Railway Track Fault Detection

Mathemathical Modeling Practice

Tamás DEMUS
XP4B9D

Fall Semester 2022

Contents

1 Introduction

Analyzing images and processing the information stored within is a key field of machine learning. Classification of different images, identifying and localizing different objects are basic problems. Several real-life cases prove the usability of such approach such as traffic sign recognition, object detection or face recognition. The target of current research is to endeavour to create an algorithm that is able to classify images taken from parts of the rail track to classify whether the rail is defect or not.

A comprehensive overview about visual inspection technologies used in the railway sector is given in [1]. It focuses on different technologies including machine learning and non machine learning applications and covers the overall railway sector including track, vehicle and infrastructure elements. The following major application categories identified: the railway track, the pantograph-catenary subsytem, the train body and rail infrastructure. The railway track is discussed in several subpoints: the detection of rail surface defects, wear and deformation of track components, identification of rail components and the detection and extraction of rails. Current summary follows this review focusing on the track inspection methods applying machine learning algorithms based on image processing.

1.1 Surface defect detection

Xue Wang et al. presents a rail surface defect detection method [2] based on a liner CCD followed by image processing and applying a Quantum Neural Network (QNN). The accuracy results were evaluated and compared between QNN, Artifical Neural Network (ANN) and Support Vector Machine (SVM).

First it detects region of interests (ROI) in the captured images by selecting pixels where the gray-level quantity is above a certain threshold. Then it searches for edges to identify the pixels that belong to certain edges. Then comparing this two set of points, a region of interest is defined by the presence of a set of pixels in both sets. Once this are marked, a rectangle can be drawn to enclose the region. Next step if to extract the features from the single ROIs. The features are defined based on the geometry of the flaw inside the ROI, the grey-level of the flaw pixels, and a so-called tansform-space, where a discrete cosine transform filter is applied. The obtained recalls are ranging between 86.7% and 100% depending on the type of flaw.

Deep convolutional neural network (DCNN) was proposed in [3] to detect surface defects of rails. Video recordings of rail tracks are processed by DCNNs with different sizes related to the size and number of the filters, number of convolutional layers and the size of the fully connected layers. Hyperbolic tangent and ReLU activation functions used. The applied dataset consits of more than 22.000 manually labelled images of 6 classes (normal surface, weld, light, moderate or severe squat and joint defects). Input image size is 100x50 pixels and converted to grayscale. The multi class accuracy is ranging between 91.17% and 92.47%. The performance increases by increasing the size of the DCNN and by applying ReLU activation functions.

Another solution for rail surface defect detection is described in [4]. Images taken by maintenance vehicles from the top and evaluated for surface defects via two SVM classifiers applied one after the another. Overall 8 defect categories differentiated including the non-defect case. 939 manually annotated images was used for the experiments. A Random Forest based edge detection is followed by a Generalized Hough transform to detect the position of the rails. This overperformed the Canny edge detection method. Defect severity level classification was traced back to a texture classification task for that a Bag of Words model is used. A set of filters are used to create the feature vectors for each pixel in each image. From this the dictionary is created via a k-means clustering resulting in a histogram of k bins containing the number of pixels in each bin. These descriptors were fed for a χ^2-kernel SVM for training. Besides the dictionary texton forests were built. Instead of predicting the class, texton forests predict the image descriptors. The two methods were combined to an ensemble method by applying stacking. Overall 5 texton forest SVM classifier and 4 texton dictionary SVM classifier provides 9 probability vector with 8 elements for the 8 classes. A second level linear-kernel SVM was fitted to these outputs. An accuracy of 82% is achieved in the end.

Santur et al. presents a laser based imaging technology combined with deep learning model [5]. The rail profile is aquired via laser scanning and afterwards a convolutional neural network is applied. The article distinguishes between 4 defects: cracks, abrasion, corrugation and headchecks, however only binary classification is implemented (faulty or not faulty image). The rail imaging is done via stereovision, time of flight and laser triangulation to obtain a 3d image. Convolutional Neural Network (CNN) was applied on the image. The mentioned system achieved 98% accuracy.

Li et al. introduces a corrugation identification system that consists of an image acquisition system and the corresponding machine learning algorithm [6]. The images are taken from the top by equipment fitted to the underframe of the vehicle. The image processing covers the localization of the rail, feature extraction and corrugation recognition steps. Rail localization is done on the observation that the rail brightness is high and even and mostly located on the center of the picture. Features extracted by applying Fourier transformation for each column (lengthwise part of the rail surface) to determine the Fourier energy spectrum after truncation and sampling to reduce the dimensionality. Observation taken that the longitudinal average gray value is relatively high and distributed uniformly. Corrugation lines represented as periodic changes of the gray value that is a sparse energy distribution in the frequency domain. Corrugation afterwards can be detected by an SVM trained on these features. The proposed solution reaches 99% accuracy.

Another model is described in [7] for surface defect detection. A CNN is applied after preprocessing of the images. Canny edge detection is applied to the grayscaled images, that is followed by removing the false edge points. The point removal is based on the dynamic range of the edgepoints that is adaptively narrowed to obtain a set of points with low standard deviation. Once the false points removed, linear fitting performed to obtain the coefficients of the rail edge linears. Inception-v3 CNN was applied with transfer learning. More than 5000 images were used for the training, more than 2000 for the validation and around 1500 for the final test. The image size for the CNN is 960x1280 pixels. The model results in 92.54% recall rate and 92.08% precision.

1.2 Component detection

Li et al. presents a methodology in his work in [8] extended in [9] and [10] to detect missing rail components focusing on the tie plate and it's surrounding localization. A fixed position camera is used in the research that allows a steady assessment of the track components. Mostly relying on edge detection methods on grayscale images, at first detecting the tie plate itself and then the other components can be identified based on the geometrical positioning.

Connection element binary classification is presented in [11] that is explained more in detail in [12]. The method is able to detect the presence of fastener bolts or hooks. Fastening elements are extracted as features from grayscale pictures taken from top view. Wavelet transformation and principal component analysis (PCA) performed to describe the features in low dimensionality. During the wavelet transformation a high-pass and low-pass filter is applied in every combination, extended by applying the same combinations on the low-pass-low-pass result. This results in a multilevel (depending on the number of extensions) wavelet transform. Two type of neural networks considered: Multilayer Perceptron and Radial Basis Function. A training set of 173 images used for the bolt detection and 221 images for the hook detection. Depending on the object to detect the wavelet transformation with levels of 3, 4, and 5 performed best, reaching accuracy rates of above 98% in almost all cases.

A commercial solution is presented in [13] utilizing GPU based hardware. It consists of an integrated image acquisition system followed by a prediction modul to identify image areas for classification and a bolt detection module. The bolt detection module uses Multilayer Perceptual Network Classifiers (MLPNC) fed by features generated through wavelet transformation. Training dataset consists of approx. 1000 images of 24x100 pixels window, that are extracted from recorded videos. The MLPNCs have a structure of 150:10:1 layers. The results of different MLPNCs combined with and AND operator. The prediction and detection modules are built on a GPU based hardware for fast execution. An accuracy rate of 99.9%, 0.1% and 95% was reached for visible, occluded and absent bolts respectively.

Another component detection is introduced in [14]. Top view rail images used for detecting the fastening elements taken by the image acquisition system taken in a way to minimise surroundings of the ROIs. The alogrithm applies preprocessing, grayscale conversion, feature point detection, feature extraction and feature matching to detect rail anchors. The images are resized to 200x200 and then cropped to 170x67 to minimise the noise caused by the surroundings. Harris-Stephen and Shi-Tomasi feature detectors were applied. The features of the train and test images were matched. If the number of matchings is above a certain threshold, than presence of a rail anchor is assumed. The method achieved an average success rate of 83.55%.

1.3 Further detection methods

In his study Kumar M. et al. compares crack detection methods that are based on different measurement solutions and image processing algorithms [15]. The preprocessing steps are explained in detail. They refer to conversion grayscale images as main apprach. Noise reduction and the sharpening of the picture is taken as first step. For further evaluation he refers to several edge detection techinques that might be applied.

1.4 Summary

no joint method, different preprocessing: grayscale, edge detection, rail position, or special feature extraction (bag of words, wavelet transform, etc.), image size vary, training sets vary Common: known shape, geometrical position and camera position, grayscale

In Section 2 the dataset is introduced without any detailed analysis to give a first glance where the study is started. Section 3 the main research questions stated. Following in Section 4 a detailed dexription of the applied methodology and toolkits introduced. Section 5 presents the results of the study divided according to the main steps of the algorithm. In Section 6 the results are reflected to the research questions, hopefully leading to positive outcomes. Finally Section 7 concludes in the overall achievements.

2 Dataset description

The dataset used for this study is taken from Kaggle webpage [16] and can be downloaded directly from https://www.kaggle.com/datasets/salmaneunus/railway-track-fault-detection [17]. The

dataset is stored in different directories related to their purpose: Train, Validation, or Test dataset. Inside each directory the classes also splitted to separate directories: Defective or Non defective. The directory structure along with the number of images can be seen in Table 1. The images are taken from

Folder	Number of images
./Train/Defective	150
./Train/Non defective	150
./Validation/Defective	31
./Validation/Non defective	31
./Test/Defective	11
./Test/Non defective	11

Table 1: Dataset directory structure

different perspectives, either from the top or from the side or from any other direction or angle. The photos show different track sections, that could be a close view on the single rail or a full picture taken from the entire rail section. The quality of the dataset is spreading between different formats, mostly consisting of jpg files. The size of the images are different ranging from very low to very high resolutions. The dataset contains only color pictures. Some examples are shown in Figure 1 and Figure 2. The total size of the dataset is 2.14 GB.

(a) Non defective (b) Non defective (c) Non defective

Figure 1: Example images of non defective track

(a) Defective (b) Defective (c) Defective

Figure 2: Example images of defective track

3 Problem statement

The algorithm targets to identify from a picture whether it represents a defective or a non defective track section. For this purpose image manipulation technics together with machine learning algorithms, in more detail neural networks applied. The problem formulation defines the main questions of the research to be answered:

Q1 What kind of defects are represented in the images?

Q2 Can these defects detected by applying image manipulation and machine learning approach?

Q3 What accuracy rate can be achieved with the algorithm?

4 Methodology

The research can be divided to several major steps, these are described in the subsequent sections. The algorithm is developed in Python programming language in a form of a Jupyter notebook. In current research the aim was to store the different stages of the processing in different file folders to allow single analytics of each processing step. In each folder the same original structure is retained. Therefore the following data folders are differentiated.

1. raw Contains the raw dataset directly copied from Kaggle.
2. data The images after data cleaning in standard format and naming.
3. augmented Additional images as result of data augmentation.
4. preprocessed The data after image processing, preprocessed for the machine learning algorithm. Includes both the original and augmented dataset.

4.1 Data cleaning

The images should be brought to the same quality in order to efficiently process them. This covers the identification of corrupted files, correct the wrong file formats and the resolution of all issues that prevents the processing of the data along the same pipeline. The result of this step should be a data structure that is easy to handle and process further. This includes the following information.

1. Type of image, whether it is intended for training, validation or testing.
2. Indicator of defective or non-defective class, both as integer and as string.
3. Path of the image directory, filename and their joints as full path.

4.2 Data exploration

The target of this step is to get a deep understanding of the pictures contentwise. The main properties, such as size, color composition, orientation, what is shown and similar need to be checked with respect to characteristic differences between the distinct classes. The size and the mean of the color components is extracted from the images and stored in the dataframe of the images. The distribution and correlation of the images color components considering the mean values analyzed. A random sample of the images is taken for visualization to study the different defects of the defective class. As a final step, the color components were investigated in detail considering the histogram of the component values taken for each pixels. This is investigated on the RGB color model and the images were converted to HSV model to get a more comprehensive view.

5 Results

5.1 Data cleaning

The raw data from the Kaggle webpage [17] is copied to the folder raw. This folder is not changed during the data processing, it is used as a starting point and to preserve the original data. During data cleaning three different file formats detected, namely webp, jpeg and jpg. The filenames show a much wider spread including random names, regular photo labeling (e.g.: DCIM) and others. As Tensorflow is not able to work with webp files and to maintain the same standard along the dataset, all files that are not in jpg format are converted to jpg. During this conversion all the files were renamed applying an integer counter. The modified dataset is then stored in the data folder in the same structure as in the original data.

Second step is to sum all the main information of each image in a single Python DataFrame to allow easy processing.

5.2 Data exploration

The distribution of the image sizes are shown in Figure 3. Please note that higher detail images can be found in the corresponding Jupyter notebook, the aim in this document is to explain the evaluation approach and summarize the results. Non defective pictures have only 2 shapes, either 3000x4000 or 4000x3000, depending whether it has portrait or landscape orientation. Defective images have wider distribution in the shapes reaching as high as 8000 pixels and as low as 148 pixels. The minimum image size in this case is 148x194 (height x width) and the maximum is 8000x6000. All images have 3 color components according to RGB color model.

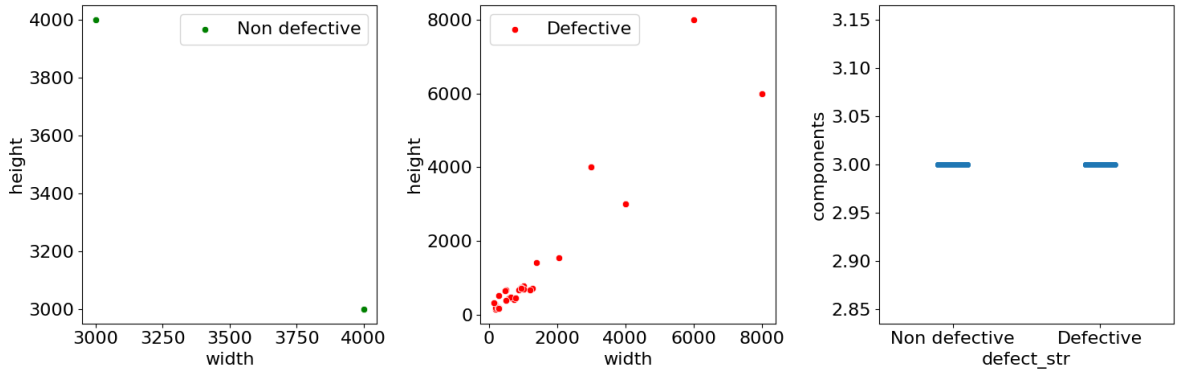

Figure 3: Distribution of the image shapes

The distribution and correlation of the mean of the color components with respect to the images is shown on Figure 4. The different color components show comparable histograms. The case of the green and blue component the non defective components show a more narrow distribution. The paired color components in both classes represent similar feature spaces.

During the study of the random samples, the following defects have been identified.

1. Rail cracks. (Figure 5a)
2. Disjoint rail connections. (Figure 5b)
3. Pitting of the rail surface. (Figure 5c)
4. Defect fastener. (Figure 5d)
5. Missing elements, e.g.: screws, springs. (Figure 5e)

The magnitude of each defect is varying in a wide range. For example in case of cracked rails, from a crack of a few millimeters up to a missing rail part of several centimeters can be found. Similarly, surface pitting can range from small cracks on the surface up to severe cases, like shown in Figure 5c.

The color component analysis on RGB and HSV colormodes revealed no exact differentiation between the classes. The component values highly depend on what is represented on the picture and how the photo is taken. For example, presence of shadow has a significant impact on the component histogram. An example of the analysis is shown on Figure 6.

6 Discussion

7 Conclusion

List of Figures

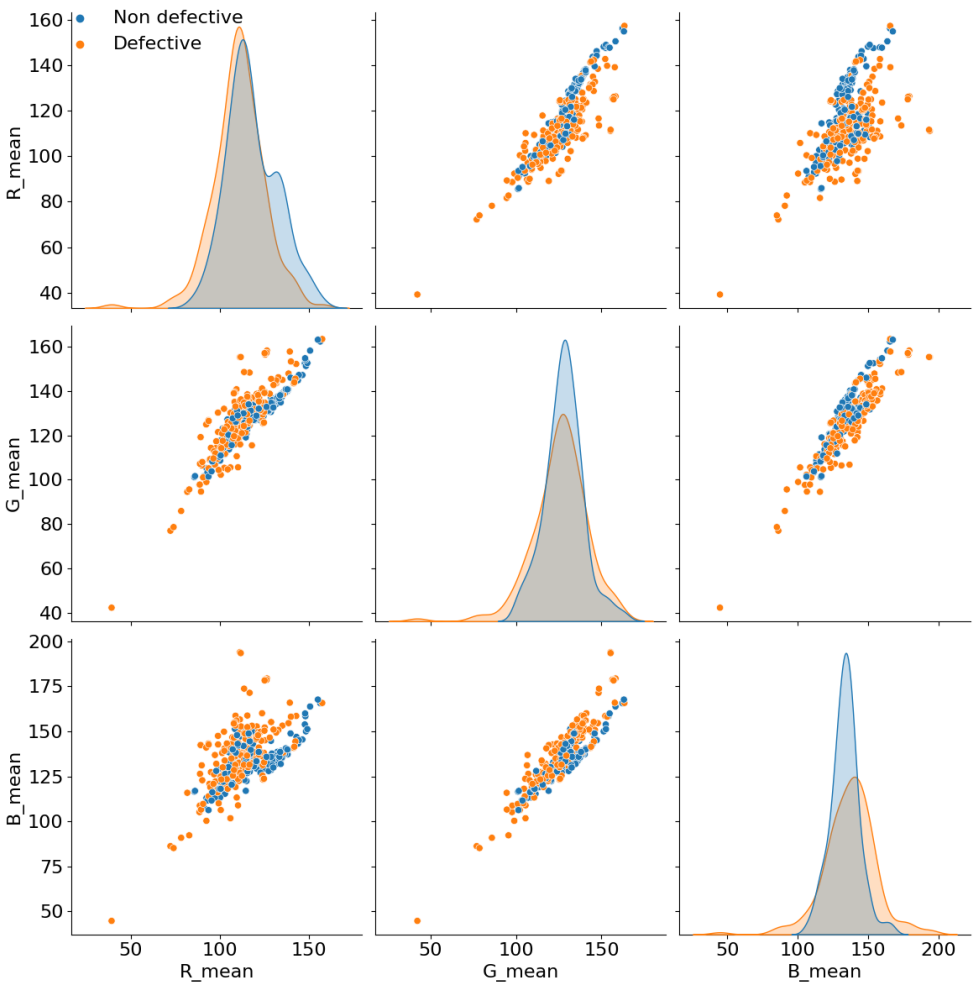

Figure 4: Color component pair analysis

(a) Cracked rail

(b) Disjoint rails

(c) Surface pitting

(d) Fastener defect

(e) Missing element

Figure 5: Identified rail defects

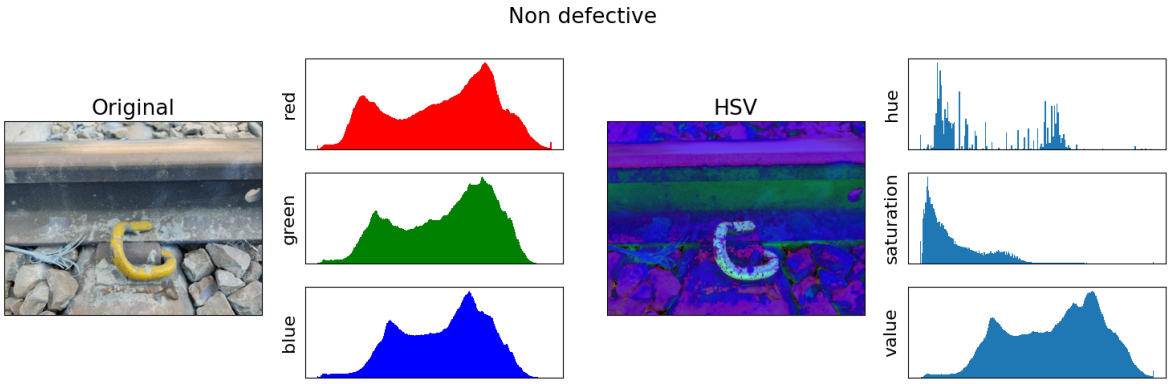

Figure 6: Color component analysis on RGB and HSV color modes

List of Tables

References

[1] Scarlett Liu, Quandong Wang, and Yiping Luo. "A review of applications of visual inspection technology based on image processing in the railway industry". In: *Transportation Safety and Environment* 1.3 (Dec. 2019), pp. 185–204. ISSN: 2631-4428. DOI: 10.1093/tse/tdz007. URL: https://academic.oup.com/tse/article/1/3/185/5714252 (visited on 12/24/2022).

[2] Xue Wang, Yike Tang, and Ping Cheng. "Machine-vision detection for rail-steel's surface flaws based on quantum neural network". In: *2008 7th World Congress on Intelligent Control and Automation*. 2008 7th World Congress on Intelligent Control and Automation. Chongqing, China: IEEE, 2008, pp. 5050–5055. ISBN: 978-1-4244-2113-8. DOI: 10.1109/WCICA.2008.4593749. URL: http://ieeexplore.ieee.org/document/4593749/ (visited on 12/25/2022).

[3] Shahrzad Faghih-Roohi et al. "Deep convolutional neural networks for detection of rail surface defects". In: *2016 International Joint Conference on Neural Networks (IJCNN)*. 2016 International Joint Conference on Neural Networks (IJCNN). Vancouver, BC, Canada: IEEE, July 2016, pp. 2584–2589. ISBN: 978-1-5090-0620-5. DOI: 10.1109/IJCNN.2016.7727522. URL: http://ieeexplore.ieee.org/document/7727522/ (visited on 12/24/2022).

[4] Ke Ma et al. "Texture classification for rail surface condition evaluation". In: *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*. 2016 IEEE Winter Conference on Applications of Computer Vision (WACV). Lake Placid, NY: IEEE, Mar. 2016, pp. 1–9. ISBN: 978-1-5090-0641-0. DOI: 10.1109/WACV.2016.7477597. URL: https://ieeexplore.ieee.org/document/7477597/ (visited on 12/25/2022).

[5] Yunus Santur, Mehmet Karakose, and Erhan Akin. "A new rail inspection method based on deep learning using laser cameras". In: *2017 International Artificial Intelligence and Data Processing Symposium (IDAP)*. 2017 International Artificial Intelligence and Data Processing Symposium (IDAP). Malatya: IEEE, Sept. 2017, pp. 1–6. ISBN: 978-1-5386-1880-6. DOI: 10.1109/IDAP.2017.8090245. URL: http://ieeexplore.ieee.org/document/8090245/ (visited on 12/25/2022).

[6] Qingyong Li et al. "A cyber-enabled visual inspection system for rail corrugation". In: *Future Generation Computer Systems* 79 (Feb. 2018), pp. 374–382. ISSN: 0167-739X. DOI: 10.1016/j.future.2017.04.032. URL: https://linkinghub.elsevier.com/retrieve/pii/S0167739X17307069 (visited on 12/25/2022).

[7] Lidan Shang et al. "Detection of rail surface defects based on CNN image recognition and classification". In: *2018 20th International Conference on Advanced Communication Technology (ICACT)*. 2018 20th International Conference on Advanced Communications Technology (ICACT). Chuncheon-si Gangwon-do, Korea (South): IEEE, Feb. 2018, pp. 45–51. ISBN: 979-11-88428-01-4. DOI: 10.23919/ICACT.2018.8323642. URL: https://ieeexplore.ieee.org/document/8323642/ (visited on 12/24/2022).

[8] Ying Li et al. "Component-based track inspection using machine-vision technology". In: *Proceedings of the 1st ACM International Conference on Multimedia Retrieval - ICMR '11*. the 1st ACM International Conference. Trento, Italy: ACM Press, 2011, pp. 1–8. ISBN: 978-1-4503-0336-1. DOI: 10.1145/1991996.1992056. URL: http://portal.acm.org/citation.cfm?doid=1991996.1992056 (visited on 12/22/2022).

[9] Hoang Trinh et al. "Enhanced rail component detection and consolidation for rail track inspection". In: *2012 IEEE Workshop on the Applications of Computer Vision (WACV)*. 2012 IEEE Workshop on Applications of Computer Vision (WACV). Breckenridge, CO, USA: IEEE, Jan. 2012, pp. 289–295. ISBN: 978-1-4673-0234-0 978-1-4673-0233-3 978-1-4673-0232-6. DOI: 10.1109/WACV.2012.6163021. URL: http://ieeexplore.ieee.org/document/6163021/ (visited on 12/25/2022).

[10] Ying Li et al. "Rail Component Detection, Optimization, and Assessment for Automatic Rail Track Inspection". In: *IEEE Transactions on Intelligent Transportation Systems* 15.2 (Apr. 2014), pp. 760–770. ISSN: 1524-9050, 1558-0016. DOI: 10.1109/TITS.2013.2287155. URL: http://ieeexplore.ieee.org/document/6662397/ (visited on 12/25/2022).

[11] P.L Mazzeo et al. "Visual recognition of fastening bolts for railroad maintenance". In: *Pattern Recognition Letters* 25.6 (Apr. 2004), pp. 669–677. ISSN: 0167-8655. DOI: 10.1016/j.patrec.2004.01.008. URL: https://linkinghub.elsevier.com/retrieve/pii/S0167865504000194 (visited on 12/25/2022).

[12] Francescomaria Marino et al. "A Real-Time Visual Inspection System for Railway Maintenance: Automatic Hexagonal-Headed Bolts Detection". In: *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)* 37.3 (May 2007), pp. 418–428. ISSN: 1094-6977. DOI: 10.1109/TSMCC.2007.893278. URL: http://ieeexplore.ieee.org/document/4154946/ (visited on 12/25/2022).

[13] P. De Ruvo et al. "A GPU-based vision system for real time detection of fastening elements in railway inspection". In: *2009 16th IEEE International Conference on Image Processing (ICIP)*. 2009 16th IEEE International Conference on Image Processing ICIP 2009. Cairo, Egypt: IEEE, Nov. 2009, pp. 2333–2336. ISBN: 978-1-4244-5653-6. DOI: 10.1109/ICIP.2009.5414438. URL: http://ieeexplore.ieee.org/document/5414438/ (visited on 12/25/2022).

[14] Rubayat Ahmed Khan, Samiul Islam, and Rubel Biswas. "Automatic detection of defective rail anchors". In: *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. 2014 IEEE 17th International Conference on Intelligent Transportation Systems (ITSC). Qingdao, China: IEEE, Oct. 2014, pp. 1583–1588. ISBN: 978-1-4799-6078-1. DOI: 10.1109/ITSC.2014.6957919. URL: http://ieeexplore.ieee.org/document/6957919/ (visited on 12/25/2022).

[15] Maneesh Kumar M. et al. "A Survey on Crack Detection Technique in Railway Track". In: *2018 Conference on Emerging Devices and Smart Systems (ICEDSS)*. 2018 Conference on Emerging Devices and Smart Systems (ICEDSS). Tiruchengode: IEEE, Mar. 2018, pp. 269–272. ISBN: 978-1-5386-3479-0. DOI: 10.1109/ICEDSS.2018.8544319. URL: https://ieeexplore.ieee.org/document/8544319/ (visited on 12/22/2022).

[16] *Kaggle: Your Home for Data Science*. URL: https://www.kaggle.com/ (visited on 12/22/2022).

[17] *Railway Track Fault Detection | Kaggle*. URL: https://www.kaggle.com/datasets/salmaneunus/railway-track-fault-detection (visited on 12/22/2022).