Railroad Surface Defect Segmentation Using a Modified Fully Convolutional Network

Hyeonho Kim[1], Suchul Lee[2], and Seokmin Han[3]*
[1,2,3] Department of Computer Science & Information Engineering
Korea National University of Transportation
Uiwang-si, Gyeonggi-do, South Korea
[e-mail: rlagusgh125@ut.ac.kr, sclee@ut.ac.kr, seokmin.han@ut.ac.kr]
* Corresponding author: Seokmin Han

Received November 5, 2019; revised May 20, 2020; revised July 16, 2020; accepted November 17, 2020; published December 31, 2020

Abstract

This research aims to develop a deep learning-based method that automatically detects and segments the defects on railroad surfaces to reduce the cost of visual inspection of the railroad. We developed our segmentation model by modifying a fully convolutional network model [1], a well-known segmentation model used for machine learning, to detect and segment railroad surface defects. The data used in this research are images of the railroad surface with one or more defect regions. Railroad images were cropped to a suitable size, considering the long height and relatively narrow width of the images. They were also normalized based on the variance and mean of the data images. Using these images, the suggested model was trained to segment the defect regions. The proposed method showed promising results in the segmentation of defects. We consider that the proposed method can facilitate decision-making about railroad maintenance, and potentially be applied for other analyses.

Keywords: Artificial Intelligence, Pattern Recognition, Computer Vision System

http://doi.org/10.3837/tiis.2020.12.008 **ISSN : 1976-7277**

1. Introduction

Trains and subways are popular forms of public transportation. Therefore, unexpected accidents and delays are of serious concern for railways making good maintenance essential. This study aims at the semantic segmentation of railroad surfaces through deep learning, permitting visualization of the size and position of defects. Defects occur owing to a variety of reasons, such as friction or the collision of parts connecting adjacent tracks, but generally increase over time [2]. Defect growth directly or indirectly causes risk factors, such as broken rails [18]. Therefore, prevention actions such as replacement or repair of the railroad track may be needed after a railroad inspection. Automated defect segmentation can help investigators find rail defects. Our main idea is illustrated in **Fig. 1**.

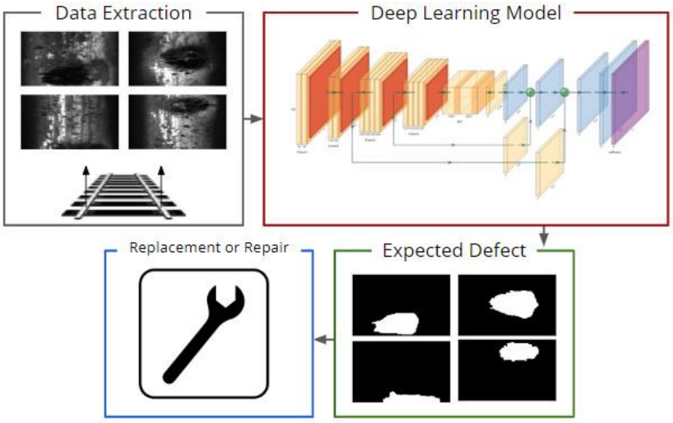


Fig. 1. Blueprint of actions such as replacement or repair, assisted by defect detection using a deep learning model

To aid in the inspection process, we designed a segmentation model by modifying a fully convolutional network (FCN) [1]. Our model adapts to the size of the input image of the railroad surface, reducing the computation cost. The proposed segmentation model was developed to balance performance and computation time. Our segmentation model is a scaled down version of an FCN based on VGG19 [12]. Because it is based on VGG19, we were able to use parameters that had learned ImageNet in advance. As a result, our segmentation model with 27 relatively low-depth layers was able to achieve F1 and intersection over union (IoU) scores as high as 90% with low computation time.

After a discussion of related work in Section 1.1, in Section 2, we describe the data, our network structure, and the proposed method. Section 3 presents the experiments we performed. In Section 4, we compare and analyze our experiments with related studies. In Section 5, we discuss the experiments conducted and conclude in Section 6.

1.1 Related work

As the need for automation of railroad inspection has increased, relevant studies have been conducted to support the automatic analysis of railroad surfaces using machine vision.

Concerning railroad defect detection, a lot of researches into machine vision has already been done [20-22]. In particular, He et al. [14] introduced the Perona–Malik diffusion model for a rail surface defect detection system. And Gan et al. [9] proposed a hierarchical inspection framework including coarse extractors and fine extractors to handle different railway elements.

And in the classification problem using neural networks, Giben et al. [3] proposed a deep convolution neural network for material classification and segmentation of rails. And Faghih-Roohi et al. [4] proposed a deep convolutional neural network (DCNN) for classification. Moreover, Soukup and Huber-Mörk [15] reinforced detection performance by training the photometric stereo images of the surface defects to a convolutional neural network (CNN). Besides, Ref [5, 23] used YOLOv3 [6] or MobileNetV2 [24] to detect the presence of defects and to the approximate location of the defects called 'bounding boxes'. However, in this case, the size of the defect is not visually accurate.

Fully convolutional network (FCN) [1] based methods have often been applied to the semantic segmentation of pixel-level classification of defects. To obtain better performance, researchers sometimes added a conditional random field to the FCN [16], or reconstructed a segmentation model such as SegNet [8] or U-net [17] inspired by the structure of the FCN. However, these models may be slower than the FCN-based method, because SegNet and U-Net contain more complex pairwise decoder architectures.

Among the segmentation methods, Liang et al. [7] has the closest relevance to our method. They made a SegNet with 59 layers to classify rail defects. In particular, Ref [7], used the same database as our research for the same purpose. Unfortunately, they did not provide a precise description of their segmentation experiment environment (e.g., GPU, parameters) or their image preprocessing method. Thus, we could not find a quantitive analysis of the method, making it difficult to directly compare performance. In this research, we provide a precise experimental description and quantitative evaluation results of the proposed method, along with an indirect comparison with Ref [7]. To the best of our knowledge, this research is the first to provide an exact experiment description and quantitative evaluation of a deep learning-based method for railroad defect detection and segmentation.

2. Methods

2.1 Model Architecture

2.1.1 Image Database Construction

The dataset used in this research is composed of rail-surface images with at least one defect [9]. The defect region is approximately 1% of the entire image. In the dataset, for each original surface image, the corresponding mask image is provided with some noise. The mask image is considered to be the ground truth. There are two types of railroad images in this study. Type 1 images are images of express rails; Type 2 images are of common or heavy haul rails.

2.1.2 Data Initialization

In order to make the railroad images suitable for use, each image was divided and cropped to obtain images of height 100 pixels, as illustrated in **Fig. 2**. From the express rail images (Type 1), we obtained 709 samples of size 100×160×1, and from the common/heavy rail images (Type 2), we obtained 1408 samples of size 100×55×1.

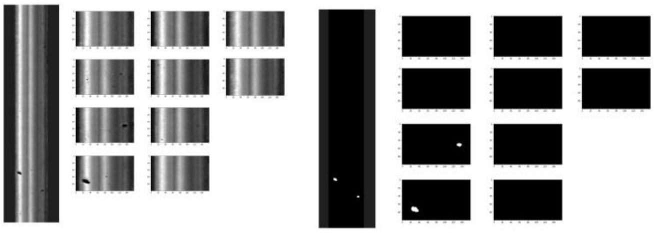


Fig. 2. Each image was divided and cropped to obtain images of height 100 pixels

In the ground truth images, noise was eliminated by simple thresholding (value 127). The ground truth images underwent the same process as the railroad images. After that, the dataset images were split into two mutually exclusive groups. 90% of the images went into the training set, and 10% into the test set.

2.1.3 Image Preprocessing

For preprocessing, we subtracted the mean value of the training set images from every image (training and test sets) and then divided the pixel values of each image by the standard deviation of the training set images. Thus, the distribution of the pixel values changed, as illustrated in **Fig. 3**.

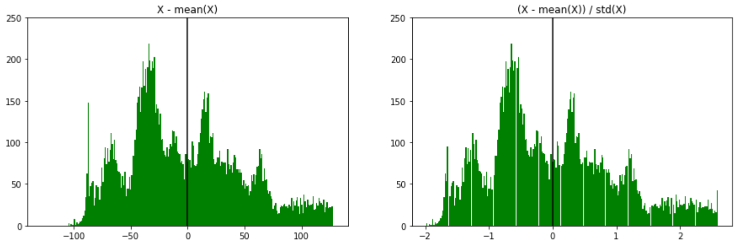


Fig. 3. Distribution of pixel values of the image before preprocessing (left) and the image after preprocessing (right)

As mentioned, the test set images were preprocessed in the same way, using the mean value and the standard deviation of training set images.

2.1.4 Fully Convolutional Network

Fully convolutional networks (FCN) [1] show excellent performance for segmentation and have been applied in various studies. Largely, the traditional FCN consists of:

1) Convolutionalization: In the supervised learning classifier, a convolutional neural network (CNN) maps the existing labels with output to a dense layer (one that is fully connected) at the end of the network. However, at the end of the network, FCN uses a convolution layer instead of a dense layer to output feature images. This is called convolutionization.

2) Transpose Convolution: In a typical CNN structure, the pooling layer subsamples the input features. Max pooling is often used for the pooling method. It takes the maximum value at each of the sub-regions of a certain size and reduces the size of input features. Then, the transpose convolution layer (sometimes called a deconvolution layer [10]) is used to upscale the reduced output features. This enables back-propagation.

3) Addition Layer: As the subsampling and upscaling steps done by the pooling and transpose convolution layer may exclude useful information, FCN models add a feature map of the prior pooling layer to the output to increase model reliability. Combining fine and coarse layers has been shown to contribute to performance improvements.

4) Dropout Layer: In deep learning, the dropout layer is a regularization layer [19]. The dropout layer randomly sets input units to 0 at a certain frequency rate, which helps prevent overfitting.

2.1.5 Modified FCN Based on VGG

AlexNet [11] or VGG [12] are often used as part of FCN architecture. In this research, we applied VGG and pre-training weights trained on ImageNet, and modified them for our purpose. Considering the difference between the Type 1 and Type 2 datasets employed here, we designed two FCN models for each data type. The details of both models are presented in **Table 1** and **Fig. 4**.

Table 1. Proposed FCN structure

Sequence of Layer	**Type 1**	**Type 2**
Convolution layer Convolution layer Max-Pooling layer	3×3×64 3×3×64 2×2, Output: I/2	3×3×64 3×3×64 2×2, Output: I/2
Convolution layer Convolution layer Max-Pooling layer	3×3×128 3×3×128 2×2, Output: I/4	3×3×128 3×3×128 2×2, Output: I/4
Convolution layer Convolution layer Convolution layer Convolution layer Max-Pooling layer	3×3×256 3×3×256 3×3×256 3×3×256 2×2, Output: I/8	3×3×256 3×3×256 3×3×256 3×3×256 2×2, Output: I/8
Convolution layer Convolution layer Convolution layer Convolution layer Max-Pooling layer	3×3×512 3×3×512 3×3×512 3×3×512 2×2, Output: I/16	3×3×512 3×3×512 3×3×512 3×3×512 2×2, Output: I/16
Convolution layer Dropout layer Convolution layer Dropout layer Convolution layer	1×1×1024 Rate: 0.5 1×1×1024 Rate: 0.5 1×1×2	1×1×1024 Rate: 0.5 1×1×1024 Rate: 0.5 1×1×2
Transpose-Convolution layer Addition layer	4×4×2, Output: I/8 With a 3rd pooling	4×4×2, Output: I/8 With a 3rd pooling
Transpose-Convolution layer Addition layer	3×2×2, Output: I/4 With a 2nd pooling	3×3×2, Output: I/4 With a 2nd pooling
Transpose-Convolution layer	4×4×2, Output: I/2	2×3×2, Output: I/2
Transpose-Convolution layer	4×4×2, Output: I	2×3×2, Output: I
Total number of parameters	**12,163,110**	**12,163,042**

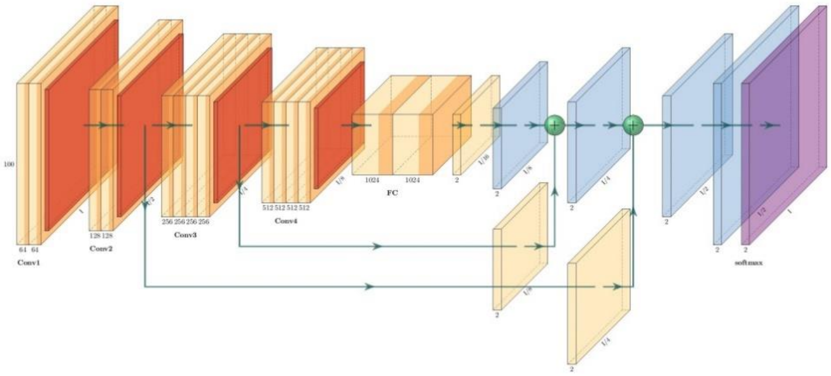


Fig. 4. Model architecture. Both modified models have reduced number of parameters

2.2 Training Methods

We used the Adam optimizer [13] because it is known to have stochastic characteristics and maintain a prior value with stability and fast convergence speed. Mean softmax-cross entropy (MSCE) was employed for the cost function. When the image data size is assumed to be $N \times W \times H$ and the number of classes is C, and the output of class i, the ith class of $N \times W \times H$ size, is assumed to be Y_i, we can define softmax as follows:

$$f(Y)_i = \frac{e^{Y_i}}{\sum_j^C e^{Y_j}} \tag{1}$$

where C is the number of classes and Y_i is the output of class i. We can also define MSCE as follows:

$$MSCE(T,Y) = \frac{1}{NWH} \sum^{NWH} (-\sum_i^C T_i log(f(Y)_i))_{nwh} \tag{2}$$

where N×W×H is the size of the input image, Ti is the label of the ith output, and we have 2 classes (C = 2). Equations (1) and (2) can be expressed as Equations (3) and (4) respectively:

$$f(Y)_i = \frac{e^{Y_i}}{e^{Y_0} + e^{Y_1}} \tag{3}$$

$$MSCE(T,Y) = \frac{1}{NWH} \sum^{NWH} (-T_0 log(f(Y)_0) - T_1 log(f(Y)_1))_{nwh} \tag{4}$$

because the same location values in class dimensions T_0 and T_1 are always either the 1 – or 0 state or the 0 – and 1 state, the loss function for a positive T_i can be expressed as follows:

$$L(T_i, Y \mid T_i = Positive) = \frac{1}{NWH} \sum^{NWH} (-log(\frac{e^{Y_i}}{e^{Y_0} + e^{Y_1}}))_{nwh} \tag{5}$$

3. Implementation

3.1 Accuracy

We used intersection over union (IoU) and the F1 score for evaluation. IoU is defined as $| T_g \cap T_p | / | T_g \cup T_p |$, where T_g is the set of “true” pixels, and T_p is the set of predicted pixels. IoU expresses the spatial similarity between the target label and the predicted result. The F1 score is defined as 2TP/(2TP + FP + FN), where FP is false positive, TP is true positive, FN is false negative, and TN is true negative. The general accuracy (which is the detection rate) is just (TP+TN)/(TP+FP+TN+FN). However, TN is not very meaningful in this research because the percentage of defects in this dataset is ~1%. Thus, TN is always high, even when the model does not predict defects well. Therefore, we treat F1 and IoU scores as more important than TN scores and calculate the F1 and IoU scores of the untrained test set for each epoch.

3.2 Experiment

We used a GeForce GTX 1080(8Gb). We performed experiments to analyze the performance and calculate the time cost. The batch size was set to 16 and the learning rate was set at 10^{-4}.

In this experiment, **Table 2** and **Fig. 5**, and **Fig. 6** show that the calculated loss decreases over time. We stopped the training of the network at epoch 300 because its training loss approached 0.001. We used this model for the test set.

Table 2. Training experiment

Model	Epoch	Iteration	Time	Training Set Loss
Type 1	1 10 100 300	40 400 4000 12000	4.5 s 46 s 470 s 1400 s	0.77 0.28 0.002 0.001
Type 2	1 10 100 300	80 800 8000 24000	5 s 50 s 500 s 1500 s	0.79 0.16 0.003 0.001

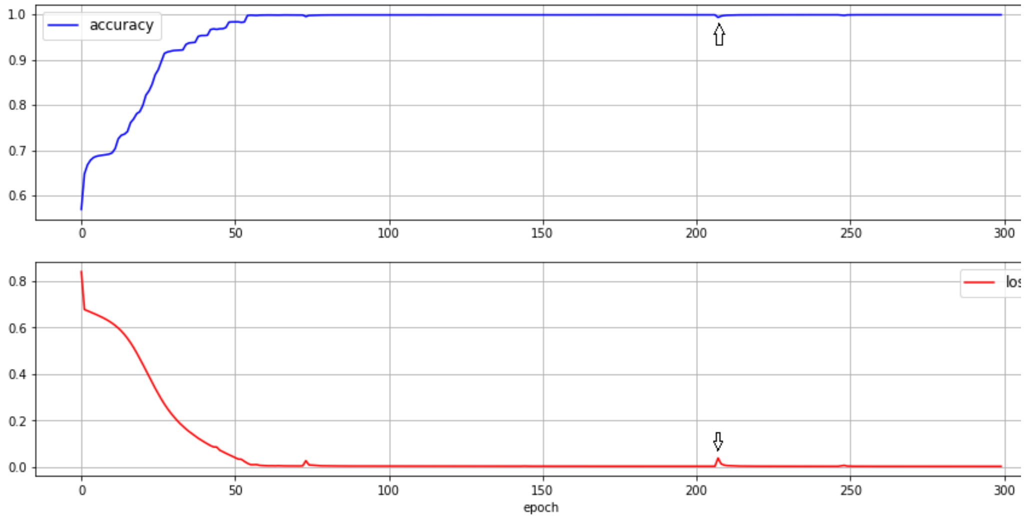


Fig. 5. Model, training accuracy and loss at each epoch for Type 1. At epoch 300, training loss approached 0.001

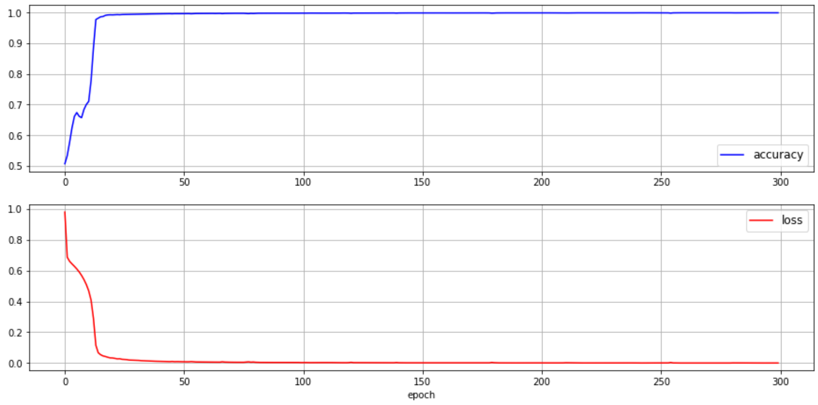


Fig. 6. Model, training accuracy, and loss for Type 2. At epoch 300, training loss approached 0.001

4. Analysis and Results

4.1 Experimental Results

The test results of the proposed network were obtained at epoch 300, batch size 16, and learning rate 0.0001. **Table 3** and **Fig. 7** show test results very close to the ground truth images.

Table 3. Test set score for trained model with learning rate 0.0001, batch size 16, and epoch 300. Type 1 and 2 test sets consisted of 71 and 141 images, respectively

Model	Accuracy	F1 score	Precision	Recall	Mean IoU	Entire Time
Type 1	0.999	0.972	0.972	0.973	0.947	9 s
Type 2	0.999	0.946	0.932	0.961	0.898	9 s

Compared with Type 1, Type 2 images of common and heavy rails have a more consistent background but more complex defects with more diverse appearances. The authors of Ref [9] reported performance differences due to this; it also occurred in our research.

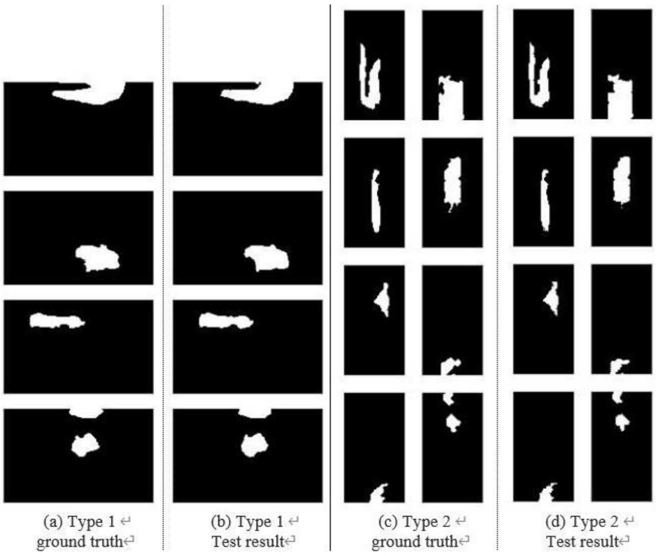


Fig. 7. Result images of Type 1 and Type 2 test sets

4.2 Comparisons and Validation

We compared the previous segmentation model [7] with the proposed model for validation. In addition, we also implemented the FCN-8s model [1] and compared it with the proposed model. FCN-8s is based on the VGG16 architecture. The previous segmentation model consists of 59 layers based on SegNet [8], as discussed in Ref [7]. However, our implementation may not match the original architecture of Ref [7]. Their model used 120 samples of size 1250×55×1 for training and nine samples for testing. Unfortunately, the authors of the segmentation model [7] did not report how to choose the test images from the dataset. Therefore, we randomly selected 120 samples for training and nine samples for testing. The experimental results of our implementation of Ref [7] are shown in **Table 4**.

Table 4. Segmentation model [7] training result

Epochs	Iteration	Time (per step)	Training Acc	Training Loss
100	3200	700 s (60 ms)	0.997	0.036
300	9600	2100 s (60 ms)	0.999	0.002

We also localized our dataset structure for the segmentation model [7] for a fair comparison. For this, we use k-fold cross-validation, which is often used for model evaluation. We used fivefold cross-validation for objective evaluation. **Table 5** and **6** show the F1 score IoU results of five subsets of three models (Base SegNet [7], FCN-8s [1], and our model, Proposed FCN). **Table 7** and **Fig. 8** show the overall comparison results of the three models.

Table 5. Fivefold cross-validation result of F1 score

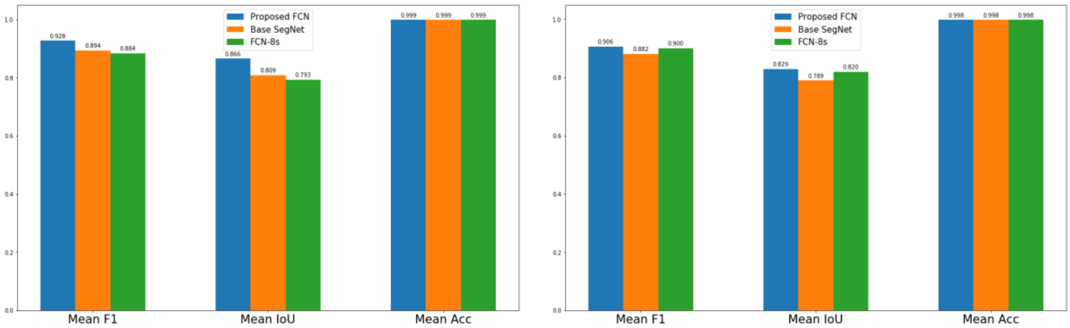
Type	Model	F1 score				
		Subset 1	Subset 2	Subset 3	Subset 4	Subset 5
Type 1	Proposed FCN	0.899	0.954	0.932	0.874	0.942
	Base SegNet [7]	0.892	0.897	0.926	0.902	0.851
	FCN-8s [1]	0.913	0.882	0.872	0.861	0.892
Type 2	Proposed FCN	0.874	0.929	0.911	0.906	0.956
	Base SegNet[7]	0.912	0.888	0.912	0.840	0.854
	FCN-8s [1]	0.847	0.902	0.932	0.896	0.921

Table 6. Fivefold cross-validation result of mean IoU

Type	Model	Mean IoU				
		Subset 1	Subset 2	Subset 3	Subset 4	Subset 5
Type 1	Proposed FCN	0.817	0.912	0.872	0.777	0.891
	Base SegNet[7]	0.805	0.814	0.862	0.822	0.741
	FCN-8s [1]	0.839	0.789	0.773	0.755	0.815
Type 2	Proposed FCN	0.777	0.896	0.847	0.829	0.917
	Base SegNet [7]	0.838	0.799	0.838	0.724	0.745
	FCN-8s [1]	0.736	0.822	0.873	0.811	0.853

Table 7. Fivefold cross-validation average comparisons

Type	Model	Mean Time Per step	Mean Validation Loss	Mean Validation Acc	Mean F1 score	Mean IoU
Type 1	Proposed FCN	7 ms	0.003	0.999	0.920	0.854
	Base SegNet [7]	13 ms	0.007	0.998	0.893	0.809
	FCN-8s [1]	8 ms	0.004	0.998	0.884	0.792
Type 2	Proposed FCN	4 ms	0.009	0.998	0.916	0.846
	Base SegNet [7]	6 ms	0.017	0.997	0.881	0.789
	FCN-8s [1]	5 ms	0.010	0.998	0.900	0.819



(a) Type 1 (b) Type 2

Fig. 8. Comparison plots

5. Discussion

Defect segmentation should be fast and accurate. We have suggested a relatively simple network and algorithm with promising results. It is observed that the proposed method can be applied to effective railroad defect detection. In our experiment, small batch size or low learning rate cases showed a tendency to converge faster in loss value than large batch size or high learning rate cases, with a widespread zigzag shape. Although we could not determine the exact reason for the widespread zigzag shape of the loss, it seems that batch size and learning rate should be considered as the tradeoff between time cost and performance in this research. Considering all the results with parameter variations in this research, we assigned a batch size of 16 and a learning rate of 10^{-4} as appropriate values.

Regarding the difference between data types 1 and 2, the difference in shape between these two data sets seems to cause a performance difference. This indicates that although we tried to normalize the railroad image data, there are still features that cause the difference in performance. To compensate for the difference in the shape and size of the railroad image data, we have to find an improved method than simple image resizing and normalization using the standard deviation and average. We are investigating such a method as the next step of this research.

Because we used the same dataset as Ref [7] and have the same purpose, we compared the results of Ref [7] with ours. To do this, we implemented their method according to Ref [7]. Unfortunately, the authors of Ref [7] did not report how to reproduce their method and evaluate their performance precisely. Although we tried to reproduce their method, we are not sure that we reproduced their method accurately. Despite this, we compared our results to previous methods in terms of F1 and IoU. Applying our dataset to train our implementation of Ref [7] as well as the proposed method, we showed that the performance of the proposed method is better than or equal to the previous method. To the best of our knowledge, this research is the first to provide the exact experimental description and quantitative evaluation of a deep learning-based method for railroad defect detection and segmentation, which is the main contribution of this research. Our future work will be to apply this method to real railroad situations, and measure the time-dependent growth of railroad defects.

6. Conclusion

The purpose of this study was to achieve rapid and accurate defect segmentation. We suggested a relatively fast-learning deep neural network model to optimize the balance between performance and calculation time cost, by building a neural network based on FCN. Because of its relatively simple and effective structure, our model's calculation cost was low, considering its promising performance. The IoU and F1 scores exceeded 90%. We hope that this research will be easily applied to industry, and we are planning to use this research in real railroad situations.

Acknowledgements

This research was supported by a National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (NRF-2017R1C1B5077068) and by Korea National University of Transportation in 2020. And, we would like to thank Editage (www.editage.co.kr) for English language editing.

References

[1] E. Shelhamer, J. Long, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 640-651, Apr. 2017. Article (CrossRef Link)

[2] R. Andersson, "Surface defects in rails: potential influence of operational parameters on squat initiation," *Thesis for the degree of licentiate of engineering/Department of Applied Mechanics, Chalmers University of Technology*, 2015. Article (CrossRef Link)

[3] X. Giben, V. M. Patel, and R. Chellappa, "Material classification and semantic segmentation of railway track images with deep convolutional neural networks," in *Proc. of 2015 IEEE International Conference on Image Processing (ICIP)*, Sep. 2015. Article (CrossRef Link)

[4] S. Faghih-Roohi, S. Hajizadeh, A. Núñez, R. Babuska, and B. De Schutter, "Deep convolutional neural networks for detection of rail surface defects," in *Proc. of 2016 International Joint Conference on Neural Networks (IJCNN)*, pp. 2584-2589, 2016. Article (CrossRef Link).

[5] S.Yanan, Z. Hui, L. Li, and Z. Hang, "Rail surface defect detection method based on YOLOv3 deep learning networks," *2018 Chinese Automati on Congress (CAC)*, pp. 1563-15668, Dec. 2018. Article (CrossRef Link)

[6] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018. Article (CrossRef Link)

[7] Z. Liang, H. Zhang, L. Liu, Z. He, and K. Zheng, "Defect detection of rail surface with deep convolutional neural networks," in *Proc. of 2018 13th World Congress on Intelligent Control and Automation (WCICA)*, July 2018. Article (CrossRef Link)

[8] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481-2495, 2017. Article (CrossRef Link)

[9] J. Gan, Q. Li, J. Wang, and H. Yu, "A Hierarchical Extractor-Based Visual Rail Surface Inspection System," *IEEE Sensors Journal*, vol. 17, no. 23, pp. 7935-7944, 1 Dec. 1, 2017. Article (CrossRef Link)

[10] M. D. Zeiler, D.Krishnan, G. W. Taylor, and R. Fergus, "Deconvolutional networks," in *Proc. of 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2582-2535, June 2010. Article (CrossRef Link)

[11] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks ," *NIPS*, vol. 60, no. 6, 2012. Article (CrossRef Link)

[12] K. Simonyan and A. Zisserman. "Very deep convolutional networks for large-scale image recognition," *ICLR*, 2015. Article (CrossRef Link)

[13] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. of Published as a conference paper at the 3rd International Conference for Learning Representations*, San Diego, 2015. Article (CrossRef Link)

[14] Z. He, U. Wang, F. Yin, and J. Liu , "Surface defect detection for high-speed rails using an inverse PM diffusion model," *Sensor Review, 2016*. Article (CrossRef Link)

[15] D. Soukup and R. Huber-Mörk, "Convolutional Neural Networks for Steel Surface Defect Detection from Photometric Stereo Images," *International Symposium on Visual Computing*, vol. 8887, pp. 668-677, 2014. Article (CrossRef Link)

[16] W. Zhao, Y. Fu, X. Wei, and H. Wang, "An improved image semantic segmentation method based on superpixels and conditional random fields," *2018 Applied Sciences*, 2018. Article (CrossRef Link)

[17] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Proc. of 2015 International Conference on Medical image computing and computer-assisted intervention*, vol 9351, 2015. Article (CrossRef Link)

[18] H. Darwin and Schafer, "Effect of train length on railroad accidents and a quantitative analysis of factors affecting broken rails," B.S. thesis, Dept. Civil Engineering, Univ. of Illinois at Urbana-Champaign, Urbana, IL, USA, 2008.

[19] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *2014 Journal of Machine Learning Research*, vol. 15, no. 1, 2014. Article (CrossRef Link)

[20] H. Zhang, X. Jin, Q. M. J. Wu, Y. Wang, Z. He, and Y. Yang, "Automatic visual detection system of railway surface defects with curvature filter and improved gaussian mixture model," *IEEE Transactions on Instrumentation and Measurement*, vol. 67, no. 7, pp. 1593-1608, July 2018. Article (CrossRef Link)

[21] M. Yongzhi, B. Xiao, J. Dang, B. Yue, and T. Cheng, "Real time detection system for rail surface defects based on machine vision," *EURASIP Journal on Image and Video Processing*, 2018. Article (CrossRef Link)

[22] Y. Wu, Y. Qin, Z. Wang, and L. Jia, "A UAV-based visual inspection method for rail surface defects," *2018 Applied sciences*, vol. 8, no. 7, 2018. Article (CrossRef Link)

[23] H. Yuan, H. Chen, S. Liu, J. Lin, and X. Luo, "A Deep Convolutional Neural Network for Detection of Rail Surface Defect," in *Proc. of 2019 IEEE Vehicle Power and Propulsion Conference (VPPC)*, pp. 1-4, 2019. Article (CrossRef Link)

[24] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proc. of the IEEE conference on computer vision and pattern recognition*, vol. 1, 2018. Article (CrossRef Link)

Hyeonho Kim received B.Sc. degree from Korea National University of Transportation in 2019 and currently proceeds an M.Sc degree there. His research interests are machine learning, digital image processing, and deep learning.

Suchul Lee is an associate professor in the Dept. of Computer Science and Information Engineering at Korea National University of Transportation, Korea. He received the B.S. and Ph.D. degrees in Computer Science and Engineering from Seoul National University, Korea in 2008 and 2014, respectively. From 2014 to 2016, he was a member of research staff in National Security Research Institute, Korea. His research interests include computer and information security, wireless and mobile systems, Internet applications, such as 802.11, cognitive radio, and traffic analysis with an emphasis on system performance optimization.

Seokmin Han received B.Sc., M.Sc., and Ph.D. degrees from Seoul National University in 2000, 2003, and 2008, respectively. He is currently an associate professor at Korea National University of Transportation. His research interests are machine learning, computer vision, and medical image analysis.