ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)
https://doi.org/10.17559/TV-20221020152721
Original scientific paper

A Deep Learning-Based Hybrid Approach to Detect Fastener Defects in Real-Time

İlhan AYDIN*, Mehmet SEVİ, Erhan AKIN, Emre GÜÇLÜ, Mehmet KARAKÖSE, Hssen ALDARWICH

Abstract: A fastener is an important component used to fix the rail in railways. Defects in this component cause the rail and ballast to remain unstable. If the defective fasteners are not replaced in time, it is inevitable that the train will derail, and serious accidents will occur. Therefore, this component should be inspected periodically. Conventional image processing-based control systems are affected by noise and different lighting conditions in the real environment. In this study, it is aimed to determine the defects of fasteners with a deep learning-based hybrid approach. The YOLOv4-Tiny method is used for fastener detection and localization. This method gives accurate results, especially for the detection of small objects. After the fastener position is determined, a new lightweight convolutional neural network model is used for defect classification. The proposed convolutional neural network for classification has a small network structure because it uses depth-wise and pointwise convolution layers. When the experimental results are compared with other known transfer learning methods, better results were obtained in terms of training/test time and accuracy.

Keywords: defect detection; deep learning; fastener; object detection; railway system

1 INTRODUCTION

In railways, the fastener is a very important component as it provides the fixation between the rail and the sleeper. Due to the friction between the train wheel and the rail, defects occur or come off in the fastener [1]. The defective fastener causes the vertical displacement of the rail and vibrations during the train transition [2]. Defective fasteners are very important for the safety of the rail and should be examined periodically. Conventional rail control systems use expert personnel to visually control the rail along the railway. Manual rail control produces false alarms because it is time-consuming, requires expert knowledge, and checks are carried out at night when the line is empty [3]. With the increase of high-speed rail lines, traditional control systems have become unusable for existing systems in terms of speed and accuracy.

In recent years, with the developments in computer vision and artificial intelligence, non-contact defect detection methods have started to be developed in railway components. Non-contact defect detection methods can be examined in two parts: computer vision [4-6] and deep learning-based methods [7-12]. Although computer vision-based techniques give good results under certain light conditions, manual feature extraction and low generalization performance are the disadvantages of this method. Deep learning-based techniques, on the other hand, are trained with large datasets and automatically extract features from images. Thanks to these features, their generalization performance is better.

Defects in railway sub-assemblies can be classified into 3 categories as rail surface defects, fastener defects, and traverse-related defects. Fastener defects are one of the most studied defect types. Feng et al. [2] used the position of the rail and sleeper to determine the position of the fastener. For this purpose, they used line segmentation and the geometric relationship between the two components. Wei et al. [11] proposed an image processing-based technique to detect rail and fastener and rail position. Aydin et al. [13] applied the LLBP method after image pre-processing to detect rail and traverse for fastener detection. Dou et al. [14] proposed a template matching-based method to determine the position of the fastener. Liu et al. [15] localized the fastener position with HOG features and a template-matching method. Franca et al. [16] presented a method based on basic image processing, feature merging, and a heuristic algorithm. The proposed method uses edge detection and entropy methods as well as Haar transform. Fan et al. [17] proposed the Line Local Binary Pattern (LLBP) method for the detection of the fastener. The fastener position is determined in the image containing rail and more than one traverse. The disadvantage of the method is that it needs to adjust the neighbourhood parameter. Feng et al. [2] presented an STM-based approach to model fasteners. The proposed method uses a probability-based approach to determine the object model. Wei et al. [11] used Dense SIFT and bags of visual word methods to classify the fastener condition.

In recent years, with the developments in the field of deep learning, new methods have been developed for the detection of rail components and defect detection. Liu et al. [18] proposed fastener detection with an object detection-based deep learning approach and defect detection with decision tree-based regional analysis. Their proposed defect classification approach identifies defects relative to the healthy fastener. Qi et al. [19] proposed a modified Tiny-YOLOv3 architecture for fastener detection in real-time. The proposed approach only detects fasteners, and no analysis has been made for defect detection. Tu et al. [3] proposed a deep learning-based approach in order to determine rail surface and fastener defects in real-time. Their method uses a YOLACT-based segmentation network for the detection of rail and fasteners. After the fasteners are segmented, the right and left fasteners are compared with template matching in a binary image. In the proposed method, the fastener must be labelled in a detailed form. With a modified SSD-based method, the fasteners were detected, the cropped fasteners were separated into regions, and defect detection was made with the Faster-RCNN [20]. The generative adversarial network-based architectures have also been recommended to increase the number of defective data in cases where the defective data related to fasteners are small [9].

In most of the methods proposed for the detection of the fastener in the literature, real-time working conditions are considered in limited studies. In particular, deep learning-based approaches include complex models. The most of approaches for the detection of fasteners have low accuracy. It is also difficult to measure real achievements because data sets are imbalanced. In this study, a Tiny-YOLOv4-based fastener is proposed to detect small objects. After the fasteners are detected, the data set is formed by cropping. The types of defects are classified by using an improved lightweight mobile network. The contributions of the proposed approach are given below.

- Detection and defect-recognition of the fasteners with lightweight integrated structure,
- Being able to work in real-time,
- Detection of defects with high accuracy.

The rest of this study is organized as follows. Section 2 explains the proposed hybrid approach. Experimental results are compared in Section 3. The results are given in Section 4.

2 THE PROPOSED HYBRID APPROACH FOR FASTENER DETECTION AND DEFECT CLASSIFICATION

The proposed hybrid approach performs fastener defect detection in two stages. The first stage is the determination of the position of the fastener and the second stage is the fastener defect detection. The block diagram of the proposed approach is given in Fig. 1.

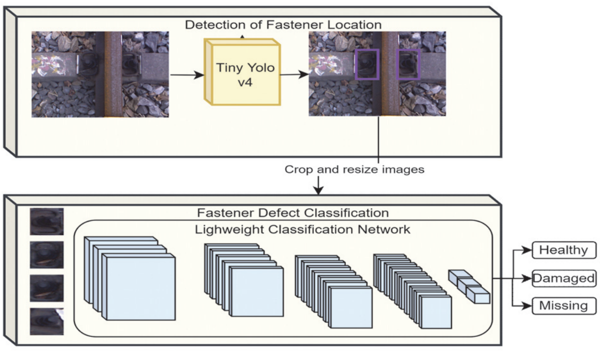


Figure 1 The proposed hybrid framework for defect detection of fastener

In the first step of the method proposed in Fig. 1, the positions of the fasteners are localized. The detection of the fastener is difficult due to the complex background of components such as rail, sleeper, and ballast in the railway image. Therefore, YOLOv4-Tiny will be used for real-time and high-accuracy detection of the fastener [21]. A lightweight convolutional neural network is proposed for the fastener state recognition. Thanks to the Inverted residual blocks and the attention module, this model is both smaller in size and can perform faster recognition.

2.1 Detection of Fastener Region

In this part, the YOLOv4-Tiny model is used to determine the position of the fastener. The fastener positioning method should yield good results in both speed and accuracy in the presence of complex backgrounds. YOLOv4-Tiny is a model designed on the YOLOv4 model and works faster for object detection. Thanks to this feature, it can be used in real-time, especially in embedded systems and mobile devices.

YOLOv4-Tiny has modified the CSPDarknet53 module as a backbone. This module uses CSPBlock instead of ReSBlock in the Residual network. In order to reduce the computational cost of CSPBlock, this cost is reduced by not using Bottleneck blocks and the accuracy and learning ability of the network are increased. Another difference between YOLOv4-Tiny compared to standard YOLOv4 is that it uses the Leaky Relu activation function instead of the Mish activation function in CSPDarknet53. Two activation functions are given in Fig. 2.

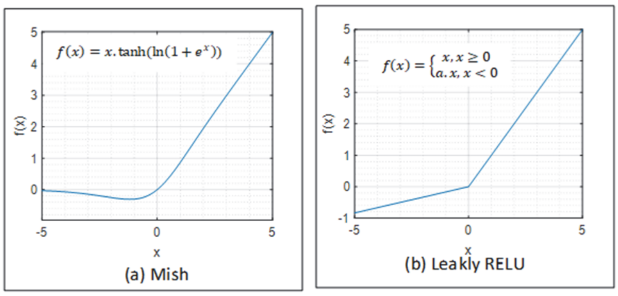


Figure 2 Activation function used in YOLOv4 versus YOLOv4-Tiny

The architecture shown in Fig. 3 uses a pyramid network structure to obtain feature maps at different scales.

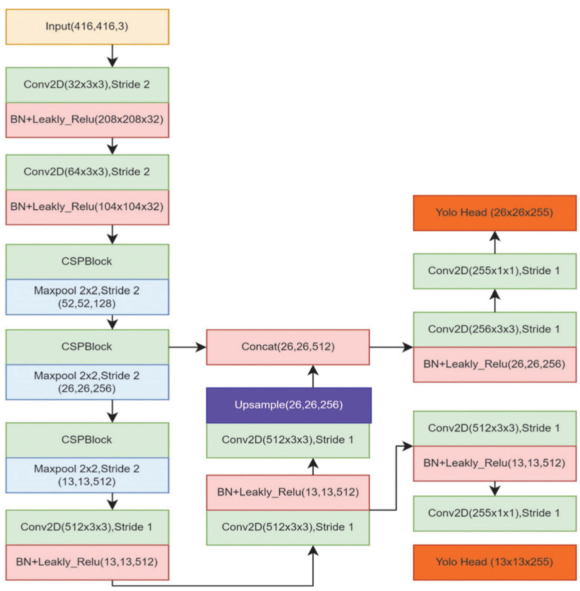


Figure 3 YOLOv4-Tiny architecture

The structure in Fig. 3 increases the detection speed and is different from the spatial pyramid used in YOLOv4. It also uses feature maps at 2 different scales, 26 × 26 and 13 × 13, for prediction. The CSP block used in the YOLOv4-Tiny model is given in Fig. 4.

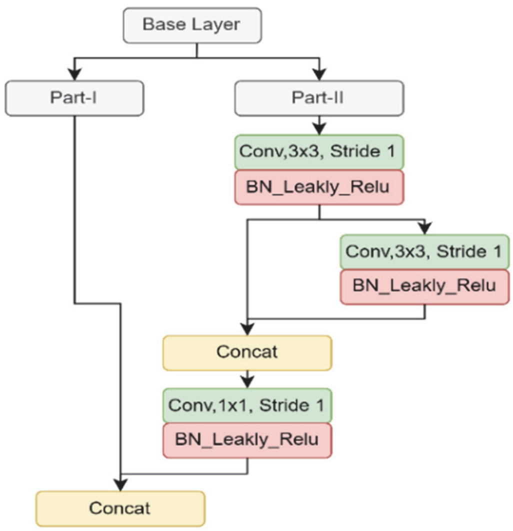


Figure 4 CSP block used in YOLOv4-Tiny

The CSP block in Fig. 4 splits the base layer into two parts. After certain convolution processing is applied to the second part, it is concatenated with the first part. There are residual blocks at the base of the structure. These blocks increase the speed of the algorithm.

In this study, the YOLOv4-Tiny model is used for the position detection of rail fasteners. An interface has been prepared for this purpose. The constructed interface is given in Fig. 5.

Figure 5 GUI design for testing the YOLOv4-Tiny model

In Fig. 5, the trained model is selected, and the fasteners are detected in the images of the selected folder. Afterward, the detected fasteners are cropped and written to a result folder.

2.2 Fastener State Recognition

After detecting the fastener with the YOLOv4-Tiny model, a lightweight network model is proposed for the detection of the defect type in the fastener. A low-weight network model using inverted residual blocks is proposed to create a network with high classification accuracy that will operate in real-time. The proposed network model is given in Fig. 6.

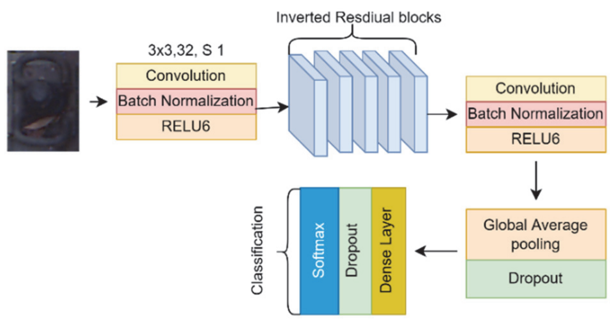


Figure 6 The proposed lightweight CNN model

The proposed method in Fig. 6 consists of a lightweight network model. The proposed model has inverted residual blocks using depth-wise convolution layers. Depth-wise convolution layers were first introduced in the Xception model and are used in lightweight networks to reduce the number of parameters [22]. The depth-wise separable convolution operation performs a 3 × 3 one-channel convolution operation for each channel, followed by a 1 × 1 M-channel pointwise convolution operation.

Fig. 7 shows the structure.

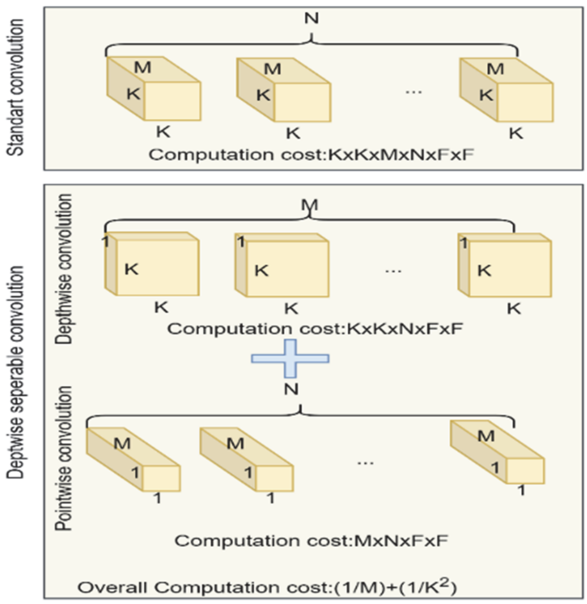


Figure 7 Standard and depth-wise separable convolutions

In Fig. 7, when using depth-wise separable convolution on an $F \times F$ size image, the computational cost is 10 times smaller than using a standard convolution layer. In many convolutional neural networks, the power of the network is increased by applying RELU activation after each convolution process. Although this operation works well in complex networks, it causes data loss in lower-sized networks. For this reason, the linear bottleneck operation, which provides the transfer of data directly to the other layer after the convolution layer, allows using the missed features with RELU. Inverted residual blocks were used after the MobileNetv2 network [23]. These blocks consist of three convolution operations. These layers consist of the bottleneck block, which increases the feature size, the depth-wise convolution layer, which provides low-weight feature extraction, and the pointwise convolution layer. This structure is shown in Fig. 8.

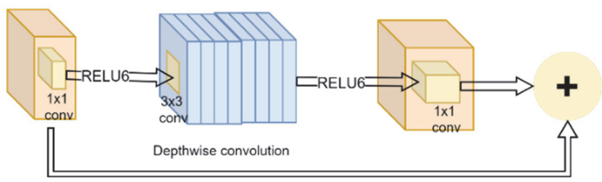


Figure 8 Inverted residual block

The model given in Fig. 8 provides higher accuracy results than traditional residual blocks. Many features are lost because the traditional residual blocks extract features after dimension reduction. In contrast, inverted residual blocks increase the dimension before feature extraction and have a low number of parameters due to the depth-wise convolution layers. In addition, a dense layer is included in the model after the global pooling layer. The structure and parameters of the proposed model are given in Tab. 1.

The t value in Tab. 1 is the channel expansion factor applied to the input tensor. For example, if the t value is taken as three and the input tensor consists of 16 channels, the output is 3 × 16 = 48. The value of n indicates how many times the blocks will be repeated. The S value is the stride parameter in the convolution operation.

Table 1 The structure and parameters of the proposed CNN

Number of filters	Layer	t	n	s
32	Convolution2d 3 × 3	-	1	1
32	Inverted residual block	1	1	1
8	Inverted residual block	6	1	2
16	Inverted residual block	6	2	1
32	Inverted residual block	6	3	2
64	Inverted residual block	6	2	1
256	Global average pooling	-	-	-
512	Dense	-	-	-

2.3 Performance Evaluation Metrics

In this study, both object detection and defect recognition are done with a YOLOv4-Tiny and a lightweight convolutional neural network; operations are performed on two different metrics. One of the most important metrics for object detection is Intersection over Union (IoU), which is represented as the ratio of the intersections of the labelled bounding box and the predicted bounding box to the union. This structure is given in Fig. 9.

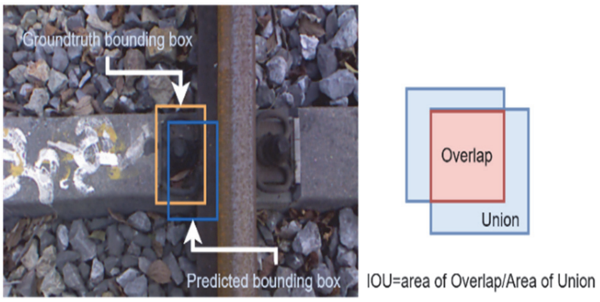


Figure 9 The calculation of IOU

In object detection, precision, recall, *mAP*, and *F*1 scores obtained from the precision-recall curve are needed, apart from the IOU. Precision and recall are two important parameters in object detection, and the curve created with these two parameters is used to calculate other metrics. Precision is the ability of a model to predict only objects of interest. Recall is the ability of a model to find all states. The equations for the two metrics are given below.

$$P = \frac{TP}{TP + FP} \tag{1}$$

$$R = \frac{TP}{TP + FN} \tag{2}$$

$$F_1 = 2 * \frac{PxR}{P + R} \tag{3}$$

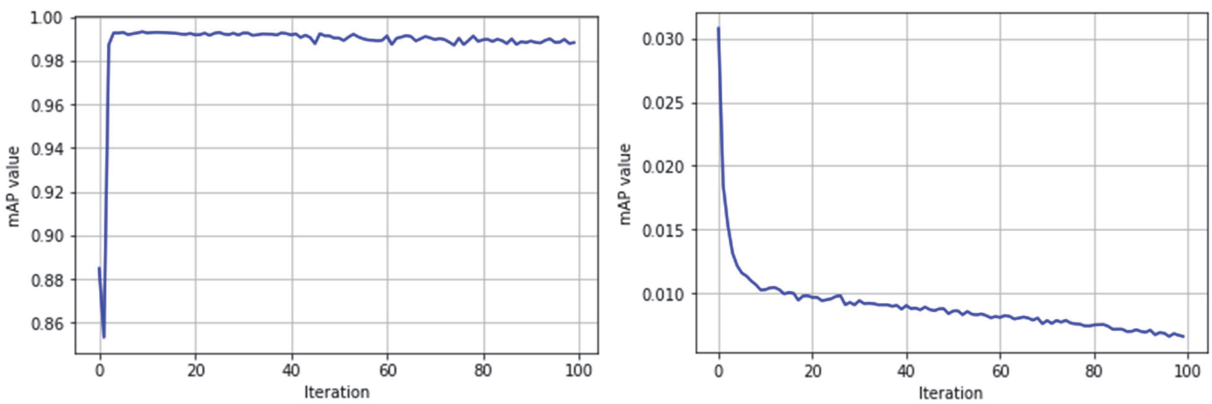
In Eqs. (1) to (3), TP represents the true positive value and means that the trained model predicts the correct positive. The *FP* value is false positive, and the trained model falsely predicts the positive class. The *FN* value is the prediction of a sample as negative when its real value is positive. The accuracy of object detection methods is usually measured by Mean Average Precision (*mAP*) for each class. The mean precision is the area under the *P-R* curve. These two parameters are given in Eq. (4) and (5). The accuracy rate in the lightweight classification network is also used for fastener defect classification.

$$AP = \int_0^1 p(r)\,\mathrm{d}r \tag{4}$$

$$mAP = \frac{1}{N} \sum AP_i \tag{5}$$

3 EXPERIMENTAL RESULTS

With the proposed deep learning-based approach, a data set with 8461 fastener images were used to detect fasteners and identify defects [24]. The dataset was taken with a camera placed on a railway vehicle. Detection and recognition networks are trained on a Mobile Workstation computer with a GTX2070 Max Q GPU card with 8 GB memory. A pre-trained model of the YOLOv4-Tiny model with transfer learning on ImageNet was used. In model training, a batch size of 64 was taken and 100 epochs of training were carried out. The learning rate was taken as $1e^{-4}$ and SGD was chosen as the optimizer.



(a) *mAP* value

(b) Training loss

Figure 10 Training loss and *mAP* graph of YOLOv4-Tiny

Fig. 10 shows the graph of *mAP* and loss in YOLOv4-Tiny training. According to the loss and *mAP* graph obtained in Fig. 10, the detection rate of the fasteners is quite high. In addition, the precision-recall curve is another metric that can be used for performance measurement. The closer this curve is to 1, the higher the performance of the model. The Precision-Recall curve for the YOLOv4-Tiny model is given in Fig. 11. According to the loss and *mAP* graph obtained in Fig. 10, the detection rate of the fasteners is quite high. In addition, the precision-recall curve is another metric that can be used for performance measurement. The closer this curve is to 1, the higher the performance of the model. The Precision-Recall curve for the YOLOv4-Tiny model is given in Fig. 11.

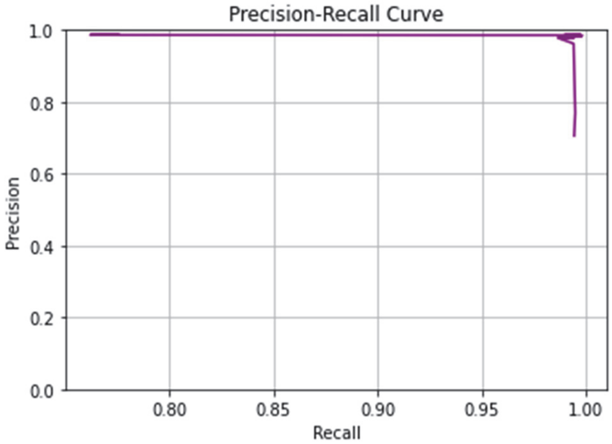
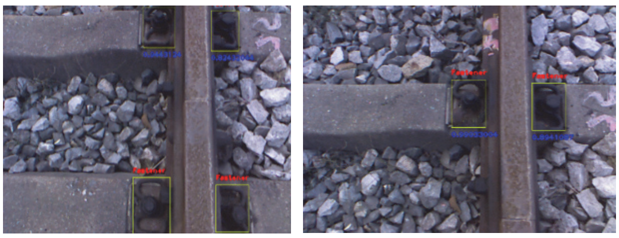


Figure 11 The precision-recall curve for YOLOv4-Tiny

In Fig. 12, defective and missing fasteners are also given in order to test the robustness of the YOLOv4-Tiny model. In cases where the fastener is missed or partially broken, the algorithm performs the detection process. After the fasteners are detected, the CNN model is run for defect detection. For the fastener, healthy, defective, and missed fasteners are taken into consideration. The number of samples in the data set for each case is given in Tab. 2.



(a) Four healthy fasteners (b) One healthy and one defected fastener

Figure 12 Fastener detection results for different conditions

Table 2 The number of samples for each class

Class name	Number of samples
Healthy	5000
Defect	1000
Missing	1000

In Tab. 2, the number of defective samples is less than the number of healthy samples. It consists of defective fasteners reverse broken fasteners, broken bottom fasteners, and broken fasteners on both sides. In particular, the data set was chosen in this way in order to take into account the class imbalance in performance metrics. The training parameters of the proposed deep neural network are given in Tab. 3.

Table 3 The parameters of Lightweight CNN

Parameter	Value
Learning rate	$1e^{-3}$
Optimizer	Adam
Batch size	32
Epoch	70

The deep learning network was run according to the parameters in Tab. 3. Accordingly, classification performance and loss function for training and validation are given in Fig. 13.

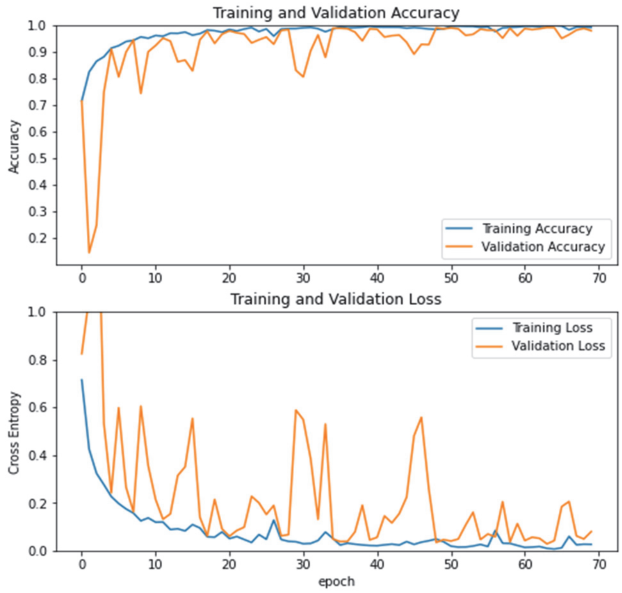


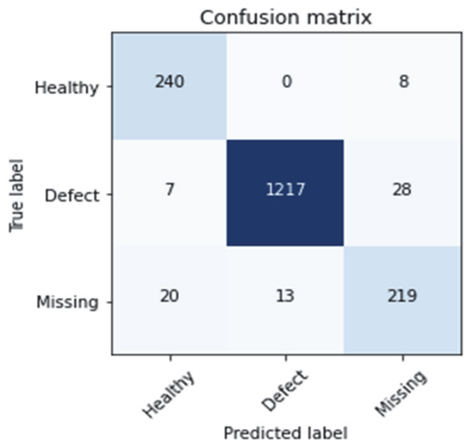
Figure 13 Accuracy and loss graph for training and validation dataset

In the accuracy and loss values given in Fig. 13, it is seen that the accuracy is fixed in both the training and validation data sets after a certain iteration. The performance of the proposed CNN-based defect classification network is also compared with the transfer learning methods known in the literature. For this purpose, the proposed method was compared with methods such as MobilenetV2 [23], InceptionV3 [25], and VGG16 [16]. The complexity matrices for each method are given in Fig. 14.

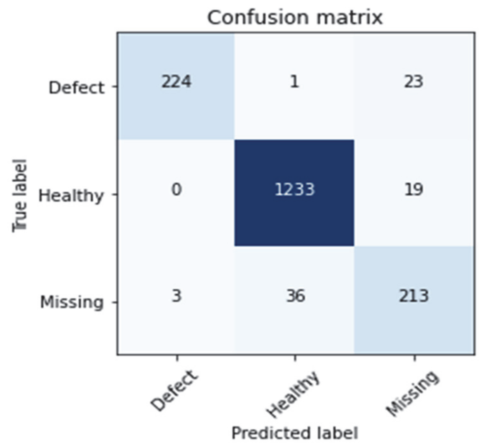
Performance metrics were compared for classification on the confusion matrices obtained in Fig. 14. For this purpose, accuracy, precision, recall, and $F1$ criteria of confusion matrices were compared. In addition, the number of parameters and processing times were evaluated for each method. These comparison results are given in Tab. 4.

In the comparison results given in Tab. 5, the proposed approach in [3] first segments the fasteners with the YOLACT-based segmentation method, and then the defective fasteners are found by pattern matching. The method requires detailed labelling and gives false results if the elements on the right and left of the rail are defective. In the study presented in [11], Faster RCNN method was used for rail fastener detection and both defect classification and detection were performed. In the study in [13], image enhancement was applied to make the position determination of the fastener more accurate. However, the size of the dataset used is small and balanced. The approach proposed in [15] proposes a two-stage convolutional neural network. In the last stage, both classification and similarity comparisons are done in the separated network structure. In [21], after the segmentation process is performed in order to increase the number of defective data, the defect is created by obtaining the fastener element and placed on a determined background.

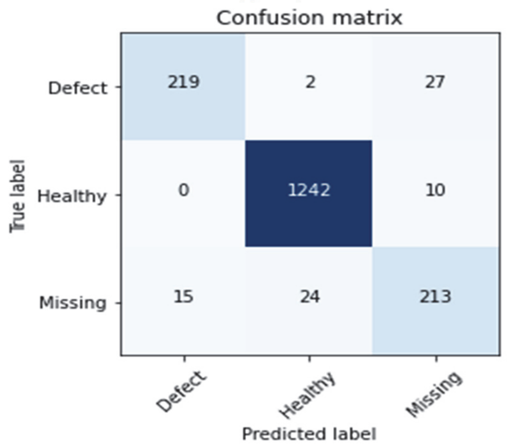
The main problem here is that the background of the fastener is fixed, the images used are similar and the network memorizes. In [22], RGBD images are obtained for classification, and taking these images brings an extra burden. In [27], an in-painting-based approach was proposed to produce defective fasteners, and the VGG19 model was used for classification. However, the images produced with defects mostly consist of broken fasteners and do not reflect the real condition. In the real world, bent and deformed fasteners are also formed. In [28], a low-weight Nano-Yolox model is proposed for the detection of fasteners with different defects. Although the proposed model is suitable for real-time operation, it requires labelling of fasteners with different defect types for object detection.



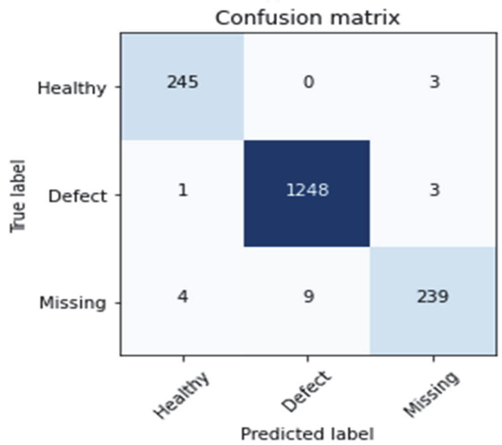
(a) InceptionV3



(b) VGG16



(c) MobileNetv2



(d) The proposed CNN

Figure 14 Confusion matrices of different CNN models

Table 4 Performance comparison with different transfer learning models

Model	Precision	Recall	*F*1	Accuracy	Processing Time / ms	Number of trainable params / millions	Model Size / MB
InceptionV3	91.66	93.66	92.33	96.00	6.0	23.8	97
VGG16	93.33	91.00	92.00	95.00	6.4	138.4	204
MobileNetV2	92.33	90.66	91.66	96.00	2.17	3.4	16.6
Proposed CNN	98.33	98.00	97.66	99.00	1.53	0.3	1.71

Table 5 The comparison results of fastener recognition methods

Reference	Used Method	Number of classes	Accuracy rate / %	FPS
[3]	Deep learning-based segmentation and template matching	3	93.30	21.10
[11]	Faster RCNN-based fastener inspection	3	97.90	260.00
[13]	Contrast enhancement and improved CNN	3	96.80	221.00
[15]	Visual similarity using two CNNs	4	92.69	196.00
[24]	RGBD-image based SVM	3	95.83	71.37
[26]	UNET-based segmentation and improved AlexNet	3	97.52	188.00
[27]	Image in-painting and VGG19	3	97.97	N/A
[28]	Yolox-Nano model	3	98.07	54.35
Ours	YOLOv4-Tiny+Developed CNN Model	3	98.57	279.06

4 CONCLUSION

Detection of rail fastener defects with a hybrid deep learning-based approach was developed in this study. For this purpose, we focused on inspecting two types of defects, including defective and missing fastener defects. The proposed framework is based on the YOLOv4-Tiny model-based fastener detection and defect classification with a developed lightweight CNN model. The inverted residual blocks used in the CNN model reduce the network size by reducing the number of trainable parameters. The effectiveness of the proposed framework has been validated with the collected dataset.

The YOLOv4-Tiny model used for real-time fastener detection achieved a *mAP* ratio of 0.99. The lightweight CNN model proposed in comparative experiments was compared with known transfer learning models such as VGG16, InceptionV3, and MobilenetV2. It has been shown that the proposed CNN model is more efficient in terms of accuracy, model size, number of parameters as well as running speed. In addition, it has been proven that the proposed approach is more effective in both accuracy and processing speed than the approaches proposed in the literature for fastener detection.

Acknowledgments

This work was supported by the Scientific Research Projects Coordination Unit of Firat University. Project number ADEP.22.02.

5 REFERENCES

[1] Molodova, M., Li, Z., Núñez, A., & Dollevoet, R. (2014). Automatic detection of squats in railway infrastructure. *IEEE Transactions on Intelligent Transportation Systems, 15*(5), 1980-1990. https://doi.org/10.1109/TITS.2014.2307955

[2] Feng, H., Jiang, Z., Xie, F., Yang, P., Shi, J., & Chen, L. (2013). Automatic fastener classification and defect detection in vision-based railway inspection systems. *IEEE transactions on instrumentation and measurement, 63*(4), 877-888. https://doi.org/10.1109/TIM.2013.2283741

[3] Tu, Z., Wu, S., Kang, G., & Lin, J. (2021). Real-time defect detection of track components: Considering class imbalance and subtle difference between classes. *IEEE Transactions on Instrumentation and Measurement, 70*, 1-12. https://doi.org/10.1109/TIM.2021.3117357

[4] Zhang, H., Jin, X., Wu, Q. J., Wang, Y., He, Z., & Yang, Y. (2018). Automatic visual detection system of railway surfaced efects with curvature filter and improved Gaussian mixture model. *IEEE Transactions on Instrumentation and Measurement*, *67*(7), 1593-1608. https://doi.org/10.1109/TIM.2018.2803830

[5] Yu, H., Li, Q., Tan, Y., Gan, J., Wang, J., Geng, Y. A., & Jia, L. (2018). A coarse-to-fine model for rail surface defect detection. *IEEE Transactions on Instrumentation and Measurement*, *68*(3), 656-666. https://doi.org/10.1109/TIM.2018.2853958

[6] Zhang, D., Song, K., Wang, Q., He, Y., Wen, X., & Yan, Y. (2020). Two deep learning networks for rail surface defect inspection of limited samples with line-level label. *IEEE Transactions on Industrial Informatics*, *17*(10), 6731-6741. https://doi.org/10.1109/TII.2020.3045196

[7] Gibert, X., Patel, V. M., & Chellappa, R. (2016). Deep multitask learning for railway track inspection. *IEEE transactions on intelligent transportation systems*, *18*(1), 153-164. https://doi.org/10.1109/TITS.2016.2568758

[8] He, Y., Wu, J., Zheng, Y., Zhang, Y., & Hong, X. (2022). Track defect detection for high-speed maglev trains via deeplearning. *IEEE Transactions on Instrumentation and Measurement*, *71*, 1-8. https://doi.org/10.1109/TIM.2022.3151165

[9] Liu, J., Ma, Z., Qiu, Y., Ni, X., Shi, B., & Liu, H. (2021). Four discriminator cycle-consistent adversarial network for improving railway defective fastener inspection. *IEEE Transactions on Intelligent Transportation Systems*, *23*(8), 10636-10645.https://doi.org/10.1109/TITS.2021.3095167

[10] Zhuang, L., Qi, H., & Zhang, Z. (2021). The automatic rail surface multi-flaw identification based on a deep learning powered framework. *IEEE Transactions on Intelligent Transportation Systems*, *23*(8), 12133-12143. https://doi.org/10.1109/TITS.2021.3109949

[11] Wei, X., Yang, Z., Liu, Y., Wei, D., Jia, L., & Li, Y. (2019). Railway track fastener defect detection based on image processing and deep learning techniques: A comparative study. *Engineering Applications of Artificial Intelligence*, *80*, 66-81. https://doi.org/10.1016/j.engappai.2019.01.008

[12] Teng, Z., Liu, F., & Zhang, B. (2016). Visual railway detection by superpixel based intracellular decisions. *Multimedia Tools and Applications*, *75*, 2473-2486. https://doi.org/10.1007/s11042-015-2654-x

[13] Aydin, I., Sevi, M., Salur, M. U., & Akin, E. (2022). Defect classification of railway fasteners using image preprocessing and alightweight convolutional neural network. *Turkish Journal of Electrical Engineering and Computer Sciences*, *30*(3), 891-907. https://doi.org/10.55730/1300-0632.3817

[14] Dou, Y., Huang, Y., Li, Q., & Luo, S. (2014). A fast template matching-based algorithm for railway bolts detection. *International Journal of Machine Learning and Cybernetics*, *5*, 835-844. https://doi.org/10.1007/s13042-013-0223-z

[15] Liu, J., Huang, Y., Zou, Q., Tian, M., Wang, S., Zhao, X., Dai, P., & Ren, S. (2019). Learning visual similarity for inspecting defective railway fasteners. *IEEE Sensors Journal*, *19*(16), 6844-6857. https://doi.org/10.1109/JSEN.2019.2911015

[16] Franca, A. S. & Vassallo, R. F. (2020). A method of classifying railway sleepers and surface defects in real environment. *IEEE Sensors Journal*, *21*(10), 11301-11309. https://doi.org/10.1109/JSEN.2020.3026173

[17] Fan, H., Cosman, P. C., Hou, Y., & Li, B. (2018). High-speed railway fastener detection based on a line local binary pattern. *IEEE Signal Processing Letters*, *25*(6), 788-792. https://doi.org/10.1109/LSP.2018.2825947

[18] Liu, J., Teng, Y., Shi, B., Ni, X., Xiao, W., Wang, C., & Liu, H. (2021). A hierarchical learning approach for railway fastener detection using imbalanced samples. *Measurement*, *186*, 110240. https://doi.org/10.1016/j.measurement.2021.110240

[19] Qi, H., Xu, T., Wang, G., Cheng, Y., & Chen, C. (2020). MYOLOv3-Tiny: A new convolutional neural network architecture for real-time detection of track fasteners. *Computers in Industry*, *123*, 103303. https://doi.org/10.1016/j.compind.2020.103303

[20] Liu, J., Liu, H., Chakraborty, C., Yu, K., Shao, X., & Ma, Z. (2021). Cascade learning embedded vision inspection of rail fastener by using a fault detection IoT vehicle. *IEEE Internet of Things Journal*. https://doi.org/10.1109/JIOT.2021.3126875

[21] Jiang, Z., Zhao, L., Li, S., & Jia, Y. (2020). Real-time object detection method based on improved YOLOv4-tiny.

[22] Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1251-1258. https://doi.org/10.48550/arXiv.1610.02357

[23] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4510-4520. https://doi.org/10.48550/arXiv.1801.04381

[24] Zhong, H., Liu, L., Wang, J., Fu, Q., & Yi, B. (2022). A real-time railway fastener inspection method using the lightweight depth estimation network. *Measurement*, *189*, 110613. https://doi.org/10.1016/j.measurement.2021.110613

[25] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for

computer vision. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2818-2826. https://doi.org/10.1109/CVPR.2016.308

[26] Liu, J., Teng, Y., Ni, X., & Liu, H. (2021). A fastener inspection method based on defective sample generation and deep convolutional neural network. *IEEE Sensors Journal*, *21*(10), 12179-12188. https://doi.org/10.1109/JSEN.2021.3062021

[27] Su, S., Du, S., & Lu, X. (2022). Geometric Constraint and Image Inpainting-Based Railway Track Fastener Sample Generation for Improving Defect Inspection. *IEEE Transactions on Intelligent Transportation Systems*, *23*(12), 23883-23895. https://doi.org/10.1109/TITS.2022.3207490

[28] Hu, J., Qiao, P., Lv, H., Yang, L., Ouyang, A., He, Y., & Liu, Y. (2022). High speed railway fastener defect detection by using improved YoLoX-Nano Model. *Sensors*, *22*(21), 8399. https://doi.org/10.3390/s22218399

Contact information:

İlhan AYDIN
(Corresponding author)
Fırat University,
Elazığ, Turkey
E-mail: iaydin@firat.edu.tr

Mehmet SEVİ
Muş Alparslan University,
Muş, Turkey
E-mail: m.sevi@alparslan.edu.tr

Erhan AKIN
Fırat University,
Elazığ, Turkey
E-mail: eakin@firat.edu.tr

Emre GÜÇLÜ
Fırat University,
Elazığ, Turkey
E-mail: eguclu@firat.edu.tr

Mehmet KARAKÖSE
Fırat University,
Elazığ, Turkey
E-mail: mkarakose@firat.edu.tr

Hssen ALDARWICH
Fırat University,
Elazığ, Turkey
E-mail: hussendarwesh6@gmail.com