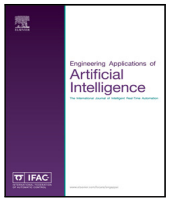
Engineering Applications of Artificial Intelligence 125 (2023) 106622

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai



TrackSafe: A comparative study of data-driven techniques for automated railway track fault detection using image datasets

Marta Garcia Minguell, Ravi Pandit *

Centre for Life-cycle Engineering and Management, Cranfield University, United Kingdom

ARTICLE INFO

Keywords:
Artificial intelligence
Object detection
Railway track faults
Defect detection
Image processing

ABSTRACT

Railway track accidents continue to occur despite manual inspections, which are often inaccurate and can lead to catastrophic events. While artificial intelligence has been applied in the railway sector, few studies have focused on defect detection using object detection tools. Additionally, there is a lack of studies that compare different models using the same dataset.

This paper proposes new data-driven techniques that identify railway track faults using three object detection models: YOLOv5, Faster RCNN, and EfficientDet. These models are compared by testing a dataset of 31 images that contain three different railway track elements (clip, rail, and fishplate), both faulty and non-faulty. Six classes were differentiated in the training of the models: one faulty and one non-faulty for each of the three classes. Image pre-processing steps included data augmentation techniques and image resizing. Results show good precision (equivalent to 1) in detecting non-defective elements, but recall values for defective elements vary among models, with Faster RCNN performing the best (0.93), followed by EfficientDet (0.81), and YOLOv5 (0.68). The full paper discusses the strengths and weaknesses of these proposed techniques for railway fault detection.

1. Introduction

Rail transport is a commonly used mode of transportation for both passengers and freight, covering both short and long distances. Although the railway sector has witnessed a decline in the number of accidents in the past decade, with a 45% decrease in the number of fatalities in the EU from 2010 to 2020 (Anon, 2021), the primary causes of accidents remain unchanged. While train accidents can occur due to various reasons, defective rails, mechanical track failures leading to derailments, and broken rails are among the most frequent causes (Sanger, 2018; Zang et al., 2019). According to recent data spanning from 2016 to 2020, track buckles and broken rails rank as the two most common precursors to accidents in the railway sector (European Union Agency for Railway, 2022; Rengel et al., 2022a).

Artificial Intelligence (AI) has become increasingly prevalent in the railway sector, with applications in seven distinct sub-domains: maintenance and inspection, safety and security, autonomous driving and control, traffic planning and management, transport policy, revenue management, and passenger mobility (Anon, 2020). In the maintenance and inspection field, the primary function of AI is to move from corrective/preventative inspection to predictive inspection (Tang et al., 2022; Sresakoolchai and Kaewunruen, 2022). A data-driven Internet of Things (IoT) based prototype, referred to as "MUHAFIZ", is presented in Shah et al. (2021) as an automated and portable TRV (track recording vehicle) with a revolutionary design based on an axle-based acceleration approach for rail track defect diagnostics. Field-based testing has demonstrated that MUHAFIZ is 87% more effective than the standard push-trolley-based TRV mechanism. Fast Fourier and wavelet transformations (WTs) are widely utilized to identify railway defects, as seen in Ghosh et al. (2021), where a methodology is proposed for real-time train line state identification. The findings indicate that fracture damage is more likely to be identified by WT than by Fast Fourier transform (FFT), while corrugation defects are more likely to be detected by FFT than WT. The accelerometer sensors employed in this study to detect vibrations are susceptible to interference from other vibration sources since they are installed on the axle boxes of service trains. The recommended approach is less effective in detecting wheel faults. In Rosyidi et al. (2022), the author proposed the use of Principle Component Analysis (PCA) to estimate the remaining useful time (RUL) of the subsystem in the automatic railroad crossing system, predicting when maintenance is required. The proposed technique was found to be effective in estimating RUL, supported by simulation results.

In recent years, digital technologies and various sensors have been utilized to detect rail defects and deterioration. For instance, vibration signals (Najeh et al., 2021) and acoustic analysis (Shah et al., 2021) have been employed. Karakose and Yaman (2020) suggested a fuzzy system-based thermography solution to mitigate the impact of weather

* Corresponding author.
E-mail address: ravi.pandit@cranfield.ac.uk (R. Pandit).

https://doi.org/10.1016/j.engappai.2023.106622
Received 3 February 2023; Received in revised form 15 April 2023; Accepted 7 June 2023
Available online xxxx

and daylight on non-contact rail system maintenance, which is compatible with technology 4.0. Dube et al. (2021) devised a novel technique for identifying cracks and counting their number on the rail surface, while Karaduman et al. (2020) employed image processing techniques to detect wear on the rail surface by removing shadows in railroad photographs. Banić et al. (2019) utilized edge and feature extraction techniques to determine the rails in the tracked railway using UAVs (Drones). Additionally, machine vision was used to detect surface defects (Min et al., 2018) and object detection to detect anomalies (Wang et al., 2021) or foreign objects on the railway track (Gasparini et al., 2020; Bhushan et al., 2017). Recently, computer vision and audio machine learning have also garnered attention in railway industries for feature extraction and analysis to optimize the system for better performance (Doshi, 2022; Rengel et al., 2022b). For instance, in Yuan et al. (2019), the authors proposed a novel automatic feature extraction, pre-processing, and analysis deep learning technique for early fault detection in railways. They followed three basic steps, namely, data pre-processing, feature extraction using spectrograms, and finally, classification model training based on feature and pre-processing datasets. Furthermore, updated research on fault detection on railway lines has been examined in a review study by Kou (2021). Hashmi et al. (2022) investigated a similar strategy by fusing conventional acoustic-based systems with deep learning models to enhance performance and reduce railway accidents. They used two CNN models – convolutional 1D and convolutional 2D – as well as one RNN model in this context. The layer of a deep learning model that produces spectrograms was used for on-the-fly feature extraction. Finally, visual examination was used in the current investigation to determine POI since no conventional approaches were used. The assessment of rail surface damage using computer vision and deep learning techniques, in addition to conventional ultrasonic and acceleration detection techniques, has the potential to significantly increase detection system effectiveness while decreasing inspection costs.

Recent research suggests that AI and data-driven algorithms using object detection have potential for identifying defects on railroad lines (Chen et al., 2008). In the past, some studies have used one-stage object detection models, such as YOLO (You Only Look Once), to detect rail, clips, and bolt defects (Wang et al., 2020), as well as surface defects such as cracks or irregularities on the railway track (Yanan et al., 2019). While the first paper compares different YOLO models, it does not differentiate between defective and non-defective elements, only paying attention to detecting unfaulty elements. The second paper, on the other hand, only studies the third version of YOLO and does not compare different models. Other studies have explored the use of CNN methods in this field, with Faster RCNN being the most commonly used model (Wei et al., 2019). While this model includes a classification of defective and non-defective elements, the study only examines one element, fasteners. Finally, AttnConv-Net has been used to validate the effectiveness of object detection models for maintenance and inspection (Wang et al., 2022). This paper focuses on comparing the results of this model with other published models to validate its usefulness. While object detection has shown promise for identifying railroad defects, further research is needed to evaluate its effectiveness across multiple types of defects and elements.

Novelty and contribution to knowledge

Regular inspections of railway tracks are crucial for ensuring safety by detecting physical flaws or design faults that could lead to serious incidents such as train derailments. However, the manual inspection approach is costly and outdated, and can result in significant downtime, especially under harsh weather conditions. In recent years, digital technologies have been explored to automate railway track inspections. Although object detection has been investigated as a potential solution (Padilla et al., 2020), there are research gaps to address. For example, there is a lack of studies that propose defects detection in railway tracks using newer object detection models, and the importance of data collection and classification is not highlighted in most research. Moreover, many studies only train models on defective elements and fail to classify non-defective elements, leading to limitations in their effectiveness.

This paper aims to address these gaps by proposing a method for defect detection in railway tracks using newer object detection models and emphasizing the importance of data collection and classification. The proposed approach aims to improve safety and reduce costs by detecting defects automatically based on image datasets. With the continuous advance in machine learning, there is a growing interest in introducing these tools in the railway sector. However, one of the biggest challenges in object detection is the need for sufficient data to train a programme, and the paper explores this challenge by comparing different models and datasets to find the most effective technique in terms of precision and recall. Moreover, the paper highlights the importance of a comparative analysis of these techniques with the same training and testing dataset. The significance of the proposed method lies in its ability to detect defects in railway tracks quickly and accurately, minimizing the risk of accidents and reducing costs associated with manual inspections. This research presents a contribution to knowledge in the field of object detection in railway track inspections and offers a new perspective on its implementation in different sectors. It also highlights the need to use more advanced and newer object detection models to improve the accuracy of railway track inspections.

The methodology employed in this study is depicted in Fig. 1 and can be described as follows. Firstly, the datasets were pre-processed and analysed to identify the most commonly occurring railway defects, which were then classified as either defective or non-defective. The selected image datasets were then used for training and testing the proposed model. Object detection models were chosen for verification, and the corresponding steps and algorithms were followed as illustrated in Fig. 1. Finally, the validation of these models was performed using the validation dataset, and the results obtained were compared for each model.

The outline of this paper is as follows: Section 1 is the introduction. Section 2 describes the datasets in this study including pre-processing, classification and labelling. Section 3 explained the methodologies used in this study. Section 4 presents the results and their critical analysis. Section 5 carries out the performance comparison of the proposed model, discussion and the limitation associated with it. Finally, Section 6 provides concluding remarks and future work.

2. Data

2.1. Data description

The data extracted to create the final dataset has been done from 3 different sources. Firstly, images are a collection from an open source (Anon, 2022) and field trips. Thus, with this dataset, 3 defects on the railway tracks were highlighted and these are broken clips (fasteners), rail track breaks and faulty railway fish plates and these defects are briefly described as follows:

➢ Broken clips: Clips are a kind of fastener in charge of fastening steel rail to the sleeper. In some cases, it has happened that the clip has broken or come loose and therefore cannot perform its function as shown in Fig. 2.

➢ Rail track breaks: There are two distinct examples of this type of defect as shown in Fig. 3. The first consists of train tracks that for any reason have been damaged and have ended up breaking, preventing the train from passing through the crack. The second consists of small cracks or deformities in the track joints. This type of defect is usually affected by the breaking of the fish plate.

➢ Faulty fish plates: Fish plates are metal plates that connect two rails by several bolts or spikes. These are considered defective when some of the bolts are missing, causing the element's inefficiency as shown in Fig. 4.

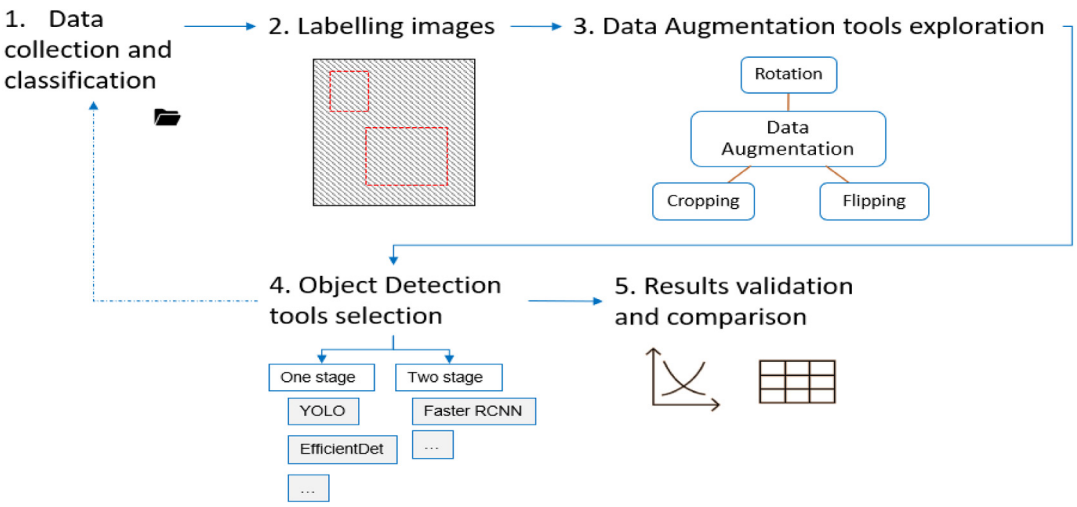


Fig. 1. Framework of the proposed methodologies for railway track fault detection techniques and their performance comparison.

Fig. 2. Correct clip v broken clip.

Fig. 3. Sample Rail track breaks.

The second source consists of images extracted by Network Rail showing defective rails and those that were consistent with the dataset built up to that point were selected. Finally, the last source of information is a set of images photographed of the Railway Innovation test area at Cranfield University by the author. A total of 329 images have been selected with a total of 467 annotations, as some of the images have more than one annotation. These images have been classified into 6 different classes, including faulty and unfaulty elements.

In short, the importance of data collection and classification cannot be overstated, as it is essential for building an accurate and reliable dataset for machine learning models. In this study, we took great care to ensure that the images collected were of high quality and represented a diverse range of real-world scenarios. Moreover, we classified the images into 6 different classes, including faulty and unfaulty elements, to provide clear differentiation between different types of defects and non-defective elements.

2.2. Data labelling and augmentation

Image datasets used in this study are coming from three different source but still not sufficient to train the proposed model, therefore, augmentation techniques employed to increase the size of datasets. In addition to that, rotation technique was selected to simulate how cameras could take pictures of the railway track from different angles

Fig. 4. Broken fishplate.

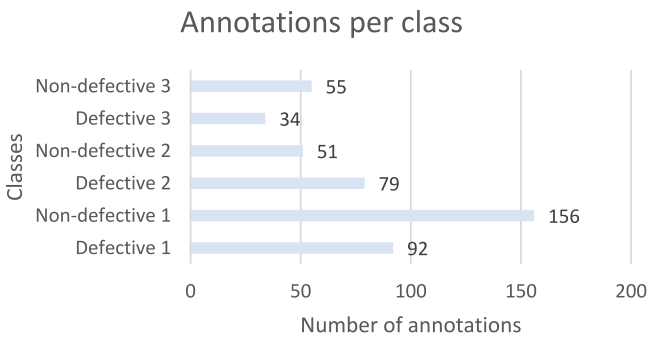


Fig. 5. Number of annotations for each class.

Table 1
Sample of object classes and definitions.

Class	Class name	Object detected	Number of annotations
1	Defective 1	Broken clips	92
2	Non-Defective 1	Non-broken clips	156
3	Defective 2	Rail Track breaks	79
4	Non-Defective 2	Common rail junctions	51
5	Defective 3	Faulty fishplates	34
6	Non-Defective 3	Correct fishplates	55

increasing then, the image dataset with more possible input data. It is very difficult to present images of correct railway tracks without cracks, as the mechanism would continuously detect an object throughout the whole railway track. For class 4 (non-defective 2) of Table 1, instead of choosing images of the rail where cracks and holes are not present, a class of holes that are correct on the railway tracks has been chosen. In the junctions of different tracks, where fishplates can be found, there exist gaps between them that are correct. In some cases, due to friction with the train, some of the track boundaries are cracked or the hole is larger than it should be, in these cases it is considered a defect and is classified as class 3 (Defective 2) of Table 1.

Fig. 5 shows the classes as a function of the number of annotations. As can be seen, for some of the classes there are not enough images, and this can affect the results as the machine cannot be correctly trained. Furthermore, Fig. A.8 of appendix shows object count per image. Further details of these datasets can be found in Anon (2022).

There are several tools available for labelling images. Labelling, which was created by python and uses QT as a graphical interface, has been used in this study (see Fig. 6). The following image shows an example of how this task has been done. For each image, the region where the figure is located must be drawn (with a bounding box) and the type of class has to be mentioned. The programme saves the given information in VOCXML or YOLOtxt format, this includes the object classification and location in annotation text. The images were labelled using Labelling and resized to square adding white margins to avoid cropping images and losing annotations during the process.

The next step is to apply data augmentation on datasets to increase the number of training data available and reduce overfitting problems. Considering that there is a limited number of images in the dataset due to limited sources, different image augmentation techniques were contemplated. Among the different techniques, only geometric transformations were tested, as the objective is to maintain the originality of the images. Some of these kinds of transformations include translation, cropping, rotation, flipping, colour space and noise. Among the different geometric techniques for image augmentation, rotation was the one chosen. The other ones can suppose an advantage for other research, but for this one, the most suitable one was to create new versions changing the perspective in which the camera can take a photo. In Fig. 7, an example of how the rotation technique applies to an image is shown. After applying this image augmentation technique, the number of images increased from 329 to 675, adding then 346 extra images after this step.

Initially, there were 329 images in total that had to be distributed between the training and the testing dataset. As this number is low and the machine must be trained with enough images to work properly, only 9% of the total number was selected for the testing dataset. This means that 298 images were selected to train the different models and the 31 remaining images to test the model's performance. These 31 images include a total of 44 annotations, as some of the images contain more than one annotation. After applying the image rotation technique, the training dataset increased its number from 298 to 644 available images. Further details on the analysis can be found in Appendix A.

2.3. Pre-processing

The final images for the dataset had different dimensions from each other and were mostly very large. The machine is more efficient and faster if it works with smaller images, so a resizing step of the images has been performed. As shown in Fig. 8, the average size was 4000 × 3000. To reduce the image size to a square, the image size was reduced to 416 × 416 (as shown in Fig. 9). The most suitable way to avoid distorting the images was to reduce them to square shape. For memory and resource reasons, a smaller size that the average size was chosen. The choice of 416 × 416 pixels was based on the model architecture and computational resources available. This size was found to provide a good balance between accuracy and efficiency for our specific task.

In order not to crop the margins of some of the photographs, thus losing some annotation information, white margins have been added so as not to distort the results. An example is shown in Fig. 10.

3. Methodologies

Object detection is a computer vision technique that aims to locate in a video or image the detected objects and classify them using bounding boxes (Xiao et al., 2020). Normally this bounding box is followed by a text and a value, the former names the class type and the latter quantifies the confidence of classification (in percentage). The traditional object detection pipeline follows 3 main stages: Informative Region Selection (1), Feature Extraction (2) and Classification (3) (Zhao et al., 2018). Precision and recall are the most commonly used metrics in general, Padilla et al. (2021) and also being used in this study to access the effectiveness of the model and datasets. Precision is defined as the fraction of detected items that are correct and recall as the fraction of correctly detected items among all that should have been detected. And, mathematically they are defined as follows,

$$Precision = \frac{\sum TP}{\sum TP + \sum FP} = \frac{\sum TP}{All\ detections} \quad (1)$$

$$Recall = \frac{\sum TP}{\sum TP + \sum FN} = \frac{\sum TP}{All\ ground\ Truths} \quad (2)$$

Fig. 6. Image labelling process.

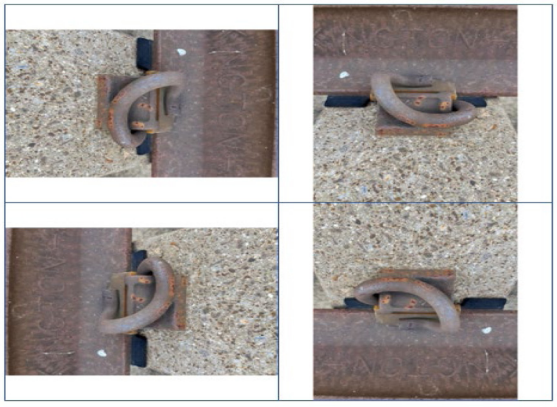


Fig. 7. Image 90° rotation.

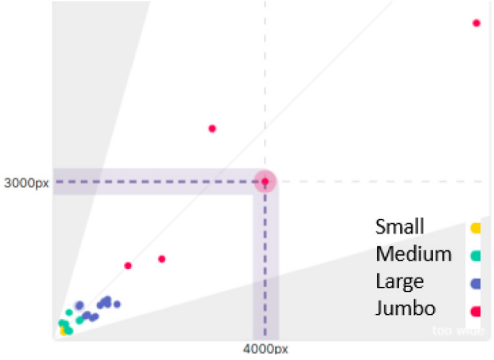


Fig. 8. Original image sizing.

Where,

- True positive (TP): Positive class predicted correctly (bounding box).
- False positive (FP): A non-positive class predicted as positive (false bounding box).
- False negative (FN): A non-detected positive class.

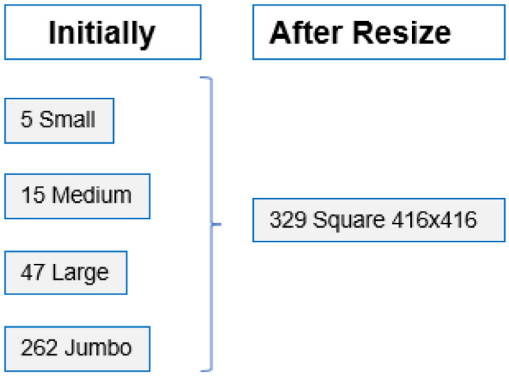


Fig. 9. Image resizing.

Furthermore, to compare the precision of the different classes, Average Precision (AP) is used. It is defined as the area that remains under the Precision-Recall curve varying the threshold (τ) accepted (Zhu et al., 2020).

$$Precision(\tau) = \frac{\sum TP(\tau)}{\sum TP(\tau) + \sum FP(\tau)} = \frac{\sum TP(\tau)}{All\ detections} \quad (3)$$

$$Recall(\tau) = \frac{\sum TP(\tau)}{\sum TP(\tau) + \sum FN(\tau)} = \frac{\sum TP(\tau)}{All\ ground\ Truths} \quad (4)$$

When the objective is to compare performance between detectors, the most common metric is mean Average Precision (mAP) which calculates performance considering all classes and is used as a unique metric for final evaluation (Zaidi et al., 2022).

There are different types of object detection models and for this research three advanced techniques namely, Faster RCNN, YOLO and EfficientDet are being used. YOLO and Faster RCNN are popular while EfficientDet is one of the best performing models in terms of accuracy and training speed. However, these techniques have never been explored in detail in railway industries to suggest the best technique among these techniques. All three models have been trained and tested using Google Colab which enables GPU use for faster results. A brief summary of these techniques is described as follows.

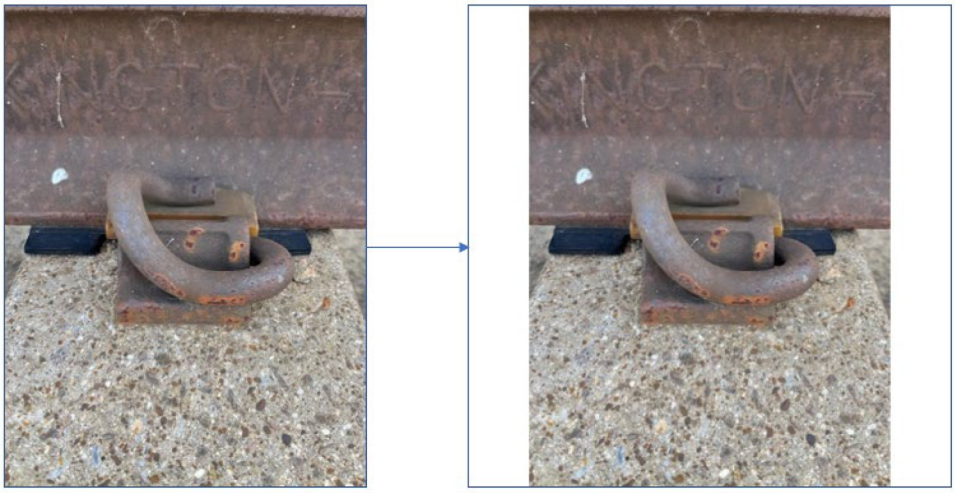


Fig. 10. White margins for final square images.

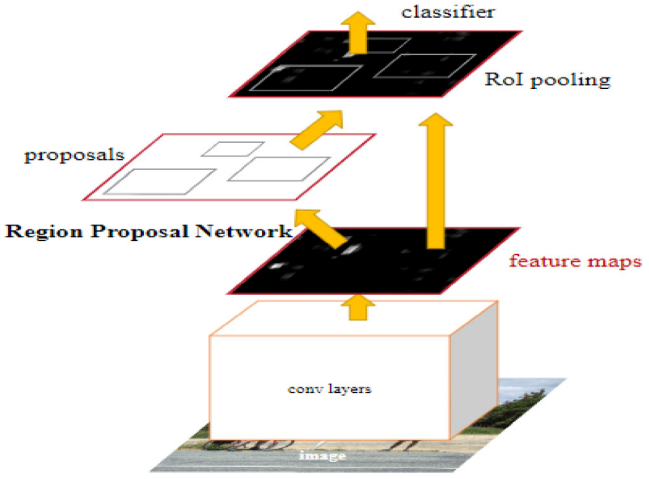


Fig. 11. Faster RCNN architecture (Ren et al., 2015).

3.1. Faster RCNN

Thanks to the incorporation of faster RCNN in CNNs, performance has been improved in terms of detection speed. Faster RCNN is a two-stage algorithm that uses the convolution network to generate the regions and boxes and shares it with the object detection network which reduces the number of possible frames by a large percentage (Salvador et al., 2016; Ren et al., 2015). As shown in Fig. 11, Faster RCNN follows 4 main parts of the structure, (Liu et al., 2017) and these are:

1. Convolutional layer: Extracts a feature map of the image.
2. RPN: The feature map is shared for subsequent Region Proposal Networks. It determines anchors while it generates a region proposal through border regression.
3. RoI pooling: Collects the feature map fixing the size by up-sampling, and the region proposal.
4. Classification and regression layer: The proposal feature maps are used to identify the class and adjust the regions of interest with bounding boxes.

3.2. You only look once version 5 (YOLOv5)

YOLO is a one-stage algorithm that predicts bounding boxes and performs class probabilities at the same time using an end-to-end neural network. It transforms the target detection problem into a regression problem solution by calculating the loss function of the classification prediction and location information (Xue-ping et al., 2019). There exist various versions of YOLO, but in this work, only v5 is contemplated as being one of the latest versions it is more flexible in terms of model sizing and data enhancement compared to the previous versions (Jiang et al., 2022). The architecture of YOLOv5 consists of four main elements (Li et al., 2022) and is described as follows.

1. Input terminal: Includes data pre-processing and data augmentation. Yolov5 can set the initial anchor frame size if the dataset is changed.
2. Backbone network: It extracts feature maps from the input image using cross-stage partial network and spatial pyramid pooling.
3. Neck network: Feature Pyramid Network (FPN) structure is used to convey semantic features and the Path Aggregation Network (PAN) structure conveys localization features. FPN structure does it from the top feature map to the lower feature map and PAN structures the other way around.
4. Output: Predicts the targets, applies bounding boxes and generates class probabilities and scores.

The output is encoded by dividing the input image into an SxS grid of cells. One grid cell oversees predicting the object in the image (the one in the centre of the object). Every grid cell predicts B bounding boxes, compounded by 5 elements (x, y, w, h, confidence), and C class probabilities (HackerNoon, 2018) where;

- (x,y): Centre of the box, relative to the grid cell location.
- (w,h): Width and height relative to the image dimensions.
- Confidence: Score that reflects the presence of the object in the image.

The output vector follows the following equation:

$$\boldsymbol{Output\ vector} = \boldsymbol{SxSx}(\boldsymbol{Bx}5 + \boldsymbol{C}) \tag{5}$$

3.3. EfficientDet

Being a state-of-art object detection model, EfficientDet is the last model considered for this project using the PyTorch implementation. EfficientDet follows the one-stage object detectors paradigm. Its structure consists of 3 main parts as shown in Fig. 12 namely, a backbone network, a feature network and a shared class/box prediction network (Tan et al., 2020) and explained as follows.

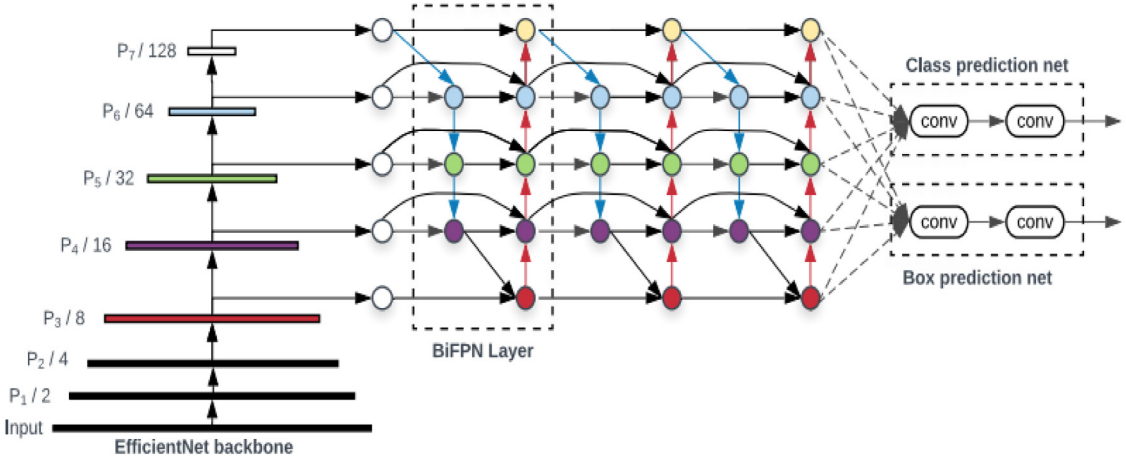


Fig. 12. EfficientDet Architecture, (Tan et al., 2020).

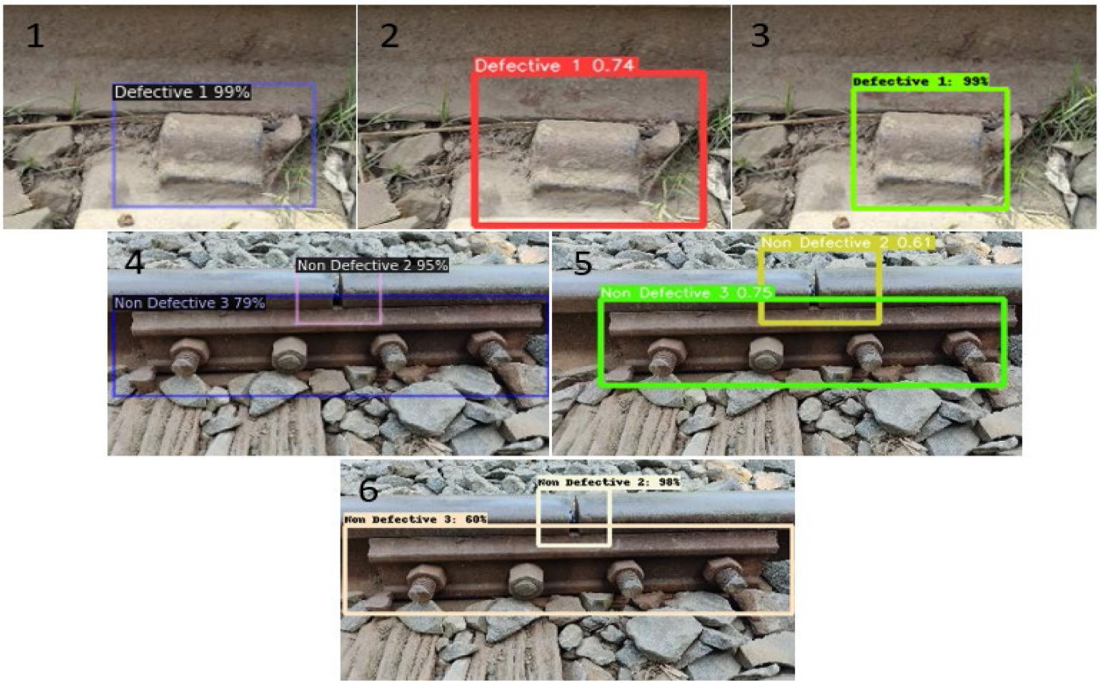


Fig. 13. (1&4) Faster RCNN, (2&5) Yolov5, (3&6) EfficientDet.

1. Backbone network: EfficientNet is employed as the backbone network which is designed to efficiently scale from the smallest model size.
2. Feature network: it extracts single images of any size as input and outputs sized feature maps. The Bi-directional Feature Pyramid Network (BiFPN) feature network is used, which is repeated multiple times.
3. Shared class/box prediction network: As the BiFPN layers, it is repeated multiple times based on different resource constraints.

The following equations show how the input image resolution, width and depth for BiFPN are scaled (Sahota, 2020). There exist 8 types of EfficientDet-Dφ, as φ can take values from 0 to 7. The one used in this project is EfficientDet-D0, being then:

$$\boldsymbol{R}_{input}(\boldsymbol{\varphi}=0)=512+\boldsymbol{\varphi}\cdot 128=512 \tag{6}$$

$$\boldsymbol{W}_{bifpn}(\boldsymbol{\varphi}=0)=64\cdot\left(1.35^{\boldsymbol{\varphi}}\right)=64 \tag{7}$$

$$\boldsymbol{D}_{bifpn}(\boldsymbol{\varphi}=0)=3+\boldsymbol{\varphi}=3 \tag{8}$$

The box/class prediction network used in EfficientDet is a softmax classifier. Although the width is the same as in BiFPN, the depth follows the following equation:

$$\boldsymbol{D}_{box}(\boldsymbol{\varphi}=0)=\boldsymbol{D}_{class}(\boldsymbol{\varphi}=0)=3+\frac{\boldsymbol{\varphi}}{3}=3 \tag{9}$$

4. Result and discussion

This section contains the results obtained after training the problem and passing the testing dataset. For each model, precision and recall have been calculated and the precision–recall curve for defective and non-defective annotations is represented. In addition, the mean average precision of the 3 models is shown. Some of the tested images are shown in Fig. 13.

4.1. You only look once (version 5)

For this first model, the precision and recall results are shown in the following Table 2. In general, the results of precision are higher than

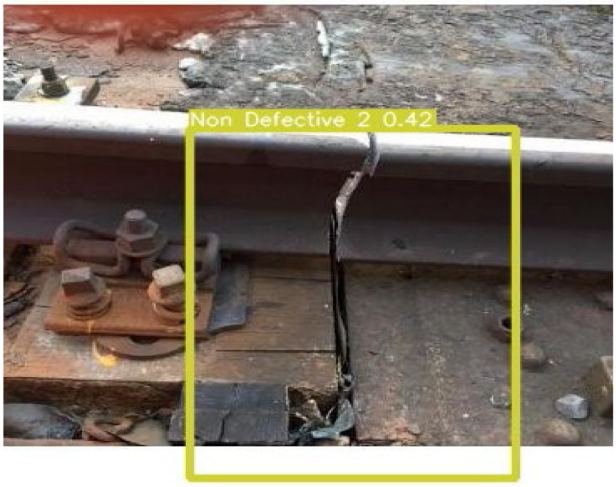


Fig. 14. Defective 2 wrongly classified as non-defective 2.

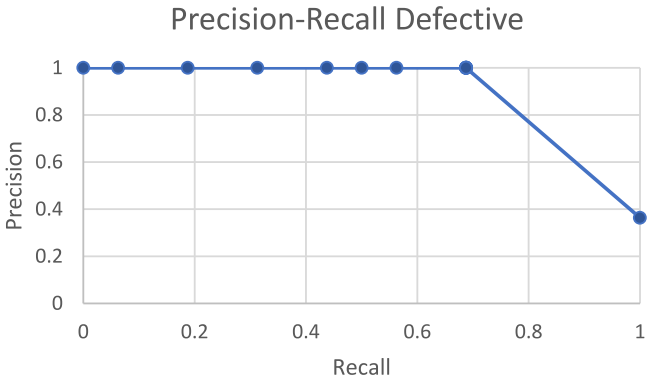


Fig. 15. Precision-Recall curve for defective annotations.

Table 2
Precision and recall (YOLOv5).

YOLOv5						
	D1	ND1	D2	ND2	D3	ND3
Precision	100%	100%	100%	75%	100%	100%
Recall	100%	95%	60%	75%	67%	80%

Table 3
Number of FP and FN.

False positive		False negative	
Defective	Non-defective	Defective	Non-defective
0	1	5	3

the ones obtained for recall, this means that the detections done by the model are highly correct compared to the number of annotations that should have been detected. In other words, the number of false positives is low compared to the number of false negatives.

Table 3 presents the number of false positives and false negatives in both categories, defective and non-defective. There has been one detected element as non-defective considered a false positive and 5 defective elements that were not detected (false negatives). Indeed, the one non-defective false positive is one of the five defective false negatives. This means that not only the machine h was unable to detect the defect, but it was also detected as unfaulty. Fig. 14 shows the image that the model was not able to detect correctly.

As the number of images in the dataset was low, the accuracy-recall curves have been made with two separate classes instead of six. The

Table 4
Precision and recall for each class.

Faster RCNN						
	D1	ND1	D2	ND2	D3	ND3
Precision	75%	100%	91%	100%	67%	100%
Recall	100%	100%	100%	100%	67%	80%

Table 5
Number of FP and FN.

False positive		False negative	
Defective	Non-defective	Defective	Non-defective
3	0	1	2

Table 6
Precision and recall for each class.

EfficientDet						
	D1	ND1	D2	ND2	D3	ND3
Precision	100%	95%	100%	100%	100%	100%
Recall	100%	100%	90%	75%	33%	60%

precision–recall curves of YOLOv5 are shown in Figs. 15 and 16. For this model, the number of false positives is low and constant when fixing different thresholds. The maximum value of recall maintaining precision to 1, is equal to 0.6875 for defective annotations and equal to 0.893 for non-defective annotations. For more information regarding precision–recall curves, and calculations go to Appendix B. The Average Precision of the defective class is equal to 0.9 and for the non-defective is 0.978. The mean Average Precision for YOLO is equal to 0.939.

4.2. Faster R-CNN

Precision and recall results for Faster RCNN are shown in Table 4. In contrast to the results obtained with yolov5, Faster RCNN shows mostly better results in Recall than in Precision. Therefore, contrary to previous results, in this case, the number of false positives is higher than the number of false negatives respectively for each class. More information about precision and recall calculations can be found in Appendix B.

Table 5 shows the total number of false positives and false negatives of defective and non-defective objects. In this case, there is only one worrying error case: the defective false negative. The machine has not been able to detect a defective class element. On the other hand, it can be verified that there are no non-defective false positives, therefore, at least the model has not detected that the element belongs to a non-defective class, it has simply not classified it in any of the two categories.

The following two graphs, Figs. 17 and 18 represent the Precision-Recall curves for Defective and Non-Defective classes.

For the defective curve, the maximum value of recall maintaining precision to 1, is 0.5625. Nevertheless, for a precision value of 0.938, recall raises to 0.938. For the non-defective curve, the recall value equals 0.929 when having perfect precision (equals 1). For the Defective class, the Average Precision has been equal to 0.945, and for the non-defective class equal to 0.987. The mean Average Precision considering the results of these two categories has been equal to 0.966.

4.3. EfficientDet

For this last model, EfficientDet, precision and recall results for each of the six classes are shown in Table 6. Similar to the results of Yolov5, the accuracy percentages are very high compared to those of recall. In fact, for the images tested for defect 3, the number of false negatives is higher respectively.

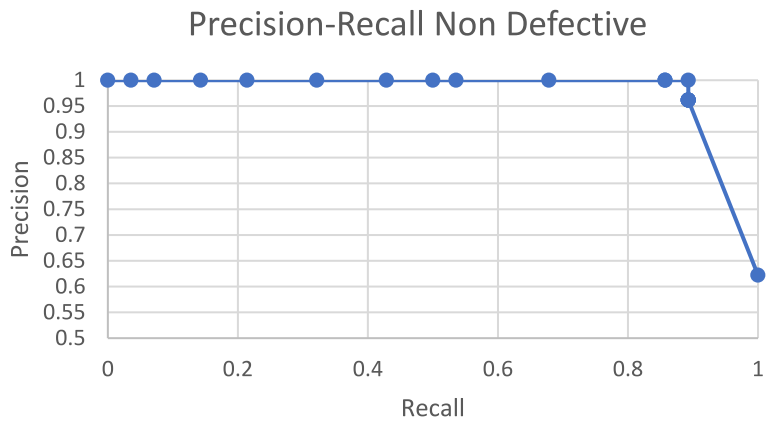


Fig. 16. Precision-Recall curve for non-defective annotations.

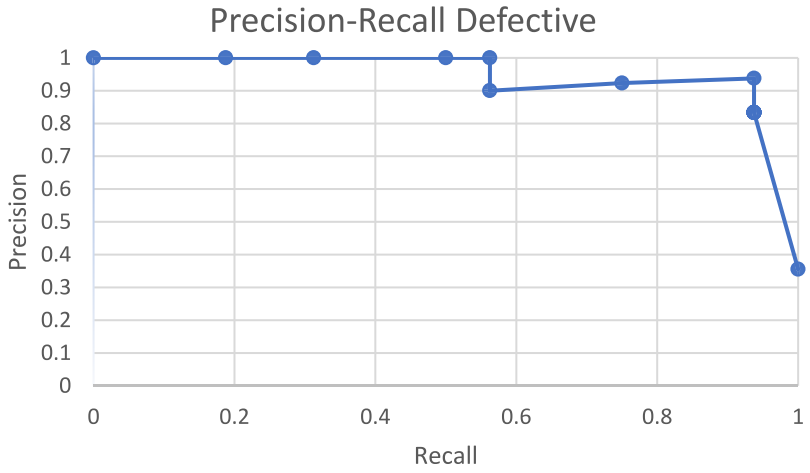


Fig. 17. Precision-Recall curve for defective annotations.

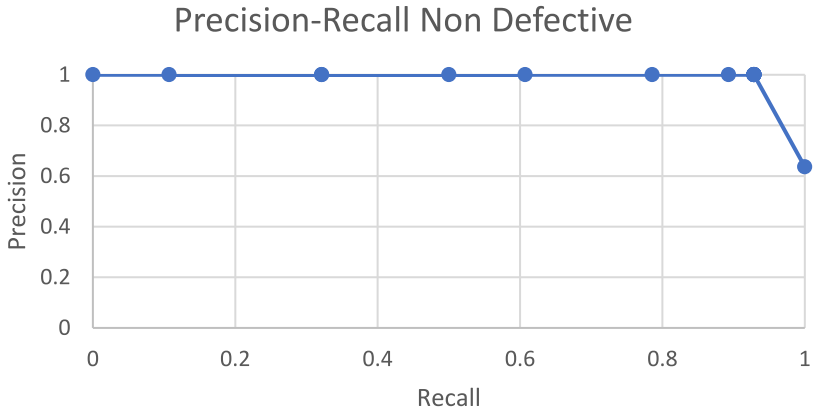


Fig. 18. Precision-Recall curve for non-defective annotations.

Table 7
Number of FP and FN.

False positive		False negative	
Defective	Non-defective	Defective	Non-defective
0	1	3	3

Table 7 shows that the total number of false positives is one and the total number of false negatives is 6. The only false positive (see Fig. 19) is classified as a non-defective but it is not one of the defective false negatives, thus not worsening the results. The other three defective false negatives, although not recognized, were at least not classified as non-defective.

In Figs. 20 and 21, Precision-Recall curves are shown. As it can be precision maintains constant and equal to 1 during most of the fixed thresholds. The maximum value of recall with perfect precision is equal to 0.8125 with a fixed accepted threshold between 0 and 80% for the Defective curve. For the non-defective curve, the maximum recall is equal to 0.89 for a precision of 1 when the threshold is fixed at 60%. More information regarding the precision–recall curves can be found in Appendix B.

Fig. 19. Non-defective false positive.

Table 8
Precision and recall results.

Methods		D1	ND1	D2	ND2	D3	ND3
YOLOv5	Precision	1	1	1	0.75	1	1
	Recall	1	0.95	0.6	0.75	0.67	0.8
Faster RCNN	Precision	0.75	1	0.91	1	0.67	1
	Recall	1	1	1	1	0.67	0.8
Efficient DetD0	Precision	1	0.95	1	1	1	1
	Recall	1	1	0.9	0.75	0.33	0.6

The Average Precision of defective equals 0.94 and 0.978 for non-defective. The mean Average Precision result for EfficientDet is 0.959.

5. Model performance comparison and discussion

The three models tested have proven to be able to obtain good results and detect defective elements on the railway tracks. In this section, a more detailed analysis of the results obtained for each model is discussed. The section includes the influence of the initial data on the results, advantages and limitations associated with this research.

5.1. False positives (FP) and False negatives (FN)

The level of confidence with which the different models have detected an element is variable. For FasterRCNN and EfficientDet, the average confidence levels are quite high compared to those obtained in YOLOv5. Fig. 22 shows the average confidence levels for each model for each class type without taking into account false positives and false negatives.

As shown in Fig. 22, the first four classes, EfficientDet leads the 3 models with an average confidence level of almost 100%. For the last two classes, on the other hand, FasterRCNN has the highest values. For all 6 classes, YOLOv5 has the lowest confidence levels. However, the confidence level is meaningless if it does not consider the false positives and false negatives that it implies. The number of false positives and false negatives for each model and class are shown in Fig. 23.

The following table shows the results obtained for precision and recall in each of the object detection models. During the evaluation of the 70 images, the total number of false positives and false negatives obtained, are shown in Table 9. The number of false negatives for the three models highlights the inefficiency in detecting elements in the images (see Table 8).

The Average Precision of the defective class has been equal to 0.9 for YOLOv5, 0.945 for FasterRCNN and 0.94 for EfficientDetD0. For the non-defective class, the average precision has been 0.978 for YOLOv5, 0.987 for FasterRCNN and 0.978 forisficientDetD0. The Mean average precision for the three models is shown in Table 10.

Table 9
Total number of false positives and false negatives.

Methods	False positive		False negative	
	Defective	Non-defective	Defective	Non-defective
YOLOv5	0	1	5	3
Faster RCNN	3	0	1	2
EfficientDetD0	0	1	3	3

Table 10
Mean Average Precision results.

Performance error metrics			
	YOLOv5	FasterRCNN	EfficientDetD0
Mean Average Precision (mAP)	0.939	0.966	0.959

Table 11
Thresholds for each model.

Methods	Threshold		Precision		Recall	
	D	ND	D	ND	D	ND
FasterRCNN	>=0.8	>0	0.9375	1	0.9375	0.9286
YOLOv5	>0	>=0.43	1	1	0.6875	0.8929
EfficientDet	>0	>=0.52	1	1	0.8125	0.8929

Table 12
False positives and negatives considering thresholds.

Methods	Threshold		False positives		False negatives	
	D	ND	D	ND	D	ND
FasterRCNN	>=0.8	>0	1	0	1	2
YOLOv5	>0	>=0.43	0	0	5	3
EfficientDet	>0	>=0.52	0	0	3	3

5.2. Thresholds

In order to improve the accuracy of the models used in this study, thresholds for each model were analysed to determine the settings that produced the best results. Specifically, the aim was to strike a balance between reducing the number of false positives and false negatives for both defective and non-defective items. Depending on the type of item being analysed, the focus was on either maximizing precision or recall. This allowed for the creation of more trusted results, which were shown in Table 11.

For defective items, it was crucial to minimize the number of false negatives in order to ensure that the model always alerted the user to the presence of a defect. Conversely, for non-defective items, the priority was to minimize the number of false positives in order to avoid erroneously categorizing these items as defective. The thresholds used to achieve these goals were presented in Table 11. Among the three models tested, Faster RCNN showed the best results when considering these priorities, with a precision of 1 and 0 false positives for non-defective items and a recall of 0.9375 with just one false negative for defective items, as shown in Table 12.

5.3. Data repercussion

The data used to train Yolov5, Faster RCNN and EfficientDet had a great impact on the results obtained. Being precision values high for most classes, the impact of the training dataset is not relevant. For recall, in contrast, the figure shows how the classes with lower images had lower values. In other words, the lower the number of images, the higher the number of false negatives. The following graphs compare the data used for the training and the results of precision (Fig. 24) and recall (Fig. 25) for each class and model considering accepted all the confidence levels.

Being precision values high for most classes, the impact of the training dataset is not relevant. For recall, in contrast, the figure shows how the classes with lower images had lower values. In other words, the lower the number of images, the higher the number of false negatives.

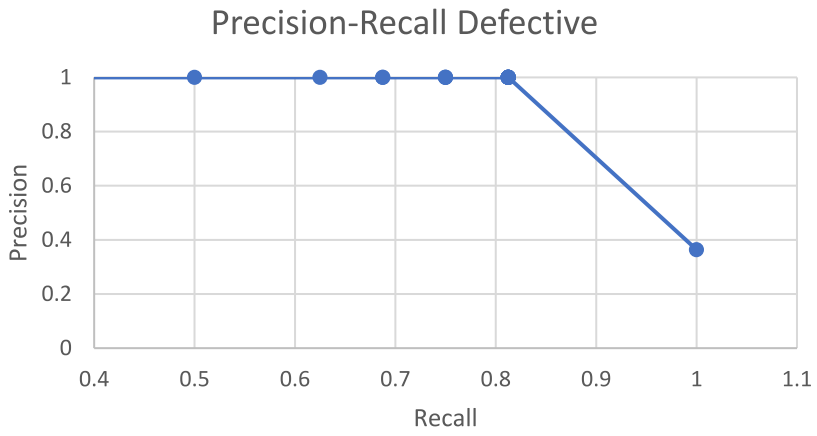


Fig. 20. Precision-Recall curve for defective annotations.

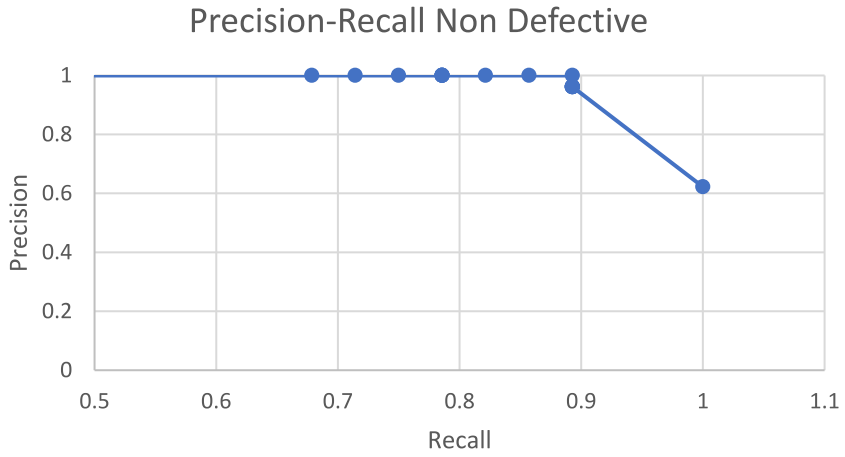


Fig. 21. Precision-Recall curve for non-defective annotations.

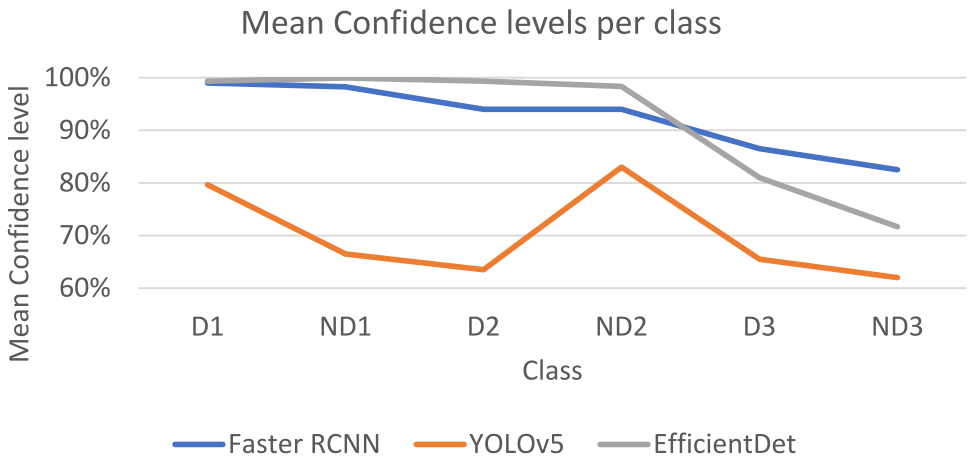


Fig. 22. Mean threshold per model.

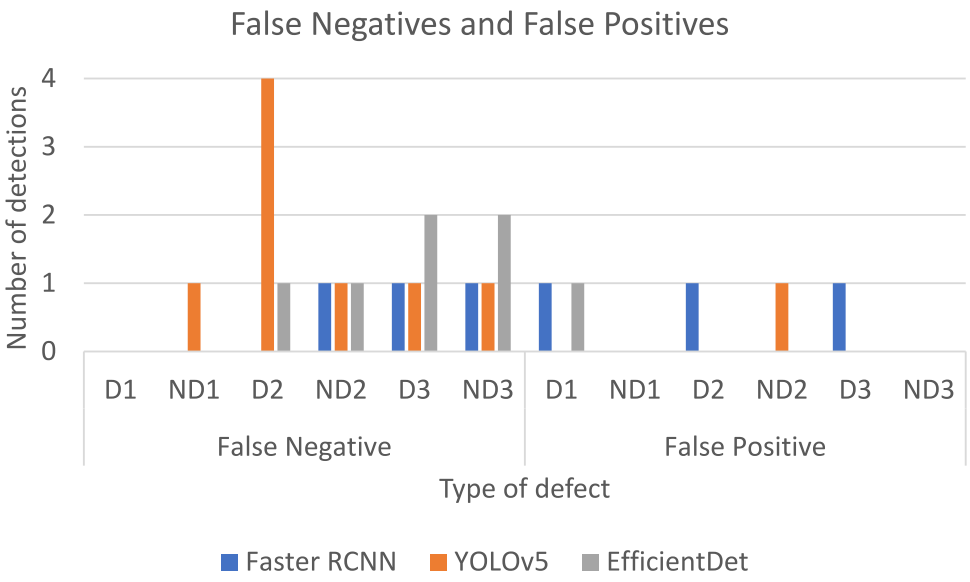


Fig. 23. FN and FP for class and model.

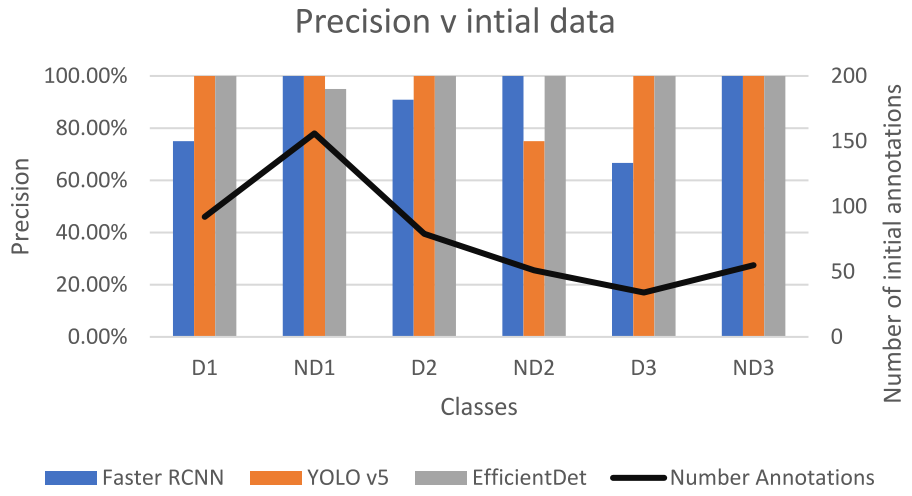


Fig. 24. Precision v original dataset.

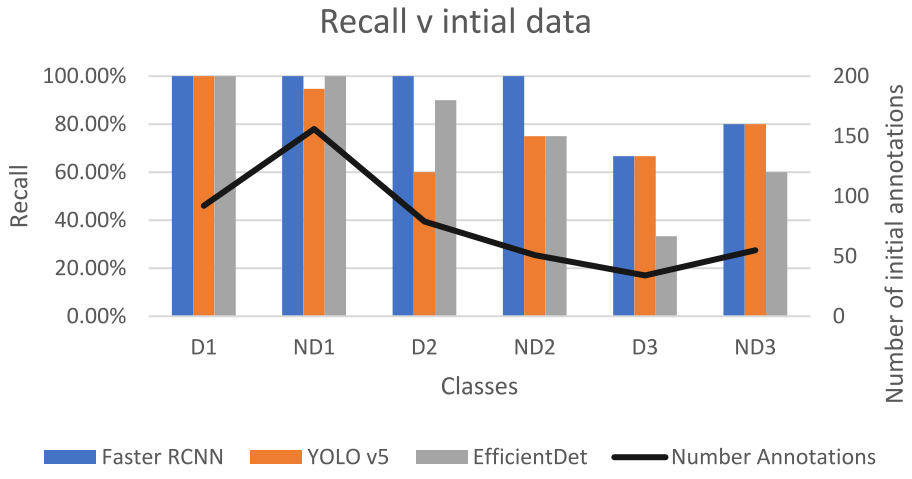


Fig. 25. Recall v original dataset.

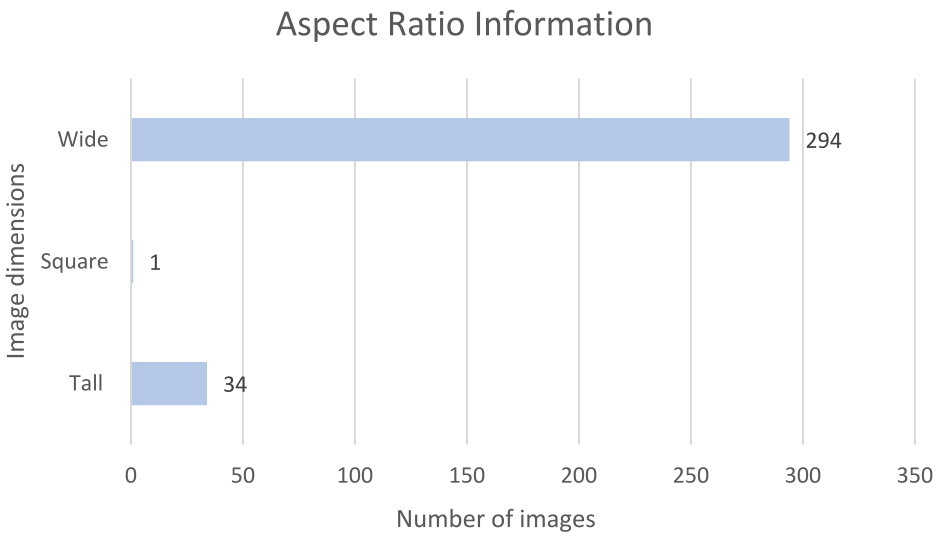


Fig. A.1. Aspect Ratio information.

5.4. Limitations

In this study, several limitations were identified and are described as follows. Firstly, it should be noted that the results obtained from the three models are based on a specific dataset. Changing this dataset could result in different outcomes than those obtained. It may be beneficial to add more images or replace some with others to improve the precision and recall obtained, especially since certain classes lack sufficient images. Additionally, this study only focuses on three types of defects found on railway tracks, and other types could be considered in future studies.

Furthermore, there are certain limitations to how the programs classify defective and non-defective elements. Classifying an item as defective does not necessarily mean that the program understands that it cannot be classified as non-defective, and vice versa. The models are trained to detect objects, and the designation of faulty or non-faulty is simply a label that is added during the training process. Therefore, the program may not understand that faulty and non-faulty are mutually exclusive categories and could detect that an object is both faulty and non-faulty at the same time, as it is simply a label. These limitations should be taken into consideration when interpreting the results of this study and when applying the models to other scenarios. There is a need to start a model which can be used for a binary and multi classification and can link that if one object is detected as "A" then that same object cannot be detected as "B".

6. Conclusions

The paper aimed to test object detection models to detect defects in railway tracks and thus considerably reduce the risk of manual inspection and maintenance. Due to this study, it has been possible to validate the effectiveness that object detection could have in the field of railway and specifically in maintenance and inspection. It should be noted that the lack of images of some of the defective elements has led to this low recall for some of the classes.

Faster RCNN has proven to be the most effective model for this dataset in terms of accuracy and recall. In addition, the importance of a dataset with enough images has been demonstrated, since if this is not the case, the results obtained by one model, or another may be affected.

To be able to improve some of the results or to continue advancing with this study, different future projects have been proposed to implement object detection techniques in the railway sector. Firstly, due to the lack of faulty railway track data, it would be interesting to be able to extend the work confidentially to a company that can provide this information. With more images, the dataset could be expanded, and more accurate and effective results could be obtained. With a wider range of photographs, the classification presented so far could be more detailed. For example, for the fishplate defects, instead of considering the lack of screws as the same defect, different classes could be made for the lack of one or more of the four screws, thus training the programme more effectively. In addition, other defects have not been considered in this project, which would also be interesting to comment on. On the other hand, it should be noted that following the risk, further investigation should be carried out to optimize the automated process. Using cameras or robots on the train tracks, these images could automatically pass through a programme that detects whether the elements are defective or not.

Future work involves exploring meta-heuristic algorithms, such as the grey wolf optimization algorithm, whale optimization algorithm, and sparrow search algorithm, to solve this problem. Additionally, compared to binary classification, incorporating multi-classification analysis has the potential to significantly enhance the accuracy of the proposed model. Therefore, this aspect has been identified as a topic for further research.

CRediT authorship contribution statement

Marta Garcia Minguell: Conceptualization, Methodology, Software, Data curation, Writing – original draft. **Ravi Pandit:** Visualization, Investigation, Supervision, Software, Validation, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data that has been used is confidential.

Appendix A. Extra image dataset information

A.1. Aspect ratio of original images

See Fig. A.1.

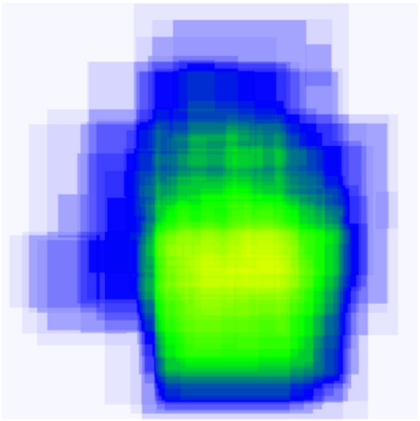


Fig. A.2. Defective 1 heatmap.

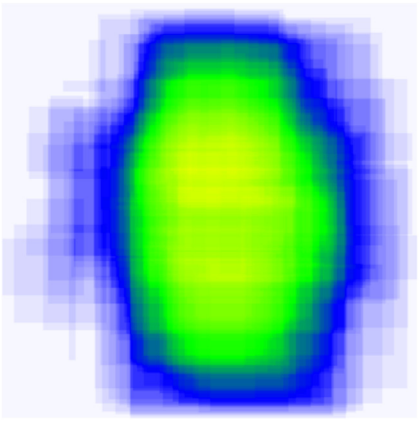


Fig. A.3. Non-defective 1 heatmap.

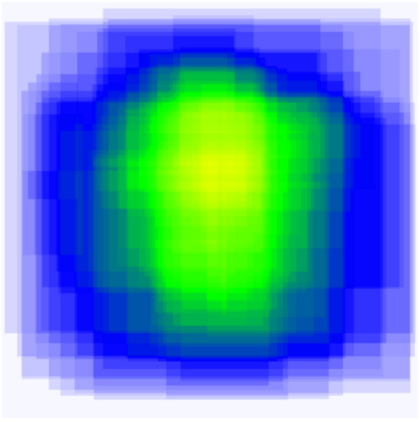


Fig. A.4. Defective 2 heatmap.

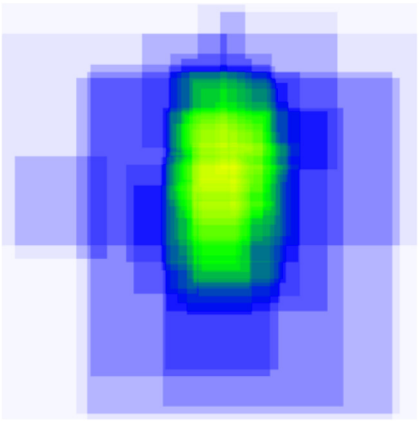


Fig. A.5. Non-defective 2 heatmap.

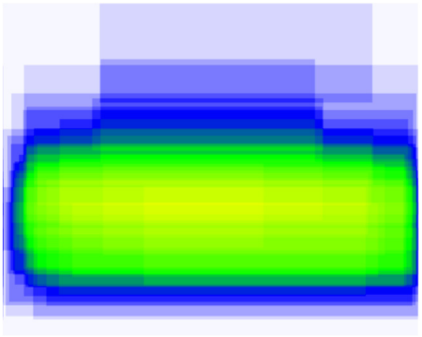


Fig. A.6. Defective 3 heatmap.

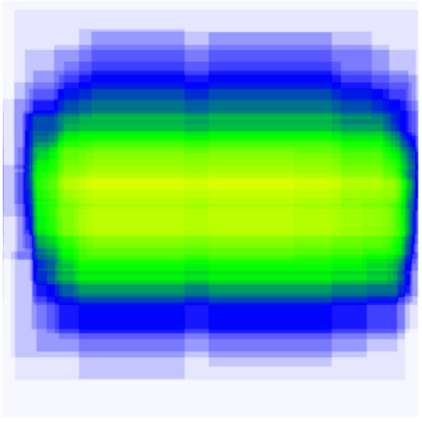


Fig. A.7. Non-defective 3 heatmap.

A.2. Annotations heatmap for each class (https://roboflow.com/)

See Figs. A.2–A.7.

A.3. Annotations per image

Fig. A.8.

Table B.1
Expected results v obtained in the models.

	Results						
	D1	ND1	D2	ND2	D3	ND3	Total
Real Boxes	3	19	10	4	3	5	44
Faster RCNN	4	19	11	3	3	4	44
Yolo v5	3	18	6	4	2	4	37
EfficientDet	3	20	9	3	1	3	39

Appendix B. Precision and recall calculations

See Tables B.1–B.4.

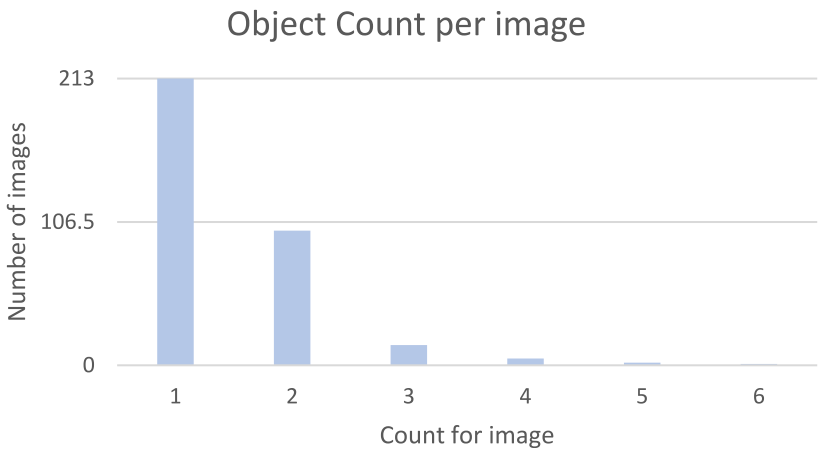


Fig. A.8. Annotation count per image.

Table B.2
Total true positives.

	True positives						
	D1	ND1	D2	ND2	D3	ND3	Total
Faster RCNN	3	19	10	3	2	4	41
Yolo v5	3	18	6	3	2	4	36
EfficientDet	3	19	9	3	1	3	38

Table B.3
Total false negatives.

	False postives						
	D1	ND1	D2	ND2	D3	ND3	Total
Faster RCNN	1		1		1		3
YOLO v5				1			1
EfficientDet		1					1

Table B.4
Total false positives.

	False negatives (not detected)						
	D1	ND1	D2	ND2	D3	ND3	Total
Faster RCNN	0	0	0	0	1	1	3
Yolo v5	0	1	4		1	1	8
EfficientDet	0	0	1	1	2	2	6

References

Anon, 2020. RAILS deliverable d 1.1: Definition of a reference taxonomy of AI in railways.

Anon, 2021. Railway Safety Statistics in the EU - Statistics Explained. Ec.europa.eu, https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Railway_safety_statistics_in_the_EU#Continued_fall_in_the_number_of_railway_accidents, (Accessed on 19th Aug 2022).

Anon, 2022. Kaggle railway track datasets. Available on https://www.kaggle.com/datasets/salmaneunus/railway-track-fault-detection. (Accessed on 1st July 2022).

Banić, M., Miltenović, A., Pavlović, M., Ćirić, I., 2019. Intelligent machine vision based railway infrastructure inspection and monitoring using UAV. Facta Univ. Ser.: Mech. Eng. 17 (3), 357–364.

Bhushan, M., et al., 2017. Automated vehicle for railway track fault detection. IOP Conf. Ser.: Mater. Sci. Eng. 263, 052045. http://dx.doi.org/10.1088/1757-899X/263/5/052045.

Chen, J., Roberts, C., Weston, P., 2008. Fault detection and diagnosis for railway track circuits using neuro-fuzzy systems. Control Eng. Pract. (ISSN: 0967-0661) 16 (5), 585–596. http://dx.doi.org/10.1016/j.conengprac.2007.06.007.

Doshi, K., 2022. Audio deep learning made simple (part 1): State-of-the-art techniques. Available online: https://towardsdatascience.com/audio-deep-learning-made-simple-part-1-state-of-the-art-techniques-da1d3dff2504 (Accessed on 1st September 2022).

Dube, U., Subramaniam, S., Subramaniam, S., 2021. Cost effective railway track fault detection using image processing.

European Union Agency for Railway, 2022. Railway Safety and Interoperability: the 2022 Report.

Gasparini, R., et al., 2020. Anomaly detection, localization and classification for railway inspection. In: 2020 25th International Conference on Pattern Recognition. ICPR, http://dx.doi.org/10.1109/ICPR48806.2021.9412972.

Ghosh, C., Verma, A., Verma, P., 2021. Real time fault detection in railway tracks using fast Fourier transformation and discrete wavelet transformation. Int. J. Inf. Technol. 14, 31–40.

HackerNoon, 2018. Understanding YOLO. https://hackernoon.com/understanding-yolo-f5a74bbc7967, (Accessed Aug. 25, 2022).

Hashmi, M.S.A., Ibrahim, M., Bajwa, I.S., Siddiqui, H.U.R., Rustam, F., Lee, E., Ashraf, I., 2022. Railway track inspection using deep learning based on audio to spectrogram conversion: An on-the-fly approach. Sensors 22, 1983.

Jiang, P., Ergu, D., Liu, F., Cai, Y., Ma, B., 2022. A review of yolo algorithm developments. Proc. Comput. Sci. 199, 1066–1073. http://dx.doi.org/10.1016/J.PROCS.2022.01.135.

Karaduman, G., Karakose, M., Aydin, I., Akin, E., 2020. Contactless rail profile measurement and rail fault diagnosis approach using featured pixel counting. Intell. Autom. Soft Comput. 26 (3), 455–463.

Karakose, M., Yaman, O., 2020. Complex fuzzy system based predictive maintenance approach in railways. IEEE Trans. Ind. Inform. 16 (9), 6023–6032.

Kou, L., 2021. A review of research on detection and evaluation of rail surface defects. EasyChair 7244, 20.

Li, Z., Tian, X., Liu, X., Liu, Y., Shi, X., 2022. A two-stage industrial defect detection framework based on improved-YOLOv5 and optimized-inception-ResnetV2 models. Appl. Sci. 12 (2), 834. http://dx.doi.org/10.3390/APP12020834.

Liu, B., Zhao, W., Sun, Q., 2017. Study of object detection based on faster R-CNN. In: 2017 Chinese Automation Congress. CAC, pp. 6233–6236.

Min, Y., Xiao, B., Dang, J., Yue, B., Cheng, T., 2018. Real time detection system for rail surface defects based on machine vision. EURASIP J. Image Video Process. http://dx.doi.org/10.1186/s13640-017-0241-y.

Najeh, T., Lundberg, J., Kerrouche, A., Chatterton, S., 2021. Deep-learning and vibration-based system for wear size estimation of railway switches and crossings. Sensors 21, http://dx.doi.org/10.3390/s21155217.

Padilla, R., Netto, S.L., da Silva, E.A.B., 2020. A survey on performance metrics for object-detection algorithms. In: 2020 International Conference on Systems, Signals and Image Processing. IWSSIP, pp. 237–242.

Padilla, R., Passos, W.L., Dias, T.L.B., Netto, S.L., da Silva, E.A.B., 2021. A comparative analysis of object detection metrics with acompanion open-source toolkit. Electronics (Basel) http://dx.doi.org/10.3390/electronics10030279.

Ren, S., He, K., Girshick, R., Sun, J., 2015. 'Faster R-CNN: Towards real-time object detection with region proposal networks'. IEEE Trans. Pattern Anal. Mach. Intell. 39 (6), 1137–1149. http://dx.doi.org/10.48550/arxiv.1506.01497.

Rengel, J., Santos, M., Pandit, R., 2022a. EfficientNet architecture family analysis on railway track defects. In: Yin, H., Camacho, D., Tino, P. (Eds.), Intelligent Data Engineering and Automated Learning – IDEAL 2022. IDEAL 2022. In: Lecture Notes in Computer Science, Vol. 13756, Springer, Cham, http://dx.doi.org/10.1007/978-3-031-21753-1_46.

Rengel, J., Santos, M., Pandit, R., 2022b. EfficientNet architecture family analysis on railway track defects. In: Yin, H., Camacho, D., Tino, P. (Eds.), Intelligent Data Engineering and Automated Learning – IDEAL 2022. IDEAL 2022. In: Lecture Notes in Computer Science, Vol. 13756, Springer, Cham, http://dx.doi.org/10.1007/978-3-031-21753-1_46.

Rosyidi, M., et al., 2022. Predictive maintenance with PCA approach for multi automated railroad crossing system (ARCS) in the framework of prognostic and health management (PHM) planning. J. Phys.: Conf. Ser. 2322, 012090. http://dx.doi.org/10.1088/1742-6596/2322/1/012090.

Sahota, H., 2020. Google's EfficientDet: An overview. Towards Data Science Accessed: Aug. 25, 2022. [Online]. Available: https://towardsdatascience.com/googles-efficientdet-an-overview-8d010fa15860, (Accessed on 1st September 2022).

Salvador, A., Giro-I-Nieto, X., Marques, F., Satoh, S., 2016. Faster R-CNN features for instance search. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops. pp. 394–401. http://dx.doi.org/10.48550/arxiv.1604.08893.

Sanger, Slack Savis, 2018. Railway Accidents Causes and Safety Measures. SDS Blog, https://www.slackdavis.com/blog/railway-accidents-causes-and-safety-measures/, (Accessed on 19th Aug 2022).

Shah, A.A., Bhatti, N.A., Dev, K., Chowdhry, B.S., 2021. MUHAFIZ: Iot-based track recording vehicle for the damage analysis of the railway track. IEEE Internet Things J. 8 (11), 9397–9406. http://dx.doi.org/10.1109/JIOT.2021.3057835.

Sresakoolchai, J., Kaewunruen, S., 2022. Railway defect detection based on track geometry using supervised and unsupervised machine learning. Struct. Health Monitor. 21 (4), 1757–1767. http://dx.doi.org/10.1177/14759217211044492.

Tan, M., Pang, R., Le, Q.v., 2020. EfficientDet: Scalable and efficient object detection. CVPR2020, Accessed: Aug. 22, 2022. [Online]. Available: https://github.com/google/automl/tree/.

Tang, R., et al., 2022. A literature review of artificial intelligence applications in railway systems. Transp. Res. C 140, http://dx.doi.org/10.1016/J.TRC.2022.103679.

Wang, T., Yang, F., Tsui, K.-L., 2020. Real-time detection of railway track component via one-stage deep learning networks. Sensors 15, http://dx.doi.org/10.3390/s20154325.

Wang, T., Zhang, Z., Yang, F., Tsui, K.-L., 2021. Intelligent railway foreign object detection: A semi-supervised convolutional autoencoder based method. IEE Sens. http://dx.doi.org/10.48550/arXiv.2108.02421.

Wang, T., Zhang, Z., Yang, F., Tsui, K.-L., 2022. Automatic rail component detection based on AttnConv-net. IEE Sens. 22 (3), http://dx.doi.org/10.1109/JSEN.2021.3132460.

Wei, X., Yang, Z., Liu, Y., Wei, D., Jia, L., Li, Y., 2019. Railway track fastener defect detection based on image processing and deep learning techniques: A comparative study. Eng. Appl. Artif. Intell. 80, 66–81. http://dx.doi.org/10.1016/J.ENGAPPAI.2019.01.008.

Xiao, Y., et al., 2020. A review of object detection based on deep learning. Multimed Tools Appl http://dx.doi.org/10.1007/s11042-020-08976-6.

Xue-ping, L., et al., 2019. Improved YOLOV3 target recognition algorithm for adaptive edge optimization. Microelectron. Comput. 36 (7), 59–64. http://dx.doi.org/10.19304/J.ISSN1000-7180.2021.1274.

Yanan, S., Hui, Z., Li, L., Hang, Z., 2019. Rail Surface Defect Detection Method Based on YOLOv3 Deep Learning Networks. IEEE, http://dx.doi.org/10.1109/CAC.2018.8623082.

Yuan, M., Li, J., Liu, Y., Gao, X., 2019. Automatic recognition and positioning of wheel defects in ultrasonic B-scan image using artificial neural network and image processing. J. Test. Eval. 48, 308–322.

Zaidi, S.S.A., Ansari, M.S., Aslam, A., Kanwal, N., Asghar, M., Lee, B., 2022. A survey of modern deep learning based object detection models. Digit. Signal Process.: Rev. J. 126, http://dx.doi.org/10.1016/J.DSP.2022.103514.

Zang, Y., Shangguan, W., Cai, B., Wang, H., Pecht, M.G., 2019. Methods for fault diagnosis of high-speed railways: A review. Proc. Inst. Mech. Eng. O 233 (5), 908–922. http://dx.doi.org/10.1177/1748006X18823932.

Zhao, Z.-Q., Zheng, P., Xu, S.-T., Wu, X., 2018. Object detection with deep learning: A review. IEEE Trans. Neural Netw. Learn. Syst. 1–21, http://dx.doi.org/10.1109/TNNLS.2018.2876865.

Zhu, H., Wei, H., Li, B., Yuan, X., Kehtarnavaz, N., 2020. A review of video object detection: Datasets, metrics and methods. Appl. Sci. http://dx.doi.org/10.3390/app10217834.

2023-06-30

TrackSafe: a comparative study of data-driven techniques for automated railway track fault detection using image datasets

Garcia Minguell, Marta

Elsevier

Minguell MG, Pandit R. (2023) TrackSafe: a comparative study of data-driven techniques for automated railway track fault detection using image datasets. Engineering Applications of Artificial Intelligence, Volume 125, October 2023, Article number 106622
https://doi.org/10.1016/j.engappai.2023.106622