TriRNet: Real-Time Rail Recognition Network for UAV-Based Railway Inspection

Lei Tong, *Student Member, IEEE*, Limin Jia, *Member, IEEE*, Yixuan Geng, *Student Member, IEEE*, Yong Qin, *Member, IEEE*, Donghai Song, Bidong Miao, Tian Tang and Zhipeng Wang, *Member, IEEE*

***Abstract*—UAVs have a broad application prospect in the field of railway inspection due to their excellent mobility and flexibility. However, it still faces challenges, such as high human labor costs and low intelligence levels. Therefore, it is of great significance to develop a real-time intelligent rail recognition algorithm that can be deployed on the onboard computing device to guide the UAV's camera to follow the target rail area and complete the inspection automatically. However, a significant issue is that rails from the perspective of UAVs may appear with changing pixel widths and various inclination angles. Concerning the issue, a general and adaptive rail representation method based on projection length discrimination (RRM-PLD) is proposed. It can always select the optimal representation direction, horizontal or vertical, to represent any kind of rails. With the RRM-PLD, a novel architecture (Real-Time Rail Recognition Network, TriRNet) is proposed. In TriRNet, a designed inter-rail attention (IRA) mechanism is presented to fuse local features of single rails and global features of other rails to accurately discriminate the geometric distribution of all rails in the image in a regressive way and thus improve the final recognition accuracy. Further, one-to-one mapping from anchor points to final feature maps is established. It greatly simplifies the model design process and improves the model's interpretability. Besides, detailed model training strategies are also presented. Extensive experiments have verified the effectiveness and superiority of the proposed formulation in terms of both network reasoning latency and recognition accuracy.**

***Index Terms*—Rail recognition; attention; UAV; anchor points; automatic railway inspection**

I. INTRODUCTION

RAILWAY inspection, especially inspection of facilities in the train running areas, is of great significance for the railway safety operation. Over the past decade, artificial intelligence (AI) empowered by deep convolutional neural networks has developed rapidly in various fields. Numerous AI models have been proposed and applied to visual tasks such as object detection [1], target tracking [2], and semantic segmentation [3, 4], and have achieved remarkable results. Researchers of the railway industry have also proposed a series of AI models for railway inspection works including rail component detection [5-8], rail surface or fastener defect detection [9-12], and environmental hazard evaluation [13], etc. But there is no doubt that the establishment of all kinds of such AI models is completely inseparable from large-scale data. Traditionally, inspection workers and dedicated inspection trains are employed to collect relevant image data. However, the work efficiency of manual inspection is extremely low while the labor cost is high. The dedicated inspection trains cannot completely cover all important infrastructures in the entire train running area. They will also affect the normal configuration of the train running diagram. In addition, inspection workers and inspection trains can only undergo the inspection during the maintenance time window at night. In that case, the inspection quality can be adversely affected by the weak light condition to a large extent.

Unmanned aerial vehicles (UAVs) can inspect objects in the air in a non-contact manner due to their excellent flexibility and maneuverability. Thus, they have been applied to a wide range of industries recently. Over the years, some researchers have noticed the potential of adopting UAVs for railway inspection applications. Previous works dedicated to UAV-based railway inspection include mainly two categories: image-based and point cloud-based inspection. The works on image-based inspection contain railroad track components inspection [7], small objects detection of railway scene [14], rail surface defects detection [9, 10], rail fastener defect inspection [15], catenary support device inspection [16], railway infrastructure monitoring [17], railway scene image dehazing to enhance the effect of railway object detection [18], and railway scene parsing [19]. The works on point cloud-based inspection contain rail track detection [20], contact wire measurement with LiDAR [21], and full-range railway environment segmentation [22]. However, in these works of adopting UAVs to conduct railway inspection, the data collection of specific objects is completed by manually manipulating the UAVs. The process takes up much time and has very low work efficiency and intelligence level because of too much manual intervention to adjust the attitudes of the UAV and the gimbal during the flight,

Manuscript received xx xx, 2022; revised xx xx, 2023; accepted xx xx, 2023. Date of publication xx xx, 2023; date of current version xx xx, 2023. This work was supported in part by the Fundamental Research Funds for the Central Universities under Grant 2022YJS111; in part by National Natural Science Foundation of China under Grant 61833002 and 62033004; in part by State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, under Contract No. RCS2022ZT005; and in part by research project of China State Railway Group Co., Ltd. under Grant P2022X001. The Associate Editor for this article was xx xx. *(Corresponding author: Zhipeng Wang; Limin Jia).*

Lei Tong, Limin Jia, Yixuan Geng, Yong Qin, and Zhipeng Wang are all with the State Key Laboratory of Advanced Rail Autonomous Operation, Beijing Jiaotong University, Beijing, 100044, China (e-mails: 20114070@bjtu.edu.cn; lmjia@bjtu.edu.cn; 17114233@bjtu.edu.cn; yqin@bjtu.edu.cn; zpwang@bjtu.edu.cn);

Donghai Song and Bidong Miao are both with the China Railway Electrification Bureau Group Co., Ltd., Beijing, 100044, China (e-mails: songdonghai1a@163.com; miaobidong@163.com);

Tian Tang are with the United Aircraft Group, Beijing, 100176, China (e-mail: tangtian@zhz.com).

which will inevitably slow down the data collection process greatly.

The onboard edge computing devices endow UAVs with computing power and make it possible to fulfill UAV-based automatic data collection of the train running area without excessive human controlling intervention. To achieve that, it is quite essential to develop rail recognition algorithms that can be deployed and run on the onboard edge computing devices in real-time to guide the UAV's camera to follow the target train running area automatically, as shown in the technical roadmap for UAV-based onboard automatic rail recognition in Fig. 1.

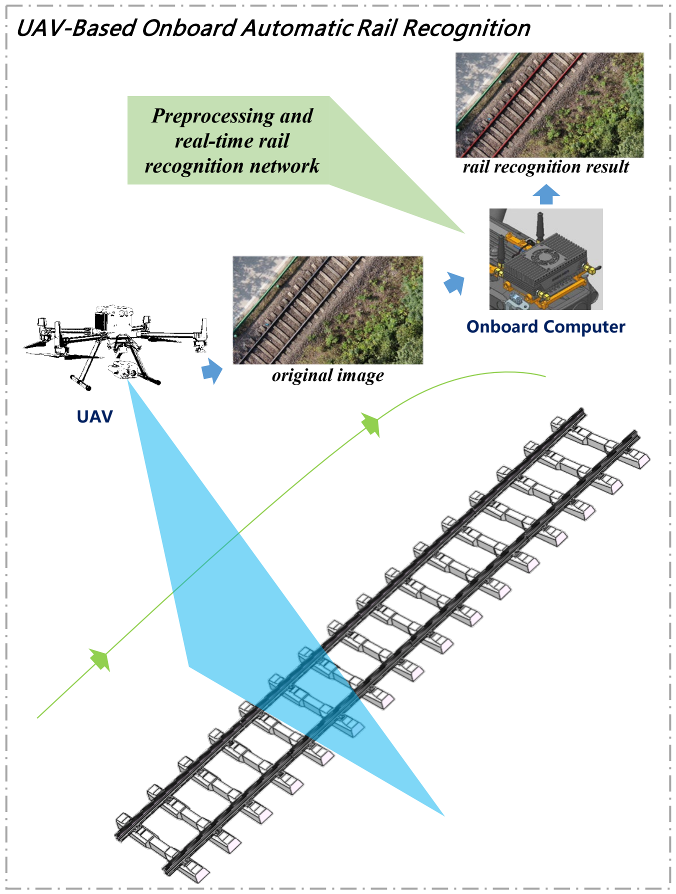

UAV-Based Onboard Automatic Rail Recognition
Preprocessing and real-time rail recognition network
rail recognition result
Onboard Computer
original image
UAV

Fig. 1. UAV-based onboard automatic rail recognition.

Different from the images by train-mounted cameras in which the geometric position distribution of the rails fluctuates within only a tiny range and remains roughly in the same relative direction. In contrast, the inclination angles of the rails in the UAV's view can change greatly as it moves and rotates all the time, as shown in Fig. 2. Moreover, since the focal length of the camera and the relative flying height between the UAV and the rails always change from time to time, the pixel width of the rail from the UAV's perspective may also vary sharply from only a few pixels to 70 more pixels at 1080p image size. Thus, these issues along with the changeable environment on both sides of the railway bring a great challenge to the task of rail recognition. Previous works [5-8] have been developed for multiple rail components detection that covers rail recognition, but they all investigate completely different kinds of rail images under a far smaller and fixed field of view, in which only one evident rail exists. This makes it easier for them to accomplish rail recognition. [23] developed a track segmentation network that is far from efficient and not capable of being deployed on onboard computing devices. Segmentation-based methods must adopt some post-processing method to obtain the final line prediction (like least squares linear regression) from many discrete segmented pixels. Thus, this kind of method is slowed down further and suffers a lot from the possible predicted FP pixels. Therefore, they cannot be directly adopted for the work involved in this paper. Concerning these issues, this paper proposes a general and adaptive rail representation method and a real-time attention-aware rail recognition network that can be run on the onboard computer in real-time. The main contributions of this paper are summarized as follows:

(1) Aiming at the challenge that the directions of the rails from the perspective of UAVs are arbitrary and vary from time to time, a general and adaptive rail representation method based on projection length discrimination (RRM-PLD) is proposed, which can always select the optimal representation direction to represent any kind of rails.

(2) To accurately discriminate and identify the geometric characteristics of rails, an attention-aware real-time rail recognition network (TriRNet) is proposed for the UAVs' onboard edge computing devices, in which the designed inter-rail attention (IRA) mechanism can fuse local features of single rails and global features of all rails and thus improve the final recognition accuracy.

(3) Focusing on the task of line-shaped structure detection, one-to-one mapping from anchor points to final customized feature maps for proposal generation is established to simplify the complexity of the network design process and improves the network interpretability.

The following part of this paper is organized as follows: Section II presents some previous works related to the research of this paper. Section III gives a detailed description of the proposed RRM-PLD based on projection length discrimination, deep rail recognition architecture, and the corresponding training strategies. Section IV makes a detailed experimental design elaborations and relevant result analysis and verifies the effectiveness and the superior performance of the proposed approach in terms of both inference latency and recognition accuracy. Section V concludes this paper.

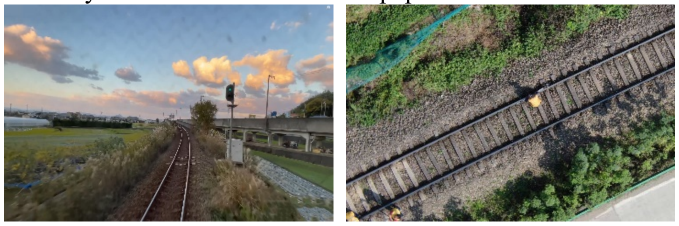


Fig. 2. Comparison between the images collected by the vehicle-mounted camera and that collected by UAVs.

II. RELATED WORKS

Different from the familiar lane detection task, rail recognition has not attracted enough attention from relevant researchers before because of the particularity that the traveling of trains is completely restricted by rails. But with the application of UAVs in the field of railway inspection, rail recognition has become essential. At present, according to our best knowledge, there is no mature algorithm that can be applied to perform rail recognition for UAV remote sensing images directly. [20] used UAV point clouds to perform 3D model construction and rail track detection but it cannot produce real-time detection results. [23] developed a railway track segmentation network that runs on a large graphics card (NVIDIA Tesla V100). Their processing scheme and reasoning speed are completely not applicable to making real-time feedback decisions to the UAV and thus limit their application to onboard computers. Besides, this kind of traditional approach considers the detection of the line-shaped rails as a simple discrete pixel-wise segmentation task, introducing unnecessary convolution calculation and resource consumption. Moreover, additional curve-fitting operations from many segmented discrete pixels are needed to obtain the final line geometry. Despite that, the lane detection task benchmarked on CULane [24], TuSimple [25], and LLAMAS [26] has made great progress recently, providing a good reference for this work.

The existing lane detection algorithms can be roughly divided into three different kinds of methods: parametric prediction methods, segmentation-based methods, and anchor-based methods. The parametric prediction methods formulate the lanes in the image as curve equations, including PolyLaneNet [27] and LSTR [28] based on transformer [29]. The parameters of the polynomial lane curve equation are predicted directly with deep neural networks. But this kind of method does not adapt to the changing inclination angles of the rails from the perspective of UAVs due to the limitations of the determined equation form. Moreover, their model accuracy still needs to be improved further. The segmentation-based methods [24, 30-35] predict the lanes at the pixel and instance levels. But the discreteness of the predicted pixels limits the accuracy of the detection results, and some post-clustering methods are developed to address the problem [31, 35]. However, the lane marker is represented as a mask, not a line-shaped object which is inefficient to describe lanes. The existing state-of-the-art architectures are basically all anchor-based methods which can be categorized into two different kinds. The first kind [36-39] determines the final line proposal for lane marker by regressing a relative location to the predefined line-shaped anchors, which are adapted from the box-shaped anchor-based object detection architectures such as Faster R-CNN [40] and Mask R-CNN [41], and thus are complex to formulate a customized and suitable anchor group. Also, some work such as [39], has developed a very effective attention mechanism to realize parameters regression and line proposal generation by fusing attention-guided global features and local features. The second kind [42-46] formulates the lane markers by a series of row anchors more efficiently and succinctly. The locations of the lane markers are determined by choosing the best location at each row of the gridded image. Therefore, the lane markers are finally predicted by a series of anchor points row-wisely which can be optimized by easily adding more line-shape-related constraints to the formulation. Despite the high performance of the anchor-based methods, they still cannot solve the challenge of diverse and changing rails from the perspective of UAVs.

III. METHODOLOGY

In this section, the proposed anchor point-based RRM-PLD is discussed firstly, which is based on the projection length discrimination and the concept of one-to-one mapping from the anchor points to the features in the final extracted feature maps. The representation method can be applied to rails at various viewing angles for the UAVs. Next, an attention-aware deep convolutional neural network architecture is presented to complete the effective and real-time rail recognition task with the proposed one-to-one mapping rail representation method. Then the method for the generation of the predicted rail proposals is presented. Next, the detailed training strategy of the whole proposed model is introduced, mainly including the balanced transpose co-training strategy and the design of the dedicated integrated loss function. Finally, the coordinates transformation between the pixel coordinates and the anchor point-based gridded coordinates is introduced.

A. Rail Representation

Inspired by the concept that lines can be represented by certain points sampled from them, this paper attempts to represent each rail by a series of uniformly sampled points from the image. But a rail can be sampled from both horizontal and vertical directions. As illustrated in Fig. 3, a horizontal and a vertical positioning line are plotted in the figure respectively, which are adopted to position the rails in the image. Both kinds of positioning lines are first discretized, that is, they are divided into many gridding cells, each of which represents a valid location where the rails can possibly pass through. All these locations are called anchor points. An additional cell is attached at the end of each positioning line to indicate the location where no rail exists on the entire positioning line. Green and red circles are exploited to represent the actual locations of different rails.

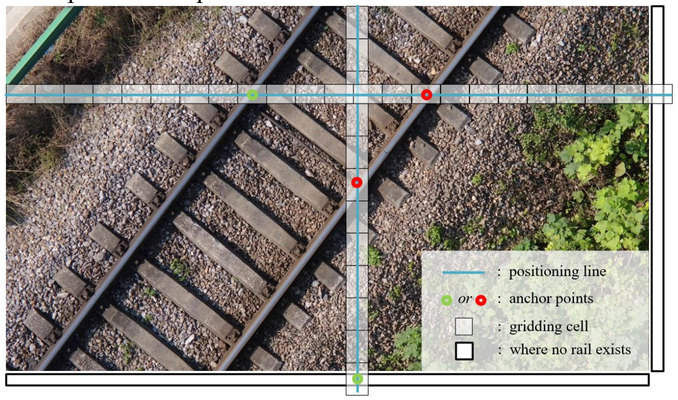


Fig. 3. Rail representation based on positioning lines and anchor points. Anchor points corresponding to gridding cells existing in the positioning lines indicate locations for rails.

More specifically, suppose the height and width of the image are h and w respectively, it is needed firstly to select a direction of the positioning line to be used, i.e., horizontal or vertical. The number of positioning lines in the horizontal or vertical direction is denoted as the sample dimension d_s. The number of gridding cells that discretize the positioning lines is denoted as the gridding dimension d_g+1. Among all the d_g+1 cells for one positioning line, the best anchor point can always be found in that positioning line to represent the location of each rail. The number of positioning lines used for sampling rails is far smaller than the image size, i.e., $d_s \ll h$ and $d_s \ll w$, which brings superior advantages to the reduction of the amount of calculation. All the rails can be represented with these $d_s\times(d_g+1)$ anchor points and each rail can be represented by d_s uniformly sampled anchor points as shown in Fig. 4. Then the j-th rail can be formulated as a set of anchor points, denoted as $P_j = \{Loc_{i,j}\}$, in which i=1, 2, …, d_s and $Loc_{i,j} \in \{1, 2, \ldots, d_g + 1\}$. Further, since different directions of positioning lines lead to different representations, rails under horizontal and vertical positioning lines are denoted as P_j^h and P_j^v respectively.

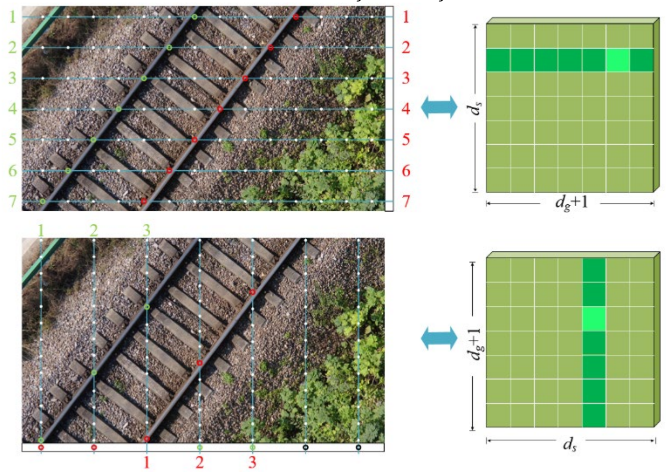


Fig. 4. The RRM-PLD with d_s = 7 and d_g = 14 and illustration of the one-to-one mapping from the anchor points to the final customized feature maps for proposal generation.

Thus, the same rail can be represented by two groups of anchor points P_j^h and P_j^v, the number of which is d_s for each group. However, not all these d_s anchor points are valid because of the existence of invalid anchor points that indicates no rail exists in the current row or column. The valid number of anchor points in the representation group is defined as the projection length, denoted as ℓ_j^h and ℓ_j^v corresponding to horizontal and vertical positioning lines respectively. For example, as shown in the left two images in Fig. 4 with a sample dimension d_s being 7, the valid projection length for the rails based on the horizontal positioning lines is 7 and the valid projection length based on the vertical positioning line is 3. It is believed that the larger the projection length is, the more sampled valid points there are, and then the closer it is to the real shape of the rail. Hence, the final anchor points-based representation for the j-th rail can be determined by projection length discrimination:

$$\ell_j = \max\left\{\ell_j^h,\ \ell_j^v\right\},\ P_j = \begin{cases} P_j^h,\ if\ \ell_j^h \geq \ell_j^v \\ P_j^v,\ if\ \ell_j^h < \ell_j^v \end{cases} \tag{1}$$

But if there exist N rails at most in the image, then all N rails can be represented by (P_1, P_2, …, P_N). Because all the N rails are involved in the model training process, the projection length for the multiple rails in a whole image is determined by:

$$\ell_h = \frac{1}{N}\sum_j \ell_j^h,\ \ell_v = \frac{1}{N}\sum_j \ell_j^v \tag{2}$$

$$\ell = \max\left\{\ell_h, \ell_v\right\},\ P_j = \begin{cases} P_j^h,\ if\ \ell_h \geq \ell_v \\ P_j^v,\ if\ \ell_h < \ell_v \end{cases} \tag{3}$$

Hence, the final projection length for the whole image in Fig. 4 is 7 and the representation with the horizontal positioning lines is more suitable to complete the recognition task. To satisfy this representation of rails, the projection lengths of the rails of different representation directions are supposed to be predicted to realize adaptive rail representation.

Additionally, the RRM-PLD has also designed a one-to-one mapping relationship established to accomplish the global rail recognition across the whole images between anchor points and the final extracted feature maps which are used to perform classification (selecting the best location along the direction of the positioning lines), as illustrated in Fig. 4. With this simple but effective one-to-one mapping, a real-time attention-aware rail recognition network is proposed.

B. Attention-Aware TriRNet Architecture

The relative spatial locations across different rails are important for rail recognition. Generally, the features of a single rail are locally limited, so it is expected that a more global and sufficient feature can be obtained to realize the discrimination of the geometric features of the rails. To determine which rail representation direction is more adaptive and representative of the currently processing image, the inter-rail attention (IRA) module is proposed to complete the spatially geometric feature discrimination of rails, which can detect rails by not only exploiting the feature of the current rail but also absorbing the features of other extra rails in the image.

As depicted in Fig. 5, the backbone makes use of the most commonly used ResNet [47] to extract rich global features from the input railway scene UAV aerial image, generating a down-sampled multi-layer feature map $F_{back} \in \mathbb{R}^{C' \times H \times W}$. To reduce the computing cost of the forward inference, an additional 1×1 convolution is attached to F_{back}, producing thinner feature maps $F_R \in \mathbb{R}^{C\times H\times W}$. F_R is then flattened and reshaped as a linear vector $F_{Lin} \in \mathbb{R}^{C\cdot H\cdot W}$. Then F_{Lin} is further functioned by a linear space transformation (also called fully connected layer) to generate a new global feature vector $F_{glob} \in \mathbb{R}^{d_s\cdot(d_g+1)\cdot N}$, which can be described by:

$$F_{glob} = \boldsymbol{W}_0 F_{Lin} + \boldsymbol{b}_0 \tag{4}$$

in which $\boldsymbol{W}_0 \in \mathbb{R}^{d_s\cdot(d_g+1)\cdot N \times C\cdot H\cdot W}$ and $\boldsymbol{b}_0 \in \mathbb{R}^{d_s\cdot(d_g+1)\cdot N}$ hold. Finally, the F_{glob} is exploited to realize the attention mechanism and the final rail recognition based on the anchor points representation.

Through a reasonable and artful design, this paper establishes a one-to-one mapping between anchor points and the final feature maps for classification along the direction of the positioning lines. However, anchor points can be represented in two ways with horizontal or vertical positioning lines, respectively. As shown in Fig. 4, these two representations correspond to two different dimension patterns of the feature maps, respectively. The dimension of the anchor points and the corresponding feature map for each rail is $d_s\times(d_g+1)$ with the horizontal positioning lines. Similarly, the dimension with the vertical positioning lines for each rail is $(d_g+1)\times d_s$. F_{glob} is reshaped to F_H and F_V at the same time, where $F_H \in \mathbb{R}^{d_s\times(d_g+1)\times N}$ and $F_V \in \mathbb{R}^{(d_g+1)\times d_s\times N}$ always hold.

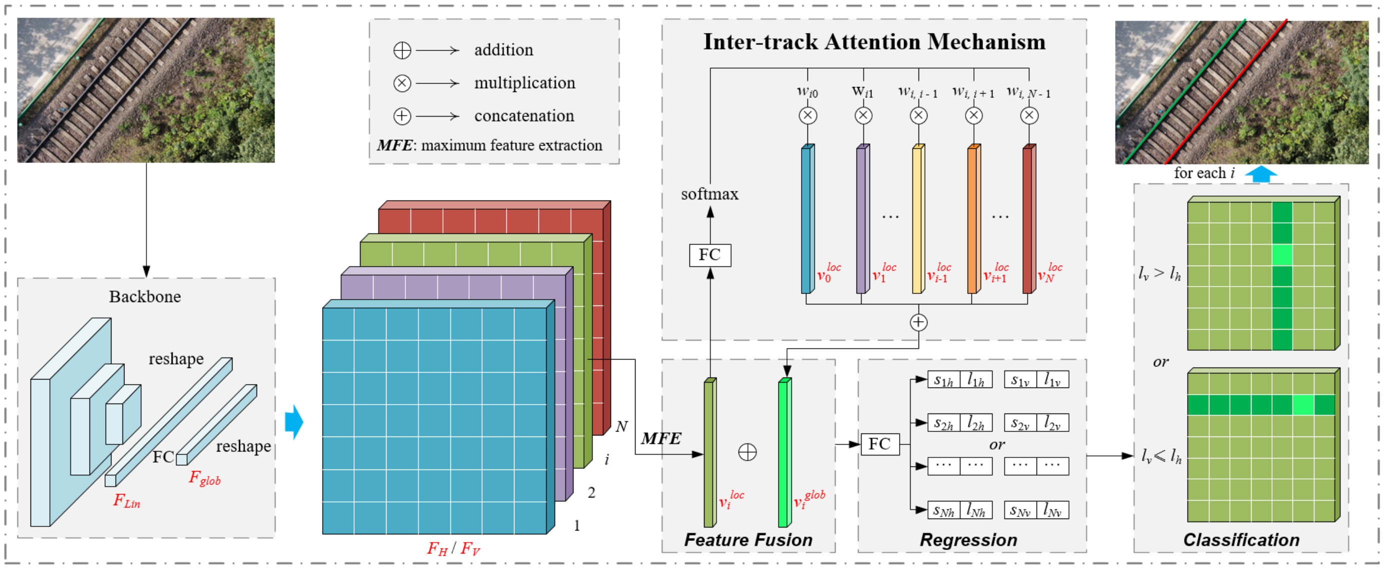


Fig. 5. The proposed deep convolutional network architecture TriRNet for UAV-based rail recognition.

There are two purposes for F_H and F_V: (1) to generate the subsequent global feature vectors by inter-rail attention module which are utilized to regress the starting location and the projection length for each rail; (2) to complete the rail proposal generation process according to the corresponding direction of positioning lines. The regressed projection lengths are supposed to discriminate the geometric features of the rails, which helps to choose a better rail representation direction. The feature maps F_H and F_V are supposed to be classified along the direction of the corresponding positioning lines. The locations of the anchor points corresponding to the largest feature value are selected. These locations can correspond one-to-one with specific pixels in the image and finally generate the predicted rail proposal. F_H and F_V exist in the network at the same time and both participate in the training process. F_{glob} generates F_H and F_V at the same time, thus forming a special dual-branch architecture (DBA).

There are two forms of maximum feature extraction (MFE) methods corresponding to the two representation directions. The maximum local feature vector $\boldsymbol{v}_i^{loc}$ corresponding to the i-th feature map of F_H or F_V can be obtained by the MFE process. However, no matter which representation method is used to obtain the final feature maps, i.e., F_H and F_V, the maximum local feature vector $\boldsymbol{v}_i^{loc}$ can be obtained according to the direction of its positioning line:

$$\begin{cases} v_{ij}^{loc} = \max_k F_{H;j,k} \\ v_{ij}^{loc} = \max_k F_{V;k,j} \end{cases} \tag{5}$$

in which $j = 1,2,...,d_s$, $k = 1,2,...,d_g + 1$, and $\boldsymbol{v}_i^{loc} \in \mathbb{R}^{d_s}$ is a local row feature vector, denoted as $\boldsymbol{v}_i^{loc} = [v_{i1}^{loc}, v_{i2}^{loc}, ..., v_{id_s}^{loc}]$. $\boldsymbol{v}_i^{loc}$ is then exploited to conduct feature fusion with the extra $N-1$ local feature vectors in a weighted way. All weights are obtained by a softmax process functioned onto another linear space transformation layer L_{att}, which transforms the d_s-dimensional feature vector $\boldsymbol{v}_i^{loc}$ to $N-1$ weight values which are finally used for inter-rail attention calculation:

$$L_{att}\left(\boldsymbol{v}_i^{loc}\right) = \tilde{\boldsymbol{W}}\boldsymbol{v}_i^{loc} + \tilde{\boldsymbol{b}} \tag{6}$$

in which $\tilde{\boldsymbol{W}} \in \mathbb{R}^{(N-1)\times d_s}$ and $\tilde{\boldsymbol{b}} \in \mathbb{R}^{N-1}$. Then the weights for $\boldsymbol{v}_i^{loc}$ to output a global feature vector $\boldsymbol{v}_i^{glob}$ can be computed as:

$$w_{i,j} = \begin{cases} softmax\left(L_{att}\left(\boldsymbol{v}_i^{loc}\right)\right)_j, & if\ j < i \\ 0, & if\ j = i \\ softmax\left(L_{att}\left(\boldsymbol{v}_i^{loc}\right)\right)_{j-1}, & if\ j > i \end{cases} \tag{7}$$

where $i = 1,2,...,N$ and $j = 1,2,...,N$. The softmax activation function is utilized to create a normalized probability distribution that can be used for conducting weighted addition operations on multiple local feature vectors. Then the global rail attention feature vectors can be calculated by:

$$\boldsymbol{v}_i^{glob} = \sum_j w_{i,j}\boldsymbol{v}_j^{loc} \tag{8}$$

where $i = 1,2,...,N$, $j = 1,2,...,N$, and $\boldsymbol{v}_i^{glob} \in \mathbb{R}^{d_s}$ is a global row feature vector with the same dimension as $\boldsymbol{v}_i^{loc}$. Let $\mathbf{V}^{loc} = \left[\boldsymbol{v}_1^{loc}, ..., \boldsymbol{v}_N^{loc}\right]$ be the local feature matrix containing all local row feature vectors and $\mathbf{W} = \left(w_{i,j}\right)_{N\times N}$ be the calculated weight matrix denoted in (7). Then the global features obtained by the inter-rail attention module can be formulated as:

$$\mathbf{V}^{glob} = \mathbf{W}\mathbf{V}^{loc} \tag{9}$$

in which $\mathbf{V}^{loc} \in \mathbb{R}^{N\times d_s}$, $\mathbf{W} \in \mathbb{R}^{N\times N}$ and $\mathbf{V}^{glob} \in \mathbb{R}^{N\times d_s}$. As can be seen, $\mathbf{V}^{loc}$ and $\mathbf{V}^{glob}$ have the same matrix dimensions. In general, the inter-rail attention mechanism proposed in this paper is a series of linear space transformations across all the local feature vectors $\boldsymbol{v}_i^{loc}$. The attention mechanism can effectively fuse all local features to generate global features $\boldsymbol{v}_i^{glob}$ across different rails, which is more beneficial to take the global context information of the whole image into account. In this way, the focus of the network can be concentrated on those key anchor points of interest and thereby improving the recognition accuracy of rails. The $\boldsymbol{v}_i^{loc}$ and $\boldsymbol{v}_i^{glob}$ are finally concatenated to perform the regression of the geometric parameters, i.e., start location and projection length of the rails in the image. The regression process is formulated as:

$$(\hat{s}_i, \hat{\ell}_i) = \mathrm{FC}(\boldsymbol{v}_i^{loc} \oplus \boldsymbol{v}_i^{glob}) \tag{10}$$

C. Proposal Generation

As denoted in (5), the local feature vectors $\boldsymbol{v}_i^{loc}$ are obtained from F_H and F_V. Also, the final predicted rail proposals are to be generated from these two feature maps in a similar principle. For the possible N rails existing in an image to be detected, each rail is connected to a feature map layer in F_H and F_V. For F_H, the predicted locations for the i-th rail can be formulated as:

$$Loc_{H;\,i,j} = \arg\max_k F_{H;\,j,k} \tag{11}$$

Combining the regressed starting location $\hat{s}_{ih}$ and projection length $\hat{\ell}_{ih}$, the predicted rail represented by horizontal anchor points can be determined. For F_V, the predicted locations for the i-th rail can be formulated as:

$$Loc_{V;\,i,j} = \arg\max_k F_{H;\,k,j} \tag{12}$$

Also, the predicted rail represented by vertical anchor points can be determined with the regressed starting location $\hat{s}_{iv}$ and projection length $\hat{\ell}_{iv}$. Thus, two kinds of different predicted rail proposals can be obtained by formulas (11) and (12). As mentioned before, the final proposals are decided by projection length discrimination. Suppose the mean projection lengths corresponding to the horizontal and vertical anchor points based representations are $\hat{\ell}_h$ and $\hat{\ell}_v$, respectively. Then they can be calculated as:

$$\hat{\ell}_h = \frac{1}{N}\sum_i \hat{\ell}_{ih},\ \hat{\ell}_v = \frac{1}{N}\sum_i \hat{\ell}_{iv} \tag{13}$$

The projection length discrimination derived from the inter-rail attention module is then applied. If $\hat{\ell}_h \geq \hat{\ell}_v$ holds, then (11) can be exploited to produce the final rail proposals; if $\hat{\ell}_h < \hat{\ell}_v$ holds, then (12) is exploited to produce the final proposals.

D. Loss Design and Model Training

As discussed before, this paper constructs a dual-branch architecture based on inter-rail attention regression, projection length discrimination, and one-to-one mapping from represented anchor points to final extracted feature maps. With the dual-branch architecture, the training of the two branches of the network is implemented in a weighted manner, and the weights are calculated according to the corresponding two kinds of regressed projection lengths, denoted as:

$$w_h = e^{\hat{\ell}_h} \Big/ \left(e^{\hat{\ell}_h} + e^{\hat{\ell}_v}\right),\ w_v = e^{\hat{\ell}_v} \Big/ \left(e^{\hat{\ell}_h} + e^{\hat{\ell}_v}\right) \tag{14}$$

In this way, the computed losses of the two branches can be weighted by w_h and w_v when performing loss back propagation during the network training process. With this weighted training mode, it is obvious that the angular distribution of rails in the image has an important influence on the training of both branches. When the inclination angles of rails distributed in the training dataset are closer to being horizontal, the training of the vertical branch that corresponds to F_V is more effective; when the inclination angles distributed in the training dataset are closer to being vertical, then the training of the horizontal branch that corresponds to F_H is more effective. In fact, the branch that corresponds to a larger projection length and representation direction always gets a larger weight to update its parameters. Basically, in the existing datasets, the angular distribution of the rails cannot be completely uniform, which is also confirmed by the statistics of the datasets used in this paper. Therefore, this paper adopts a balanced transpose co-raining strategy (BTCS) to effectively train both branches of the network in a more balanced way, so as to adapt to the uneven distribution of the inclination angles of rails existing in the images in the dataset.

Now given an image I, consider the image as a matrix, and its transposed image is denoted as IT. If the geometric direction

Algorithm 1: Training and testing of TriRNet

```
Input: Training or testing set $\mathcal{X}$ and its rail label set $\mathcal{T}$:
    $(L^h, L^v)$, in which $L_k^h = \{loc_{i,j}^h\}$ and $L_k^v = \{loc_{i,j}^v\}$, $i$ =
    1, …, $N$ and $j$ = 1, …, $d_s$. $N \leftarrow$ max number of rails.
Output: Network parameters $\Theta$.
1. Let $\eta_0 \leftarrow$ initial learning rate.
2. Let backbone mapping function $\mathcal{F}_0$: $I \rightarrow (F_H, F_V)$, in which
    $F_H \in \mathbb{R}^{d_s \times (d_g+1) \times N}$ and $F_V \in \mathbb{R}^{(d_g+1) \times d_s \times N}$.
3. for $epoch$ in range ($N_{epoch}$) do
4.   for image and rail label $\{n, (I, L^h, L^v)\}$ in $(\mathcal{X}, \mathcal{T})$ do
5.     Let $I^T$ and $(L^{hT}, L^{vT}) \leftarrow$ the transpose of $I$ and its rail
       label derived from $(L^h, L^v)$.
6.     step = $|\mathcal{X}| \cdot epoch + n$
7.     Let $\ell_h, \ell_v \leftarrow$ the mean num of valid points in $L^h$, $L^v$.
8.     $(F_H, F_V) = \mathcal{F}_0(I)$, $(\mathbf{V}_H^{loc}, \mathbf{V}_V^{loc})$ = MFE$(F_H, F_V)$
9.     $(\mathbf{V}_H^{glob}, \mathbf{V}_V^{glob})$ = IRA$(\mathbf{V}_H^{loc}, \mathbf{V}_V^{loc})$
10.    $(\tilde{\mathbf{V}}_H, \tilde{\mathbf{V}}_V) = (\mathbf{V}_H^{glob}, \mathbf{V}_V^{glob}) \oplus (\mathbf{V}_H^{loc}, \mathbf{V}_V^{loc})$
11.    $(\hat{\ell}_h, \hat{\ell}_v) = mean[\mathrm{FC}(\tilde{\mathbf{V}}_H, \tilde{\mathbf{V}}_V)]$  // fully connected layer
12.    $Loc_{H;\,i,j} = \arg\max_k F_{H;\,j,k}$, $Loc_{V;\,i,j} = \arg\max_k F_{H;\,k,j}$
13.    if testing then
14.      output rail proposal $Loc_H$ if $\hat{\ell}_h > \hat{\ell}_v$ else $Loc_V$
15.      continue
16.    end if
17.    $w_h = e^{\ell_h} / \left(e^{\ell_h} + e^{\ell_v}\right)$, $w_v = e^{\ell_v} / \left(e^{\ell_h} + e^{\ell_v}\right)$
18.    $\mathcal{L}_{cls}^O \leftarrow w_h \cdot classify(Loc_H, L^h) + w_v \cdot classify(Loc_V, L^v)$
19.    $\mathcal{L}_{reg}^O = smoothL1(\hat{\ell}^h, \ell^h) + smoothL1(\hat{\ell}^v, \ell^v)$
20.    Repeat 7-19 for $I^T$, $(L^{hT}, L^{vT})$ to obtain $Loc_H^T$, $Loc_V^T$,
       $\mathcal{L}_{cls}^T$, $\mathcal{L}_{reg}^T$.
21.    $\mathcal{L}_{cls} = (\mathcal{L}_{cls}^O + L_{cls}^T)/2$, $\mathcal{L}_{reg} = (\mathcal{L}_{reg}^O + \mathcal{L}_{reg}^T)/2$
22.    $\mathcal{L}_{TC} \leftarrow mean(\|Loc_H - Loc_V^T\| + \|Loc_V - Loc_H^T\|)$
23.    $\mathcal{L} = \alpha\mathcal{L}_{cls} + \beta\mathcal{L}_{reg} + \gamma\mathcal{L}_{TC}$, $\nabla\Theta_t = bp(\mathcal{L})$
24.    $\eta \leftarrow LRScheduler(steps, \eta_0)$, $\Theta_{t+1} = \Theta_t - \eta\nabla\Theta_t$
25.  end for
26. end for
27. return $\Theta$
```

of the rails in the image I is closer to being horizontal, then the vertical branch in the network will be trained more effectively. But if I^T is also input into the network for training, the horizontal branch in the network can also be trained to the same extent. On the other hand, if the geometric direction of the rails in the image is closer to being vertical, then the horizontal branch in the network will be trained more effectively. But if I^T is also fed into the network for training, then the vertical branch in the network can also be trained to the same extent in the same way. Therefore, this paper proposes to input the original image and the transposed image into the network simultaneously when training the network. In this way, the weights w_h and w_v are just reversed, and both branches have more balanced reasoning capabilities.

In fact, based on the labels of I and I^T, the horizontal representation of the image I should be completely consistent with the vertical representation of the image I^T. Similarly, the vertical representation of the image I should be completely consistent with the horizontal representation of the image I^T. However, based on the predictions of I and I^T, there exist specific differences between them. These differences should be gradually reduced during the training process. Based on this difference, we propose the transpose consistency loss (TCL) function. In this paper, the loss consists of three parts: classification loss, regression loss, and transpose consistency loss.

Classification loss. As denoted in (11) and (12), the F_H and F_V are the final feature maps adopted to generate the rail proposals through classification. Thus, both the horizontal branch and the vertical branch have their own classification results, as well as classification losses. The two kinds of classification losses for the original input image are combined in a weighted way and formulated as follows:

$$\mathcal{L}_{cls}^{O} = w_h \mathcal{L}_{clsh} + w_v \mathcal{L}_{clsv} \tag{15}$$

in which $\mathcal{L}_{cls}^{O}$ is the general classification loss. $\mathcal{L}_{clsh}$ and $\mathcal{L}_{clsv}$ are the classification loss corresponding to the two branches, respectively. Similarly, the classification losses for the transposed image are computed as follows:

$$\mathcal{L}_{cls}^{T} = w_h \mathcal{L}_{clsh}^{T} + w_v \mathcal{L}_{clsv}^{T} \tag{16}$$

Then, the final classification loss is defined as the mean of the classified loss for the original image and the transposed one.

$$\mathcal{L}_{cls} = (\mathcal{L}_{cls}^{O} + \mathcal{L}_{cls}^{T}) / 2 \tag{17}$$

Regression loss. As discussed before, the local feature vector $\boldsymbol{v}_i^{loc}$ and the global feature vector $\boldsymbol{v}_i^{glob}$ are finally concatenated to regress the start location $\hat{s}_i$ and the projection length $\hat{\ell}_i^h$. Here the Sooth L1 loss function is adopted to achieve the regression process. The general regression loss $\mathcal{L}_{reg}^{O}$ is computed as:

$$\mathcal{L}_{reg}^{O} = \mathcal{L}_{regh} + \mathcal{L}_{regv} \tag{18}$$

in which $\mathcal{L}_{regh}$ and $\mathcal{L}_{regv}$ are the regression loss corresponding to the two branches, respectively. Similarly, the counterpart regression loss for the transposed image is computed as follows:

$$\mathcal{L}_{reg}^{T} = \mathcal{L}_{regh}^{T} + \mathcal{L}_{regv}^{T} \tag{19}$$

Consequently, the final regression loss for the start location and the projection length is defined as the mean of the regression loss for the original input image and its transposed counterpart.

$$\mathcal{L}_{reg} = (\mathcal{L}_{reg}^{O} + \mathcal{L}_{reg}^{T}) / 2 \tag{20}$$

Transpose consistency loss. Along with the balanced transpose co-training strategy and based on the correspondence between the predicted rail proposals of I and I^T, a consistency loss can be formulated by modeling the distance between the predictions of I and I^T . Let $Loc_{H;\,i,j}$ and $Loc_{V;\,i,j}$ obtained from F_H and F_V denote the two predictions of the image I. Let $Loc_{H;\,i,j}^{T}$ and $Loc_{V;\,i,j}^{T}$ denote the two predictions of image I^T. Then the transpose consistency loss L_{TC} which is a second-order difference constraint can be computed as:

$$\mathcal{L}_{TC} = \frac{1}{Nd_s} \sum_{i=1}^{N} \sum_{j=1}^{d_s} \left(\left\| Loc_{H;\,i,j} - Loc_{V;\,i,j}^{T} \right\| + \left\| Loc_{V;\,i,j} - Loc_{H;\,i,j}^{T} \right\| \right) \tag{21}$$

in which $Loc_{i,j}$ and $Loc_{i,j}^{T}$ are the predicted j-th sampled point of the i-th rail for image I and I^T respectively. It is worth noting that the transpose consistency loss is constructed based on the above-mentioned balanced transpose co-training strategy.

Then, an integrated structural loss function is presented as:

$$\mathcal{L} = \alpha \mathcal{L}_{cls} + \beta \mathcal{L}_{reg} + \gamma \mathcal{L}_{TC} \tag{22}$$

in which α, β and γ are loss coefficients. In this paper, this integrated loss function is finally exploited to train the whole dual-branch architecture. The three coefficients are all taken simply as 1.0 when used in the training process.

Pseudo code for training and testing of TriRNet. A detailed pseudo code for the training and testing of TriRNet is given in Algorithm 1. The main steps for forward inference, the strategy for the use of the transposed images, the computation of the proposed structural loss, and the learning rate scheduler are involved.

IV. EXPERIMENTS

A. Datasets and Parameter Setting

To evaluate the effectiveness and overall performance of the proposed architecture, this paper establishes a new dataset on rail recognition. All the images are collected by a multi-rotor UAV DJI M300 RTK carrying a multi-sensor integrated payload DJI Zenmuse H20T. The flight height of the UAV relative to the railway plane is usually set at 50-80m and a lateral distance of 10-20m is also adopted to avoid the accidental drop of the UAV on the railway line and thus ensure the safety of railway normal operations. The moving speed of the UAV is usually set to 1-3m/s to ensure the good quality of the collected images. All images used in the experiments are taken from Beijing-Shanghai high-speed railway and several ordinary railway scenes under this distance and moving speed setting. The constructed dataset includes 1116 images in all, in which the training set includes 893 images and the test part includes 223 images. The datasets contain abundant images with changeable pixel width and inclination angles of rails to enhance the generalization ability of the proposed models. Also, the images in the dataset have various changing backgrounds, as shown in Fig. 6, for the sake of a fairer evaluation of the model capabilities.

The proposed TriRNet and its incomplete versions in *Ablation Study* are all trained with the same hyperparameter setting. Their training is implemented by Adam Optimizer with a momentum of 0.9. The batch size, initial learning rate, and epoch are 16, 10^{-2}, and 500, respectively. If the initial weights of the network are transferred from another trained model, then the initial learning rate is set to 10^{-4}. The learning rate is adjusted dynamically during training by a multi-step learning rate scheduler. It is reduced to one-tenth and one-hundredth of the initial learning rate at the 25th and 38th training steps. All neural network modules involved in this paper are implemented in PyTorch deep learning framework.

B. Evaluation Metric

F1 measure is the most used metric in the lane detection task of road traffic. For the evaluation of different rail recognition algorithms, the F1 measure is adopted, which is defined as:

$$F1 = \frac{2 \times P \times R}{P + R} \tag{23}$$

where $P = \frac{TP}{TP + FP}$, $R = \frac{TP}{TP + FN}$.

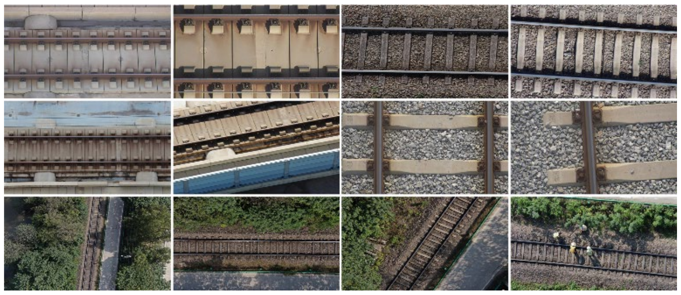


Fig. 6. Various kinds of railway scenes from the perspective of UAVs, in which rails can have utterly different pixel widths, inclination angles, and changing backgrounds.

As depicted in Fig. 7, a rail proposal of TriRNet is composed of a series of classified and activated anchor points. For a single anchor point, it is classification. But for a whole rail proposal, it is composed of a series of activated anchor points. Given a predefined pixel width, these grouped points are then considered as a whole to perform IoU calculation with the corresponding ground truth rail target, which is also composed of a series of sampled points. In CULane [24] lane detection dataset, each lane is treated as a 30-pixel-width line. In this paper, all rails are treated as 28-pixel-width lines, considering that 28 pixels are the 3/4 quantile in the statistics counted on the pixel widths of all rails in images of the whole dataset. As shown in Fig. 7, The IoU value between the predicted rail proposal and the corresponding ground truth is calculated. When the computed IoU is larger than θ, the proposal is treated as true positives (TP); when the IoU is smaller than θ, the proposal is then treated as false positives (FP). The rails actually exist but are not detected are counted as false negatives (FN). θ represents the IoU threshold and is usually taken as 0.5 in the CULane lane detection dataset. Here the mF1 measure [46] is finally adopted to evaluate the algorithms, which is defined as:

$$mF1 = \left(F1@30 + F1@50 + F1@75\right) / 3 \tag{24}$$

where F1@30, F1@50, and F1@75 are F1 metrics when IoU thresholds θ are taken as 0.3, 0.5, and 0.75, respectively. The reasonable range for mF1 and other F1 metrics (including the adopted F1@30, F1@50, and F1@75) are expected to be distributed between 0 and 1.0. For ease of comparison, they are all scaled between 0 and 100 in the following sections, which is 100 times the original values.

C. Ablation Study

In this section, a series of ablation studies on the network designs are performed. Starting from a baseline model, the effectiveness of the inter-rail attention (IRA) design is first presented. Then the two dimensions, i.e., the sampling dimension and the gridding dimension of the RRM-PLD, are investigated. At last, the effects of the proposed designs of the network and the two dimensions of the rail representation on the inference speed of the network are studied.

1) Ablation on the Inter-Rail Attention Mechanism

Starting from a baseline [43], which only applies one branch to conduct the rail recognition, the design of DBA and IRA on the network architecture and the design of BTCS and TCL on the model training are all discussed. From the designing process of the proposed architecture, it can be easily concluded that the BTCS can only be applied when the DBA is added to the architecture, and the TCL can only be applied when the BTCS is adopted in the network training process. The IRA design is completely independent of the other designs and can be applied to the network whenever needed.

As shown in the top three rows of Table I, the original baseline models with different backbones are presented. The results show that the capability of the baseline model is far from satisfactory and this is caused by the architecture itself, not by the ability of the backbones. Then the designs of DBA, BTCS, and TCL are gradually applied to the baseline models, as listed in the next three rows in Table I. As can be seen, the mF1 measure is increased from the original 12.75 to 51.35, 59.90, and 67.09 respectively, which demonstrates the effectiveness of the DBA, BTCS, and TCL design. Next, the IRA module is further added to the architecture. At first, only the DBA and the IRA are adopted. The mF1 value has been greatly improved compared to the baseline series and outperforms the model that only applies the DBA design, which is illustrated in row 7 of Table I. The results also imply the effectiveness of the single IRA design. Then the four designs are applied to the architecture, as shown in row 8 of Table I, the mF1 measure is increased to 66.72, which is very close (almost equal) to row 6 without transferring the network weights of the trained model in row 5. Because TCL is a kind of unsupervised loss function that can bring unexpected fluctuations to the network training process, transferring pre-trained network weights, and training with a lower learning rate can reduce the adverse effect of this fluctuation on the network training process. Therefore, both row 6 and row 9 take advantage of the weights transferred from the trained model of row 5. With all these four designs integrated into the architecture and initializing the network with the transferred weights, the accuracy of rail recognition has reached a higher level of 73.68. And the F1@30, F1@50, and F1@75 all reach a higher level, i.e., 85.49, 79.25, and 56.29 respectively, compared to other formulations. DBA, BTCS, and TCL are three basic designs to formulate the whole rail architecture while IRA performs as an enhancement module to improve the recognition accuracy in an attention-aware way. Rows 4-8 also indicate that the backbone resnet18 has enough capability to conduct rails recognition for UAV aerial images.

2) Ablation on the Sampling Dimension

As discussed in the *Methodology* section, a one-to-one mapping from the anchor points and the final extracted feature

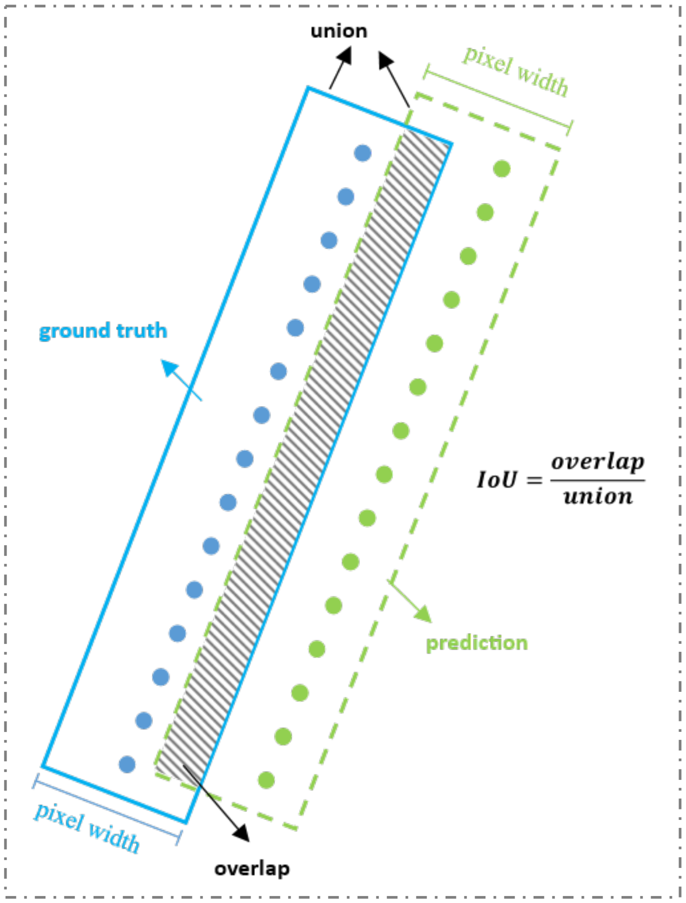


Fig. 7. Illustration of the IoU calculation between the predicted proposal and corresponding ground truth. The proposal is treated as TP / FP case when IoU ≥ θ/ IoU < θ.

maps which are used to perform classification is established as illustrated in Fig. 4, in which there exist two different dimensions to locate the sampled rail points and indicate the size of final feature maps. One is the sampling dimension, denoted as d_s and the other is the gridding dimension, denoted as d_g. Ablations on these two dimensions have been investigated. These two parameters determine the density at which the rails in the image are sampled and the number of grids to locate them. It is concluded that the larger d_s is, the closer the anchor point-based representation is to the real shape of rails; the larger d_g is, the more precise the locating of the rails along the direction of positioning lines. However, the larger d_s and d_g are, the more parameters there are in the final feature maps. This means longer latency to classify in the forward inference process and lower optimization efficiency in the training process. Therefore, reasonable settings of d_s and d_g play an important role in parameter optimization, reasoning accuracy, and inference efficiency.

The ablation on the sampling dimension is first presented. Because of the possible fluctuations that the unsupervised TCL design may bring to the network training, only the designs of DBA, BTCS, and IRA are employed in the network, the training of which does not need to transfer any weights from a trained model. As presented in Table II, all the models are trained with backbone resnet18 and initialized with a random parameter setting with the gridding dimension 100. The sampling dimension is taken from 5 to 25 at an interval of 5. As seen from the table, all the mF1 and other F1 measures have improved as the sampling dimension increased from 5 to 20 and obtain the highest performance when d_s is taken as 20. But when the sampling dimension reaches 25, the performance of the network begins to drop significantly from the highest 68.9 to 55.0. Therefore, it can be concluded that it is not that the larger the sampling dimension is, the better performance the trained model will have. Only an appropriate sampling dimension setting can make the network have a good performance as expected.

3) Ablation on the Gridding Dimension

Similarly, the ablation study on the gridding dimension is also conducted. The results are shown in Table III. Also, all the models are trained with backbone resnet18, sampling dimension 10, and random parameter setting for initialization. Only the designs of DBA, BTCS, and IRA are adopted for the same reason as mentioned above.

The gridding dimension is taken from 50 to 250 at an interval of 50. As illustrated in Table III, the mF1 and other F1-related measures of the network all show a trend of increasing first and then decreasing similar to the ablation study on the sampling dimension. All the F1-related measures achieve the highest when the gridding dimension is taken as 200, which are 78.3, 65.1, 29.9, and 57.8 for F1@30, F1@50, and F1@75 respectively. However, all these metrics start to drop when the gridding dimension is increased to 250. Therefore, it is not that the larger the grid dimension, the more beneficial it is to improve the network capability. It is analyzed that a quite large gridding dimension can lead to insufficient optimization of network parameters and thus influence the final extracted features for classification although it can locate the rails along the direction of the positioning lines more precisely in theory. Thus, an appropriate gridding dimension also plays a crucial role in improving the network inference capability.

4) Ablation on the Network Inference Speed

The proposed architecture is supposed to run on the onboard edge computing devices which are carried on a UAV to make real-time intelligent reasoning of video streams passed from the camera mounted on the UAV. Therefore, the inference speed of the network is very important for the onboard deployment of the trained model. The speed evaluation in this paper does not use any C++ extensions and acceleration toolkits or libraries such as TensorRT.

The effect of several customized designs on the network inference speed is studied first, as shown in Table IV. The experiments are conducted on an NVIDIA Jetson Xavier NX and an NVIDIA Jetson TX2 which are both embedded edge computing devices respectively with a sampling dimension of 15 and a gridding dimension of 150. The inference speed of the baseline, the baseline with a single DBA design applied, the baseline with a single IRA design applied, and the baseline with both DBA and IRA designs applied are tested with different backbones, i.e., resnet18, resnet34, resnet50, and resnet101. The BTCS and TCL designs are not structural designs on the architecture and can only be used during the training process, so they will not influence the reasoning latency of the whole architecture and thus are not included in this ablation study part. As can be concluded from Table IV which gives a series of values on frames per second (FPS), the DBA and IRA design will cause certain latency to the original baseline models, in which the IRA design will slow down the inference speed more than the DBA design. The inference speed becomes slower as the larger backbones are adopted gradually. The performance of NX is far better than TX2, the reasoning speed of which is almost 3-4 times that of TX2. When the DBA and IRA are applied to the baseline simultaneously, the NX can still achieve an FPS of 33.9 with backbone network resnet18, which is sufficient to meet the practical needs for engineering deployment. And the TX2 platform can achieve an FPS of 8.9 under the same network configuration, which also has a good reference value for the engineering applications of the proposed networks. In general, both NX and TX2 have the expected real-time performance for our proposed architecture and are fully capable of meeting current deployment needs.

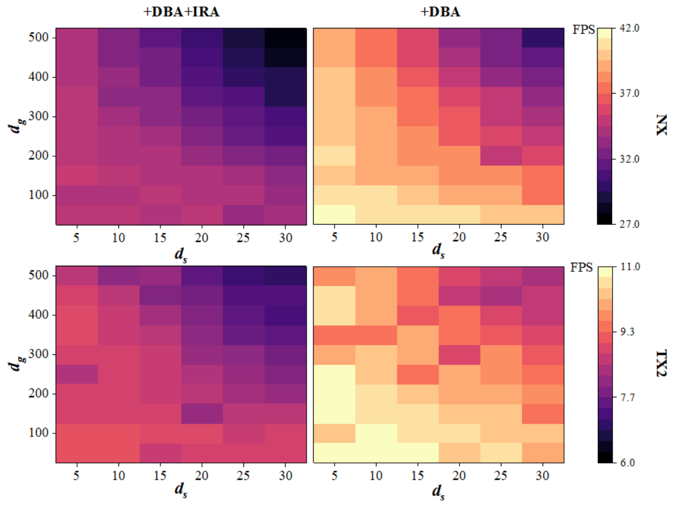


Fig. 8. The inference speed test results for both "+DBA+IRA" and "+DBS" settings on the embedded edge computing devices NVIDIA Jetson NX and TX2.

Further, the effect of different settings of sampling dimension and gridding dimension on network latency is also studied, as presented in Fig. 8. The FPS values of different dimension configurations on both NVIDIA Jetson Xavier NX and NVIDIA Jetson TX2 devices are reported in the form of heat maps. The top two heat maps are tested on NX and the bottom two matrices are tested on TX2. The two matrices on the left column are both with the "+DBA+ IRA" design and the ones on the right column are only with the "+DBA" design. As can be concluded from Fig. 8, within a certain range, the FPS value of the network has a linear negative correlation with the d_s and d_g basically. The inference speed of the network on the NX device is distributed in the interval of 27~42 FPS while the speed of the network on the TX2 device is distributed in the interval of 6~11 FPS, which further shows that the computing power of NX is much higher than that of TX2. Comparing the heat maps on the left and right sides in Fig. 8, applying the proposed IRA module to the network brings only a very limited increase in latency, and will not influence the real-time performance of the network greatly.

D. Comprehensive Comparison

This paper has also performed a comprehensive comparison with some other popular algorithms, i.e. SCNN [24], RESA [48], and UFLD [43], which are originally developed for lane detection tasks, in terms of recognition accuracy and network latency. In all experiments in this section, the two dimensions of the proposed network, d_s and d_g, are configured as 15 and 150, respectively.

There are many algorithms developed for lane detection tasks, but most of them are not suitable for rail recognition for images taken by UAVs in this paper. We also tried to directly use some more complicated approaches such as LaneATT [39], CondLane [45], PINet [35] for rail recognition, but the test results are far from meeting the requirements of the task. To be concrete, LaneATT backboned with resnet34 only achieves F1@30 of18.88 in the task, PINet only achieves F1@50 of 20.54, and the training of CondaLane fails to converge despite training for a long time.

It is analyzed that this may be caused by the great difference between the lane detection task based on the cameras mounted on cars and the rail recognition task based on UAV-mounted cameras. Firstly, the changing height of UAVs can cause large variations in the pixel width of rails in the image, resulting in large fluctuations in their scale presented in the image. Secondly, lanes in the images collected by the vehicle-mounted camera generally do not change too much, often the lane marker is rotated and translated slightly, but the rails in the images taken by UAVs may change greatly. The background may be another important factor. The background in the lane detection task is relatively simple, generally including vehicles and roads only, while the backgrounds for the rails in the UAV remote sensing images vary greatly which could be various and unpredicted especially when the picture is expanded to include the surrounding environment on both sides of the railway lines. Therefore, those popular algorithms are not suitable for the rail recognition task in this paper. UFLD is the baseline model used in this paper, SCNN and RESA are the most used segmentation-based methods, so they are selected as the final comparative models.

As shown in Table V, the mF1, F1@30, F1@50, and F1@75 are adopted to evaluate the recognition accuracy of the proposed attention-aware architecture and other comparative algorithms, i.e., UFLD, SCNN, and RESA. UFLD is also the baseline model adopted in this paper. The SCNN and RESA model series achieve almost the same level in terms of recognition accuracy. Their best-performing models achieve only mF1 scores of 37.85 and 37.77, respectively. It is analyzed that this is because the representation methods adopted in the SCNN and RESA models cannot adapt to the rails with variable inclination angles, especially those whose inclination angles are close to horizontal. Thus, they can only recognize only a part of the rails in the datasets with certain inclination angle distribution. UFLD models perform the worst among all these comparative algorithms. Despite that, our proposed approach based on this baseline obtains higher F1-related scores by adopting the proposed anchor points-based RRM-PLD, outperforming all other algorithms in terms of recognition accuracy. Concretely, the proposed TriRNets backboned by resnet18, resnet34, resnet50, and resnet101 achieve much higher mF1s, i.e., 73.68, 74.68, 74.83, and 73.70. Once the proposed architecture designs (DBA and ITA) and network training strategy designs (BTCS and TCL) are applied to the baseline model, the recognition accuracy of the network is greatly improved, which indicates the superiority of the proposed designs and formulation. Overall, the size of the backbone does not have a great impact on the performance of the network, and a larger backbone network will not bring greater gains to the network as expected. A small-size backbone resnet18 can satisfy the requirements for efficient feature extraction for the images in the constructed dataset.

The developed models are supposed to be deployed on the onboard computers carried by UAVs. Hence, the latency vs. accuracy diagram for the different algorithms is presented in Fig. 9 to give more intuitive observations on the comprehensive performance among all these model series. The top figure and the bottom figure correspond to the evaluation results on NX and TX embedded computing devices respectively. As depicted, these model series can be divided into three levels: low-level models, middle-level models, and high-level models. The low-level models contain the UFLD series, which cannot adapt to the rail recognition task. The middle-level models include SCNN and RESA series, which can recognize quite a part of the rails in the images in the test set. But they are still limited to the recognition of the rails that are not close to being horizontal in the image. Our proposed architecture is determined as a high-level algorithm because it outperforms other algorithms in terms of both recognition accuracy and inference latency and is completely suitable for practical deployment on the onboard

TABLE I
THE ABLATION STUDY RESULTS FOR THE PROPOSED DESIGNS: DUAL-BRANCH ARCHITECTURE (DBA), THE BALANCED TRANSPOSE CO-TRAINING STRATEGY (BTCS), THE TRANSPOSE CONSISTENCE LOSS (TCL), AND THE INTER-RAIL ATTENTION (IRA) MECHANISM WITH SAMPLING DIMENSION 15, GRIDDING DIMENSION 150.

No.	model	backbone	initialize	DBA	BTCS	TCL	ITA	F1@30	F1@50	F1@75	mF1
1	baseline	resnet18	Rand.	×	×	×	×	19.51	13.72	4.31	12.51
2	baseline	resnet34	Rand.	×	×	×	×	19.27	14.34	3.74	12.45
3	baseline	resnet50	Rand.	×	×	×	×	19.38	14.74	4.14	12.75
4	+DBA	resnet18	Rand.	✓	×	×	×	69.84	57.90	26.31	51.35
5	+DBA+BTCS	resnet18	Rand.	✓	✓	×	×	72.22	56.80	50.68	59.90
6	+DBA+BTCS+TCL	resnet18	Para.5	✓	✓	✓	×	82.09	72.51	46.66	67.09
7	+DBA+IRA	resnet18	Rand.	✓	×	×	✓	72.78	59.98	23.36	52.04
8	+DBA+BTCS+TCL+IRA	resnet18	Rand.	✓	✓	✓	✓	83.22	75.13	41.80	66.72
9	+DBA+BTCS+TCL+IRA	resnet18	Para.5	✓	✓	✓	✓	**85.49**	**79.25**	**56.29**	**73.68**

TABLE II
THE ABLATION STUDY RESULTS FOR THE SAMPLING DIMENSION WITH GRIDDING DIMENSION 100 WHEN DBA, BTCS, AND IRA ARE APPLIED TO THE ARCHITECTURE AND MODEL TRAINING. THE NETWORK PERFORMS BEST WHEN THE SAMPLING DIMENSION IS TAKEN AS 20.

No.	model	backbone	initialize	d_s	d_g	F1@30	F1@50	F1@75	mF1
1	+DBA+BTCS +IRA	resnet18	Rand.	5	100	82.5	71.3	10.5	54.8
2	+DBA+BTCS +IRA	resnet18	Rand.	10	100	77.6	64.2	23.5	55.1
3	+DBA+BTCS +IRA	resnet18	Rand.	15	100	85.0	77.0	42.7	68.2
4	+DBA+BTCS +IRA	resnet18	Rand.	20	100	**85.8**	**79.0**	41.9	**68.9**
5	+DBA+BTCS +IRA	resnet18	Rand.	25	100	73.5	61.1	30.3	55.0

TABLE III
THE ABLATION STUDY RESULTS FOR THE GRIDDING DIMENSION WITH SAMPLING DIMENSION 10 WHEN DBA, BTCS, AND IRA ARE APPLIED TO THE ARCHITECTURE AND MODEL TRAINING. THE NETWORK PERFORMS BEST WHEN THE GRIDDING DIMENSION IS TAKEN AS 200.

No.	model	backbone	initialize	d_s	d_g	F1@30	F1@50	F1@75	mF1
1	+DBA+BTCS +IRA	resnet18	Rand.	10	50	74.8	55.6	8.3	46.2
2	+DBA+BTCS +IRA	resnet18	Rand.	10	100	77.6	64.2	23.5	55.1
3	+DBA+BTCS +IRA	resnet18	Rand.	10	150	75.7	62.8	27.7	55.4
4	+DBA+BTCS +IRA	resnet18	Rand.	10	200	**78.3**	**65.1**	**29.9**	**57.8**
5	+DBA+BTCS +IRA	resnet18	Rand.	10	250	71.2	52.6	17.8	47.2

TABLE IV
THE FORWARD INFERENCE SPEED EVALUATION ON THE DBA DESIGN. "NX" MEANS NVIDIA JETSON XAVIER NX AND TX2 MEANS NVIDIA JETSON TX2. THE FRAMES PER SECOND (FPS) VALUES ARE REPORTED WITH SAMPLE DIMENSION 15, GRIDDING DIMENSION 150.

Backbone	Baseline		+DBA		+IRA		+DBA+IRA	
	TX2	NX	TX2	NX	TX2	NX	TX2	NX
ResNet18	10.9	40.5	10.5	39.5	9.6	36.7	8.9	**33.9**
ResNet34	6.5	23.0	6.0	22.7	5.9	21.7	5.5	20.6
ResNet50	3.3	12.7	3.3	12.6	3.0	12.1	2.9	11.7
ResNet101	1.9	7.3	1.9	7.2	1.8	7.1	1.7	6.8

NX or TX2. Despite the not-bad mF1 scores for SCNN and RESA, they still cannot be deployed directly onto the onboard computers because of their too-long inference latency on both NX and TX2 platforms. UFLD also cannot be employed to accomplish engineering applications due to its far too low recognition accuracy.

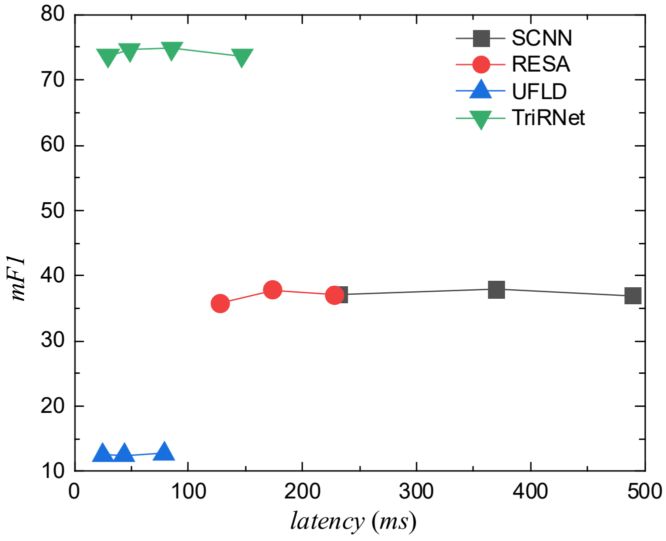
TABLE V
COMPREHENSIVE COMPARISONS BETWEEN SOME POPULAR ALGORITHMS WITH "VERTICAL SAMPLING + HORIZONTAL POSITIONING" REPRESENTATION AND THE PROPOSED ATTENTION-AWARE ARCHITECTURE WITH RRM-PLD.

model	backbone	F1@30	F1@50	F1@75	mF1
UFLD	ResNet18	19.51	13.72	4.31	12.51
UFLD	ResNet34	19.27	14.34	3.74	12.45
UFLD	ResNet50	19.38	14.74	4.14	12.75
SCNN	ResNet50	40.59	39.85	30.71	37.05
SCNN	ResNet101	41.32	39.85	30.71	36.81
SCNN	VGG16	48.52	46.05	18.98	37.85
RESA	ResNet18	40.51	40.15	26.64	35.77
RESA	ResNet34	41.17	40.44	31.69	37.77
RESA	ResNet50	40.80	40.43	29.87	37.03
TriRNet	ResNet18	85.49	79.25	56.29	**73.68**
TriRNet	ResNet34	87.98	80.27	55.78	**74.68**
TriRNet	ResNet50	89.34	80.05	55.10	**74.83**
TriRNet	ResNet101	87.30	79.37	54.42	**73.70**

E. Visual Quality Comparison

To demonstrate the superiority and excellent performance of the proposed algorithm more intuitively, several visual examples are given to compare with other models, as shown in Fig. 10. SCNN and RESA are backboned by resnet50 and resnet34 respectively. TriRNet adopts resnet50 as the backbone network. For the vertically or near-vertically distributed rails as shown in columns 1-3 in Fig. 10, all three models achieve good recognition results. However, for the horizontally or near-horizontally distributed rails as shown in the first four rows of columns 4-6 of Fig. 10, SCNN and RESA have fairly poor recognition performance. In contrast, the proposed TriRNet can still make accurate predictions of rails with near-horizontal inclination angles in UAV remote sensing images. It is analyzed that the recognition accuracy is limited by their corresponding representation methods and the algorithms themselves. Thus, these results have verified that SCNN and RESA are not suitable for rails with changing geometric distribution and inclination angles from the perspective of UAVs, although they have achieved good results in the lane detection task. Meanwhile, the results also verify the effectiveness of the proposed RRM-PLD based on anchor points and positioning lines and the proposed inter-rail attention-guided architecture. Rows 2-4 of columns 1-3 of Fig. 10 give samples where some rail recognition points are missing for the segmentation-based SCNN. This can be caused by the fact that the rails in this case have relatively small pixel widths, which can bring a very adverse impact on existing segmentation-based recognition methods.

Compared with SCNN and RESA, TriRNet also has good performance in detecting occluded rails with high accuracy, as illustrated in row 1 of columns 4-6 of Fig.10. The higher recognition accuracy further proves the effectiveness of the proposed IRA module and network architecture. It is analyzed that the IRA module can effectively extract the correlations across all rails in the image and fuse the extracted local features of a single rail and the calculated global features of all rails to obtain accurate identification of the geometric distribution of rails in the image. The last row of columns 4-6 gives an example that one rail is almost occluded completely by trees. The SCNN and RESA model can percept only one rail in the image with low accuracy. But our TriRNet can still recognize two rails due to the IRA mechanism although the recognition result is not very precise. the recognition accuracy of the proposed method in the case of tree occlusion still has a relatively large gap from the expectation which should be addressed in future works. It is worth mentioning that although most of the rails in the images of our dataset used in this paper are near-straight lines, this does not mean that the proposed architecture is only applicable to the near-straight rail recognition tasks. Since the proposed RRM-PLD has no limits to the shapes of rails, it is especially suitable for not only rails but other line-shaped objects that are collected from changeable viewing angles and have different distribution directions theoretically. This will be further verified in future works.



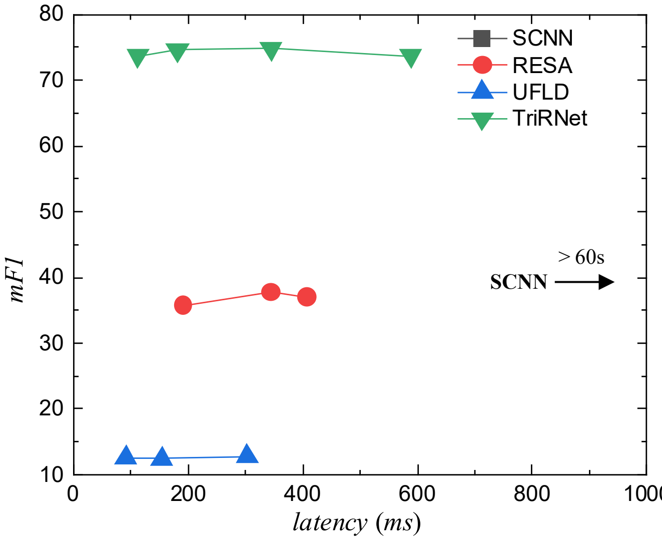


Fig. 9. The latency vs. accuracy diagram for the different model series. The curves on the top figure are all tested on NX device and the ones on the bottom figure are all tested on TX2 device.

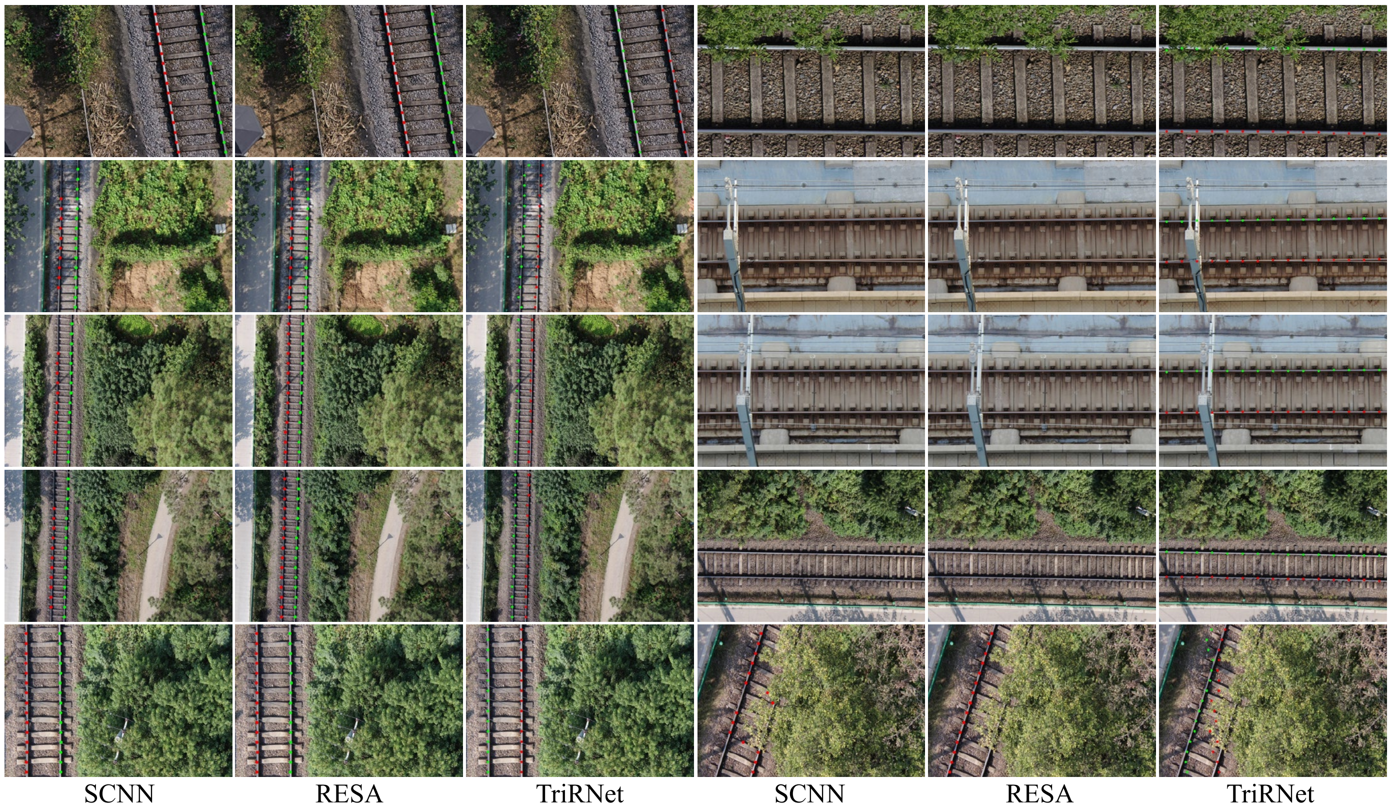


Fig. 10. Several visual examples of the proposed TriRNet compared to SCNN and RESA.

F. Adaptability Test on More Scenarios

To better evaluate the adaptability to more railway scenarios under different environment conditions, datasets RailAug and Rail4Track are further constructed. The sizes of them are shown in Table VI. RailAug with a larger size is used to simulate the different weather, lighting, and movement conditions that may occur in the UAVs' practical railway inspection. It includes the augmentation of motion blurring, random rain, random snow, random fog, and brightness transforms. Rail4Track contains images of the turnout area with a more complex layout and each of them is with four rails inside. The resolution of images in both datasets is 1920×1080. The 3/4 quantiles of pixel width to perform IoU computation are 28 and 21, respectively.

TriRNet backboned by ResNet34 is evaluated on these two datasets. The experimental results presented in Table VII show the proposed TriRNet's good capability to adapt to different environmental conditions. As can be seen, under the dimension setting of d_s=15 and d_g =150, TriRNet can achieve mF1 metrics of 68.38 and 71.70 on the Rail4Track and RailAug, respectively. Especially, when the threshold is taken as 0.3, TriRNet can obtain an F1 measure of up to 88.26 and 85.96 on the two datasets, respectively. Additionally, to give a more intuitive impression of the accuracy of F1's changing with the IoU threshold θ, the F1 *vs*. θ curves for Rail4Track and RailAug are illustrated in Fig. 11. The area under the curve can reflect the comprehensive ability of the algorithm. TriRNet backboned by ResNet34 can perform almost equally well on the two datasets of very different sizes, indicating the great potential of TriRNet.

Fig. 12 gives more abundant visual samples on the datasets Rail4Track (a-c) and RailAug (d-g) by TriRNet-ResNet34. The samples of the first two columns of Rail4Track show the good capability of the proposed algorithms for the turnout area (Y-shaped rail area) with a more complex rail layout. All the visual examples of Rail4Track have proved the good performance of the proposed TriRNet on the scenarios of more rails. More randomly motion blurred, brightness transformed, snow-added, rain-added, and fog-added examples in RailAug are presented in the bottom four rows of Fig. 12. These visual results indicate the superior adaptability of the proposed TriRNet to the practical UAV-based inspection scenarios under different weather, lighting, and UAV's movement conditions.

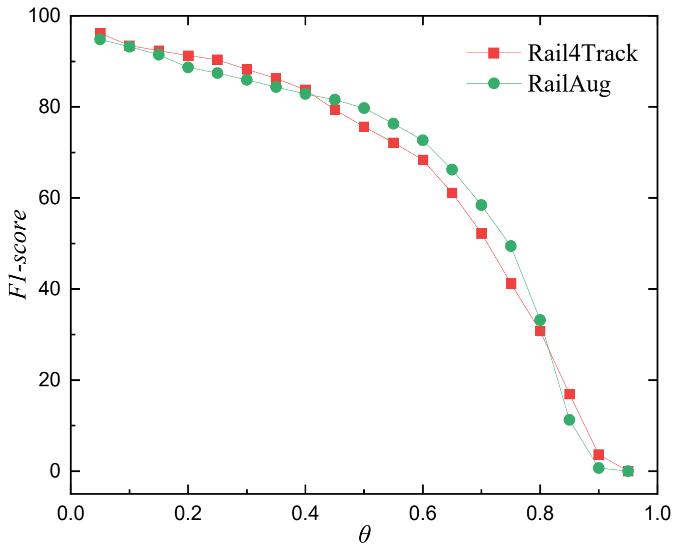


Fig. 11. F1 *vs*. θ curves for Rail4Track and RailAug.

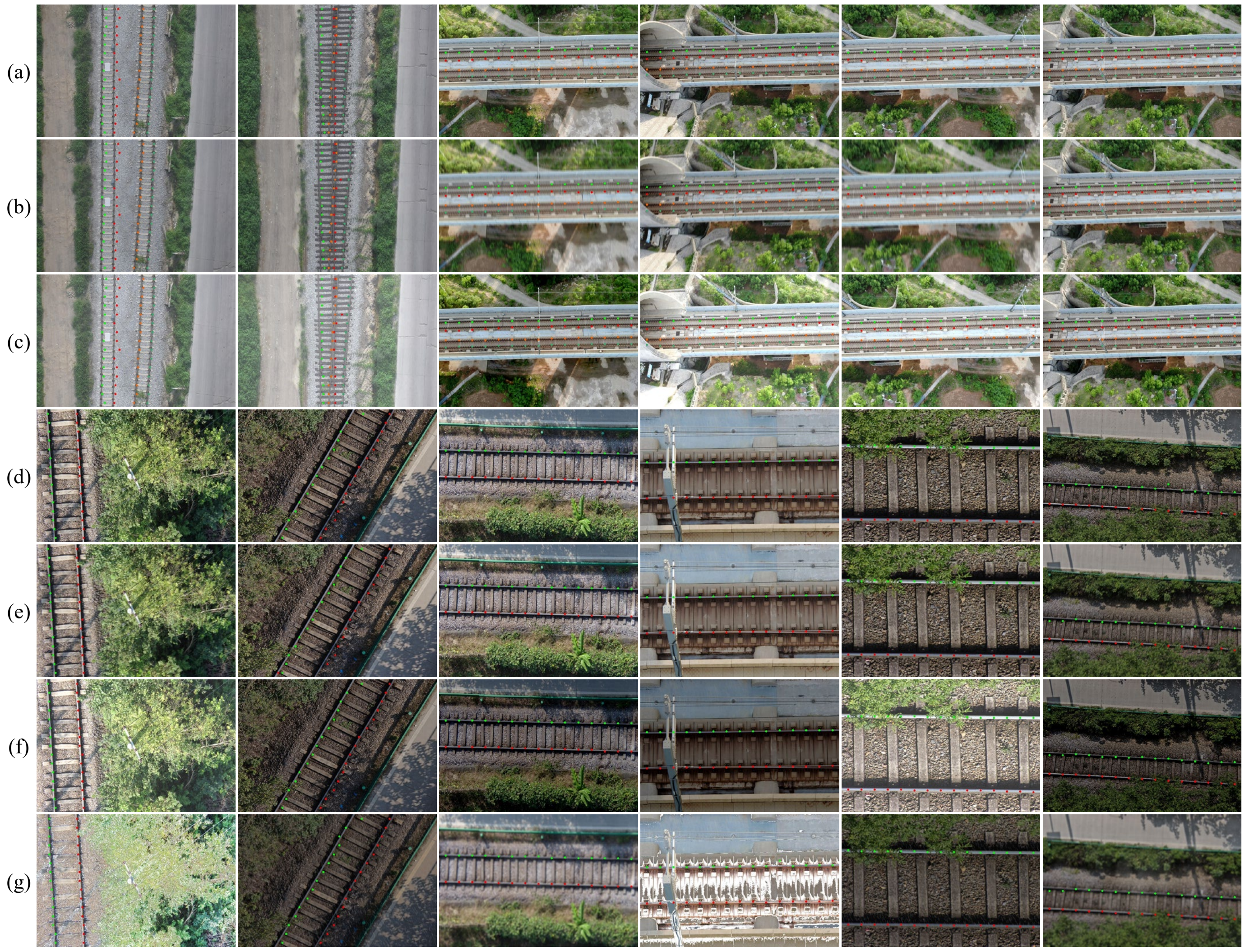


Fig. 12. Visual examples on the datasets Rail4Track (a-c) and RailAug (d-g) by TriRNet-ResNet34. (a) and (d) are original images; (b) and (e) are motion-blurred images; (c) and (f) are brightness-transformed images; (g) are images augmented with random snow, random rain, or random fogs, respectively.

In the motion-blurred example given in row (e) of the second column of Fig. 12, a prediction point for one rail is mixed with the prediction points for another. Despite that, such kind of mixed prediction results can be easily refined by some simple outlier removal techniques.

TABLE VI
THE SIZE OF THE TRAINING, VALIDATION, AND TEST SET FOR DATASET RAIL4TRACK AND RAILAUG.

dataset	training	validation	test	in total
Rail4Track	1290	270	290	1850
RailAug	4465	1115	1115	6695

TABLE VII
EVALUATION RESULTS OF TRIRNET BACKBONED BY RESNET34 ON DATASETS RAIL4TRACK AND RAILAUG UNDER THE DIMENSION SETTING OF $d_s = 15$ AND $d_g = 150$.

dataset	d_s	d_g	F1@30	F1@50	F1@75	mF1
Rail4Track	15	150	88.26	75.65	41.22	68.38
RailAug	15	150	85.96	79.73	49.42	71.70

The statistical comparison of recognition effect under different weathers, line types, rail sections, etc., conditions is further investigated on the two datasets, as presented in Table VIII. Compared to the original split of the RailAug test set, it is evident that inclement weather conditions, i.e., rain and snow, can indeed exert a discernible adverse influence on rail recognition accuracy. Unexpectedly, the fog split yields a better recognition performance. This can be attributed to the fact that the simulated fog is thin mists rather than dense fog, which, to some extent, mitigates the impact of complex changeable background interference in the vicinity of the railways. In the presence of motion blurring and brightness change splits, small declines in recognition accuracy which primarily occurs within the high-reliability recognition requirement zone, i.e., F1@75, can also be observed. The dataset is also split into two parts for comparison: ballasted and ballastless. It can be seen that the accuracy on the ballastless split far surpasses its ballasted counterpart and the F1 value even reaches an impressive 98.22 at an IoU threshold of 0.3 for the ballastless split. Naturally, the homogenous and consistent context of ballastless railway lines make it more conducive to complete rail recognition with less noise interference compared to the ballasted ones.

TABLE VIII

STATISTICAL COMPARISON OF RECOGNITION EFFECT UNDER DIFFERENT WEATHER, LINE TYPE, RAIL SECTION, ETC., CONDITIONS. BECAUSE IN PRACTICE THERE IS STILL A SLIGHT CURVATURE THAT IS DIFFICULT TO OBSERVE FOR SOME OF RAILS IN THE DATASET, THE "***straight****" HERE IS INTENDED TO INDICATE THAT THE RAILS ARE JUST APPROXIMATELY STRAIGHT.

Dataset	**RailAug**						**RailAug**		**Rail4Track**	
Category	**Weather conditions**			**Other transforms**			**Line types**		**Rail sections**	
Split	***rain***	***snow***	***fog***	***blurring***	***brightness***	***original***	***ballasted***	***ballastless***	***straight****	***turnout***
Number	71	75	77	223	446	223	580	535	240	50
F1@30	81.23	85.41	88.52	86.69	86.01	86.01	74.44	98.22	89.38	82.63
F1@50	78.50	76.16	81.97	80.09	79.75	80.09	67.11	93.15	81.56	45.79
F1@75	49.15	41.28	57.05	45.28	50.63	51.19	33.47	66.38	47.40	10.00
mF1	69.63	67.62	75.85	70.69	72.13	72.43	58.34	85.92	72.78	46.14

Additionally, a comparison was made to assess the impact of different rail shape conditions on the recognition effect. The test set of Rail4Track was divided into two splits: straight sections and turnout sections, as indicated in the last two columns of Table VIII. In a general assessment, the recognition accuracy within turnout sections consistently falls below that observed in straight sections. To be more specific, the composite metric mF1 value for turnout sections, at 46.14, is notably lower than the one for straight sections, which stands at 72.78. It can also be noted that when the recognition confidence is comparatively low, particularly with an IoU threshold of 0.3, the F1 score remains relatively high, i.e., 82.63. However, as it moves to higher confidence intervals, such as IoU thresholds of 0.5 or 0.75, the F1 score experiences a rapid decrease. It is analyzed that the decrease may arise from the potential confusion introduced to the model by changes in the order of multiple rails present in turnout sections. Additionally, the lower quantity of images in turnout sections, as compared to straight sections, can also introduce a certain degree of this kind of imbalance that could impact the model's ability to achieve robust recognition. More images from railway turnout areas are expected to be included in the future's evaluation and experiments to eliminate the impact of unbalanced sample size on the model.

V. CONCLUSION

This paper mainly discusses real-time rail recognition towards UAV-based practical and automatic railway inspection. According to the geometric characteristics of rails from the perspective of UAVs, a general and adaptive rail representation method based on projection length discrimination (RRM-PLD) is proposed, which can always select an optimal representation direction to represent any kind of rails. Along with that, this paper also proposes a novel real-time rail recognition network (TriRNet) architecture, in which the proposed inter-rail attention (IRA) mechanism can fuse the local features of single rails and the global features across all rails to accurately discriminate the geometric distribution of rails in a regressive way and thus improve the final recognition accuracy. Especially, one-to-one mapping from the constructed anchor points to the final feature maps for proposal generation is established, which can greatly simplify the model design process and improves the interpretability of the model. To train the network in a more balanced and efficient way, detailed loss designing and model training strategies are also introduced.

Experiments have proven that the proposed formulation can recognize rails with any inclination angle in the UAV aerial images more efficiently in terms of both reasoning latency and recognition accuracy compared to other algorithms. Specifically, the proposed TriRNet can achieve an mF1 of 74.83, and the inference speed has obtained an FPS of higher than 38 without utilizing any acceleration libraries, which meets the practical requirements of the onboard edge computing devices. It is also worth noting that the rail recognition architecture proposed in this paper has excellent universality and can be transferred to accomplish the detection of other linear-shaped structures in theory. It provides a significant reference for the detection of other line-shaped structures. More investigations will be conducted to verify the engineering usability, practical performance, model robustness of the proposed algorithm in the future to better adapt to more challenging railway scenarios.

ACKNOWLEDGMENT

The authors are grateful to Prof. Yap-Peng Tan (IEEE Fellow, School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore) for his invaluable guidance and to the China Scholarship Council for their financial support. Their contributions have been crucial to the completion of this paper.

REFERENCES

[1] Z. Q. Zhao, P. Zheng, S. T. Xu, and X. Wu, "Object Detection With Deep Learning: A Review," *IEEE Trans Neural Netw Learn Syst,* vol. 30, no. 11, pp. 3212-3232, Nov 2019, doi: 10.1109/TNNLS.2018.2876865.

[2] W. Luo, J. Xing, A. Milan, X. Zhang, W. Liu, and T.-K. Kim, "Multiple object tracking: A literature review," *Artificial Intelligence,* vol. 293, 2021, doi: 10.1016/j.artint.2020.103448.

[3] Y. Mo, Y. Wu, X. Yang, F. Liu, and Y. Liao, "Review the state-of-the-art technologies of semantic segmentation based on deep learning," *Neurocomputing,* vol. 493, pp. 626-646, 2022, doi: 10.1016/j.neucom.2022.01.005.

[4] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez, "A review on deep learning techniques applied to semantic segmentation," in *CVPR*, 2017.

[5] T. Wang, Z. Zhang, F. Yang, and K.-L. Tsui, "Automatic Rail Component Detection Based on AttnConv-Net," *IEEE Sensors Journal,* vol. 22, no. 3, pp. 2379-2388, 2022, doi: 10.1109/jsen.2021.3132460.

[6] L. Zhuang, H. Qi, T. Wang, and Z. Zhang, "A Deep-Learning-Powered Near-Real-Time Detection of Railway Track Major Components: A Two-Stage Computer-Vision-Based Method," *IEEE Internet of Things Journal,* vol. 9, no. 19, pp. 18806-18816, 2022, doi: 10.1109/jiot.2022.3162295.

[7] F. Guo, Y. Qian, Y. Wu, Z. Leng, and H. Yu, "Automatic railroad track components inspection using real‐time instance segmentation," *Computer-Aided Civil and Infrastructure Engineering,* vol. 36, no. 3, pp. 362-377, 2021.

[8] D. Wei, X. Wei, Q. Tang, L. Jia, X. Yin, and Y. Ji, "RTLSeg: A novel multi-component inspection network for railway track line based on instance segmentation," *Engineering Applications of Artificial Intelligence,* vol. 119, 2023, doi: 10.1016/j.engappai.2023.105822.

[9] Y. Wu, Y. Qin, Y. Qian, F. Guo, Z. Wang, and L. Jia, "Hybrid deep learning architecture for rail surface segmentation and surface defect detection,"

Computer-Aided Civil and Infrastructure Engineering, vol. 37, no. 2, pp. 227-244, 2022.
[10] Y. Wu, Y. Qin, Z. Wang, and L. Jia, "A UAV-Based Visual Inspection Method for Rail Surface Defects," *Applied Sciences,* vol. 8, no. 7, pp. 1028-1047, 2018.
[11] L. Zhuang, H. Qi, and Z. Zhang, "The Automatic Rail Surface Multi-Flaw Identification Based on a Deep Learning Powered Framework," *IEEE Transactions on Intelligent Transportation Systems,* vol. 23, no. 8, pp. 12133-12143, 2022, doi: 10.1109/tits.2021.3109949.
[12] X. Wei, Z. Yang, Y. Liu, D. Wei, L. Jia, and Y. Li, "Railway track fastener defect detection based on image processing and deep learning techniques: A comparative study," *Engineering Applications of Artificial Intelligence,* vol. 80, pp. 66-81, 2019, doi: 10.1016/j.engappai.2019.01.008.
[13] Y. Wu, F. Meng, Y. Qin, Y. Qian, F. Xu, and L. Jia, "UAV imagery based potential safety hazard evaluation for high-speed railroad using Real-time instance segmentation," *Advanced Engineering Informatics,* vol. 55, 2023, doi: 10.1016/j.aei.2022.101819.
[14] Y. Li, H. Dong, H. Li, X. Zhang, B. Zhang, and Z. Xiao, "Multi-block SSD based on small object detection for UAV railway scene surveillance," *Chinese Journal of Aeronautics,* vol. 33, no. 6, pp. 1747-1755, 2020.
[15] P. Chen, Y. Wu, Y. Qin, H. Yang, and Y. Huang, "Rail fastener defect inspection based on UAV images: A comparative study," in *Proc. EITRT*, Qingdao, China, 2019, vol. 640, in Lecture Notes in Electrical Engineering, pp. 685-694.
[16] J. Liu, Z. Wang, Y. Wu, Y. Qin, X. Cao, and Y. Huang, "An Improved Faster R-CNN for UAV-Based Catenary Support Device Inspection," *International Journal of Software Engineering and Knowledge Engineering,* vol. 30, no. 07, pp. 941-959, 2020.
[17] M. Banić, A. Miltenović, M. Pavlović, and I. Ćirić, "Intelligent Machine Vision Based Railway Infrastructure Inspection and Monitoring Using Uav," *Facta Universitatis, Series: Mechanical Engineering,* vol. 17, no. 3, pp. 357-3640, 2019.
[18] Y. Wu, Y. Qin, Z. Wang, X. Ma, and Z. Cao, "Densely pyramidal residual network for UAV-based railway images dehazing," *Neurocomputing,* vol. 371, pp. 124-136, 2020.
[19] L. Tong *et al.*, "Fully Decoupled Residual ConvNet for Real-Time Railway Scene Parsing of UAV Aerial Images," *IEEE Transactions on Intelligent Transportation Systems,* vol. 23, no. 9, pp. 14806 - 14819, 2022, doi: 10.1109/tits.2021.3134318.
[20] S. Sahebdivani, H. Arefi, and M. Maboudi, "Rail Track Detection and Projection-Based 3D Modeling from UAV Point Cloud," *Sensors (Basel),* vol. 20, no. 18, pp. 1-15, Sep 13 2020.
[21] Y. Geng *et al.*, "UAV-LiDAR-Based Measuring Framework for Height and Stagger of High-Speed Railway Contact Wire," *IEEE Transactions on Intelligent Transportation Systems,* vol. 23, no. 7, pp. 7587-7600, 2022, doi: 10.1109/tits.2021.3071445.
[22] Y. Geng *et al.*, "3DGraphSeg: A Unified Graph Representation-Based Point Cloud Segmentation Framework for Full-Range Highspeed Railway Environments," *IEEE Transactions on Industrial Informatics,* pp. 1-13, 2023, doi: 10.1109/tii.2023.3246492.
[23] H. Yang, X. Li, Y. Guo, and L. Jia, "RT-GAN: GAN Based Architecture for Precise Segmentation of Railway Tracks," *Applied Sciences,* vol. 12, no. 23, 2022, doi: 10.3390/app122312044.
[24] X. Pan, J. Shi, P. Luo, X. Wang, and X. Tang, "Spatial as deep: Spatial CNN for traffic scene understanding," in *Proc. AAAI*, New Orleans, LA, USA, 2018, pp. 7276-7283.
[25] TuSimple, "Tusimple: lane detection benchmark, https://github.com/TuSimple/tusimple-benchmark," ed, 2017.
[26] K. Behrendt and R. Soussan, "Unsupervised Labeled Lane Markers Using Maps," presented at the ICCVW, 2019.
[27] L. Tabelini, R. Berriel, T. M. Paixao, C. Badue, A. F. de Souza, and T. Oliveira-Santos, "PolyLaneNet: Lane estimation via deep polynomial regression," in *Proc. ICPR*, Virtual, Milan, Italy, 2020, pp. 6150-6156.
[28] R. Liu, Z. Yuan, T. Liu, and Z. Xiong, "End-to-end lane shape prediction with transformers," in *Proc. WACV*, Virtual, Online, United states, 2021, pp. 3693-3701.
[29] A. Vaswani *et al.*, "Attention is all you need," presented at the Proc. NIPS, Long Beach, California, USA, 2017.
[30] S. Lee *et al.*, "VPGNet: Vanishing Point Guided Network for Lane and Road Marking Detection and Recognition," presented at the 2017 IEEE International Conference on Computer Vision (ICCV), 2017.
[31] D. Neven, B. De Brabandere, S. Georgoulis, M. Proesmans, and L. Van Gool, "Towards End-to-End Lane Detection: An Instance Segmentation Approach," in *Proc. IV*, Changshu, Suzhou, China, 2018, vol. 2018-June, pp. 286-291.
[32] M. Ghafoorian, C. Nugteren, N. Baka, O. Booij, and M. Hofmann, "EL-GAN: Embedding Loss Driven Generative Adversarial Networks for Lane Detection," in *Computer Vision – ECCV 2018 Workshops*, (Lecture Notes in Computer Science, 2019, ch. Chapter 15, pp. 256-272.
[33] Y. Hou, Z. Ma, C. Liu, and C. C. Loy, "Learning lightweight lane detection CNNS by self attention distillation," in *Proc. ICCV*, Seoul, Korea, Republic of, 2019, vol. 2019-October, pp. 1013-1021.
[34] H. Abualsaud, S. Liu, D. B. Lu, K. Situ, A. Rangesh, and M. M. Trivedi, "LaneAF: Robust Multi-Lane Detection With Affinity Fields," *IEEE Robotics and Automation Letters,* vol. 6, no. 4, pp. 7477-7484, 2021.
[35] Y. Ko, Y. Lee, S. Azam, F. Munir, M. Jeon, and W. Pedrycz, "Key Points Estimation and Point Instance Segmentation Approach for Lane Detection," *IEEE Transactions on Intelligent Transportation Systems,* vol. 23, no. 7, pp. 8949-8958, 2022.
[36] Z. Chen, Q. Liu, and C. Lian, "PointLaneNet: Efficient end-to-end CNNs for Accurate Real-Time Lane Detection," in *Proc. IV*, Paris, France, 2019, pp. 2563-2568.
[37] X. Li, J. Li, X. Hu, and J. Yang, "Line-CNN: End-to-End Traffic Line Detection With Line Proposal Unit," *IEEE Transactions on Intelligent Transportation Systems,* vol. 21, no. 1, pp. 248-258, 2020.
[38] H. Xu, S. Wang, X. Cai, W. Zhang, X. Liang, and Z. Li, "CurveLane-NAS: Unifying Lane-Sensitive Architecture Search and Adaptive Point Blending," in *Proc. ECCV*, Cham, 2020, pp. 689-704.
[39] L. Tabelini, R. Berriel, T. M. Paixão, C. Badue, A. F. D. Souza, and T. Oliveira-Santos, "Keep your Eyes on the Lane: Real-time Attention-guided Lane Detection," in *Proc. CVPR*, 2021, pp. 294-302.
[40] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 39, no. 6, pp. 1137-1149, Jun 2017.
[41] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," presented at the ICCV, 2017.
[42] J. Philion, "FastDraw: Addressing the Long Tail of Lane Detection by Adapting a Sequential Prediction Network," in *Proc. CVPR*, 2019, pp. 11574-11583.
[43] Z. Qin, H. Wang, and X. Li, "Ultra Fast Structure-Aware Deep Lane Detection," in *Proc. ECCV*, Cham, 2020, pp. 276-291.
[44] S. Yoo *et al.*, "End-to-end lane marker detection via row-wise classification," in *CVPRW*, Virtual, Online, USA, 2020, vol. 2020-June, pp. 4335-4343.
[45] L. Liu, X. Chen, S. Zhu, and P. Tan, *CondLaneNet: a Top-to-down Lane Detection Framework Based on Conditional Convolution*. 2021.
[46] T. Zheng, Y. Huang, Y. Liu, W. Tang, and Z. Yang, "CLRNet: Cross Layer Refinement Network for Lane Detection," in *Proc. CVPR*, 2022 2022, pp. 888-897.
[47] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proc. CVPR*, 2016, vol. 2016-December, pp. 770-778. [Online]. Available: https://arxiv.org/abs/1512.03385. [Online]. Available: https://arxiv.org/abs/1512.03385
[48] T. Zheng *et al.*, "RESA: Recurrent Feature-Shift Aggregator for Lane Detection," in *Proc. AAAI*, 2021, vol. 4B, pp. 3547-3554.

LEI TONG (Student Member, IEEE) received his B.E. degree from Beijing Jiaotong University, Beijing, China, in 2019. He is a Ph.D. candidate in the State Key Laboratory of Advanced Rail Autonomous Operation, Beijing Jiaotong University, Beijing, China. He is also a visiting Ph.D. student currently at School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. His research interests include deep learning, image processing, intelligent railway detection, and UAV-based automatic railway inspection.

LIMIN JIA (Member, IEEE) is currently a Professor with the State Key Laboratory of Advanced Rail Autonomous Operation, Beijing Jiaotong University, Beijing, China. His research interests include intelligent control, system safety, and fault diagnosis and their applications in a variety of fields, such as rail traffic control and safety, and transportation.

YIXUAN GENG (Student Member, IEEE) received his B.E. and Ph.D. degrees from Beijing Jiaotong University, China. He is currently a lecturer at the School of Traffic and Transportation, Beijing Jiaotong University, Beijing, China. His research interests include machine learning, fault diagnosis, and point cloud processing.

YONG QIN (Member, IEEE) received his B.E. and M.E. degrees from Tongji University, Shanghai, China, in 1993 and 1996, respectively, and the Ph.D. degree from the China Academy of Railway Sciences, Beijing, China, in 1999. He is currently a Professor at the State Key Laboratory of Advanced Rail Autonomous Operation, Beijing Jiaotong University, Beijing, China.

DONGHAI SONG graduated from Shijiazhuang Tiedao University in 2012. He is currently a senior engineer at China Railway Electrification Bureau Group Co., Ltd. His research interests include reliability improvement and efficiency optimization of high-speed railway power supply systems.

BIDONG MIAO received his Bachelor's degree from Hunan University of Science and Technology, China, in 2006. He is currently a Senior Engineer at China Railway Electrification Bureau Group Co., Ltd. His research interests include fault distance measurement optimization and safety assurance of high-speed railway power supply systems.

TIAN TANG received the B.E. and M.S. degree from Beijing Institute of Technology and Chinese Aeronautical Establishment, Beijing, China, in 2005 and 2011, respectively. He is currently working as a VP in the United Aircraft Group, Beijing, China. His research interests include high-reliability UAV/Drone design and its application in railway inspection.

ZHIPENG WANG (Member, IEEE) received the B.S. and Ph.D. degree from Northeast University and Beihang University, China, in 2008 and 2014, respectively. He is currently an associate professor at the State Key Laboratory of Advanced Rail Autonomous Operation, Beijing Jiaotong University, Beijing, China. His research interests include UAV-based railway inspection and prognostics & health management.