ESKİŞEHİR TECHNICAL UNIVERSITY JOURNAL OF SCIENCE AND TECHNOLOGY
A- APPLIED SCIENCES AND ENGINEERING

Estuscience – Se, 2024, 25 [4] pp. 557-566, DOI: 10.18038/estubtda.1479970

RESEARCH ARTICLE

THE OPTIMIZATION OF RAILWAY FASTENER DEFECT DETECTION VIA ACTIVATION FUNCTION ADAPTATIONS

Ridvan OZDEMIR [1, 2], Mehmet KOC [3 *]

[1] Eti Makine, Eskisehir, Turkey.
[2] Institution of Graduate Schools, Bilecik Seyh Edebali University, Bilecik, Turkey
ridvan.ozdemir@etimakine.com.tr - 0000-0002-8599-1709

[3] Department of Computer Engineering, Faculty of Engineering, Eskisehir Technical University, Eskisehir, Turkey
mehmetkoc@eskisehir.edu.tr - 0000-0003-2919-6011

Abstract

Manual control of rail defect detection is slow and costly. Deep learning methods can detect some of these defects to a certain extent. However, existing systems produce too many false positives due to environmental factors, resulting in labor and cost losses. One of the most important components in railway systems is the fastener, and their failure can lead to severe accidents. In this study, we developed a deep learning-based method that is designed to remain robust against foreign objects and environmental conditions when detecting railway fasteners. By employing various activation functions and expanding the training dataset through data augmentation techniques, our method significantly reduces false alarms. The best-performing activation function in our tests achieved an F1-score of 0.99 and a mean average precision (mAP) of 100%. Testing on a dataset provided by TCDD Railway Research & Technology Centre (DATEM) confirms the efficacy of our approach, demonstrating a notable decrease in unnecessary work and associated costs.

Keywords

YOLOv4,
Railway component,
Deep learning,
Activation function,
Fastener defect

Time Scale of Article

Received :07 May 2024
Accepted : 27 October 2024
Online date : 27 December 2024

1. INTRODUCTION

Railway transportation has a critical role in both passenger and freight transportation since it is economical and safe. The importance of high-speed trains in passenger transportation is increasing day by day due to the comfort and speed they provide. The increasing use of rail transportation also makes the detection and maintenance of railroad faults more critical. In the past, manual fault detection and control operations were both costly and slow. With the development of technology, these manual processes can now be performed much faster using vision and artificial intelligence-based algorithms. Vision-based methods are widely adopted for railway fastener detection because of their precision and ease of integration with existing monitoring systems. Other techniques, such as acoustic and laser-based methods, have also been explored in some studies. [1, 2]. However, the defect detection models used have not yet reached the desired level. Single-Stage Object Detection (SSD) is the general name for object detection methods that perform object detection and classification simultaneously within a single neural network architecture and are effective in real-time applications. The main advantages of SSD are its simplicity, speed, and suitability for detection of objects with different sizes. In this study, the SSD model You Look Only Once v4 (YOLOv4) is used to detect defects in fasteners that are critical for railway safety. The performance of the SSD model is analyzed using different activation functions. Activation functions directly affect the performance of the model due to the nonlinearity they add to the

*Corresponding Author: mehmetkoc@eskisehir.edu.tr

model. [3]. This feature distinguishes them from simple linear models and enables them to provide successful results for challenging real-world problems.

In 2019, Lin et al. published a paper in which they mentioned that railway fasteners are one of the most important components of the railway, and their damage control is done manually. They trained the YOLOv3 model with 20 km of GoPro images and achieved 89% and 95% precision and recall, respectively [4]. In 2020, Qi et al. stated in their article that real-time detection of defects in railway fasteners would be a significant improvement in this field. They also noted that this task is challenging due to the memory and processor limitations of embedded maintenance systems. With the MYOLOv3-Tiny model they developed, the accuracy value reached 99.33% and the memory consumption was reduced by 43% compared to the YOLOv3-Tiny model [5]. Güçlü et al. introduced a novel method utilizing fuzzy logic and YOLOv4 in their study. They achieved a success rate of 99.25% in classifying faulty fasteners on the test dataset [6]. In 2021, Liao et al. investigated the effects of activation functions on the learning process with benchmarks and successfully detected surface defects with the hybrid model they developed, achieving an of 98.64% mAP value in their study [7]. Şener et al. propose an AI-based model using the Tensorflow library and deep learning approaches to identify problems in railway tracks with an overall accuracy of 92.2% [8]. Ozdemir and Koc used a semi-supervised deep learning strategy with a student-teacher model and YOLOv4 to detect various defects encountered in railways [9]. They used a new dataset from the Turkish State Railways (TCDD), which contains five different defects, including fastener faults. Their method automates the process of adding appropriately pseudo-labeled images to the training dataset, enhancing model performance, and lowering the labor-intensive task of manual labeling. Sevi et al. proposed a deep learning-based approach to classify defects in railway fasteners by generating defective data from images of healthy rail fasteners. They used CNN, VGG16, and ResNet50 models to classify fastener defects, achieving a 100% accuracy rate with the proposed method [10]. He et al. added a SE attention mechanism, replaced the backbone with a lightweight MobileNet-V2 network, and included Mixup data augmentation to improve the Yolov4 model for track fastening service status detection [11]. As compared to the conventional Yolov4 model, their results demonstrated a 67.39% increase in processing speed and an 83.2% MAP increase in detection accuracy. Yılmazer et al. aimed to detect three different fault classes, including missing fasteners, along with the state of healthy sleepers, with the model's accuracy determined to be 95% in the performed tests. [12].Zengzhen Mi et al. found that their improved version of the YOLOv4 model outperformed YOLOv3, YOLOv4, YOLOv5, YOLOv6, Faster RCNN, and SSD models, achieving an F1 score of 0.925 compared to YOLOv6's 0.914 [13]. In our work, we use the original YOLOv4 model to evaluate the impact of different activation functions on railway fastener detection.

The dataset provided by TCDD was obtained using a track inspection system called VCUBE. There were many false positives in the detections made by V-CUBE, with the precision value being about 11%; this means that only 301 out of 2730 detections were true. The false positives (FPs) may occur due to the ballasts on the fasteners and other environmental conditions, resulting in very low performance. This study, focusing on fastener defects, was conducted to address this problem. Additionally, the effects of activation function selection and data augmentation on the model's performance were analyzed.

2. MATERIALS and METHODS

2.1. Dataset

One of the most critical factors in the success of deep learning models is the dataset [14]. The more accurate and consistent the labels and tags in the dataset used to train the model, and the higher the class diversity, the more successful the model will be. The dataset in this study, which was created using railroad images provided by TCDD, was obtained using a track inspection system. These images were analyzed, and the marking and labeling were done carefully. After analyzing all the images, the following three classes were created:

- **Sleeper**: Placed under the rails and used to support the rails. It stabilizes the rails by distributing the load under the rails and ensures the contact of the rails with the ground.
- **Fastening OK**: Fasteners are used to attach rails and rail components. These parts connect the rails to the bearings and substructure and ensure that the track is safe and stable. Various types of fasteners can be used to ensure that the rails are held in place and secured. These systems increase the life of the track by absorbing vibrations and carrying the loads of the rails. Fastening systems typically consist of rail clips, screws, wedges, and other hardware.
- **Fastening NOK**: If the fastener is deformed for any reason, or if it rotates and loses contact with the rail, the safety of the rail line is compromised. Fasteners in the images under this condition are labeled as "Fastening NOK".

Labeled sample images of the Sleeper, Fastening OK, and Fastening NOK classes are shown in Figure 1.

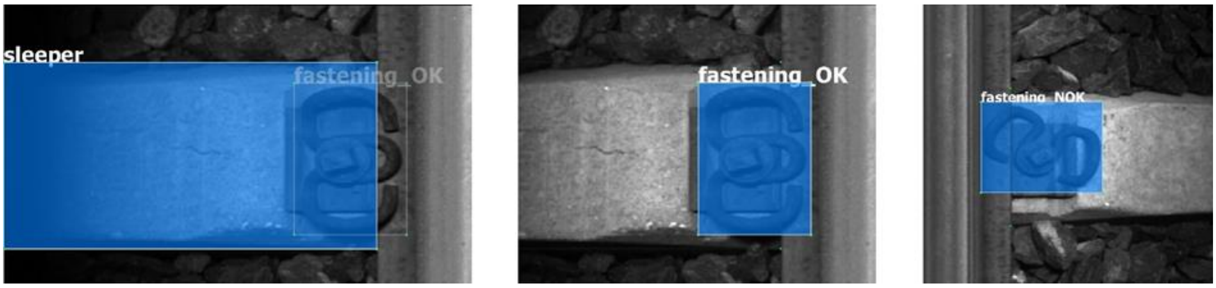

Figure 1. Examples of Sleeper, Fastening OK and Fastening NOK Classes

After analyzing the railroad images received from TCDD, approximately 89% of the classifications were found to be incorrect. For example, Figure 2 shows images that were initially classified as incorrect in the dataset but were actually correct. By analyzing all images, those that were truly defective were identified and separated from the non-defective ones, as demonstrated in Figure 3.

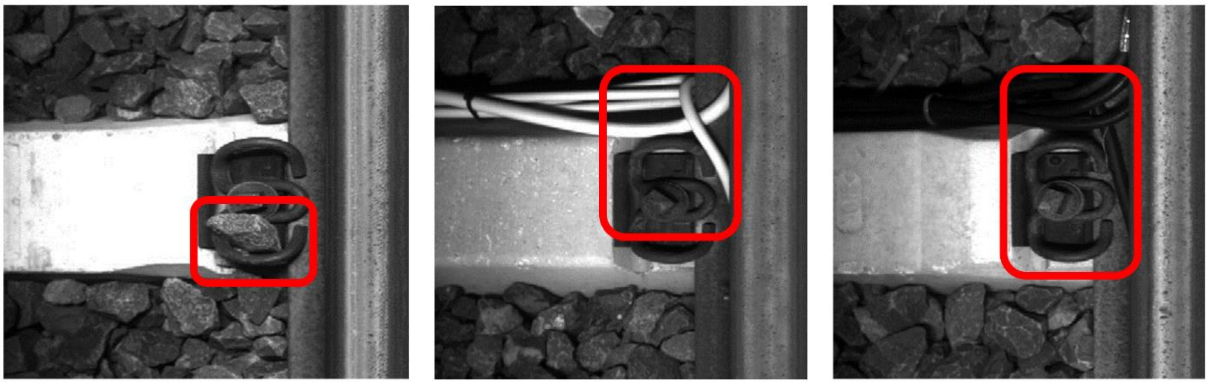

Figure 2. Normal images that are labeled as defective

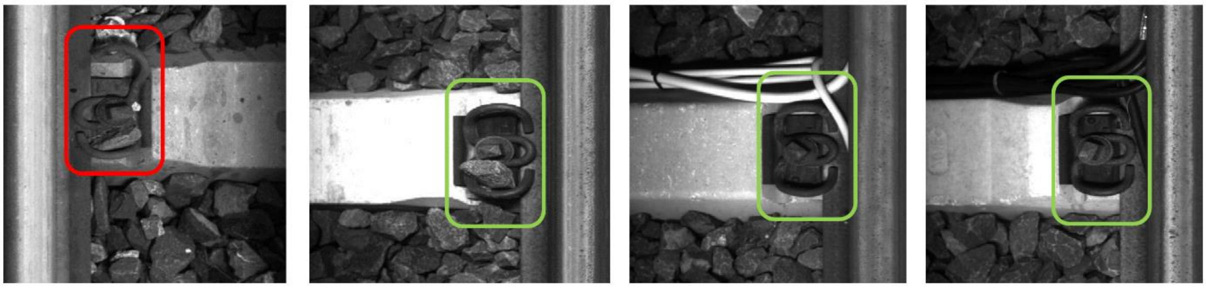

Figure 3. Checking for images that are labeled as defective

Training, validation, and test datasets were then created from the acquired images, which were labeled in YOLO format using the labelImg program. In addition to labeling the fasteners, the sleepers in the images were also labeled, creating a dataset that facilitates the detection of missing fastener defects if needed. The numbers of images in the training, validation, and test datasets, along with the number of classes in these images, are provided in Table 1.

Table 1. Number of Images in Training, Validation, and Test Datasets

Dataset	image	sleeper	fastening_OK	fastening_NOK
Training	812	812	571	241
Validation	100	100	68	32
Test	100	100	72	28

The use of data augmentation techniques, specifically adjustments in contrast and orientation, expanded the training dataset to four times its original size by increasing the number of images. The augmented training dataset has 3248 images which contains 3248 sleepers, 2284 fastening_OK, and 964 fastening_NOK samples.

2.2. YOLOv4

YOLOv4 is an object detection algorithm that provides advantages such as high accuracy, speed, and scalability [15]. In this study, the YOLOv4 detection algorithm is used to detect the defects of railway fasteners with high accuracy and reduce the number of false positives.

The architecture of YOLOv4 generally consists of Backbone, Neck, Head sections. The tasks of these sections are briefly described below:

- **Backbone**: In YOLOv4, it is based on CSPDarknet53, a pre-trained deep learning model. CSPDarknet53 [16] is a Convolutional Neural Network (CNN) based architecture and includes a series of convolution and pooling layers to transform the image into smaller feature maps.
- **Neck**: This section processes the output from the backbone network and contains several convolution and scaling layers that combine and compress feature maps at different scales. Typically, a neck consists of several paths from the bottom up and several paths from the top down. In the neck part of the architecture of YOLOv4, SPP (Spatial Pyramid Pooling) add-on module and PANet path aggregation were chosen. PAN (Path Aggregation Network) in this section [17] is a network that performs the operations of merging and expanding feature maps. SPP [18] layer contains max-pooling outputs with 1×1, 5×5, 9×9, 13×13 kernel sizes and adds imaginary connections, so the model can be trained shorter time also had a higher performance.
- **Head:** The head section is the dense prediction layer. It contains bounding boxes and the class of each box is estimated. The models implemented in YOLOv3 are used here. Thanks to the

mechanism that YOLOv4 has, the model is able to recognize both small and large objects with the same level of accuracy.

A figure summarizing the structure of the YOLOv4 algorithm is shown in Figure 4. Yolov4 uses a composite loss function which is made up of three components: (*i*) Bounding box regression loss ($\mathcal{L}_{\text{loc}}$) improves the overlap, center distance, and aspect ratio consistency to ensure that predicted bounding boxes in the object detection tasks are accurately aligned with the ground truth. (*ii*) Confidence loss ($\mathcal{L}_{\text{conf}}$) penalizes incorrect object evaluations by using binary cross-entropy to determine the model's confidence about whether the bounding box contains an object. (*iii*) Classification loss ($\mathcal{L}_{\text{clc}}$) quantifies how well a model can classify an object inside a bounding box. The total loss is the weighted sum of these three loses:

$$\mathcal{L} = \lambda_{\text{loc}}\mathcal{L}_{\text{loc}} + \lambda_{\text{conf}}\mathcal{L}_{\text{conf}} + \lambda_{\text{cls}}\mathcal{L}_{\text{cls}} \quad (1)$$

where λ_{loc}, λ_{conf}, and λ_{cls} are the corresponding weight coefficients.

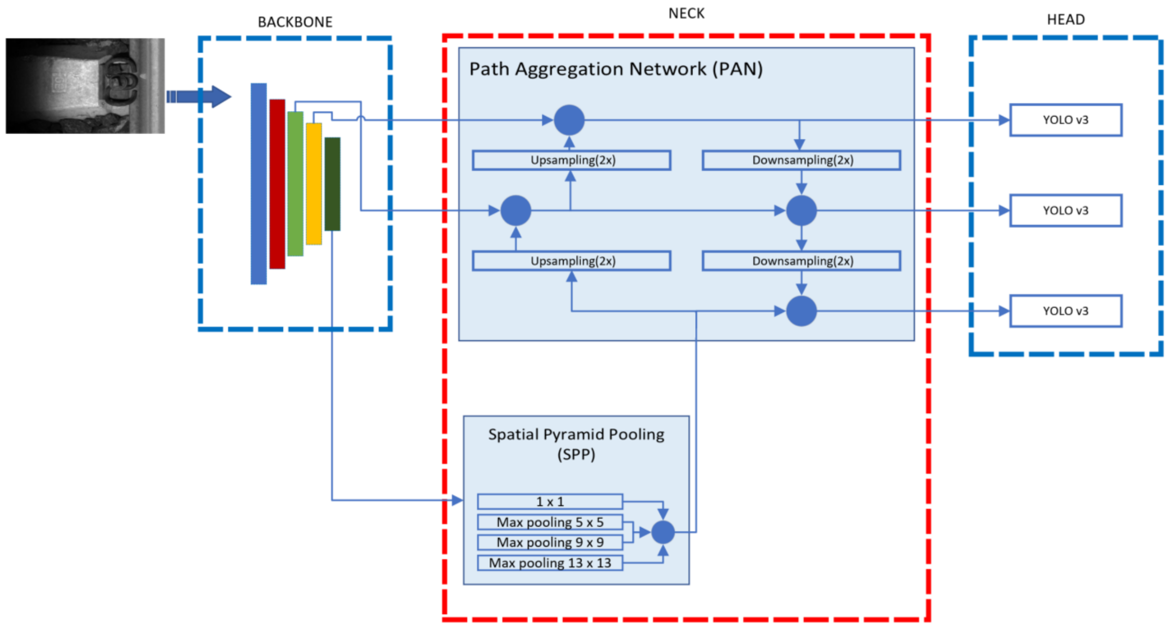

Figure 4. YOLOv4 Architecture [19]

2.3. Activation Functions

Some activation functions commonly used in the literature are selected to investigate the learning performance of the model and their performances are compared on the test dataset. The activation functions are designed to optimize the training of the model, deal with gradient vanishing/exploding problems, and improve the learning process. The ideal activation function may differ based on the model and application. Figure 5 depicts some of the most frequent activation functions used to train deep learning models.

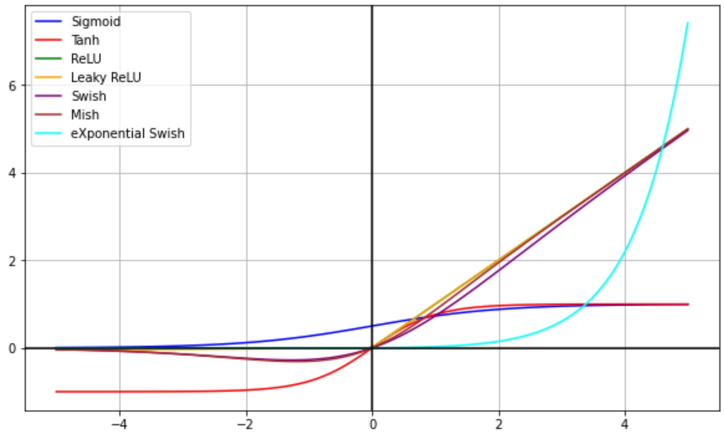

Figure 5. Comparison of Activation Functions

In this study, the activation functions Leaky ReLU (Rectified Linear Unit), Swish, x-Swish (Exponential Swish) and Mish were employed during the model training process. Each of these activation functions is described in detail below:

Leaky ReLU: It multiplies the input by a small slope value when the input is negative. This means that $x < 0$ when the Leaky ReLU is multiplied by α. Adding a small slope to negative inputs that are zero is intended to increase the model's ability to learn.

$$f(x) = \begin{cases} x & x \geq 0 \\ \alpha x & x < 0 \end{cases} \tag{2}$$

Swish: It applies the sigmoid function to the input and then multiplies the input by the result. The Swish function produces a smooth version of the sigmoid as the value of x increases. Swish has smoother derivatives, which can make training more stable.

$$f(x) = x \ \sigma(x) = \frac{x}{1 + \mathrm{e}^{-x}} \tag{3}$$

x-Swish (eXponential Swish): x-Swish has been proposed to improve the Swish function. x-Swish aims to improve the performance of Swish by adding an additional scaling factor.

$$f(x) = \beta x \sigma(x) = \frac{\beta x}{1 + \mathrm{e}^{-x}} \tag{4}$$

Mish: Mish is a nonlinear function that multiplies the input by the hyperbolic tangent (tanh) function of x. Mish has a smoother curve than other activation functions, especially Swish and ReLU.

$$f(x) = x\tanh(\ln(1 + \mathrm{e}^{x})) \tag{5}$$

Mish*: Mish* activation function is combined with both the Mish activation function and the SAM (Spatial Attention Module) module. This can help the model to focus on important regions in the feature map. The mathematical formulation of this special function may vary depending on the application context, but in general it includes mish activation function and spatial attention features. Mish* aims to achieve better performance in tasks such as object detection by combining the spatial attention feature with the Mish function. Unlike the Mish activation function, the plug-in modules have not only SPP, but also SAM, and the head part uses the linear activation function instead of the logistic one.

3. EXPERIMENTS

To compare the performance of the fault detection model developed with the YOLOv4 object detection algorithm trained on our custom railway fastener dataset, we conducted 10 different experiments using various activation functions. In these experiments, the YOLOv4 model is trained using hyperparameters with an input size of 352 × 352 pixels, a batch size of 64, and a mini-batch size of 32. The learning rate is set at 0.001 and the momentum at 0.949. The training is conducted over 12,000 iterations with the Adam optimizer. The experiments are conducted on a desktop PC operating Windows 11, equipped with an Intel i7-11700 CPU at 2.50GHz, an RTX3060 Ti GPU, and 16GB of RAM.

Evaluation Metrics:

The evaluation of detection accuracy involves the utilization of metrics such as Precision, Recall, F_1-score, Intersection over Union (IoU), and mean average precision (mAP), which are delineated as follows:

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{6}$$

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{7}$$

$$\mathrm{F_1} - \mathrm{score} = 2 \times \frac{\mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}} \tag{8}$$

$$\mathrm{IoU} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \tag{9}$$

$$\mathrm{mAP} = \frac{1}{\mathrm{C}} \sum_{\mathrm{i}=0}^{\mathrm{C}} \mathrm{AP_i} \tag{10}$$

Here C represents the total number of categories within the image dataset. TP, FN, and FP denote true positive, false negative, and false positive, respectively. In this study, the prediction anchor box is classified as true positive (TP) if its Intersection over Union (IoU) value equals or exceeds a predefined threshold, denoted as T, which is set at 0.5.

Table 2. Comparison of Activation Function Performances

Activation	F_1-score	recall	precision	IoU	mAP
leaky	0.99	0.99	0.98	93.07	98.62
swish	0.86	0.93	0.80	75.80	99.20
x-swish	0.92	0.95	0.90	87.13	97.97
mish	**0.93**	**1.00**	**0.87**	**81.83**	**99.24**
mish*	0.98	0.99	0.98	92.57	98.70

To investigate the effects of activation functions on the performance of the model, Swish, x-Swish, Mish, and Mish* activation functions were selected one at a time instead of Leaky ReLU and the model was trained on the training dataset. As a result of this training, performance comparisons were made on the test dataset with the models obtained using different activation functions. The results of these tests were compared with the overall performance indices in Table 2, and the class-based performance results are shown in Table 3 on the dataset. With the Mish activation function detecting 27 out of 28 defective fasteners, it achieved the best result, marking an improvement of 3.57% and attaining an F_1-score of 0.93. Although the best result was obtained in the detection of faulty fasteners, the overall performance was degraded due to the high number of false positives in the "sleeper" and "fastening_OK" classes.

Table 3. Class Based Performance of Activation Functions

	Sleeper			Fastening_OK			Fastening_NOK		
activation	**AP**	**TP**	**FP**	**AP**	**TP**	**FP**	**AP**	**TP**	**FP**
leaky	100.00	100	0	99.67	72	4	96.18	26	0
swish	100.00	89	36	99.12	70	7	98.49	26	2
x-swish	100.00	94	11	99.01	71	6	94.89	26	5
mish	**100.00**	**100**	**24**	**100.00**	**72**	**6**	**97.73**	**27**	**1**
mish*	100.00	100	0	99.67	72	5	96.43	26	0

To improve the performance of the error detection model, the number of images in the training dataset was increased using data augmentation methods including flipping, contrast reduction, and contrast enhancement. In this way, a training data set was obtained that is more adaptable to changing light conditions at different times of the day, when the weather is cloudy or sunny. As a result of these operations, the number of images was increased, and the training dataset was expanded to four times its original size. With the extended training dataset, the models with leaky ReLU, Swish, X-Swish, Mish, and Mish* activation functions were retrained. The performance of the new models was compared on the test dataset. The results were compared with the overall performance indices listed in Table 4 and on a class-by-class basis in Table 5. Analysis of the tables shows that the models trained with both the Swish and Mish* activation functions successfully detected all defective images. However, the model trained with the Mish* activation function outperformed the Swish-activated model on the F1-score index, achieving 0.99 compared to 0.93. This superiority is attributed to the Swish model producing more false positives and its failure to detect some true positives in the "sleeper" and "fastening_OK" classes. As a result, Mish* was found to be the most successful function.

Table 4. Comparison of Activation Function Performances on an Extended Training Dataset

Activation	F_1-score	recall	precision	IoU	mAP
leaky	0.99	0.99	0.98	92.93	100.00
swish	0.93	0.95	0.90	84.48	100.00
x-swish	0.93	0.93	0.93	87.16	99.96
mish	0.97	0.98	0.96	90.51	99.94
mish*	**0.99**	**1.00**	**0.99**	**93.51**	**100.00**

Table 5. Class Based Performance of Activation Functions

	Sleeper			Fastening_OK			Fastening_NOK		
Activation	**ap**	**TP**	**FP**	**ap**	**TP**	**FP**	**ap**	**TP**	**FP**
leaky	100.00	100	0	100.00	72	4	100.00	26	0
swish	100.00	94	14	100.00	68	5	100.00	28	1
x-swish	100.00	87	13	100.00	72	2	99.88	27	0
mish	100.00	96	5	99.94	72	4	99.88	27	0
mish*	**100.00**	**100**	**0**	**100.00**	**72**	**3**	**100.00**	**28**	**0**

4. CONCLUSION

The ballasts on the fasteners and other environmental factors cause a lot of false positives (FPs) in the current system that generates the dataset for TCDD, which leads to poor defect detection performance. YOLOv4 models incorporating Leaky ReLU, Swish, x-Swish, Mish, and Mish* activation functions were trained on this new dataset, which was organized considering the environmental conditions. When evaluating the performance of the 'Railway Fastener Fault Detection Module' on the test dataset, it was initially observed that one faulty fastener was missed. However, following the implementation of data augmentation improvements, all faulty fasteners were successfully detected. This module achieved an improvement of 1% in recall and precision values and 1.38% in mAP by using different activation functions compared to the initial setup. Consequently, many false positives were avoided, unnecessary work loss was prevented, and productivity was increased.

ACKNOWLEDGMENTS

This research received funding from the Eskisehir Technical University Scientific Research Projects Commission, under grant number 21GAP081. The authors express their gratitude to TCDD Railway Research & Technology Centre (DATEM) for supplying the dataset. For access to the training and test datasets utilized in the experiments, please contact the authors.

CONFLICT OF INTEREST

The authors stated that there are no conflicts of interest regarding the publication of this article.

CRediT AUTHOR STATEMENT

Ridvan Ozdemir: Methodology, Software, Validation, Data Curation, Writing – Original Draft, Visualization, Conceptualization. **Mehmet Koc:** Writing – Review & Editing, Supervision, Resources, Conceptualization.

REFERENCES

[1] Dinhobl G, Petz M, Kümmritz S, Hutterer H. On Measurement of Railway Noise: Usability of Acoustic Camera. In: Degrande G, Lombaert G, Anderson D, de Vos P, Gautier P-E, Iida M, Nelson JT, Nielsen JCO, Thompson DJ, Tielkes T, Towers DA, editors. Noise and Vibration Mitigation for Rail Transportation Systems. Cham: Springer International Publishing, 2021, pp. 234–241.

[2] Assali P, Viguier F, Pollet N. Contribution of Terrestrial Laser Scanning for monitoring and inspection of railway infrastructure. In Proceedings of the World Congress on Railway Research; 2013 Nov 25-28; Sydney, Australia. 2013.

[3] Szandała T. Review and comparison of commonly used activation functions for deep neural networks. In: Bio-inspired Neurocomputing 903, Bhoi AK, Mallick PK, Liu CM, Balas VE, editors. Studies in Computational Intelligence, vol. 903. Singapore: Springer Singapore, 2021, pp. 203–224.

[4] Lin Y-W, Hsieh C-C, Huang W-H, Hsieh S-L, Hung W-H. Railway track fasteners fault detection using deep learning. In: 2019 IEEE Eurasia Conference on IOT, Communication and Engineering, 2019, pp. 187–190.

[5] Qi H, Xu T, Wang G, Cheng Y, Chen C. MYOLOv3-Tiny: A new convolutional neural network architecture for real-time detection of track fasteners. Computers in Industry; 2020; 123: 103303.

[6] Güçlü E, Aydin İ, Şahbaz K., Akin E, Karaköse M. Detection of defects in railway fasteners using YOLOv4 and fuzzy logic (article in Turkish with an abstract in English). Railway Engineering 2021; 14: 249-262.

[7] Liao X, Lv S, Li D, Luo Y, Zhu Z, Jiang C. YOLOv4-MN3 for PCB surface defect detection. Applied Sciences 2021; 11(24):1-17.

[8] Şener A, Ergen B, Toğaçar M. Fault detection from images of railroad lines using the deep learning model built with the tensorflow library. Turkish Journal of Science & Technology 2022; 17(1): 47-53.

[9] Ozdemir R, Koc M. On the enhancement of semi-supervised deep learning-based railway defect detection using pseudo-labels. Expert Systems With Applications 2024; 251: 124105.

[10] Sevi̇ M, Aydın İ, Karaköse M. Classification of railway fasteners by deep learning methods (article in Turkish with an abstract in English). European Journal of Science and Technology 2022; 35: 268-274.

[11] He J, Wang W, Yang N. Research on track fastener service status detection based on improved YOLOv4 model. Journal of Transportation Technologies; 14: 212-223.

[12] Yılmazer M, Karaköse M, Aydın İ, Akın E. Multiple fault detection in railway components with mask R-CNN deep neural network (article in Turkish with an abstract in English) Cukurova University Journal of the Faculty of Engineering 2022; 37(4); 1103-1111.

[13] Mi Z. Research on steel rail surface defects detection based on improved YOLOv4 network. Frontiers in Neurorobotics 2023; 17: 1-11.

[14] Alzubaidi L, Zhang J, Humaidi AJ, Al-Dujaili A, Duan Y, Al-Shamma O, Santamaria J, Fadhel MA, Al-Amidie M, Farhan L. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions 2021; 8(1); 1-74.

[15] Bochkovskiy A, Wang C-Y, Liao H-YM. YOLOv4: Optimal speed and accuracy of object detection. arXiv preprint 2020; 2004.10934.

[16] Wang C-Y, Liao H-YM, Wu YH, Chen P-Y, Hsieh J-W, Yeh I-H, CSPNet: a new backbone that can enhance learning capability of CNN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops; 2020; pp. 390-391.

[17] Liu S, Qi L, Qin H, Shi J, Jia J. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2018; pp. 8759-8768.

[18] He K, Zhang X, Ren S, Sun J. Spatial pyramid pooling in deep convolutional networks for visual recognition IEEE Transactions of Pattern Analysis and Machine Intelligence 2015; 37(9): 1904-1916.

[19] Getting Started with YOLOv4 - MATLAB & Simulink. Accessed: Feb. 23, 2024. [Online]. Available: https://www.mathworks.com/help/vision/ug/getting-started-with-yolo-v4.html