Research, Society and Development, v. 11, n. 8, e12211830252, 2022
(CC BY 4.0) | ISSN 2525-3409 | DOI: http://dx.doi.org/10.33448/rsd-v11i8.30252

An in-depth assessment of convolutional neural networks for rail surface defect detection

Uma avaliação abrangente de rede neural convolucional para detecção de defeitos na superfície ferroviária

Una evaluación en profundidad de las redes neuronales convolucionales para la detección de defectos en la superficie del carril

Received: 05/12/2022 | Reviewed: 06/09/2022 | Accept: 06/10/2022 | Published: 06/12/2022

Rebeca Alves da Silva Lemos Passos
ORCID: https://orcid.org/0000-0001-6821-0686
Military Institute of Engineering, Brazil
E-mail: rebecapassos@ime.eb.br
Matheus Pinheiro Ferreira
ORCID: https://orcid.org/0000-0003-0687-4422
Military Institute of Engineering, Brazil
E-mail: matheus@ime.eb.br
Ben-Hur de Albuquerque e Silva
ORCID: https://orcid.org/0000-0001-6933-498X
Military Institute of Engineering, Brazil
E-mail: benhur@ime.eb.br
Luiz Antonio Silveira Lopes
ORCID: https://orcid.org/0000-0001-9325-4330
Military Institute of Engineering, Brazil
E-mail: laslopes@ime.eb.br
Hugo Ribeiro
ORCID: https://orcid.org/0000-0002-1710-4899
MRS Logística, Brazil
E-mail: Hugo.Ribeiro@mrs.com.br
Romero Pereira dos Santos
ORCID: https://orcid.org/0000-0001-5220-1451
MRS Logística, Brazil
E-mail: Romero.Santos@mrs.com.br

Abstract

The consistent monitoring of rails is based on correctly identifying defects to support corrective measures. Recently, convolutional neural networks (CNN), a deep learning method, have been providing outstanding results for the automatic detection of defects. However, several aspects of CNN-based approaches such as network architecture, transfer learning and processing time remains not fully understood. In this work, we performed an in-depth assessment of ten widely used CNN models with the objective of finding the one with the best performance in identifying defects in rail surface images. The classification results are promising, reaching an average accuracy of 83.7% on detection of mild defects and squat. The Inceptionv3 network provided the best results by correctly identifying 92% of images with severe squat defects.
Keywords: Rail inspection; Squat; CNN.

Resumo

O monitoramento consistente dos trilhos baseia-se na identificação correta dos defeitos para apoiar as medidas corretivas. Recentemente, as redes neurais convolucionais (CNN), um método de aprendizado profundo, vêm apresentando excelentes resultados para a detecção automática de defeitos. No entanto, vários aspectos das abordagens baseadas em CNN, como arquitetura de rede, aprendizado de transferência e tempo de processamento, ainda não são totalmente compreendidos. Neste trabalho, realizamos uma avaliação aprofundada de dez modelos CNN amplamente utilizados, com o objetivo de encontrar aquele com melhor desempenho em identificar defeitos em imagens de superfície do trilho. Os resultados da classificação são promissores, atingindo uma acurácia média de 83,7% na detecção de defeitos leves e agachamento. A rede Inceptionv3 forneceu os melhores resultados ao identificar corretamente 92% das imagens com defeitos graves de squat.
Palavras-chave: Inspeção ferroviária; Squat; CNN.

Resumen
El monitoreo consistente de los rieles se basa en identificar correctamente los defectos para respaldar las medidas correctivas. Recientemente, las redes neuronales convolucionales (CNN), un método de aprendizaje profundo, han proporcionado resultados sobresalientes para la detección automática de defectos. Sin embargo, varios aspectos de los enfoques basados en CNN, como la arquitectura de la red, el aprendizaje de la transferencia y el tiempo de procesamiento, aún no se comprenden por completo. En este trabajo, realizamos una evaluación en profundidad de diez modelos CNN ampliamente utilizados con el objetivo de encontrar el que tenga el mejor rendimiento en la identificación de defectos en las imágenes de la superficie del carril. Los resultados de la clasificación son prometedores, alcanzando una precisión media del 83,7 % en la detección de defectos leves y achaparrados. La red Inceptionv3 brindó los mejores resultados al identificar correctamente el 92 % de las imágenes con graves defectos de posición en cuclillas.
Palabras clave: Inspección ferroviaria; Okupa; CNN.

1. Introduction

The propagation of cracks in railway tracks gives rise to fractures that could lead to catastrophic events. Therefore, one should conduct safety inspections to detect the crack formation and propagation before the fracture occurs. Moreover, consistent monitoring of railways is required to provide reliable data to maintenance teams for planning future corrective actions. Thus, achieving operational security and eliminate existing defects (MRS, 2008).

Recently, automated defect detection in railway tracks has been increasingly studied due to the development of computer vision and as an exciting alternative to manual monitoring, which is slow, exhausting, subjective, and costly (Yanan et al., 2018). Among the automated detection methods, those using railway images and convolutional neural networks (CNNs) are promising (Faghih-Roohi et al., 2016). CNNs constitute a class of deep artificial neural networks that rely on local linear operations (convolutions) followed by non-linear transformations, creating different representations of the input data. The convolutional layers are filters that extract low-level features (e.g., object edges) and high-level features (e.g., object shapes), considering the spatial context. A non-linear activation function is usually applied to the output of a convolutional layer, followed by a pooling (downsampling) operation to reduce its dimension. After several convolutional and pooling layers, a fully connected (FC) layer might be included to exploit the high-level features learned. The FC layer could be seen as hidden layers of a multilayer perceptron (MLP) network. Finally, the last layer is often a softmax classifier that outputs class membership probabilities for each class. A comprehensive overview of CNNs and deep learning can be found in Ponti et al. (2017). To rail surface defect detection, some studies employed CNN-based methods for scene classification and object detection. Scene classification aims to identify a defect in the rail given an image as input (usually a grayscale photograph). At the same time, object detection approaches draw a bounding box around the defect, finding it in the image. Faghih-Roohi et al. (2016), for example, proposed three deep CNN (DCNN) for scene classification to identify defects in railway tracks. The authors successfully classified normal rail, small defects, and squats with almost 92% accuracy.

Similarly, Jamshidi et al. (2017) developed a DCNN model to classify images representing normal rail, trivial defects (seed squats), and squats. For a binary classification problem (squat vs. normal), the authors obtained a classification accuracy of 96.9%. Object detection networks such as the YOLOv3 (Redmon and Farhadi, 2018) have been employed to retrieve rail surface defects in grayscale images. Yanan et al. (2018) detected defects in rails with 97% accuracy with YOLOv3. Yuan et al. (2019) combined YOLOv3 network with MobileNetV2 (Sandler et al., 2018) to retrieve three types of rail surface defects. The MobileNetV2 architecture is used as a backbone network to extract image features, whereas YOLOv3 works on the regression prediction. The experiments showed that the combination of MobileNetV2 with YOLOv3 achieves higher detection accuracy and robustness when compared to the YOLOv3 alone, achieving 87.40% of average accuracy. Rodrigues (2020) used the supervised machine learning algorithm SVM (Support Vector Machine) to detect defects on the surface of the billet through images. After training, the network was able to identify tracks in grinding condition and tracks with severe damage. The model achieved more than 95% accuracy in image classification.

Efforts have been made to detect irregularities in rail components, such as fasteners, for example. In a recent study, Yuan et al. (2021) designed a one-dimensional CNN to inspect the fasteners from the time domain recorded by the accelerometer of a rail with fasteners in different degradation conditions. The authors report that the model achieves high detection accuracy and good noise flexibility.

Some studies employed CNNs for semantic segmentation, assigning a label to each image pixel, thus performing pixel-level classification of defects. Liang et al. (2018) proposed an image processing pipeline based on the SegNet (Badrinarayanan et al., 2017) semantic segmentation architecture and obtained results with a 100% detection rate. More recently, Kim et al. (2020) modified the AlexNet (Krizhevsky et al., 2012) and the visual geometry group (VGG) (Simonyan & Zisserman, 2014) networks for semantic segmentation and achieved 99% of accuracy. Given the outstanding results of CNN-based methods to automatically detect rail surface defects in the last years, an in-depth assessment of the most used architectures is needed. Such an assessment may provide valuable insights for the real-world application of CNNs aiming at reliable and fast detection of rail defects. In this work, we assess ten CNN architectures. We focused on identifying a severe type of defect called squat, caused by rolling contact fatigue at the wheel-rail interface and is characterized by the shattering of the gauge corner (MRS, 2008). We trained the networks with and without transfer learning. We analyze features such as computation time, accuracy, and the number of model parameters. This is the first work that assessed different CNNs to identify rail surface defects to the best of our knowledge. Moreover, most studies have focused on detecting the presence or not of a defect, with its classification remaining a challenge.

2. Methodology

2.1 *Image database and pre-processing*

The images used in this work were collected from a critical section of Barra do Piraí's railway line in Rio de Janeiro, Brazil. The railway line is under the concession of MRS Logística S.A, which captured the images using a railway inspection vehicle (RIV) (Figure 1). The grayscale images captured by the RIV have a dimension of 1600×1200 pixels. A total of 244 images were collected, of which 80 were taken from railways without any defects while the others presented some flaws. The image labeling was performed according to the fracture and defect identification guidelines of railways from MRS (2008) by an expert from the company.

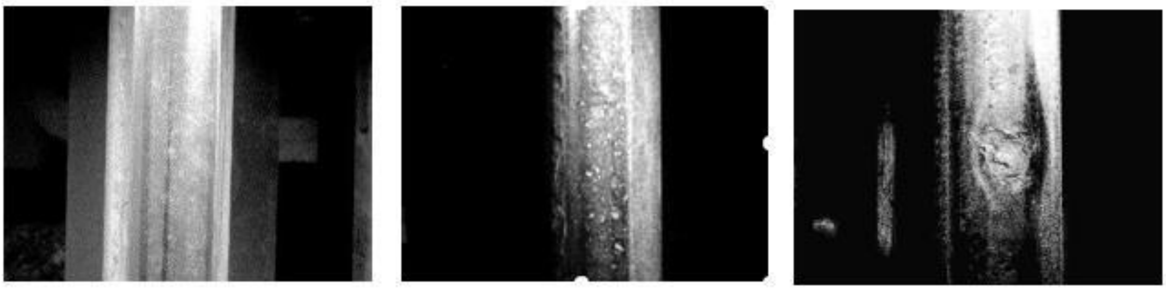
Figure 1. Rail Inspect Vehicle (RIV).

Source: Adapted from Loram (2018).

(CC BY 4.0) | ISSN 2525-3409 | DOI: http://dx.doi.org/10.33448/rsd-v11i8.30252

The dataset contains many defects such as head checking, flaking spalling, and squat. The railway lines without defects were grouped in a class called “Normal” (Figure 2a). The less severe superficial defects were arranged in “Mild Defects” (Figure 2b). Images with squat defects were arranged in a group called “Squat” (Figure 2c).

Figure 2. Examples of the dataset (a) class “Normal” (b) class “Moderate” (c) Squat.



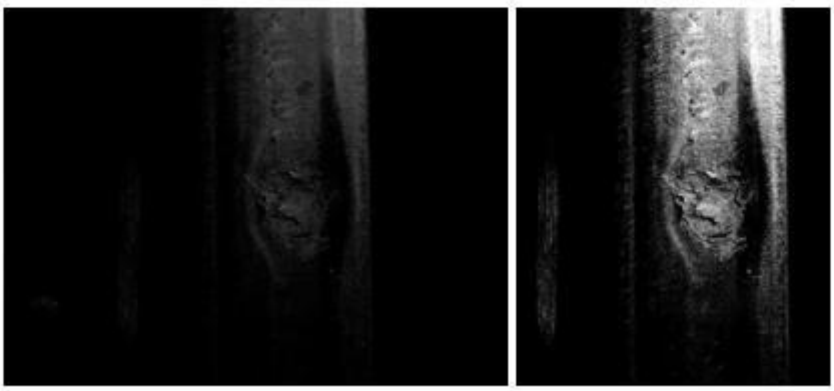
(a) (b) (C)

Source: MRS (2020).

As shown in Figure 2b, the “moderate” class is represented by a slight loss of billet material due to the high stresses of the wheel-rail contact. The “Squat” class (Figure 2c) is represented by cracks and holes in a large area on the rail surface, caused by contact fatigue and rail irregularities, such as weld and billet widening (MRS, 2008).

We cropped 300 pixels from each side of the original images to avoid processing areas without the railway. The cropped images have a dimension of 1001 x 1200 pixels that should be reduced by half, i.e., 500 x 600 pixels, to reduce the processing time. Moreover, we improved the contrast of the images to highlight the railway. We performed a simple linear contrast enhancement, excluding the bottom 1% and the top 1% of all pixel values. The original and final images are shown in Figure 3.

Figure 3. (a) Original image and (b) after pre-processing.



(a) (b)

Source: Adapted from MRS (2020).

Through Figure 3b, it is possible to observe that the contrast change performed made the light colors of the image lighter and the dark colors darker, improving the visual quality of the image and, consequently, facilitating the identification of the defect by the neural network.

2.2 CNNArchitectures

We assessed ten CNN architectures widely used in computer vision and image classification tasks. Table 1 summarizes the characteristics of each architecture, including depth (number of layers), size, parameters, image input size, and the reference work.

Table 1. List of CNN architectures investigated.

Network	Depth (n. layers)	Size (MB)	Parameters (Million)	Image input size (pixels)	Reference
Squeezenet	18	5,2 MB	1.24	227-by-227	Iandola et al. (2016)
Googlenet	22	27 MB	7	224-by-224	Szegedy et al. (2015)
Inceptionv3	48	89 MB	23.9	299 by 299	Lin et al. (2019)
Densenet201	201	77 MB	20	224-by-224	Huang et al. (2017)
Mobilenetv2	53	13 MB	3.5	224-by-224	Sandler et al (2018)
Resnet18	18	44 MB	11.7	224-by-224	He et al. (2016)
Resnet50	50	96 MB	25.6	224-by-224	He et al. (2016)
Resnet101	101	167 MB	44.6	224-by-224	He et al. (2016)
Xception	71	85 MB	22.9	299 by 299	Chollet (2017)
Efficientnetb0	82	20 MB	5.3	224-by-224	Tan & Quoc (2019)

Source: MATHWORKS (2020).

2.3 Experimental setup

We first evaluated the use of transfer learning, which refers to initializing the weights of the CNN models with values obtained after training them for a different classification problem. Transfer learning proved helpful in reducing computation time and improving accuracy in several image classification tasks (Shin et al., 2016). We initialized the weights of the networks using pre-trained values of the ImageNet database (Deng et al., 2009) and with random values following a uniform distribution, which we refer to from "scratch." We used a desktop computer with an Intel Core i7-8700 3.2GHz CPU, 24GB of main memory, and an NVIDIA® GeForce Titan V GPU with 12GB of dedicated memory for training and inference. We implemented all image processing procedures in the MATLAB® environment.

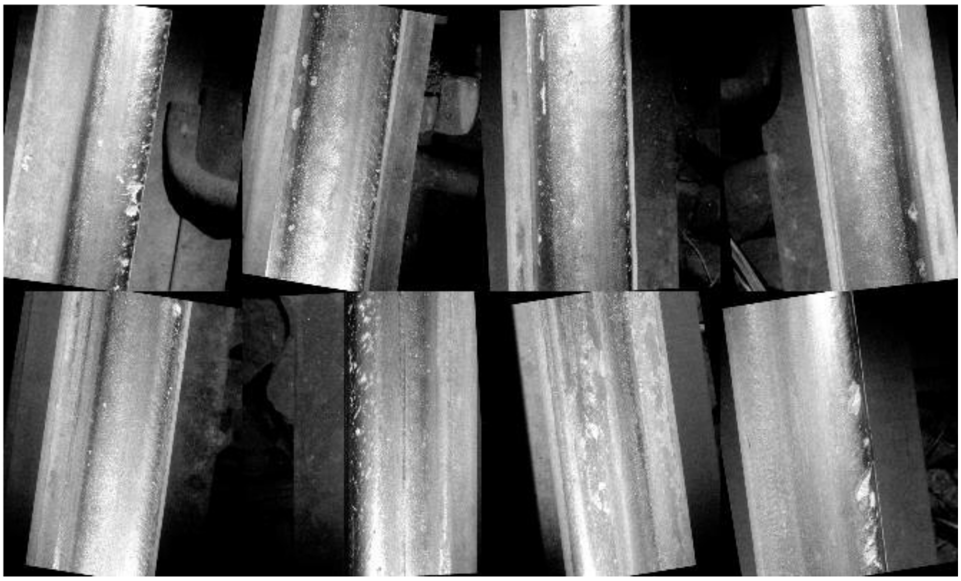
Due to the small number of images, we used image augmentation on the training dataset, which is composed of 244 images or 80% of the total number of images. The augmentation methods comprised image rotations between -10◦ and 10◦ and a translation of three pixels on the y- and x-axes. The augmented dataset was composed of 1025 images. Details of the dataset and samples that went through the process are shown in Table 2 and Figure 4, respectively.

Table 2. Number of training and testing images.

Class	Nº of training images	Nº of training images after augmentation	Nº of Test images
Normal	80	320	16
Moderate	76	304	15
Squat	88	352	18

Source: MATHWORKS (2020).

Figure 4. Examples of training images after augmentation.



Source: Adapted from MRS (2020).

Figure 4 shows eight training images generated with the application of rotation and translation. These images were created within the network and discarded after its training. The greater the number of examples available for training the network, the greater its ability to generalize learning to new situations, reducing the occurrence of overfitting, a natural tendency that networks have to memorize training examples (Demuth, 2000).

We tested all CNN models with and without transfer learning. To use transfer learning, we replaced the FC layer with a new one with the number of classes of our dataset. Initial tests performed with the Resnet-18 network revealed that low initial learning (< 0.001) produced overfitting. The best accuracy in the network was achieved using an initial learning rate of 0.01 with a decaying factor of 0.5 after 90 epochs with a minibatch of 12 and 200 epochs. We trained all networks with these settings. To avoid biased training due to class imbalance, we used class weights.

2.4 Accuracy assessment

We performed the accuracy assessment with the testing images that were not used to train the models. We computed the mean accuracy and F1-score for each class. To evaluate the network accuracy in a binary classification, the classes “Mild defects” and “Squats” were considered one group.

3. Results and Discussion

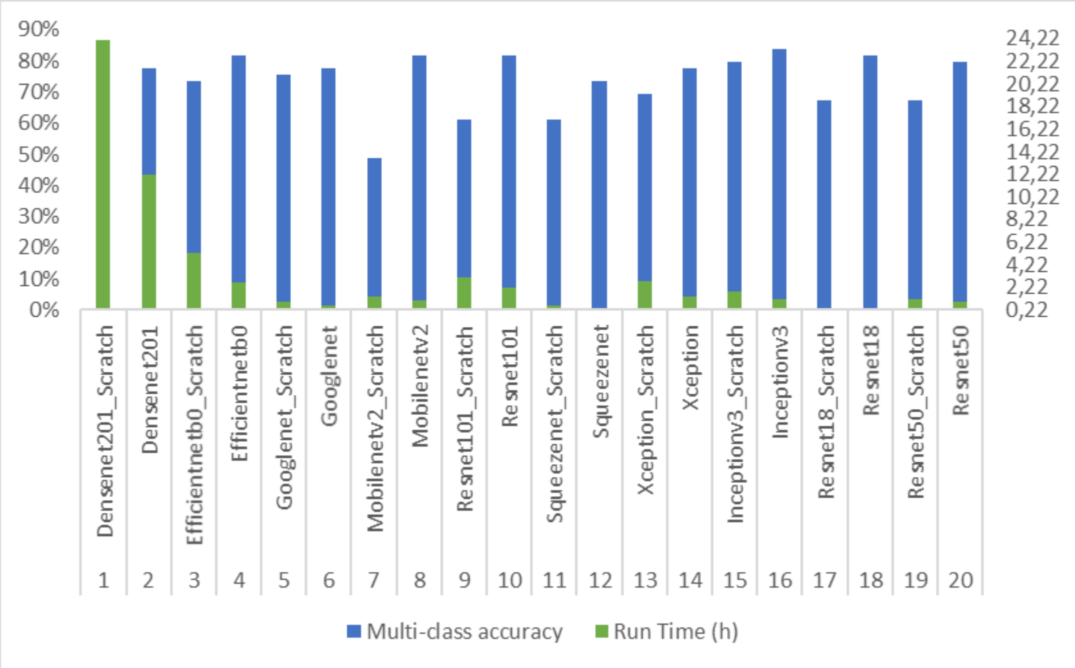
Tables 3 and Figure 5 show the multi-class classification result of the trained CNNs. The untrained versions are identified by the ending “Scratch”. The Inceptionv3 obtained the best classification Median accuracy, F1-score "Normal" and F1-score "Squat" from the test set, reaching 83.67%, 86,49%, and 91,89%, respectively. Its training time was 4.419 seconds, 5.5 times more than the fastest network, SqueezeNet. On the other hand, the network with the worst performance was Mobilenetv2_Scratch, with 40% accuracy in classifying railway lines with mild defects, 63.15% accuracy for normal tracks, and 40% for the ones with squat defects.

Table 3. Classification accuracy results using transfer learning.

	CNN	Multi-class accuracy	F1-score "Normal"	F1-score "Mild defects"	F1-score "Squat"	Run Time (s)	Relative Run Time
1	Densenet201_Scratch	71.42%	76.92%	60.00%	75.86%	86,733	108.6
2	Densenet201	77.55%	88.88%	62.06%	78.78%	43,791	54.9
3	Efficientnetb0_Scratch	73.46%	82.05%	62.06%	73.33%	18,781	23.5
4	Efficientnetb0	81.63%	84.21%	69.23%	88.23%	9,718	12.2
5	Googlenet_Scratch	75.51%	84.21%	54.54%	78.94%	3,353	4.2
6	Googlenet	77.55%	88.88%	68.75%	73.33%	2,166	2.7
7	Mobilenetv2_Scratch	48.97%	63.15%	40.00%	40%	5,084	6.4
8	Mobilenetv2	81.63%	84.21%	64.00%	91.42%	3,813	4.8
9	Resnet101_Scratch	61.22%	80%	51.42%	43.47%	11,108	13.9
10	Resnet101	81.63%	86.48%	66.66%	88.23%	8,009	10.0
11	Squeezenet_Scratch	61.22%	75.67%	46.66%	58.06%	2,158	2.7
12	Squeezenet	73.46%	88.88%	60.00%	68.75%	798	1.0
13	Xception_Scratch	69.38%	78.04%	51.85%	73.33%	10,04	12.6
14	Xception	77.55%	84.21%	58.33%	83.33%	5,148	6.4
15	Inceptionv3_Scratch	79.59%	91.42%	68.75%	77.41%	6,54	8.2
16	Inceptionv3	83.67%	86.48%	66.66%	91.89%	4,419	5.5
17	Resnet18_Scratch	67.34%	76.19%	55.17%	66.66%	1,012	1.3
18	Resnet18	81.63%	88.88%	69.23%	83.33%	943	1.2
19	Resnet50_Scratch	67.34%	76.19%	55.17%	66.66%	4,287	5.4
20	Resnet50	79.59%	84.21%	66.66%	84.84%	3,604	4.5

Source: Author himself (2021).

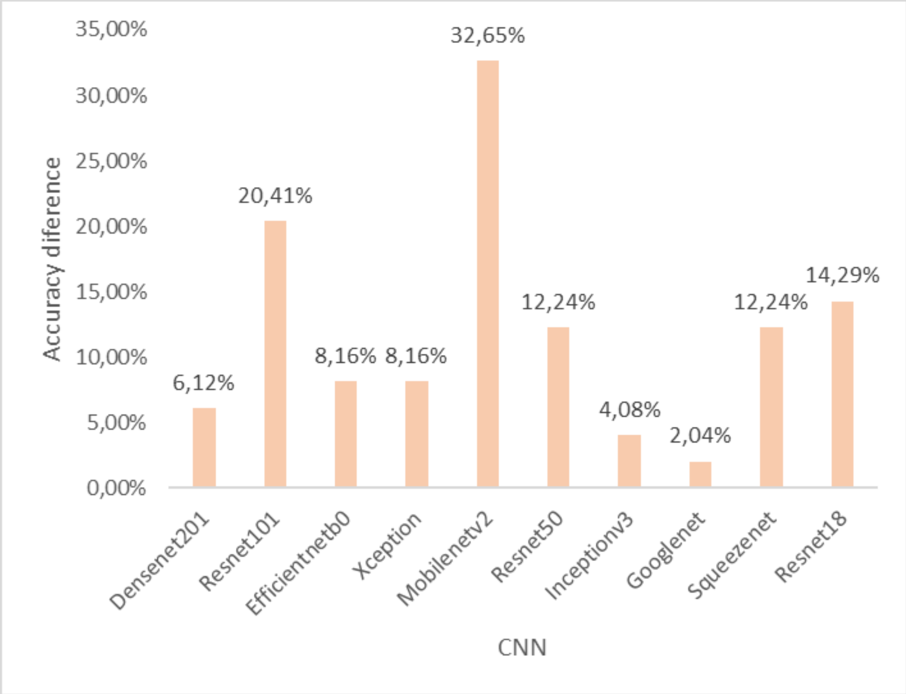
Figure 5. Median accuracy and Run time.



Source: Author himself (2021).

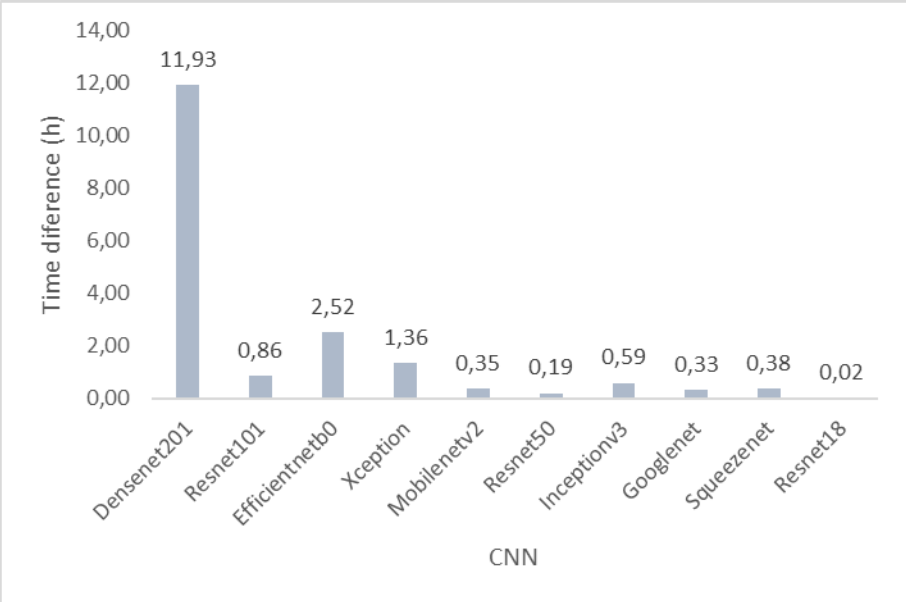
All networks reached better classification accuracy in their pre-trained versions, with an average deviation of 12% concerning the untrained ones. Mobilenetv2 is the network that deviates the most, with an accuracy difference of 32.7% between the untrained and pre-trained versions (Figure 6). Similarly, all networks reached a shorter run time in their pre-trained versions. Densenet201 is the network with the greatest run time difference between the versions (53,8 h), as shown in Figure 7.

Figure 6. Accuracy difference between pre-trained and untrained neural networks.



Source: Author himself (2021).

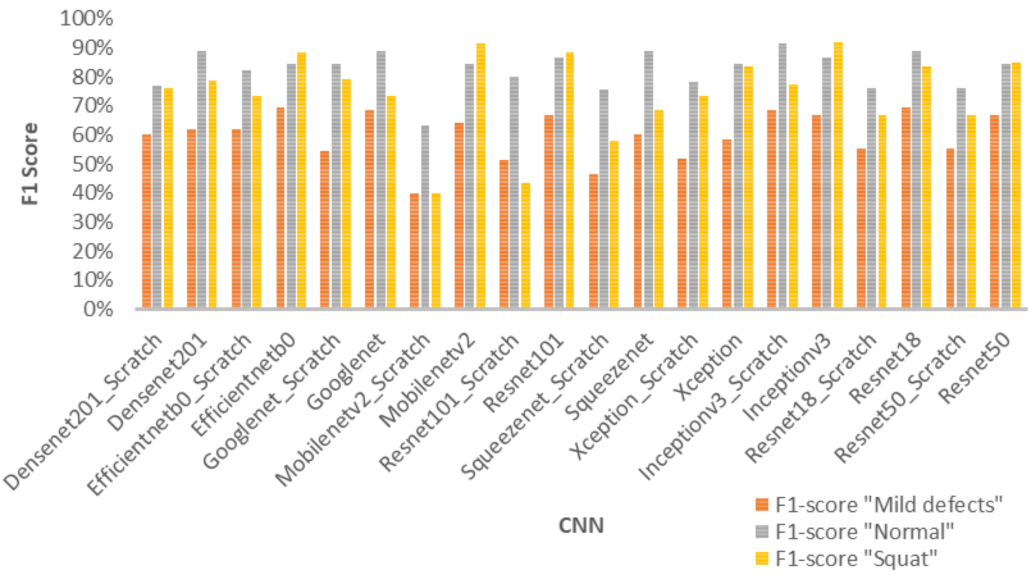
Figure 7. Run time difference between pre-trained and untrained neural networks.



Source: Author himself (2021).

Figure 8 shows that the networks were not able to identify images containing railway lines with mild defects. Conversely, railway lines with Squat were easier to identify.

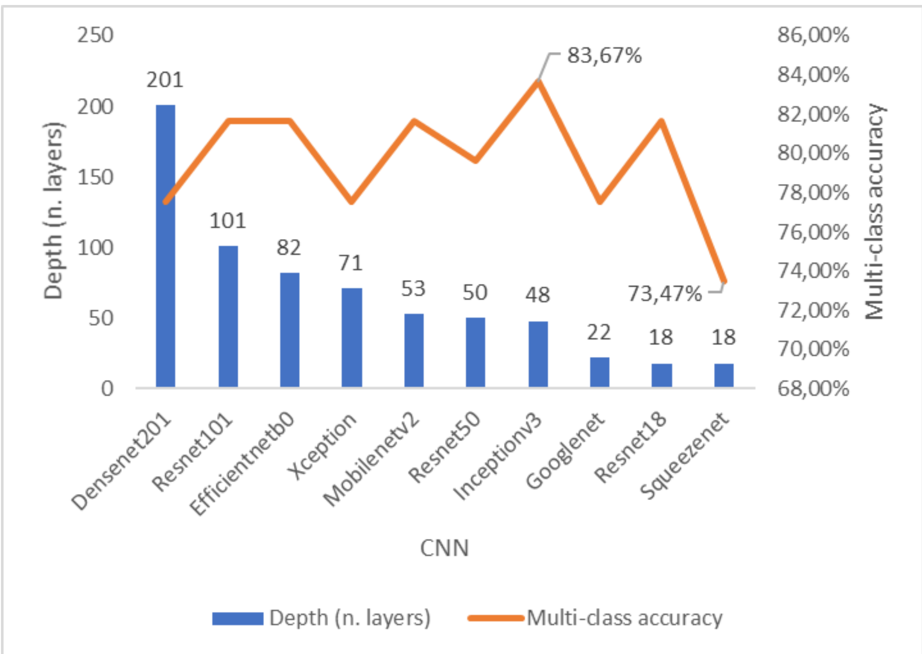
Figure 8. F1-score of trained CNNs.



Source: Author himself (2021).

Figure 9 and Figure 10 show the influence of network depth (number of layers) on the validation accuracy and training time, respectively.

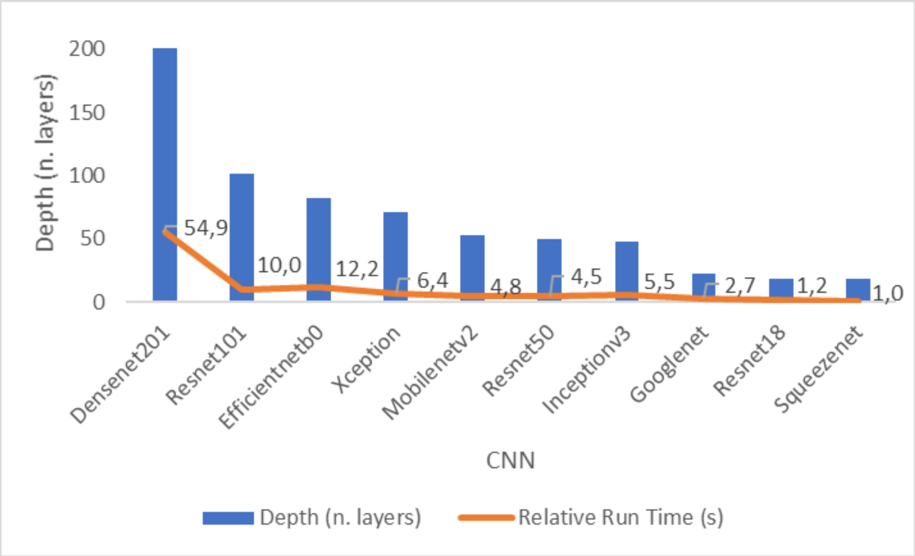
Figure 9. Median accuracy x depth (n. layers).



Source: Author himself (2021).

The median accuracy x depth graph, illustrated in Figure 9, arranges the CNNs in descending order according to the number of layers. It is possible to observe that the line representing the average accuracies obtained by the networks in the classification of the test images does not decrease as the number of layers decreases. The highest average accuracy was achieved by the 48-layer Inceptionv3 and the lowest average accuracy, but not far from the Inceptionv3, was achieved by the 18-layer squeezenet.

Figure 10. Run time x depth (n. layers).



Source: Author himself (2021).

The run time x depth graph, illustrated in Figure 10, arranges the CNNs in descending order according to the number of layers. It is possible to observe that the line representing the training time used by the networks tends to decrease as the number of layers decreases.

Finally, Table 4 presents a binary classification of the accuracy of “Normal railway lines” and “Railway lines with defects”. In the binary classification, the methodology correctly predicted 88.48% of the images using Inceptionv3.

Table 4. Binary accuracy values of CNNs.

DCNN	Binary accuracy	Binary F1-score
Densenet201_Scratch	78,94%	0.7142
Densenet201	83,82%	0.7755
Efficientnetb0_Scratch	80,59%	0.7346
Efficientnetb0	86,95%	0.8163
Googlenet_Scratch	82,22%	0.7551
Googlenet	83,82%	0.7755
Mobilenetv2_Scratch	59,01%	0.4897
Mobilenetv2	86,95%	0.8163
Resnet101_Scratch	70,31%	0.6122
Resnet101	86,95%	0.8163
Squeezenet_Scratch	70,31%	0.6122
Squeezenet	80,59%	0.7346
Xception_Scratch	77,27%	0.6938
Xception	83,82%	0.7755
Inceptionv3_Scratch	85,4%	0.7959
Inceptionv3	88,48%	0.8367
Resnet18_Scratch	75,57%	0.6734
Resnet18	86,95%	0.8163
Resnet50_Scratch	75,57%	0.6734
Resnet50	85,4%	0.7959

Source: Author himself (2021).

4. Conclusion

We analyzed the performance of ten untrained and pre-trained artificial neural network structures in detecting rails without defects, with mild defects, and squat. The pre-trained networks were trained with more than 1 million images from the ImageNet dataset, and they learned features to classify 1000 object classes. An augmentation technique was employed to increase the number of images used during training. Additionally, the images underwent some pre-processing to crop the non-important regions and improve the contrast.

The best median accuracy was 83.67% with Inceptionv3, the highest F1-score “Squat” (91.89%). The Mobilenetv2_Scratch showed lower performance with a median accuracy of 48.98% and the Resnet-18 an excellent cost of time x accuracy with a median of 81.63% and 1,2 of Relative Run Time.

Defect segmentation should be fast and accurate. In our experiment, we conclude that simple network and transfer learning can be applied in the effective rail surfaces defect detection, how the squat can achieve promising results in mild defects detection. We suggest increasing the number of examples of mild defects to affect the methodology as a preventive maintenance tool in future works.

Acknowledgments

We thank MRS Logística S.A for the partnership in this research project and our funding agency CAPES.

References

Badrinarayanan, V., & Kendall, A., & Cipolla, R. (2017). *Segnet*: *A deep convolutional encoder-decoder architecture for image segmentation*. IEEE transactions on pattern analysis and machine intelligence 39, 2481–2495.

Chollet, F. (2017). Xception: *Deep learning with depthwise separable convolutions*, in*: Proceedings of the IEEE conference on computer vision and pattern recognition*, 1251–1258.

Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., & Fei-Fei, L. (2009*). Imagenet: A large-scale hierarchical image database,* in: 2009 *IEEE conference on computer vision and pattern recognition,* IEE, 248–255.

Demuth H. B. *Neural network toolbox; for use with MATLAB; computation, visualization, programming; user's guide, version 4*. [S.l.], 2000. Disponível em: <http://cda.psych.uiuc.edu/matlab_pdf/nnet.pdf>.

Faghih-Roohi, S., Hajizadeh, S., Núñez, A., Babuska, R., & De Schutter, B. (2016). *Deep convolutional neural networks for detection of rail surface defects, in: 2016 International joint conference on neural networks* (IJCNN), IEEE, 2584–2589.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). *Deep residual learning for image recognition*, *in: Proceedings of the IEEE conference on computer vision and pattern recognition,* 770–778.

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K.Q. (2017). *Densely connected convolutional networks*, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 4700–4708.

Iandola, F.N., Han, S., Moskewicz, M.W., Ashraf, K., Dally, W.J., & Keutzer, K. (2016). *Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size*. arXiv preprint arXiv:1602.07360.

Jamshidi, A., Faghih-Roohi, S., Hajizadeh, S., Núñez, A., Babuska, R., Dollevoet, R., Li, Z., & De Schutter, B. (2017). *A big data analysis approach for rail failure risk assessment*. Risk Analysis 37, 1495–1507.

Kim, H., Lee, S., & Han, S. (2020*). Railroad surface defect segmentation using a modified fully convolutional network*. *KSII Transactions on Internet & Information Systems* 14.

Krizhevsky, A., Sutskever, I., & Hinton, G.E. (2012). *Imagenet classification with deep convolutional neural networks, in: Advances in neural information processing systems*, 1097–1105.

Liang, Z., Zhang, H., Liu, L., He, Z., & Zheng, K. (2018). *Defect detection of rail surface with deep convolutional neural networks,* in: 2018 13th *World Congress on Intelligent Control and Automation* (WCICA), IEEE, 1317–1322.

Lin, C., Li, L., Luo, W., Wang, K.C., & Guo, J. (2019). *Transfer learning based traffic sign recognition using inception*-v3 model. *Periodica Polytechnica Transportation Engineering* 47, 242–250.

Loram. (2018). Riv. *URL: https://www.loram.com/products/ quality-management/riv.*

MRS. (2008). MRS Logística S.A. *Guia de Identificação de Defeitos e Fraturas em Trilhos*.

Ponti, M.A., Ribeiro, L.S.F., Nazare, T.S., Bui, T., & Collomosse, J., (2017). *Everything you wanted to know about deep learning for computer vision but were afraid to ask*, in: 2017 30th *SIBGRAPI conference on graphics, patterns and images tutorials* (SIBGRAPI-T), IEEE, 17–41.

Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767.

Rodrigues, Paulo Cézar Lobo. *Detecção de anomalias em trilho utilizando visão computacional*. *Instituto Federal do Espírito Santo, Programa de Pós-graduação em Tecnologias Sustentáveis*, Vitória,2020.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.C. (2018). *Mobilenetv2: Inverted residuals and linear bottlenecks*, in: Proceedings of *the IEEE conference on computer vision and pattern recognition,* 4510–4520.

Shin, H. C., Roth, H. R., Gao, M., Lu, L., Xu, Z., Nogues, I., Yao, J., Mollura, D., & Summers, R. M. (2016). *Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning*. *IEEE transactions on medical imaging* 35, 1285–1298.

Silva, P. et al. (2014). *Use of artificial neural networks and geographic objects to classify remote sensing images. CERNE*, 20(2), 267-276.

Simonyan, K., & Zisserman, A. (2014). *Very deep convolutional networks for large-scale image recognition*. arXiv preprint arXiv:1409.1556.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). *Going deeper with convolutions*, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1–9.

Tan, M., & Quoc, V. L. (2019). *Efficientnet: Rethinking model scaling for convolutional neural networks*. arxiv e-prints, art. arXiv preprint arXiv:1905.119461.

Yanan, S., Hui, Z., Li, L., & Hang, Z. (2018). *Rail surface defect detection method based on yolov3 deep learning network*s, in: 2018 *Chinese Automation Congress (CAC),* IEEE, 1563–1568.

Yuan, H., Chen, H., Liu, S., Lin, J., & Luo, X. (2019). *A deep convolutional neural network for detection of rail surface defect,* in: 2019 IEEE *Vehicle Power and Propulsion Conference* (VPPC), IEEE, 1–4.

Zhandong Y., Shengyang Z., Xuancheng Y., & Wanming Z. (2021). *Vibration-based damage detection of rail fastener clip using convolutional neural network: Experiment and simulation. Engineering Failure Analysis*. Volume 119.