applied sciences

MDPI

Article

DWPIS: Dynamic-Weight Parallel Instance and Skeleton Network for Railway Centerline Detection

Xiaofeng Li [1], Yuxin Guo [1], Han Yang [1,*], Qixiang Ye [2] and Limin Jia [3]

[1] The School of Traffic and Transportation, Beijing Jiaotong University, Beijing 100044, China; xfengli@bjtu.edu.cn (X.L.); 20114023@bjtu.edu.cn (Y.G.)
[2] The School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing 100049, China; qxye@ucas.ac.cn
[3] The State Key Lab of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing 100044, China; lmjia@bjtu.edu.cn
* Correspondence: 17114237@bjtu.edu.cn

Abstract: The primary premise of autonomous railway inspection using unmanned aerial vehicles is achieving autonomous flight along the railway. In our previous work, fitted centerline-based unmanned aerial vehicle (UAV) navigation is proven to be an effective method to guide UAV autonomous flying. However, the empirical parameters utilized in the fitting procedure lacked a theoretical basis and the fitted curves were also not coherent nor smooth. To address these problems, this paper proposes a skeleton detection method, called the dynamic-weight parallel instance and skeleton network, to directly extract the centerlines that can be viewed as skeletons. This multi-task branch network for skeleton detection and instance segmentation can be trained end to end. Our method reformulates a fused loss function with dynamic weights to control the dominant branch. During training, the sum of the weights always remains constant and the branch with a higher weight changes from instance to skeleton gradually. Experiments show that our model yields 93.98% mean average precision (mAP) for instance segmentation, a 51.9% F-measure score (F-score) for skeleton detection, and 60.32% weighted mean metrics for the entire network based on our own railway skeleton and instance dataset which comprises 3235 labeled overhead-view images taken in various environments. Our method can achieve more accurate railway skeletons and is useful to guide the autonomous flight of a UAV in railway inspection.

Keywords: UAV railway inspection; railway detection; instance segmentation; skeleton detection; dynamic weight

Citation: Li, X.; Guo, Y.; Yang, H.; Ye, Q.; Jia, L. DWPIS: Dynamic-Weight Parallel Instance and Skeleton Network for Railway Centerline Detection. *Appl. Sci.* **2023**, *13*, 6133. https://doi.org/10.3390/app13106133

Academic Editor: Diogo Ribeiro

Received: 16 April 2023
Revised: 15 May 2023
Accepted: 15 May 2023
Published: 17 May 2023

1. Introduction

The operating mileage of China's railways has reached 160,000 km, and how to ensure the health of track infrastructure and the safety of the train operation environment is a very complex task. Due to technical limitations, the current safety-inspection work for rail transit still relies mainly on manual labor. However, the complex and dangerous terrain environment can make manual inspection operations difficult. A mainstream alternative solution is to utilize autonomous patrol and inspection using unmanned aerial vehicles (UAVs), which is both safer and more cost-effective. To accomplish this, it is necessary to ensure that the UAVs can fly along the railway autonomously.

Related research and applications are still in the early stages of development, and there is no mature theoretical method to achieve the autonomous GPS-independent flight of UAVs along railways. Among the studies that have been conducted, some auxiliary sensors are utilized to guide the autonomous flight when satellite navigation is not available. Magnetic sensors are used to determine the relative position between the UAV and transmission lines in [1]. Infrared markers are leveraged to estimate the relative position of the UAV in [2]. Binocular visual sensors are used to realize the 3D autonomous perception

of power lines in [3]. These methods rely on auxiliary sensors and are not suitable for our research. We have conducted several relevant studies using vision-based methods to detect the railways [4–6] and have proven their effectiveness in guiding UAVs' flight through experiments. There is still much room for improvement in our previous work

Railway detection is challenging for three main reasons. First, the structure of railways is complex, and includes the track, sleeper, ballast, railroad switch, and other structures. Second, the appearance of railways can vary depending on changes in weather and lighting conditions. Finally, railway lines have large aspect ratios, which can cause affine distortion when viewed from various angles [4].

To solve these issues, conventional methods of railway detection mainly use handcrafted features, including color, gradient, structure tensor, strip filter, and ridge shapes [7,8]. Heuristic algorithms are commonly used to segment railway targets, such as the Hough transform [9], K-means filter [10], steerable filter [11], and Kalman filter [10]. Morphological operators [12], visual saliency [13], and Markov random fields [14,15] are also used, in some studies, to determine the location and extract the shape of rail surface defects. However, using handcrafted features in conventional methods always needs a human to decide which kind of features is most suitable and needs a human to adjust the parameters. With the rise in deep learning, railway target detection has been designed as an end-to-end training task. A faster region-based convolutional neural network (R-CNN) is utilized for objective location in [14,16]. A deep convolution neural network (DCNN) is utilized for material classification in [17], for fasteners defect detection in [18–20] and for surface defect detection in [16]. A generative adversarial network (GAN) is utilized for defect detection in [21].

The basic research of this paper proposes a discretization-filtering-reconstruct (DFR) method [4] to fit a polynomial curve that represents the centerline of the railway, using the segmentation result of a lightweight CNN that includes a split-recursion-merge (SRM) module. While guiding a UAV's autonomous flight based on the fitted centerlines is proven effective, this method still has some drawbacks. The fitting procedure relies on several empirical parameters, including the number of splitted bins used to segment the mask, the threshold for the association between nodes representing the discrete trapezoidal blocks, the length of filter lines, and the number of filter strong nodes. They are determined through human experience and fixed in advance, thus lacking a theoretical basis. In addition , the fitted curves are not coherent and smooth, resulting in oscillation during the UAV's flight. To avoid these problems, we propose obtaining the centerline of the railway directly using an end-to-end method.

Whether the railway in the image is straight or curved, the structure of the railway remains symmetrical. Skeleton is a descriptor that can reveal the symmetry of an object [22]. Therefore, the centerline of the railway, which is needed to guide the UAV's autonomous flight, can be viewed as the skeleton of railway. Through an end-to-end skeleton-detection neural network, the uncertainty caused by the human-determined parameters mentioned above can be reduced. However, extracting the features of the railway skeleton is not an easy task, as objects with a similar shape to the skeleton may be misidentified due to the diversity of the railway's environment. To exclude the skeletons of other useless objects, an end-to-end instance-segmentation neural network is added in parallel to detect the target railway. This parallel network is utilized to extract the features of the useful region, which contains the correct skeleton.

Based on the ideas presented above, this paper proposes a novel method called the dynamic-weight parallel instance and skeleton network (DWPIS). The DWPIS network is composed of an instance-segmentation branch based on SOLOv2 [23] and a skeleton-detection branch based on our previous work, AdaLSN [24]. The two branches are combined using a fused loss function with dynamic weight. The purpose of this work is illustrated in Figure 1 and the contributions of this paper are summarized as follows:

- Given the constant shape and structure of the railways, their centerline can be viewed as a skeleton to guide UAVs' autonomous flight. This paper extracts the skeleton

directly through an end-to-end skeleton-detection neural network and locates the target through an instance-segmentation neural network, rather than computing and fitting the centerline using detection results.

- For the instance-segmentation branch, this paper changes the backbone to the ELAN-based [25] backbone as in YOLOv7 [26], and adds the attention module SimAM [27] after each level of features in the backbone. Experimental results show that the detection accuracy of the instance segmentation is improved.
- For the skeleton-detection branch, this paper changes the loss function to Dice loss [28] due to the serious imbalance between skeleton pixels and background. Furthermore, a threshold function is added after the sigmoid function in the inference process to enhance the skeleton results.
- This paper designs a fused loss function to adjust the weight of the two parallel branches during training, where the sum of the weights for the two parts remains one. The weight of the loss for the instance-segmentation branch decreases from a large value to quickly extract the necessary features. Meanwhile, the weight of the loss for the skeleton-detection branch gradually increases, to become dominant.

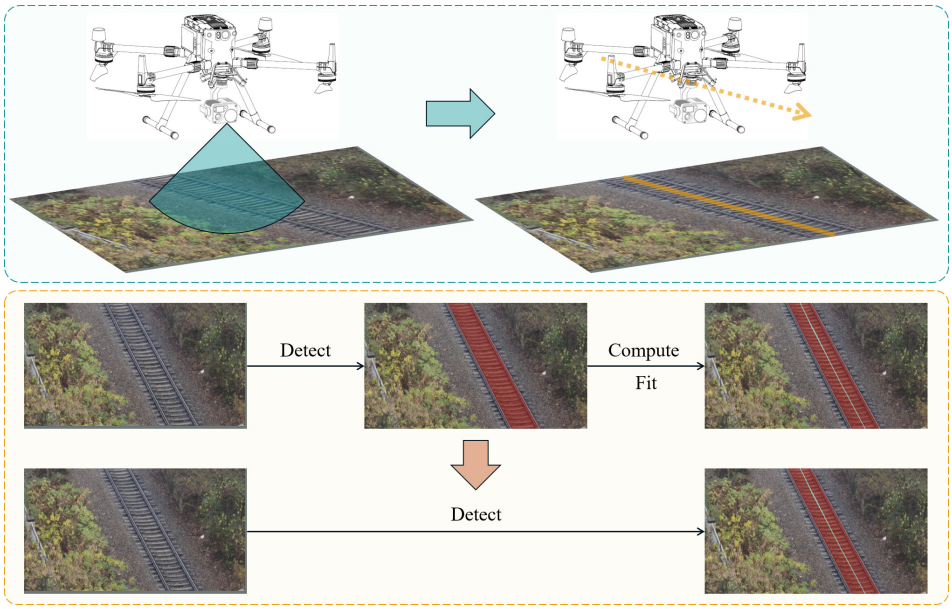

Figure 1. Our task: (**Top**) UAVs rely on vision-baesd detection methods to obtain the centerline of railways and guide the UAV's autonomous flight. (**Bottom**) Use of end-to-end method to replace the operations of computing and fitting, which rely on empirical parameters.

The rest of this paper is organized as follows. In Section 2, the work related to the instance-segmentation and skeleton-detection methods is outlined. In Section 3, the architecture of the proposed DWPIS is presented, including the two parallel branches and the novel fused loss function with dynamic weight. In Section 4, the experiments and analysis are discussed. In Section 5, the conclusion is presented.

2. Related Work

Our network has two branches for separate instance segmentation and skeleton detection. Existing works on instance segmentation have yielded excellent results in terms of both segmentation accuracy and inference speed. Research on skeleton detection in images with simple backgrounds has also made some progress.

2.1. Instance Segmentation

Instance segmentation is the combination of two tasks, semantic segmentation and object detection, and requires the classification of pixels and location of different instances. It includes two types of methods, two-stages methods and on-stage methods [29].

The two-stages methods can be further divided into two categories, top-down methods based on detection and bottom-up methods based on semantic segmentation. Top-down methods first locate the bounding box of an instance using object detection, and then perform semantic segmentation for each detection box. One such method, Mask RCNN [30], extends Faster-RCNN by adding a branch for predicting an object mask in parallel with the existing branch for bounding-box recognition. Bottom-up methods first perform semantic segmentation at the pixel level, and then distinguish different instances using clustering, metric learning, or other methods. In [31], masks for all objects are obtained through semantic segmentation and training is performed using a discriminative loss function, which makes it easy to cluster the image into instances.

The one-stage instance-segmentation methods include anchor-based methods inspired by YOLO [32] and RetinaNet [33], and anchor-free methods inspired by FCOS [34]. In anchor-based methods, the main idea is to classify and regress candidate target regions called anchors, produced by sliding windows. YOLACT [35] generates a dictionary of prototype masks and predicts per-instance linear combination coefficients. YOLACT++ [36] optimizes the prediction head and adds a novel, fast mask re-scoring branch. SOLO [29] reformulates the instance segmentation as two sub-tasks: category prediction and instance-mask generation problems. SOLOv2 [23] uses the matrix non-maximum suppression (NMS) technique and object mask generation is decoupled into a mask kernel prediction and mask feature learning. The main idea behind anchor-free methods is to transform them into keypoint-based methods by locating the keypoint, or into region-based methods by locating the center of an object and predicting the contours of the object. PolarMask [37] predicts the contour of an instance through instance-center classification and dense distance regression in a polar coordinate.

As one-stage methods usually have a quicker inference speed than two-stage methods, they are more suitable for real-time scenarios involving guiding the autonomous flight of UAVs. While the segmentation accuracy of one-stage methods for small targets may be less satisfactory, the railways in overhead-view images captured by UAVs are not considered small objects. Therefore, in this paper, we propose using SOLOv2 as the base architecture for the instance-segmentation branch, considering both accuracy and speed.

2.2. Skeleton Detection

The skeleton is a structure-based object descriptor that reveals local symmetry as well as connectivity between object parts [38,39]. Skeleton detection has been used in many applications, including object recognition and retrieval, pose estimation, hand-gesture recognition, shape matching, scene text detection, and road detection in aerial scenes [22]. Among them, the most common and popular use is detecting and locating key points of the human body to recognise different body movements effectively.

A pioneer work of skeleton-detection methods, the edge-detection method HED [40] turns pixel-wise edge classification into image-to-image prediction. The side-output residual network (SRN) [41] leverages the side-output residual units to build short connections between adjacent side-output branches for matching object symmetry at different scales. For the problem that the scales of object skeletons may dramatically vary among objects and object parts, Hi-Fi [42] introduces a novel hierarchical feature-integration mechanism to capture high-level features from deeper layers and low-level details from shallower layers, which essentially establishes dense side-output branches. To cope with object parts of large widths, Ref. [22] proposes a “skeleton context flux” representation, which encodes the relative position of skeletal pixels to semantically meaningful entities.

Our previous work [43] proposes to formulate the pixel-wise binary classification tasks as linear reconstruction problems within a linear span network architecture (LSN) consisting

of three components: feature linear span, resolution alignment, and subspace linear span. Each component contains several linear-span units implemented by a concatenation layer, a convolutional layer and a slice layer to minimize the reconstruction error. Building on top of an LSN, our improved research, adaptive linear-span network (AdaLSN) [24], defines a mixed unit-pyramid search space. A genetic architecture search is applied to jointly optimize unit-level operations and pyramid-level connections for adaptive feature-space expansion.

Compared to the state of the art, AdaLSN with sufficient feature-space expansion achieves significantly higher accuracy by utilizing complementary feature extraction and architecture optimization. Considering the serious imbalance between positive and negative samples in railway-skeleton images, Dice loss function [28] is more suitable for our research. Therefore, in this paper, we propose using our previous work, AdaLSN, with a Dice loss function as the base architecture for the skeleton-detction branch.

3. Dynamic-Weight Parallel Instance and Skeleton Network

The central idea of our network is changing the dominant of the two branches, the instance-segmentation branch and skeleton-detection branch, through a fused loss function with dynamic weight during training, to locate the targets and extract the skeletons more accurately. The architecture of the network is illustrated in Figure 2.

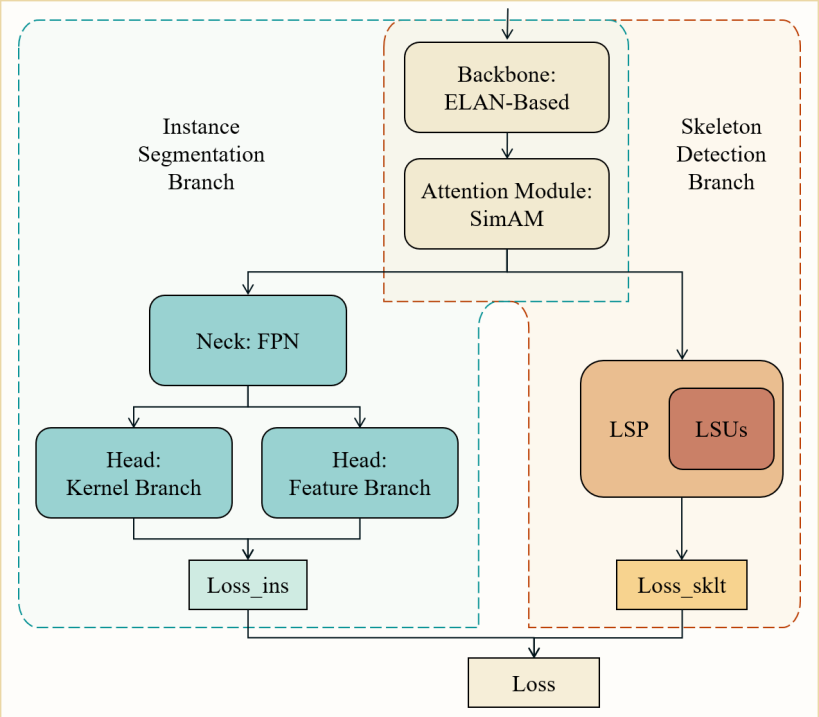

Figure 2. Architecture of our network.

3.1. Instance-Segmentation Branch

The target of the instance-segmentation branch in our work is to determine the location of the railway. This paper uses SOLOv2 [23] as the base architecture for this part, which converts the location prediction task into a classification task. In SOLOv2, the image is divided into $S \times S$ cells, resulting in S^2 location classes, and each instance can be assigned to one of them as its location category.

The backbone of SOLOv2 used to extract the feature is the traditional residual neural network (ResNet), which requires deep architectures, making it computationally expensive. This paper uses a more advanced backbone mainly composed of ELAN [25], which is similar to YOLOv7 [26], as shown in Figure 3. ELAN, a layer aggregation architecture with efficient gradient propagation paths, is mainly composed of VoVNet [44] combined with

CSPNet [45]. It optimizes the gradient length of the overall network with the structure of a stack in a computational block [25].

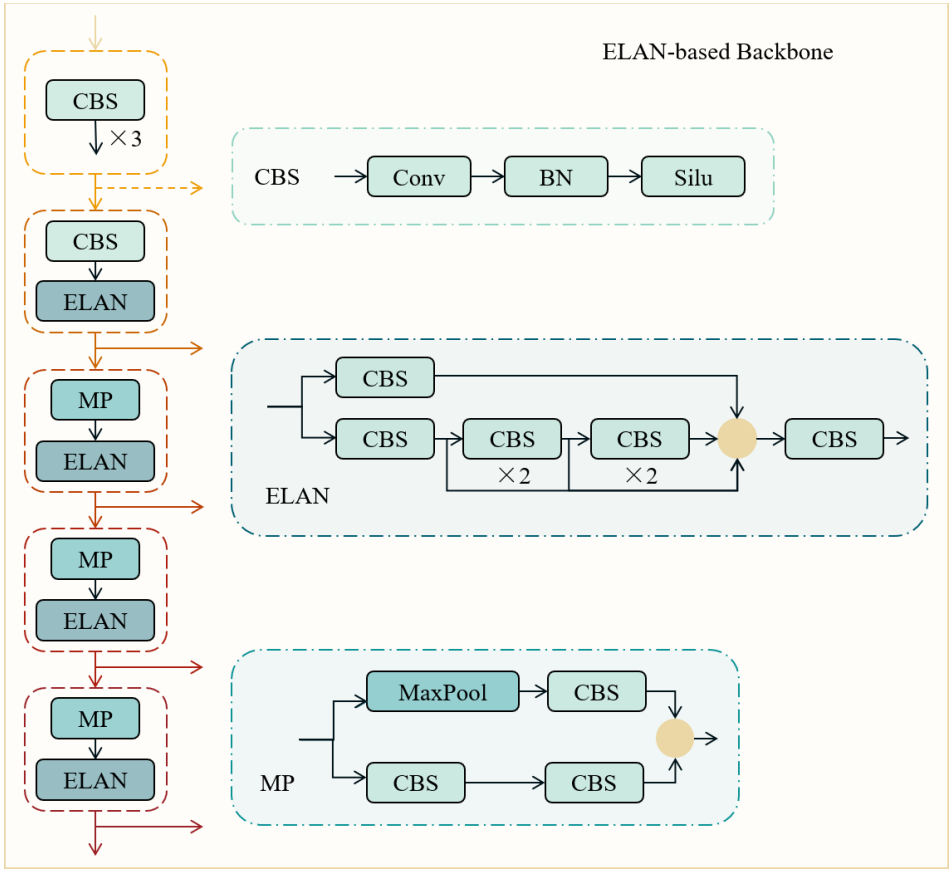

Figure 3. Backbone of our network.

After the backbone, this paper adds the attention module SimAM [27] at each level of the feature map to divert attention to the most important regions and disregard the irrelevant parts. It is a type of channel-spatial-attention module [46] and a significant advantage of it is that it does not introduce additional parameters.

To assign objects of varying sizes to different levels of feature maps, SOLOv2 employs a feature pyramid network (FPN) [47] as the neck of the framework. SOLOv2 has two heads: the kernel branch which predicts the semantic category and mask kernel, and the feature branch which predicts the mask feature. Therefore, the training loss function is defined as follows:

$$L_{ins} = L_{cate} + \lambda L_{mask} \tag{1}$$

where L_{ins} is for the whole instance-segmentation network, L_{cate} is for semantic-category classification, and L_{mask} is for mask prediction.

3.2. Skeleton-Detection Branch

The target of our skeleton-detection branch is to extract the skeleton of the railway. This paper uses our previous work AdaLSN [24] as the base architecture for this branch. AdaLSN consists of two components, the backbone and the linear span pyramid (LSP). The LSP is constructed by attaching the linear span unit (LSU) to the convolutional layer of the backbone in five stages, which can be viewed as a side-out branch. The key innovation in AdaLSN is the search for four classes of architecture encoding, including the connection between the backbone and LSUs, the inner edges and operators in each LSU, the connection between LSUs in the LSP, and the connection between loss and the LSU. The focus of this

paper is not on searching for the best network architecture through a genetic algorithm; thus, one fixed architecture is utilized.

AdaLSN uses Inception-v3 [48] as the best backbone. In this paper, a new backbone composed of ELAN is used, which is the same as the backbone of the instance-segmentation branch, to connect the two parallel branches.

This paper uses a Dice loss layer [28] instead of the binary cross-entropy loss, which is utilized in the original AdaLSN architecture and may result in a sizeable loss at the start of our training process. Dice is suitable for situations where there is a serious imbalance between positive and negative samples, as is the case in our skeleton-detection task. The skeleton pixels, which are positive samples, only represent a small fraction of the whole image compared to the background pixels, which are negative samples. Therefore, the Dice loss is much more appropriate for our mission. The training loss function is defined as follows:

$$L_{unit} = \sum_{i=1}^{5} L_i \tag{2a}$$

$$L_{sklt} = L_{unit} + L_{fuse} \tag{2b}$$

where L_{unit} is the sum of loss of every LSU; L_i is the loss of each LSU; L_{sklt} is the loss for the whole skeleton-detection network; and L_{fuse} is the loss of the fused LSU, whose input is the output from all the LSUs.

During the inference process, the original method in AdaLSN limits the fused output of the LSP to the range of 0 to 1 using the sigmoid function, but it changes the shape of results significantly. To obtain better skeleton results against complex backgrounds, this paper designs an extension threshold function to be added after the sigmoid function, defined as:

$$p_{th} = \begin{cases} 1, & if \ \ p_{sig} >= \theta \\ 0, & if \ \ p_{sig} < \theta \end{cases} \tag{3}$$

where p_{th} is the value of each pixel after our threshold function, p_{sig} is the value of each pixel after the sigmoid function, and θ is the threshold filtering the pixels. All the pixels with a value less than θ are changed to zero, and the others are changed to one.

3.3. Architecture of DWPIS

Previous skeleton-detection methods commonly use datasets in which the target occupies most of the image and there are no other objects in the simple background. However, our target object, a railway, is located in a complex environment in most of images to be evaluated, including other objects of a similar shape. This increases the difficulty of detection and the error rate. Additionally, the height of a UAV is not fixed due to the different inspection-mission requirements. As the UAV flies higher, the railways in the image become smaller and the background becomes more complex.

To address the aforementioned problems, this paper proposes the novel dynamic-weight parallel instance and skeleton network. This network is divided into two parallel branches, one for instance segmentation and another for skeleton detection separately. They use the same backbone and attention module to extract features from input images first. Then, before a fused loss function merges the two branches, the loss of each respective branch is obtained. The detailed network, including training, inference and post-process, is shown in Figure 4.

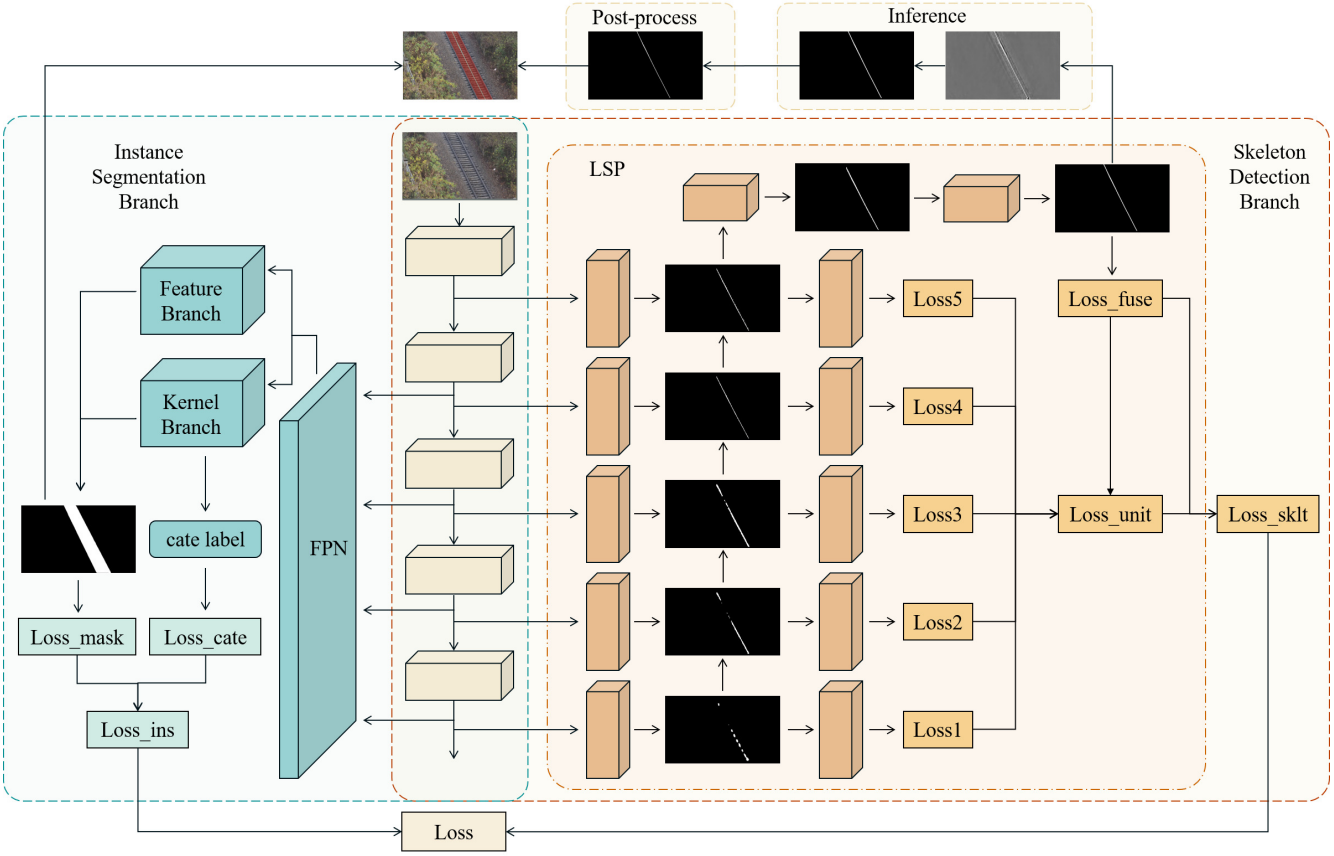

Figure 4. Detailed network including training, inference and post-process.

At the beginning of training, the network primarily relies on the instance-segmentation branch to extract useful target features, which guides the network to quickly determine the features of the correct location. At the same time, the skeleton-detection branch is optimized mainly through the former. As training progresses, the instance-segmentation branch should achieve good-enough results after only a few epochs and the focus of the network should gradually shift to the skeleton-detection branch. Since the training epochs required for skeleton detection are much more than that for instance segmentation, the skeleton-detection branch maintains dominance for a long time after the result of the instance-segmentation branch is stabilized.

In order to achieve the above training strategy, this paper reformulates a fused loss function with dynamic weight, which is defined as:

$$\omega = \frac{1}{1 + e^{-\frac{\epsilon - \alpha}{\beta}}} \tag{4a}$$

$$L = (1 - \omega) L_{ins} + \omega L_{sklt} \tag{4b}$$

where ω is the weight for the loss of the skeleton-detection branch, ϵ is the current training epoch, parameter α represents the epoch where the weights of the two branches are both 0.5, parameter β determines the rate of weight changes, and L is the total loss for our network. As the function demonstrates, the sum of the weights for the two branches always remains at one. As the training epoch increases, the weight of the instance-segmentation branch decreases gradually while the weight of the skeleton-detection branch increases.

4. Experiments and Analysis

4.1. Experimental Setting

Datasets. A total of 3235 overhead-view images of railways were collected in Ma'anshan, Nanjing, Qinghai–Tibet Railway and our laboratory. These images were divided into 2277 images for the training dataset, 655 images for the validation dataset, and 303 images for the test dataset at a ratio of 7:2:1. Figure 5 shows some examples.

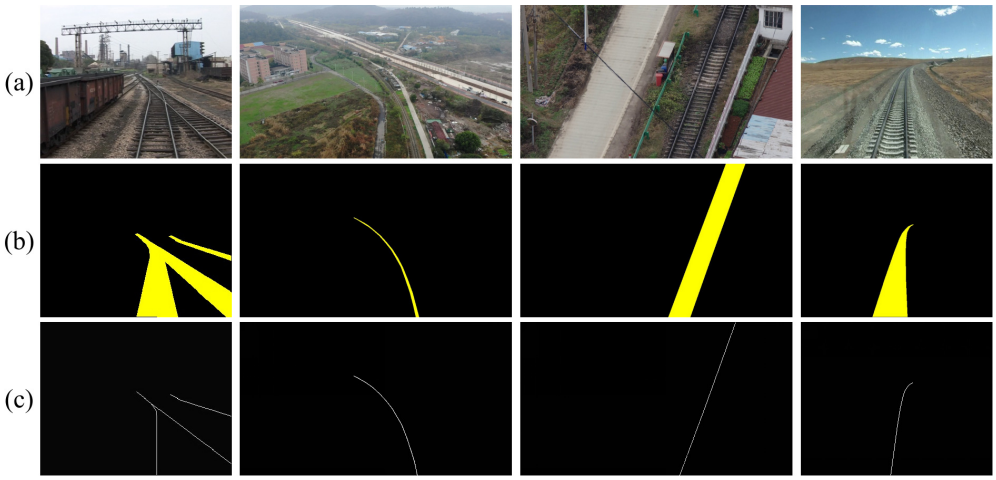

Figure 5. Examples of our dataset. (**a**): Original images. (**b**): Instance labels. (**c**): Skeleton labels.

Implementation details. Our network was implemented using PyTorch and was run on three NVIDIA TITAN RTX GPUs (24 GB RAM). For training, this paper used the stochastic gradient descent (SGD) optimizer with an initial learning rate of 0.01, a momentum of 0.9, and a weight decay of 0.0001. The final model was trained for 360 epochs and the models for ablation experiments were trained for 36 epochs. Pre-processing operations included resizing, random flipping, normalization and padding. After inference, this paper performed erosion as the post-processing method to remove the noise points or lines for skeleton detection.

Evaluation protocol. Average precision (AP), average recall (AR) and mean average precision (mAP) were utilized as evaluation metrics for instance segmentation. For skeleton detection, the F-measure score (F-score) was used as the evaluation metric. A weighted-mean metric W_{mean} was designed for our dynamic-weight parallel instance and skeleton network, which consisted of mAP for the instance-segmentation branch and F-score for the skeleton-detection branch, defined as:

$$W_{mean} = \gamma mAP + (1 - \gamma)F_{score} \tag{5}$$

where γ is the weight value of 0.2 for mAP.

4.2. Main Result

Our network achieved 93.98% mAP for instance segmentation, 51.9% F-score for skeleton detection and 60.32% W_{mean} for the whole task. The evaluation metrics are shown in Table 1 and some classic examples for detection masks are shown in Figure 6.

Table 1. Evaluation metrics results of our network.

Network	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	mAP	F-Score	W_{mean}
Ours	93.6	98.3	96.3	-	86.5	95.2	93.98	51.9	60.32

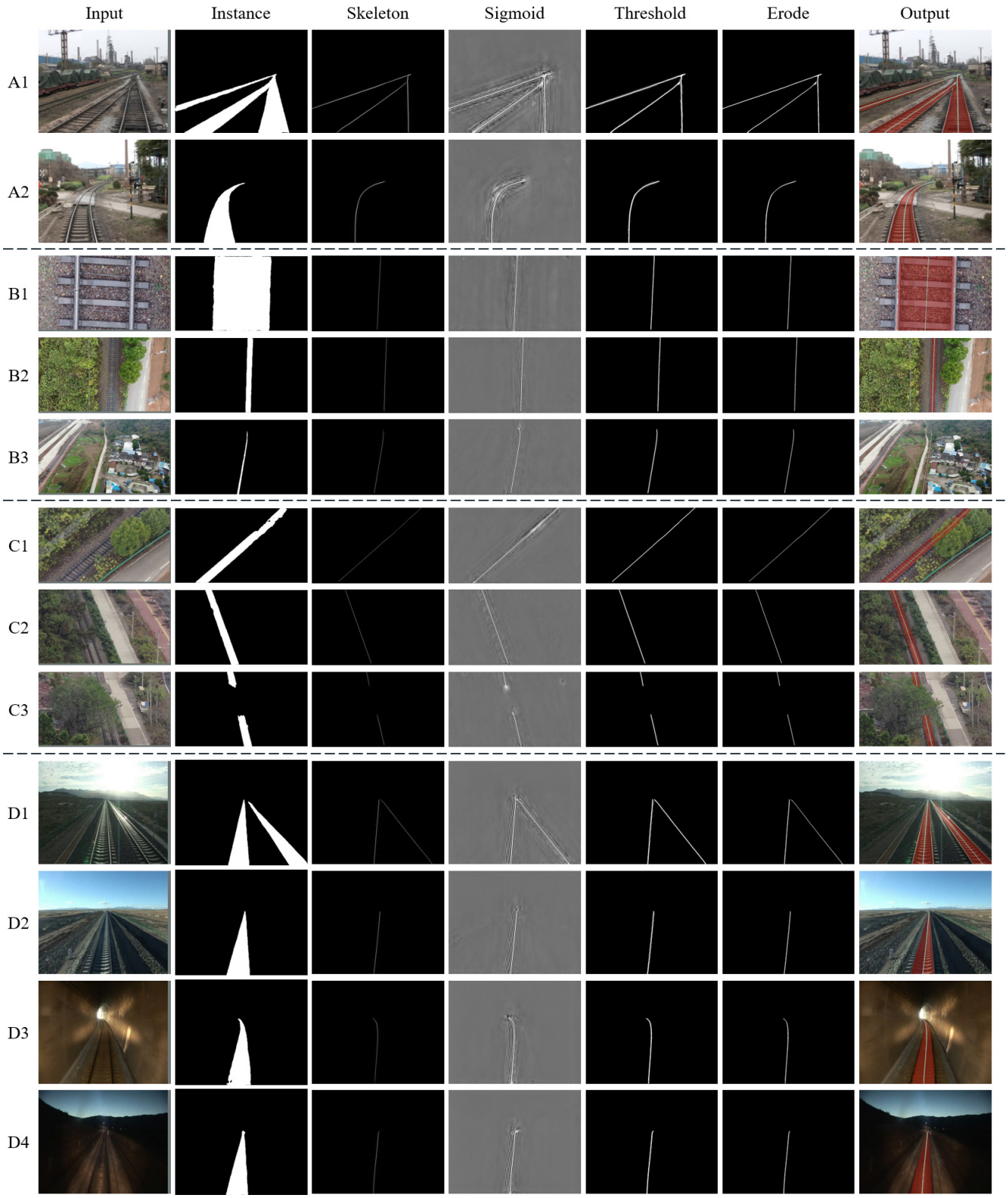

Figure 6. Examples of results of our network. (**A1**,**A2**) Single railway, multiple railways, straight railways and curved railways. (**B1**–**B3**) Different flying heights of UAVs. (**C1**–**C3**) Obscured railway. (**D1**–**D4**) Different lighting conditions.

The model performs well for images with a single railway or multiple railways, as well as straight railways or curved railways, as indicated by Figure 6(A1,A2). The UAVs need to fly at different heights to carry out the tasks, so the model must be suitable for various scales

of railways in the images. As shown in Figure 6(B1–B3), the model can always accurately identify the railways and skeletons when the flying height changes from low to high. In certain environments, the railway may be shaded by trees or other objects, which increases the difficulty of the task. As shown in Figure 6(C1,C2), the railways and skeletons are fully detected even though the railway is partly shaded by sparse trees. However, in Figure 6(C3), the detected results are truncated by a tree because it is so dense that it completely obscures the railway. In order to operate during different times of the day and in various weather conditions, the UAVs must be able to detect the skeleton of a railway under different lighting conditions. As demonstrated in Figure 6(D1–D4), the results of instance segmentation and skeleton detection are always right, whether the light is strong or dim.

Compared to the original SOLOv2 with ResNet-50 backbone, our changes to the backbone and attention module result in a 3.58% increase in mAP for instance segmentation. Compared to the original AdaLSN with a fixed architecture and Inception-v3 backbone, our novel network achieves a 2.2% improvement in skeleton detection.

4.3. Ablation Experiments

This paper investigates and compares the following five aspects in our methods:

Threshold function. To improve the process of inference using a model that only requires a few epochs of training, a threshold function is added after the original sigmoid function to simplify the features. We compare the inference results with and without a threshold function, in which the threshold is $1 - 10^{-1}$, $1 - 10^{-3}$, 1–10^{-5} and $1 - 10^{-7}$. As shown in Figure 7, the inference achieves significant improvement through the addition of a threshold function. The inference results are good enough when the threshold value increases to $1 - 10^{-7}$ which is ultimately chosen as our threshold. As expected, the skeleton of each railway becomes more obvious and the F-score is much higher than the inference result without our threshold function, as shown in Table 2.

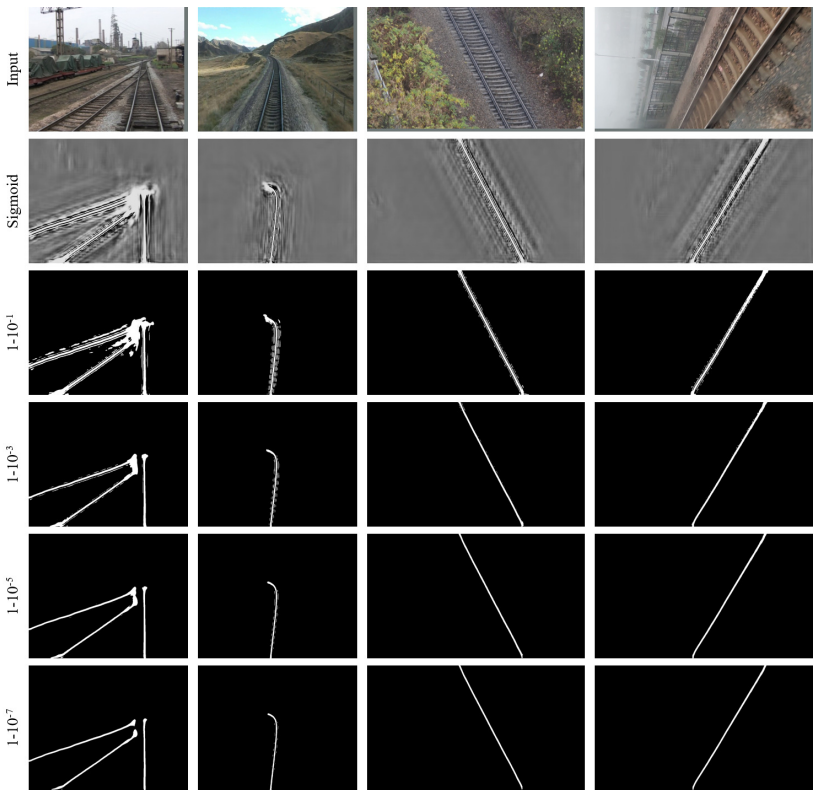

Figure 7. Comparison of the results of with and without our threshold function during the inference process. The first row: input images. The second row: results without our threshold function. The other rows: results with our threshold function and the thresholds are $1 - 10^{-1}$, $1 - 10^{-3}$, $1 - 10^{-5}$, and $1 - 10^{-7}$, respectively.

Table 2. Comparison of with and without threshold function.

Inference	mAP
Sigmoid function	1.18
Sigmoid function + threshold function	51.0

Backbone. According to previous research, the best backbone for SOLOv2 is RseNet, and Inception-v3 is the best for AdaLSN. To further compare the impact of the backbone, this paper trains SOLOv2 (only for the instance-segmentation branch), AdaLSN with a fixed architecture (only for the skeleton-detection branch), and our dynamic-weight parallel instance and skeleton network with different backbones, including ELAN-based backbone, Inception-v3, ResNet-50 and ResNet-101.

Table 3 shows the evaluation metrics for only the instance-segmentation branch. As demonstrated, the network using the ELAN-based backbone achieves a higher mAP compared to the networks that use other backbones. The network achieves a 3.56% mAP improvement over ResNet-50, indicating that the ELAN-based architecture is the optimal choice among them for the instance-segmentation branch.

Table 3. Comparison of backbone and attention module for instance-segmentation network.

Network	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	mAP
Inception-v3	90.6	96.3	92.2	-	79.7	93.3	90.42
ResNet-50	87.9	97.0	89.9	-	72.1	91.7	87.72
ResNet-100	85.4	95.4	89.3	-	69.1	89.4	85.72
ELAN-based	91.2	97.2	92.3	-	82.4	93.3	91.28
ELAN+SimAM	91.3	97.0	93.2	-	81.4	93.6	91.3

As shown in Table 4, the ELAN-based backbone does not achieve the best F-score compared to the networks using other backbones, for only the skeleton-detection branch. However, the main target of changing the backbone is to improve the detection results of the instance-segmentation branch. Therefore, this paper pays more attention to the metrics of the instance-segmentation network and our parallel network.

Table 4. Comparison of backbone and attention module for skeleton-detection network.

Network	F-Score
Inception-v3	48.8
ResNet-50	35.8
ResNet-100	37.0
ELAN-based	48.2
ELAN+SimAM	47.9

The evaluation metrics for our novel network are shown in Table 5. The W_{mean} of the ELAN-based backbone network is 0.06 smaller than that of the Inception-v3 backbone network, which is mainly due to the smaller AP. However, as shown in Table 3, the mAP of the ELAN-based backbone network is larger. The different results may because the weight of the instance-segmentation branch is always less than one in our network. Therefore, this paper still uses the ELAN-based backbone for more epochs of the training.

Table 5. Comparison of backbone and attention module for our network.

Network	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	F-Score	W_{mean}
Inception-v3	91.2	97.3	93.1	-	81.0	93.6	50.8	58.89
ResNet-50	86.6	96.4	90.2	-	70.8	90.5	45.8	54.02
ResNet-100	86.9	96.1	90.2	-	69.3	91.1	46.1	54.22
ELAN-based	90.8	96.9	92.7	-	81.3	93.0	50.8	58.83
ELAN+SimAM	90.9	96.9	93.1	-	81.0	93.3	51.0	59.01

Attention module. This paper compares the effect of adding SimAM to the network using an ELAN-based backbone. For only always instance-segmentation network, the mAP is slightly larger when SimAM is added, as shown in Table 3. As expected, adding SimAM also improves our novel network with the instance-segmentation branch and skeleton-detection branch, as shown in Table 4. However, it performs poorly for the network for skeleton detection, as shown in Table 5.

Loss function parameter. This paper designs a novel fused loss function to adjust the dynamic weight of the two branches during training. To increase the weight of the skeleton-detection branch as the loss of the instance-segmentation branch gradually stabilizes, the sigmoid function is chosen as the base. The sum of the weights for the two branches is always kept at one. This paper adds two parameters to adjust the function: alpha, which determines when the weight is 0.5, and beta, which determines the rate of weight increases.

Through experiments, the segmentation results are already satisfactory after training for 36 epochs. Therefore, this paper adjusts both parameters to be related to this epoch number: α is changed to be $36/\alpha'$, and β is changed to be $36/\beta'$. The results of the experiments for finding the best parameters are shown in Figure 8.

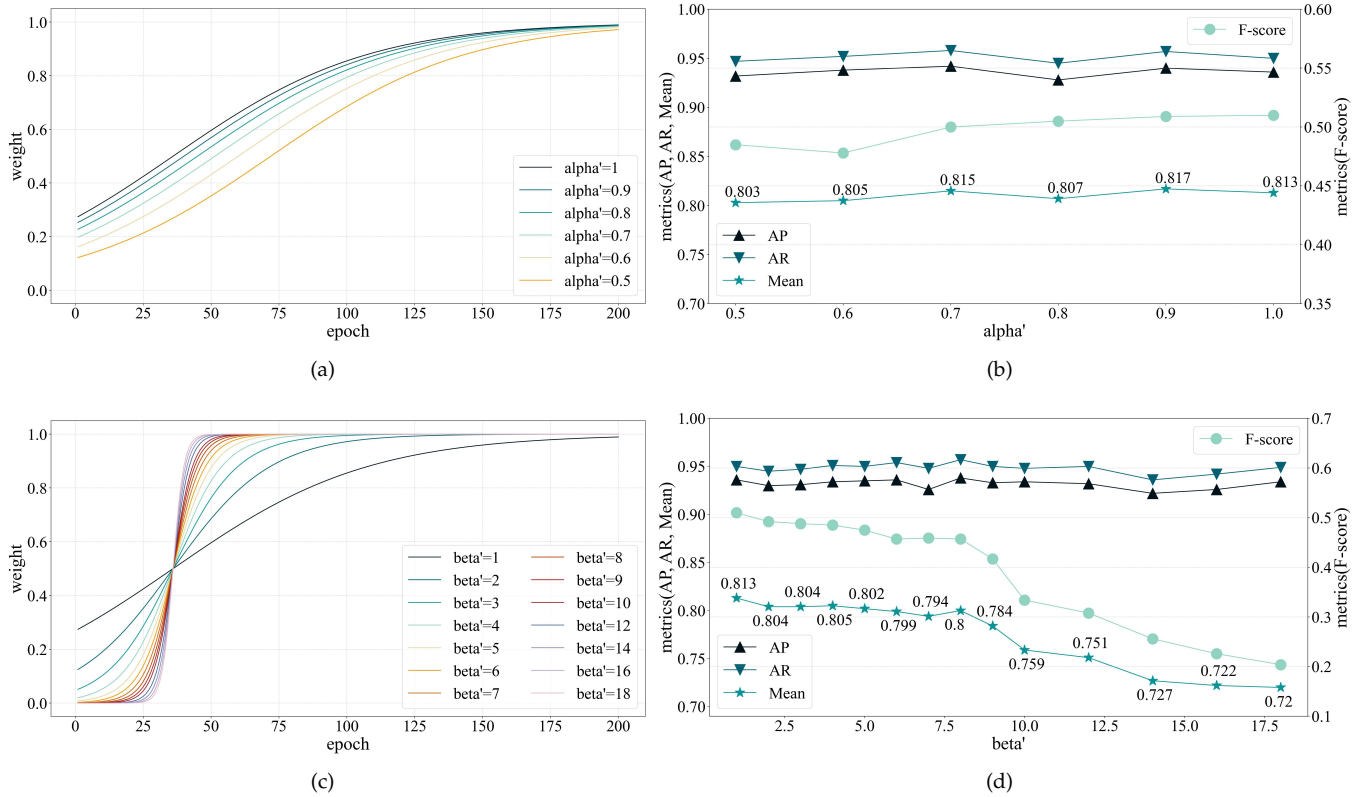

Figure 8. Comparison of different parameters of our loss function. (**a**) Loss function curves of varied α' parameters with fixed β'. (**b**) Evaluation metrics of networks with the loss functions in (**a**). (**c**) Loss-function curves of different β' parameters with fixed α'. (**d**) Evaluation metrics of networks with the loss functions in (**c**).

This paper varies β' from 1 to 18 while keeping α' fixed at 1, and the curves for each value are shown in Figure 8c. When β' is equal to one, the overall rate of weight increases changes the slowest during training. A new mean metric was calculated by assigning the weights of 0.35 to AP and AR, and 0.3 to F-score. It is mainly because the instance-segmentation branch is more important at the beginning of training. As shown in Figure 8d, the best result is obtained when the β' is set to one.

The value of α' was changed from 0.5 to 1 while keeping β' fixed at its best value, and the curves for each value are shown in Figure 8a. Decreasing $alpha'$ means the weight of the skeleton-detection branch is lower during the same training epoch. The mean metrics is calculated in the same way as evaluating β'. As shown in Figure 8b, the best value for α' is 0.9.

Parallel network. The main idea behind our network is to add a parallel instance-segmentation branch to remove the skeleton of the wrong target, on the base of the skeleton-detection network. This paper compares the results of our novel network to the skeleton-detection network. As shown in Table 6, our network achieves a 3.1% improvement in F-score compared to the skeleton detection network with same backbone and attention model. It is a 2.2% improvement compared to the fixed AdaLSN with the original backbone. As expected, the results of the skeleton-detection network show skeletons of objects that are not railways, which are not present in our novel parallel network shown in Figure 9.

Table 6. Comparison of with and without instance branch.

Network	F-Score
Fixed AdaLSN (only skeleton)	48.8
Ours (only skeleton)	47.9
Ours (two branches)	51.0

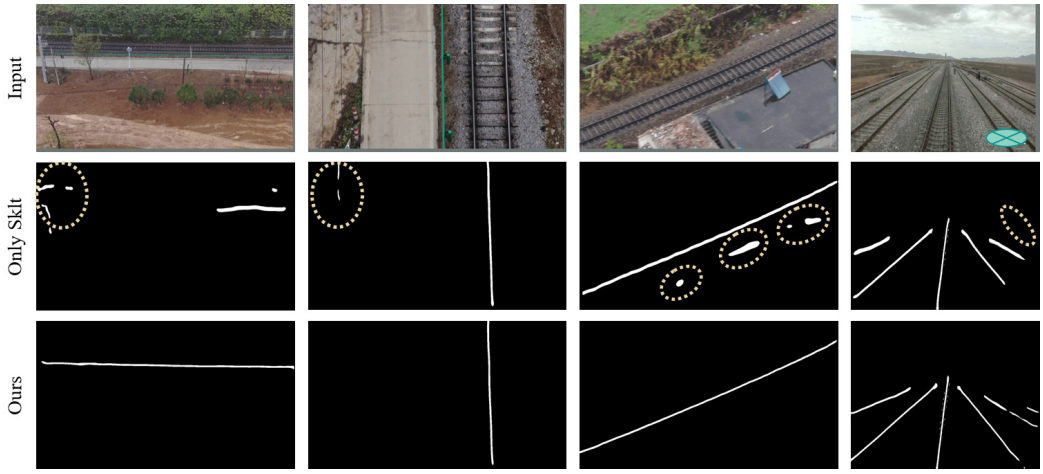

Figure 9. Comparison of the results of our network with and without instance-segmentation branch. The first row: input images. The second row: results of our network without instance-segmentation branch, in which the dashed ellipses represent the misidentified or undetected skeletons. The third row: results of our network with instance-segmentation branch.

4.4. Analysis

Experimental results show that our DWPIS with two branches can obtain stronger railway skeletons than the base architecture of the skeleton-detection network. Upon careful analysis, these enhancements come from three main sources: the instance-segmentation branch, fused loss function, and inference function.

Firstly, the instance segmentation locates the railway target. Through training the instance-segmentation branch, the network extracts the useful features of the railways containing the skeleton and the parameters of the shared backbone are optimized.

Secondly, the fused loss function with dynamic weight decided, using training-epoch changes, the dominance of the training. Through controlling the dominant branch, the network is mainly trained for instance segmentation first and then mainly for skeleton detection for a long time. For the skeleton-detection branch, the Dice loss function is more suitable for the serious imbalance between positive and negative samples and increases the convergence speed of training.

Finally, the threshold function, the subject of the skeleton-detection branch in the inference procedure, optimizes the skeleton results. By adding a threshold function after the original sigmoid function, better results can be obtained by filtering noise after fewer training epochs.

5. Conclusions

In this work, which focuses on the task of detecting the centerline of a railway to guide UAV autonomous flight, this paper introduced the dynamic-weight parallel instance and skeleton network, which is an end-to-end multi-task branch method.

- This paper proposed a novel network with two parallel branches, including an instance-segmentation branch and a skeleton-detection branch. The instance-segmentation branch is to determine the location of the railway, which is improved in the backbone and attention module. The skeleton-detection branch extracts the skeleton of the railway and is improved in the loss function and inference process.
- This paper designed a fused loss function with dynamically changing weights during the training process to change the dominant task. The weight of the instance-segmentation branch decreases from initially being large to extract the correct location. In contrast, the weight of the skeleton-detection branch increases from a very small value to being extremely close to one to focus on skeleton features.

This paper compares the evaluation metrics of the following architectures: (1) the different backbones of the instance-segmentation network, skeleton-detection network and two-branches network, (2) with and without an attention module, (3) different parameters of fused loss function with dynamic weight, (4) with and without a threshold function in the inference procedure, and (5) with and without the instance-segmentation branch.

Overall, experiments based on our own railway datasets taken in various places and times demonstrate that our network performs well in different environments. However, the skeleton results may be truncated when there is a dense occlusion in the field of view, which will influence the smoothness of the UAV's autonomous flight. Apart from this, the speed of the inference of our network may not meet the stringent requirements of real-time flight of lightweight UAVs. Considering the obscured railway and the complexity of network, there is still much room for improvement in both accuracy and speed in future work. To solve the issue of the truncated skeleton, future work can focus on two aspects: connecting the truncated parts utilizing the structure features of the skeleton in the post-process; and obtaining a complete skeleton directly through optimizing the network. To increase the speed of inference, the two branches can be merged in the bottleneck to decrease the complexity of network in future work.

Author Contributions: Conceptualization, X.L. and Y.G.; methodology, X.L. and Y.G.; software, X.L. and Y.G.; validation, X.L. and Y.G.; formal analysis, X.L. and Y.G.; investigation, X.L., Y.G. and H.Y.; resources, X.L. and Q.Y.; data curation, X.L., Y.G. and H.Y.; writing—original draft preparation, X.L. and Y.G.; writing—review and editing, X.L. and Y.G.; visualization, X.L. and Y.G.; supervision, X.L., Q.Y. and L.J.; project administration, X.L. and L.J.; funding acquisition, X.L. and L.J.; All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by National Natural Science Foundation of China (No. 62033004).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to express their sincere appreciation to the editors and the reviewers for their constructive comments.

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

Abbreviations

The following abbreviations are used in this manuscript:

UAV	unmanned aerial vehicle
GPS	global positioning system
CNN	convolutional neural network
GAN	generative adversarial network
FPN	feature pyramid network
LSN	linear span network
LSP	linear span pyramid
LSU	linear span unit
RAM	random access memory
SGD	stochastic gradient descent
AP	average precision
AP_{50}	AP at IoU = 0.50
AP_{75}	AP at IoU = 0.75
AP_S	AP for small objects
AP_M	AP for medium objects
AP_L	AP for large objects
AR	average recall
mAP	mean average precision
F-score	F-measure score

References

1. Wu, Y.; Luo, Y.; Zhao, G.; Hu, J.; Gao, F.; Wang, S. A novel line position recognition method in transmission line patrolling with UAV using machine learning algorithms. In Proceedings of the 2018 IEEE International Symposium on Electromagnetic Compatibility and 2018 IEEE Asia-Pacific Symposium on Electromagnetic Compatibility (EMC/APEMC), Suntec City, Singapore, 14–18 May 2018; pp. 491–495.
2. Wang, Z.; Liu, S.; Chen, G.; Dong, W. Robust visual positioning of the UAV for the under bridge inspection with a ground guided vehicle. *IEEE Trans. Instrum. Meas.* **2021**, *71*, 1–10. [CrossRef]
3. Xu, C.; Li, Q.; Zhou, Q.; Zhang, S.; Yu, D.; Ma, Y. Power line-guided automatic electric transmission line inspection system. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 1–18. [CrossRef]
4. Yang, H.; Li, X.; Guo, Y.; Jia, L. Discretization–Filtering–Reconstruction: Railway Detection in Images for Navigation of Inspection UAV. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 1–13. [CrossRef]
5. Yang, H.; Li, X.; Guo, Y.; Jia, L. RT-GAN: GAN Based Architecture for Precise Segmentation of Railway Tracks. *Appl. Sci.* **2022**, *12*, 12044. [CrossRef]
6. Guo, Y.; Li, X.; Jia, L.; Qin, Y. An efficient rail recognition scheme used for piloting mini autonomous UAV in railway inspection. In Proceedings of the 2020 International Conference on Sensing, Diagnostics, Prognostics, and Control (SDPC), Beijing, China, 5–7 August 2020; pp. 284–290.
7. Bar Hillel, A.; Lerner, R.; Levi, D.; Raz, G. Recent progress in road and lane detection: A survey. *Mach. Vis. Appl.* **2014**, *25*, 727–745. [CrossRef]
8. Shin, B.S.; Klette, R. *Visual Lane Analysis—A Concise Review*; The University of Auckland, Multimedia Imaging: Auckland, New Zealand, 2013.
9. Tapia-Espinoza, R.; Torres-Torriti, M. Robust lane sensing and departure warning under shadows and occlusions. *Sensors* **2013**, *13*, 3270–3298. [CrossRef] [PubMed]
10. Zhang, J.; Liu, L.; Wang, B.; Chen, X.; Wang, Q.; Zheng, T. High speed automatic power line detection and tracking for a UAV-based inspection. In Proceedings of the 2012 International Conference on Industrial Control and Electronics Engineering, Xi'an, China, 23–25 August 2012; pp. 266–269.
11. Liu, Y.; Mejias Alvarez, L.; Li, Z. Fast power line detection and localization using steerable filter for active UAV guidance. In *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XXXIX-B3: XXII ISPRS Congress*; ISPRS-International Society for Photogrammetry and Remote Sensing; Elsevier: Amsterdam, The Netherlands, 2012; pp. 491–496.
12. Yu, H.; Li, Q.; Tan, Y.; Gan, J.; Wang, J.; Geng, Y.a.; Jia, L. A coarse-to-fine model for rail surface defect detection. *IEEE Trans. Instrum. Meas.* **2018**, *68*, 656–666. [CrossRef]

13. Nieniewski, M. Morphological detection and extraction of rail surface defects. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 6870–6879. [CrossRef]
14. Jin, X.; Wang, Y.; Zhang, H.; Zhong, H.; Liu, L.; Wu, Q.J.; Yang, Y. DM-RIS: Deep multimodel rail inspection system with improved MRF-GMM and CNN. *IEEE Trans. Instrum. Meas.* **2019**, *69*, 1051–1065. [CrossRef]
15. Zhang, H.; Jin, X.; Wu, Q.J.; Wang, Y.; He, Z.; Yang, Y. Automatic visual detection system of railway surface defects with curvature filter and improved Gaussian mixture model. *IEEE Trans. Instrum. Meas.* **2018**, *67*, 1593–1608. [CrossRef]
16. Kang, G.; Gao, S.; Yu, L.; Zhang, D. Deep architecture for high-speed railway insulator surface defect detection: Denoising autoencoder with multitask learning. *IEEE Trans. Instrum. Meas.* **2018**, *68*, 2679–2690. [CrossRef]
17. Giben, X.; Patel, V.M.; Chellappa, R. Material classification and semantic segmentation of railway track images with deep convolutional neural networks. In Proceedings of the 2015 IEEE International Conference on Image Processing (ICIP), Quebec City, QC, Canada, 27–30 September 2015; pp. 621–625.
18. Tu, Z.; Wu, S.; Kang, G.; Lin, J. Real-time defect detection of track components: Considering class imbalance and subtle difference between classes. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1–12. [CrossRef]
19. Chen, J.; Liu, Z.; Wang, H.; Nunez, A.; Han, Z. Automatic defect detection of fasteners on the catenary support device using deep convolutional neural network. *IEEE Trans. Instrum. Meas.* **2017**, *67*, 257–269. [CrossRef]
20. Zhong, J.; Liu, Z.; Wang, H.; Liu, W.; Yang, C.; Han, Z.; Nunez, A. A looseness detection method for railway catenary fasteners based on reinforcement learning refined localization. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1–13. [CrossRef]
21. Zhang, D.; Gao, S.; Yu, L.; Kang, G.; Wei, X.; Zhan, D. DefGAN: Defect detection GANs with latent space pitting for high-speed railway insulator. *IEEE Trans. Instrum. Meas.* **2020**, *70*, 1–10. [CrossRef]
22. Wang, Y.; Xu, Y.; Tsogkas, S.; Bai, X.; Dickinson, S.; Siddiqi, K. Deepflux for skeletons in the wild. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 5287–5296.
23. Wang, X.; Zhang, R.; Kong, T.; Li, L.; Shen, C. Solov2: Dynamic and fast instance segmentation. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 17721–17732.
24. Liu, C.; Tian, Y.; Chen, Z.; Jiao, J.; Ye, Q. Adaptive linear span network for object skeleton detection. *IEEE Trans. Image Process.* **2021**, *30*, 5096–5108. [CrossRef] [PubMed]
25. Wang, C.Y.; Liao, H.Y.M.; Yeh, I.H. Designing Network Design Strategies Through Gradient Path Analysis. *arXiv* **2022**, arXiv:2211.04800.
26. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv* **2022**, arXiv:2207.02696.
27. Yang, L.; Zhang, R.Y.; Li, L.; Xie, X. Simam: A simple, parameter-free attention module for convolutional neural networks. In *International Conference on Machine Learning*; PMLR: London, UK, 2021; pp. 11863–11874.
28. 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.
29. Wang, X.; Kong, T.; Shen, C.; Jiang, Y.; Li, L. Solo: Segmenting objects by locations. In Proceedings of the Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Part XVIII 16; Springer: Berlin/Heidelberg, Germany, 2020; pp. 649–665.
30. 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.
31. De Brabandere, B.; Neven, D.; Van Gool, L. Semantic instance segmentation with a discriminative loss function. *arXiv* **2017**, arXiv:1708.02551.
32. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
33. 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.
34. Tian, Z.; Shen, C.; Chen, H.; He, T. Fcos: Fully convolutional one-stage object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9627–9636.
35. Bolya, D.; Zhou, C.; Xiao, F.; Lee, Y.J. Yolact: Real-time instance segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9157–9166.
36. Zhou, C. *Yolact++ Better Real-Time Instance Segmentation*; University of California, Davis: Davis, CA, USA, 2020.
37. Xie, E.; Sun, P.; Song, X.; Wang, W.; Liu, X.; Liang, D.; Shen, C.; Luo, P. Polarmask: Single shot instance segmentation with polar representation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 12193–12202.
38. Marr, D.; Nishihara, H.K. Representation and recognition of the spatial organization of three-dimensional shapes. *Proc. R. Soc. Lond. Ser. B Biol. Sci.* **1978**, *200*, 269–294.
39. Dickinson, S.J.; Leonardis, A.; Schiele, B.; Tarr, M.J. *Object Categorization: Computer and Human Vision Perspectives*; Cambridge University Press: Cambridge, UK, 2009.
40. Xie, S.; Tu, Z. Holistically-nested edge detection. In Proceedings of the IEEE International Conference on Computer Vision, Boston, MA, USA, 7–12 June 2015; pp. 1395–1403.

41. Ke, W.; Chen, J.; Jiao, J.; Zhao, G.; Ye, Q. SRN: Side-output residual network for object symmetry detection in the wild. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Venice, Italy, 22–29 October 2017; pp. 1068–1076.
42. Zhao, K.; Shen, W.; Gao, S.; Li, D.; Cheng, M.M. Hi-fi: Hierarchical feature integration for skeleton detection. *arXiv* **2018**, arXiv:1801.01849.
43. Liu, C.; Ke, W.; Qin, F.; Ye, Q. Linear span network for object skeleton detection. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 133–148.
44. Lee, Y.; Hwang, J.W.; Lee, S.; Bae, Y.; Park, J. An energy and GPU-computation efficient backbone network for real-time object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Long Beach, CA, USA, 15–20 June 2019.
45. Wang, C.Y.; Liao, H.Y.M.; Wu, Y.H.; Chen, P.Y.; Hsieh, J.W.; Yeh, I.H. CSPNet: A new backbone that can enhance learning capability of CNN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 390–391.
46. Guo, M.H.; Xu, T.X.; Liu, J.J.; Liu, Z.N.; Jiang, P.T.; Mu, T.J.; Zhang, S.H.; Martin, R.R.; Cheng, M.M.; Hu, S.M. Attention mechanisms in computer vision: A survey. *Comput. Vis. Media* **2022**, *8*, 331–368. [CrossRef]
47. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
48. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.