Measurement* **2010**, *43*, 513–519. [CrossRef]
23. Hsieh, Y.A.; Tsai, Y.J. Machine learning for crack detection: Review and model performance comparison. *J. Comput. Civ. Eng.* **2020**, *34*, 04020038. [CrossRef]
24. Cha, Y.J.; Choi, W.; Büyüköztürk, O. Deep learning-based crack damage detection using convolutional neural networks. *Comput.-Aided Civ. Infrastruct. Eng.* **2017**, *32*, 361–378. [CrossRef]
25. Dung, C.V. Autonomous concrete crack detection using deep fully convolutional neural network. *Autom. Constr.* **2019**, *99*, 52–58. [CrossRef]
26. Dong, S.; Tan, H.; Liu, C.; Hu, X. Apparent disease detection of bridges based on improved YOLOv5s. *J. Chongqing Univ.* **2024**, 1–12 . Available online: https://link.cnki.net/urlid/50.1044.N.20230331.1847.002 (accessed on 21 July 2024).
27. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1137–1149. [CrossRef]
28. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In *Computer Vision—ECCV 2016, Proceedings of the 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016*; Proceedings, Part I 14; Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37. [CrossRef]
29. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934. [CrossRef]
30. Zeiler, M.D.; Fergus, R. Visualizing and understanding convolutional networks. In *Computer Vision—ECCV 2014, Proceedings of the 13th European Conference, Zurich, Switzerland, 6–12 September 2014*; Proceedings, Part I 13; Springer: Berlin/Heidelberg, Germany, 2014; pp. 818–833. [CrossRef]
31. Geng, Q.; Zhou, Z.; Cao, X. Survey of recent progress in semantic image segmentation with CNNs. *Sci. China Inf. Sci.* **2018**, *61*, 051101 . [CrossRef]
32. Zhang, Y.; Yuen, K.V. Review of artificial intelligence-based bridge damage detection. *Adv. Mech. Eng.* **2022**, *14*, 16878132221122770. [CrossRef]
33. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 640–651. [CrossRef]
34. Hao, S.; Zhou, Y.; Guo, Y. A brief survey on semantic segmentation with deep learning. *Neurocomputing* **2020**, *406*, 302–321. [CrossRef]
35. Shi, J.; Dang, J.; Cui, M.; Zuo, R.; Shimizu, K.; Tsunoda, A.; Suzuki, Y. Improvement of damage segmentation based on pixel-level data balance using vgg-unet. *Appl. Sci.* **2021**, *11*, 518. [CrossRef]
36. Zhou, Z.; Rahman Siddiquee, M.M.; Tajbakhsh, N.; Liang, J. Unet++: A nested u-net architecture for medical image segmentation. In Proceedings of the Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support, Granada, Spain, 20 September 2018; Springer International Publishing: Cham, Switherland, 2018; pp. 3–11. [CrossRef]
37. Deng, W.; Mou, Y.; Kashiwa, T.; Escalera, S.; Nagai, K.; Nakayama, K.; Matsuo, Y.; Prendinger, H. Vision based pixel-level bridge structural damage detection using a link ASPP network. *Autom. Constr.* **2020**, *110*, 102973. [CrossRef]
38. Zheng, S.; Lu, J.; Zhao, H.; Zhu, X.; Luo, Z.; Wang, Y.; Fu, Y.; Feng, J.; Xiang, T.; Torr, P.H.; et al. Rethinking Semantic Segmentation from a Sequence-to-Sequence Perspective with Transformers. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 19–25 June 2021; pp. 6877–6886. [CrossRef]
39. 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. https://doi.org/10.48550/arXiv.2102.04306.
40. Thisanke, H.; Deshan, C.; Chamith, K.; Seneviratne, S.; Vidanaarachchi, R.; Herath, D. Semantic segmentation using Vision Transformers: A survey. *Eng. Appl. Artif. Intell.* **2023**, *126*, 106669. [CrossRef]
41. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 10012–10022. [CrossRef]

42. He, X.; Zhou, Y.; Zhao, J.; Zhang, D.; Yao, R.; Xue, Y. Swin transformer embedding UNet for remote sensing image semantic segmentation. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 4408715. [CrossRef]
43. Wang, W.; Xie, E.; Li, X.; Fan, D.P.; Song, K.; Liang, D.; Lu, T.; Luo, P.; Shao, L. Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction Without Convolutions. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021, pp. 568–578. [CrossRef]
44. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861. [CrossRef].
45. Chollet, F. Xception: Deep Learning With Depthwise Separable Convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017. [CrossRef]
46. Fang, M.; Liang, X.; Fu, F.; Song, Y.; Shao, Z. Attention mechanism based semi-supervised multi-gain image fusion. *Symmetry* **2020**, *12*, 451. [CrossRef]
47. 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 (CVPR), Las Vegas, NV, USA, 27–30 June 2016. [CrossRef]
48. Islam, M.A.; Jia, S.; Bruce, N.D. How much Position Information Do Convolutional Neural Networks Encode? In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019. [CrossRef]
49. Chen, L.C.; Papandreou, G.; Schroff, F.; Adam, H. Rethinking Atrous Convolution for Semantic Image Segmentation. *arXiv* **2017**, arXiv:1706.05587. [CrossRef].
50. 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 (CVPR), Salt Lake City, UT, USA, 18–23 June 2018. [CrossRef]
51. Wang, W.; Dai, J.; Chen, Z.; Huang, Z.; Li, Z.; Zhu, X.; Hu, X.; Lu, T.; Lu, L.; Li, H.; et al. Internimage: Exploring large-scale vision foundation models with deformable convolutions. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 14408–14419. [CrossRef]
52. 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; IEEE: Piscataway, NJ, USA, 2016; pp. 565–571. [CrossRef]
53. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015, Proceedings of the 18th International Conference, Munich, Germany, 5–9 October 2015*; Proceedings, Part III 18; Springer: Cham, Switherland, 2015; pp. 234–241. [CrossRef]
54. 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]
55. 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 (CVPR), Seattle, WA, USA, 14–19 July 2017. [CrossRef]
56. 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. [CrossRef]
57. 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 (CVPR), Long Beach, CA, USA, 15–20 June 2019. [CrossRef]
58. Howard, A.; Sandler, M.; Chu, G.; Chen, L.C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for MobileNetV3. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019. [CrossRef]
59. Guo, M.H.; Lu, C.Z.; Hou, Q.; Liu, Z.; Cheng, M.M.; Hu, S.M. SegNeXt: Rethinking Convolutional Attention Design for Semantic Segmentation. *arXiv* **2022**, arXiv:2209.08575. [CrossRef].
60. 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. [CrossRef]
61. Cao, H.; Wang, Y.; Chen, J.; Jiang, D.; Zhang, X.; Tian, Q.; Wang, M. Swin-Unet: Unet-Like Pure Transformer for Medical Image Segmentation. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 205–218. [CrossRef]
62. Zhu, L.; Wang, X.; Ke, Z.; Zhang, W.; Lau, R.W. BiFormer: Vision Transformer with Bi-Level Routing Attention. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 10323–10333. [CrossRef]
63. Xia, C.; Wang, X.; Lv, F.; Hao, X.; Shi, Y. ViT-CoMer: Vision Transformer with Convolutional Multi-scale Feature Interaction for Dense Predictions. *arXiv* **2024**, arXiv:2403.07392. [CrossRef].
64. Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; He, K. Aggregated residual transformations for deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1492–1500. [CrossRef]
65. 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 (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520. [CrossRef]

66. Woo, S.; Debnath, S.; Hu, R.; Chen, X.; Liu, Z.; Kweon, I.S.; Xie, S. Convnext v2: Co-designing and scaling convnets with masked autoencoders. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 16133–16142. [CrossRef]
67. Liu, Z.; Hu, H.; Lin, Y.; Yao, Z.; Xie, Z.; Wei, Y.; Ning, J.; Cao, Y.; Zhang, Z.; Dong, L.; et al. Swin transformer v2: Scaling up capacity and resolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 12009–12019. [CrossRef]