Article

BoltVision: A Comparative Analysis of CNN, CCT, and ViT in Achieving High Accuracy for Missing Bolt Classification in Train Components

Mujadded Al Rabbani Alif *, Muhammad Hussain, Gareth Tucker and Simon Iwnicki

Department of Computing and Engineering, Huddersfield University, Queensgate, Huddersfield HD1 3DH, UK; m.hussain@hud.ac.uk (M.H.); g.j.tucker@hud.ac.uk (G.T.); s.iwnicki@hud.ac.uk (S.I.)

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

Abstract: Maintenance and safety inspection of trains is a critical element of providing a safe and reliable train service. Checking for the presence of bolts is an essential part of train inspection, which is currently, typically carried out during visual inspections. There is an opportunity to automate bolt inspection using machine vision with edge devices. One particular challenge is the implementation of such inspection mechanisms on edge devices, which necessitates using lighter models to ensure efficiency. Traditional methods have often fallen short of the required object detection performance, thus demonstrating the need for a more advanced approach. To address this challenge, researchers have been exploring the use of deep learning algorithms and computer vision techniques to improve the accuracy and reliability of bolt detection on edge devices. High precision in identifying absent bolts in train components is essential to avoid potential mishaps and system malfunctions. This paper presents “BoltVision”, a comparative analysis of three cutting-edge machine learning models: convolutional neural networks (CNNs), vision transformers (ViTs), and compact convolutional transformers (CCTs). This study illustrates the superior assessment capabilities of these models and discusses their effectiveness in addressing the prevalent issue of edge devices. Results show that BoltVision, utilising a pre-trained ViT base, achieves a remarkable 93% accuracy in classifying missing bolts. These results underscore the potential of BoltVision in tackling specific safety inspection challenges for trains and highlight its effectiveness when deployed on edge devices characterised by constrained computational resources. This attests to the pivotal role of transformer-based architectures in revolutionising predictive maintenance and safety assurance within the rail transportation industry.

Keywords: safety inspection; train components; missing bolt classification; machine learning; transformer models; vision transformer; pre-trained models; transfer learning; railway safety; model evaluation; safety-critical applications; edge devices; data augmentation; confusion matrix analysis; real-world deployment; rail; rolling stock

1. Introduction

The importance of safety inspection of trains lies at the core of maintaining railway operational integrity and minimising potential risks in the modern transportation industry. Systems embedded within trains, such as the axle assemblies, the motive power units, and the braking mechanisms, are among the critical components that constitute the running gear of these transport systems. Since many of these elements are situated externally, particularly those affixed underneath the trains, they are highly susceptible to functional impairments. Their vulnerabilities come from factors like continual vibrations, progressive ageing, and various unpredictable environmental impacts, leading to potential disruptions such as parts displacement, deformation, bolt loss, and fluid leakage. Efficient maintenance and reliable inspection processes are pivotal to ensuring the safe commute of millions
of passengers daily and efficient transportation of goods, highlighting its undeniable importance for societal functioning and economic stability. Identifying missing bolts assumes particular significance among the countless components subject to these inspection processes. Although small, bolts are crucial in securely holding the train components together. Over time, due to wear and tear or even a manufacturing anomaly, bolts may become missing, leading to loose connections and posing a severe safety hazard, including possible derailment. Therefore, timely detection of missing bolts is indispensable for maintaining the reliability and safety of trains, preventing accidents, and minimising the risk of catastrophic failures. This paper addresses the challenges associated with existing methods of bolt detection, emphasising the need for advanced technological solutions, particularly in the context of edge devices, where conventional approaches often fall short.

While deep learning models primarily designed for detecting missing bolts have shown potential, their extensive architecture often makes for unfeasible deployment, notably causing inefficiencies on-device. The standard model compression techniques used to address issues of memory footprint and latency frequently cause a decrease in analytical capabilities, limiting them from achieving the vital high-accuracy inspection of critical train components. Furthermore, traditional techniques reliant on feature extraction or rule-based solutions prove equally unsuitable. Models crafted to cater to standard cloud computing systems face many challenges when transitioning for execution on edge devices, which are characterised by constrained resources. Specifically, convolutional layers in deep CNNs entail substantial MACs (multiply and accumulate) operations. Local receptive fields focus on partial context. Two-stage object detectors like Faster R-CNN require costly region proposal generation. To address these, we propose evaluating convolutional vision transformers like ViT and CCT. Their attention mechanism captures global context relations while needing fewer parameters. ViT patchification and CCT hybrid design enhance computational performance. Compared to CNNs, transformers demonstrate generalised reasoning, not just statistical learning. Pre-training on large datasets improves robustness. We benchmark transformer architectures against CNNs to identify optimal edge-based missing bolt classifiers attaining high accuracy under tight resource constraints. Moreover, factors such as limited computing capacity, finite memory and power quotas, stringent requirements of runtime efficiency, and connection considerations for isolated operations render several existing solutions inefficient. Weak processors struggle with floating point operations, leading to slow inference that is unsuitable for real-time needs. Minimal storage is inadequate for retaining huge datasets commonly relied on by data-hungry deep-learning algorithms. The absence of cloud connectivity impedes model updates or offloading. Together, these hurdles necessitate highly optimised architectures. The proposed vision transformers offer solutions through multiple advantages. Their attention mechanism reduces parameters in comparison to convolutional kernels. Patch embedding projections shrink the input footprint. Positional encoding circumvents rely solely on data. Hybrid designs fuse the benefits of CNN and transformer architectures. Additionally, large-scale pre-training provides generalisable feature representations, limiting data needs during fine-tuning. Overall, the transformers require fewer parameters, operations, and data samples. Our study empirically evaluates them against standard CNNs on missing bolt classification to quantify computational accuracy trade-offs. The experiments inform guidelines for transitioning state-of-the-art deep learning to safety-critical railway inspection while accounting for edge hardware realities.

Conventionally, computer vision methodologies rely heavily on object detection for bolt localisation, which inevitably places further constraints. Dense object detectors notably strain edge devices with limited resources, given their high computational demand. Single-stage detectors, such as YOLO [1] and SSD [2], fall short in localisation accuracy compared to their dual-stage counterparts. However, two-stage frameworks like Faster R-CNN [3] often present high latency periods, making them unfit for real-time inference applications. Hence, these detector architectures can easily overwhelm edge devices due to intricate multiphase computations, leading to substandard performance even with
elaborate parameter tuning. Since safety-critical tasks demand precise localisation, object detection presents considerable challenges on edge hardware. As a result, classification has emerged as a more reliable and efficient alternative for pinpointing missing bolts on such devices. Recent years have seen significant progress in image classification, bolstered by the introduction of sophisticated neural network structures, including convolutional neural networks (CNNs) [4] and recurrent neural networks (RNNs) [5]. These advancements have driven groundbreaking developments in computer vision, natural language processing, and speech recognition.

Particularly, CNNs have shown exemplary performance in image classification tasks, aided by advanced models like VGGNet [6], ResNet [7], and Inception [8], among others. These models draw their power from deep layers of interconnected neurons, which are made feasible by large-scale datasets and potent computational resources such as GPUs and cloud computing. This has enhanced accuracy, expedited training times, and expanded capabilities to address complex real-world problems such as diabetic retinopathy detection [9], pallet racking inspection [10], emotion detection [11], etc. Moreover, transfer learning [12] and pre-trained models [13] have gained popularity, enabling knowledge to shift from one domain to another with minimal training on limited datasets. Novel deep learning models like vision transformers (ViTs) [14], which leverage self-attention mechanisms, have also demonstrated promising outcomes. We aim to combine the strength of these deep learning algorithms to devise a precise and robust system capable of identifying and localising damage in rail component systems. By predominantly focusing on the recognition aspect, classification models optimise accuracy without imposing substantial computational costs for localisation. With their streamlined predictions, these models seem well-suited to undertake the task of missing bolt detection on edge devices.

Through our research, we aspire to significantly elevate the precision of railway component inspections and minimise potential transport hazards, thereby contributing to overall safety within the industry. Furthermore, while transformer-based solutions such as the vision transformer (ViT) [14] and compact convolutional transformer (CCT) [15] have shown significant promise for complex vision tasks, our research aims to investigate and introduce these advanced architectures as viable alternatives to existing bolt detection methodologies. Notably, these transformer models can potentially achieve high-accuracy detection with fewer computational resources than conventional mechanisms, providing suitability for deployment on resource-constrained edge devices. As part of our objectives, we intend to evaluate tailored variants of ViT and CCT optimized specifically for the bolt classification task at hand so as to integrate these innovative solutions into mainstream safety inspection practices within the railway sector, potentially spearheading a comprehensive transformation in monitoring approaches. The core contributions encompass (a) the creation of a realistic bolts image dataset, (b) quantitative benchmarking of CNNs, ViTs, and CCTs for missing bolt identification capability and computational viability analysis, (c) determination of the optimal technique, and (d) assessment of deployability constraints to inform potential adoption procedures.

This paper is structured as follows: Section I covers the introduction, emphasising the significance of safety inspections for trains while highlighting the limitations of traditional machine vision techniques for missing bolt identification. It establishes the context, needs, and objectives of developing an accurate computer vision-based solution. Section 2 reviews existing literature on computer vision approaches for detection and classification tasks related to train components, fasteners, and prior work leveraging vision transformers. This discussion of current research contextualises BoltVision’s innovations and distinguishing factors. Section 3 presents the methodology, dataset details, proposed architectures, and our training process. Section 4 documents the quantitative results, comparative analysis, key inferences, and critical discussion regarding the evaluation of the CNN, ViT, and CCT models from our experiments. Section 5 discusses the efficacy of BoltVision for addressing railway inspection use cases, outlines its merits over existing solutions, and considers
the potential scope for further enhancements in future work. Finally, Section 6 concludes the study.

By presenting a detailed comparative study between convolutional and transformer architectures for missing bolt classification in trains, validated through rigorous experimentation, this paper makes significant contributions to research at the intersection of computer vision, predictive maintenance, and transportation infrastructure monitoring. It offers unique insights into the promise of vision transformers, so far unexplored, for precision safety inspection in railways. The findings reveal their superiority in striving for accuracy and computational efficiency, which are vital for practical edge deployment to enable real-time embedded assessments during train runs. By highlighting BoltVision’s capabilities in a novel application context, this work expands the academic knowledge base for leveraging state-of-the-art deep learning in the railway industry to build safer, more reliable next-generation transportation networks.

2. Related Work

2.1. Bolt Detection Using Machine Learning

Safety inspection in transportation, particularly within the context of train components, has undergone significant evolution over the years. Historically, safety protocols have adapted to the growing complexity of train structures, emphasising the critical importance of thorough inspections to ensure the reliability and security of these systems. Based on visual inspections, detecting missing bolts on trains is typically a manual activity; however, machine vision using rule-based algorithms is starting to be used. Manual inspections, although traditional, are inherently limited by human capabilities and subjectivity. On the other hand, rule-based algorithms often need more adaptability to discern nuanced variations in bolt configurations, leading to challenges in achieving the desired level of accuracy.

Specifically in the context of bolts and threaded fasteners, Dou et al. proposed a fast template matching-based algorithm for detecting missing bolts in railway maintenance [16]. The researchers achieved high accuracy and efficiency in detecting missing bolts, and they overcame the disadvantages of traditional feature extraction and classification techniques. However, the algorithm had some limitations, such as the manual selection of templates and the requirement for many samples to train the algorithm for a new railway. In a similar research field, Liu et al. proposed a visual inspection system that can automatically inspect the status of fastening bolts on freight trains, reducing costs and avoiding traffic accidents [17]. The proposed hierarchical detection framework comprises fault area extraction and fastening bolt detection using a grey projection method, a fastening bolt detector with gradient-orientation-based features, and a support vector machine classifier. However, the researchers also mentioned that while the proposed system achieves high accuracy and inspection speed, it may perform better in certain environmental conditions or with specific fastening bolts. Ramana et al. proposed a novel approach to identifying loose civil structural bolts using a smartphone camera and support vector machines [18]. The researchers trained the Viola–Jones algorithm using two datasets of images—one with bolts and one without bolts where the aim was to locate all the bolts in the images. The algorithm calculated features that were then used to train a support vector machine. The support vector machine generated a decision boundary that separates the loosened bolts from the tight bolts.

Nonetheless, the researchers scrutinised the proposed method and identified several limitations. These included imprecise bolt localisations due to fluctuating illuminations, spurious feature extractions caused by intense reflections at the bolt head, and unreliable Hough transforms that impeded precise feature extraction. Ultimately, these limitations resulted in a significant decline in performance when distinguishing between bolts that were only slightly loosened versus wholly tightened. They also found that the vertical angle changes affected the accuracy of the classification, and a few false classifications resulted from the wrong ellipse detection on the bolt head.
2.2. Bolt Detection Using Deep Learning

With the rise of deep learning, a new chapter commenced in object detection. Multiple research works have explored convolutional neural networks (CNNs) for detecting different objects, including bolts in industrial components. Wang et al. introduced a novel approach for detecting bolts on critical components of high-speed trains using ResNet [19]. The proposed method has high precision, more robust universality, and robustness and reduces the cost of the workforce while improving the stability of detection. The researchers argued that the proposed method is more effective than traditional computer vision methods. Additionally, the paper notes that previous computer vision-based methods for detecting abnormal targets by contrasting them with standard images are heuristic and can result in many false alarms. Zhou et al. propose a method that combines traditional visual inspection with deep learning by implementing a stacked auto-encoder convolutional neural network (SAE-CNN) to improve inspection accuracy for automated visual inspection of target parts for train safety [20]. The researchers achieved an accuracy rate of over 98% in their final experiment using the proposed visual inspection method combining traditional feature extraction with deep learning.

Similarly, Wang et al. proposed a new method for detecting defects in high-speed trains using bi-temporal images and two convolutional neural networks [21]. The researchers noted the lack of accurate multiclassification datasets, which posed a challenge. However, they overcame this issue by approaching change classification as a binary classification task and focusing on whether changes are hazardous. As mentioned in the research, convolutional neural networks (CNNs) have been successfully used in various image recognition tasks, including object detection. However, the resource-intensive nature of these models is a cause for concern, particularly when deploying them on edge devices with limited computational capabilities.

2.3. Transformer Based Applications

This literature review also highlights the growing relevance of transformer-based architectures in machine learning. The vision transformer (ViT), known for its attention-based mechanism, offers a paradigm shift in image recognition tasks, demonstrating superior performance in various applications [22–24]. The inherent ability of ViT to capture global contextual information through self-attention mechanisms makes it particularly promising for tasks involving complex visual patterns [25], such as those encountered in safety inspection for train components. Its success in image classification tasks has spurred interest in its applicability to the intricate nature of identifying missing bolts within the context of train structures. However, while ViT shows great promise, there remains a need for a focused exploration of its effectiveness in safety inspection scenarios specific to train components. A critical analysis of existing literature reveals a gap in current research, specifically in addressing the challenges of safety inspection in train components, especially the need for lightweight models suitable for edge devices. This gap motivates the exploration presented in this paper. BoltVision leverages transformer-based architectures to achieve high accuracy in missing bolt classification, addressing the identified limitations in current safety inspection methodologies.

In summary, this literature review establishes the historical context of safety inspection in train components, critiques existing methods, explores advances in machine learning, and identifies the gaps BoltVision aims to fill in the pursuit of more accurate and efficient safety inspection processes. From Table 1, it can be observed as evident that while existing literature has made advancements in bolt detection, limitations persist due to reliance on hand-crafted features, constrained operating conditions, and lack of generalization capability. A key innovation in this paper is the exploration of transformer-based deep learning techniques like ViT and CCT that offer enhanced representational learning to automatically distinguish between missing and intact bolts across diverse inspection scenarios. This study benchmarks these data-driven models against CNNs to deliver practical insights into advancing safety-critical railway monitoring leveraging AI.
Table 1. Comparison of existing works.

<table>
<thead>
<tr>
<th>Research</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dou et al. [16]</td>
<td>- High accuracy and efficiency</td>
<td>- Needs manual template selection</td>
</tr>
<tr>
<td></td>
<td>- Overcomes limitations of traditional techniques</td>
<td>- Many samples needed to train for any new railway</td>
</tr>
<tr>
<td>Liu et al. [17]</td>
<td>- High accuracy and inspection speed</td>
<td>- Performance varies across environment and bolt type</td>
</tr>
<tr>
<td></td>
<td>- Hierarchical detection framework</td>
<td></td>
</tr>
<tr>
<td>Ramana et al. [18]</td>
<td>- Novel smartphone-based approach</td>
<td>- Unreliable under varying illumination</td>
</tr>
<tr>
<td></td>
<td>- Leverages SVM for classification</td>
<td>- False classifications due to wrong ellipse detection</td>
</tr>
<tr>
<td>Wang et al. [19]</td>
<td>- High Precision</td>
<td>- A multiclassification dataset was available, and some results were ambiguous</td>
</tr>
<tr>
<td></td>
<td>- Robust universality</td>
<td></td>
</tr>
<tr>
<td>Zhou et al. [20]</td>
<td>- Combines traditional and deep learning</td>
<td>- No assessment of model efficiency</td>
</tr>
<tr>
<td></td>
<td>- Accuracy over 98%</td>
<td></td>
</tr>
<tr>
<td>Wang et al. [21]</td>
<td>- Handles multiclassification</td>
<td>- Very resource intensive</td>
</tr>
<tr>
<td></td>
<td>- Focuses on hazardous changes</td>
<td>- No computational profiling</td>
</tr>
</tbody>
</table>

3. Methodology

3.1. Dataset

As with any machine learning effort, an indispensable aspect of our research strategy is the selection and compilation of a robust dataset to train and assess the effectiveness of our “BoltVision” model. The dataset used in this study has been collected from actual train bogies axles using a 12 MP iPhone camera manufactured by Apple Inc., headquartered in Cupertino, California. This camera was chosen to simulate a Raspberry Pi imaging sensor that may be deployed for edge-based inference in practical railway environments. The author manually annotated the entire assemblage to categorize each image as either containing bolts or missing bolts. Careful consideration was given to capturing visuals reflecting the complexities of real-world inspection conditions, including occlusions, rust, oil stains, scaling, and variability across train component subsections. The images focus specifically on sections containing bolts, depicting samples of intact and missing bolts.

The images were captured with meticulous attention to the operating environment and the positioning of the device. The dataset comprises a diverse set of 145 images, resized to a resolution of $224 \times 224$ for standardisation across the dataset, and evenly divided into two categories—bolt-present and bolt-missing imagery. This specific categorisation allows the model to learn and recognise patterns associated with the two states of bolt-related scenarios in train components.

During the image capture process, every effort was made to emulate the viewpoint of the proposed edge device in its prospective deployment setting. This careful curation approach ensures practical training and evaluation of our “BoltVision” system applied to real-world scenarios. The dataset provides an essential foundation for the proposed bolt detection mechanism, adapting convolutional neural network models, vision transformers (ViTs), and compact convolutional transformers (CCTs) to analyse images accurately. In executing this dataset curation strategy, we aimed to devise a ground-truth set that would serve as a solid basis for our research, catering to various real-world conditions found in railway component inspection scenarios.
Figure 1 displays representative images from the two categories of our dataset—instances with bolts present and those with missing bolts. In images corresponding to the bolt-present class (Figure 1A), the classification task is direct, as the model’s primary objective is to discern the presence of bolts amidst the background components. In contrast, images belonging to the bolt-missing class (Figure 1B) are characterised by the absence of bolts in several instances. For example, the bolt absence is strikingly apparent in the right image showcased in Figure 1B. Alternatively, the middle image of Figure 1B contains an instance where a missing bolt is not immediately identifiable compared to the images located furthest to the right and left. It is crucial to recognise that all images within our dataset comprise a background context, which must be accounted for when implementing augmentation techniques. This preserves the environmental context during the modelling process.

![Figure 1](image_url)

**Figure 1.** Data sample of a bolt dataset: (A) bolt-present and (B) bolt-missing.

Through the compilation of this fundamental dataset, our endeavours aim to establish a robust base for training and assessment of the proposed “BoltVision” system employing convolutional neural network models supplemented by attention mechanisms. Our dataset incorporates diverse images depicting the present and missing bolts, enabling the proposed model to learn and extrapolate patterns associated with the different bolt conditions. However, potential sources of bias may stem from the limited dataset size as well as singleton manual annotation, allowing the possibility of labelling errors or missed edge cases. Usage of a consistent camera type under similar lighting conditions can potentially overfit models to the specific visual signature. However, the diversity of components, damage modes, and imaging angles coupled with balanced category distribution helps mitigate these risks and facilitate more robust model learning translatable to real testbeds. Future work should focus on expanding the visual data corpus through additional railway asset images under various operational and environmental settings to further enhance generalisation.

### 3.2. Data Augmentation

Data augmentation techniques are leveraged during training to expand the diversity of images available for learning robust data representations. In the context of our paper on BoltVision, data augmentation assumes paramount importance as it serves as a pivotal strategy to enhance the model’s efficacy in safety inspection for train components. The diverse and realistic variations introduced through data augmentation simulate the dy-
dynamic conditions that the model may encounter during deployment. In the construction of BoltVision, a dual-framework approach was employed, utilising Keras with TensorFlow for the custom model and the PyTorch framework for pre-trained models like the vision transformer (ViT). The data augmentation strategies varied between the custom model and PyTorch-based pre-trained models.

For the custom model developed using Keras and TensorFlow, feature-wise normalisation, centre shearing, and a suite of other augmentations ensure the model’s adaptability to varying perspectives, lighting conditions, and potential distortions in real-world scenarios, as shown in Figure 2.

Figure 2. The data augmentation techniques are applied to an image sample for custom models.

3.2.1. Feature-Wise Normalization

\[ x_{\text{norm}} = \frac{x - \mu}{\sigma} \]  

(1)

This rescales the pixel intensities across channels as per Equation (1), where \( x \) is the input, \( \mu \) is the mean, and \( \sigma \) is the standard deviation. By normalising each channel independently, model overfitting is reduced.

3.2.2. Feature-Wise Centre

\[ x_{\text{centered}} = x - \mu \]  

(2)

Similar to normalisation, feature-wise centring independently shifts the mean of each channel to zero; based on Equation (2), data distribution centring further improves generalisation by removing the mean image.

3.2.3. Shearing Transformation

Geometric image warping is applied by selectively skewing rows of pixels horizontally or vertically. By generating sheared versions (Figure 3), model robustness against affine transformations improves.
3.2.3. Shearing Transformation

Geometric image warping is applied by selectively skewing rows of pixels horizontally or vertically. By generating sheared versions (Figure 3), model robustness against position variations.

3.2.4. Additional Augmentations

Apart from normalisation, centring, and shearing, several other augmentations are applied to the image dataset to enable robust model generalisation:

**Width/Height Shifting:** Images are shifted horizontally or vertically by fractions of the image dimensions as in Equation (3). This induces translational invariance against position variations.

\[
\hat{x} = shift(x, \Delta w, \Delta h)
\]

(3)

where \(\Delta w = \alpha * W, \Delta h = \alpha * H\), and \(\alpha\) is chosen randomly between [0,0.2].

**Rotation:** Arbitrary image rotation by angle \(\theta\) Equation (4) increases model invariance to bolt orientation. Values of \(\theta\) vary randomly for each epoch between \(-45^\circ\) to \(+45^\circ\).

\[
H = \begin{bmatrix}
\cos \theta & -\sin \theta \\
\sin \theta & \cos \theta
\end{bmatrix}
\]

(4)

**Zooming:** Images are resized by a random scale factor \(S\) as per Equation (5) to simulate distance variation. This augments scale invariance.

\[
\hat{x} = zoom(x, S)
\]

(5)

where \(S\) is randomly picked in [0.5,1.5].

**Brightness:** By increasing or decreasing brightness, as in Equation (6), the model becomes agnostic to illumination variations.

\[
\hat{x} = x + \beta
\]

(6)

where \(\beta\) varies randomly between \([-0.2,0.2]\).

Along with flipping and fill-mode padding, these augmentations equip the model to handle practical complexities associated with missing bolt classification from in situ edge deployment perspectives.

For the vision transformer architecture leveraged through PyTorch, we utilise the default augmentations employed during the original ViT pre-training. These include random cropping on input images of size \(224 \times 224\), along with horizontal flipping. The default augmentations aim to simulate real-world variations encountered during image capture, facilitating the model’s ability to generalise across different scenarios. Specifically, random horizontal flips are applied to replicate mirror images, rotations simulate varying orientations, and adjustments in brightness and contrast emulate changes in lighting conditions [26]. By reusing the same strategies, transforming invariances learned during pre-training transfer leads to performance improvements on our bolt classification task.

---

**Figure 3.** Shearing applied to an image sample.

---
In summary, this dual-strategy approach ensures that the custom model benefits from tailored augmentations while pre-trained models like ViT retain the augmentation strategies embedded in their original training, collectively fostering a robust and versatile training dataset for BoltVision. Our data augmentation strategy, with its diverse set of techniques, was designed to avoid overfitting, improve model generalisation, and eventually enhance the overall performance of our BoltVision system.

3.3. Model Architectures

Within machine learning, the choice of model architecture forms the fundamental building block driving performance capability on a given task. Architectural innovations focused on specialised components for improved representation learning and clever inductive biases for efficient generalisation remain pivotal to advancing the frontiers of computer vision. In our study, BoltVision explores a diverse set of model architectures, experimenting with convolutional neural networks (CNNs), both in sequential and attention-based configurations, a vision transformer (ViT) in its pre-trained and custom-formulated variants, as well as the custom compact convolutional transformer (CCT).

The selection of the specific architectures exploited in “BoltVision” in the form of convolutional neural networks (CNNs), vision transformers (ViTs), and compact convolutional transformers (CCTs) was underpinned for several reasons. CNNs have been at the forefront of image classification tasks for years due to their proven practical ability to self-learn and extract salient features from images, making them a natural choice in developing a solution for the bolt classification \[27\]. However, while CNNs are robust in spatial feature extraction, they need the ability to understand the broader contextual information within images, a gap aptly filled by transformer architectures \[28\]. Vision transformers (ViTs) and compact convolutional transformers (CCTs) can capture complex feature relationships across an entire image due to their inherent self-attention mechanism, enabling superior understandings of contextual patterns compared to traditional networks. Furthermore, ViT and CCT are less reliant on computationally expensive convolution operations, making them a desirable option for edge device deployment where computational resources are limited \[29\]. Therefore, the combination of these architectures in our study capitalised on the strengths of both CNNs and transformer models to develop an efficient, high-performing, and context-aware system for bolt detection.

3.3.1. Sequential CNN

As a baseline model, we implement a sequential convolutional neural network (CNN) architecture for the binary missing bolt classification task. The model comprises a series of convolutional blocks sequenced hierarchically for progressively extracting spatial visual features and mapping them into successively abstract representational spaces. The architecture of the sequential CNN is given in Figure 4.

![Figure 4. The architecture of the sequential CNN.](image)

**Input Layer:** The input colour images of dimension 224 × 224 pixels are normalised but retain their three channels (RGB).

**Convolutional Block:** Each convolutional block consists of a 2D convolutional layer, activating through a ReLU non-linearity, followed by max-pooling for spatial downsam-
pling. Convolution operation for feature extraction is defined in Equation (7), where \( l \) denotes layer index, \( x \) is input, \( k \) refers to learned filters, \( b \) is bias, and \( * \) denotes convolution. Multiple such blocks enhance representational capacity.

\[
z_l = ReLU \left( k_l * x + b_l \right)
\]  

(7)

**Flatten and Fully Connected Blocks:** The convolutional feature hierarchy gets flattened into a unidimensional vector fed as input to a sequence of fully connected layers for classification. Dropout regularisation is utilised during training.

**Output Layer:** Final binary bolt-missing/present classification probabilities are computed through a SoftMax output layer as in Equation (8), where \( z \) is the pre-activation input and the output denotes the probability prediction for each class.

\[
softmax(z_i) = \frac{e^{z_i}}{\sum e^{z_j}}
\]  

(8)

Through systematic experimentation and analysis of this CNN architecture, we determine an effective baseline missing bolt classification performance to quantify the benefits of more complex vision models.

### 3.3.2. Attention-Based Convolutional Neural Network

In addition to the sequential CNN, we experimented with an attention-mechanism-enhanced CNN model for missing bolt classification. The architecture explored comprises 15 CNN convolutions for enriched multi-scale representational learning as proposed by the paper [30]. In each connection, the input undergoes a series of operations, including a convolutional operation, batch normalisation to stabilise training, and a max-pooling operation for spatial downsampling. The resulting feature maps from the connections are concatenated to form a comprehensive representation of spatial hierarchies captured at different scales. The architecture of the attention-based CNN can be seen in Figure 5.

This concatenated feature map is then processed through a dense layer with a single unit and a SoftMax activation function.

**Input Layer:** Based on the specifications, the input layer of the model is designed to accept grayscale images with a dimension of \( 224 \times 224 \) and three channels, representing the red, green, and blue colour channels. The model can process images with high detail and colour accuracy, allowing for more precise image recognition and analysis. The dimensions of the input layer are critical in determining the quality and accuracy of the model’s output, making it essential to ensure that the input images are of the correct size and colour depth for optimal results.

**Convolutional Layers:** The attention CNN architecture in BoltVision incorporates 16 connections of CNN layers, each utilising a distinct kernel size \( 3 \times 3 \). These connections allow the model to capture features at multiple scales, enabling a more comprehensive understanding of spatial hierarchies within the input images. The ReLU activation function is applied to all layers. Every parallel connection has a batch normalisation layer and a max-pooling layer with a size of \( 2 \times 2 \).

**Attention Mechanism:** The output of the 7th, 10th, and 13th convolutional layer has been subjected to an attention mechanism that employs a dense layer of 1 unit and SoftMax activation to obtain attention weights. The resulting attention output is obtained by multiplying the concatenated output and the attention weights. This is represented by Equations (9) and (10) in the attention mechanism equation.

\[
E = \text{Softmax} \left( \frac{QW_o K W_k^T}{\sqrt{d_k}} \right)
\]  

(9)

\[
C = E(VW_o)
\]  

(10)
In this equation, we use \( Q \) to represent the query vector, which reflects the current state of the network or decoder. \( K \) represents the key vectors that reflect the input or encoder states, while \( V \) represents the value vectors associated with those states. \( E \) is the attention matrix, which assigns importance weights to the input states. We apply the Softmax function to the scaled dot product of \( Q \) and \( K \) to calculate \( E \). The resulting attention matrix reflects the importance assigned to each input state. Finally, we calculate the context vector \( C \) by multiplying the attention matrix with the value vectors.

**Dense Layer with SoftMax Activation:** The result of the attention is then passed through a dense layer with a single unit. The SoftMax activation function is applied to the output, ensuring the final prediction represents the attention-weighted combination of features from different kernel sizes. This step is crucial for the model to make informed decisions regarding the presence or absence of missing bolts.

**Flatten Layer:** Flattening the attention output is crucial to ensure seamless processing. This enables it to be easily inputted into dense layers, ultimately leading to better results.

**Fully Connected Layers:** Following the convolutional layers and the attention component, the architecture incorporates two dense layers that are fully connected. The first layer comprises 512 units and the second has 256 units, both using Relu activation. Moreover, each layer involves a batch normalisation layer to ensure performance regularisation.

**Output Layer:** The output is produced by a final layer with three units and SoftMax activation.

**Figure 5.** The architecture of the attention-based CNN.

### 3.3.3. Compact Convolutional Transformer (CCT)

The compact convolutional transformer (CCT) is a novel architecture developed by BoltVision that effectively combines the benefits of convolutional layers and transformer-based models. This innovative approach successfully bridges the gap between the two architectures, resulting in a high-performing and efficient model [29]. The CCT’s design
prioritises efficiency without sacrificing performance, making it ideal for real-time data processing applications. By integrating the strengths of both architectures, CCT provides a powerful tool for a range of tasks that require image or sequence processing. Figure 6 presents the architecture of the CCT.

![Figure 6. The architecture of the CCT.](image)

**Convolutional Tokenizer:** This section aims to extract critical characteristics from the given image. This is accomplished through a series of convolutional layers, utilising a kernel size of 3 and a stride of 1 and incorporating 1-unit padding. Then, a pooling operation is executed. This procedure results in a collection of patches, each representing a distinct image segment. These patches are subsequently transmitted to the transformer encoder block for further refinement.

**Transformer with Sequence Pooling:** The first component of this layer is known as the transformer encoder, which is designed to understand the relationship between different patches obtained by the convolutional tokenizer block. It comprises eight transformer layers and four attention heads, with 64 projection dimensions. This section utilises a multi-head attention method that enables the model to focus on specific areas of the image while considering the entire image. The output of this process is then directed to sequence pooling, which uses an attention-based technique to pool over the sequence of tokens. This refinement reduces computation slightly by forwarding one less token.

The transformer encoder block improves the features extracted by the multi-head attention mechanism using a feed-forward network (MLP). The transformer units of the transformer encoder have been set to 128. The output of the transformer encoder block is then processed through several FC layers to produce the final classification. These FC layers consist of three dense layers—the first dense layer has 512 units, the second dense layer has 256 units, and the final dense layer has 3 units. The last layer applies a SoftMax activation function to output the probability of each class.

The CCT architecture in BoltVision balances computational efficiency and performance, providing a promising solution for safety inspection tasks. Integrating convolutional and transformer components allows CCT to capture local and global features effectively, making it a valuable contender in exploring models for missing bolt classification.

### 3.3.4. Vision Transformer (ViT)

The vision transformer (ViT) implemented in BoltVision represents a pioneering architecture that leverages the transformer model’s success in natural language processing for image classification tasks inspired by previous research [14]. ViT replaces traditional convolutional layers with self-attention mechanisms, enabling the model to capture global contextual information efficiently. The architecture of ViT is illustrated in Figure 7.
The transformer encoder block improves the features extracted by the multi-head attention mechanism using a feed-forward network (MLP). The transformer units of the transformer encoder have been set to 128. The output of the transformer encoder block is then processed through several FC layers to produce the final classification. These FC layers consist of three dense layers—the first dense layer has 512 units, the second dense layer has 256 units, and the final dense layer has 3 units. The last layer applies a SoftMax activation function to output the probability of each class.

The CCT architecture in BoltVision balances computational efficiency and performance, providing a promising solution for safety inspection tasks. Integrating convolutional and transformer components allows CCT to capture local and global features effectively, making it a valuable contender in exploring models for missing bolt classification.

3.3.4. Vision Transformer (ViT)

The vision transformer (ViT) implemented in BoltVision represents a pioneering architecture that leverages the transformer model’s success in natural language processing for image classification tasks inspired by previous research [14]. ViT replaces traditional convolutional layers with self-attention mechanisms, enabling the model to capture global contextual information efficiently. The architecture of ViT is illustrated in Figure 7.

**Embedding Layer:** The ViT begins with an embedding layer that linearly projects flattened image patches into embedding vectors. This step allows the model to treat the image as a sequence of embeddings, enabling the application of the transformer architecture. Mathematically, the embedding operation is represented by Equation (11).

\[
\text{Embed}(X) = X \cdot W_{\text{embed}} + b_{\text{embed}}
\] (11)

where \(X\) is the flattened image patches, \(W_{\text{embed}}\) is the weight matrix, and \(b_{\text{embed}}\) is the bias term.

**Transformer Encoder:** The core of ViT is the transformer encoder, which is composed of multiple layers of self-attention mechanisms. The attention mechanism allows the model to capture dependencies between different patches, enabling an understanding of the global relationship within the image. Mathematically, the attention-weighted sum is expressed with Equation (12) [14].

\[
\text{Attention}(Q, K, V) = \text{softmax} \left( \frac{QK^T}{\sqrt{d_k}} \right) \cdot V
\] (12)

where \(Q\), \(K\), and \(V\) represent the query, key, and value matrices, respectively, and \(d_k\) are the dimensions of the key vectors.

**Classification Head:** Following the transformer encoder, ViT employs a classification head, typically consisting of fully connected layers, to produce the final output. The output is then processed through a SoftMax activation function for image classification.

3.3.5. Pre-Trained Vision Transformer (Pre-trained ViT)

For enhanced performance and efficiency, BoltVision leverages a pre-trained ViT. Transfer learning is employed, where a ViT model is pre-trained on a large dataset, such as ImageNet, before fine-tuning the specific task of missing bolt classification.

The equations associated with pre-trained ViT involve the same embedding, transformer encoder, and classification head operations described above, with the parameters already optimised during pre-training.
The utilisation of ViT and pre-trained ViT in BoltVision reflects a commitment to harnessing the power of transformer-based architectures for image classification tasks, providing a solid foundation for the accurate identification of missing bolts in train components.

3.4. Model Hyperparameters and Training Setup

3.4.1. Sequential CNN, Attention CNN, CCT, and ViT Models

The key hyperparameters used for training the custom CNN architectures are summarised in Table 2.

Table 2. Common hyperparameters for custom models.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Batch Size</td>
<td>32</td>
</tr>
<tr>
<td>Initial Learning Rate</td>
<td>0.001</td>
</tr>
<tr>
<td>Weight Decay</td>
<td>0.001</td>
</tr>
<tr>
<td>Optimizer</td>
<td>Adam</td>
</tr>
<tr>
<td>Training Epochs</td>
<td>150</td>
</tr>
<tr>
<td>Batch Size</td>
<td>32</td>
</tr>
</tbody>
</table>

A batch size of 32 was utilised during training of the custom models. This enables a sufficient estimate of the gradient for optimising the model while maximising GPU hardware utilisation by leveraging vectorised operations and parallel processing capabilities. The initial learning rate was set to 0.001, which was identified empirically to enable adequate adaptation of model weights without causing instability or fluctuation in the optimisation trajectory, which can lead to poor solutions. A weight decay of 0.001 helps regularise the model by penalising excessive growth in parameter magnitudes selected through successive experiments to balance underfitting and overfitting risks. The Adam optimiser was chosen as it adaptively sets individual adaptive learning rates for each parameter based on estimates of first- and second-order moments, making it well-suited for non-convex deep learning objectives. The training regimen spanned 150 epochs to provide adequate iterations for convergence. Early stopping was used as regularisation by monitoring validation accuracy and halting training if metrics failed to improve by at least 0.0001 in 15 successive epochs. This avoids over-optimization beyond the empirically identified peak. The final model retains the best-performing weights on the validation set for testing.

The TensorFlow library’s CosineDecay function was the learning rate scheduler to enhance optimisation and avert overfitting [31]. The initial learning rate was set at 0.001, gradually decreasing over small steps during training. This strategy ensured swift convergence by commencing with a higher learning rate, progressively fine-tuning the model.

3.4.2. Pre-trained ViT Model

As highlighted in Table 3, an Adam optimiser was utilised for the pre-trained model with a learning rate of 0.001, betas ranging from 0.9 to 0.999, and a batch size of 16. The weight decay parameter was set at 0.1. Additionally, the ReduceLROnPlateau technique from PyTorch was incorporated, dynamically adjusting the learning rate when the validation loss failed to decrease. Except for ReduceLROnPlateau, all the pre-trained model parameters were taken from their original paper [14].

Through these designated hyperparameters and training setups, we ensured effective training of BoltVision with both custom and pre-trained models. The key hyperparameters used for training the models are standardised as per Tables 2 and 3, ensuring parity. All models undergo multiple restarts, with the best validation accuracy run selected for testing to mitigate optimisation disparity. Identical train-validation-test splits enable fair benchmarking.
Table 3. Common hyperparameters for the pre-trained model.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Batch Size</td>
<td>16</td>
</tr>
<tr>
<td>Initial Learning Rate</td>
<td>0.001</td>
</tr>
<tr>
<td>Weight Decay</td>
<td>0.1</td>
</tr>
<tr>
<td>Betas (0.9,0.999)</td>
<td></td>
</tr>
<tr>
<td>Optimizer</td>
<td>Adam</td>
</tr>
<tr>
<td>Training Epochs</td>
<td>150</td>
</tr>
</tbody>
</table>

3.5. Data Partition

In order to facilitate the robust training of our models, it was imperative to partition our dataset into three distinct subsets systematically: training, validation, and testing. Our initial allocation reserved 80% of the dataset for training purposes, with the remaining 20% earmarked exclusively for model testing. However, recognising the importance of optimising model accuracy and mitigating the risk of overfitting, a further subdivision was undertaken within the training set. Expressly, 20% of the training images were meticulously set aside for the validation set, while the remaining 80% constituted the primary subset for actual model training. This meticulous partitioning strategy ensured the highest achievable accuracy during the model development phase and played a pivotal role in averting overfitting risks. The validation set, distinct from the training set, functioned as a crucial monitoring tool, allowing for the continuous evaluation of model performance and facilitating adjustments in the training process. Table 4 displays the number of images per section after the procedure.

Table 4. Data partition result.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training</td>
<td>101</td>
</tr>
<tr>
<td>Validation</td>
<td>28</td>
</tr>
<tr>
<td>Testing</td>
<td>16</td>
</tr>
</tbody>
</table>

Notably, this meticulous data partitioning strategy was consistently applied to the custom and pre-trained models, ensuring a standardised and equitable evaluation framework for both model types.

3.6. Experiment Setup

The experiments conducted in this study leverage a laptop testbed with an AMD Ryzen 9 5900HX CPU having 16 GB DDR4 RAM. An NVIDIA GeForce RTX 3070 GPU with 8 GB GDDR6 memory is utilised to accelerate deep learning computations. Software implementations utilise the Python ecosystem, using Keras and TensorFlow [version 2.13.1] [32] libraries for model development and PyTorch [version 2.0.1+cu118] [33] for pre-trained architectures. Supplementary visualisation and analysis rely on Matplotlib [version 3.8.0] [34] and Pandas [version 2.1.0] packages [35]. This computational framework enabled rapid iteration and validation of complex CNN, ViT, and CCT model architectures, using suitable hardware resources while retaining software flexibility for customisation specific to the missing bolt classification task and dataset. The combined advantages facilitated rigorous benchmarking essential for our comparative assessment.

4. Experiment Result

4.1. Evaluation Metrics

To facilitate extensive performance benchmarking and comparative assessment of the convolutional and transformer architectures examined in this study, we utilise a range of comprehensive evaluation metrics spanning both quantification and visualisation techniques.
4.1.1. Accuracy

The accuracy metric provides an overall measure of correct predictions by comparing the number of true positives, true negatives, false positives and false negatives. It gives a ratio of the number of correct positive predictions made by the model to the total number of actual positive cases in the dataset.

4.1.2. Precision

Precision measures the accuracy of positive predictions, representing the proportion of correctly predicted positive instances among all instances predicted as positive.

4.1.3. Recall

Recall, also known as sensitivity or true positive rate, assesses the model’s ability to capture all positive instances, representing the proportion of correctly predicted positive instances among all actual positive instances.

4.1.4. F1-Score

The F1-score balances precision and recall, offering a harmonic mean of the two. It is beneficial when there is an uneven class distribution.

Collectively, these evaluation metrics offer a holistic understanding of the models’ performance, considering aspects of correctness, the balance between precision and recall, discriminatory power, and accuracy in classifying missing bolts within train components.

4.2. Quantitative Performance Analysis

The quantitative performance analysis aims to comprehensively understand the models’ effectiveness in classifying missing bolts in train components. This section encompasses an overall accuracy comparison, a detailed class-wise performance breakdown, and an assessment of the models’ generalisation capabilities.

4.2.1. Overall Accuracy Comparison

Table 5 summarises the overall accuracy achieved by each of the explored model architectures on the missing bolt classification task using the curated image dataset.

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequential CNN</td>
<td>38.00%</td>
</tr>
<tr>
<td>Attention CNN</td>
<td>71.43%</td>
</tr>
<tr>
<td>CCT</td>
<td>67.85%</td>
</tr>
<tr>
<td>Custom Vit</td>
<td>65.04%</td>
</tr>
<tr>
<td>Pre-trained ViT</td>
<td>92.86%</td>
</tr>
</tbody>
</table>

We find that the fundamental sequential CNN architecture underperforms drastically for this challenging task, with just 38% test accuracy, reflecting its inability to adequately distinguish between missing and intact bolts as may be encountered in complex practical inspection scenarios.

Interestingly, integrating multi-scale visual feature extraction and attention modules into the CNN model leads to a considerable 33% absolute improvement in classification accuracy. Nonetheless, 71% accuracy also signifies substantial scope for improvement over the augmented CNN approach.

With 68% test accuracy, the hybrid CCT model combining merits of convolutions and self-attention outperforms the standalone custom ViT model, achieving 65% accuracy. However, surpassing even the hybrid architecture by clear margins is the predefined pre-trained ViT model, which attains a remarkable 93% classification accuracy, confirming the superiority of vision transformer models for accurate image-based missing bolt detection as required in mission-critical railway monitoring applications.
In summary, the overall accuracy comparison reveals the varying degrees of success achieved by each model architecture in addressing the critical task of missing bolt classification within train components. The substantial accuracy of the pre-trained vision transformer highlights the significance of transfer learning and sets a benchmark for future advancements in safety inspection mechanisms.

4.2.2. Computational Efficiency Analysis

Table 6 compares the parametric and memory requirements of our evaluated models. As the real-world viability of machine learning solutions depends heavily on computational constraints, we empirically profile the shortlisted models to quantify their parameter, memory, and operations intensity crucial for comprehending edge compatibility.

<table>
<thead>
<tr>
<th>Model</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequential CNN</td>
<td>51.8 M</td>
</tr>
<tr>
<td>Attention CNN</td>
<td>154 M</td>
</tr>
<tr>
<td>CCT</td>
<td>0.24 M</td>
</tr>
<tr>
<td>Custom ViT</td>
<td>0.31 M</td>
</tr>
<tr>
<td>Pre-trained ViT</td>
<td>85.8 M</td>
</tr>
</tbody>
</table>

The analysis reveals that large capacity models like the pre-trained ViT (86 M) and attention-augmented CNN (154 M), which demonstrate strong accuracy performance, incur proportionally large parametric scale and memory storage needs during inference. Specifically examining the pre-trained ViT model, while its sizable parameter count enables rich representational power from extensive pre-training, this also causes substantial resource intensity unsuitable for constrained edge devices. However, techniques like knowledge distillation through student–teacher learning and weight quantisation have shown the potential to compress vision transformers by over 4× with minimal accuracy loss. Specialised AI accelerators on edge devices also offer optimised dataflow and reduced precision computations to feasibly execute such larger models.

In conclusion, architectural innovations like the pre-trained ViT that push accuracy boundaries remain pivotal. But corresponding co-design innovations in model compression and specialised inferencing hardware are equally crucial to enable ubiquitous, low-latency deployment critical for safety inspections during train runs. The insights on pre-trained ViT quantify accuracy vs. efficiency trade-offs to inform practical adoption guidelines balancing safety assurance needs with computational realities.

4.2.3. Class-Wise Performance Breakdown

Beyond aggregate overall accuracy, analysing model performance explicitly within each class provides more fine-grained insights. Tables 7 and 8 summarise the precision, recall, and F1-scores on the bolt-present and bolt-missing classes, respectively, for each architecture.

<table>
<thead>
<tr>
<th>Model</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequential CNN</td>
<td>0.38</td>
<td>1.00</td>
<td>0.55</td>
</tr>
<tr>
<td>Attention CNN</td>
<td>0.61</td>
<td>0.72</td>
<td>0.66</td>
</tr>
<tr>
<td>CCT</td>
<td>0.56</td>
<td>0.81</td>
<td>0.66</td>
</tr>
<tr>
<td>Custom ViT</td>
<td>0.66</td>
<td>0.18</td>
<td>0.29</td>
</tr>
<tr>
<td>Pre-trained ViT</td>
<td>1.0</td>
<td>0.82</td>
<td>0.90</td>
</tr>
</tbody>
</table>
Table 8. Bolt-missing class performance.

<table>
<thead>
<tr>
<th>Model</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequential CNN</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Attention CNN</td>
<td>0.8</td>
<td>0.70</td>
<td>0.75</td>
</tr>
<tr>
<td>CCT</td>
<td>0.83</td>
<td>0.59</td>
<td>0.69</td>
</tr>
<tr>
<td>Custom ViT</td>
<td>0.64</td>
<td>0.94</td>
<td>0.76</td>
</tr>
<tr>
<td>Pre-trained ViT</td>
<td>0.89</td>
<td>1.0</td>
<td>0.94</td>
</tr>
</tbody>
</table>

**Bolt-Present Classification:** In the context of bolt-present classification, the sequential CNN exhibited a relatively low precision of 0.38, signifying that while it accurately classified instances where bolts were present, it also produced a notable number of false positives. The attention CNN displayed a balanced performance with a precision of 0.61, indicating improved accuracy in positively identifying bolt-present instances. The compact convolutional transformer (CCT) model demonstrated a commendable balance between precision (0.56) and recall (0.81), yielding an F1-score of 0.66, showcasing its ability to identify bolts while minimising false positives accurately. However, the vision transformer (ViT) showed challenges in precision (0.66) and recall (0.18), resulting in a lower F1-score of 0.285. In stark contrast, the pre-trained vision transformer excelled with perfect precision (1.0) and a high recall (0.82), yielding an impressive F1-score of 0.90, showcasing its efficacy in accurately classifying instances with bolts present.

**Bolt-Missing Classification:** On the other hand, the bolt-missing classification presents a distinct set of challenges. The sequential CNN, unfortunately, exhibited an inability to correctly classify instances where bolts were missing, as indicated by a precision, recall, and F1-score of 0.0. The attention CNN showcased significant improvement with a precision of 0.8, a recall of 0.70, and an F1-score of 0.75, indicating enhanced performance in identifying instances with missing bolts. The CCT model demonstrated notable precision (0.83) and recall (0.59), resulting in an F1-score of 0.69. While the vision transformer (ViT) displayed a high recall (0.94), its precision (0.64) resulted in a slightly lower F1-score of 0.761. The trained vision transformer once again outperformed with a precision of 0.894, perfect recall (1.0), and an impressive F1-score of 0.94, showcasing its robustness in correctly identifying instances with missing bolts.

**Comparison and Analysis:** For the bolt-present class, the sequential CNN displays very high recall (1.0), indicating that all relevant samples are correctly classified. However, the precision is extremely low (0.38), implying many false positives. The F1-score of 0.55 encapsulates this imbalance. In contrast, the attention CNN offers more balanced precision (0.61) and recall (0.72), reflected in the higher F1-score (0.66). This demonstrates enhanced discrimination between true bolt-present cases versus misclassified anomalies. The CCT model follows similar trends as the attention CNN with slightly lower precision (0.56) balanced by higher recall (0.81), culminating in an F1-score of 0.66. This highlights that the hybrid architecture correctly identifies the majority of relevant bolts while also limiting false alarms. Comparatively, the custom ViT model reverses characteristics with higher precision (0.66) but drastically lower recall (0.18), quantified by the poor F1-score of just 0.29. This signals an inability to retain all bolts reliably, possibly due to overfitting on the small dataset. Finally, the pre-trained ViT outdoes previous models on both fronts with perfect precision and 0.82 recall—best summarised by the impressive 0.9 F1 measure, substantiating well-balanced and accurate bolt-present classification capability.

For missing bolts, while CCT offers competitive precision (0.83) and recall (0.59), pre-trained ViT dominates across metrics with 0.894 precision, flawless 1.0 recall, and an F1-score of 0.94. The superiority underscores its efficacy in surface defect detection for transport systems by reliably distinguishing bolt anomalies.

As evident, both CNN models suffer from very low recall in detecting bolt-present class, unable to adequately retain relevant samples. ViT demonstrates the opposite behaviour with poorer precision but high recall. Pre-trained ViT emerges as the most balanced. For missing bolt cases, CNNs and transformers offer complementary precision–recall trade-
offs. But pre-trained ViT again dominates on both fronts to reliably identify safety-critical missing fasteners.

4.3. Key Experimental Inferences

The extensive comparative evaluation of diverse CNN and transformer architectures using multiple performance metrics and visualisations leads to several critical inferences aligned with the original objectives.

4.3.1. Summary Analysis Relative to Objective

This study’s primary objective was to develop an effective model for the safety inspection of trains, explicitly focusing on the classification of missing bolts. The models were rigorously evaluated based on overall accuracy, class-wise performance, and confusion matrix analysis. Notably, the pre-trained vision transformer emerged as the top performer, as shown in Figure 8, showcasing its remarkable accuracy and robustness in addressing the core objective of the research.

![Top Accuracy](image)

**Figure 8.** Top accuracies of the models.

4.3.2. Transformer Superiority Factors

The experimentation highlighted the prominence of transformer-based architectures in the realm of the safety inspection of trains. The vision transformer (ViT) and its pre-trained variant exhibited competitive performance, emphasising the efficacy of self-attention mechanisms and transformer architectures in capturing intricate features crucial for accurate classification. The transformer’s ability to model complex relationships within the image data proved advantageous, particularly in scenarios where traditional convolutional neural networks (CNNs) may face limitations.

In addition to aggregate performance metrics, confusion matrices provide further fine-grained insights into model prediction behaviour by tabulating the actual vs. predicted classification outcomes.

Figure 9 illustrates sample confusion matrices for the best and worst performing models, pre-trained ViT and sequential CNN, respectively, on the test dataset. The sequential CNN matrix reflects very high false negatives, with most missing bolt cases being incorrectly predicted as bolt-present. This signifies the model’s inability to capture discriminative visual patterns characteristic of absent bolts adequately.
Figure 9. Confusion matrices of the best (A) and worst (B) model.

In contrast, the pre-trained ViT confusion matrix demonstrates high diagonals and low off-diagonals, indicating consistency between ground truth and predicted labels for both classes. The minimal class imbalance drives both precision and recall to higher equivalency.

Analysing model-wise confusion matrices reveals how false positives vs. false negatives skew precisions and recalls, allowing customised tuning based on application priority. Transformer superiority stems from balance across correct and incorrect classification proportions for both standard and abnormal cases.

4.3.3. Custom vs. Pre-Trained Model Trade-Offs

The comparative analysis between custom-designed models and pre-trained models provided valuable insights into the trade-offs associated with each approach. While custom models, such as sequential CNN and attention CNN, demonstrated reasonable performance, they required substantial training epochs to achieve competitive accuracy. On the other hand, pre-trained models, especially the pre-trained vision transformer, leveraged transfer learning to excel in accuracy with significantly fewer training epochs. The trade-offs encompass factors such as training time, model complexity, and the availability of large pre-existing datasets for transfer learning. The accuracy curve and the loss curve can be seen in Figures 10 and 11.

In conclusion, the key experimental inferences underscore the pivotal role of transformer architectures, particularly the pre-trained vision transformer, in achieving superior accuracy in safety inspection tasks for train components. The trade-offs between custom and pre-trained models inform decision-making considerations for deploying efficient and effective models in real-world applications. This empirical BoltVision assessment thus delivers practical guidelines for leveraging deep learning to automate critical quality checks during rail maintenance for promptly identifying missing/faulty bolts before they endanger safe transportation.
Figure 10. Train accuracy vs. validation accuracy of models.

Figure 11. Train loss vs. validation loss of models.
5. Discussion

The results obtained carry significant implications for the field of train safety inspection. The superior performance of the pre-trained vision transformer highlights its potential for immediate deployment in real-world applications. The accuracy achieved, particularly in the classification of missing bolts, suggests that leveraging pre-existing knowledge through transfer learning substantially enhances model effectiveness. These implications underscore the practical viability of implementing advanced transformer-based models for safety-critical tasks in the railway industry.

While transformer-based architectures demonstrate significant promise, deploying them for real-time safety inspection requires addressing certain limitations. For instance, the high computational requirements of large vision transformers in terms of FLOPs and latency pose a practical constraint for efficiency-critical edge devices. Additionally, over-reliance on pre-trained weights could inhibit adaptability to newer inspection scenarios not covered in the original training data. There also remain open questions regarding explaining the predictions of such black-box models for trustworthiness. More concerning, the risk of uncontrolled false negatives—though minimal, as evidenced through the accuracy metrics—can severely impact public safety if not handled judiciously. Furthermore, false positives raise maintenance overhead. Quantifying and mitigating these failure modes through explainability, uncertainty estimation, and risk-based assessment of model readiness levels remain crucial areas for future work alongside pursuing pure accuracy gains. Fine-tuning hyperparameters, exploring additional data augmentation strategies, and investigating model ensembling techniques could contribute to refining the model’s performance. Additionally, incorporating domain-specific knowledge and conducting more extensive experiments with diverse datasets could enhance the model’s adaptability to varied conditions encountered in real-world scenarios.

The outcomes of this study pave the way for several promising avenues in future research. Firstly, investigating the interpretability of transformer-based models in safety inspection tasks can contribute to building trust in model predictions. Exploring the robustness of the models under different environmental conditions, such as varying lighting and camera perspectives, remains an essential direction for ensuring reliable performance in practical settings. Furthermore, extending the research to address multi-class classification scenarios and exploring the integration of additional sensor data could broaden the applicability of the models in comprehensive safety inspection frameworks. Additionally, while this study focused specifically on missing bolt classification for train components, the transformer-based models showcase the promising potential for generalisation to other defect detection tasks across railway inspection scenarios. For instance, the visual attention mechanisms allow the models to ignore clutter and focus only on relevant regions containing the components of interest. This facilitates adaptation even to new complex metallic surface types by retraining just the final classifier layers while leveraging the generalised feature representations from pre-training. The ability to classify multiple defect patterns also enables multi-category component damage detection without architectural redesign. In effect, the entire deep neural hierarchy distills a holistic understanding of the ‘anomaly’ itself. Such knowledge transfer could allow tailoring these models to detect issues like cracks, corrosion damage, faulty welds, or leaks across broader locomotive assets like axles, frames, couplers, brakes, and bogies following the reannotation of small additive datasets. In future work, we aim to empirically evaluate such adaptability to related safety-critical visual inspection categories prevalent in rail maintenance routines.

The discussion section concludes by emphasising the practical implications of the results, identifying areas for immediate improvement, and outlining potential research directions to advance the field of safety inspection for train components. The findings contribute to the current understanding of model performance and serve as a catalyst for continued exploration and innovation in this critical domain.
6. Conclusions

In conclusion, the research endeavours in the safety inspection of trains, particularly in the context of missing bolt classification, have yielded valuable insights and promising outcomes. The rigorous evaluation of diverse model architectures has illuminated the potential of transformer-based models, with the pre-trained vision transformer emerging as a standout performer, achieving an impressive accuracy of 92.86%. This success underscores the significance of leveraging transfer learning to enhance the efficiency and accuracy of safety inspection mechanisms.

The discussion of result implications emphasises the immediate applicability of advanced models in real-world scenarios, with the pre-trained vision transformer showcasing its practical viability. The discussion also recognises the scope for improvements, suggesting avenues such as hyperparameter fine-tuning and further exploration of data augmentation strategies to enhance model robustness.

Future research directions are delineated, offering a roadmap for continued exploration and innovation. Addressing interpretability challenges, assessing model robustness under diverse environmental conditions, and expanding the research to encompass multi-class classification scenarios represent pivotal areas for further investigation. Additionally, combining vision transformer ensembles or transferring intermediate representations from large-scale pre-trained models can further optimise accuracy and efficiency, which is crucial for ubiquitous edge deployment.

In summary, this research contributes to the advancement of safety inspection technologies for train components and sets the stage for future developments in the application of transformer-based models. The pre-trained vision transformer’s exceptional performance establishes a benchmark, emphasising the potential for continued advancements in the intersection of machine learning and railway safety. As deep learning technology continues to evolve [36], these findings are poised to catalyse further research and foster the implementation of efficient and reliable safety inspection mechanisms in the railway industry. This research can also be used as a springboard for other domains where lightweight, non-invasive detection is required, such as manufacturing [37] and education [38]. As vision-based AI technology continues to progress rapidly across applications like civil infrastructure monitoring [39,40], the findings from comparative assessment of models like BoltVision will help guide similar safety-critical adoption in the railway industry. By benchmarking diverse deep learning approaches and demonstrating state-of-the-art transformer architectures, this work delivers practical insights into pursuing further accuracy and efficiency gains. Specifically, exploring multi-modal sensor fusion, uncertainty quantification, dataset expansion through railway image synthesis, and optimised edge deployment present promising areas for subsequent investigation. We hope the BoltVision evaluations and discussions of real-world viability will catalyse greater integration of automated, non-invasive inspection to enrich reliability across large-scale transportation systems.

Author Contributions: Conceptualization, M.A.R.A. and M.H.; methodology, M.A.R.A. and G.T.; software, M.A.R.A.; validation, M.A.R.A. and M.H; formal analysis, M.A.R.A.; investigation, M.A.R.A.; resources, M.A.R.A. and G.T.; data curation, M.A.R.A.; writing—original draft preparation, M.A.R.A. and M.H.; writing—review and editing, M.H., G.T. and S.I.; visualisation, 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.

Acknowledgments: We would like to acknowledge Haris M. Khalid for his contribution to the conceptual layout of the research and for providing insights into the layout of the manuscript.

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


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.