Research on the Segmentation and Extraction of Scenes along Railway Lines Based on Remote Sensing Images of UAVs

Lei Tong[1, 2, 3], Limin Jia[1, 2, 3], Zhipeng Wang[1, 2, 3(✉)], Yunpeng Wu[1] and Ning Wang[1]

[1] State Key Lab of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing, China
[2] National Engineering Laboratory for System Safety and Operation Assurance of Urban Rail Transit, Beijing Jiaotong University, Guangdong, China
[3]Beijing Research Center of Urban Traffic Information Sensing and Service Technologies, Beijing Jiaotong University, Beijing, China
zpwang@bjtu.edu.cn

Abstract. At present, the manual inspection along railway lines is still a major method to ensure railway operation safely, but the cost is high and work efficiency is low. Therefore, unmanned aerial vehicles (UAVs) patrol inspection is required. This paper presents the effective segmentation of scenes along railway lines (SRL) from remote sensing perspective of UAVs based on the full convolutional networks (FCN). Firstly, the datasets needed in this research are collected and produced from Langfang section of the Beijing-Shanghai high-speed rail-way. The datasets are expanded by using data augmentation to constrain the overfitting in the training process. Secondly, the segmentation model FCN-8s for SRL is developed and trained. The related setting and hardware environment in the training process are described in this paper. The experimental results show that a single image prediction needs 151.2ms, to achieve 6.6 fps when input size is 384×384. Good accuracy is obtained on the test dataset, i.e., 55.8% MIoU and 70.2% MPA, which meets the expectations of FCN. At the same time, it is also found that the segmentation of railway area achieves the best result thus the railway area is extracted accordingly.

Keywords: Railway Scenes Segmentation, Semantic Segmentation, UAVs.

1 Introduction

In recent years, the UAVs remote sensing technology has been applied to the transmission line inspection [1], and it also provides a fast and effective means for the inspection of railway lines. While in the patrol inspection work, images collected by UAVs are broad, rich in content and high in resolution and both railway infrastructure and surroundings have vital impact on the safe operation of the railway. Therefore, the segmentation and extraction of SRL based on UAVs remote sensing plays an important role in the safety of monitoring railway lines.

In the future, computer vision will become a key tool for the analysis of images

collected by patrol inspection of UAVs along railway lines. As a branch of computer vision, semantic segmentation technology will provide strong support for the segmentation of scenes along railways. Long et al. proposed FCN [2] for image semantic segmentation by adapting the structure of VGG [3] model and add some upsampling layers. On this basis, many models are developed to improve accuracy with different techniques [4-11]. In the work of segmentation of railway scenes using semantic segmentation techniques, Wang et al. proposed an architecture for segmentation of railway regions and optimized contours of orbital regions using polygonal fitting method [12]. He et al. also proposed a semantic segmentation network for segmenting railway scenes and verified the superiority of the network over UNet and FCN networks [13].

However, all these works don't put too much attention on ensuring the safety of railway operation. Furthermore, railway scenes defined in Ref. [12-13] are very different from this research. We concentrate on SRL from the perspective of UAVs while they define railway scenes from the perspective of train cabs. Because FCN is the base model of many of the above models, this study aims to achieve effective segmentation and further extraction of SRL by using FCN model.

2 Dataset

2.1 Build of Dataset

Three predefined locations, i.e. areas A, B and C of the Langfang section of the Beijing-Shanghai high-speed railway, are selected for image collection and the weather conditions are very well on the day of collecting images. According to the requirements of visual range and resolution of the captured images, the flying height of UAVs is set between 80m and 200m. After images collection, images with resolution 3648×5472 (h×w) of SRL are selected, removing inapplicable ones that are repeated, blurred, and angularly offset. 220 images from 430 selected images are used to create semantic segmentation datasets for segmentation of SRL, in which 200 images collected from A and B are used to build the training and validation dataset, and 20 images left collected from C are used to build test dataset.

In the built dataset, scenes are classified into five categories: background, buildings, vegetation, railways and roads, i.e., every pixel in images from the dataset is labelled as one of the above categories. In order to improve the efficiency of labelling process, all original images are scaled to 512×768 and all annotations are implemented using the image annotation tool LabelMe.

In the annotated images, colour black represents background, and colour red, green, yellow, and blue represent buildings, vegetation, railways, and roads respectively. These annotated images need to be converted to grayscale images when training the model.

2.2 Data Augmentation

At present, there are still many difficulties to collect images of SRL with UAVs and the amount of 220 images in the built dataset is far from enough for the training of

deep learning model. In order to improve the training effect, constrain the overfitting in the training process and improve generalization ability of the model, the annotated dataset is expanded by using data augmentation.

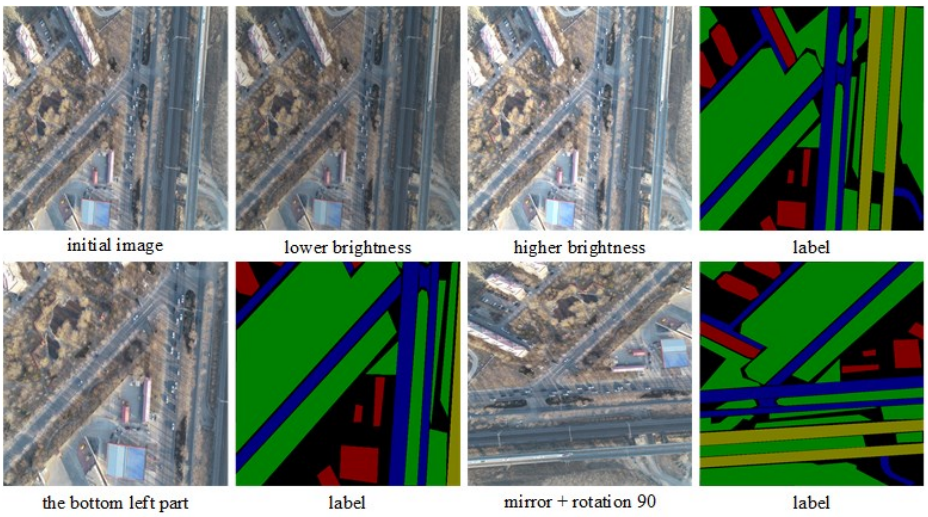

Fig. 1. Examples of data augmentation.

The built dataset can be expanded to 17 times of original size with rotating & horizontal mirroring operation (8 types), brightness transformation operation (4 types) and cropping operation (5 types) of the original images and annotations respectively. Among them, the rotation and horizontal mirroring operation include 8 types in all, i.e., horizontal mirrors (yes, no) × rotation angle (0, 90, 180, 270) where the initial image is represented when the rotation angle is 0 and horizontal mirroring is not used; brightness transformation operation include 5 types in all, i.e., brightness (0.5, 0.8, 1.2, 1.5); cropping operation include 5 types in all, i.e., cropping position (bottom left, bottom right, center, top left and top right). Part of the data augmentation process are shown in Fig. 1. For ease of display, the images are scaled to a square shape. Through above operations, 3400 images are obtained for training and validation and 340 images for test.

3 Segmentation Model for SRL

3.1 Network Architecture of the Model

The FCN model can use a variety of network architecture as its encoder [11], the key part of which is the convolutionalization[1] of fully connected layers and upsampling layers. In the Ref. [2], the authors use different network architectures to com-

[1] For further understanding about the word "convolutionalization" in Fig. 2 of Ref. [2].

pare achieved accuracy, among which the FCN network based on VGG [3] model performs best. Besides, the authors adopt different skip structures in the process of upsampling to the resolution of the input image, i.e., FCN-32s, FCN-16s and FCN-8s, and point out FCN-8s achieves best accuracy. Therefore, FCN-8s model based on VGG-19 is implemented in this paper, which means there is no difference in the first five convolutional and pooling modules between VGG-19 and FCN-8s here, i.e., modules A to E shown in Fig. 2. Also, it is modules from A to E to fine-tune in the transfer learning process which will be mentioned further in the following section 3.2. Subsequent three modules F1 to F3 are all adapted from fully connected layers in VGG-19.

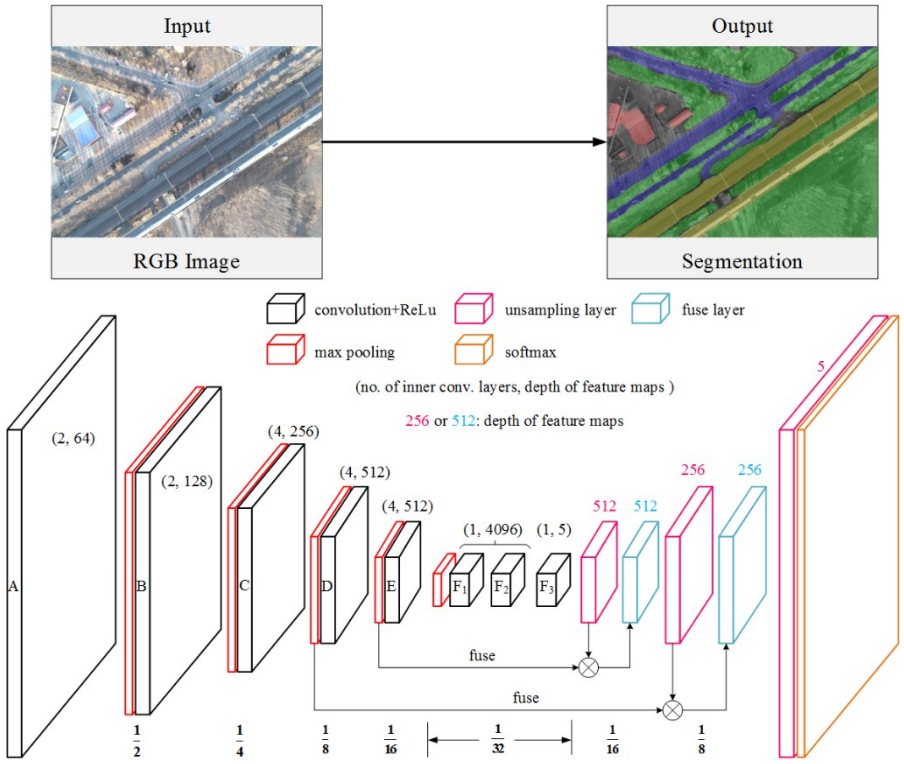

Fig. 2. Network architecture of segmentation model for SRL.

As illustrated in Fig. 2, three upsampling layers are built in the model to increase the resolution of feature maps. The first two upsampling layers restore the resolution of feature maps to 1/16 and 1/8 times of resolution of the original input images respectively, while the third one restores the resolution of input image completely. Obviously, we know that there is the same resolution between the input image and the output image. The results of upsampling layers are fused with the corresponding pooling layers in the encoder [2, 11], combining fine local features in the lower layers and coarse global features in the higher layers to achieve precise feature extraction. The

two fuse layers are obtained by adding the upsampling layer and corresponding pooling layer. Transposed convolution is an important way to achieve upsampling. In this paper, upsampling is implemented with transposed convolution and deep learning framework TensorFlow is used to implement FCN-8s model.

3.2 Training of the Model

Transfer Learning & Cross Validation. Constantly fine-tuning the weights from a pre-trained model is one of the main ways of transfer learning [14]. In addition, it is higher layers to be fine-tuned not lower layers since the lower one tends to contain more generic features that most models share. It is also important to choose an appropriate learning rate, generally a smaller one, when using transfer learning.

In this paper, the pre-trained model VGG-19 is used to accelerate the training process. The initialization of the parameters of the FCN-8s network model is finished by transforming fully connected layers of VGG-19 to convolutional ones and randomly initializing other newly built higher layers, e.g., unsampling layers, etc. In the subsequent training process, training of the model is completed by continuously fine-tuning the parameters in the network with smaller learning rate, especially the high-level parameters that are randomly initialized.

Cross-validation is a method to evaluate the performance of a model. It is mainly achieved by dividing dataset into training part and validation part with different composition. In this paper, K-fold Cross Validation method is adopted to evaluate FCN-8s model. The idea of cross-validation is to divide the dataset into $\boldsymbol{K}$ parts equally, then each of the equal parts is used as validation dataset in turn and the remaining equal parts used as training one respectively. In such case, $\boldsymbol{K}$ trained models can be obtained through multiple training processes. And the average of the accuracies of $\boldsymbol{K}$ models is used as the final performance indicator. In this paper, $\boldsymbol{K}$ is set to 10.

Setup & Environment. The images in the original built dataset are all scaled to 512×768. In order to speed up the training process, the input image is further scaled to a square shape 384×384 into the model during the training process. Some other hyperparameter settings are shown in Tab. 1. The selected Batch Size and Learning Rate are 2 and 10^{-4} respectively and the setting of Iterations No. depends on the training process till the model converges.

Table 1. Hyperparameters setting in the training process.

Hyperparameters	Batch Size	Learning Rate	Iterations No.
Value	2	10^{-4}	Convergence

The training process is finished on the NVIDIA RTX 2080 GPU. All experiments were performed in the TensorFlow deep learning framework, and the model were trained until the loss function converges. And we use the cross-entropy loss function as objective function. In the input mode with a batch size of 2, the total loss is the sum

of the losses of all pixels of all images in the input batch. Training process employed Adam optimizer with a dropout rate of 15% in every convolutional layer of the network.

4 Segmentation & Extraction of SRL

4.1 Experimental Results

Runtime of the Model. After calculation, the running speed of the model built in this paper is summarized in Tab. 2. A single image prediction needs 151.2ms, achieving 6.6 fps when input size is 384×384.

Table 2. Runtime and environment of the model.

GPU	framework	input size	run time	fps
NVIDIA RTX 2080	TensorFlow	384×384	151.2ms	6.6

Accuracy on Validation Dataset. As shown in Tab. 3 and Tab. 4, mean values of the 10 trained models' evaluation metrics is obtained through cross-validation and segmentation performance of SRL is perfect. Since MIoU is a standard metric for semantic segmentation tasks, cross-validation results for different classes of IoU are listed here in particular, as shown in Tab. 4. From Tab. 4, it is illustrated that IoU of the railway area is the highest and the segmentation performance is the best, which has a close relationship with more regular geometry shape and specific color characteristics of the railway area. Prediction results on validation dataset is shown in Fig. 3 (a) and perfect performance on segmentation work is realized as we can see.

Table 3. Cross-Validation results on validation dataset.

Metrics	PA	MPA	MIoU	FWIoU
Mean value	91.3	90.5	81.9	84.5

Table 4. Cross-Validation results of all classes on validation dataset.

Class	background	building	plant	railway	road
Mean of IoU	75.7	72.6	88.0	92.2	81.8

Accuracy on Test Dataset. Since the images in test dataset are collected in area C, which is different from the images from training set and validation dataset (A and B), the evaluation on the test dataset seems to be more objective and accurate, as shown in Tab. 5. Compared with the accuracy metrics on the validation dataset, the accuracy metrics on the test dataset has a certain decline, but still meet the expectation on the accuracy FCN architecture can achieve itself, which is 56% [3]. As illustrated in Tab. 6, the IoU value of the railway area on the test dataset is still much better than the other classes, which maintains a high level compared to the results on the validation

dataset. This objectively and realistically shows that the built model is better for segmentation of the railway area than other classes. Prediction results on test dataset is shown in Fig. 3 (b). However, there is still some fault predicted part, e.g., predicting part of road area as railway and predicting part of building area as road.

Table 5. Cross-validation results on test dataset.

Metrics	PA	MPA	MIoU	FWIoU
Mean value	70.9	70.2	55.8	56.9

Table 6. Cross-validation results of all classes on test dataset.

Class	background	building	plant	railway	road
Mean of IoU	53.3	39.8	57.2	78.6	52.3

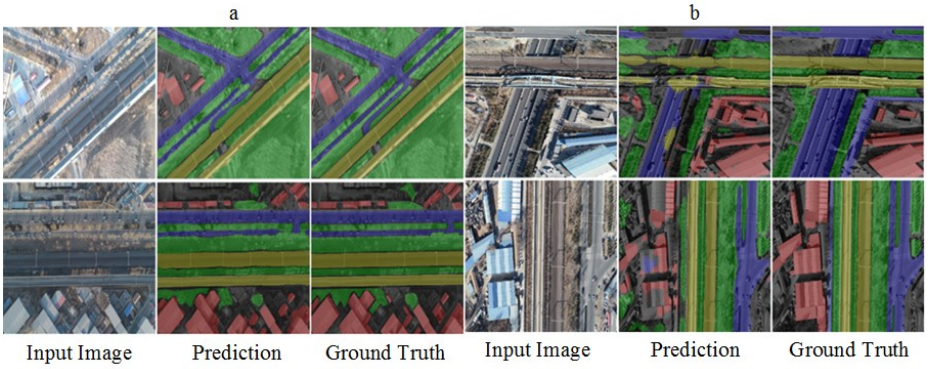

Fig. 3. Prediction results on validation and test dataset. (a) validation part. (b) test part.

4.2 Extraction of Railway Area

According to the segmented image that segmentation model for SRL predict, the different SRL can be extracted separately. Since the model has a high accuracy for the segmentation of railway area, the railway area is extracted here. As illustrated in Fig. 4, the left image is an annotation of some image in the dataset; the right image shows that three areas filled in white are different regions of the same class: green rectangle represents a rectangle that can include a target subarea along horizontal and vertical axis directions of the image, respectively; red rectangle represents a rectangle that can include the a target subarea and has also the smallest area. Some concepts covered in this section are defined as follows:

Rectangular Subgraph: a rectangular part obtained by cropping images in the horizontal and vertical axis directions, e.g., rectangles in green in Fig. 4(b).

Target Subarea: a rectangular part which includes one of the regions representing the same class and has the smallest area, e.g., rectangles in red in Fig. 4(b).

Mask: a grayscale image with the same resolution with the initial image, in which only specific region's values of pixels are set to 1 and the remaining set to 0.

Sub-mask: *Mask* corresponding to a specific *Rectangular Subgraph*.

Sub-label: the label image or predicted image of a specific *Target Subarea*, that is, the corresponding part that *Target Subarea* reflect on the original label image or the predicted image.

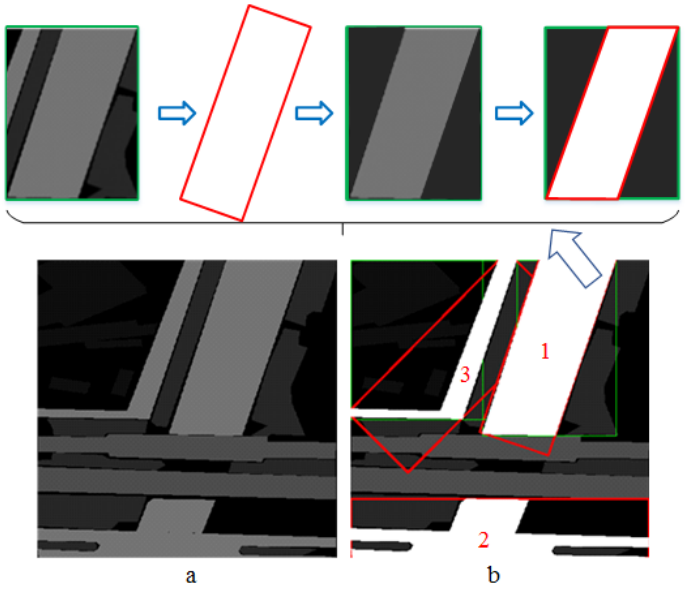

Fig. 4. Concepts for *Rectangular Subgraph*, *Sub-mask*, *Target Subarea* and *Sub-label*. (a) A grayscale label image from the built dataset. (b) Key rectangular areas with different meanings. The extraction process of *Target Subarea* No. 1 is displayed on the top area.

It is easy to find that the *Target Subarea* is contained in its corresponding *Rectangular Subgraph*. Suppose original image as *IMG* and corresponding label image or predicted image as *LBL*. For a specific *Target Subarea*, suppose the *Mask* corresponding to the inner area of its green rectangle as MG_i, the *Mask* on corresponding to the inner area of its red rectangle as MR_i and the *Mask* on *LBL* corresponding to its all pixels belonging to the current class as ML_c. As shown in Fig. 5, the extraction process of all the railway regions can be expressed as follows:

Step 1: Extract the *Rectangular Subgraph* from the original image, i.e., *subgraph*.

$$subgraph(i) = Crop_{MG_i}(IMG) \tag{1}$$

Step 2: Extract the part that MR_i falls on the *subgraph*, i.e., *submask*:

$$submask(i) = Crop_{MG_i}(MG_i \cap MR_i) \tag{2}$$

Step 3: Extract the *Target Subarea*, i.e., *subarea*:

$$subarea(i) = submask(i) \cap subgraph(i) \tag{3}$$

Step 4: Extract the *Sublabel* corresponding to the *Target Subarea*, i.e., *sublabel*:

$$sublabel(i) = submask(i) \cap Crop_{MG_i}(LBL \cap ML_c) \tag{4}$$

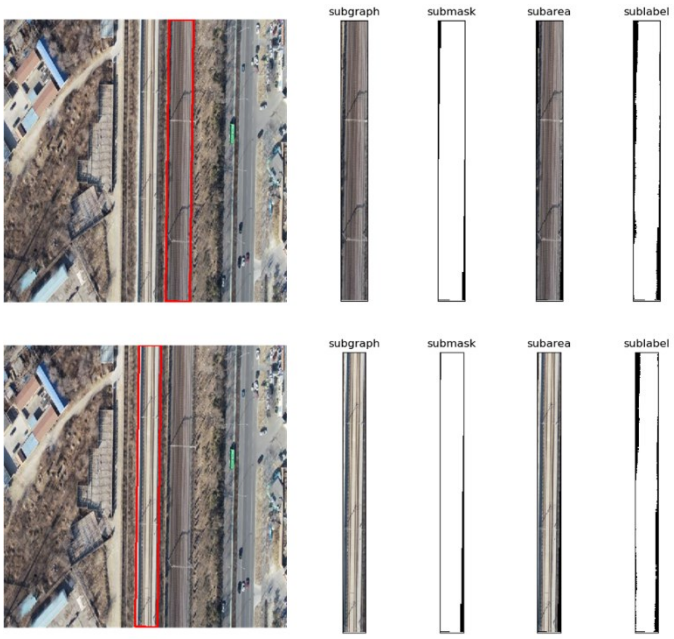

Fig. 5. The extraction of railway area. The *subgraph*, *submask*, *subarea* and *sublabel* here corresponding to the concepts in step 1-4.

In all above formulas, $Crop_{M(i)}(\cdot)$ represents cropping the image in parentheses according to the rectangle indicated by $M(i)$ and the rectangle is required to be in the horizontal and vertical axis directions, while i represents the i-th aggregation area of the pixels belonging to class C. The process of extracting the railway area is illustrated in Fig. 5. Two corresponding sets of *Rectangular Subgraph*, *Sub-mask*, *Target Subarea* and *Sub-label* are respectively extracted.

5 Discussion and Conclusions

Efficient monitoring of SRL can ensure effective and safe operation of railway. This research combines patrol inspection of UAVs and semantic segmentation techniques with safety monitoring along the railway for the first time. Segmentation and extraction of SRL based on UAVs remote sensing images is realized in this paper and the experimental results have demonstrated the effectiveness of the built model. In further work, the accuracy of the model needs to be improved. At the same time, the runtime of the model needs to be further reduced to meet the practical needs.

6 Acknowledgement

This research is supported by the National Key R&D Program of China (No.2016YFB1200203).

References

1. Zhao Z. , Li S., Qi Y., et al.: A Semantic Segmentation Method for Aerial Image of Transmission Line with Improved FCN Model. J. China Sciencepaper 13(14), 1614-1620 (2018). (in Chinese).
2. Long J., Shelhamer E., Darrell T.: Fully Convolutional Networks for Semantic Segmentation. In: Proceedings of 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3431-3440. IEEE, Piscataway (2015).
3. Simonyan K., Zisserman A.: Very Deep Convolutional Networks for Large-scale Image Recognition. In: Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015), arXiv preprint arXiv: 1409.1556v6 (2015).
4. Chen L. C., Papandreou G., Kokkinos I., et al.: Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs. In. Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015), arXiv preprint arXiv: (2015).
5. Chen L. C., Papandreou G., Kokkinos I., et al.: DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. J. IEEE Transactions on Pattern Analysis & Machine Intelligence 40(4), 834-848 (2016).
6. Yu F., Koltun V.: Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122v3, (2016).
7. Paszke A., Chaurasia A., Kim S., et al: Enet: A Deep Neural Network Architecture for Realtime Semantic Segmentation. arXiv preprint arXiv:1606.02147, (2016).
8. Roy A., Todorovic S.: A Multi-scale CNN for Affordance Segmentation in RGB Images. In: European Conference on Computer Vision, pp. 186-201. Springer, Heidelberg (2016).
9. Eigen D., Fergus R.: Predicting Depth, Surface Normals and Semantic Labels with A Common Multi-scale Convolutional Architecture. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2650–2658. IEEE, Piscataway (2015).
10. Bian X., Lim S. N., Zhou N.: Multiscale Fully Convolutional Network with Application to Industrial Inspection. In: 2016 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1-8. IEEE, Piscataway (2016).
11. Badrinarayanan V., Kendall A., Cipolla R.: SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. J. IEEE Transactions on Pattern Analysis and Machine Intelligence 39(12), 2481-2495 (2017).
12. Wang Z., Wu X., Yu G., et al.: Efficient Rail Area Detection Using Convolutional Neural Net-work. J. IEEE Access vol. 6, 77656-77664 (2018).
13. He Z., Tang P., Jin W., et al.: Deep Semantic Segmentation Neural Networks of Railway Scene. In: 2018 37th Chinese Control Conference (CCC), pp. 9095-9100. Elsevier, Amsterdam, (2018).
14. Garcia A. G., Escolano S. O., Oprea S. O., et al.: A Review on Deep Learning Techniques Applied to Semantic Segmentation. arXiv preprint arXiv:1704.06857, (2017).