

Defect classification of railway fasteners using image preprocessing and a lightweight convolutional neural network

İlhan AYDIN^{1,*}, Mehmet SEVİ¹, Mehmet Umut SALUR², Erhan AKIN¹

¹Department of Computer Engineering, Faculty of Engineering, Firat University, Elazığ, Turkey

²Department of Computer Technologies, Vocational School of Technical Sciences, Gaziantep Islam Science and Technology University, Gaziantep, Turkey

Received: 08.06.2021

Accepted/Published Online: 24.10.2021

Final Version: 21.03.2022

Abstract: Railway fasteners are used to securely fix rails to sleeper blocks. Partial wear or complete loss of these components can lead to serious accidents and cause train derailments. To ensure the safety of railway transportation, computer vision and pattern recognition-based methods are increasingly used to inspect railway infrastructure. In particular, it has become an important task to detect defects in railway tracks. This is challenging since rail track images are acquired using a measuring train in varying environmental conditions, at different times of day and in poor lighting conditions, and the resulting images often have low contrast. In this study, a new method is proposed for the classification of defects on rail track fasteners. The proposed approach uses image enhancement to first filter the rail images and obtain a high contrast image. Then, the rail track and sleeper positions are determined from the high contrast image. The location of the fastener is determined by applying the line local binary pattern method and the defects of the fastener are classified using an improved lightweight convolutional neural network (LCNN) model. Features are extracted from two fully connected layers of the developed LCNN model and the feature vector is constructed by concatenating these layers. The concatenated features are processed using a number of machine learning methods and the optimum classifier is chosen. Experimental results show that Cubic SVM gives the best results with a detection accuracy rate of 99.7%.

Key words: Railways, fastener, defect detection, CNN, line local binary pattern, support vector machines

1. Introduction

Railways are structures that enable passenger and freight transportation with vehicles moving on rails. Railway transportation, which is one of the most important means of transportation used in recent years, has seen significant improvements with the development of safe and comfortable high-speed trains [1]. The health of the rails and fasteners is very important for the safe travel of trains on the rail. Environmental conditions and friction between the train wheels and the rail cause the generation of defects in the rail fasteners [2]. Worn or missing fasteners cause the connection between the rail and the ballast to loosen. This can result in collapses or serious defects occurring due to the pressure exerted during the passage of trains. Therefore, railway inspection is one of the most important safety activities to detect defects that may lead to accidents [3].

The rail industry currently uses a range of defect inspection practices to ensure safe rail transportation. Traditional inspection of railways is carried out by an experienced engineer through visual inspection along the track. Recent advances in high-resolution cameras and computer vision enable automatic inspection systems

*Correspondence: iaydin@firat.edu.tr

for railway safety to be developed. An automatic rail inspection system consists of four tasks: rail profile measurement, detection of rail surface defects, measurement of gauges, and detection of defects in fasteners. In addition to visual inspection, there are inspection methods such as ultrasonic rail flaw detection [4], and rail geometry measurement systems [5]. However, these methods are quite expensive, and inspection takes place only a few times a year. Therefore, there is a need to develop a new method that can reduce the time between fault incidence and fault detection. The development of vision-based techniques has progressed rapidly in recent years due to the availability of increased computing power and new processing techniques.

Image processing and deep learning-based techniques have been proposed for the detection of defects on the rail surface [6–8, 27]. Liu et al. [9] proposed a similarity-based deep mesh model to detect defects in fasteners. In their method, both the localization of the fastener is determined, and the classification of the fasteners is provided. Feng et al. [10] proposed a method for fastener inspection with the Hybrid Yolov4-based object detection algorithm. The proposed hybrid approach focuses on the detection of different fasteners under different lighting and scaling conditions, rather than identifying defects [10]. Bai et al. [11] proposed a faster detection and defect classification method with a modified Faster RCNN (region based convolutional neural networks) and a support vector data identification-based method. Defective and robust fasteners were taken into consideration during the detection and classification stages. A histogram-based method has been proposed to detect the connector from rail images taken with laser cameras [12]. The proposed method in [12] is based on a hybrid approach that combines the histogram peak control method and principal component analysis. He et al. [13] proposed a directional gradient histogram and a fuzzy cluster-based technique for the detection and classification of the fastener according to its defects under different lighting conditions. Their method divides the fastener component into different regions and determines the severity of the defect according to the presence of each region. Wang et al. [14] used two-dimensional kernel principal components and support vector machines (SVM) to detect lost nuts of the rail fastener. Xia et al. [15] proposed a template matching-based method to detect fastener region, and features extracted from these regions with the Haar-like method were classified with the Adaboost algorithm. However, healthy fastener models are not useful because they cannot model damaged or missed fasteners. Geometric features, Sobel edge detection, and Hough transformation were used to locate the rail fasteners [16]. Then, the obtained 2-D fastener images were transformed into a 1-D signal using the Gabor filter, and the fastener defects were classified with the multiple signal classification algorithm. Although techniques based on the Gabor filter [8] and edge detection [16, 17] are used to locate the fastener, rail, and sleeper components, these techniques are often affected by noise and lighting conditions.

A rail fastener is an important component consisting of different parts, and it is very important to localize this component and to determine the defects that occur by obtaining the local features related to this component. Recent advances in deep learning have seen significant improvements in image feature extraction and image classification results. Comparing deep learning methods with classical feature extraction methods such as SIFT, SURF, and HOG, deep learning methods have advantages in terms of accuracy and distinctive feature extraction. A multi-tasking method has been proposed for the determination of tie and fastener defects using a deep learning method [8]. After classifying different materials, three different fasteners were classified. However, in the study, no discussion was given regarding the detection rate. Liu et al. [18] proposed a segmentation-based deep neural network to generate defective and missing fasteners to eliminate class imbalance. The generated fasteners were then classified by defect type using a modified Alexnet model. Two techniques based on Dense-SIFT and convolutional neural network (CNN) were compared for defective fastener detection [19]. The results showed that the dense-SIFT-based approach gives better results than CNN.

In studies on fastener defect detection, preliminary studies to determine the location of the fastener are very limited. Defect detection with pretrained networks is not suitable for real-time operation due to the complexity of the network structures required. Existing deep learning-based object detection methods are mostly used for detection of healthy fasteners and do not consider different defect types. Therefore, determination of fastener location, computational efficiency and robustness are important problems still to be solved.

We propose a solution using new image processing and deep learning-based methods for image enhancement, fastener detection, defect detection and classification. The main contributions of the proposed solution are given below:

- A new image enhancement method for accurate detection of rail components,
- More accurate and noise insensitive position detection using the line local binary pattern (LLBP) method to detected traverses instead of finding the position of the fastener in the whole image,
- A new lightweight convolutional neural network (LCNN) model that uses fewer layers and features for classification of fastener defects,
- Integrating features in two fully connected layers and selecting the most appropriate machine learning method for classification,
- The performance of the proposed deep neural network is comparable to other pre-trained state-of-art networks.

The paper is organized as follows. In Section 2 the details of fastener detection and deep learning-based defect classification are described. Section 3 explains the experimental results and performance analysis. Finally, the conclusion is given in Section 4.

2. The proposed method for defect classification of fasteners

In this study, a novel approach has been proposed for the classification of defects in rail fasteners, consisting of four important steps. In the first step, images are collected from the railways with the help of a camera. The lighting conditions are not optimal because the rail images are usually taken from a camera mounted under the train, therefore, rail images contain a lot of noise. In the second step, an image enhancement method is applied to obtain a high contrast rail image. Moreover, noise and lighting effects are reduced in the image obtained. In the third step, the fastener component is precisely detected in the overall image which contains rails, sleepers, infrastructure, and fasteners. After determining the rail and sleeper position, the location of the fastener is obtained by applying the LLBP method to the sleeper part. In the final step, using the features obtained by an LCNN, the healthy, damaged and missing fastener types are classified. A block diagram of the proposed method is given in Figure 1.

In Figure 1, first are shown images of the rail components including rail, sleeper, and fastener, collected with a high-speed camera mounted under the train. Histogram equalization is applied to the captured images in the filtering step. With this process, the most common pixel values are spread over the full grayscale range, thereby expanding the intensity range of the image. Next, an image enhancement process is performed on the image of the rail, and the noise is removed. In the third step, the positions of the rail and sleepers are determined and the LLBP algorithm is applied to the images determined to contain sleeper blocks. Although the LLBP algorithm reveals fastener components, a morphological operator is applied to close any holes. In the

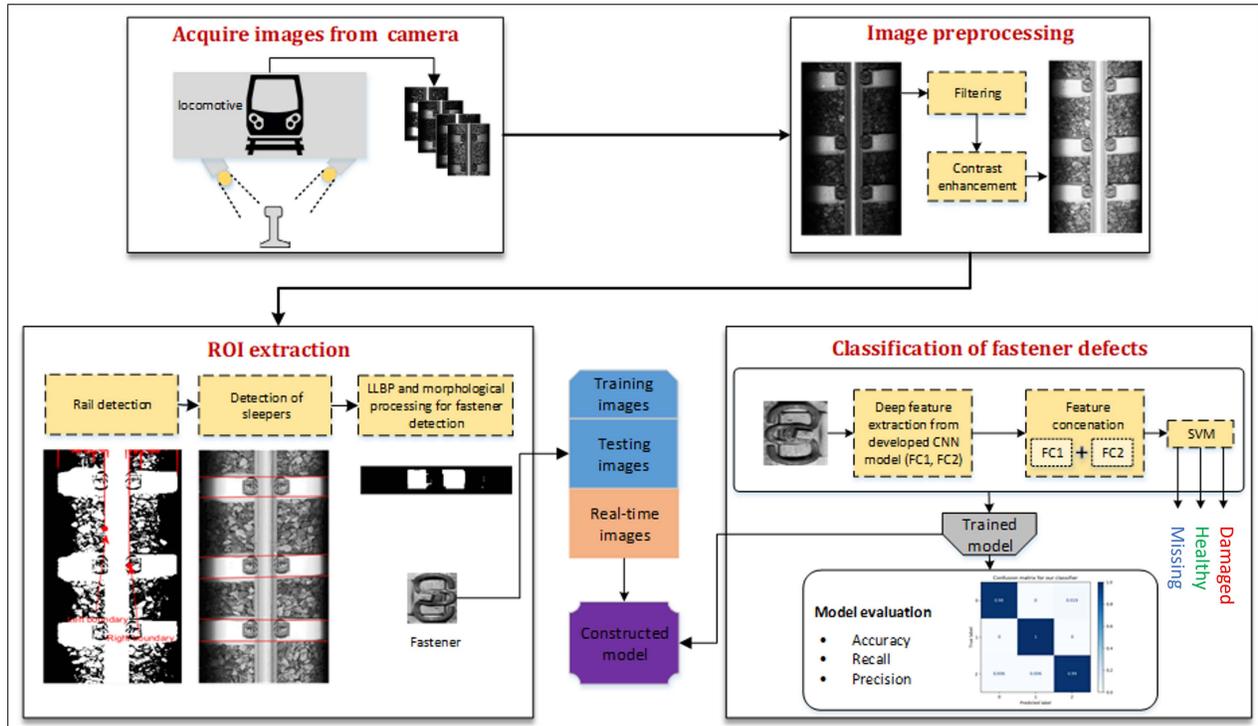


Figure 1. The block scheme of the proposed system.

final step, the located fastener components are given to the newly developed LCNN, and the features obtained are classified using SVM.

2.1. Image preprocessing and enhancement

A color image consists of three components, Red (R), Green (G), and Blue (B). Each of these matrices holds the intensity values associated with the image. The value of the pixel at any (x, y) position in the image is represented by the values in $R(x, y)$, $G(x, y)$ and $B(x, y)$. Generally, three coefficients are needed when converting a color image to a gray scale image using the following transformation:

$$I(x, y) = k_r R(x, y) + k_b B(x, y) + k_g G(x, y) \tag{1}$$

The values of these coefficients k_r , k_g , and k_b are usually selected as $k_r = 0.2989$, $k_g = 0.5870$ and $k_b = 0.1140$.

Since the contrast value of the images obtained from the bottom of the train is low, an image enhancement process is applied as the first step. Appropriate coefficients should be determined to obtain an image with high contrast and low homogeneity [13]. In Eq. (1), the value of k_b can be chosen as 1 without losing this generalization. Small details are lost in the high-contrast image obtained, however, this provides easier detection of rail and sleeper components. To obtain an image with low homogeneity, a monochrome image with a high variance should be created. At first, the pixel values of the image are normalized to be between 0 and 1 according to Eq. (2).

$$J(x, y) = (I(x, y) - I_{min}) / (I_{max} - I_{min}) \tag{2}$$

In Eq. (2), I_{max} and I_{min} represent the maximum and minimum values of the monochrome image, respectively. In order to increase the contrast value of the image, its homogeneity should be low. Therefore, it is necessary to find the k_r and k_g values that will maximize the standard deviation of the image [20]. This objective function is given in Eq. (3).

$$(\hat{k}_r, \hat{k}_g) = \arg \max_{(k_r, k_g)} (\sigma_j^2) \quad (3)$$

The Nelder-Mead simplex method can be used to maximize the standard deviation given in Eq. (3). For our purpose, the values of k_r and k_g must be found that minimize $-\sigma_j^2$. The Nelder-Mead method is a nonlinear unlimited optimization method and is implemented in Matlab through the *fminsearch* function. For the *fminsearch* function, the initial values of k_r and k_g are taken as 1.

The high contrast image consists of values between 0 and 1. To get the inverse of this image, it is necessary to subtract all pixel values from the value 1. As a result of this process, contrast is improved and an image with reduced lighting problems is obtained, enabling us to detect the rail components more easily for segmentation in the next step.

2.2. Region of interest (ROI) extraction and fastener detection

After the contrast-enhanced image is obtained, the rail and sleeper components must be detected. For this purpose, the obtained image is converted into a binary image by determining a threshold point with the Otsu method [21]. Then, the rail position is determined using the horizontal projection method. A rail extraction algorithm shown in Algorithm 1 is applied to detect the left and right rail.

Algorithm 1 Rail extraction algorithm

```

function[lefboundary,rightboundary]=railextraction(S)
[w,h] ← size(S) // w: width of image, h: height of image
p ← zeros(1,h)
for i ← 1 to h do
    for j ← 1 to w do
        P(i) ← P(i) + s(i,j)
    end for
end for
DVP ← zeros(1,h-1)
for i ← 2 to h do
    DVP(i-1) ← P(i) - P(i-1)
end for
[lefboundary,rightboundary] ← first two max values of DVP
    
```

In Algorithm 1, $S(i, j)$ represents the pixel value of the segmented image at position (x, y) . The segmented image is a binary image consisting of zeros or ones. The algorithm first calculates the histogram of the average intensity of each column in the binary rail image in vector P . Afterwards, the difference between consecutive columns in the histograms of this vector is calculated and stored in the vector DVP . Since there is only one rail in the image, the histogram differences between the starting column of the rail and the previous column and the ending column of the rail and the next column will be larger than the other differences. Therefore, these two maximum values indicate the start and end points of the rail. Detection of the sleeper component is slightly different from the rail component. This is because there is more than one sleeper in the image. The starting

and ending point of each sleeper refer to the region of interest (ROI) where the fastener can be in this region. Starting and ending positions for each sleeper are determined using a sliding window from top to bottom from a point near the left side of the rail. Illustrations of rail and sleeper localizations are given in Figure 2.

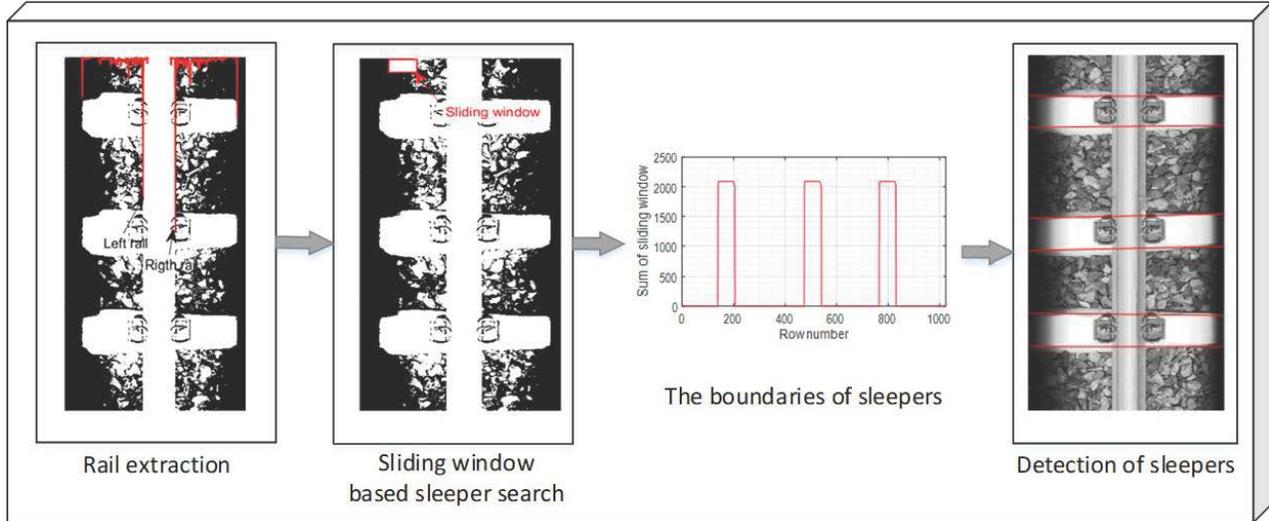


Figure 2. The illustration of rail and sleeper detection.

As shown in Figure 2, after determining the rail and sleeper components, the locations of the fasteners in the image must be determined. For this purpose, the LLBP method has been used. The original local binary pattern method produces a value by comparing the central pixel and neighboring pixels. If the central pixel value is p_c and a neighboring pixel value is p_n , if $p_n > p_c$, the corresponding pixel takes the value 0, or the value 1 if not [22]. Then the weighted sum of each bit around the center pixel gives the value of the center pixel.

It is difficult to detect the fastener in the image obtained with the local binary pattern. Therefore, the LLBP method is used in this study. The difference in the LLBP method is that it makes a comparison in the direction of the line [23]. Fan et al. [23] used the LLBP method for fastener detection. However, in [23] the LLBP method is applied to the whole image and since the image includes components such as ballasts and rails, it is difficult to detect the fasteners directly. Therefore, a separate sliding window filter is required to search for the fastener component. The size of this filter needs to be optimized. In our method, LLBP is applied only to the area where the sleeper is located, and it does not require any search process for the detection of fasteners. The mathematical calculation of this method is given in Eq. (4).

$$LLBP = sign(p_{c-n} - p_c) \times sign(p_{c+n} - p_c) \tag{4}$$

In Eq. (4), p_c and $p_{c\pm n}$ show the central pixel and its two neighbors. LLBP has only one parameter, n , which represents the neighborhood size. If this parameter is chosen to be too small, it becomes difficult to determine the position of the fastener due to noisy data. The larger the value of n , the clearer the fastener is revealed. Choosing an appropriate value depends on the size of the fastener in the image.

Figure 3 demonstrates the LLBP method and its application for different values of n . When the n value is small a very noisy image is obtained, and the noisy pixel values make it difficult to detect the fastener. On the other hand, when a high value of n is chosen, the noise decreases, and fastener detection becomes easier.

After detecting the fastener on the sleeper, the edge points are found by drawing the image contour. Figure 4 shows how fasteners are detected by the LLBP method after sleeper detection. When segmentation-based techniques are used, noisy regions can easily be detected as objects. This occurs frequently in areas of shadow on the sleeper sides. In this study, the most suitable value of n is determined by trial and error to be 10.

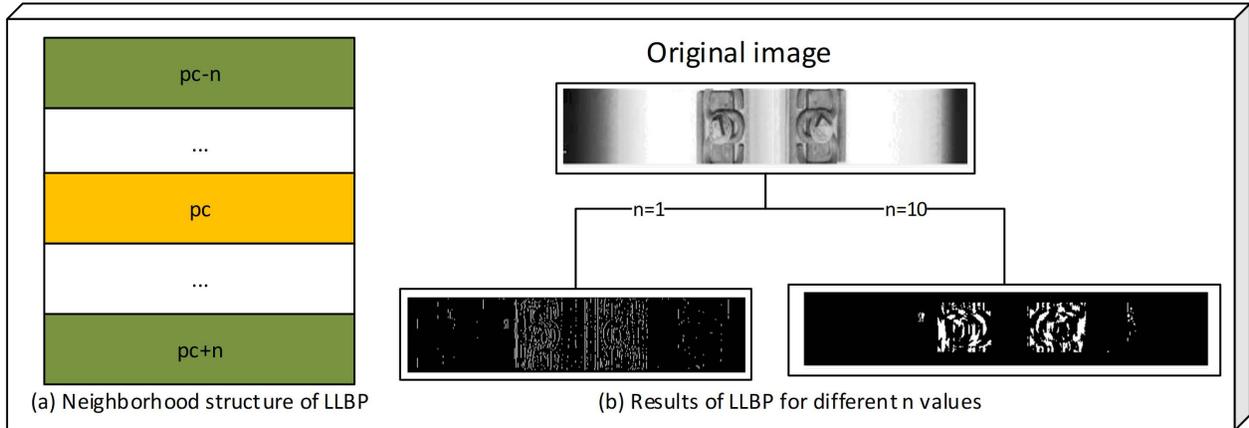


Figure 3. LLBP structure and its application for different n values.

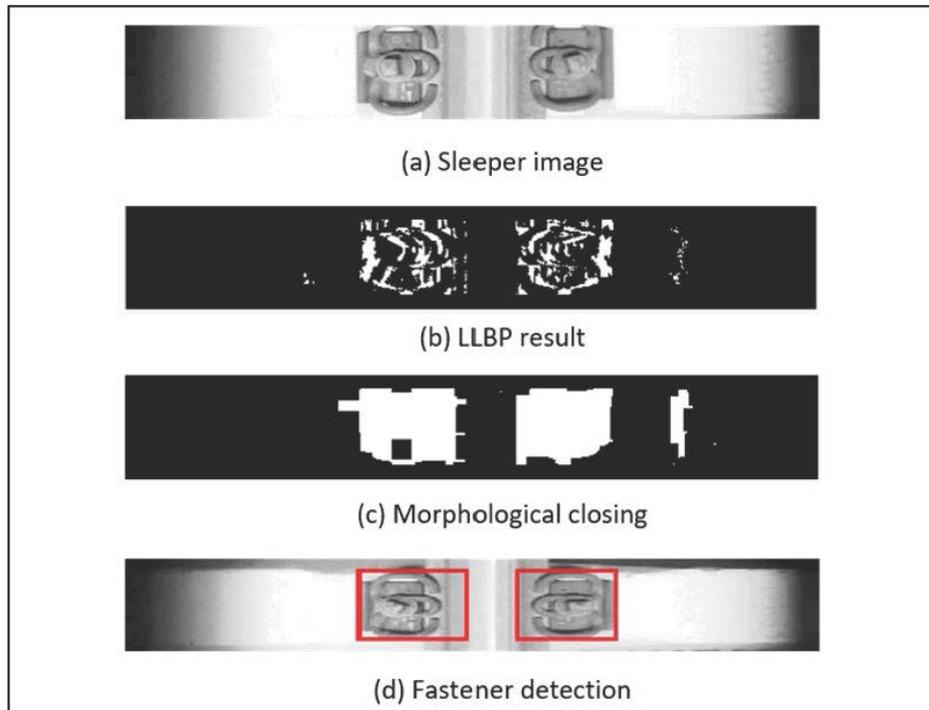


Figure 4. Fastener detection steps from a to d, respectively.

2.3. Lightweight convolutional neural network

Convolutional neural networks (CNNs) are multi-stage frameworks consisting of different layers. In this framework, each stage consists of tensors expressed in multi-channel matrices known as feature maps [24]. In image classification applications, the input layer of a CNN is usually a 2-dimensional, three-channel image. A CNN consists of three layer types. These layers are usually convolution, pooling, and fully connected layers. The convolution layer is the most fundamental element of a CNN architecture and is used for feature extraction and nonlinear processing [25]. Negative values resulting from convolution are eliminated by applying a nonlinear process to the resulting image of the weighted sum. Nonlinear operators can be implemented in different ways, namely sigmoid, hyperbolic tangent (tanh), and rectified linear units (ReLU). In this case, a pooling layer is applied after ReLU processing and is used to down-sample the image size. The maximum pooling algorithm is most commonly used for this purpose, where a feature mapped image is subsampled by taking the maximum value of each subregion.

The fully connected layer is similar to feed-forward neural networks, and several fully connected layers are placed towards the ends of a CNN. A fully connected layer gathers information from previous layers, providing a high level of abstraction. Since there are too many parameters here, it leads to an overfitting problem. That is why dropout is applied to break some of the connections. The softmax process is applied to generate class prediction scores from the outputs taken from the fully connected layer. Softmax function is given in Eq. (5).

$$\sigma_{softmax}(z^{(i)}) = \frac{e^{z^i}}{\sum_{k=1}^c e^{(z_k)^i}} \quad (5)$$

In Eq. (5), $\sigma_{softmax}$ is calculated for each input z^i . A batch normalization layer is used to reduce the training time of the CNN. This layer is also required to manage network initialization sensitivity. The normalization process is followed as defined in Eq. (6)

$$x_{normalized} = \frac{(x - \mu)}{(\sqrt{\sigma^2 + \epsilon})} \quad (6)$$

where the mini-batch mean and variance are denoted by μ and σ , respectively, and ϵ is a small threshold value chosen to avoid division by zero.

At this stage, it is intended to recognize the detailed class of fasteners whose ROI can be extracted. A deep neural network model has been proposed to achieve high classification accuracy and reduced computing time. A transfer learning algorithm is usually applied to pretrained networks to train small-sized datasets, however, these networks have many layers and complex structures. Therefore, a smaller size network model is proposed in this study. Convolution, batch normalization, ReLU, and max pooling operations are applied to the images taken from the input layer. After this process is repeated five times, 3 fully connected layers are used at the end of the network. The features obtained from two fully connected layers are combined and given to machine learning-based classifiers as input. The block diagram of the proposed method is given in Figure 5.

The CNN shown in Figure 5 has 26 layers. The number of features was determined as 300 in the first fully connected layer and 250 in the next fully connected layer. These feature numbers have been found by experiments. The number of features in the last fully connected layer is 3, indicating the number of classes. The features obtained in the first two layers are combined. Then, the fastener condition is determined by giving

it to the classification model. For classification purposes, different models are tested, and the best classifier is selected.

The performance of the proposed CNN-based defect detection system is evaluated using fastener images taken from a measurement train. The performance of the classifier is calculated using the following four standard metrics

$$Accuracy(Acc) = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{7}$$

$$Recall(R) = \frac{TP}{(TP + FN)} \tag{8}$$

$$Precision(P) = \frac{TP}{(TP + FP)} \tag{9}$$

$$F1 = \frac{2 * PxR}{(P + R)} \tag{10}$$

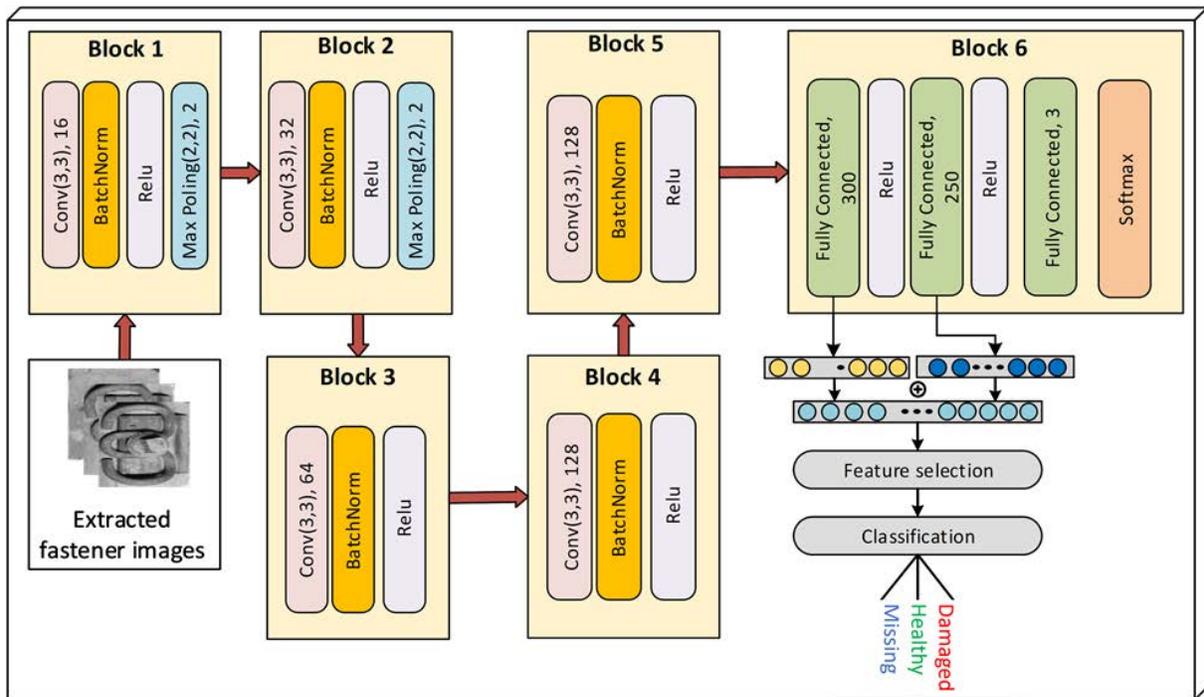


Figure 5. The architecture of the proposed LCNN model and classification.

where TP is true positive, TN is true negative, FP is false positive, and FN is false negative. A single metric is not enough to determine the performance of the classifier. Accuracy is used to measure the performance of input data over all classes. Other metrics are specific to each class and calculate the recall rate on the classification algorithm of the respective class.

3. Experimental results

A fastener defect detection system consists of a high-speed image acquisition system, an image processing and recognition system, a data management system, and a central control system.

The image acquisition system consists of high-speed cameras and illumination devices placed under a measurement train to collect images continuously. The collected images are transmitted to the central control system and recorded with GPS information. In this study we acquired the required data, at different times, using The Republic of Turkey State Railways Research Center (DATEM) experimental system mounted on a measurement train and any information about camera calibration is not provided. Figure 6 represents the measurement system from which the data is obtained.

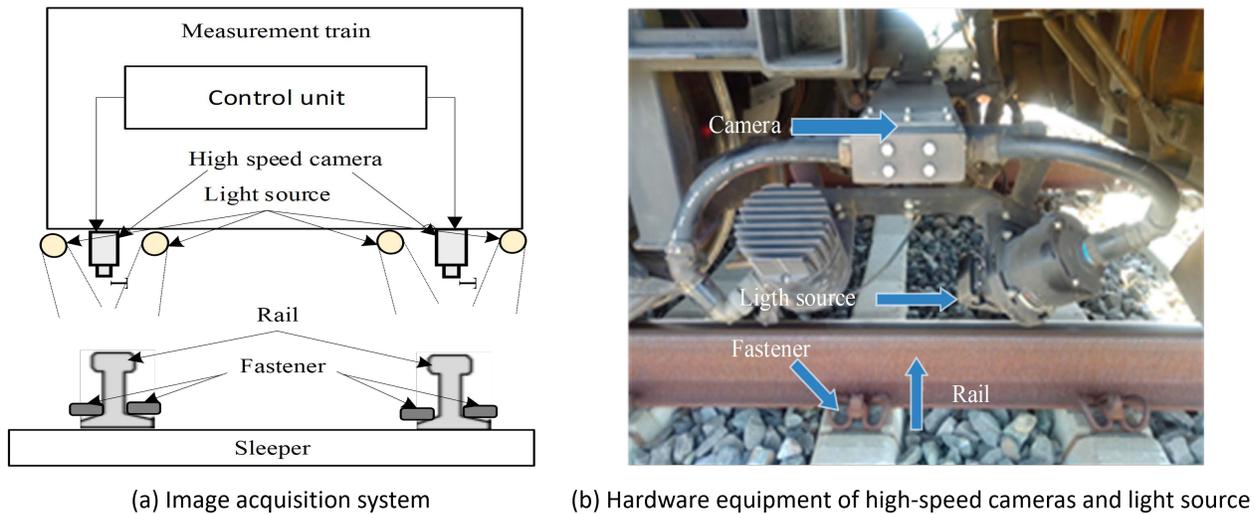


Figure 6. The structure of measurement system.

As shown in Figure 6, fastener images are collected with two line scene cameras placed at a vertical angle under the measurement train. Right and left ray images are collected with separate cameras. Each camera is equipped with two LED lights to eliminate light variations in the outdoor environment. The camera resolution is 2096 pixels, the line rate of the camera is 68,000 lines per s and the image resolution is 1 mm x 0.15 mm. There are 525 images of healthy fasteners, 594 images of damaged fasteners, and 479 images of missing fasteners in the dataset. All images are collected along the Ankara-Konya and Ankara-Eskişehir railway lines in Turkey and are manually labeled.

It is found that there are two types of fastener defect in the dataset. One type is a partially deformed fastener and the other type is a completely lost fastener. Examples are given in Figure 7 for each case.

In the fastener images shown in Figure 7 for three different situations, fastener components that are partially worn or completely missed cause more serious problems during the train transition, negatively affecting train operating safety. The images taken with the image acquisition system consist of rails, sleepers, fasteners, and ballasts. Since the purpose of this study is to determine the fastener defects, the fastener part should be extracted from the image. However, the obtained images contain noise due to the lighting conditions under the train and the contrast differences caused by the images being taken at different times of the day. Therefore, a contrast enhancement method has been applied to the original image. The initial parameters of the *fminsearch*

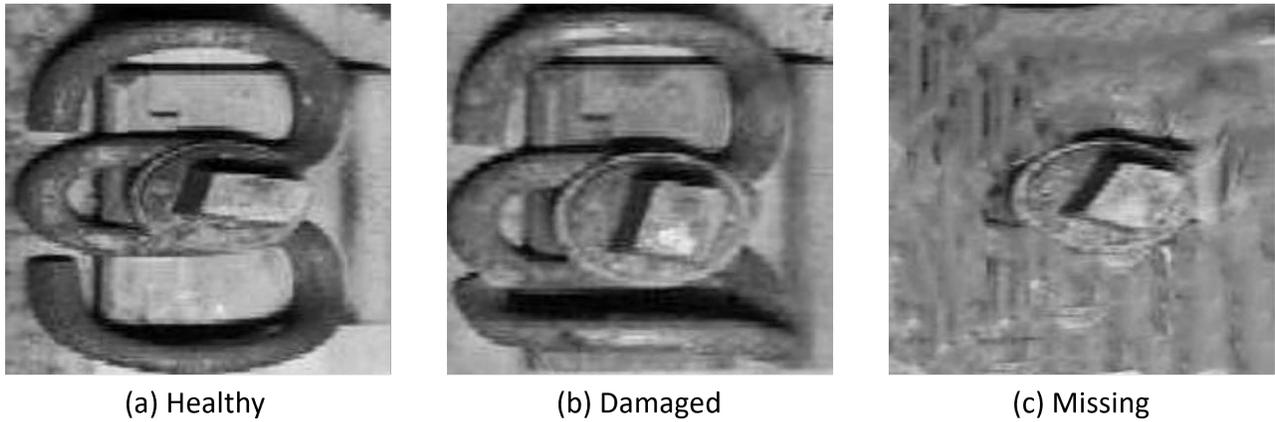


Figure 7. Three different conditions of fastener.

function used for the image enhancement stage were selected as $k_r = 1$ and $k_g = 1$.

For fastener detection, both rail and sleeper must first be detected. To obtain rail and sleeper positions a high contrast image is segmented using Otsu method; the rail position is determined and the positions of the sleepers are determined by using a filter from a point close to the left of the rail; the LLBP algorithm is applied to the sleepers obtained from the high contrast image; and finally the fastener locations are obtained from the cropped sleeper images by applying morphological processes. Morphological close was applied to determine the position of the fasteners in the binary image obtained as a result of LLBP. For this process, the structural element was chosen as a square and its neighborhood value was taken as 15.

After the fastener region is obtained from the rail image, the data set is constructed. The trained network model is obtained by giving the constructed data set to the LCNN model with the following training parameter values: initial learning rate = 10^{-4} , max epochs = 30, mini batch size = 32, validation frequency = 15 and optimization = stochastic gradient descent with momentum.

During training, both enhanced images and images taken directly from under the train without any preprocessing are used. In addition, data augmentation such as reflection and scaling is applied in both directions. The dataset is partitioned as 70% training and 30% validation. Validation performance is compared for each condition and results are given in Table 1.

Table 1. Comparative results before and after image enhancement.

Dataset	Validation accuracy (%)
Original images	91.37
Data augmentation on original images	92.58
Enhanced images	93.15
Data augmentation and enhanced images	96.82

In Table 1, it is seen that the validation results of the model training on the enhanced image are better than the original image. These results are not surprising because, in the original images, features representing the image are difficult to extract due to the effects of lighting conditions. In addition, there is an increase in performance after the image augmentation step. Model training with fully connected, softmax, and classification

layers does not provide sufficient performance. For our purpose, the features obtained from the two fully connected layers before the classification layer will be combined to select the best classification method. The output size of fully connected layers also affects network performance. The last fully connected layer represents the number of classes and it was chosen as three because there are three classes in our study. However, fully connected layer-1 (FC1) and fully connected layer-2 (FC2) must be selected correctly. In Table 2, the classification performance of the features obtained according to the sizes selected for FC1 and FC2 is given.

Table 2. The performance of proposed CNN for different fully connected layers.

Size of fully connected layers		Size of combined features	Accuracy rate of combined features
FC1	FC2		
100	50	150	93.11
150	100	250	94.15
200	150	350	95.00
250	200	450	96.61
300	250	550	99.70
350	300	650	97.20
400	350	750	97.90
450	400	850	97.90
500	450	950	97.10

In Table 2, selected features are given to the Matlab classification learner tool for classification. Here, the classification performance of the combined features according to the classifier giving the highest performance is selected. In Table 2, the best performance is obtained when 300 and 250 features are selected for FC1 and FC2, respectively. When there is not enough data to train CNNs, the features obtained from the layer of a pretrained CNN before the classification stage are given to a machine learning method, and classification is made. Since feature extraction from a pretrained network does not require further training, it can be done even on a low-specification computer. However, the dimensionality of the features obtained in these pretrained networks is quite high. In this study, fastener defect classification is made by extracting a smaller size feature. In Table 3, the number of trainable parameters in millions (M) and the feature extraction performance of the proposed CNN architecture are compared with state-of-art pretrained networks.

Table 3. The feature extraction process for proposed CNN and baseline pretrained CNN models.

The CNN model	Number of learnable layers	Number of learnable parameters	Feature layer	Number of features	Time(ms)
SqueezeNet	18	1.24M	pool10	1000	8.46
GoogleNet	22	6.7M	pool5-drop_7x7_s1	1024	18.00
VGG19	16	143M	avg_pool	2048	38.47
LCNN	8	0.9M	FC1-FC2	550	6.58

The pretrained networks given in Table 3 were chosen specifically from those with small dimensions and fewer parameters. For example, although SqueezeNet performs close to Alexnet, it runs 50 times faster.

GoogleNet's architecture has 22 layers and the model size is low compared to other CNNs. Likewise, the VGG19 model is also a network model with 16 layers and has a slightly higher accuracy rate on the Imagenet dataset. In Table 3, feature extractions are compared on 737 samples and feature extraction time is given for one image. As can be seen here, the feature extraction performance of the proposed network model is better than other models, and it seems possible to be used in real-time applications. T-distributed stochastic neighbor embedding (t-SNE), a statistical method, is used to show the distribution of features obtained from two fully connected layers of the proposed CNN model [26]. The technique shows high-dimensional data obtained from a network activation layer in a two-dimensional way. The t-SNE method preserves distances while performing the nonlinear mapping. Thus, by learning the distribution of the features obtained from a particular layer, samples showing erroneous distribution can be determined. In Figure 8, a t-SNE representation of the FC1 and FC2 layer of the proposed CNN architecture is given.

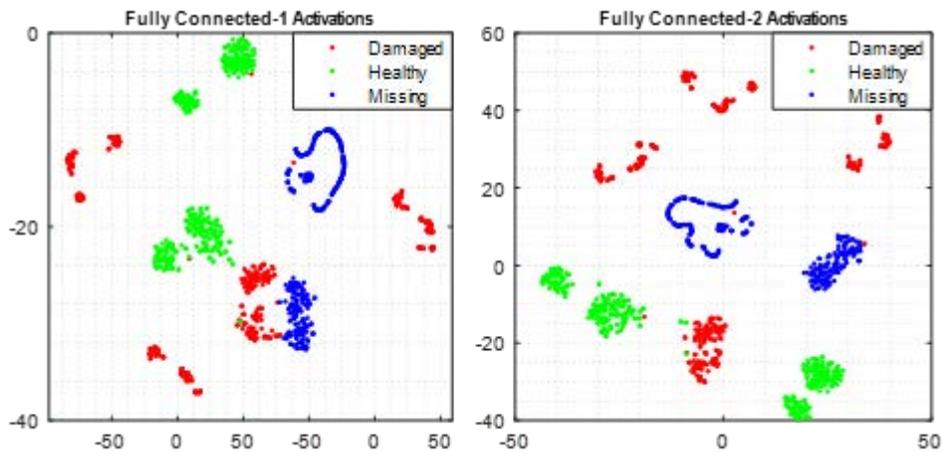


Figure 8. The t-sne representation of features extracted from FC1, and FC2.

As shown in Figure 8, it seems more difficult to distinguish between healthy and defective data in the t-SNE representation of FC1 layer. In the t-SNE representation of the FC2 layer, it is seen that some intact and deformed examples can be distinguished more clearly. The proposed method combines the features obtained from FC1 and FC2 and classifies them with machine learning methods. For this purpose, different classification methods have been used. The results of eight classification methods giving the highest classification accuracy are given. The proposed method is compared with both the evaluation of extracted features by using machine learning methods and the fully connected and softmax layers of the CNN. The parameters are the same for all pretrained networks. In addition, 5-fold cross-validation was performed at the classification stage. In 5-fold cross validation, the data set was divided into 5 parts, one part at a time for testing and the remaining 4 parts for training. The accuracy value was obtained by taking the average of the performances obtained for the test data. The comparison results of the proposed CNN model with different pretrained networks are given in Table 4.

When the results obtained from the relevant layers of the pretrained networks selected in Table 4 are compared with the results of the proposed method, it is seen that the methods proposed in seven of the nine classification methods give better results. The proposed CNN architecture uses fewer features and its performance is higher than the state-of-art pretrained networks. Cubic SVM gave the best performance among machine learning methods. The accuracy rate for Cubic SVM is 99.7%.

Table 4. Comparison results with different pretrained CNN models.

Classifiers	Parameter	Value	Accuracies of CNN Models (%)			
			SqueezeNet	GoogleNet	VGG19	Ours
Fine Tree	Number of splits	100	89.2	92.3	90.5	93.9
Linear Discriminant	Covariance	Full	86.3	89.6	88.8	92.6
Gaussian Naïve Bayes	Distribution	Gaussian	91.4	95.4	96.7	96.8
Cubic SVM	Kernel function	Cubic	96.9	97.8	98.4	99.7
Weighted KNN	Number of neighbors	10	93.3	94.4	95.6	96.2
Ensemble Subspace KNN	Number of learners	30	96.2	97.9	97.2	96.3
Ensemble Boosted Tree	Number of splits	20	50.6	53.3	60.4	51.4
Ensemble RusBoosted Trees	Number of splits	20	90.7	93.4	94.7	94.8
LCNN	Activation function	Softmax	92.4	93.7	94.8	96.8

To demonstrate the effectiveness of the proposed method, we have also compared the relevant methods in the literature. These methods consist of traditional computer vision techniques [2, 11] and deep learning-based techniques [9, 10, 19]. Fastener defect classification results with different methods are given in Table 5.

Table 5. Comparison of the proposed method with the baseline methods in the literature.

Reference	Method	Class	P	R	F1	Acc (%)	FPS
[2]	Probabilistic structure modeling	Healthy	0.97	0.97	0.97	96.50	14
		Damaged	0.83	0.80	0.81		
		Missing	0.82	0.88	0.85		
[9]	Similarity based deep learning method	Healthy	0.85	0.85	0.85	92.91	196
		Damaged	0.98	0.96	0.97		
		Missing	0.98	0.99	0.99		
[10]	Hybrid YoloV4	Healthy	0.78	1.00	0.87	94.40	78
		Damaged	0.91	0.98	0.95		
		Missing	0.82	0.89	0.85		
[12]	Template matching based nearest neighbor	Healthy	0.92	0.97	0.94	96.34	6
		Damaged	0.67	0.87	0.75		
		Missing	0.83	0.87	0.85		
[18]	Improved Alexnet	Healthy	0.97	0.92	0.95	97.52	188
		Damaged	0.96	1.00	0.98		
		Missing	1.00	0.99	0.99		
LCNN	Contrast enhancement and lightweight convolutional neural networks	Healthy	1.00	1.00	1.00	99.70	221
		Damaged	0.99	0.99	0.99		
		Missing	0.99	0.99	0.99		

In the classification results in Table 5, it is seen that the proposed model provides better performance than existing models applied to the same problem. In addition, it is seen that deep learning methods provide better performance in the classification stage. The probabilistic structure model performs defect detection by creating a model for the fastener. However, this model requires the adjustment of many parameters. The pattern matching-based nearest neighbor algorithm requires high computational power because of the large size of the features obtained from the histogram of oriented gradients. Therefore, the frame rate using this method is low. In the implementation using Hybrid Yolov4, the fastener component is labeled as a geometric object. This causes the labeling to be slow as it requires precise labeling. In addition, Yolov4 is sensitive to

lighting conditions and its performance is poor when the number of samples is distributed unevenly across the classes. The similarity-based deep learning method uses sample fastener pairs while generating defect data. However, since the defect types created are very similar to each other, only the properties of fasteners with limited defects are learned even if the number of samples is increased. For our method, the sample numbers are evenly distributed. In addition, since the problems related to lighting were solved in the preprocessing stage, the performance of the proposed deep learning method increased. During the comparison of the proposed method with the existing methods in the literature, frames per second (FPS) values were considered. Since the proposed deep learning network has fewer layers than pretrained network structures, it has a relatively high frame rate (221 FPS) in the implementation reported here.

4. Conclusions

Advanced detection of defects in train rails is a very important issue for both human safety and economic performance. Detecting defective fasteners on train rails is one of the major challenges. In this study, a novel CNN-based deep network model is proposed to solve this problem. The proposed model has four main stages: (1) images are collected from the railways, (2) image enhancement is applied to obtain high contrast rail images, (3) fastener components are precisely detected in the rail images, and (4) fastener images are classified using a lightweight convolutional neural network. The proposed LCNN architecture has two main advantages. First, it is architecturally simple and requires less computational time. Second, it shows improved classification performance compared to the major classification models in the literature. The most known machine learning algorithms are used to classify the features obtained from the proposed deep model.

A series of experiments were carried out to verify the performance of the proposed model. In these experiments, the performances of major existing deep architectures were compared with the performance of the proposed model. As a result of the experimental studies, it has been observed that the performance of the proposed model is better than the state-of-art models in the literature. The proposed method was also compared with deep learning-based fastener detection methods and better results were obtained. Deep learning-based detection methods are more focused on the detection of healthy fasteners and are less well fitted to the detection of defective fasteners. The main reason for this is that most data collected in the field contains healthy fasteners and the number of defective or missing fasteners is very limited. Therefore, future studies will aim to improve the balance of training data by using generative deep learning techniques to produce defective and missing fasteners under different illumination conditions.

Acknowledgment

This work was supported by TÜBİTAK (The Scientific and Technological Research Council of Turkey) under Grant No: 120E097.

References

- [1] Zuwen L. Overall comments on track technology of high-speed railway. *Journal of Railway Engineering Society* 2007; 1: 41-54.
- [2] Feng H, Jiang Z, Xie F, Yang P, Shi J et al. Automatic fastener classification and defect detection in vision-based railway inspection systems. *IEEE Transactions on Instrumentation and Measurement* 2013; 63 (4): 877-888.
- [3] Peng Z, Wang C, Ma Z, Liu H. A multi-feature hierarchical locating algorithm for hexagon nut of railway fasteners. *IEEE Transactions on Instrumentation and Measurement* 2019; 69 (3): 693-699.

- [4] Utrata D, Clark R. Groundwork for rail flaw detection using ultrasonic phased array inspection. In AIP Conference Proceedings 2003; 657 (1): 799-805.
- [5] Chen Q, Niu X, Zuo L, Zhang T, Xiao F et al. A railway track geometry measuring trolley system based on aided INS. *Sensors* 2018; 18 (2): 538.
- [6] Gan J, Wang J, Yu H, Li Q, Shi Z. Online rail surface inspection utilizing spatial consistency and continuity. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 2018; 50 (7): 2741-2751.
- [7] Zhang H, Jin X, Wu QJ, Wang Y, He Z et al. Automatic visual detection system of railway surface defects with curvature filter and improved gaussian mixture model. *IEEE Transactions on Instrumentation and Measurement* 2018; 67 (7): 1593-1608.
- [8] Gibert X, Patel VM, Chellappa R. Deep multitask learning for railway track inspection. *IEEE Transactions on Intelligent Transportation Systems* 2016; 18 (1): 153-164.
- [9] Liu J, Huang Y, Zou Q, Tian M, Wang S et al. Learning visual similarity for inspecting defective railway fasteners. *IEEE Sensors Journal* 2019; 19 (16): 6844-6857.
- [10] Guo F, Qian Y, Shi Y. Real-time railroad track components inspection based on the improved YOLOv4 framework. *Automation in Construction* 2021; 125: 103596.
- [11] Bai T, Yang J, Xu G, Yao D. An optimized railway fastener detection method based on modified Faster R-CNN. *Measurement* 2021; 182. doi: 10.1016/j.measurement.2021.109742.
- [12] Dou Y, Huang Y, Li Q, Luo S. A fast template matching-based algorithm for railway bolts detection. *International Journal of Machine Learning and Cybernetics* 2014; 5 (6): 835-844.
- [13] He B, Luo J, Ou Y, Xiong Y, Li B. Railway fastener defects detection under various illumination conditions using fuzzy c-means part model. *Transportation Research Record* 2020; 2675 (4): 271-280.
- [14] Wang L, Zhang B, Wu J, Xu H, Chen X et al. Computer vision system for detecting the loss of rail fastening nuts based on kernel two-dimensional principal component–two-dimensional principal component analysis and a support vector machine. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit* 2016; 230 (8): 1842-1850.
- [15] Xia Y, Xie F, Jiang Z. Broken railway fastener detection based on adaboost algorithm. In 2010 International Conference on Optoelectronics and Image Processing 2010; 1: 313-316.
- [16] Ni X, Liu H, Ma Z, Wang C, Liu J. Detection for rail surface defects via partitioned edge feature. *IEEE Transactions on Intelligent Transportation Systems* 2021; 1-17. doi: 10.1109/TITS.2021.3058635.
- [17] Karaduman G, Karaköse M, Aydın İ, Akın E. Contactless rail profile measurement and rail fault diagnosis approach using featured pixel counting. *Intelligent Automation and Soft Computing* 2020; 26 (3): 455-463.
- [18] Liu J, Teng Y, Ni X, Liu H. A fastener inspection method based on defective sample generation and deep convolutional neural network. *IEEE Sensors Journal* 2021; 21 (10): 12179-12188.
- [19] Wei X, Yang Z, Liu Y, Wei D, Jia L et al. Railway track fastener defect detection based on image processing and deep learning techniques: a comparative study. *Engineering Applications of Artificial Intelligence* 2019; 80: 66-81.
- [20] Mery D, Pedreschi F. Segmentation of colour food images using a robust algorithm. *Journal of Food Engineering* 2005; 66 (3): 353-360.
- [21] Otsu N. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics* 1979; 9 (1): 62-66.
- [22] Ojala T, Pietikainen M, Maenpää T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2002; 24 (7): 971-987.
- [23] Fan H, Cosman PC, Hou Y, Li B. High-speed railway fastener detection based on a line local binary pattern. *IEEE Signal Processing Letters* 2018; 25 (6): 788-792.

- [24] Singh N, Sabrol H. Convolutional neural networks-an extensive arena of deep learning. A comprehensive study. Archives of Computational Methods in Engineering 2021; 1-26. doi: 10.1007/s11831-021-09551-4
- [25] Mahalakshmi P, Fatima NS. Ensembling of text and images using deep convolutional neural networks for intelligent information retrieval. Wireless Personal Communications 2021; 1-19. doi: 10.1007/s11277-021-08211-x
- [26] Van der Maaten L, Hinton G. Visualizing data using t-SNE. Journal of Machine Learning Research 2008; 9 (86): 2579-2605.
- [27] Santur Y, Karaköse M, Akin E. An adaptive fault diagnosis approach using pipeline implementation for railway inspection. Turkish Journal of Electrical Engineering & Computer Sciences 2018; 26 (2): 987-998.