


Article

# Lightweight Convolutional Network with Integrated Attention Mechanism for Missing Bolt Detection in Railways

Mujadded Al Rabbani Alif \* and Muhammad Hussain 

Department of Computer Science, Huddersfield University, Queensgate, Huddersfield HD1 3DH, UK; m.hussain@hud.ac.uk

\* Correspondence: u2276977@unimail.hud.ac.uk

**Abstract:** Railway infrastructure safety is a paramount concern, with bolt integrity being a critical component. In the realm of railway maintenance, the detection of missing bolts is a vital task that ensures the stability and safety of tracks. Traditionally, this task has been approached through manual inspections or conventional automated methods, which are often time-consuming, costly, and prone to human error. Addressing these challenges, this paper presents a state-of-the-art solution with the development of a lightweight convolutional neural network (CNN) featuring an integrated attention mechanism. This novel model is engineered to be computationally efficient while maintaining high accuracy, making it particularly suitable for real-time analysis in resource-constrained environments commonly found in railway inspections. The proposed CNN utilises a distinctive architecture that synergises the speed of lightweight networks with the precision of attention-based mechanisms. By integrating an attention mechanism, the network selectively concentrates on regions of interest within the image, effectively enhancing the model's capability to identify missing bolts with remarkable accuracy. Comprehensive testing showcases a remarkable 96.43% accuracy and an impressive 96 F1-score, substantially outperforming existing deep learning frameworks in the context of missing bolt detection. Key contributions of this research include the model's innovative attention-integrated approach, which significantly reduces the model complexity without compromising detection performance. Additionally, the model offers scalability and adaptability to various railway settings, proving its efficacy not just in controlled environments but also in diverse real-world scenarios. Extensive experiments, rigorous evaluations, and real-time deployment results collectively underscore the transformative potential of the presented CNN model in advancing the domain of railway safety maintenance.



**Citation:** Alif, M.A.R.; Hussain, M. Lightweight Convolutional Network with Integrated Attention Mechanism for Missing Bolt Detection in Railways. *Metrology* **2024**, *4*, 254–278. <https://doi.org/10.3390/metrology4020016>

Academic Editor: David Henry

Received: 28 March 2024

Revised: 3 May 2024

Accepted: 8 May 2024

Published: 10 May 2024



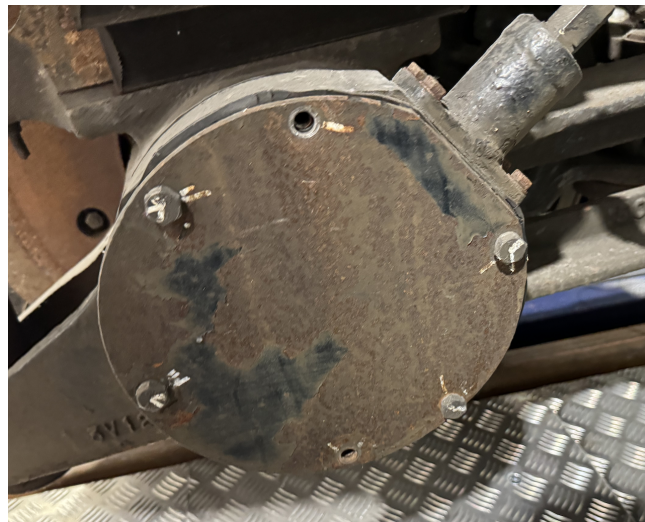
**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Keywords:** lightweight convolutional neural network (CNN); integrated attention mechanism; missing bolt detection; railway infrastructure safety; real-time analysis; safety inspection automation; high-accuracy detection systems; deep learning in real-world scenarios

## 1. Introduction

Railway maintenance and safety are foundational to the rail industry, ensuring the reliable and safe transportation of goods and passengers [1]. The railway system comprises a complex network of components that necessitate regular inspection and maintenance to avert malfunctions and accidents. Among the vast array of elements in this infrastructure, the integrity of fastening components, such as bolts, is crucial for maintaining track stability and alignment. Bolts, as seen in Figure 1, serve a pivotal role in securing rails to sleepers and joining rail sections, which, if compromised, can precipitate catastrophic failures, leading to derailments or other severe accidents [2]. Over time, bolts can become loose or go missing due to vibration, thermal expansion, and contraction, as well as mechanical stress, which can lead to the misalignment of tracks, the reduced functionality of the joints, and, in the worst cases, derailments [3]. Thus, the detection and timely replacement of such bolts are critical. This approach requires not just accuracy but also swiftness to preclude

the development of hazardous conditions. The traditional approach to bolt detection, involving manual inspections, is vulnerable to human error and becomes less practical with the ever-expanding scale of modern rail networks. Automated bolt detection methods promise greater consistency and efficiency. However, their efficacy is dependent on the sophistication of the detection technology used. In this context, the development and implementation of advanced detection systems capable of quickly and reliably identifying missing or damaged bolts is crucial. Despite advances in technology and the development of automated systems for maintenance checks, the detection of missing bolts remains a challenge due to the complexity of the environment in which railways operate, which encompasses factors like varying light conditions, weather influences, and the presence of dirt and grease [4]. Moreover, with the expansion of railway networks and a concurrent increase in traffic, the demand for more efficient and reliable maintenance techniques has escalated. Thus, the advent of a lightweight, attention-enhanced convolutional neural network represents a significant innovation in the field, tackling the urgent need for improved safety measures and maintenance practices in the railway industry. Such a system, by enhancing detection accuracy and efficiency, could represent a significant advancement in railway maintenance protocols, directly contributing to the safety and reliability of rail transport [5,6].



**Figure 1.** Real-life example of a bolt used in railway infrastructure.

With the advent of traditional machine learning techniques, automation has attempted to address the drawbacks of manual inspections. While these methods increase scalability, they still demand extensive feature engineering and are often constrained by the quality and quantity of data. Machine learning models rely on predefined features extracted from the data, which may fail to capture the complexity of real-world scenarios, resulting in suboptimal detection accuracy [7]. The adoption of deep learning approaches, notably convolutional neural networks (CNNs) [8], constitutes a significant step forward, as they offer end-to-end learning capabilities and automatic feature extraction. CNNs have demonstrated remarkable success in various computer vision tasks, including object detection and classification [9]. For instance, CNNs have been widely adopted for pedestrian detection [10], face recognition [11], handwriting detection [12,13], and defect detection in manufacturing [14]. In the domain of transportation, CNNs have been employed for vehicle detection and classification [15], road sign recognition [16], and crack detection in roads and bridges [17]. However, deep learning models, particularly those with deeper architectures, require considerable computational resources, which may hinder their deployment in resource-constrained environments. Other deep learning architectures, such as Vision Transformers (ViTs) [18] and You Only Look Once (YOLO) [19], have also gained popularity for object detection tasks. ViTs leverage self-attention mechanisms to capture

long-range dependencies in the input data, while YOLO is a real-time object detection system that combines object localisation and classification into a single neural network. For example, ViTs have been successfully applied to medical image analysis tasks, such as tumour detection and segmentation [20], pallet rack defect detection [21], as well as remote sensing applications, including land cover classification and change detection [22]. On the other hand, YOLO has been widely adopted in surveillance systems for real-time object tracking [23], as well as in autonomous driving for pedestrian and vehicle detection [24]. Other notable examples of deep learning architectures for object detection include Faster R-CNN [25], which combines a region proposal network with a CNN for accurate object localisation and classification, and single-shot detectors like SSD [26] and RetinaNet [27], which perform object detection in a single pass over the input image, making them computationally efficient. Nevertheless, these models often have high computational complexity and memory requirements, which may pose challenges for real-time applications in railway maintenance. This necessitates balancing model performance and the feasibility of deployment in the field, where computational constraints are prevalent. Furthermore, these models' limited interpretability and "black box" nature can undermine trust and adoption in safety-critical applications [28]. Additionally, the effectiveness of deep learning models relies heavily on the availability of large annotated datasets, which are scarce in the domain of railway maintenance. The diversity of bolt types and the infrequency of missing bolts present further challenges in dataset acquisition. Hence, despite advancements in automated bolt detection, there remains a pressing need for a solution that provides high detection accuracy while being resource-efficient and interpretable for real-world application in railway maintenance.

To address these pressing limitations, we propose a groundbreaking solution: a lightweight convolutional neural network (CNN) with an integrated attention mechanism. This innovative architecture, based on supervised learning, is designed to address the dual challenges of accuracy and computational efficiency in the detection of missing bolts in railway infrastructure. Our design philosophy behind the lightweight CNN is to refine network complexity while enhancing the model's predictive power. By employing a more compact and optimised set of layers, our CNN operates with fewer computational resources, which is ideal for on-site deployment where hardware limitations must be considered. The attention mechanism elevates the model's performance by focusing on the most salient features of the input data, mimicking the human visual system's ability to concentrate on pertinent parts of an image to extract information. Our CNN leverages the attention mechanism to effectively emphasise different regions in the image of railway tracks, thus improving the detection of missing bolts by directing computational resources to the most critical areas. This targeted approach not only boosts the model's accuracy but also reduces the need for excessive computing power, typically associated with deeper, more complex networks. The synergy of a lightweight neural network with an attention mechanism is a pioneering approach in railway maintenance. It promises to significantly elevate the current state of the art in automated bolt detection, delivering a practical, efficient, and highly accurate system that fulfils the rigorous demands of railway safety inspections.

In essence, the primary objective of this paper is to present an efficient, accurate, and reliable method for detecting missing bolts in railway infrastructure, embodied in the deployment of our novel lightweight CNN with an integrated attention mechanism. Aiming to overcome the limitations of manual inspections and traditional automated methods, this solution is both computationally light and robust against the real-world complexities of railway maintenance. The scope of this research includes the development and evaluation of the CNN architecture, tailored to operate with high efficiency and deployable in various settings, notably those with limited computational resources. This paper will detail an extensive comparative analysis of our model against the current state of the art, demonstrating its superior performance in bolt detection tasks. Alongside a comprehensive methodology for model training and validation, the paper will highlight potential applications and discuss the wider implications of this technology for the field of railway safety

and automated visual inspections. Our work strives to set a new benchmark in railway maintenance technology and to offer a model adaptable to various railway systems worldwide. Moreover, this research aims to contribute to the broader field of machine learning and computer vision, introducing an architecture that reconciles high accuracy with operational efficiency—two often-competing priorities in deep learning model development.

## 2. Related Work

The advancement of automated inspection systems in the railway industry marks a significant stride towards enhancing operational safety and efficiency. At the heart of these developments lies the integration of computer vision and deep learning technologies, which have shown remarkable capabilities in detecting and diagnosing infrastructural anomalies. Particularly, the detection of missing bolts is a critical component in ensuring the structural integrity of railway tracks. This presents unique challenges that necessitate innovative solutions.

### 2.1. Traditional Bolt Detection Methods

The maintenance of railway components, including bolt detection, has historically been a manual task requiring significant human effort and time. There are inherent risks of oversight and inaccuracies. Automated methods using traditional computer vision techniques have been developed to improve efficiency and accuracy. These methods, however, often need help with the complexities of real-world railway environments, such as varying lighting conditions and occlusions.

Traditionally, railway inspection, including bolt detection, was conducted manually by trained workers who would walk along the railway lines to identify any potential risks. While thorough, this manual process has been criticised for being slow, costly, dangerous, and subject to human error [4]. Automated vision-based systems have been developed to improve the manual process. Early systems utilised techniques like wavelet transforms and principal component analysis for image preprocessing to detect the absence of fastening bolts with a high degree of reliability and robustness [29].

Marino et al., in their research, proposed real-time visual inspection systems for railway maintenance, like the Visual Inspection System for Railway (VISyR), which employs digital line-scan cameras and neural classifiers to detect fastening bolts with high accuracy [30]. Similarly, L. Liu et al. proposed machine vision approaches to automatically inspect the status of fastening bolts on freight trains, achieving high inspection accuracy and real-time performance [31]. Recent trends in bolt detection are shifting towards more advanced techniques that include the use of deep learning and convolutional neural networks (CNNs) for improved accuracy and efficiency in detecting various components of rail tracks, including bolts [32]. Sun et al. proposed innovative methods using binocular vision, which have also been proposed for detecting bolt-loosening on key components of running trains, thereby significantly improving fault detection efficiency [33].

The development of automated bolt detection systems marks a significant step towards enhancing the safety and reliability of railway infrastructure, making the process more efficient and less dependent on manual labour while reducing the risk of human error. Traditional methods, while being thorough, are limited by their time-consuming nature and high labour costs. Automated systems using traditional computer vision techniques, although more efficient, still struggle with complex real-world conditions such as varying lighting and occlusions. These methods include the use of digital line-scan cameras and simple neural classifiers, which, while improving over manual inspections, do not consistently handle the complexity of diverse environmental conditions.

### 2.2. Convolutional Neural Networks in Bolt Detection

The field of automated bolt detection has been revolutionised by the introduction of convolutional neural networks (CNNs), which have significantly improved the accuracy and speed of detecting and classifying bolts in railway maintenance.

A particularly noteworthy development is the use of CNNs in a hierarchical detection framework proposed by L. Liu et al., which includes extracting the fault area containing the target from a complex background and then employing gradient-orientation-based features alongside a support vector machine classifier for bolt detection. This approach has led to impressive accuracy levels in automated status inspections of fastening bolts on freight trains [31]. In another instance, Gibert et al. proposed a deep multitask learning framework for railway track inspection. Combining multiple detectors improves accuracy in detecting defects in railway ties and fasteners. It shows that when networks are trained to recognise multiple, distinct patterns simultaneously, performance can be significantly enhanced, suggesting a beneficial path for future CNN-based bolt detection methodologies [34].

The use of deep learning extends to transforming the problem of abnormal bolt detection into a bolt number detection issue. Wang et al. proposed that Faster R-CNN and YOLO have been adapted for this purpose, and they developed a new network based on ResNet to count the number of bolts, which serves as an indicator of bolt presence and condition [35]. Moreover, CNN models have been successfully implemented in systems that inspect train wheels, showcasing the potential of deep learning in automating the bolt inspection process. Li et al., in their research, trained a model to distinguish between bolt and non-bolt images, with the ability to adapt to various situations encountered by bolts in real-life scenarios [36].

Additionally, another method that has significantly improved the performance of deep learning models is attention mechanisms. It is inspired by human visual attention and focuses on the most informative parts of input by dynamically weighing the significance of different features. This capability can significantly enhance CNN performance, especially in tasks requiring the differentiation of fine-grained details or when dealing with noisy and complex datasets. Wang et al. proposed an innovation called AttnConvnet, which integrates an attention mechanism within a deep CNN to detect multiple rail components, including bolts. The use of positional embedding and cascading attention blocks allows for learning the local context of rail components, simplifying the detection pipeline by removing the need for pre- and post-processing [37]. Similarly, Alif et al. proposed Boltvision, which demonstrated effective uses of transformer-based architecture. By performing a comparative analysis of CNNs, ViTs, and CCTs, the study contributes to the field by emphasising the practical implications of deploying such models on edge devices where computational resources are limited. The utilisation of a pre-trained ViT base within BoltVision and achieving 93% accuracy in classifying missing bolts is particularly notable [38].

Overall, attention mechanisms have vast potential to enhance CNN performance, offering a pathway to more intelligent and efficient models capable of high-fidelity detection tasks in railway maintenance and beyond. The further exploration and integration of attention into CNN architectures are likely to yield significant improvements in automated visual inspection systems.

### *2.3. Limitations of Existing Deep Learning Approaches*

While deep learning and CNNs have advanced bolt detection in railway maintenance, they are not without limitations. A significant challenge lies in the balance between computational efficiency and accuracy. Deep learning models, particularly those with complex architectures, require substantial computational resources, which can be a bottleneck when deployed in real-time systems or on edge devices with limited processing capabilities.

In the context of real-world applications, intricate models may struggle to maintain high-speed performance without sacrificing accuracy. This is evident in systems that demand on-the-fly analysis, such as those used for inspecting high-speed railways where detection must occur in milliseconds. According to the research performed by Gibert et al., although the deep multitask learning framework improves accuracy, it highlights the trade-off between computational demand and real-time processing efficiency [34]. Another concern is the robustness of these models in varied environmental conditions.

The performance of CNNs can degrade when faced with images captured in poor lighting, from different angles, or with occlusions. This was observed in the hierarchical detection framework, where complex backgrounds and varying lighting conditions posed significant challenges [31]. Furthermore, the requirement for large labelled datasets for training can be a limitation. Deep learning models are data-hungry, and the need for labelled examples, especially of rare defects, can hinder the model's ability to learn effectively. This has been noted as an issue by Wang et al. in systems designed for fault detection, where the models may not generalise well to unseen defects or new types of bolts [35].

The challenges associated with deep learning for bolt detection extend into areas such as the overfitting of models to training data, which can reduce their effectiveness in practical scenarios. Overfitting occurs when a model learns details and noise in the training data to the extent that it negatively impacts the performance of the model on new data, leading to less accuracy when deployed in real-world conditions. This issue becomes more pronounced when the variation in the operational environment is significant and not fully represented in the training set [39]. Additionally, the interpretability of deep learning models remains a limitation. While these models can perform with high accuracy, understanding the decision-making process is often difficult. This "black box" nature of deep learning models can be a barrier, particularly in safety-critical applications like railway maintenance, where explainability is important for trust and diagnostics [36]. Transfer learning has been proposed as a solution to address the issue of limited labelled data, but it is not without drawbacks. While it allows a pre-trained model to be applied to new but related tasks, there is still a requirement for fine-tuning the model with a sufficient amount of target data to achieve the desired level of performance. Moreover, if the new task diverges significantly from the source task, the effectiveness of transfer learning can be compromised, resulting in suboptimal performance [35].

To sum up, while deep learning offers promising directions for automated bolt detection, computational efficiency model robustness and generalisation capabilities need to be improved. Addressing these limitations is crucial for developing more sophisticated and practical inspection systems that can operate under a broad spectrum of conditions and constraints.

#### *2.4. Comparison of Bolt Detection Technologies*

Bolt detection has long been a critical concern in railway maintenance. The safety and reliability of railway infrastructure are heavily dependent on the accurate and timely identification of missing or loose bolts. The methods employed for this task have evolved significantly over the years, reflecting advancements in technology and an increased understanding of the challenges inherent in railway environments.

Initially, bolt detection was predominantly a manual process conducted by human inspectors. This method, while thorough, is fraught with drawbacks—primarily, it is labour-intensive, time-consuming, and prone to human error. Inspectors are subject to fatigue, which can compromise the accuracy of inspections, particularly under adverse conditions. As technology advanced, automated systems employing traditional computer vision techniques were developed. These systems, utilizing tools like wavelet transforms and principal component analysis, marked a significant improvement over manual inspections by increasing efficiency and reducing human error. However, their effectiveness was often limited by their inability to handle the complexities of real-world conditions such as variable lighting, weather effects, and physical obstructions [29].

Further developments led to the adoption of more sophisticated machine vision systems. Technologies like the Visual Inspection System for Railway (VISyR), which utilizes digital line-scan cameras and basic neural classifiers, improved accuracy [30]. Nevertheless, these systems sometimes struggle to perform consistently across the diverse and dynamically changing environments found along railway tracks.

The introduction of deep learning, particularly the use of convolutional neural networks (CNNs), has revolutionized bolt detection. CNNs significantly enhance both the

accuracy and speed of detection by directly learning from large quantities of data to identify complex patterns that characterize bolt presence or absence. When enhanced with attention mechanisms, these models further refine their predictive capabilities. The attention layers allow the models to focus selectively on the most relevant features of the input data, mirroring human visual attention and effectively improving the detection outcomes in noisy or complex scenes [40].

For a detailed comparison of these methodologies, see Table 1, which summarizes the technologies used, advantages, and limitations of each bolt-detection method. This table helps illustrate how each approach has contributed to advancements in the field, setting the context for our proposed CNN model's development.

**Table 1.** Comparison of bolt detection technologies.

| Method                             | Technology  | Advantages   | Limitations  |
|------------------------------------|---|--|--|
| Manual Inspection [4]              | Human visual inspection                               | Highly thorough  | Time-consuming, costly, subject to human error                               |
| Traditional Automated Systems [29] | Digital line-scan cameras, wavelet transforms, PCA    | Improved speed and reliability                                 | Struggles with complex environments, limited by basic CV techniques          |
| Early CNN Approaches [29,30,32,33] | Basic CNN models                                      | Better accuracy and efficiency in ideal conditions             | Require substantial computational resources struggles in varied environments |
| Advanced CNN Techniques [31,34,40] | Hierarchical detection frameworks, multitask learning | High accuracy, capable of complex pattern recognition          | Still demands high computational power, can be costly to implement           |
| Innovative Approaches [37,38]      | Attention mechanisms, Transformer-based architectures | Enhanced focus on relevant features, suitable for edge devices | May require extensive training data, complex to implement                    |

Deep learning models, especially those equipped with attention mechanisms, surpass the performance of traditional and machine vision systems not only in terms of accuracy and reliability but also by demonstrating superior adaptability to new and unseen conditions—critical for real-time applications in railway maintenance, where anomalies must be detected swiftly to prevent failures.

The promising results obtained from deep learning models suggest a pathway toward more intelligent, efficient, and robust bolt detection systems. Future research can further explore the integration of these advanced models into comprehensive railway inspection systems that operate autonomously under a wide range of conditions. Moreover, the ongoing development of these technologies could potentially extend to other critical infrastructure inspection tasks, where the ability to quickly and accurately detect anomalies is paramount.

### 3. Methodology: Custom Convolutional Development Pipeline

#### 3.1. Dataset

The study's dataset was collected using a sophisticated setup featuring a robotic arm equipped with a high-resolution camera, specifically chosen to mimic the diverse angles and positions from which images are typically captured within an operational railway setting (see Figure 2). This setup includes a Raspberry Pi camera module equipped with an infrared sensor. The use of the Raspberry Pi camera with an infrared sensor ensures that images are captured consistently and accurately, even in challenging lighting conditions commonly encountered in railway environments. The parameters of the high-resolution camera are given in Table 2.



**Figure 2.** Robotic arm equipped with a high-resolution camera for automated image acquisition.

**Table 2.** Parameters of the high-resolution camera.

| Parameter           | Value          |
|---------------------|----------------|
| Resolution          | 12 megapixels  |
| Lens focal length   | 3.6 mm         |
| Aperture            | f/2.0          |
| Sensor type         | CMOS           |
| Field of view       | 60 degrees     |
| Image sensor size   | 1/2.3 inch     |
| ISO sensitivity     | ISO 100-3200   |
| Shutter speed range | 1/1000 to 30 s |

Such a setup is crucial in bypassing the inconsistencies associated with handheld devices like smartphones and ensures standardised, reproducible image acquisition methods, making it well-suited for integration with edge computing devices.

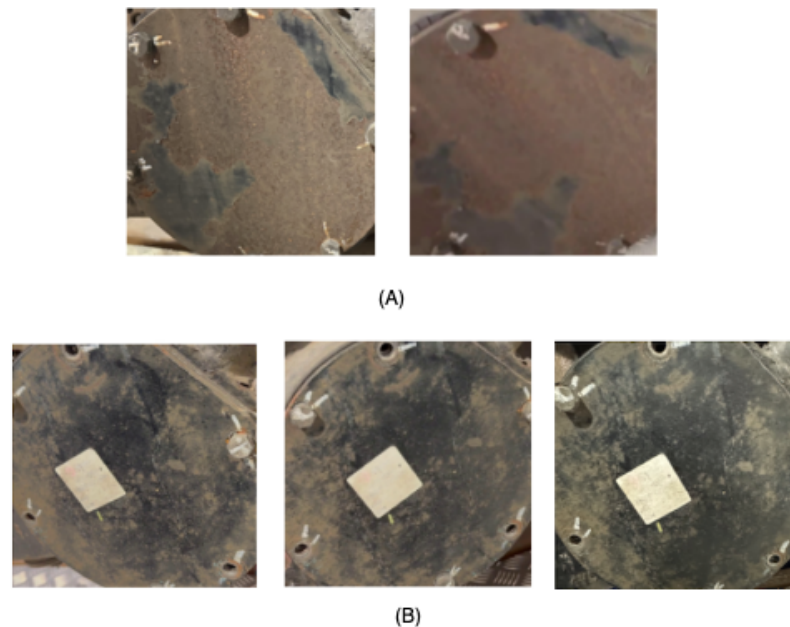
The dataset comprises 145 images, each meticulously resized to a uniform resolution of  $224 \times 224$  pixels. The images are categorized into two distinct groups: bolt-present and bolt-missing scenarios. The bolt-present category includes instances where the bolts are clearly visible and intact (as shown in the top row A of Figure 3). Conversely, the bolt-missing category encompasses scenarios where the bolts are absent, which could signify a potential maintenance issue (as demonstrated in the bottom row B of Figure 3).

Each image in the dataset was captured with careful consideration for the operating environment and the positioning of the device to ensure a realistic simulation of on-site conditions. The images encapsulate the intricacies encountered in real-world inspection settings, including factors such as occlusions, rust, oil stains, and the presence of scale, while also accommodating the variability observed across different subsections of train components.

To facilitate focused learning for the model, images were cropped to centre around bolt-containing sections of the train components, with equal representation given to samples showcasing both intact bolts and scenarios where bolts are missing. This deliberate emphasis enables the model to learn and distinguish patterns pertinent to the presence or absence of bolts, aiding in the accurate detection of potential safety hazards.



The curation of the dataset was geared towards providing the convolutional network with a rich and diverse set of visual inputs, thereby optimizing the performance of the attention mechanism integrated within the network for reliable bolt detection. The diversity and quality of the dataset are instrumental in training a model that can accurately identify missing bolts, a critical aspect of railway maintenance and safety inspections.



**Figure 3.** Sample images from the bolt detection dataset: row (A) showcases instances with bolts properly in place, while row (B) illustrates scenarios where bolts are missing.

### 3.2. Data Augmentation

Data augmentation plays a critical role in the development of robust machine learning models, especially in the realm of computer vision. For our paper, augmentation is particularly significant due to the high variability and unpredictability of real-world railway maintenance scenarios. By artificially expanding the dataset through various transformations, data augmentation enhances the diversity of training examples. This process helps to prevent overfitting, improve model generalizability, and ultimately increase the robustness of the predictive model against unseen data. Such enhancements are vital for an architecture designed to operate in the field, where it will encounter a wide range of conditions not perfectly mirrored in the initial dataset.

In the context of our lightweight convolutional network, data augmentation ensures that the model is not only trained on a wide spectrum of possible scenarios, including different angles, lighting conditions, and occlusions, but also learns to identify bolts regardless of such variations. The attention mechanism within the model benefits from augmented data by learning to focus effectively on relevant features amidst diverse and challenging inputs.

For this project, we utilized PyTorch, a leading deep-learning library, to implement our data augmentation strategy effectively. PyTorch offers a comprehensive suite of data augmentation tools that provide a flexible platform for easily applying a wide array of augmentation techniques. The original dataset consisted of 145 images, and through systematic augmentation, we expanded this significantly to simulate an extensive and varied set of visual scenarios. This expanded dataset, showcased in Figure 4, underpins the training process, ensuring that our architecture is both rigorously trained and thoroughly tested. The result is a more effective and reliable bolt detection system that is adept at operating under the variable conditions typical in railway environments. This detailed approach to data augmentation underscores our commitment to creating a highly adaptable and resilient model capable of performing reliably in real-world settings, where extensive data collection is often impractical. By emphasizing the transformative effect of our data

augmentation techniques, we reinforce the robustness and adaptability of our model to meet the dynamic demands of railway safety maintenance.

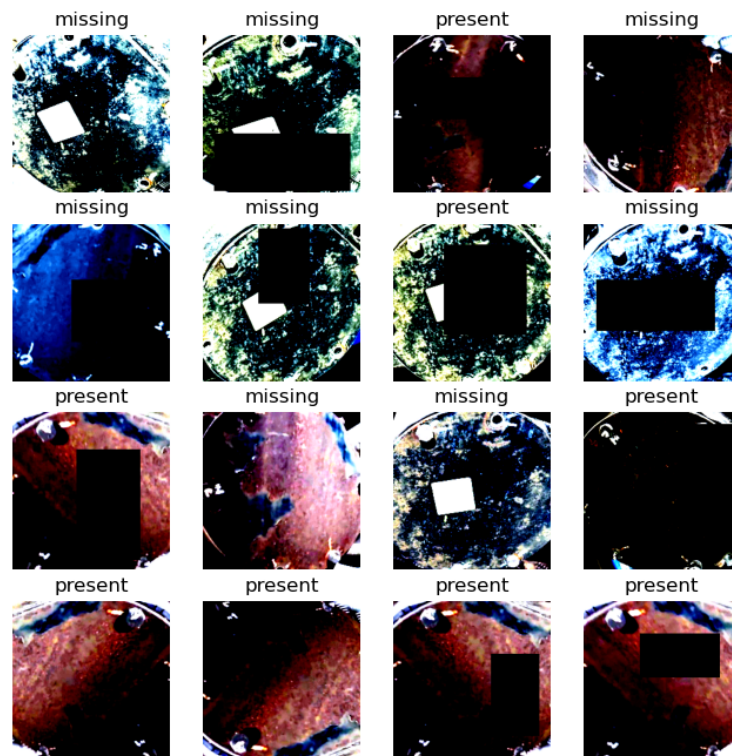


Figure 4. Examples of augmented images from the bolt detection dataset.

### 3.2.1. Random Vertical Flip

The horizontal flip augmentation reflects an image along the vertical axis, effectively creating a mirror image. This technique addresses the scenario where the edge device captures images from opposing directions. The probability  $p$  determines the chance of any image being flipped during the augmentation process, as given in Equation (1).

$$I' = \text{flip}_{\text{horizontal}}(I) \tag{1}$$

### 3.2.2. Random Horizontal Flip

Similarly, the vertical flip mirrors the image along the horizontal axis as per Equation (2). While less common in bolt inspection scenarios, vertical flips ensure the model’s robustness against unusual but possible situations, such as upside-down camera mounting.

$$I' = \text{flip}_{\text{vertical}}(I) \tag{2}$$

### 3.2.3. Random Rotation

This augmentation rotates the image by a random angle  $\theta$  within a specified range Equation (3), typically between  $-90$  and  $90$  degrees. This imitates the varying angles at which images might be captured, particularly in less controlled environments.

$$I' = \text{rotate}(I, \theta) \tag{3}$$

The rotation transformation can be mathematically described using the rotation matrix  $R(\theta)$  for a 2D image as given in Equation (4):

$$R(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \tag{4}$$

### 3.2.4. Random Auto Contrast

This augmentation is another powerful technique used to enhance the performance of our model. Auto-contrast automatically adjusts the image contrast so that the histogram of the output image is spread out, improving the visibility of features and details that are important for the model's learning process. The following Equation (5) can represent the process of applying auto-contrast:

$$I' = \frac{I - \text{Min}(I)}{\text{Max}(I) - \text{Min}(I)} \times (L_{\text{max}} - L_{\text{min}}) + L_{\text{min}} \quad (5)$$

where  $I$  is the original image,  $I'$  is the image after applying auto-contrast,  $\text{Min}(I)$  and  $\text{Max}(I)$  are the minimum and maximum pixel intensity values in the original image respectively, and  $L_{\text{max}}$  and  $L_{\text{min}}$  are the maximum and minimum pixel intensity values desired in the output image.

### 3.2.5. Normalization

Normalization is a vital data augmentation strategy, especially in preparing datasets for convolutional neural networks (CNNs). It involves adjusting the pixel intensity values so the dataset has a mean of zero and a standard deviation of one. Standardizing the dataset helps stabilize the learning process and ensures that the model trains faster and performs better by reducing internal covariate shifts. For our dataset, the normalization process can be described mathematically in Equation (6):

$$I' = \frac{I - \mu}{\sigma} \quad (6)$$

Here,  $I$  is the original image,  $I'$  is the normalized image,  $\mu$  is the mean pixel intensity computed over the entire dataset, and  $\sigma$  is the corresponding standard deviation.

### 3.2.6. Random Grayscale

This data augmentation technique converts colour images to grayscale with a certain probability. By introducing grayscale images during training, the model becomes invariant to colour, focusing instead on texture and shape, which are critical for bolt detection. The transformation process can be formalized as given in Equation (7):

$$I' = \begin{cases} 0.299 \times R + 0.587 \times G + 0.114 \times B, & \text{with probability } p \\ I, & \text{with probability } (1 - p) \end{cases} \quad (7)$$

Here,  $I$  represents the original RGB image,  $I'$  is the transformed image,  $R$ ,  $G$ , and  $B$  denote the red, green, and blue channel values of  $I$ , respectively, and  $p$  is the probability with which the transformation is applied, set at 0.1. This equation utilizes the luminance channel in the YUV colour space, which is a weighted sum of the RGB channels and is known to approximate the perception of grayscale by the human eye.

### 3.2.7. Random Erasing

This data augmentation strategy is designed to improve the robustness of the model by simulating occlusions and forcing the network to learn more representative features. It randomly selects a rectangle region in an image and erases its pixels with random values. The operation of random erasing can be defined by Equation (8).

$$I'(x, y) = \begin{cases} \text{random\_value}, & \text{if } (x, y) \in \text{erase\_region} \\ I(x, y), & \text{otherwise} \end{cases} \quad (8)$$

where  $I$  is the original image,  $I'$  is the image after applying random erasing,  $(x, y)$  are the coordinates of a pixel within the image, and  $\text{erase\_region}$  is the randomly selected region for

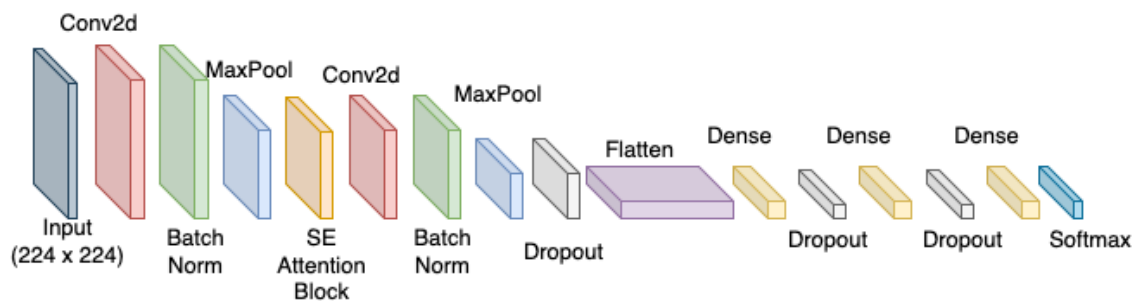
erasure. The size, aspect ratio, and location of the erase\_region are randomly determined, and random\_value is typically a pixel value drawn from a uniform distribution.

Notably, we also employed a resizing strategy where input images are first resized to  $256 \times 256$  pixels before being center-cropped to  $224 \times 224$  pixels. This particular approach ensures that the network encounters variations in the scale and framing of the subject matter, mirroring the changes a model would experience in the field. Specifically, the resizing and cropping techniques are implemented to accommodate for positional variances and focus the model's attention on the central aspects of the image, which are of primary interest in bolt detection tasks. By adopting these augmentation methods, we introduce an element of spatial variance to the model, which aids in teaching the network to recognize and localize relevant features across a range of perspectives and scales.

This comprehensive approach to data augmentation is fundamental to our convolutional network's training process. It allows the model to become invariant to the scale and position of bolts within the images. As a result, it helps mitigate overfitting and bolster the model's ability to generalize from the training data to real-world scenarios. The combination of these augmentation techniques forms a robust dataset instrumental in refining the performance of our lightweight convolutional network tailored for missing bolt detection in railway maintenance.

### 3.3. Proposed CNN Architecture

The proposed architecture presents a novel and efficient model tailored for high-accuracy bolt detection. It is designed to be lightweight for deployment on edge devices. It incorporates a series of convolutional layers, batch normalization, ReLU activations, pooling layers, dropout layers for regularization, and a crucial Squeeze-and-Excitation (SE) block that implements the attention mechanism. An example of the architecture can be seen in Figure 5.



**Figure 5.** Schematic of the CNN architecture.

The model's structure is meticulously organized into two primary sections: the body and the head. The body comprises a sequence of layers responsible for feature extraction, while the head focuses on classification.

#### 3.3.1. Feature Extraction

**Convolutional Layer:** The initial part of the model starts with a Convolutional (Conv2d) layer, transforming the input image size from  $224 \times 224 \times 3$  to  $222 \times 222 \times 5$ , with a slight parameter count of 140, emphasizing the model's efficiency. Conv2d layers are designed to automatically and adaptively learn spatial hierarchies of features from input images. In the context of bolt detection, the Conv2d layer extracts essential features such as edges, textures, and patterns that are indicative of the presence or absence of bolts. By using filters (also known as kernels), Conv2d layers apply a convolution operation to the input that captures the local dependencies in the image. Conv2d layers are also efficient in terms of the number of parameters. Unlike a fully connected layer that connects every input to every output (which would be computationally intensive and prone to overfitting), Conv2d layers share weights across the spatial dimensions. This weight sharing signifi-

cantly reduces the number of parameters, making the network less complex and faster to train without compromising its ability to learn.

In our architecture, the Conv2d layer is a fundamental building block that captures the visual essence of bolt presence, facilitating the task of identifying missing bolts with high accuracy and ensuring the reliability of the inspection process. This layer's ability to extract and learn from the visual data makes it indispensable for the success of our lightweight convolutional network.

**Batch Normalization:** Following Conv2d, batch normalization and ReLU activation encourage model stability and non-linearity. The Batch Normalization (BatchNorm) layer is an important component in deep neural networks, particularly in complex architectures like our CNN for detecting missing bolts. BatchNorm is used to normalize the inputs of each layer so that they have a mean of zero and a standard deviation of one. By doing so, BatchNorm addresses the issue of internal covariate shift, where the distribution of each layer's inputs changes as the parameters of the previous layers change during training. Normalizing the inputs helps to mitigate the problem of gradients becoming too small (vanishing) or too large (exploding), which can cause the training to stall or diverge. The operation performed by a BatchNorm layer for a given input  $x$  can be described as Equations (9)–(12):

$$\mu_B = \frac{1}{m} \sum_{i=1}^m x_i \quad (9)$$

$$\sigma_B^2 = \frac{1}{m} \sum_{i=1}^m (x_i - \mu_B)^2 \quad (10)$$

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \quad (11)$$

$$y_i = \gamma \hat{x}_i + \beta \quad (12)$$

Here,  $m$  is the batch size,  $x_i$  is the input to be normalized,  $\mu_B$  is the mean of the batch,  $\sigma_B^2$  is the variance,  $\epsilon$  is a small constant added for numerical stability, and  $y_i$  is the output of the BatchNorm layer. In the context of our proposed CNN, the use of BatchNorm after convolutional layers is critical for maintaining stable training, improving model performance, and ensuring that our lightweight network can be trained efficiently and effectively to detect missing bolts in railway components.

**MaxPool Layer:** A subsequent MaxPool2d layer reduces the spatial dimension to  $111 \times 111 \times 5$ . Max pooling is a critical operation within convolutional neural networks (CNNs), including our proposed architecture for missing bolt detection in railways. By reducing the spatial dimensions of the input feature maps, max pooling serves several essential purposes that significantly contribute to the effectiveness and efficiency of the model. While the operation reduces the resolution of the feature maps, it preserves essential contextual information by retaining the most significant signals from each region. This balance between detail reduction and context preservation is crucial in a task like bolt detection, where the goal is to discern between relatively small and potentially subtle differences in visual patterns that indicate the presence or absence of bolts.

**SE Block:** Further distilled by an SE block, which plays a pivotal role in enhancing relevant features through the adaptive re-calibration of channel-wise feature responses. The Squeeze-and-Excitation (SE) Block represents a form of attention mechanism that has been integrated into our convolutional neural network to enhance the model's capacity for feature recalibration. This block allows the network to perform dynamic channel-wise feature reweighting, significantly increasing its representational power.

The SE Block works in two main steps: squeeze and excitation. The squeeze operation aggregates the spatial information of each channel into a single descriptor, typically by using global average pooling. This results in a compact feature descriptor that summarizes

the global distribution of the feature responses for each channel, which can be expressed with Equation (13):

$$z_c = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W u_c(i, j) \quad (13)$$

where  $u_c$  is the input feature map for channel  $c$  and  $H$  and  $W$  are the height and width of the feature map, respectively.

Following the squeeze operation is the excitation step, which is a fully connected feed-forward network that learns a channel-specific descriptor. The descriptor serves as a gating mechanism, capturing channel-wise dependencies and allowing the network to learn which features to emphasize or suppress. The excitation function, typically involving a sigmoid activation, can be formalized in Equation (14):

$$s_c = \sigma(\mathbf{W}_1(\delta(\mathbf{W}_2 z_c))) \quad (14)$$

where  $\mathbf{W}_1$  and  $\mathbf{W}_2$  represent the weights of the two fully connected layers within the SE block,  $\delta$  is the ReLU activation function,  $\sigma$  is the sigmoid activation function, and  $s_c$  is the recalibration coefficient for channel  $c$ . In the context of bolt detection, the SE Block is crucial for a few reasons:

- **Focus on Informative Features:** By explicitly modelling interdependencies between channels, the SE Block allows the network to focus on the most informative features for the task of bolt detection, which can include specific shapes, edges, or textures indicative of a bolt or its absence.
- **Enhanced Representation:** The recalibration of features afforded by the SE Block helps the network to adaptively enhance representations that are important for distinguishing between bolts and no bolts and to suppress less useful ones. This adaptability is particularly valuable given the high variability in visual appearance due to lighting, orientation, and occlusion in railway environments.
- **Improved Model Generalization:** The attention mechanism provided by the SE Block enables the model to generalize better from the training data to unseen data. It effectively allows the network to make more nuanced decisions based on the relative importance of different features in the context of the specific task.
- **Robustness to Noise and Variability:** In practical railway maintenance scenarios, images captured for bolt detection can have noise and variability, such as rust, grease, or shadows. The SE Block helps the model maintain robustness against such noise by emphasizing relevant features that are indicative of the target classes.

By integrating the SE Block into our CNN architecture, the network becomes more discriminative and efficient. It can learn complex feature interdependencies without a significant increase in computational burden, which aligns with the need for deploying such models on edge devices with limited computational resources. This capability positions our architecture as particularly suitable for the real-time, accurate detection of missing bolts, a critical safety concern in railway maintenance operations.

Following the SE block, another convolution layer, along with batch normalization and ReLU activation, continues the feature extraction, succeeded by a MaxPool2d and a Dropout layer. The dropout rate of 0.1 helps prevent overfitting by randomly omitting a subset of features during training.

### 3.3.2. Classification

Transitioning to the classification segment, the model reshapes the extracted features into a vector of size 32076, which then passes through a dense layer, reducing it to 100 dimensions. This step is crucial for condensing the information into a more manageable form for classification. The inclusion of ReLU activation and dropout layers maintains the consistency of the regularization strategy, further refined by subsequent linear layers

progressively narrowing down the output to a two-dimensional vector representing the presence or absence of bolts.

### 3.3.3. Parameters and Computational Efficiency

Our proposed model represents a significant stride in computational efficiency, with only 3,213,530 trainable parameters. When we compare this to several state-of-the-art (SOTA) architectures, the contrast becomes even more striking, as shown in Table 3. As the table illustrates, traditional architectures like VGG and AlexNet have parameter counts an order of magnitude larger than our model. Even more efficient designs like GoogleNet and ResNet-18 are more than three times larger. The Vision Transformer (ViT), an advanced architecture leveraging attention mechanisms across the entire network, typically requires upwards of 86 million parameters. This reflects its expansive capacity but also highlights its computational demands, which may not be as well-suited for edge computing scenarios. The efficient design of our model is not limited to just the number of parameters but also extends to computational operations, with total multi-adds reaching a mere 16.12 megabytes. This optimization is crucial for deployment on edge devices, where efficiency in both memory usage and processing is paramount.

**Table 3.** Comparison of trainable parameters in our model versus other state-of-the-art architectures.

| Model                 | Parameters (Millions) |
|-----------------------|-----------------------|
| VGG-16                | 134.70                |
| VGG-19                | 143.67                |
| AlexNet               | 61.00                 |
| Inception v1          | 13.00                 |
| ResNet-18             | 11.69                 |
| ResNet-34             | 21.50                 |
| ViT-B-16              | 87                    |
| ViT-B-32              | 88                    |
| Proposed Architecture | 3.21                  |

In summary, the proposed architecture is a finely tuned ensemble of convolutional layers, attention mechanisms, and classification layers, all orchestrated to deliver exceptional performance in detecting missing bolts in railway components. The model's thoughtful design, combining depth and efficiency, demonstrates its potential for real-world application in railway maintenance and safety inspections. Moreover, the architecture is not only competitive in terms of learning capability but also stands out for its tailored efficiency. It is designed to operate within the constraints of edge computing, delivering high-performance bolt detection with a fraction of the parameters and computational cost typically associated with SOTA models. This deliberate balance positions our model as a prime candidate for real-world application, particularly in resource-constrained environments.

### 3.4. Training Process

The training process of our convolutional neural network, designed for detecting missing bolts, employs a robust strategy that encompasses meticulous data preparation, a well-defined set of hyperparameters, and the utilization of specific loss functions and optimizers to refine the model's performance.

#### 3.4.1. Data Preparation

The dataset comprises 101 images for training, 28 for validation, and 16 for testing, clearly divided into two classes: "missing" and "present". This categorization ensures that the model is exposed to a balanced view of both scenarios it needs to recognize.

### 3.4.2. Hyperparameters and Optimization

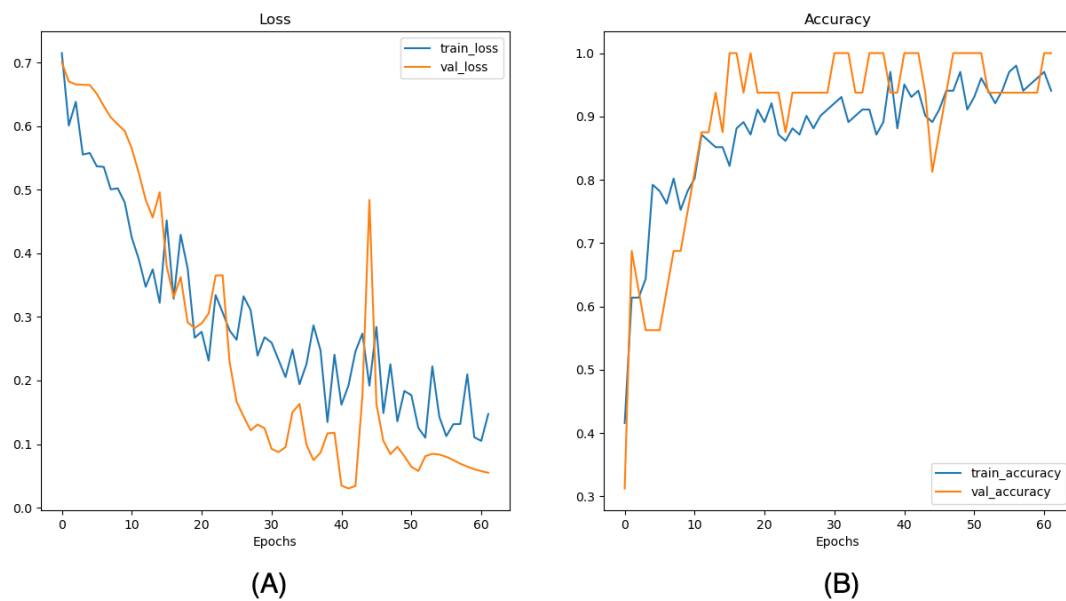
As given in Table 4, a comprehensive set of hyperparameters underpins the training process: the model is trained for 1000 epochs with images resized to  $224 \times 224$  pixels and batched into sizes of 256. The Stochastic Gradient Descent (SGD) optimizer, chosen for its effectiveness in navigating the complex landscapes of high-dimensional data, is applied with a learning rate (lr) of 0.02. The CrossEntropyLoss function is selected for its proficiency in handling binary classification tasks, ensuring that the model accurately discriminates between the two classes. To enhance the training dynamics, a ReduceLROnPlateau scheduler adjusts the learning rate based on the validation loss, introducing a patience mechanism to counteract plateaus in model improvement. Furthermore, early stopping is employed with a patience of 20 epochs and a minimum delta of 0, ensuring that training ceases when the validation loss fails to improve, thereby preventing overfitting.

**Table 4.** Detailed summary of the hyperparameters utilized in the training process.

| Hyperparameter               | Value              |
|------------------------------|--------------------|
| Epochs                       | 1000               |
| Image Size                   | $3224 \times 224$  |
| Batch Size                   | 256                |
| Optimizer                    | SGD                |
| Learning Rate (lr)           | 0.02               |
| Loss Function                | Cross Entropy Loss |
| Scheduler                    | ReduceLROnPlateau  |
| Early Stopping Patience      | 20                 |
| Early Stopping Minimum Delta | 0                  |

### 3.4.3. Training Dynamics

Figure 6 illustrates the training dynamics of the convolutional neural network model over the course of 60 epochs. The graphs show a clear trend in both the loss and accuracy metrics for the training (train loss and train accuracy) and validation (val loss and val accuracy) datasets.



**Figure 6.** Training dynamics of the CNN model, displaying trends in loss (A) and accuracy (B) for both training and validation datasets.

In the loss graph, we can observe a consistent decrease in both training and validation loss over time, indicating that the model is learning effectively from the data. Notably,



the training loss shows a smoother decline, while the validation loss exhibits some volatility, suggesting that the model may be encountering a variety of challenges present in the validation data that are not as prevalent in the training set.

The accuracy graph reveals a steady increase in both training and validation accuracy, with training accuracy climbing more smoothly compared to validation accuracy, which displays some fluctuations. However, both accuracy measures plateau towards the later epochs, which could signify that the model is approaching its maximum learning capacity given the current data and hyperparameter settings.

Overall, the depicted training process suggests a successful learning phase, with the model showing promising generalization capabilities. The accuracy levels out at a high value, which is indicative of the model's strong performance on both the training and validation datasets. The fluctuations and peaks in validation metrics also emphasize the model's resilience and adaptability to new data, which are essential for robust bolt detection in varying real-world conditions.

Overall, the training process, characterized by strategic data handling, the careful selection of hyperparameters, and a methodical approach to optimization, culminates in a model that is efficient and effective in bolt detection and ready to be deployed in scenarios demanding high accuracy and computational efficiency.

## 4. Experimental Setup and Results

### 4.1. Experimental Setup

The experimental setup for evaluating the proposed convolutional neural network was executed on a Razer Blade 14 laptop, which was selected for its combination of high-end specifications and portability. The system was powered by an AMD Ryzen 9 5900HX processor with a base clock speed of 3.30 GHz and eight cores, providing the computational muscle required for deep learning tasks. Complementing this processing power, the machine was equipped with 16 GB of DDR4-3200 MHz onboard memory, facilitating efficient data processing and model training operations. Graphics processing was handled by an Nvidia GeForce RTX 3070 GPU, which boasts 8 GB of VRAM and a 100 W Total Graphics Power rating, ensuring ample capacity for training complex neural network models. The system's storage needs were addressed with a 1 TB SSD, leveraging the M.2 NVMe PCIe 3.0  $\times$ 4 interface for rapid data access, which is crucial when dealing with large datasets and iterative model training sessions.

On the software front, the experiments were underpinned by PyTorch 2, utilizing CUDA 11.8 to harness the laptop's GPU capabilities for accelerated model training. Data visualization and analysis were conducted using Matplotlib 3.8.3, allowing for a detailed examination of the model's learning trends and performance metrics. To enhance the dataset and introduce necessary variability, the Imgaug library, a versatile augmentation library in Python, was employed. The entire setup ran on Ubuntu 22.04, a stable and widely supported Linux distribution favoured for machine learning and deep learning applications.

This configuration offered a harmonious blend of power and flexibility, enabling rigorous testing and the validation of the convolutional network model's ability to detect missing bolts in railway maintenance scenarios. The hardware and software synergy provided a robust platform for developing and assessing the model, ensuring the reliability of the experimental results and the reproducibility of the research.

### 4.2. Comparative Analysis

In the comparative analysis of our experimental results, it is essential to note that our proposed convolutional neural network not only achieved a high accuracy of 96.43% but also did so with a model architecture designed to be lightweight. This is in contrast to other state-of-the-art models that, while potent, often come with a considerably higher number of parameters. As detailed earlier, our model's parameter count is significantly lower than those of many benchmark models, making it particularly well-suited for deployment in edge computing environments where resources are constrained.

As shown in Table 5, when compared to the likes of ResNet-18, AlexNet, Inception v3, MobileNet, and the pretrained Vision Transformer (ViT) model BoltVision, our proposed model stands out for maintaining competitive performance with a fraction of the complexity. For example, the accuracy achieved by Inception v3 was marginally higher at 97%, yet this comes at the cost of increased model complexity. Similarly, MobileNet, which is optimized for mobile and edge devices, achieved an accuracy of 92% with a design that prioritizes efficiency. BoltVision’s accuracy of 93% demonstrates the potential of transformer-based architectures but still lags slightly behind our model in terms of performance.

**Table 5.** Comparative analysis of model performance with emphasis on lightweight architecture.

| Model                        | Accuracy | Parameters (Millions) |
|------------------------------|----------|-----------------------|
| ResNet-18                    | 94.00%   | 11.69                 |
| AlexNet                      | 74.00%   | 61.00                 |
| Inception v3                 | 97.00%   | 13.00                 |
| MobileNet                    | 92.00%   | 4.20                  |
| BoltVision (pre-trained ViT) | 93.00%   | 87.00                 |
| Our Proposed Model           | 96.43%   | 3.21                  |

The strategic design of our model is a testament to the effectiveness of a carefully tuned convolutional network that leverages advanced features like the Squeeze-and-Excitation (SE) block for channel-wise attention while remaining lean in its parameter usage. This approach underscores our aim to provide a model that not only excels in accuracy but also in operational efficiency, making it an optimal solution for real-time applications in settings where computational power and storage are limited.

This balance of performance and efficiency places our model as an attractive solution within the realm of railway maintenance and safety inspections, potentially revolutionizing the domain with a practical and deployable AI-driven system. The performance of our lightweight model in comparison to these larger and more complex models showcases the advancements in creating more efficient neural networks without sacrificing accuracy, positioning our work as a significant contribution to the field.

#### 4.3. Evaluation Metrics

Our model’s performance was evaluated using a suite of metrics that offer a comprehensive view of its classification abilities. These metrics included accuracy, precision, recall, and the F1-score, each providing unique insights into the model’s effectiveness from different angles.

##### 4.3.1. Accuracy

This is the most intuitive performance measure, and it is simply a ratio of correctly predicted observations to the total observations. It gives a general sense of how often the model is correct. Equation (15) formulates the accuracy calculation.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (15)$$

where TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives, respectively.

##### 4.3.2. Precision

Also known as positive predictive value, precision is the ratio of correctly predicted positive observations to the total predicted positives. High precision relates to a low false positive rate. The precision for a class is defined by Equation (16):

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (16)$$

### 4.3.3. Recall

This metric is also known as sensitivity or the true positive rate, and it measures the proportion of actual positives that were identified correctly. The equation for the recall is given in Equation (17):

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (17)$$

### 4.3.4. F1 Score

This is the weighted average of precision and recall. It takes both false positives and false negatives into account. It is a measure of a test's accuracy and is defined in Equation (18):

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (18)$$

According to the classification report given in Table 6, the model achieved perfect precision for the "missing" class and high precision for the "present" class, indicating a strong ability to identify positive cases with minimal false positives correctly. The recall scores show the model's strength in capturing all the relevant cases, particularly for the "present" class, where it reached a perfect score. The F1-scores for both classes reflect a harmonious balance of precision and recall, with the "missing" class scoring slightly higher, which is indicative of the model's adeptness at classifying the more challenging class.

**Table 6.** Classification report showcasing precision, recall, F1-score, and support for the "missing" and "present" classes.

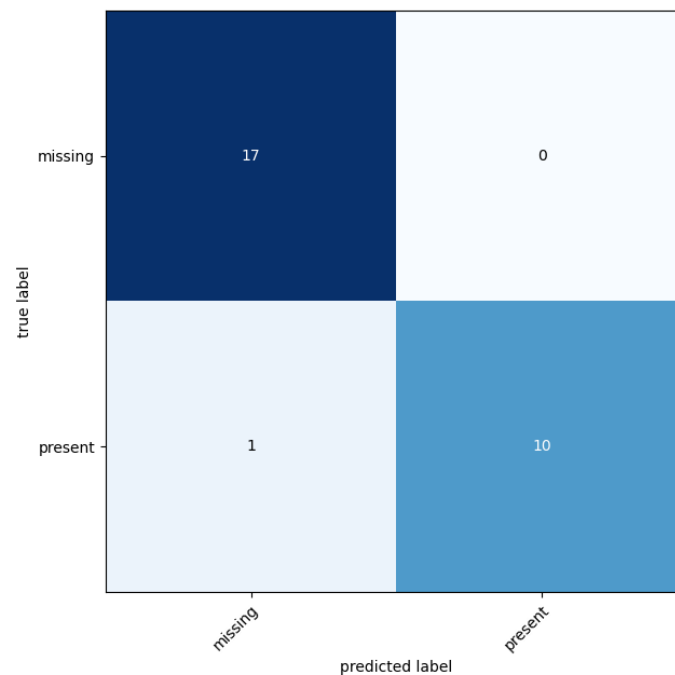
| Class            | Precision | Recall | F1-Score | Support |
|------------------|-----------|--------|----------|---------|
| Missing          | 1.00      | 0.94   | 0.97     | 18      |
| Present          | 0.91      | 1.00   | 0.95     | 10      |
| Accuracy         |           |        | 0.96     | 28      |
| Macro Average    | 0.95      | 0.97   | 0.96     | 28      |
| Weighted Average | 0.97      | 0.96   | 0.96     | 28      |

## 5. Discussion

### 5.1. Confusion Matrix Analysis

The discussion on the model's computational efficiency and deployment capabilities is crucial, particularly in the context of its application in real-time settings such as railway maintenance. The confusion matrix, depicted in Figure 7, not only provides evidence of the model's high accuracy but also underscores its efficiency. With 17 true positives and 10 true negatives out of 28 test cases, and notably only 1 false negative and 0 false positives, the model demonstrates both high reliability and precision.

The absence of any false positive predictions is particularly noteworthy, as this suggests that when the model identifies a bolt as "missing", it is highly likely to be correct. This is essential in a real-world deployment scenario where false alarms could be costly and inefficient. The solitary false negative indicates a scenario where the model predicted a bolt as "present" when it was, in fact, "missing". While this does need attention to avoid potential safety oversights, the low number is a testament to the model's effectiveness. This model's computational efficiency is highlighted by its lean architecture, which, as previously mentioned, involves significantly fewer parameters than many other state-of-the-art models. This efficiency facilitates rapid processing and reduced memory footprint, making the model suitable for deployment in edge devices, which often have limited computational resources. When considering deployment capabilities, the model's size and efficiency allow it to be integrated into a range of hardware solutions, from high-end servers for centralized processing to embedded systems aboard trains for on-the-go diagnostics. This flexibility is essential for practical applications, enabling scalable and adaptable solutions that can be tailored to specific operational requirements.



**Figure 7.** Confusion matrix for the bolt detection model.

Overall, the confusion matrix presented and the model’s architectural design converge to illustrate a system that is not only accurate but also practical for deployment. The model is a significant step towards realizing the real-time, on-site detection of missing bolts, which is key to maintaining the high safety standards required in railway operations.

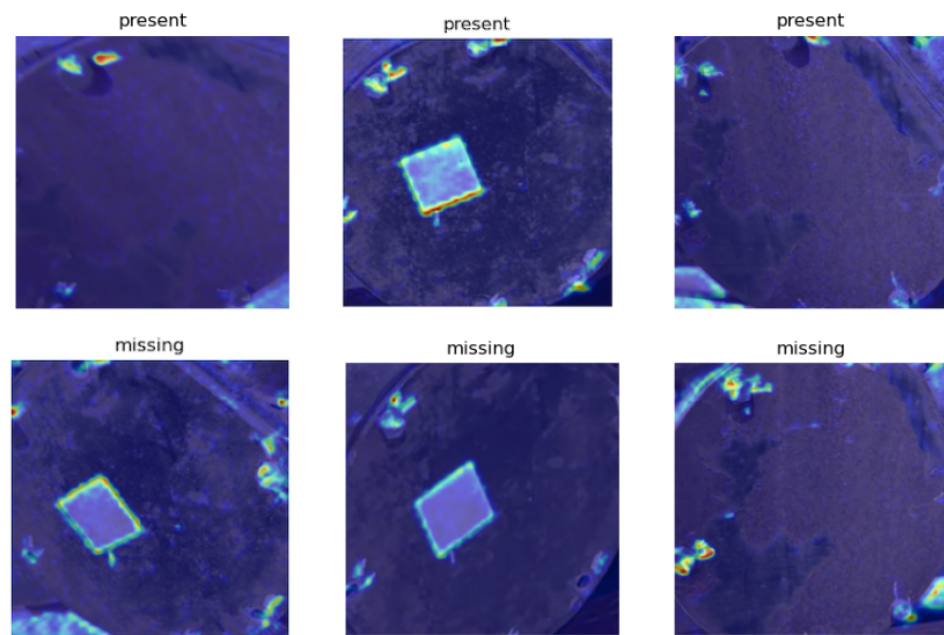
### 5.2. Interpreting Model Decisions

The Gradient-weighted Class Activation Mapping (Grad-CAM) outputs, as visualized in the overlaid heatmaps on the test images in Figure 8, provide a compelling narrative about the model’s ability to focus on relevant regions for bolt detection. These heatmaps are instrumental in understanding the model’s decision-making process, revealing the areas within the images that most significantly influence its predictions.

From a computational efficiency standpoint, the clarity of these heatmaps suggests that the model has learned to identify distinguishing features with a high degree of focus, even with a comparatively lightweight parameterization. This indicates that the network has effectively distilled knowledge into a concise, computationally efficient form. This efficiency is crucial for deployment scenarios, particularly in edge computing environments, where resources are limited and processing must be performed swiftly to keep up with real-time demands.

Furthermore, the accurate localization of the ‘missing’ and ‘present’ bolt classifications by the heatmaps underscores the model’s potential for deployment in practical applications. The ability of the model to not only provide a binary classification outcome but also visually indicate the basis of its decision enhances trust and interpretability, which are critical for maintenance teams relying on the model’s outputs to make informed decisions on railway inspections.

Grad-CAM visualizations thus offer more than just a window into the model’s internal workings; they also provide evidence of the model’s readiness for real-world application. The alignment of these insights with the model’s computational efficiency and deployment capabilities reinforces the practicality of the proposed CNN in operational settings. This combination of performance, efficiency, and interpretability positions the model as an advanced tool for enhancing the safety and reliability of railway infrastructure.



**Figure 8.** Grad-CAM visualizations representing the model’s focus areas during bolt detection.

### 5.3. Streamlined for Deployment

The model’s design as a lightweight convolutional neural network is central to its computational efficiency and deployment capabilities. Its architecture, which requires significantly fewer parameters compared to many state-of-the-art models, allows it to perform rapid inference while maintaining high accuracy. This efficiency is crucial when deploying the model in real-world scenarios, especially in edge computing environments where computational resources are often limited.

The reduced complexity of the model also means that it demands less memory for storage and less power for processing, which is beneficial for deployment on portable devices with limited battery life or on embedded systems within railway infrastructure. The lightweight nature of the model also translates to faster load times and lower latency during operation, which are key considerations for real-time applications, such as continuous monitoring and immediate fault detection in railway maintenance [41].

Furthermore, the model’s streamlined design does not compromise its ability to learn and generalize from the data, as evidenced by its performance metrics. The incorporation of techniques like Squeeze-and-Excitation blocks effectively allows the model to focus on the most relevant features without the need for additional computational heft. This selective focus enhances the model’s accuracy and interpretability, as demonstrated by the Grad-CAM visualizations, while still adhering to the constraints of a lightweight framework.

In summary, the model’s computational efficiency and deployment capabilities make it an exemplary candidate for on-device machine learning applications, as seen by the attributes in Table 7. Its ability to provide immediate, actionable insights in a resource-efficient manner makes it particularly suited for the critical task of ensuring safety in railway systems.

**Table 7.** Attributes of the lightweight model.

| Attribute                 | Benefit  |
|---------------------------|--|
| Reduced Parameter Count   | Enhances computational speed, and reduces memory usage         |
| Low Latency               | Enables real-time processing and decision-making               |
| Power Efficiency          | Ideal for battery-operated or embedded systems                 |
| Fast Inference            | Crucial for timely fault detection in critical applications    |
| Generalization Capability | Maintains high accuracy despite architectural simplicity       |
| Interpretability          | Grad-CAM visualizations aid in model trust and diagnostics     |
| Deployment Readiness      | Suited for edge computing with limited computational resources |

#### 5.4. Potential Limitations

While the proposed model demonstrates significant strengths in computational efficiency and accuracy, it is essential to recognize its potential limitations and areas that could benefit from future improvements.

One potential limitation of the model is its reliance on the quality and diversity of the data it was trained on. If the dataset lacks variety in terms of lighting conditions, angles, and bolt types, the model may not generalize well to all real-world scenarios. Consequently, future work could focus on expanding the dataset, incorporating more varied images, and potentially using synthetic data generation to enhance the model's robustness. Another area for improvement is the model's interpretability. While Grad-CAM visualizations provide some insights into the decision-making process, the model could be further refined to offer more granular explanations of its predictions, thereby increasing trust and ease of use for end-users.

Additionally, the current model architecture, although efficient, may be pushed to its limits when dealing with extremely large-scale or high-resolution images. Exploring more advanced compression and optimization techniques could result in even faster inference times and lower resource consumption without sacrificing accuracy. The model could also be extended to detect more nuanced categories of faults in railway infrastructure beyond the binary classification of bolt presence. Incorporating multi-class detection capabilities and finer-grained fault classifications would make the model more comprehensive and applicable to a broader range of maintenance tasks. Lastly, while the model performs well within the constraints of current hardware, ongoing advancements in processor and GPU technology offer opportunities to improve its performance and efficiency further. Leveraging these advancements could allow the model to handle more complex tasks, such as real-time video analysis, with greater ease and precision.

In summary, while the model stands as a promising tool for railway bolt detection, there is ample potential for enhancement in terms of data diversity, interpretability, scalability, functionality, and leveraging technological advancements, which can be addressed in future research and development efforts.

## 6. Conclusions

In conclusion, this research addresses the vital challenge of detecting missing bolts in railway systems—a significant issue for maintaining safety and operational integrity. The proposed solution, a lightweight convolutional neural network, stands as an effective response, achieving an impressive 96.43% accuracy in identifying the presence or absence of bolts. This level of precision marks a considerable advancement in automated fault detection, offering a dependable and efficient alternative to manual inspections. The effectiveness of the proposed model is further underscored by its computational efficiency, which allows it to be deployed in edge computing environments—crucial for real-time monitoring applications. Its capability to deliver high performance on limited computational resources makes it an excellent candidate for on-site deployment in various railway maintenance scenarios.

Key findings of this research reveal that it is indeed feasible to deploy a deep learning model in resource-constrained environments without compromising on the quality of outcomes. The model's contributions to railway safety are significant; by providing a reliable method for detecting missing bolts, the model can help prevent potential track failures and accidents, thereby enhancing the overall safety of railway operations. Moreover, the model's deployment capabilities position it as a valuable tool for the broader field of deep learning. Its efficiency and accuracy demonstrate that high-performance models can be both computationally economical and operationally viable. This aligns with an increasing need in the field for models that can operate at the edge, close to where data are captured, thus opening up new possibilities for real-time, on-site analytics. This research contributes to the deep learning domain by showcasing a practical application of convolutional neural networks tailored to a specific and critical real-world problem. It illustrates

the potential of neural networks to not only perform complex tasks with high accuracy but also do so with an architecture designed for efficiency and practical deployment.

Looking ahead, there are several promising avenues for future research inspired by the findings and successes of the present study. A key focus should be on the further optimization of the model to enhance its efficiency and accuracy. This could involve exploring novel neural network architectures, more advanced attention mechanisms, or cutting-edge training techniques that could refine the model's ability to discern between different states of bolt presence with even greater precision. Real-world testing represents another critical area for future research. While the model has demonstrated high accuracy in a controlled experimental setup, deploying it in operational railway environments would provide invaluable insights into its performance under varying conditions. Such testing could uncover new challenges, such as dealing with extreme weather conditions, lighting variations, and different types of wear and tear on the bolts, which would inform further model refinements.

Moreover, expanding the application of the model to cover a broader range of railway safety and maintenance tasks holds considerable potential. Beyond detecting missing bolts, the model could be adapted or extended to identify other types of structural faults and wear patterns or even to monitor the health of railway infrastructure more broadly. Each of these applications could benefit from the model's lightweight architecture and computational efficiency, making it a versatile tool for a variety of safety-critical maintenance tasks. In addition to technical optimizations and applications, future research should also consider the integration of the model into comprehensive railway maintenance systems. This could involve developing interfaces and integration protocols that enable seamless communication between the model, sensor data, and maintenance workflows. By embedding the model within the larger ecosystem of railway safety technologies, its findings could directly inform maintenance decisions, scheduling, and resource allocation, further enhancing the safety and reliability of railway operations.

Lastly, the principles and methodologies developed in this study have implications beyond railway safety, suggesting broader applicability to other domains where safety and maintenance are concerns, such as in warehouse pallet detection [42], PV crack detection [43], aerospace component detection [44], automotive [45], and micro-cracks in equipment [46]. Investigating these cross-domain applications could not only broaden the impact of the current research but also drive innovation in the field of deep learning and its practical applications. In summary, future research directions offer exciting prospects for enhancing the model's capabilities, validating its effectiveness in real-world settings, and exploring new applications that extend its utility across the maintenance and safety domains.

**Author Contributions:** Conceptualization, M.H. and M.A.R.A.; methodology, M.A.R.A.; software, M.A.R.A.; validation, M.A.R.A.; formal analysis, M.A.R.A.; investigation, M.A.R.A.; resources, M.A.R.A.; data curation, M.H.; writing—original draft preparation, M.A.R.A.; writing—review and editing, M.H.; visualization, M.A.R.A.; supervision, M.H.; project administration, M.H.; funding acquisition, M.H. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** Data are available on request.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Thaduri, A.; Kumar, U. Integrated RAMS, LCC and risk assessment for maintenance planning for railways. In *Advances in RAMS Engineering: In Honor of Professor Ajit Kumar Verma on His 60th Birthday*; Springer: Cham, Switzerland, 2020; pp. 261–292.
2. Liu, X.; Saat, M.R.; Barkan, C.P. Analysis of causes of major train derailment and their effect on accident rates. *Transp. Res. Rec.* **2012**, *2289*, 154–163. [[CrossRef](#)]
3. Ngamkhanong, C.; Kaewunruen, S.; Costa, B.J.A. State-of-the-art review of railway track resilience monitoring. *Infrastructures* **2018**, *3*, 3. [[CrossRef](#)]

4. Feng, H.; Jiang, Z.; Xie, F.; Yang, P.; Shi, J.; Chen, L. Automatic fastener classification and defect detection in vision-based railway inspection systems. *IEEE Trans. Instrum. Meas.* **2013**, *63*, 877–888. [CrossRef]
5. Liu, Z.; Song, Y.; Gao, S.; Wang, H. Review of Perspectives on Pantograph-Catenary Interaction Research for High-Speed Railways Operating at 400 km/h and above. *IEEE Trans. Transp. Electr.* **2023**. <https://doi.org/10.1109/TTE.2023.3346379>. [CrossRef]
6. Song, Y.; Lu, X.; Yin, Y.; Liu, Y.; Liu, Z. Optimization of Railway Pantograph-Catenary Systems for Over 350 km/h Based on an Experimentally Validated Model. *IEEE Trans. Ind. Inform.* **2024**, *20*, 7654–7664. [CrossRef]
7. Ahmed, H.; La, H.M.; Gucunski, N. Review of non-destructive civil infrastructure evaluation for bridges: State-of-the-art robotic platforms, sensors and algorithms. *Sensors* **2020**, *20*, 3954. [CrossRef] [PubMed]
8. O’shea, K.; Nash, R. An introduction to convolutional neural networks. *arXiv* **2015**, arXiv:1511.08458.
9. Rashid, M.; Khan, M.A.; Sharif, M.; Raza, M.; Sarfraz, M.M.; Afza, F. Object detection and classification: A joint selection and fusion strategy of deep convolutional neural network and SIFT point features. *Multimed. Tools Appl.* **2019**, *78*, 15751–15777. [CrossRef]
10. Zhang, S.; Benenson, R.; Schiele, B. Citypersons: A diverse dataset for pedestrian detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3213–3221.
11. Schroff, F.; Kalenichenko, D.; Philbin, J. Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 815–823.
12. Alif, M.A.R. State-of-the-Art Bangla Handwritten Character Recognition Using a Modified Resnet-34 Architecture. *Int. J. Innov. Sci. Res. Technol.* **2024**, *9*, 438–448.
13. Alif, M.A.R.; Ahmed, S.; Hasan, M.A. Isolated Bangla handwritten character recognition with convolutional neural network. In Proceedings of the 2017 20th International Conference of Computer and Information Technology (ICCIT), Dhaka, Bangladesh, 22–24 December 2017; pp. 1–6.
14. Masci, J.; Meier, U.; Ciresan, D.; Schmidhuber, J.; Fricout, G. Steel defect classification with max-pooling convolutional neural networks. In Proceedings of the 2012 International Joint Conference on Neural Networks (IJCNN), Brisbane, Australia, 10–15 June 2012; pp. 1–6.
15. Arora, N.; Kumar, Y.; Karkra, R.; Kumar, M. Automatic vehicle detection system in different environment conditions using fast R-CNN. *Multimed. Tools Appl.* **2022**, *81*, 18715–18735. [CrossRef]
16. Zhu, Y.; Zhang, C.; Zhou, D.; Wang, X.; Bai, X.; Liu, W. Traffic sign detection and recognition using fully convolutional network guided proposals. *Neurocomputing* **2016**, *214*, 758–766. [CrossRef]
17. Zhang, L.; Yang, F.; Zhang, Y.D.; Zhu, Y.J. Road crack detection using deep convolutional neural network. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; pp. 3708–3712.
18. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S. An image is worth 16 × 16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
19. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
20. Hossain, S.; Chakrabarty, A.; Gadekallu, T.R.; Alazab, M.; Piran, M.J. Vision transformers, ensemble model, and transfer learning leveraging explainable AI for brain tumor detection and classification. *IEEE J. Biomed. Health Inform.* **2023**, *28*, 1261–1272. [CrossRef] [PubMed]
21. Alif, M.A.R. Attention-Based Automated Pallet Racking Damage Detection. *Int. J. Innov. Sci. Res. Technol.* **2024**, *9*, 728–740.
22. Zhang, Y.; Song, Y.; Hu, H.; Zhang, Y. Mixed-former: Multi-fusion remote sensing change detection. *Int. J. Remote Sens.* **2023**, *44*, 3507–3528. [CrossRef]
23. Wu, J.; Su, X.; Yuan, Q.; Shen, H.; Zhang, L. Multivehicle object tracking in satellite video enhanced by slow features and motion features. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 1–26. [CrossRef]
24. Sarda, A.; Dixit, S.; Bhan, A. Object detection for autonomous driving using yolo [you only look once] algorithm. In Proceedings of the 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV). IEEE, Tirunelveli, India, 4–6 February 2021; pp. 1370–1374.
25. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*. Available online: [https://papers.nips.cc/paper\\_files/paper/2015/hash/14bfa6bb14875e45bba028a21ed38046-Abstract.html](https://papers.nips.cc/paper_files/paper/2015/hash/14bfa6bb14875e45bba028a21ed38046-Abstract.html) (accessed on 28 March 2024). [CrossRef] [PubMed]
26. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part I 14; Springer: Berlin/Heidelberg, Germany, pp. 21–37.
27. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
28. Arrieta, A.B.; Díaz-Rodríguez, N.; Del Ser, J.; Bennetot, A.; Tabik, S.; Barbado, A.; García, S.; Gil-López, S.; Molina, D.; Benjamins, R. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Inf. Fusion* **2020**, *58*, 82–115. [CrossRef]
29. Mazzeo, P.L.; Nitti, M.; Stella, E.; Distanto, A. Visual recognition of fastening bolts for railroad maintenance. *Pattern Recognit. Lett.* **2004**, *25*, 669–677. [CrossRef]



30. Marino, F.; Distante, A.; Mazzeo, P.L.; Stella, E. A real-time visual inspection system for railway maintenance: Automatic hexagonal-headed bolts detection. *IEEE Trans. Syst. Man, Cybern. Part C (Appl. Rev.)* **2007**, *37*, 418–428. [[CrossRef](#)]
31. Liu, L.; Zhou, F.; He, Y. Automated status inspection of fastening bolts on freight trains using a machine vision approach. *Proc. Inst. Mech. Eng. Part F J. Rail Rapid Transit* **2016**, *230*, 1629–1641. [[CrossRef](#)]
32. Wang, T.; Zhang, Z.; Yang, F.; Tsui, K.L. Automatic Detection of Rail Components via A Deep Convolutional Transformer Network. *arXiv* **2021**, arXiv: 2108.02423.
33. Sun, J.; Xie, Y.; Cheng, X. A fast bolt-loosening detection method of running train's key components based on binocular vision. *IEEE Access* **2019**, *7*, 32227–32239. [[CrossRef](#)]
34. Gibert, X.; Patel, V.M.; Chellappa, R. Deep multitask learning for railway track inspection. *IEEE Trans. Intell. Transp. Syst.* **2016**, *18*, 153–164. [[CrossRef](#)]
35. Wang, Z.X.; Tu, X.J.; Gao, X.R.; Peng, C.Y.; Luo, L.; Song, W.W. Bolt detection of key component for high-speed trains based on deep learning. In Proceedings of the 2019 Far East NDT New Technology & Application Forum (FENDT), Qingdao, China, 24–27 June 2019; pp. 192–196.
36. Li, C.; Wei, Z.; Xing, J. Online inspection system for the automatic detection of bolt defects on a freight train. *Proc. Inst. Mech. Eng. Part F J. Rail Rapid Transit* **2016**, *230*, 1213–1226. [[CrossRef](#)]
37. Wang, T.; Zhang, Z.; Yang, F.; Tsui, K.L. Automatic rail component detection based on AttnConv-net. *IEEE Sensors J.* **2021**, *22*, 2379–2388. [[CrossRef](#)]
38. Alif, M.A.R.; Hussain, M.; Tucker, G.; Iwnicki, S. BoltVision: A Comparative Analysis of CNN, CCT, and ViT in Achieving High Accuracy for Missing Bolt Classification in Train Components. *Machines* **2024**, *12*, 93. [[CrossRef](#)]
39. Dou, Y.; Huang, Y.; Li, Q.; Luo, S. A fast template matching-based algorithm for railway bolts detection. *Int. J. Mach. Learn. Cybern.* **2014**, *5*, 835–844.
40. Wang, T.; Yang, F.; Tsui, K.L. Real-time detection of railway track component via one-stage deep learning networks. *Sensors* **2020**, *20*, 4325. [[CrossRef](#)] [[PubMed](#)]
41. Hussain, M.; Hill, R. Custom lightweight convolutional neural network architecture for automated detection of damaged pallet racking in warehousing & distribution centers. *IEEE Access* **2023**, *11*, 58879–58889.
42. Zahid, A.; Hussain, M.; Hill, R.; Al-Aqrabi, H. Lightweight convolutional network for automated photovoltaic defect detection. In Proceedings of the 2023 9th International Conference on Information Technology Trends (ITT), Dubai, United Arab Emirates, 24–25 May 2023; pp. 133–138.
43. Hussain, M.; Al-Aqrabi, H.; Hill, R. PV-CrackNet architecture for filter induced augmentation and micro-cracks detection within a photovoltaic manufacturing facility. *Energies* **2022**, *15*, 8667. [[CrossRef](#)]
44. Liu, Y.; Zhou, X.; Han, H. Lightweight CNN-based method for spacecraft component detection. *Aerospace* **2022**, *9*, 761. [[CrossRef](#)]
45. He, J.; Chen, J.; Liu, J.; Li, H. A lightweight architecture for driver status monitoring via convolutional neural networks. In Proceedings of the 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO), Dali, China, 6–8 December 2019; pp. 388–394.
46. Animashaun, D.; Hussain, M. Automated Micro-Crack Detection within Photovoltaic Manufacturing Facility via Ground Modelling for a Regularized Convolutional Network. *Sensors* **2023**, *23*, 6235. [[CrossRef](#)] [[PubMed](#)]

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.