IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 30, 2020, accepted August 17, 2020, date of publication August 25, 2020, date of current version September 17, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3019318

Vanishing Point Detection and Rail Segmentation Based on Deep Multi-Task Learning

XINGXIN LI, LIQIANG ZHU, (Member, IEEE), ZUJUN YU, BAOQING GUO, AND YANQIN WAN
School of Mechanical, Electronic, and Control Engineering, Beijing Jiaotong University, Beijing 100044, China
Key Laboratory of Vehicle Advanced Manufacturing, Measuring and Control Technology, Beijing Jiaotong University, Beijing 100044, China

Corresponding author: Liqiang Zhu (lqzhu@bjtu.edu.cn)

This work was supported by the Fundamental Research Funds for the Central Universities of China under Grant 2020JBZD003.

ABSTRACT In modern railway systems, video surveillance and machine vision analysis have been widely used to detect perimeter intrusions. For pan–tilt–zoom (PTZ) cameras, the machine vision system needs to detect adjustments in PTZ cameras and then automatically determine the new alarm region in real time. In this paper, we propose a deep multi-task learning based algorithm for simultaneous vanishing point (VP) detection and rail segmentation, which can identify camera adjustment from changes in VP, and then automatically determine the alarm region from segmented rails. The multi-task based neural network consists of a feature extraction base network and three sub-task networks. The first sub-task network is a convolution regression network for VP detection. The second sub-task network utilizes an encoder-decoder structure for vanishing region (VR, a fixed region centered on VP) segmentation. The third sub-task network shares the encoder-decoder structure with the VR segmentation task and is used for rail segmentation. The VR segmentation task is activated only at the training stage, serving as an auxiliary task to enhance feature learning ability and increase VP detection accuracy. To further improve the accuracies of VP detection and rail segmentation, low-level features is modulated by high-level semantic information before feeding to the decoder stage. With the help of shared feature extraction and auxiliary training, the proposed VP prediction method needs very small training dataset and outperforms other methods in both efficiency and accuracy.

INDEX TERMS Vanishing point detection, rail segmentation, intrusion detection, multi-task learning, deep learning.

I. INTRODUCTION

Video surveillance system (VSS) is an important subsystem in modern railways which are susceptible to many types of intruding foreign objects, e.g., trespassing passengers and terrorists, landslide or falling cargo from overhead bridge. Intelligent video analysis has been widely used to detect objects intruding into the track region of railway. The general procedure of intrusion detection based on intelligent video analysis consists of four sequential steps: manual determination of alarm region that is often defined as rail track zone between leftmost and rightmost rails, intruding object detection by separating foreground objects from background [1], normal train object exclusion and final intrusion detection by deciding whether foreground object locates inside the alarm region or not, as shown in Fig.1. Due to the fact that many cameras installed along railway lines are pan–tilt–zoom (PTZ) cameras and their monitoring scenes may be adjusted from time to time by different staff, it is desirable for VSS to be able to detect the change of monitoring scene and then determine the alarm region automatically in real-time. In this paper, we propose to estimate the position changes of main vanishing point (VP) to detect monitoring scene adjustment and then segment rails to find the alarm region. The proposed intrusion detection procedure is illustrated in Fig.2.

VP, defined as a point where the parallel lines in the 3D world intersect in the 2D image plane, contains rich scene information and is essential for a good understanding of 3D geometry. The traditional VP detection approaches consist of multiple steps, including handcrafted feature extraction and decision strategies, and have been used in many applications. For example, in automatic driving, VP can provide fundamental information about road geometry for vison-based navigation, such as road region extraction [2] and road sign

The associate editor coordinating the review of this manuscript and approving it for publication was Huazhu Fu.

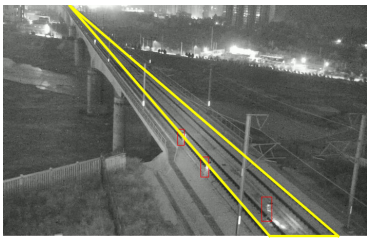


FIGURE 1. The detection of objects intruding into railway. Inside the yellow triangle is the alarm region. The objects in red bounding box are located inside alarm region and regarded as intrusions.

recognition [3]. VP was also used in camera calibration in [4] where three orthogonal VPs were estimated based on the Manhattan world assumption. In [5], dominant VPs were detected in landscape images for image retrieval. Recently, deep learning based technology has been proposed to realize end-to-end VP detection. Deep learning was first used to detect VP in [6], where image was divided into equal size blocks and used convolutional neural network (CNN) to directly predict which block contains a VP. CNN was also used to predict the dominate VP in natural landscape images that lack obvious edges or lines [7]. Deep learning approach learns feature representation through the end-to-end supervised training, which is a data-driven method and performs better than handcrafted features of traditional methods in many applications. Deep learning is also widely applied in intelligent monitoring and inspection system for high-speed railway to ensure the safe operation of railway [8]–[13].

For VSS in railways, applying VP detection methods faces two major difficulties. First, complications arise due to the low image quality in unstable illumination conditions, non-Manhattan lines typical in curved railway sections and occlusion of running trains. In these situations, directly predicting VP with handcrafted or CNN-outputted features is unstable and has poor accuracy, making the training of the underlying algorithm non-trivial. Second, VP detection, alarm region extraction and object detection for numerous cameras have to be done in real time. Thus, more efficient processing framework has to be used to remove redundant operations in these processing steps. In this paper, we propose a deep multi-task learning framework for simultaneous VP detection and rail segmentation. This framework can improve feature presentation and model generalization through complementary information from related tasks. The VP detection accuracy is further increased by adding a vanishing region (VR, a fixed size region centered at VP) segmentation task, which is an auxiliary training task. Sharing feature representations between different tasks can prevent over-fitting and save computing resources. The extracted features can be further used by subsequent object detection, significantly improving the efficiency. Experimental results show that the proposed algorithm outperforms existing algorithms.

The contributions of this paper include:

1) A deep multi-task learning framework integrating regression task and segmentation task is proposed to detect VP and rails only through a forward pass.
2) The auxiliary task VR segmentation is added to improve the performance of VP detection. The VR segmentation task, removed in the test phase, shares the feature with rail segmentation in up-sampling process, which only adds a small amount of computation load. At the same time, the low-level features reused in the up-sampling stage are modulated by relative high-level semantic information to improve both the detection and segmentation accuracy.
3) Our new algorithm is superior to other VP detection algorithms on railway scenes and its processing speed meets real-time requirements.
4) The alarm region is extracted automatically with detected VP and rails. It is beneficial to reliable intrusion detection and safer railway operation.

The rest of the paper is organized as follows. The Section II introduces the related works about VP detection and deep multi-task leaning. The multi-task learning framework is described in Section III. The Section IV shows the experimental results and Section V draws conclusions.

II. RELATED WORKS

This section provides a brief review on existing works related to the two main topics covered in this paper, namely: VP detection and multi-task learning.

A. VP DETECTION

Traditional VP detection methods rely on handcrafted features and voting or clustering processes. The authors of [2] computed dominant texture orientation at each pixel using Gabor filters. VP was estimated with soft-voting after discarding unrelated pixels. The authors of [14] and [15] utilized the improved Weber Local Descriptor to compute the texture and orientation features, and then located VP by a linear-voting scheme. In [3], authors proposed a probabilistic voting procedure to find VP from intersections of multiple

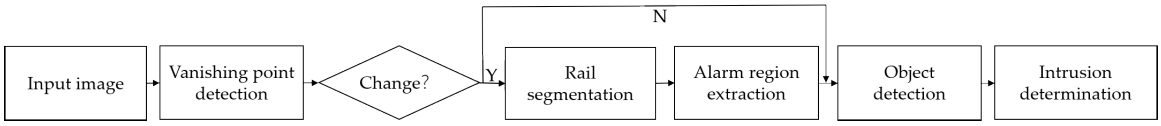


FIGURE 2. The flow of automatic intrusion detection. The alarm region is recalculated automatically depending on vanishing point detection and rail segmentation.

line segments. In [16], the J-Linkage algorithm was used to compute a set of VP candidates from detected edges, and final VP was refined using EM algorithm. In [17] and [18], the highway scene was divided into rough road region, sky region and vertical region with a dark channel prior based segmentation method and vertical envelope lines analysis. Then, road lines were extracted with several own-defined constraints and VP was estimated through mean-shift clustering. VP detection on railway platform scene was first studied in [19]. Main straight lines were detected by Canny edge detector and Hough transform. The geometric structure information of the railway platform was utilized during line clustering. Although the above-mentioned traditional methods achieved satisfied results in corresponding applications, these methods are prone to accumulate errors because of multi-step processing. Handcrafted features are also fragile and susceptible to noise. Deep learning as an end-to-end manner can learn powerful features from data [20], [21] and has shown excellent performance in many applications. VP detection was transformed as a CNN classification task in [6] and [22]. The input image was divided into equal size blocks and each block acted as a sample to be classified, then the block containing VP was target class. In this method, high prediction accuracy requires small block size. However, if the size is too small, the computation is time-consuming and not affordable. In [23], authors proposed an end-to-end neural network for simultaneous VP detection and road mark recognition through multi-task learning. The whole image was divided into four rectangles defined by VP position and four image corners, and VP was located at the intersection of the four rectangles. So, VP prediction was transformed as a pixel-level prediction task and an auxiliary task that could also provide global information for other tasks. This method is proved to be suitable for images captured by vehicle camera, where scene structure is relatively fixed and simple. Compared with the traditional methods or the classification network, regression CNN that directly predicted VP's coordinates was more accurate [24]. ResNet was used in [25], because it was beneficial to preserve geometric information of input data with fewer pooling layers and fully connected layers. A multi-task framework that combined classification network and regression network to predict VP in natural landscape image was proposed in [7]. The classification network predicted which image block contains VP, and the regression network predicted the coordinates of VP.

B. MULTI-TASK LEARNING

Multi-task learning (MTL) methods can obtain multi-task predictions simultaneously by a forward pass. MTL benefits from the extra information in the related tasks [26], [27]. The most commonly used MTL structure in deep learning is hard parameter sharing structure [28]. In this structure, all tasks share multiple feature representation layers and keep several task-specific output layers. MTL realizes implicit data augmentation because different tasks have different noise patterns, which could improve generalization ability. Auxiliary information from related tasks in MTL makes some important features easy to learn or more focused. MTL prefers feature representation to all tasks and acts as a regularization to prevent over-fitting. In computer vision, multi-class classification can be viewed as an MTL that integrating multiple binary classification tasks, such as handwritten digits classification [29], but MTL mostly focuses on producing different types of outputs including class probability and specific regression value. In object detection, Fast R-CNN [30] simultaneously predicted class probability and offsets for each region of interest (ROI), and all ROIs shared feature representation. Faster R-CNN [31] introduced a Region Proposal Network (RPN) to generate ROI and RPN shared convolutional features with the detection network instead of training two separate networks. Experiments showed that both of the detection accuracy and speed were increased. Mask R-CNN [32] added a branch for predicting an object mask based on Faster R-CNN. Both object detection and instance segmentation could be achieved by only a small amount of increased computational cost. The framework made segmentation task easier to train because of existing object detection task. It could easily be extended to human pose estimation further. A framework called HyperFace was proposed for simultaneous face detection, landmark localization, pose estimation and gender recognition in [33]. The results showed that the MTL boosted individual performance through learning more discriminative features. Authors in [34] payed attention to the training problem in MTL. Generally, multi-task loss is a weighted linear sum of each task loss, but the weights are difficult to select manually. To solve this problem, the authors in [34] took Bayesian probabilistic theory into deep MTL and utilized homoscedastic uncertainty, also called task uncertainty, which is task-dependent, to guide the learning of each task weight. In most situations, related tasks as auxiliary tasks are used to boost main task performance and will be removed during the testing phase. For example, head pose estimation and facial attribute inference were used to improve the robustness of the face key point detection in [35], especially when there existed severe occlusion and pose variation. The results showed that multi-task model were more efficient and effective than cascaded deep models. For human pose estimation [36], body part detection was added to the original body key-point regression task. The detection task improved the regression accuracy of the main task. The joint training converged to a better minimum and enhanced the generalization ability. MTL is also applied in high-speed railway inspection. In [13], it realized railway ties and fasteners inspection. A deep multi-task neural network that integrates a material classifier and a denoising autoencoder was used to detect insulator surface defect [12].

III. THE PROPOSED FRAMEWORK

Images from VSS in railways are mostly of low quality and have to be processed in multiple real-time tasks, including VP detection, track segmentation and intrusion detection. However, existing methods cannot share or reuse the

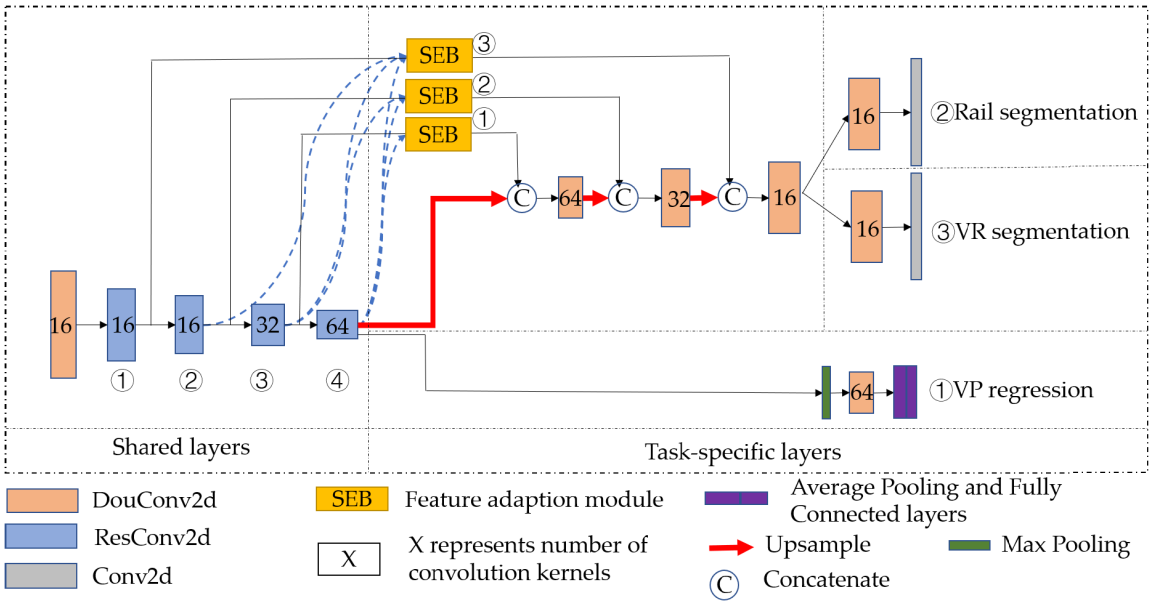


FIGURE 3. The proposed deep multi-task learning framework for simultaneous VP detection and rail segmentation. DouConv2d and Conv2d represent double and one convolutional layer respectively. ResConv2d represents residual connection block.

information in these independent tasks to accelerate the training and testing stages. In this work, we propose an end-to-end deep multi-task learning framework that realizes simultaneous VP detection and rail segmentation. The proposed framework contains three tasks, i.e., VP detection, rail segmentation and VR segmentation. As shown in Fig.3. and Table 1, the network structure consists of shared feature extraction layers and task-specific layers. The shared layers include two convolutional layers (DouConv2d) and four layers of residual connection block (ResConv2d) [21], forming a feature-extracting base network. While VP is predicted by a convolution regression network using the output of the base network, two segmentation tasks share an encoder-decoder structure and employ a few task-specific layers to generate segmentation results. Low-level features are reused in the decoder part through long-skip connections to supplement detailed information for segmentation. Meanwhile, in order to eliminate noisy information in low-level features, the semantic embedding branch (SEB) [37] is introduced to modulate the low-level features with high-level semantic information. VR segmentation acts as an auxiliary task to improve VP detection accuracy and will be removed during the testing phase.

A. VP REGRESSION

The base network for VP regression contains two convolutional layers (DouConv2d) and four residual connection blocks (ResConv2d) and is shared between different tasks, as shown in Table 1. VP regression-specific layers consist of one max-pooling layer, two convolutional layers, one average-pooling layer and one fully connected layer that directly outputs the coordinates of the VP, as shown in

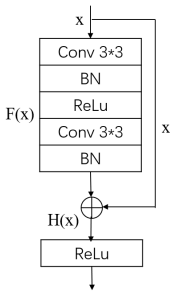


FIGURE 4. Residual connection unit. F(x) is residual mapping and H(x) is identity mapping. Residual connection blocks (ResConv2d) is formed by three cascaded residual connection units.

the last row of Table 1. Each ResConv2d block in Fig.3. is formed by three cascaded residual connection units showed in Fig. 4. The residual connections in ResConv2d are shortcut connections between convolutional layers, fitting them to residual mappings F(x), as shown in Fig.4. It is easier to learn residual mapping that approximates to zero rather than original identity mapping H(x). If input x has different size or channel with residual F(x), the x should firstly go through a convolutional layer to get desired size or channel before addition.

B. RAIL SEGMENTATION

The encoder-decoder structure is used for rail segmentation to combine the shared feature extraction layers and task layers. The base network encodes the input information, while the decoder part and task layers map the encoded low-resolution semantic features into original input size. In the decoding

TABLE 1. The proposed network structure contains shared base network (encoder part), decoder and three task-specific layers.

	input	block	channel	stride
shared base network encoder	3*300*300	DouConv2d	16	1
	16*300*300	ResConv2d①	16	1
	16*300*300	ResConv2d②	16	2
	16*150*150	ResConv2d③	32	2
	32*75*75	ResConv2d④	64	2
decoder	32*75*75, 64*38*38	SEB①	32	1
	64*38*38	Upsample	-	-
	96*76*76	DouConv2d	64	1
	16*150*150, 32*75*75, 64*38*38	SEB②	16	1
	64*76*76	Upsample	-	-
	80*152*152	DouConv2d	32	1
	16*300*300, 16*150*150, 32*75*75, 64*38*38	SEB③	16	1
	32*152*152	Upsample	-	-
	48*304*304	DouConv2d	16	1
VR segmentation	16*304*304	DouConv2d	16	1
	16*304*304	Conv2d	2	1
rail segmentation	16*304*304	DouConv2d	16	1
	16*304*304	Conv2d	2	1
VP regression	64*38*38	Max Pooling	-	2
	64*19*19	DouConv2d	64	2
	64*10*10	Average Pooling	-	1
	64*1	Fully Connected layer	-	-

The kernel sizes of Max Pooling and Average Pooling are 3 and 10 respectively. The Conv2d block consists of Conv 3*3, batch normalization and ReLu. A DouConv2d block is cascaded with two Conv2d blocks.

process, if only the high-level semantic features are used, the detailed information will be lost, resulting in inaccurate predictions. Since low-level features generated by encoding process contain more detailed information, we concatenate multiple low-level features with up-sampled high-level semantic features and then feed them into subsequent convolution layers. Until the size of up-sampled feature is the same as the one of original input, pixel-level predictions are realized through the last convolution layer. As shown in Table 1, the size of up-sampled output is 304*304, 4 pixels more than the original size, so the ground-truth image should pad 2 pixels around each border. Because the low-level features are noisy and we hope the reused low-level information is relatively semantic, the information modulation module SEB [37] is introduced by adding multi-level semantic information to low-level features. In this case, the inputs of each SEB consist of only one lower-level feature Lf and several higher-level features $Hf_i, i \in \{1, 2, \ldots, L\}$, that come from the outputs of ResConv2ds, as shown in Table 1 and Fig.3. There are $L = 1, 2, 3$ higher-level features for three SEB modules respectively, as shown in blue dotted lines in Fig.3. The only one lower-level feature Lf is shown in the black solid line in Fig.3. Thus, the output of each SEB is the superposition of lower-level features modulated by several higher-level features:

$$\begin{aligned} out &= \sum_{i=1}^{L} Lf \times F(Hf_i), \\ F(Hf_i) &= \text{Upsample}(\text{Conv2d}(Hf_i)). \end{aligned} \tag{1}$$

where operation $\times$ represents element-wise multiplication. The Conv2d(Hf_i) has the same number of convolution kernels as the channels of Lf for element-wise multiplication in (1). The details of the SEB② in Fig.3 is shown in Fig.5.

C. VR SEGMENTATION

The VR segmentation is an auxiliary task, used to improve VP detection accuracy in the training phase and will be removed during the test phase. Since VR is a fixed area centered on the VP, the information of VR can be very helpful to VP prediction. Because VR segmentation and the rail segmentation almost share the whole encoder-decoder structure, except for the last few convolution layers, so the framework is still computationally efficient.

D. LOSS FUNCTION OF MULTI-TASK LEARNING

In the process of multi-task learning, the proposed network model is optimized with respect to three objective losses.

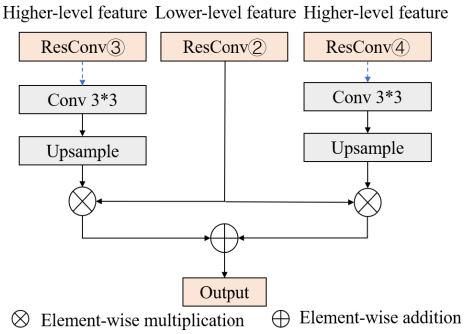


FIGURE 5. The details of the SEB② in Fig 3. The input contains one lower-level feature from ResConv2d② and two higher-level features that are ResConv2d③ and ResConv2d④.

Here we use the weighted linear sum of losses for each individual task as the total loss function:

$$L_{\text{total}}(\boldsymbol{W}) = \alpha_1 L_{\text{VP}}(\boldsymbol{W}) + \alpha_2 L_{\text{VR}}(\boldsymbol{W}) + \alpha_3 L_{\text{rail}}(\boldsymbol{W}) \quad (2)$$

where $\alpha_1, \alpha_2, \alpha_3$ are weights and $\boldsymbol{W}$ represents network parameters. In general, weight selection affects the performance of each task significantly and has to be made with great cautions. Here we adopt Bayesian probabilistic theory and treat weights as independent variables that can be learned through training [34]. We assume that the multiple learning tasks have homoscedastic uncertainty and can be modeled by Gaussian likelihood. The homoscedastic uncertainty is approximately constant and depends on different tasks instead of input data, which is also called task uncertainty.

For a regression task, the likelihood fits a Gaussian distribution with a variance σ_{VP} and a mean given by network output. The likelihoods of classification tasks are represented as a scaled version of the model output with scalars $\sigma_{\text{VR}}, \sigma_{\text{rail}}$ squashed through a Softmax function. According to the above-mentioned assumptions and the derivations of [34], the final adaptive weighted loss can be formulated as follows:

$$\begin{aligned} &L_{\text{total}}(\boldsymbol{W}, \sigma_{\text{VP}}, \sigma_{\text{VR}}, \sigma_{\text{rail}}) \\ &= \frac{1}{2\sigma_{\text{VP}}^2} L_{\text{VP}}(\boldsymbol{W}) + \frac{1}{\sigma_{\text{VR}}^2} L_{\text{VR}}(\boldsymbol{W}) + \frac{1}{\sigma_{\text{rail}}^2} L_{\text{rail}}(\boldsymbol{W}) \\ &\quad + \log \sigma_{\text{VP}} + \log \sigma_{\text{VR}} + \log \sigma_{\text{rail}} \end{aligned} \quad (3)$$

the last three terms in (3) can be regarded as regularization terms for the variables $\sigma_{\text{VP}}, \sigma_{\text{VR}}, \sigma_{\text{rail}}$. In order to avoid zero denominator in (3), we define $s_{\text{VP}} = \log \sigma_{\text{VP}}^2, s_{\text{VR}} = \log \sigma_{\text{VR}}^2, s_{\text{rail}} = \log \sigma_{\text{rail}}^2$, so the final formulation for multi-task loss function is:

$$\begin{aligned} &L_{\text{total}}(\boldsymbol{W}, s_{\text{VP}}, s_{\text{VR}}, s_{\text{rail}}) \\ &= \frac{1}{2\exp(s_{\text{VP}})} L_{\text{VP}}(\boldsymbol{W}) + \frac{1}{\exp(s_{\text{VR}})} L_{\text{VR}}(\boldsymbol{W}) \\ &\quad + \frac{1}{\exp(s_{\text{rail}})} L_{\text{rail}}(\boldsymbol{W}) + \frac{s_{\text{VP}}}{2} + \frac{s_{\text{VR}}}{2} + \frac{s_{\text{rail}}}{2} \end{aligned} \quad (4)$$

each optimal weight can be learned through the loss function optimization process during network training. We set initial values $s_{\text{VP}} = s_{\text{VR}} = s_{\text{rail}} = 5$, which were also used in [34].

In our case, VP regression task adopts SmoothL1 loss:

$$\begin{aligned} L_{\text{VP}}(p, t) &= \sum_{i \in \{x,y\}} Sm_{\text{L1}}(p_i - t_i), \\ Sm_{\text{L1}}(p_i - t_i) &= \begin{cases} 0.5(p_i - t_i)^2 & \text{if } |p_i - t_i| < 1 \\ |p_i - t_i| - 0.5 & \text{otherwise} \end{cases} \\ t_{\text{x}} &= \frac{T_{\text{x}}}{\text{w}}, \quad t_{\text{y}} = \frac{T_{\text{y}}}{\text{h}} \end{aligned} \quad (5)$$

here $(T_{\text{x}}, T_{\text{y}})$ are absolute coordinates, (w, h) are image size, p_i is the predicted value and t_i is the ground truth. SmoothL1 loss is often used in object detection, which can avoid gradient explosion in training. Moreover, the VP coordinates are normalized in loss computation, resulting in a fast convergence. Image segmentation is a pixel-level classification task. In our case, it is a binary classification task and the loss function adopts cross entropy. Because of unbalanced object and background pixels, we adopt weighted cross entropy:

$$\begin{aligned} L_{\text{VR}}(p, t) &= L_{\text{rail}}(p, t) \\ &= -\sum_{i \in \text{w*h}} \beta p_i \ln t_i + (1 - p_i) \ln(1 - t_i) \end{aligned} \quad (6)$$

where p_i is the predicted value and t_i is the ground truth, (w, h) are image size and β is set to 5 empirically.

IV. EXPERIMENTS AND RESULTS

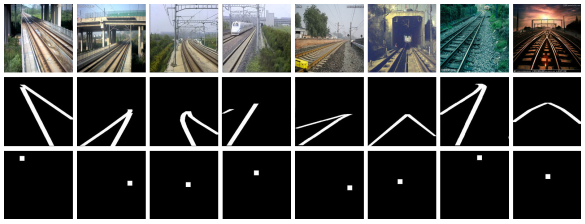
A. DATASET

Images in the dataset used in this paper come from three sources. First, we collect 3800 images captured by 12 cameras installed on the Shanghai-Ningbo Railway Line near the Shanghai Hongqiao Station, China. The cameras are fixed on catenary posts, about 2.5 meters above the rail surface. There exists illumination change, camera jitter, background dynamics and train occlusion in these images, as shown in Columns 1-3 of Fig. 6(a). Second, we collect 750 images from two PTZ cameras, fixed temporarily on tripods at Nanjing Railway Line during nighttime and Beijing Ring Railway Test Base during daytime, respectively, as shown in Columns 4-5 of Fig. 6(a). Third, we download 500 images of railway scenes from Google Image and Baidu Image. These web pictures have not only various railway scenes but also different color styles, as shown in last column of Fig. 6(a). In the experiment, we collected most of the typical railway scenes. The dataset contain:

1) different weather conditions: sunny, rainy, foggy and wind
2) different line types: straight line and curved line
3) different surrounding environment: cross bridge, tunnel, buildings, target occlusion
4) different light conditions: day and night
5) different style scenes from website: different color style and different content

For each image in the collected dataset, the VP position and rails are labeled manually. Since the alarm region is the area between the two outermost rails, only two outermost rails

(a) Collected dataset. Columns 1-3: Images from cameras installed along the railway line. Columns 4-5: Images from PTZ cameras on the railway scene. Column 6: Images from website.



(b) Dataset labels. The original images are shown in the first row. The second row shows two outermost rails determining the alarm region. VR is shown in the last row.

FIGURE 6. Dataset.

are labeled, as shown in Fig. 6(b). VR is labeled as a square region centered on VP, the size of which is 0.1 times of input image size. We resize the input images to 300*300, so the VR size is 30*30, as shown in Fig. 6(b). In order to simulate the camera angle changes on the fixed scenes, images whose VPs are more than 30 pixels away from the image center are horizontally flipped. There are totally 5400 images for training and 1000 images for testing.

B. METRIC

VP detection accuracy is defined as the proportion of testing samples whose predicted VP falls into the neighborhood of ground truth:

$$A_e = \frac{1}{N}\sum_{i=1}^{N} I(x_i, y_i, \hat{x}_i, \hat{y}_i),$$
$$I(x_i, y_i, \hat{x}_i, \hat{y}_i) = \begin{cases} 1 & \text{if } |x_i - \hat{x}_i| < e \text{ and } |y_i - \hat{y}_i| < e \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

where x_i and y_i are predicted coordinate of VP, $\hat{x}_i$ and $\hat{y}_i$ are ground truth, and e is the error range that defines the size of the neighborhood, or the error tolerance level. In order to evaluate the performance of the VP regression model for different error tolerance levels, we select three error ranges, i.e., 0.05, 0.1, and 0.15. For input image of size 300*300, the absolute errors are 15, 30 and 45 pixels respectively.

For rail segmentation, although only the leftmost and rightmost rails in the scene are labeled, the unlabeled rails will also be detected since the network can learn the general rail features during the training. Almost all the unwanted but detected rails are located between the leftmost and rightmost rails and do not affect our alarm region extraction. The outermost rails are closely related to True Positive and False Negative. For this reason, we adopt Recall as the rail segmentation accuracy to measure completeness of the two outermost rails that will be used to determine the alarm region:

$$\text{Recall} = \frac{TP}{TP + FN} \quad (8)$$

where TP is true positive, FN is false negative.

C. EXPERIMENTAL RESULTS AND MODEL ANALYSIS

For all experiments, we use an initial learning rate of 0.01, weight decay rate of 0.95, regularization term of 0.0001 and batch size of 16. The network parameters are randomly initialized and the visual VP detection results on various scenes, including straight rail lines, curved rail lines, nighttime and train objects scenes, are shown in Fig. 7. For each sample image in Fig. 7, the original image containing VP ground truth (blue circle) and detection result (red rectangle), rail ground truth and the rail segmentation result are listed from left to right. It can be seen that, although only leftmost and rightmost rails in the main line are labeled in ground truth, the trained network can detect other rails. Some feature maps of the ResConv2d③ in Fig. 3 are showed in Fig. 8. It can be found that convolution kernels are sensitive to different objects, such as VP in the second column, rails in the columns 6-7 and both objects in columns 3-5.

To analyze the impacts of the auxiliary training task and the information modulation module SEB on VP detection and rail segmentation, we compared the following four network structures on the railway dataset:

1) VP regression task/rail segmentation, marked as VPreg/RAILseg
2) VP regression and rail segmentation, marked as (VPreg, RAILseg)
3) VP regression, rail segmentation and VR segmentation, marked as (VPreg, RAILseg, VRseg)
4) VP regression, rail segmentation, VR segmentation and SEB module, marked as (VPreg, RAILseg, VRseg, SEB)

The comparisons for VP prediction and rail segmentation are shown in Table 2 and Table 3 respectively, where $A_{0.05}$, $A_{0.1}$ and $A_{0.15}$ stand for error ranges of 0.05, 0.1 and 0.15, used in (7) to evaluate the accuracy of VP prediction, respectively.

As shown in Table 2, the introduction of multi-task training and SEB module can improve VP prediction. This is due to the fact that auxiliary training makes the model easier to learn relevant features. This process is further strengthened by adding SEB modules that introduces semantics into low-level information. We add the setup (VPreg, RAILseg, SEB) to further compare the effects of VRseg and SEB in VP detection. The SEB module has greater effects than VRreg. At the same time, the SEB module enhances the segmentation performance, as shown in Table 3. Because the low-level information is noisy, unrelated information can be discarded after semantic modulation. The memory usage and computation load of the network are shown in Table 4. Because of the shared feature extraction layers, the auxiliary task only occupies 10.29% of the memory usage and 7.5% of the floating-point operations during training.

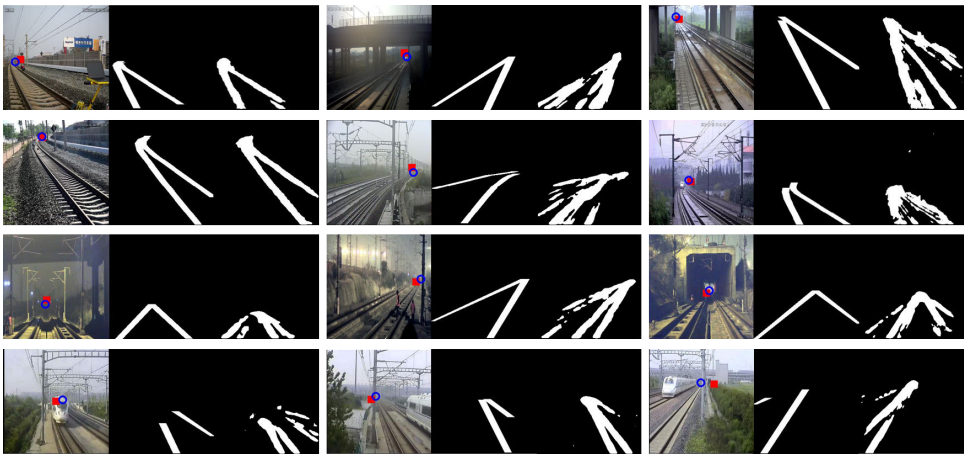


FIGURE 7. VP detection and rail segmentation results on various scenes. The result for each sample image contains the original image, ground truth of two outermost rails and segmented rails. The red rectangles and blue circles in original images are predicted VP and ground truth respectively. Fist row: scenes of straight rail lines; Second row: scenes of curved rail lines; Third row: scenes in nighttime; Last row: scenes with trains.

FIGURE 8. Feature maps from deep multi-task leaning model.

TABLE 2. The influence of auxiliary training and information modulation module SEB for VP detection.

Network architecture	VP prediction		
	$A_{0.05}$	$A_{0.1}$	$A_{0.15}$
VPreg	0.44	0.79	0.88
(VPreg, RAILseg)	0.51	0.81	0.89
(VPreg, RAILseg, VRseg)	0.55	0.80	0.95
(VPreg, RAILseg, SEB)	0.56	0.83	0.97
(VPreg, RAILseg, VRseg, SEB)	0.57	0.87	0.97

D. COMPARISON WITH OTHER VP DETECTION METHODS

The proposed VP detection method is also compared with single regression network (SRN) [24], dominate VP detection network (DON) [7], region prediction network (RPN) [23] and Hough transform method (HFT) [19]. The SRN used similar residual network as ours for feature extraction. DON used a two-stream network that fused multi-level features at different stages and formulated VP detection with a regression task and a classification task. RPN defined four rectangle regions according to VP and image corners and classified each pixel into one of these classes. The network is similar to our encoder-decoder network without SEB. Final VP is the intersection of the predicted regions. The performances of different VP detection algorithms on our railway dataset are shown in Fig. 9(a). Some prediction examples are shown in Fig. 9(b).

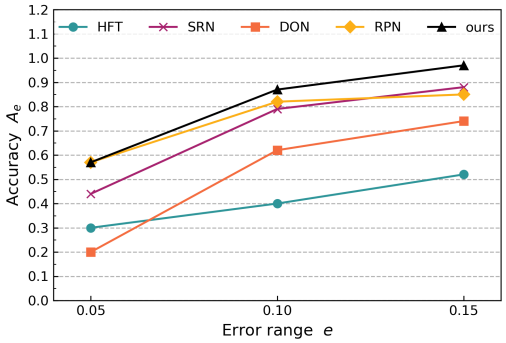
TABLE 3. The influence of auxiliary training and information modulation module SEB for rail segmentation.

Network architecture	Recall
RAILseg	0.85
(VPreg, RAILseg)	0.85
(VPreg, RAILseg, VRseg)	0.85
(VPreg, RAILseg, VRseg, SEB)	0.86

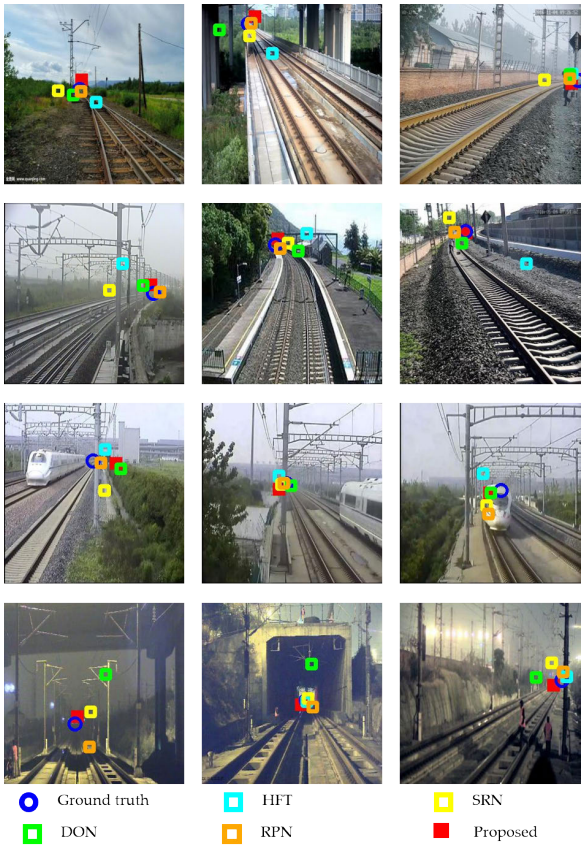
TABLE 4. The memory usage and computation load for different phases.

	Memory size(MB)	GFlops
Train	335.87	5.86
Test	301.3	5.42
Test (VP regression only)	156.82	2.45

It can be seen that, among all five methods, HFT and DON have the worst performances. The traditional method HFT is based on edge detection and Hough Transform and is prone to accumulated errors, resulting in a worse performance compared with other end-to-end neural networks. Although it is an unsupervised method, it still needs prior knowledge from samples to set up proper restrictions and parameters manually, so it is only suitable for uniform scenes. For the two-stream structure of DON, there exists the problem of unbalanced samples in classification task, especially when block size is small and thus VP hardly locates near the bottom of image. SRN and RPN have similar performances, especially for medium and large tolerance errors. The prediction result of RPN relies on the intersection of different regions. If predictions were noisy, the final computed VP was noisy



(a) The comparison results.



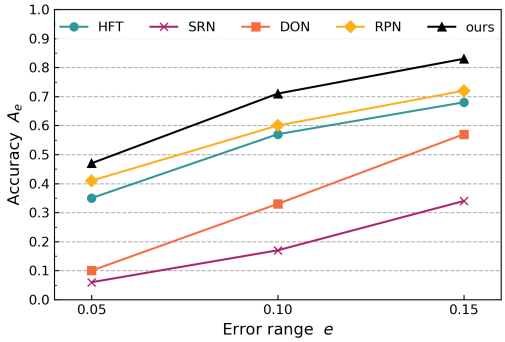
(b) Examples of VP detection results. The VP computed by HFT in left sample of the last row was not found inside image because of strong noise.

FIGURE 9. Compared with different VP detection methods (HFT, SRN, DON, RPN) on railway test data.

as well. This method was mainly applied to on-board camera images for automatic driving where the scenes are relatively uniform with only one VP. However, the railway scenes are much more complicated because of two facts: (1) railway cameras are installed on much higher positions and can see wider and further surrounding environment; (2) there are abundant distractions, such as catenary posts and overhead lines and beams. SRN has similar performance as our method at small error tolerance level. However, at medium and large error tolerance levels, our method achieves better performance. Compared with SRN, multi-task learning of our method introduces implicit data augmentation and regularization terms, so over-fitting can be prevented by balancing different noise patterns, resulting in a better generalization.

Besides railway dataset, we also evaluate the effectiveness of our method on 600 highway images selected from [22], as shown in Fig. 10(a). It can be seen from Fig. 10(b) that the proposed model achieves the highest test accuracy among all five methods on the highway dataset as well. Compared with other network structures, the deeper network structure and multi-task training of our method enhance the feature representation and generalization abilities. It can also be found that, because the road scenes are more uniform, parameter tuning in HFT is easier and higher detection accuracy can be achieved.

(a) Sample images of highway test dataset.



(b) The comparison results.

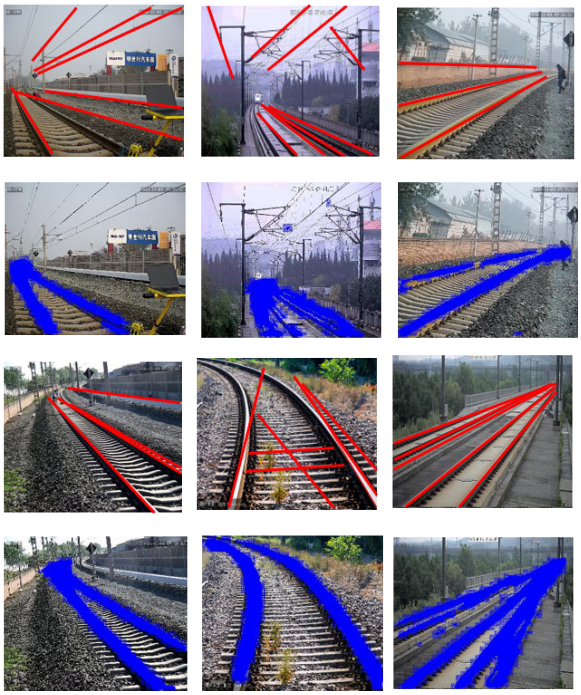
FIGURE 10. Compared with different VP detection methods (HFT, SRN, DON, RPN) on highway test dataset.

In the system configuration of pytorch 0.4-win10-NVIDIA GeForce GTX 1080 Ti, we only compare the speed of deep learning based VP detection methods. Because they run on GPU and are faster than HFT by one order of magnitude, which only runs on CPU. The SRN(389 FPS) and DON(402 FPS) have similar speed with ours(345 FPS). The RPN(199 FPS) is obviously lower than others, because the network contains a time-consuming decoder part.

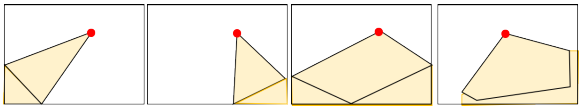
E. APPLICATION OF THE PROPOSED METHOD IN RAILWAY PERIMETER INTRUSION DETECTION

The detection of foreign object intrusion into railway lines requires the knowledge of alarm region that can be determined by using the leftmost and rightmost rails. As shown in Fig. 11(a), it is difficult for the traditional Hough transform based method to distinguish rails from overhead lines and catenary posts, while the results of VP detection and rail segmentation produced by the proposed method can provide accurate information.

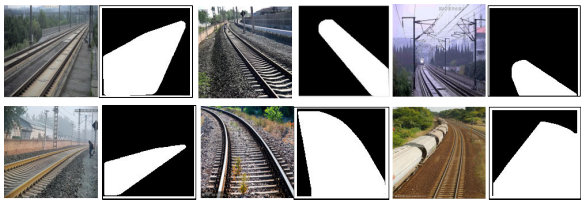
Alarm region extraction process based on VP and rail segmentation consists of a series of morphological operations, such as connected region analysis and convex hull analysis. The procedure is summarized as follows:



(a) Hough-based (red line) and deep learning-based (blue line) rail extraction.



(b) Alarm region extraction based on morphological operations. Red circle represents VP. Black border polygon is convex hull. Yellow polygon is final alarm region.



(c) Extracted alarm region in railway.

FIGURE 11. Railway perimeter intrusion detection based on VP detection and rail segmentation.

1) Open operation for de-noising
2) Connected region analysis for removing small regions or regions above the VP
3) Convex hull analysis: the major generated shapes are black border polygons in Fig. 11(b), red circles are predicted VPs.
4) The complete railway area is extracted according to the relative position of the convex hull endpoint and the image border, as shown in yellow polygons in Fig 11(b). The final extracted alarm region in railway is shown in Fig 11(c).

V. CONCLUSION

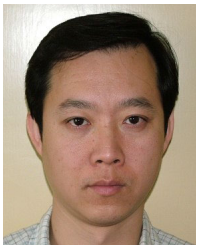
This paper proposed a multi-task learning framework for simultaneous detection of VP and rails. It has small computation load, which is of paramount importance for real-time processing. In the applications of railway perimeter intrusion detection, the detected VP and rails can be used in alarm region determination, and VP position can also be used to guide adaptive multi-scale object detection. By introducing multiple relevant learning tasks and learning more independent noise patterns, the proposed deep network can greatly improve its generalization ability, which is very important for applications with very small dataset. For the case of VP at the intersection of curves, it is difficult for traditional HFT based method that depends on extracted features, such as edges, and special defined constraints, while the proposed algorithm can still detect VP reliably. For cases where VP is located outside of the image, existing deep learning methods may fail to detect VP, while the proposed method can still be used for rail segmentation and alarm region determination. The existing image database basically covers most of typical application conditions, listed in Section IV-A. However, there is still a lack of samples under extreme weather conditions such as rainstorm, snowstorm and fog, which are difficult to obtain and will continue to be collected in the future. The vision image is sensitive to light changes and viewpoint, so multi-source data fusion is worth exploring, such as lidar data and vision data fusion used in traffic sign detection [38]. More questions about multi-task learning, such as optimal separation between shared network and task-specific networks, and the interaction among tasks, need further study.

REFERENCES

[1] T. Bouwmans, "Traditional and recent approaches in background modeling for foreground detection: An overview," *Comput. Sci. Rev.*, vols. 11–12, pp. 31–66, May 2014.
[2] H. Kong, J.-Y. Audibert, and J. Ponce, "Vanishing point detection for road detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 96–103.
[3] J. H. Yoo, S.-W. Lee, S.-K. Park, and D. H. Kim, "A robust lane detection method based on vanishing point estimation using the relevance of line segments," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 12, pp. 3254–3266, Dec. 2017.
[4] H. Wildenauer and A. Hanbury, "Robust camera self-calibration from monocular images of manhattan worlds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2831–2838.
[5] Z. Zhou, F. Farhat, and J. Z. Wang, "Detecting dominant vanishing points in natural scenes with application to composition-sensitive image retrieval," *IEEE Trans. Multimedia*, vol. 19, no. 12, pp. 2651–2665, Dec. 2017.
[6] A. Borji, "Vanishing point detection with convolutional neural networks," 2016, *arXiv:1609.00967*. [Online]. Available: http://arxiv.org/abs/1609.00967
[7] X. Zhang, X. Gao, W. Lu, L. He, and Q. Liu, "Dominant vanishing point detection in the wild with application in composition analysis," *Neurocomputing*, vol. 311, pp. 260–269, Oct. 2018.
[8] W. Liu, Z. Liu, H. Wang, and Z. Han, "An automated defect detection approach for catenary rod-insulator textured surfaces using unsupervised learning," *IEEE Trans. Instrum. Meas.*, early access, Apr. 13, 2020, doi: 10.1109/TIM.2020.2987503.
[9] Q. Guo, L. Liu, W. Xu, Y. Gong, X. Zhang, and W. Jing, "An improved faster R-CNN for high-speed railway dropper detection," *IEEE Access*, vol. 8, pp. 105622–105633, 2020.
[10] D. He, Z. Yao, Z. Jiang, Y. Chen, J. Deng, and W. Xiang, "Detection of foreign matter on high-speed train underbody based on deep learning," *IEEE Access*, vol. 7, pp. 183838–183846, 2019.
[11] D. Wei, D. Suo, L. Jia, and Y. Li, "Multi-target defect identification for railway track line based on image processing and improved yolov3 model," *IEEE Access*, vol. 8, pp. 61973–61988, 2020.

[12] G. Kang, S. Gao, L. Yu, and D. Zhang, "Deep architecture for high-speed railway insulator surface defect detection: Denoising autoencoder with multitask learning," *IEEE Trans. Instrum. Meas.*, vol. 68, no. 8, pp. 2679–2690, Aug. 2019.
[13] X. Gibert, V. M. Patel, and R. Chellappa, "Deep multitask learning for railway track inspection," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 1, pp. 153–164, Jan. 2017.
[14] W. Yang, B. Fang, and Y. Y. Tang, "Fast and accurate vanishing point detection and its application in inverse perspective mapping of structured road," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 48, no. 5, pp. 755–766, May 2018.
[15] W. Yang, X. Luo, B. Fang, D. Zhang, and Y. Yan Tang, "Fast and accurate vanishing point detection in complex scenes," in *Proc. 17th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2014, pp. 93–98.
[16] J.-P. Tardif, "Non-iterative approach for fast and accurate vanishing point detection," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 1250–1257.
[17] Y. Li, W. Ding, X. Zhang, and Z. Ju, "Road detection algorithm for autonomous navigation systems based on dark channel prior and vanishing point in complex road scenes," *Robot. Auto. Syst.*, vol. 85, pp. 1–11, Nov. 2016.
[18] W. Ding, Y. Li, and H. Liu, "Efficient vanishing point detection method in unstructured road environments based on dark channel prior," *IET Comput. Vis.*, vol. 10, no. 8, pp. 852–860, Dec. 2016.
[19] K. Tarrit, J. Molleda, G. A. Atkinson, M. L. Smith, G. C. Wright, and P. Gaal, "Vanishing point detection for visual surveillance systems in railway platform environments," *Comput. Ind.*, vol. 98, pp. 153–164, Jun. 2018.
[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.
[22] C.-K. Chang, J. Zhao, and L. Itti, "DeepVP: Deep learning for vanishing point detection on 1 million street view images," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2018, pp. 1–8.
[23] S. Lee, J. Kim, J. S. Yoon, S. Shin, O. Bailo, N. Kim, T.-H. Lee, H. S. Hong, S.-H. Han, and I. S. Kweon, "VPGNet: Vanishing point guided network for lane and road marking detection and recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 1947–1955.
[24] Y. Shuai, Y. Tiantian, Y. Guodong, and L. Zize, "Regression convolutional network for vanishing point detection," in *Proc. 32nd Youth Academic Annu. Conf. Chin. Assoc. Autom. (YAC)*, May 2017, pp. 634–638.
[25] H.-S. Choi, K. An, and M. Kang, "Regression with residual neural network for vanishing point detection," *Image Vis. Comput.*, vol. 91, Nov. 2019, Art. no. 103797, doi: 10.1016/j.imavis.2019.08.001.
[26] R. Caruana, "Multitask learning," *Mach. Learn.*, vol. 28, no. 1, pp. 41–75, 1997.
[27] J. Baxter, "A bayesian/information theoretic model of learning to learn via multiple task sampling," *Mach. Learn.*, vol. 28, no. 1, pp. 7–39, 1997.
[28] S. Ruder, "An overview of multi-task learning in deep neural networks," 2017, *arXiv:1706.05098*. [Online]. Available: http://arxiv.org/abs/1706.05098
[29] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
[30] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1440–1448.
[31] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.
[32] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2961–2969.
[33] R. Ranjan, V. M. Patel, and R. Chellappa, "HyperFace: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 1, pp. 121–135, Jan. 2019.
[34] R. Cipolla, Y. Gal, and A. Kendall, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7482–7491.
[35] Z. Zhang, P. Luo, C. C. Loy, and X. Tang, "Facial landmark detection by deep multi-task learning," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2014, pp. 94–108.
[36] S. Li, Z.-Q. Liu, and A. B. Chan, "Heterogeneous multi-task learning for human pose estimation with deep convolutional neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2014, pp. 482–489.
[37] Z. Zhang, X. Zhang, C. Peng, X. Xue, and J. Sun, "Exfuse: Enhancing feature fusion for semantic segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 269–284.
[38] C. Liu, S. Li, F. Chang, and Y. Wang, "Machine vision based traffic sign detection methods: Review, analyses and perspectives," *IEEE Access*, vol. 7, pp. 86578–86596, 2019.

XINGXIN LI received the B.S. degree from Beijing Jiaotong University, in 2014, where she is currently pursuing the Ph.D. degree with the School of Mechanical, Electronic, and Control Engineering. Her main research interests include machine vision detection technology and deep learning.



LIQIANG ZHU (Member, IEEE) received the Ph.D. degree from Arizona State University, USA, in 2004. He is currently an Associate Professor with the School of Mechanical, Electronic, and Control Engineering, Beijing Jiaotong University. He is also a member of the Key Laboratory of Vehicle Advanced Manufacturing, Measuring and Control Technology, Beijing Jiaotong University. His main research interest includes intelligent detection technology. He has been engaged in the research of intelligent information processing, machine vision, and nondestructive testing for a long time. His related achievements have been widely used in the safety detection systems of high-speed railway, subway, and other rail transit systems.

ZUJUN YU received the Ph.D. degree from Beijing Jiaotong University, in 2008. He is currently a Professor with Beijing Jiaotong University. His main research interest includes measurement and control systems.

BAOQING GUO received the Ph.D. degree from Beijing Jiaotong University, in 2009. He is currently an Associate Professor with Beijing Jiaotong University. His main research interests include detection and diagnosis of railway infrastructure, and machine vision detection technology.

YANQIN WAN received the B.S. degree from Beijing Jiaotong University, in 2014, where she is currently pursuing the Ph.D. degree. Her main research interests include action recognition and deep learning.