Turkish Journal of Science & Technology
17(1), 47-53, 2022

Research Paper
https://doi.org/10.55525/tjst.1056283

Fault Detection from Images of Railroad Lines Using the Deep Learning Model Built with the Tensorflow Library

Abdullah ŞENER[1]*, Burhan ERGEN[2], Mesut TOĞAÇAR[3]
[1]Department of Computer Technologies, Genc Vocational School, Bingol University, Bingol, Turkey
[2]Department of Computer Engineering, Faculty of Engineering, Firat University, Elazig, Turkey
[3]Department of Computer Technology, Vocational School of Technical Sciences, Fırat University Elazig, Turkey
[*1] asener@bingol.edu.tr, [2]bergen@firat.edu.tr,[3]mtogacar@firat.edu.tr

(Geliş/Received: 11/01/2022 **Kabul/Accepted: 11/02/2022)**

Abstract: A means of transportation is the way in which an object, person, or service is transported from one place to another. Rail transportation occupies an important place in terms of cost and reliability. Most train accidents are caused by faults in railroad tracks. Detecting faults in railroad tracks is a difficult and time-consuming process compared to conventional methods. In this study, an artificial intelligence based model is proposed that can detect faults in railroad tracks. The dataset used in the study consists of defective and non-defective railroad images. The proposed model consists of foldable neural networks developed using the Tensorflow library. Softmax method was used as a classifier. An overall accuracy of 92.21% was achieved in the experiment.

Key words: Decision support systems, Deep learning, Rail fault detection, Artificial intelligence.

Tensorflow Kütüphanesi Kullanılarak Oluşturulan Derin Öğrenme Modeli ile Demiryolu Hattı Görüntülerinden Arıza Tespitinin Gerçekleştirilmesi

Öz: Ulaşım aracı, bir nesnenin, bireyin veya hizmetin bir yerden başka bir yere aktarılmasını sağlayan vasıtadır. Demiryolu ulaşımı maliyet ve güvenirlilik açısından önemli bir yere sahiptir. Tren kazaların çoğu demiryolu raylarında meydana gelen arızalardan kaynaklanmaktadır. Demiryolu hatlarındaki arızaların tespiti geleneksel yöntemlere göre zor ve zaman alıcı bir süreçtir. Bu çalışmada demiryolu hatlarında meydana gelen arızaların tespitini gerçekleştirebilen yapay zekâ tabanlı bir model önerilmiştir. Çalışmada kullanılan veri kümesi arızalı ve arızalı olmayan ray görüntülerinden oluşmaktadır. Önerilen model Tensorflow Kütüphanesi kullanılarak tasarlanmış evrişimsel sinir ağlarından oluşmaktadır. Sınıflandırıcı olarak Softmax yöntemi kullanılmıştır. Gerçekleştirilen deneyde %92,21 genel doğruluk başarısı elde edilmiştir.

Anahtar kelimeler: Karar destek sistemleri, Derin öğrenme, Ray arıza tespiti, Yapay zekâ.

1. Introduction

Road transport is the preferred mode of transport compared to sea and air transport [1]. The transportation sector directly affects factors such as the transportation of people and goods, the national and international economy, the construction and development of residential areas, and the functioning and order of other sectors. Developments in the transportation sector will also have a positive impact on the country's economy. Rail transport occupies an important place in road transport. Compared to Turkey, rail transport enjoys more attention in European countries. In our country, the share of passengers using railroads annually is about 1.1%, depending on the type of transport, while the share of freight transport is 4.6% [2].

Most accidents that occur in rail transportation are due to undetected defects in the tracks [3]. It is important to control reliability by considering rail transportation, scheduling, and economics [4]. Detecting faults in railroad tracks is a costly and time-consuming task. Traditionally, fault detection has been performed by humans. Recently, artificial intelligence-based technological infrastructure systems have come to the forefront to maximize fault detection, reduce costs, and increase safety levels. With artificial intelligence, it is possible to quickly detect faults on railroad tracks [5], [6]. Recently, many artificial intelligence-based studies have been conducted to detect faults on railroad tracks. Some of these studies are:

Ya-Wen Lin et al. examined railroad link images and detected errors using their proposed Deep Learning model. They created the dataset from the data captured by the GoPro device. Ya-Wen Lin et al. trained the dataset with the Yolo-V3 model and then performed the fault detection with test images. Ya-Wen Lin et al. achieved a success rate of 89% considering the precision metric in their studies. [7]. Yang et al. found that they could use their proposed approach to proactively maintain faults that may occur on railroad lines.

*Corresponding author: asener@bingol.edu.tr. ORCID Number of authors: [1]0000-0002-8927-5638, [2]0000-0003-3244-2615, [3]0000-0002-8264-3899

They used the ResNet model and fully foldable networks in their proposed approach. The analysis was performed to detect the connection points and faults on the railroad lines. Yang et al. used two data sets in their study. As a result of the analysis, they achieved success in the range of 74% to 91% considering the f-score metrics. [8]. Hovad et al. used the Yolo-V3 model in experimental analyzes to detect defects by examining images of railroad track fasteners. Considering the f-score criterion for detecting defective surfaces in the data set, they achieved a success rate of 84% [9]. Sysyn et al. used image preprocessing and principal component analysis (TBA) methods together to perform a contact fatigue analysis of railroad rails. Sysyn et al. processed the images morphologically and then performed efficient feature selection among the features obtained from the images using the PCA method. They showed that they could obtain accurate predictions of up to 11 meters in the surface images in the regression processes they performed with the surface crack images. [10]. Rajagopa et al. used artificial intelligence approaches to detect defects and cracks in railroad lines. Rajagopa et al. used the gray leveling matrix (GLCM) and local binary patterns (LBP) as a preprocessing step and classified the obtained features using the neural network model they developed. They achieved an overall accuracy of 94.9% in detecting defects in railroad lines. [11].

In this paper, an artificial intelligence-based deep network model is developed that can detect faulty rails on railroad tracks. In order to achieve successful results in the experimental analysis, a new convolutional neural network (ESA) model was created by testing all neural network layers of the Tensorflow library. The summary of the other parts of the paper is as follows: Information about the material and the proposed approach can be found in Chapter 2, and information about the analysis of the experiments, the preferred parameters, and the hardware information are described in Chapter 3. Information about the comparison of the study, the advantages and disadvantages of the scientific research and the results are given in Chapters 4 and 5.

2. Method

This section provides detailed information about the data set used for the experimental analysis of the study and the proposed approach.

2.1. Dataset

The dataset consists of railroad images forming the railroad lines. The dataset is public and consists of two classes. The classes consist of defective and non-defective categories. The resolution of each image is not fixed and the resolution ratio is generally of good quality. The image depth is 24 bits and the file extensions are JPG. In the defective class include loose nuts, missing parts, missing fasteners, gaps in connection points, etc. Defects. Figure 1 shows an example subset of images in the dataset. 191 of the dataset consist of defective images that occur on railroad tracks, and 192 consist of non-defective images of rails. In total, there are 383 images. [12]. In the dataset of this study, the data reserved for training, testing, and validation were combined into a single folder. Then, 80% of the dataset was randomly assigned as training data and 20% as test data for analysis. To validate the accuracy of the analysis result, the proposed approach was trained several times with the dataset and average accuracy was obtained.

(a) (b)

Figure 1. Dataset classes and sample images;
a) Non-defective images, b) Defective images

2.2. Designed ESA Model

ESA models can be designed by software developers using open source codes to perform operations such as regression, classification, species detection, abnormal region detection, and segmentation. Open source ESA models are developed using various software languages (Python, MATLAB, R, Shell, etc.). [13]. Common layers used in ESA models are convolutional layers, pooling layers, and fully connected (dense) layers. The convolution layer allows the input image to be convolved by circulating it with the preferred filter parameter. In this way, filter activation maps are created, which are circulated in each region. The activation maps are used to determine the feature values of each region in the input data. [14]. Thanks to the pooling layer, the ESA model is guided through a simpler training process. With the pooling layer, the size of the input data is reduced. In general, pooling layers are used after convolution layers. [15]. The convolutional layer pooling layer can also be used continuously across multiple layers. The dense/fully linked layer is used as the last layer of the ESA model. The dense layer collects the features obtained from the previous layers in the form of a single row. To perform the classification/regression function, which is the next step, it prepares the formation of probability values for each input data [16].

During training of the ESA model, parameters such as normalization and dropout can also be used to prevent overlearning of the model. In addition, activation functions (ReLU, Sigmoid, tanh, etc.) are used to activate not all features obtained from the layers, but the features that have a certain threshold value. While the activation features facilitate the training of the model, they also contribute to the overall performance of the model [17].

Softmax method can be used as an activation function between layers of ESA models and is generally preferred to perform the classification function after the dense layer of models. Softmax processes the input features obtained from the dense layer and determines probability scores between 0 and 1 according to the class types. In a final processing step, Softmax assigns the input data to the class type with the highest score [18].

The proposed approach was developed using the Tensorflow library and the input size was set to 300×300 pixels. The proposed approach includes convolutional layers, pooling layers, density layers and features. Softmax method was used for image classification. In composing the model, the ADAM optimization method was preferred. Detailed information about the layers and functions used in the proposed approach can be found in Table 1.

Table 1. Designed ESA model

Layer/Function	Number of Input Channels	Step Size/Value
Input	-	300×300
Convolutional	256	3×3/150×150
ReLU	-	-
Maximum Pooling	-	3×3
Convolutional	128	3×3/150×150
Normalization	-	-
ReLU	-	-
Average Pooling	-	2×2
. . .		
Convolutional	64	3×3/38×38
Normalization	-	-
ReLU	-	-
Average Pooling	-	2×2
Dilution	-	0,2
Convolutional	32	3×3/19×19
Normalization	-	-
ReLU	-	-

Average Pooling	-	2×2
Dilution	-	0,25
Convolutional	16	3×3/10×10
ReLU	-	-
Average Pooling	-	2×2
Dilution	-	0,25
Dense	-	-
Softmax	Output=2 (defective / non-defective)	

3. Findings

The Jupyter Notebook interface was used for the experimental analysis of this study and the proposed approach was created using the Google Collaboratory server. Hardware information of the computer used in the study NVIDIA GeForce® RTX™ 3060 6GB graphics card, 11th generation Intel® Core™ i7 2.3GHz (24M cache, up to 4.6GHz, 8 cores) processor and 16GB memory. The Python programming language was used in the development of this approach. The most important reason why Google Colab servers are preferred is the ability to use Tensor Processing Unit (TPU). TPUs are capable of processing faster than the graphics processing unit (GPU) [19]. In performing the analysis, the TPU unit was preferred instead of the GPU. The value of the mini-batch parameter, which allows the model to process the images simultaneously, was set to 4. [20]. The number of cycles (epoch) 32 was preferred for training the dataset. Also, the early stop parameter was used to prevent overfitting during model compilation [21]. The complexity matrix and metrics were used to measure the analysis of the proposed approach. The formulas given between equation 1 and equation 4 were used to calculate the metrics. True (T), false (F), positive (P), and negative (N) variables were used in the formulas. [22], [23]. In addition, the Matthews correlation coefficient was preferred to compare the results of the analysis. Equation 5 was used to calculate Matthews [24].

$$\text{Sensitivity} = \frac{TP}{TP+FN} \tag{1}$$

$$\text{Specificity} = \frac{TN}{TN+FP} \tag{2}$$

$$\text{F-score} = \frac{2\times TP}{2\times TP+FP+FN} \tag{3}$$

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{4}$$

$$\text{Matthews} = \frac{(TP\times TN-FP\times FN)}{\sqrt{((TP+FP)\times(TP+FN)\times(TN+FP)\times(TN+FN))}} \tag{5}$$

In the experimental analysis, 80% of the dataset was trained using the proposed approach as training data. 20% of the dataset was used as test data. The training time of the model was 5671 seconds. The graphs of the overall accuracy of the training and test data are shown in Figure 2

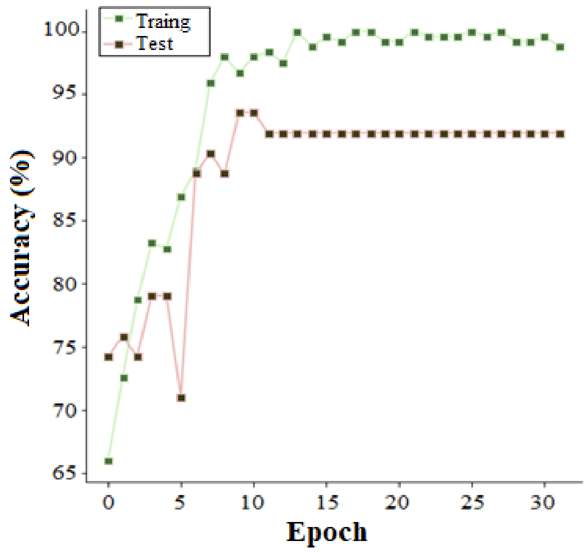

Figure 2. Training-test accuracy graph of the proposed approach

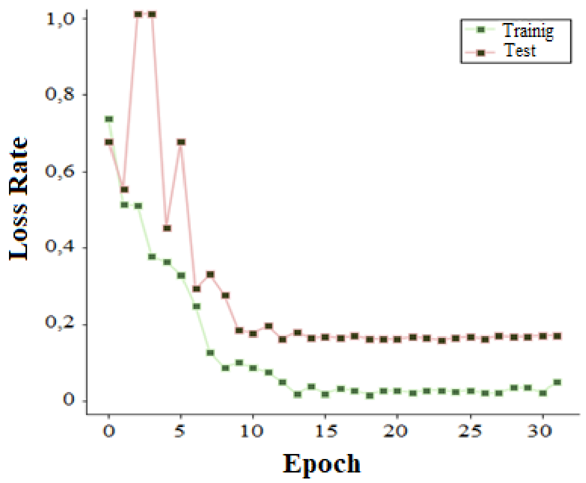

Figure 3. The training-test loss graph of the proposed approach

The result of the experimental analysis was an overall accuracy of 92.21%. The complexity matrix used to calculate the measurement metrics is shown in Figure 4. The analysis results of the measurement metrics are shown in Table 2.

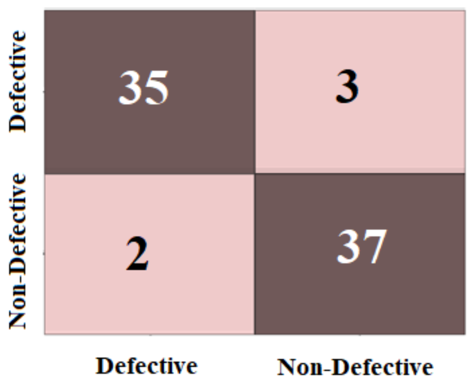

Figure 4. The complexity matrix obtained as a result of the experimental analysis

Table 2. Experimental analysis results (%)

Dataset Types	Sensitivity	Specificity	F-score	Accuracy	Matthews
Defective & Non defective	95,12	88,89	92,86	92,21	84,41

4. Discussion

The study proposed an artificial intelligence-based ESA model to prevent train accidents. The fact that this study is designed with open source codes and end-to-end makes the original aspect of the study. In the proposed approach, the analyzes were performed using TPU, which resulted in an increase in performance speed and time saving. Moreover, the advantage of the proposed approach is that it provides a decision support system that avoids wasting time in error detection and provides instantaneous results for multiple images. The disadvantages of this study are that no preprocessing steps are applied to the input images, the number of images in the dataset is small, and the hardware features are only used for a limited time (limit of free use of Google Colab, etc.). The analyzes of this study were compared with similar studies in the literature and the results of the analysis are shown in Table 3.

Table 3. Comparison of this study with similar studies in the literatüre

Article	Year	Model	Accuracy (%)
Ya-Wen Lin vd. [7]	2019	Yolo-V3	89
Hovad vd. [9]	2021	Yolo-V3	84
Rajagopa vd. [11]	2018	GLCM, LBP, ESA	94,9
This study	2022	Designed ESA	92,21

Ya-Wen Lin et al. and Hovad et al. used the Yolo-V3 model in their work. It was found that the Yolo-V3 model used in both studies limited success. Preprocessing steps (image enhancement techniques, feature extraction methods, etc.) could be preferred to increase the success of the analysis results. Indeed, Rajagopa et al. observed that they were able to increase the overall success thanks to the preprocessing steps (GLCM, LBP) they used in their studies.

5. Conclusion

Troubleshooting railroad lines is a difficult and time-consuming process. Troubleshooting with conventional methods prolongs this process. With models based on artificial intelligence, the detection of defective rails on railroad lines can be easier and faster than with conventional methods. Defects that occur on rails account for the largest share of train accidents. To prevent train accidents, an artificial intelligence-based ESA model is proposed in this study. In the study of the analysis results obtained by the proposed approach, the success in sensitivity was 95.12%, in specificity 88.89%, in f-score 92.86% and in Matthews 84.41%. The approach proposed in the next study is developed using preprocessing steps (noise removal, segmentation, resolution enhancement techniques). The features obtained from the developed ESA model are transferred to the classification process using feature selection techniques. Thus, they contribute to the performance of the model. Moreover, the analysis of the ESA model to be developed is performed using freely available datasets.

References

[1] G. Sarang, "Replacement Of Stabilizers By Recycling Plastic In Asphalt Concrete," in Use of Recycled Plastics in Eco-efficient Concrete, Elsevier, 2019, pp. 307–325.

[2] T. Deniz, "Türkiye'de Ulaşım Sektöründe Yaşanan Değişimler Ve Mevcut Durum," Doğu Coğrafya Derg., vol. 21, no. 36, p. 135, Aug. 2016.
[3] A. Welankiwar, S. Sherekar, A. P. Bhagat, and P. A. Khodke, "Fault Detection in Railway Tracks Using Artificial Neural Networks," in 2018 International Conference on Research in Intelligent and Computing in Engineering (RICE), 2018, pp. 1–5.
[4] A. James et al., "TrackNet - A Deep Learning Based Fault Detection for Railway Track Inspection," in 2018 International Conference on Intelligent Rail Transportation (ICIRT), 2018, pp. 1–5.
[5] R. Shafique et al., "A Novel Approach to Railway Track Faults Detection Using Acoustic Analysis," Sensors, vol. 21, no. 18, p. 6221, Sep. 2021.
[6] X. Wei, Z. Yang, Y. Liu, D. Wei, L. Jia, and Y. Li, "Railway Track Fastener Defect Detection Based on Image Processing and Deep Learning Techniques: A Comparative Study," Eng. Appl. Artif. Intell., vol. 80, pp. 66–81, 2019.
[7] Y.-W. Lin, C.-C. Hsieh, W.-H. Huang, S.-L. Hsieh, and W.-H. Hung, "Railway Track Fasteners Fault Detection using Deep Learning," in 2019 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE), 2019, pp. 187–190.
[8] C. Yang, Y. Sun, C. Ladubec, and Y. Liu, "Developing Machine Learning-Based Models for Railway Inspection," Appl. Sci., vol. 11, no. 1, p. 13, Dec. 2020.
[9] E. Hovad et al., "Deep Learning for Automatic Railway Maintenance," 2021, pp. 207–228.
[10] M. Sysyn, U. Gerber, O. Nabochenko, D. Gruen, and F. Kluge, "Prediction of Rail Contact Fatigue on Crossings Using Image Processing and Machine Learning Methods," Urban Rail Transit, vol. 5, no. 2, pp. 123–132, 2019.
[11] M. Rajagopal, M. Balasubramanian, S. Palanivel, "An Efficient Framework to Detect Cracks in Rail Tracks Using Neural Network Classifier," Computacion y Sistemas, vol. 22, no. 3, pp. 943–952, 2018.
[12] S. I. Eunus, "Railway Track Fault Detection," Kaggle, 2021. [Online]. Available: https://www.kaggle.com/salmaneunus/railway-track-fault-detection. [Accessed: 28-Nov-2021].
[13] H. Selçuk, T. Ç. Akıncı, and Ş. S. Şeker, "Derin Evrişimli Sinir Ağı Modellerinin Açık Kaynak Kodlu Yazılım Platformlarında Tasarımının Değerlendirilmesi," İstanbul Sabahattin Zaim Üniversitesi Fen Bilim. Enstitüsü Derg., Apr. 2021.
[14] T.-C. Lu, "CNN Convolutional Layer Optimisation Based On Quantum Evolutionary Algorithm," Conn. Sci., vol. 33, no. 3, pp. 482–494, Jul. 2021.
[15] C. F. G. dos Santos, T. P. Moreira, D. Colombo, and J. P. Papa, "Does Removing Pooling Layers from Convolutional Neural Networks Improve Results?," SN Comput. Sci., vol. 1, no. 5, p. 275, Sep. 2020.
[16] P. Sperl, C.-Y. Kao, P. Chen, X. Lei, and K. Böttinger, "DLA: Dense-Layer-Analysis for Adversarial Example Detection," in 2020 IEEE European Symposium on Security and Privacy (EuroS&P), 2020, pp. 198–215.
[17] C. Garbin, X. Zhu, and O. Marques, "Dropout vs. Batch Normalization: An Empirical Study of Their Impact to Deep Learning," Multimed. Tools Appl., vol. 79, no. 19, pp. 12777–12815, 2020.
[18] S. Maharjan, A. Alsadoon, P. W. C. Prasad, T. Al-Dalain, and O. H. Alsadoon, "A novel enhanced softmax loss function for brain tumour detection using deep learning," J. Neurosci. Methods, vol. 330, p. 108520, 2020.
[19] N. P. Jouppi et al., "In-Datacenter Performance Analysis Of A Tensor Processing Unit," in 2017 ACM/IEEE 44th Annual International Symposium on Computer Architecture (ISCA), 2017, pp. 1–12.
[20] S. Khirirat, H. R. Feyzmahdavian, and M. Johansson, "Mini-Batch Gradient Descent: Faster Convergence Under Data Sparsity," in 2017 IEEE 56th Annual Conference on Decision and Control (CDC), 2017, pp. 2880–2887.
[21] X. Ying, "An Overview Of Overfitting And Its Solutions," J. Phys. Conf. Ser., vol. 1168, p. 022022, Feb. 2019.
[22] Z. Cömert, "Fusing Fine-Tuned Deep Features For Recognizing Different Tympanic Membranes," Biocybern. Biomed. Eng., vol. 40, no. 1, pp. 40–51, Jan. 2020.
[23] E. Başaran, Z. Cömert, A. Şengür, Ü. Budak, Y. Çelik, and M. Toğaçar, "Chronic Tympanic Membrane Diagnosis based on Deep Convolutional Neural Network," in 2019 4th International Conference on Computer Science and Engineering (UBMK), 2019, pp. 1–4.
[24] S. Boughorbel, F. Jarray, and M. El-Anbari, "Optimal Classifier For Imbalanced Data Using Matthews Correlation Coefficient Metric," PLoS One, vol. 12, no. 6, p. e0177678, Jun. 2017.