Journal of
Imaging

MDPI

Article

RailTrack-DaViT: A Vision Transformer-Based Approach for Automated Railway Track Defect Detection

Aniwat Phaphuangwittayakul [1,2], Napat Harnpornchai [3,*], Fangli Ying [4] and Jinming Zhang [1]

[1] International College of Digital Innovation, Chiang Mai University, Chiang Mai 50200, Thailand; aniwat.ph@cmu.ac.th (A.P.); jinming.z@alumni.cmu.ac.th (J.Z.)
[2] Lancang-Mekong Digital Intelligence (Shijiazhuang) Technology Research Center, Shijiazhuang 051230, China
[3] Faculty of Economics, Chiang Mai University, Chiang Mai 50200, Thailand
[4] State Key Laboratory of Bioreactor Engineering, Department of Computer Science and Engineering, East China University of Science and Technology, Shanghai 200237, China; yfangli@ecust.edu.cn
[*] Correspondence: napateconcmu@gmail.com

Abstract: Railway track defects pose significant safety risks and can lead to accidents, economic losses, and loss of life. Traditional manual inspection methods are either time-consuming, costly, or prone to human error. This paper proposes RailTrack-DaViT, a novel vision transformer-based approach for railway track defect classification. By leveraging the Dual Attention Vision Transformer (DaViT) architecture, RailTrack-DaViT effectively captures both global and local information, enabling accurate defect detection. The model is trained and evaluated on multiple datasets including rail, fastener and fishplate, multi-faults, and ThaiRailTrack. A comprehensive analysis of the model's performance is provided including confusion matrices, training visualizations, and classification metrics. RailTrack-DaViT demonstrates superior performance compared to state-of-the-art CNN-based methods, achieving the highest accuracies: 96.9% on the rail dataset, 98.9% on the fastener and fishplate dataset, and 98.8% on the multi-faults dataset. Moreover, RailTrack-DaViT outperforms baselines on the ThaiRailTrack dataset with 99.2% accuracy, quickly adapts to unseen images, and shows better model stability during fine-tuning. This capability can significantly reduce time consumption when applying the model to novel datasets in practical applications.

Keywords: railway track inspection; vision transformer; computer vision; transportation safety; public transportation monitoring

check for updates

Citation: Phaphuangwittayakul, A.; Harnpornchai, N.; Ying, F.; Zhang, J. RailTrack-DaViT: A Vision Transformer-Based Approach for Automated Railway Track Defect Detection. *J. Imaging* **2024**, *10*, 192. https://doi.org/10.3390/jimaging10080192

Academic Editor: Mrinal Mandal

Received: 22 June 2024
Revised: 2 August 2024
Accepted: 5 August 2024
Published: 7 August 2024

1. Introduction

Railways contribute significantly to economic growth, especially in developing countries [1]. Moreover, railway transportation systems are a more environmentally friendly mode, with less noise pollution and energy consumption compared to airways, roads, and waterways [2,3]. Railway tracks play a crucial role in ensuring the safe and efficient operation of railway transportation [4]. However, railway track defects are a common cause of train accidents, which can lead to injuries, fatalities, and significant economic losses. Therefore, the regular inspection of railway tracks is essential to identify and address potential defects [5]. Traditional railway track inspection methods involve visual and manual inspection by human experts or using specialized vehicles equipped with sensors. However, these methods are time-consuming, costly, and may not always be effective at detecting defects [6,7].

There are alternative methods to detect damage and monitor the railway tracks. Loveday [8] explores a method to detect axial stress in railway tracks by using guided waves. Hashmi et al. [9] replaced the manual fault identification on the railway, which is performed by a railway engineer with on-the-fly feature extraction. This feature extraction is based on a deep learning model to generate spectrograms that can ensure they are less time-consuming and prone to error than traditional systems. Three deep learning models, including Convolutional 1D, Convolutional 2D, and Long-Short Term Memory (LSTM), are used to analyze different lengths of audio samples.

Computer vision is another technique that can be applied to railway track inspection and monitoring. There are a number of approaches proposed for inspecting and monitoring railway tracks using computer vision techniques. Ruvo et al. [10] present a Visual Inspection System for railway maintenance that can detect and track the rail head in a video sequence, reducing the area to be analyzed by using FPGA technology, which is highly flexible and configurable as it is based on classifiers that can be easily reconfigured for different types of rails. Ritika and Rao [11] propose a method to detect track anomalies, such as vegetation overgrowth and sun kinks, using a camera mounted on a moving locomotive and a simulated image pipeline. The Inception V3 model is used for the binary classification of vegetation overgrowth and sun kinks, and the trained model is tested on professionally simulated track videos. Although the model shows that the proposed method can classify track anomalies with high precision, the method relies on a camera mounted on a moving locomotive, which may not be feasible in all situations or for all types of tracks. Moreover, the simulated images used for training the model may not perfectly represent real-world track anomalies, which could affect the accuracy of the model. Gasparini et al. [12] proposed a vision-based framework for detecting obstacles on railways during the night using RGB or thermal images. The framework uses a rail drone equipped with cameras and external light sources to collect data. The collected data are used to train a deep learning model that can accurately detect obstacles. The experiments show that the proposed approach is suitable for implementation on a self-powered drone. Nonetheless, the use of drones to collect data is limited by their unpredictable position and short battery life. Gasparini et al. [13] proposed another framework for the automatic inspection of railways during the night using thermal images. The framework consists of three modules for detecting, localizing, and classifying anomalies. The authors also introduce a new dataset called Vesuvio, which was acquired from a rail drone specifically created for anomaly detection tasks in a railway scenario during the night. However, the power consumption of light sources used for night vision is limited to rail drones that are self-powered. Acquisition cameras with a high spatial resolution are needed to detect even small-sized objects. Gibert et al. [14] proposed a new method for fastener detection in railway tracks using a combination of linear Support Vector Machine (SVM) classifiers and a histogram of oriented gradients features to improve the classification margin. The system can inspect ties for missing or defective rail fastener problems for missing and broken components with grayscale images. Even though these two methods show high accuracy for classification, they are only applicable for grayscale image datasets. In 2016, Gibert et al. [15] proposed an algorithm for the automated inspection of railway tracks using deep convolutional neural networks and a multi-task learning framework to classify different natural color images of materials and fasteners. The proposed algorithm achieved high accuracy in detecting fasteners and ties, demonstrating its effectiveness in detecting different types of fasteners and materials. However, the proposal for the method does not discuss the computational requirements, which could be a limitation in practical applications. Liu et al. [16] present a method to improve the performance of rail fastener defect inspection for multiple railways to ensure the safety of railway operation, including a fastener region location method based on an online learning strategy and a fastener defect recognition method based on a deep convolutional neural network. One limitation of the proposed method is that the increase in the maximum queue length of the online template library improves the detection rate but reduces the detection speed, thus affecting the system's efficiency. Eunus et al. [4] introduced a novel Deep Learning (DL) algorithm named ECARRNet for automatic fault detection in railway tracks. This method aims to reduce accidents on railway tracks, save lives, and prevent disasters.

All the above works rely on Convolutional Neural Networks (CNNs). However, CNNs struggle with capturing long-range dependencies due to their local receptive field limitations [17]. Images with thin structures, complex spatial layouts, or multiple views tend to exhibit long-range dependencies [18,19]. Capturing long-range dependencies is crucial for the tasks that handle the complex and interconnected-structure images [20]. Vision Transformer (ViT) [21] is one method that can effectively handle this type of image. Vision Transformers are currently applied to various visual inspection tasks such as road tunnel defect classification [22] and structural condition assessment [23]. In this study, we employ Vision Transformer instead of CNNs for the railway track inspection task. ViT is a deep learning model architecture that is designed for image recognition tasks. Traditional CNN uses a set of convolutional filters to extract image features, which are then processed by fully connected layers to produce a final prediction. ViT replaces the convolutional layers of CNN with a self-attention [24] mechanism. It processes the input image directly by dividing an image into a sequence of non-overlapping patches of equal size. Then, each patch is embedded linearly in a low-dimensional feature space. These embedded patches form the input sequence for Transformer blocks. In the Transformer blocks, the self-attention mechanism is used to monitor all input patches and capture their dependencies. The output of the self-attention mechanism is then passed through a feedforward neural network to produce the block's final output. The output of the last block is then fed to a classification head, which produces the final prediction. In addition to patch embeddings, ViT includes positional embeddings that provide information about the spatial location of each patch. This enables the model to encode the spatial relationships between patches in an image and capture the long-range dependencies, enabling stronger global context modeling [25–27]. Additionally, ViT can handle images of any size because it processes image patches independently. This makes ViT more adaptable than CNNs, which require the image to be resized or cropped to a fixed size. Furthermore, ViT has shown promising results not only on large datasets but also on small datasets [28]. Moreover, ViT has demonstrated state-of-the-art (SOTA) performance on a variety of image classification tasks, outperforming CNNs on certain benchmark datasets [21].

This paper proposes a novel transformer-based deep learning approach, called RailTrack-DaViT, for detecting defects in railway track images. Railway track images inherently contain long-range dependencies due to the extended, curved geometry of the tracks themselves. Capturing global context and relationships between distant regions is crucial for the classification task. The key contributions are listed as follows:

- RailTrack-DaViT is a novel approach that applies a Vision Transformer (ViT) to railway track defect binary classification. RailTrack-DaViT is not only able to capture long-range dependencies but also effectively models both global and local information in railway track images.
- RailTrack-DaViT incorporates a customized classification head and training pipeline that can be effectively trained and tested on datasets containing limited data. These modifications enable RailTrack-DaViT to achieve greater stability and quicker adaptation to unseen images during the fine-tuning process compared to baseline methods.
- The ThaiRailTrack dataset is constructed by collecting Thai railway track images from two sources: the National Science and Technology Development Agency (NSTDA) and the Passenger Service Department (Operation) of the State Railway of Thailand.
- A comprehensive analysis of model performance is presented, including metrics such as a confusion matrix, training history visualization, and classification metrics. For extensive evaluation, the proposed model is evaluated on various datasets, including Rail, Fastener and fishplate, Multi-faults, and ThaiRailTrack datasets.

2. Method

2.1. Overview of RailTrack-DaViT Architecture

In this study, we propose a deep learning approach for defecting defects in railway tracks, called RailTrack-DaViT. The RailTrack-DaViT utilizes a pre-trained Dual Attention Vision Transformers (DaViT) [29] base model as the feature extractor backbone. DaViT is a recently proposed vision transformer variant that achieves a strong performance on ImageNet [30] classification while using fewer parameters and demonstrating improved data efficiency compared to other Vision Transformer (ViT) models [29] architectures. With spatial window and channel group attentions, DaViT efficiently captures not only local representation but also global context. To leverage the powerful representations learned by DaViT on ImageNet while reducing the risk of overfitting to our smaller railway dataset, we froze all layers of the base model initially and used it as a fixed feature extractor. The original 1000-class classification head was replaced with a custom multilayer perceptron (MLP) classification head for a binary classification task. The completed network architecture of RailTrack-DaViT is shown in Figure 1.

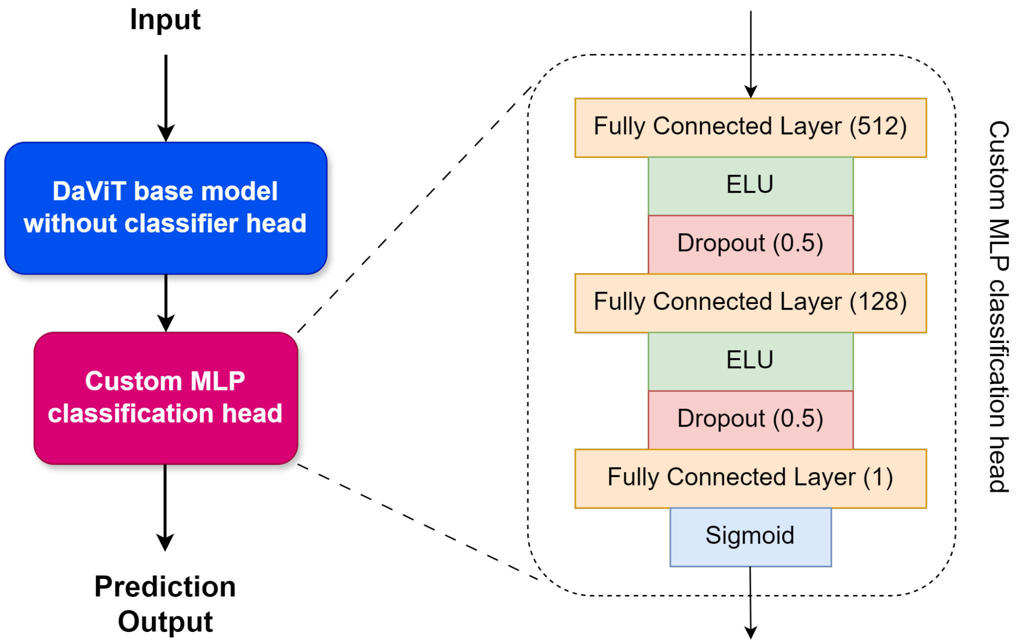

Input
DaViT base model without classifier head
Custom MLP classification head
Prediction Output
Fully Connected Layer (512)
ELU
Dropout (0.5)
Fully Connected Layer (128)
ELU
Dropout (0.5)
Fully Connected Layer (1)
Sigmoid
Custom MLP classification head

Figure 1. RailTrack-DaViT network architecture.

The MLP classification head consists of two fully connected layers with 512 and 128 units, respectively. The ELU [31] activation functions were utilized for non-linearity as shown in Equation (1).

$$f(z) = \begin{cases} z & \text{for } z \geq 0 \\ \beta \cdot (e^z - 1) & \text{for } z \leq 0 \end{cases} \tag{1}$$

where:

- z is the feature between layers used as input for the ELU activation function;
- β is a hyperparameter that controls the value to which an ELU saturates for negative inputs.

ELU has been shown to speed up learning in deep neural networks and tends to have a significantly better generalization performance compared to other activation functions. Moreover, dropout regularization ($p = 0.5$) between each layer was applied. The final output is a single sigmoid unit representing the probability of a defect being present in the input image.

2.2. *RailTrack-DaViT Model Training Process*

Figure 2 illustrates the overall process in training RailTrack-DaViT. The RailTrack-DaViT was trained in two stages. In the first stage, only the classification head was trained for 90 epochs while the RailTrack-DaViT backbone remained frozen. Binary cross-entropy loss was used as the loss function. The equation of binary cross-entropy loss can be derived as:

$$L(y, \hat{y}) = -\frac{1}{N} \sum_{i=1}^{N} [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] \tag{2}$$

where:

- $L(y, \hat{y})$ is the binary cross-entropy loss;
- N is the number of samples;
- y_i is the true label of the i-th sample (0 or 1);
- $\hat{y}_i$ is the predicted probability of the i-th sample belonging to class 1.

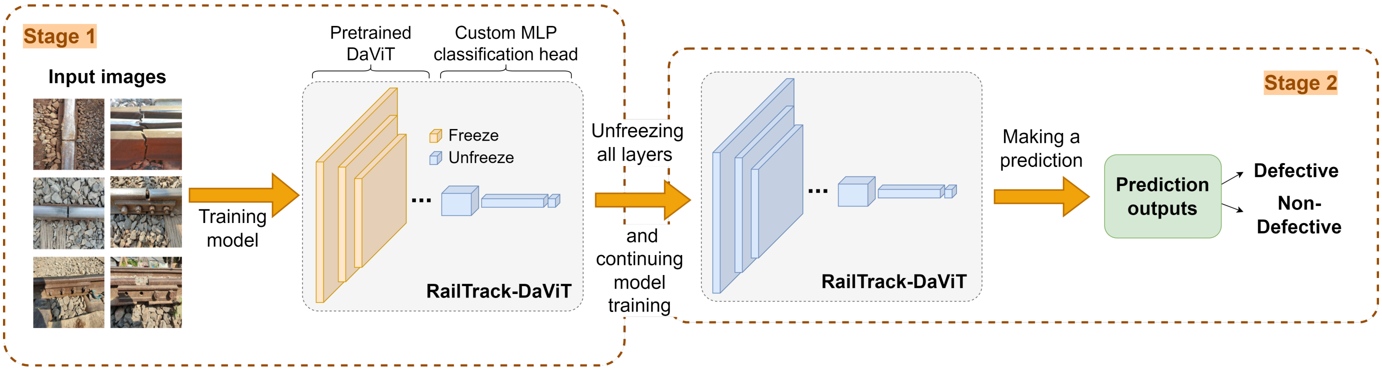


Figure 2. Overall process for training RailTrack-DaViT.

The loss function penalizes the model for making incorrect predictions. It encourages the predicted probabilities to be close to the true labels, minimizing the difference between them. The logarithmic terms ensure that the loss is high when the predicted probability is far from the true label and low when it is close.

The optimization was performed using the Adam optimizer [32] with a learning rate of 0.001. The one-cycle learning rate policy [33] was employed and linearly increased the learning rate from a low value to a maximum of 0.003 in the first 30% of iterations, then linearly decreased it back to the minimum in the remaining iterations. The one-cycle learning rate policy can be expressed as Equation (3). This learning rate schedule has been shown to speed up convergence and achieve better final accuracy compared to a fixed learning rate.

$$\alpha(t) = \begin{cases} \alpha_{\text{min}} + \frac{t}{0.3T}(\alpha_{\text{max}} - \alpha_{\text{min}}), & 0 \leq t < 0.3T \\ \alpha_{\text{max}} - \frac{t-0.3T}{0.7T}(\alpha_{\text{max}} - \alpha_{\text{min}}), & 0.3T \leq t \leq T \end{cases} \tag{3}$$

where:

- $\alpha(t)$ is the learning rate at iteration t;
- α_{min} is the minimum learning rate (initial learning rate, 1×10^{-3});
- α_{max} is the maximum learning rate (3×10^{-3});
- T is the total number of iterations (steps per epoch × number of epochs).

In the second stage, the entire RailTrack-DaViT model was unfrozen and fine-tuned end-to-end for an additional 10 epochs. The model was trained during these 10 epochs by utilizing the Adam optimizer with an initial learning rate of 0.0001 and a one-cycle learning rate policy with a maximum learning rate of 0.001. This fine-tuning allows the visual features extracted by RailTrack-DaViT to adapt to specific defect patterns present in the dataset, while the lower learning rate helps limit overfitting.

To further improve training stability, gradient clipping [34] with a maximum L2 norm regularization of 1.0 was applied to the gradients at each optimization step. Gradient clipping is a technique that limits the magnitude of gradients to prevent unstable training. It prevents the gradients from growing too large and causing training to diverge. It is typically formulated as

$$\mathbf{g}_{\text{clipped}} = \frac{\mathbf{g}}{\max\left(1, \frac{\|\mathbf{g}\|_2}{c}\right)} \tag{4}$$

where:

- $\mathbf{g}$ is the gradient vector;
- $\|\mathbf{g}\|_2$ is its L2 norm;
- c is the clipping threshold.

The clipping threshold value c determines the maximum allowed L2 norm of the gradient. In this study, we set the clipping threshold equal to 1.0. This means that if the gradients have an L2 norm larger than 1.0, they will be clipped to have an L2 norm of 1.0 while preserving direction. Gradients with an L2 norm less than or equal to 1.0 will remain unchanged.

3. Dataset

There are four datasets that are utilized to evaluate the performance of our approach in this work. The datasets consist of Rail, Fastener and fishplate, Multi-faults, and ThaiRailTrack datasets. The first three datasets (Rail, Fastener and fishplate, and Multi-faults datasets) are public datasets that can be accessed through Kaggle (https://www.kaggle.com/datasets (accessed on 19 April 2024)), which is an online community platform for data scientists and machine learning practitioners. These datasets were constructed by Eunus et al. [4]. ThaiRailTrack dataset is a private dataset that is collected from Thailand's railway organizations. Figure 3 presents the distribution of images of railway track defect detection across the four datasets used in our study. ThaiRailTrack (Before) represents the number of images before oversampling. ThaiRailTrack (After) presents the number of images after performing the oversampling technique.

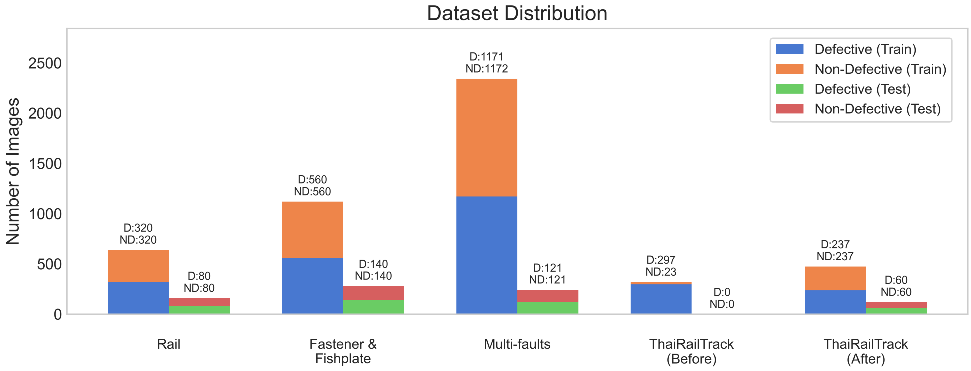


Figure 3. The number of defective and non-defective images in both the training and test sets of Rail, Fastener and fishplate, Multi-faults, and ThaiRailTrack datasets. "D" refers to the defective images category and "ND" refers to the non-defective images category.

The details of each dataset are described as follows:

3.1. Rail Dataset

The rail dataset [35] contains images of regular railway tracks and faults on railway tracks. There are 800 defective and non-defective images. The dataset was divided into 640 images for the training set and 160 images for the test set. Examples of defective images on rails are illustrated in Figure 4.

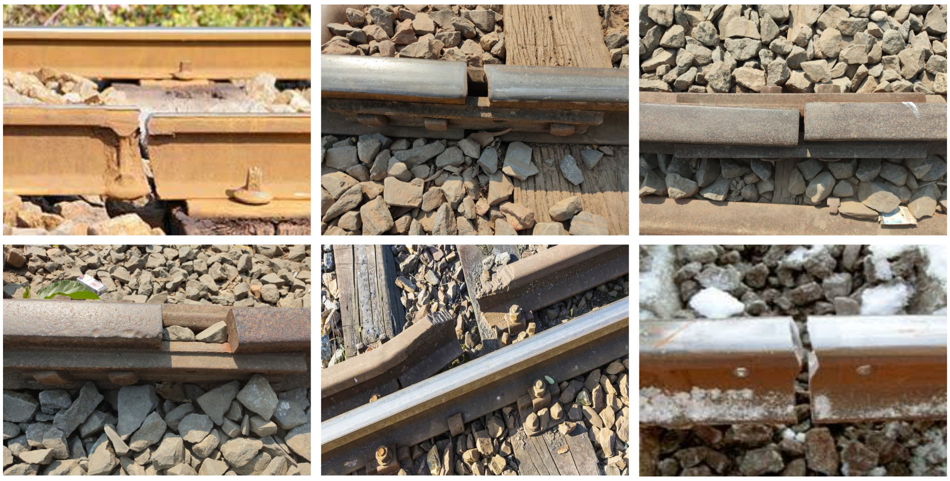


Figure 4. The sample images of railway tracks with faults (defect). The dataset consists of images of different views, including top and side views.

3.2. Fastener and Fishplate Dataset

The Fastener and fishplate dataset [36] contains images of regular railway tracks, broken clips (fasteners), and broken fishplates. Clips are a type of fastener responsible for securing the steel rail to the sleeper. Occasionally, the clip may fracture or become detached, making it incapable of serving its intended function [37]. Rail fishplates, one of the most common types of rail fastener [38], are often small copper or nickel silver plates employed to connect two rails together in a railway track. They ensure the proper alignment and continuity of the rails. In total, there were 1300 images of defective and non-defective fasteners and fishplates. The images were divided into 1120 for the training set and 280 for the test set. The samples of broken fasteners and fishplates on railway tracks are illustrated in Figure 5.

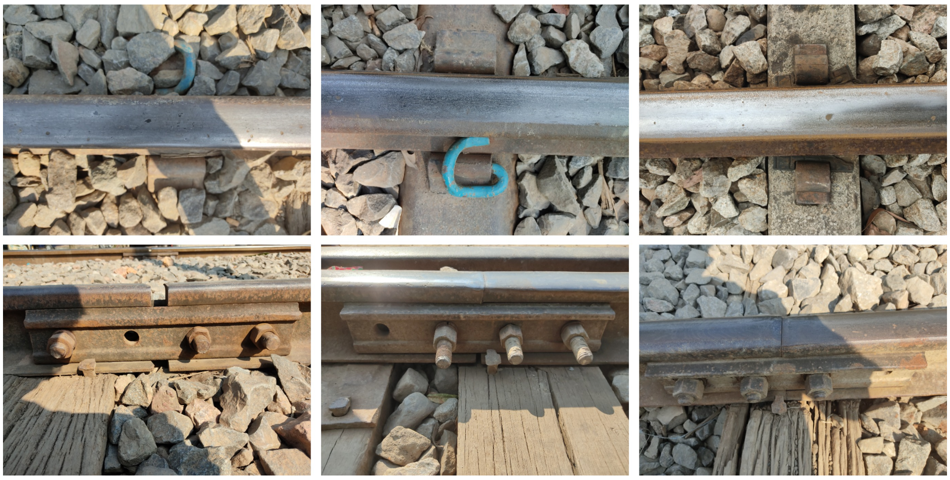


Figure 5. The sample images of broken fasteners and fishplates. The dataset consists of images from different views including top and side views. The first row presents the images of broken fasteners on railway tracks. The second row represents the images of broken fishplates on railway tracks.

3.3. Multi-Faults Dataset

The Multi-faults dataset is a combination of three datasets—Rail [35], Fastener and fishplate [36], and Railway Track fault Detection [39] datasets. There are 2584 images of both defective and non-defective railway tracks. The images were split into 2343 for the training set and 242 for the test set.

3.4. ThaiRailTrack Dataset

The ThaiRailTrack dataset consists of railway track images in Thailand, including both defective and non-defective tracks collected from two sources—the National Science and Technology Development Agency (NSTDA) and the Passenger Service Department (Operation) of the State Railway of Thailand. The original dataset had 297 images of defective railway tracks and 23 of non-defective railway tracks. To prevent a class imbalance problem when evaluating the model, we applied an oversampling technique by using random data augmentation for images of non-defective tracks. Finally, we obtained 594 images. The images were divided into 474 images for the training set and 120 images for the test set. The example images from the ThaiRailTrack dataset are illustrated in Figure 6.

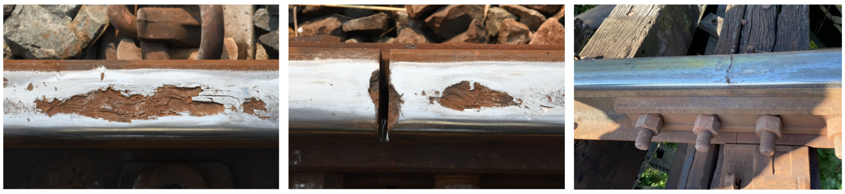


Figure 6. The top and side image views of Thai railway tracks collected from two Thailand organizations, NSTDA and the Passenger Service Department (Operation) of the State Railway of Thailand.

4. Experiment and Result Analysis

4.1. Performance Confusion Matrix

A confusion matrix is a table used to describe the performance of a classification model on a set of test data where the true values are known [40]. Figure 7 illustrates the confusion matrix for binary classification utilized in this study. The matrix consists of two axes—x-axis and y-axis—with corresponding labels. The 0 label represents positive or defective samples, which are images of broken rail tracks. The 1 label denotes negative or non-defective samples, which are images of regular rail tracks. TP, TN, FP, and FN refer to True Positive, True Negative, False Positive, and False Negative, respectively.

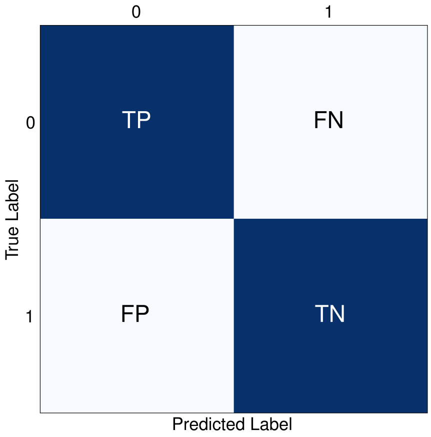


Figure 7. The confusion matrix of binary classification consisting of x-axis, y-axis, and the values.

4.2. Performance Evaluation Metrics

In this study, we utilized key performance evaluation metrics for classification models. The metrics include precision, recall, specificity, F1 score, and accuracy. These key performance evaluation metrics can be calculated with the following equations:

Precision measures the proportion of true positive predictions among all positive predictions:

$$\text{Precision} = \frac{TP}{TP + FP} \times 100\%. \tag{5}$$

Recall or sensitivity measures the proportion of actual positives that are correctly identified by the model:

$$\text{Recall} = \frac{TP}{TP + FN} \times 100\%. \tag{6}$$

Specificity measures the proportion of actual negatives that are correctly identified by the model:

$$\text{Specificity} = \frac{TN}{TN + FP} \times 100\%. \tag{7}$$

The F1 score provides a balanced measure of a model's performance, especially when the classes are imbalanced:

$$\text{F1 score} = \frac{2 \times Precision \times Recall}{Precision + Recall} \times 100\% = \frac{2 \times TP}{2 \times TP + FP + FN} \times 100\% \tag{8}$$

Accuracy measures the overall correctness of model's predictions:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \times 100\%. \tag{9}$$

4.3. Data Pre-Processing

In the beginning, we prepared four different datasets including Rail, Fastener and fishplate, Multi-faults, and ThaiRailTrack. The Rail, Fastener and fishplate, and ThaiRailTrack datasets were divided into 80 percent for the training set and 20 percent for the test set. For the Multi-faults dataset, we used 10 percent as the test set. Four types of data augmentation were randomly applied, including horizontal rotation, vertical rotation, 90 or 270-degree rotations, and brightness adjustment during model training. The images were performed randomly center-cropped to 256 × 256, resized to 224 × 224, and then randomly shuffled with a seed value equal to 42 before model training. In this study, we set the batch size to 16.

4.4. Comparison Models Overview

In this study, there are different CNN-based models as the baseline, compared with our RailTrack-DaViT for image classification. The models include Xception [41], ResNet-18 [42], ResNet-50 [42], EfficientNet-B0 [43], and EfficientNet-B7 [43]. The models were implemented using the PyTorch library in Python and executed on an NVIDIA GeForce RTX 3060 GPU. To ensure a fair comparison, we established consistent parameters for training both RailTrack-DaViT and the baseline models. The training parameters were set as follows: learning rate of 0.001, Adam optimizer, and batch size of 16.

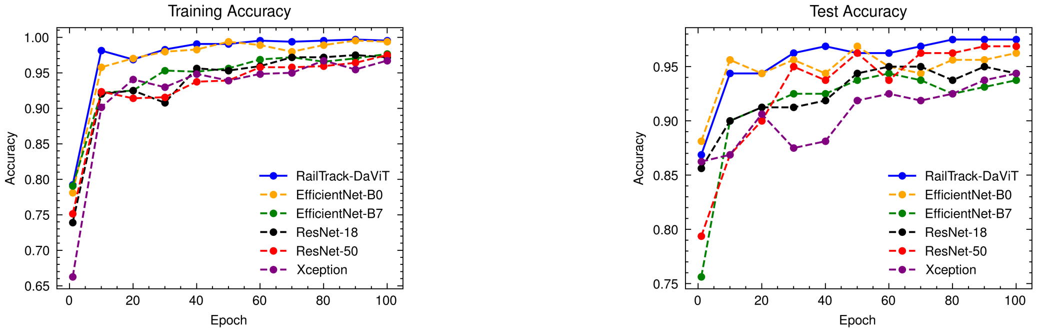
Xception [41]: A deep convolutional neural network architecture inspired by Inception [44]. It replaces the standard Inception modules with depthwise separable convolutions, which results in a more efficient use of model parameters.

ResNet or Residual Networks [42]: The key innovation of ResNet is the introduction of "identity shortcut connections" that skip one or more layers, allowing the network to learn residual functions with reference to the layer inputs. ResNet-18 and ResNet-50 are two specific architectures with 18 and 50 layers, respectively.

EfficientNet [43]: A family of models that are designed to achieve better accuracy and efficiency. The key idea is to uniformly scale the network width, depth, and resolution with a set of fixed scaling coefficients. EfficientNet-B0 is the baseline model and EfficientNet-B7 is a larger model obtained by scaling up the baseline.

4.5. The Performance Evaluation of RailTrack-DaViT and Conventional CNN-Based Models on Rail Dataset

Figure 8 illustrates the training and test accuracy of baseline and RailTrack-DaViT on the Rail dataset. The models were trained on the Rail dataset with 100 epochs. In terms of training accuracy, RailTrack-DaViT achieved the highest accuracy from the early epoch. It achieved peak accuracy compared to other models in five epochs and constantly increased the accuracy up to 100 epochs. Additionally, RailTrack-DaViT performed the highest test accuracy, which was evaluated on the test set of the rail dataset after training the model for 30 epochs. The performance curves for precision, recall, specificity, and F1 score of the baseline models and RailTrack-DaViT on the Rail dataset are presented in Appendix A.



(**a**) Training accuracy on Rail dataset (**b**) Test accuracy on Rail dataset

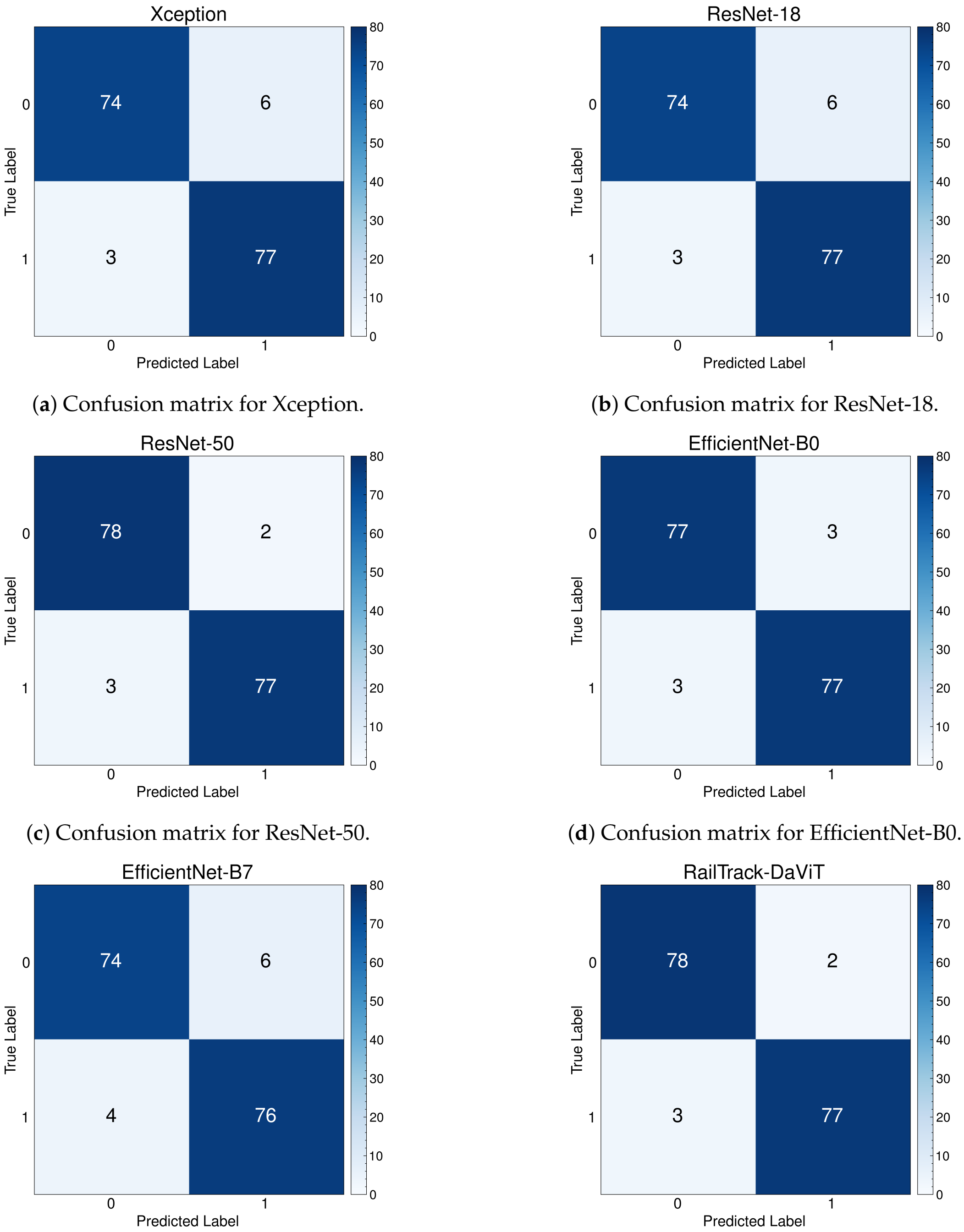
Figure 8. Comparative analysis of training and test accuracies for different classification models on the Rail dataset over 100 training epochs.

Table 1 demonstrates that RailTrack-DaViT and ResNet-50 achieve an impressive performance across all evaluation metrics tested on the Rail dataset. Specifically, for the "Defective" category, it attains a precision of 96.3%, a recall of 97.5%, a specificity of 96.3%, and an F1 score of 96.9%. The overall average accuracy of RailTrack-DaViT is 96.9%, which is among the highest in the table, indicating its exceptional performance in defect classification.

Table 1. Classification report of various models for defective and non-defective images of the Rail dataset on different metrics.

Model	**Category**	**Precision**	**Recall**	**Specificity**	**F1 Score**	**Accuracy**
Xception	Defective	96.1	92.5	96.3	94.3	94.4
	Non-defective	92.8	96.3	92.5	94.5	94.4
	Average	94.4	94.4	94.4	94.4	94.4
ResNet-18	Defective	96.1	92.5	96.3	94.3	94.4
	Non-defective	92.8	96.3	92.5	94.5	94.4
	Average	94.4	94.4	94.4	94.4	94.4
ResNet-50	Defective	96.3	97.5	96.3	96.9	96.9
	Non-defective	97.5	96.3	97.5	96.9	96.9
	Average	96.9	96.9	96.6	96.6	96.9
EfficientNet-B0	Defective	96.3	96.3	96.3	96.3	96.3
	Non-defective	96.3	96.3	96.3	96.3	96.3
	Average	96.3	96.3	96.3	96.3	96.3
EfficientNet-B7	Defective	94.9	92.5	95.0	93.7	93.7
	Non-defective	92.7	95.0	92.5	93.8	93.7
	Average	93.8	93.7	93.7	93.7	93.7
RailTrack-DaViT	Defective	96.3	97.5	96.3	96.9	96.9
	Non-defective	97.5	96.3	97.5	96.9	96.9
	Average	96.9	96.9	96.9	96.9	96.9

Figure 9 represents the confusion matrices for the baseline and our proposed model on the test set of the Rail dataset. The results show that RailTrack-DaViT and ResNet-50 can produce the highest true positives and true negatives, which equal 78 and 77, respectively, while preserving the lowest false negatives and false positives, which equal 2 and 3, respectively. This demonstrates that our proposed model is comparative to the traditional CNN-based baseline model for this Rail dataset.



(**a**) Confusion matrix for Xception.

(**b**) Confusion matrix for ResNet-18.

(**c**) Confusion matrix for ResNet-50.

(**d**) Confusion matrix for EfficientNet-B0.

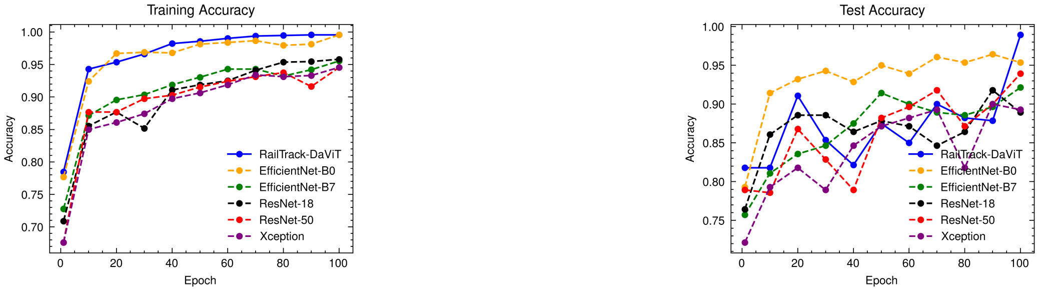
(**e**) Confusion matrix for EfficientNet-B7.

(**f**) Confusion matrix for RailTrack-DaViT.

Figure 9. Confusion matrices for baseline and our RailTrack-DaViT on the Rail dataset.

4.6. The Performance Evaluation of RailTrack-DaViT and Conventional CNN-Based Models on Fastener and Fishplate Dataset

Figure 10 presents a comprehensive analysis of the training and test accuracy of CNN-based deep learning models and our RailTrack-DaViT over 100 training epochs. The RailTrack-DaViT model consistently outperforms other models in training accuracy. Despite fluctuations in test data performance due to the optimization landscape of the Transformer-based model itself or dataset complexities arising from a combination of fastener and fishplate images, the accuracy significantly improves after unfreezing the model's weights (the last 10 epochs). This demonstrates the impact of unfrozen layer operations on model efficiency. The additional performance curves are presented in Appendix B.



(**a**) Training accuracy on Fastener and fishplate dataset.

(**b**) Test accuracy on Fastener and fishplate dataset.

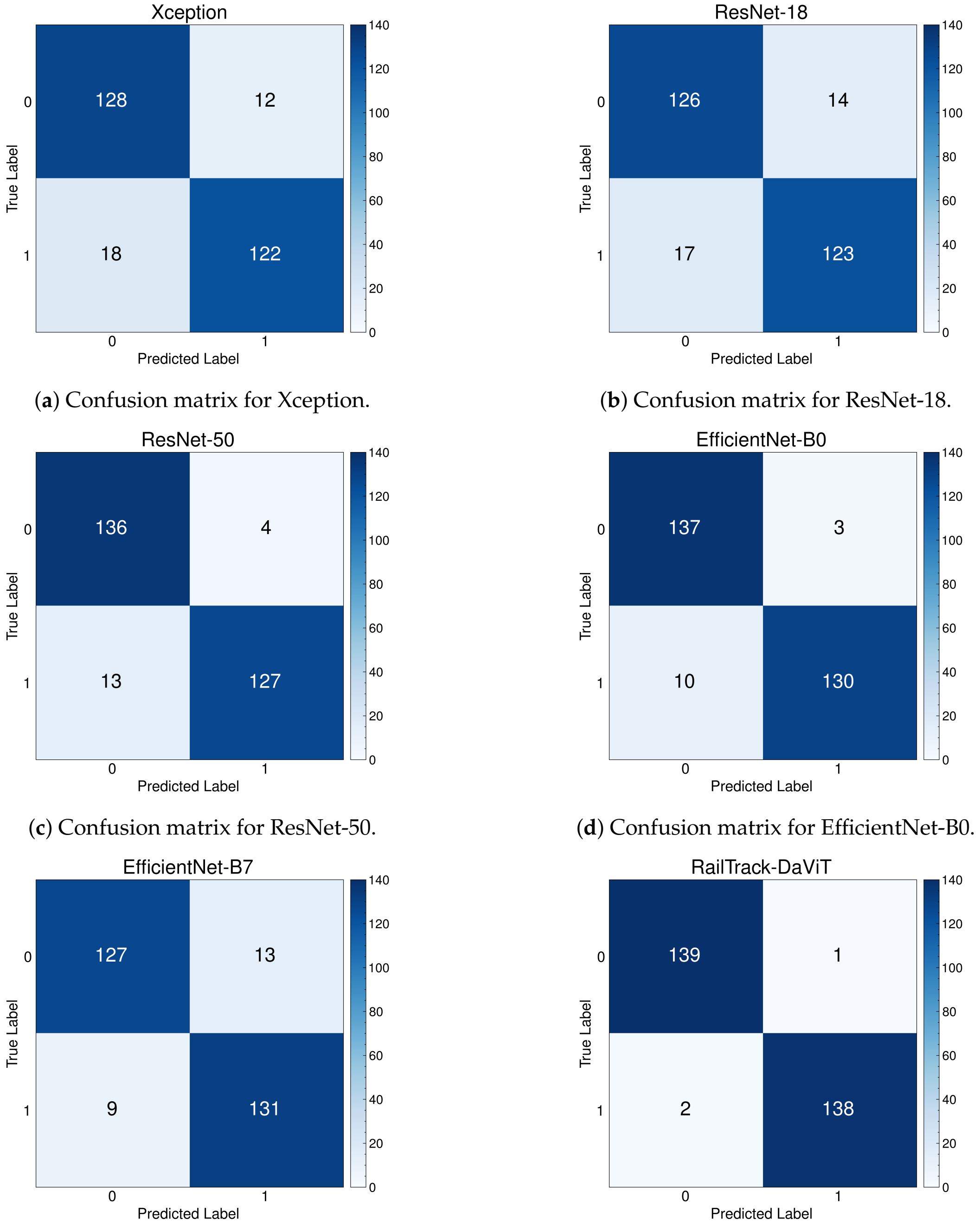
Figure 10. Comparative analysis of training and test accuracies for different classification models on Fastener and fishplate dataset over 100 training epochs.

Table 2 represents the performance metrics for CNN-based classification models and our proposed model on the Fastener and fishplate dataset. According to the table, the RailTrack-DaViT model demonstrates exceptional performance across both the defective and non-defective categories, with consistently high values for precision, recall, specificity, F1 score, and accuracy. The overall average metrics further underscore the robust and balanced performance of the model.

Table 2. Classification report of various models for defective and non-defective images of the Fastener and fishplate dataset on different metrics.

Model	Category	Precision	Recall	Specificity	F1 Score	Accuracy
Xception	Defective	87.7	91.4	87.1	89.5	89.3
	Non-defective	91.0	87.1	91.4	89.1	89.3
	Average	89.4	89.3	89.3	89.3	89.3
ResNet-18	Defective	88.1	90.0	87.9	89.0	88.9
	Non-defective	89.8	87.9	90.0	88.8	88.9
	Average	88.9	88.9	88.9	88.9	88.9
ResNet-50	Defective	96.9	90.7	97.1	93.7	93.9
	Non-defective	91.3	97.1	90.7	94.1	93.9
	Average	94.1	93.9	93.9	93.9	93.9
EfficientNet-B0	Defective	93.2	97.9	92.9	95.5	95.4
	Non-defective	97.7	92.9	97.9	95.2	95.4
	Average	95.5	95.4	95.4	95.4	95.4
EfficientNet-B7	Defective	93.4	90.7	93.6	92.0	92.1
	Non-defective	91.0	93.6	90.7	92.3	92.1
	Average	92.2	92.1	92.1	92.1	92.1
RailTrack-DaViT	Defective	98.6	99.3	98.6	98.9	98.9
	Non-defective	99.3	98.6	99.3	98.9	98.9
	Average	98.9	98.9	98.9	98.9	98.9

Figure 11 represents the confusion matrices for baseline models and our proposed model on the test set of the Fastener and fishplate dataset. Based on 280 images of the test set, the confusion matrix reveals that the RailTrack-DaViT outperforms the other models by correctly classifying between defective and non-defective images with the highest true positives and true negatives.



(**a**) Confusion matrix for Xception.

(**b**) Confusion matrix for ResNet-18.

(**c**) Confusion matrix for ResNet-50.

(**d**) Confusion matrix for EfficientNet-B0.

(**e**) Confusion matrix for EfficientNet-B7.

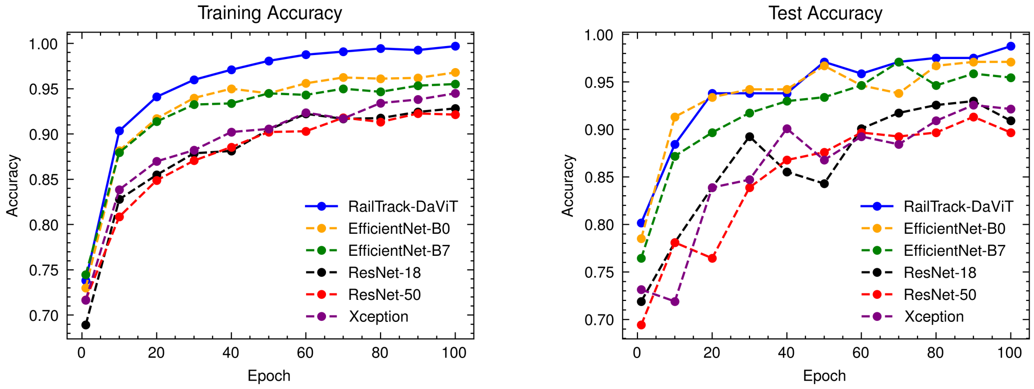
(**f**) Confusion matrix for RailTrack-DaViT.

Figure 11. Confusion matrices for baseline and our RailTrack-DaViT on the Fastener and fishplate dataset.

4.7. The Performance Evaluation of RailTrack-DaViT and Conventional CNN-Based Models on Multi-Faults Dataset

Figure 12 presents a comprehensive evaluation of the training and test accuracy for various CNN-based models and our RailTrack-DaViT model on the Multi-faults dataset. Based on the training and test accuracy curve, the RailTrack-DaViT model (represented by the blue line) exhibits a consistent and steady increase in accuracy as the number of epochs progresses. It achieves the highest training and test accuracy at the beginning of model training. The RailTrack-DaViT can maintain a competitive performance and outperform the EfficientNet-B0 model in the later stages of training according to the test accuracy.

The performance curves for precision, recall, specificity, and F1 score of the baseline models and RailTrack-DaViT on the Multi-faults dataset are presented in Appendix C.



(**a**) Training accuracy on Multi-faults dataset. (**b**) Test accuracy on Multi-faults dataset.

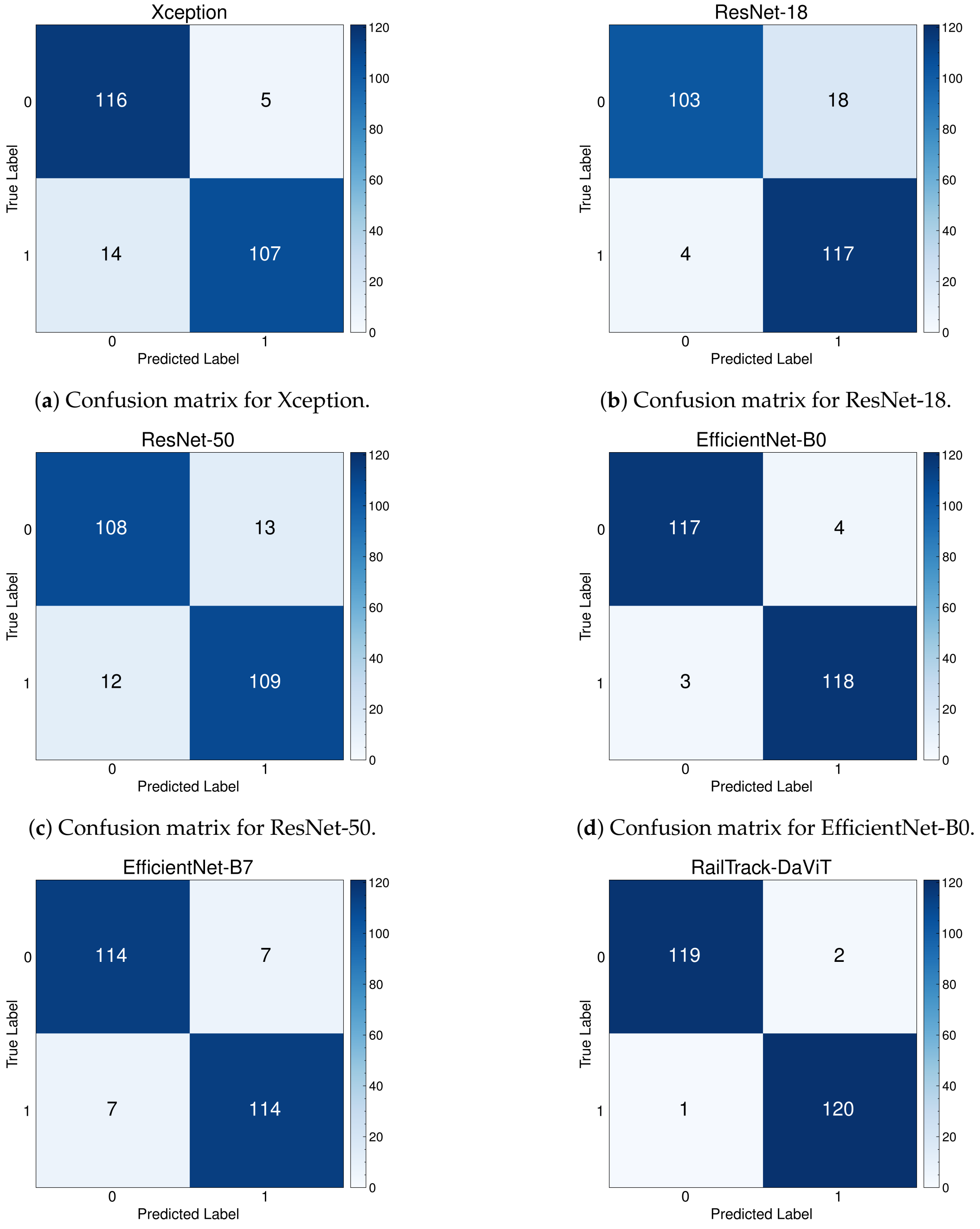
Figure 12. Comparative analysis of training and test accuracies for different classification models on the Multi-faults dataset over 100 training epochs.

Table 3 presents a comprehensive performance analysis of several models on various evaluation metrics on the Multi-faults dataset. The RailTrack-DaViT model demonstrates an excellent performance in distinguishing between defective and non-defective instances, with high scores across all evaluation metrics. This indicates its robustness and reliability in the defect defection task.

Table 3. Classification report of various models for defective and non-defective images of the Multi-faults dataset on different metrics.

Model	Category	Precision	Recall	Specificity	F1 Score	Accuracy
Xception	Defective	89.2	95.9	88.4	92.4	92.1
	Non-defective	95.5	88.4	95.9	91.8	92.1
	Average	92.4	92.1	92.1	92.1	92.1
ResNet-18	Defective	96.3	85.1	96.7	90.4	90.9
	Non-defective	86.7	96.7	85.1	91.4	90.9
	Average	91.5	90.9	90.9	90.9	90.9
ResNet-50	Defective	90.0	89.3	90.1	89.6	89.7
	Non-defective	89.3	90.1	89.3	89.7	89.7
	Average	89.7	89.7	89.7	89.7	89.7
EfficientNet-B0	Defective	97.5	96.7	97.5	97.1	97.1
	Non-defective	96.7	97.5	96.7	97.1	97.1
	Average	97.1	97.1	97.1	97.1	97.1
EfficientNet-B7	Defective	94.2	94.2	94.2	94.2	94.2
	Non-defective	94.2	94.2	94.2	94.2	94.2
	Average	94.2	94.2	94.2	94.2	94.2
RailTrack-DaViT	Defective	99.2	98.3	99.2	98.8	98.8
	Non-defective	98.4	99.2	98.3	98.8	98.8
	Average	98.8	98.8	98.8	98.8	98.8

Figure 13 presents a comprehensive evaluation of the classification performance of various models, including our RailTrack-DaViT model, through a series of confusion matrices. For the 242 images in the test set in the Multi-faults dataset, the confusion matrix demonstrates that the RailTrack-DaViT model achieves the highest number of true positives and true negatives, equaling 119 and 120, respectively.



(**a**) Confusion matrix for Xception.

(**b**) Confusion matrix for ResNet-18.

(**c**) Confusion matrix for ResNet-50.

(**d**) Confusion matrix for EfficientNet-B0.

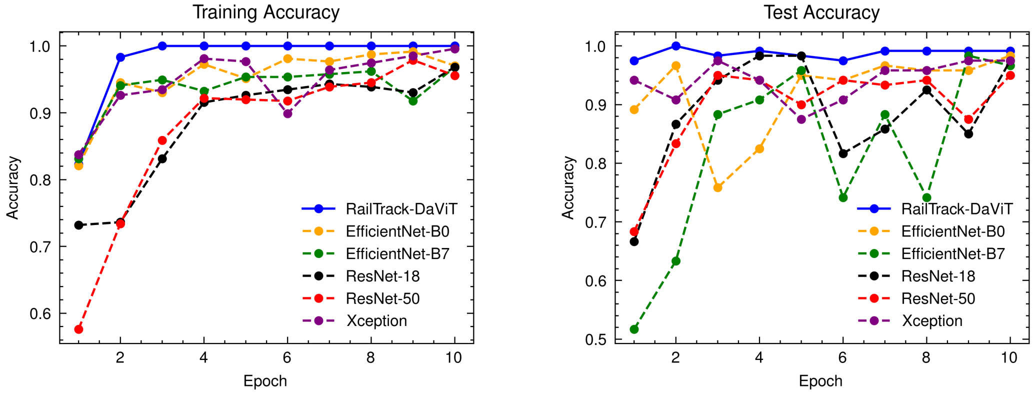
(**e**) Confusion matrix for EfficientNet-B7.

(**f**) Confusion matrix for RailTrack-DaViT.

Figure 13. Confusion matrices for baseline and our RailTrack-DaViT on the Multi-faults dataset.

4.8. The Performance Evaluation of RailTrack-DaViT and Conventional CNN-Based Models on ThaiRailTrack Dataset

Figure 14 presents a comparative analysis of the training and test accuracy of various models, including RailTrack-DaViT, over 10 epochs. It highlights the superior performance of RailTrack-DaViT in terms of both training and test accuracy, demonstrating its effectiveness in transfer learning and generalizing from the given data within only 10 epochs. Notably, the RailTrack-DaViT shows the highest test accuracy compared to baseline methods from the initial stages of fine-tuning. This suggests that the model adapts quickly to unseen images compared to other models. The performance curves for precision, recall, specificity, and F1 score of the baseline models and RailTrack-DaViT on the ThaiRailTrack dataset are presented in Appendix D.



(**a**) Training accuracy on ThaiRailTrack dataset. (**b**) Test accuracy on ThaiRailTrack dataset.

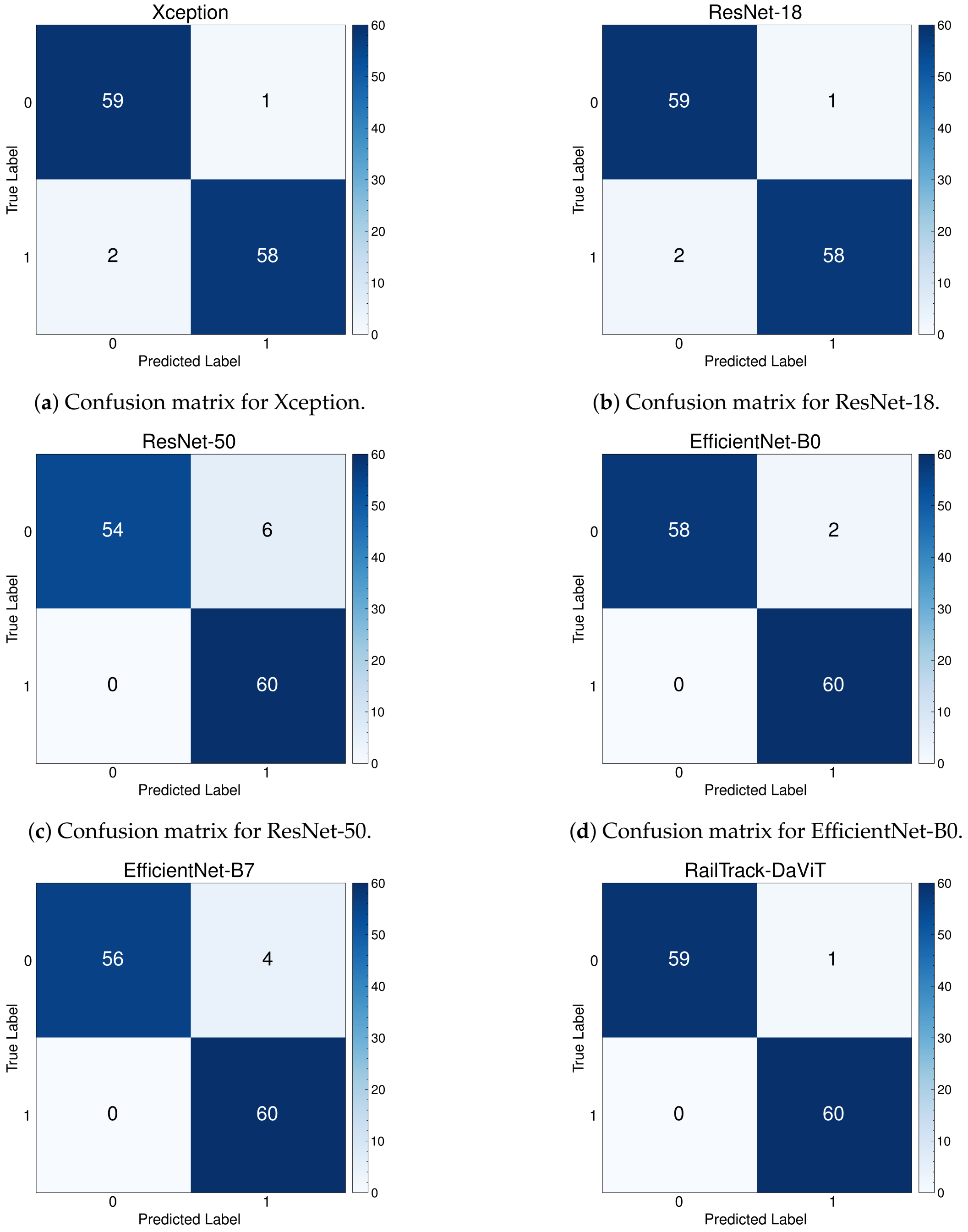
Figure 14. Comparative analysis of training and test accuracies for different classification models on ThaiRailTrack dataset over 10 training epochs.

Table 4 presents a comprehensive performance analysis of several models on the ThaiRailTrack dataset for various evaluation metrics. The RailTrack-DaViT achieves a remarkable performance across all metrics. Specifically, for the "Defective" category, it attains a perfect precision and specificity score of 100.0%, indicating no false positives and an excellent ability to identify true defective instances. Considering the average performance across categories, RailTrack-DaViT maintains its exceptional performance, achieving an average precision, recall, specificity, F1 score, and accuracy of 99.2%. This highlights the model's robustness and ability to generalize effectively across diverse instances.

Table 4. Classification report of various models for defective and non-defective images of ThaiRailTrack dataset on different metrics.

Model	Category	Precision	Recall	Specificity	F1 Score	Accuracy
Xception	Defective	96.7	98.3	96.7	97.5	97.5
	Non-defective	98.3	96.7	98.3	97.5	97.5
	Average	97.5	97.5	97.5	97.5	97.5
ResNet-18	Defective	96.7	98.3	96.7	97.5	97.5
	Non-defective	98.3	96.7	98.3	97.5	97.5
	Average	97.5	97.5	97.5	97.5	97.5
ResNet-50	Defective	100.0	90.0	100.0	94.7	95.0
	Non-defective	90.9	100.0	90.0	95.2	95.0
	Average	95.5	95.0	95.0	95.0	95.0
EfficientNet-B0	Defective	100.0	96.7	100.0	98.3	98.3
	Non-defective	96.8	100.0	96.7	98.4	98.3
	Average	98.4	98.3	98.3	98.3	98.3
EfficientNet-B7	Defective	100.0	93.3	100.0	96.6	96.7
	Non-defective	93.8	100.0	93.3	96.8	96.7
	Average	96.9	96.7	96.7	96.7	96.7
RailTrack-DaViT	Defective	100.0	98.3	100.0	99.2	99.2
	Non-defective	98.4	100.0	98.3	99.2	99.2
	Average	99.2	99.2	99.2	99.2	99.2

Figure 15 presents a comprehensive evaluation of the classification performance of baseline and our proposed models through a series of confusion matrices. For the 120 images in the test set of the ThaiRailTrack dataset, the confusion matrix demonstrates that the RailTrack-DaViT model outperforms other models in terms of achieving the highest number of both true positives and true negatives. This indicates that our proposed model performs best on both defective and non-defective images.



(**a**) Confusion matrix for Xception.

(**b**) Confusion matrix for ResNet-18.

(**c**) Confusion matrix for ResNet-50.

(**d**) Confusion matrix for EfficientNet-B0.

(**e**) Confusion matrix for EfficientNet-B7.

(**f**) Confusion matrix for RailTrack-DaViT.

Figure 15. Confusion matrices for baseline and our RailTrack-DaViT on the ThaiRailTrack dataset.

5. Ablation Study

In this section, we analyze the impact of each optimizer, five-fold cross validation, and alternative network designs on the Multi-faults dataset. We focus on this dataset because it encompasses both the Rail dataset and the Fastener and fishplate dataset and contains the highest number of image samples.

5.1. Network Design

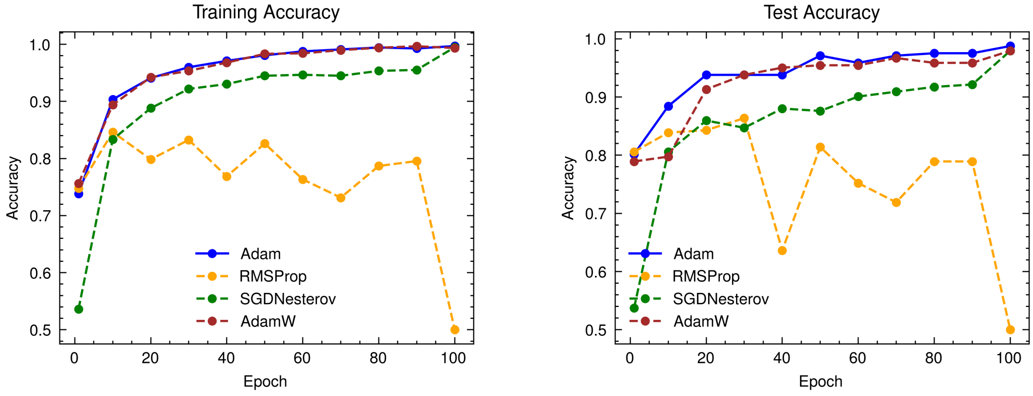
Table 5 presents the classification performance for each component and operation in the RailTrack-DaViT model, evaluated based on the training and test accuracies over 100 epochs. It demonstrates that RailTrack-DaViT without pre-trained weights achieves the lowest accuracies, indicating that the model fails to converge without pre-trained weights. Moreover, the results show that RailTrack-DaViT—with either an unfrozen layer operation or a custom MLP, or incorporating both—outperforms the traditional DaViT model.

Table 5. Accuracy of different network architectures and operations. The highest training and test accuracies are highlighted in bold.

Model	Pre-Training	Unfreeze	Custom Head	Training Accuracy	Test Accuracy
DaViT	✓	✗	✗	97.0	91.7
RailTrack-DaViT	✗	✗	✓	49.2	50.0
	✗	✓	✓	50.7	50.0
	✓	✗	✓	99.4	97.5
	✓	✓	✗	99.5	97.5
	✓	✓	✓	**99.7**	**98.8**

5.2. Optimizer

We selected the optimal optimizer based on its performance scores for both training and test sets. The RMSProp, SGD with Nesterov momentum, AdamW, and Adam optimizers were evaluated for model performance. Figure 16 illustrates the training and test accuracies of RailTrack-DaViT utilizing different optimizers. The Adam optimizer demonstrates the best performance and greater stability compared to the others on both training and test sets.



(**a**) Training accuracy for different optimizers.

(**b**) Test accuracy for different optimizers.

Figure 16. Comparative analysis of training and test accuracies for different optimizers on the Multi-faults dataset over 100 training epochs.

5.3. Five-Fold Cross Validation

In addition to standard model training, we conducted five-fold cross-validation to monitor the average performance accuracy metrics on both training and test sets. In this study, the training set was randomly split into k folds, where $k = 5$. The model was then trained on $k - 1$ folds, while one fold was left for model validation.

Table 6 summarizes the comparison of training and test accuracies between standard model training and five-fold cross-validation. Our RailTrack-DaViT model achieves the highest scores in both training and test accuracies while maintaining model training stability, as evidenced by the lowest standard deviation values (0.1 for training accuracy and 0.4 for test accuracy) among baseline models.

Table 6. Accuracy of various models using standard training and five-fold cross-validation on both training and test sets. "Standard" refers to standard model training, and "Five-fold" refers to five-fold cross-validation. The highest training and test accuracies are highlighted in bold.

Model	Technique	Training Accuracy	Test Accuracy
Xception	Standard Five-fold	94.5 93.5 ± 1.0	92.1 90.5 ± 1.3
ResNet-18	Standard Five-fold	92.8 93.3 ± 0.3	90.9 91.7 ± 1.8
ResNet-50	Standard Five-fold	92.1 91.2 ± 1.1	89.7 88.1 ± 1.0
EfficientNet-B0	Standard Five-fold	96.8 96.4 ± 0.3	97.1 96.2 ± 0.5
EfficientNet-B7	Standard Five-fold	95.5 92.0 ± 0.2	94.2 92.4 ± 1.7
RailTrack-DaViT	Standard Five-fold	**99.7** **99.7 ± 0.1**	**98.8** **98.6 ± 0.4**

6. Discussion and Conclusions

In this paper, we proposed RailTrack-DaViT, a novel vision transformer-based deep learning approach for detecting defects from railway track images. By employing a Dual Attention Vision Transformer (DaViT) architecture, RailTrack-DaViT effectively captures both global and local information, addressing the limitations of traditional CNN-based models in capturing long-range dependencies on railway track datasets. The customized classification head and training pipeline enable the model to adapt pre-trained DaViT features for binary defect identification.

Extensive evaluations on various datasets, including Rail, Fastener and fishplate, Multi-faults, and ThaiRailTrack datasets, demonstrate the superior performance of RailTrack-DaViT compared to conventional CNN-based models used in this paper including Xception, ResNet-18, ResNet-50, EfficientNet-B0, and EfficientNet-B7. Overall, the proposed approach consistently achieves high precision, recall, specifically, F1 score, and accuracy across all datasets, highlighting its robustness and generalization capabilities. Moreover, when fine-tuning the model on the ThaiRailTrack dataset, RailTrack-DaViT demonstrates its capability for handing unseen data through its ability to quickly adapt to novel images. This rapid adaptation can significantly reduce time consumption in practical applications.

The ability of RailTrack-DaViT to capture long-range dependencies and effectively model both global and local information makes it a promising solution for automated railway track defect detection. By automating the inspection process, RailTrack-DaViT has the potential to significantly reduce the time and cost associated with manual inspections while improving the accuracy and reliability of defect detection.

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

Funding: This research was partially funded by China-Laos-Thailand Education Digitization International Joint Research and Development Center of Yunnan Province (Project Number: 202203AP140006).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Rail dataset can be found at—https://www.kaggle.com/datasets/salmaneunus/railway-track-fault-detection (accessed on 19 April 2024). Fastener and fishplate dataset can be found at—https://www.kaggle.com/datasets/ashikadnan/railway-track-fault-detection-dat aset2fastener/data (accessed on 19 April 2024). ThaiRailTrack dataset is not publicly available due to privacy restrictions.

Acknowledgments: This research was fully supported by CMU Junior Research Fellowship Program. The authors would like to acknowledge Kosit Wongpinkaew and Duangrada Yutthakamthon from the National Science and Technology Development Agency (NSTDA), Thailand, as well as Kitchawat Khuanphoonphol and Passenger Service Department(Operation) of State Railway of Thailand for their support of this research through the provision of the ThaiRailTrack dataset.

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

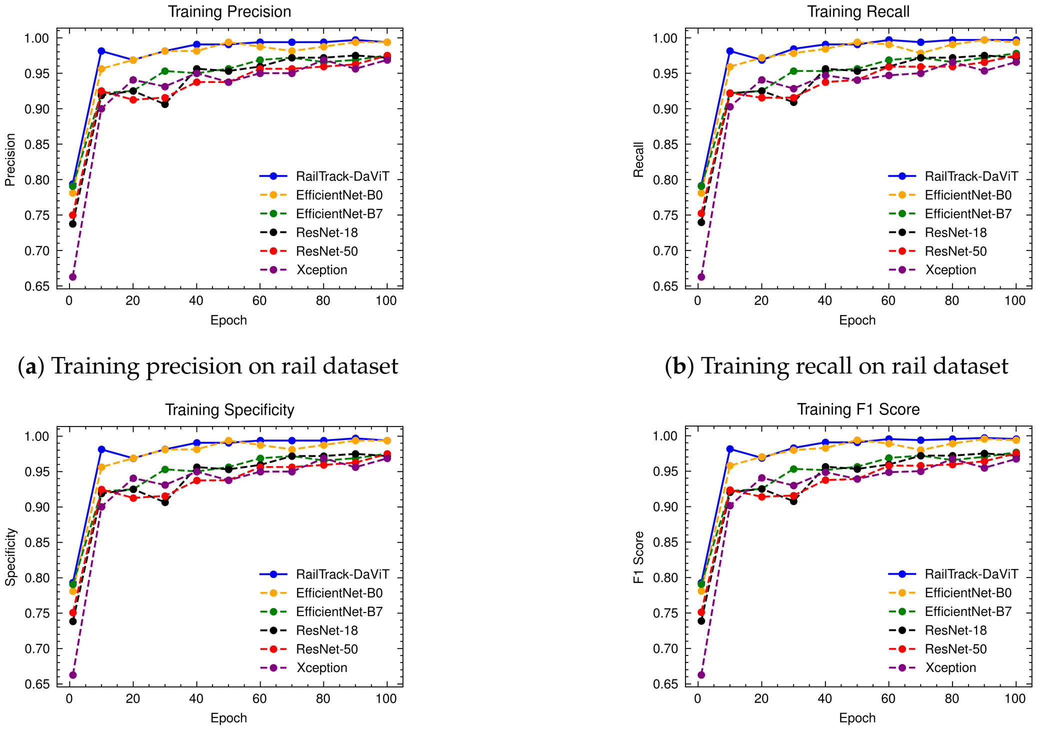
Abbreviations

The following abbreviations are used in this manuscript:

CNN	Convolutional Neural Network
ViT	Vision Transformer
DaViT	Dual Attention Vision Transformers

Appendix A. The Curves of Precision, Recall, Specificity, and F1 Score for Baselines and RailTrack-DaViT on Rail Dataset

Figure A1 illustrates training curves for precision, recall, specificity, and F1 score for our RailTrack-DaViT model and baseline models on rail dataset. The results demonstrate that RailTrack-DaViT and EfficientNet-B0 perform comparably; however, RailTrack-DaViT achieves the highest scores across all four metrics from the early training epochs onward.



(**a**) Training precision on rail dataset

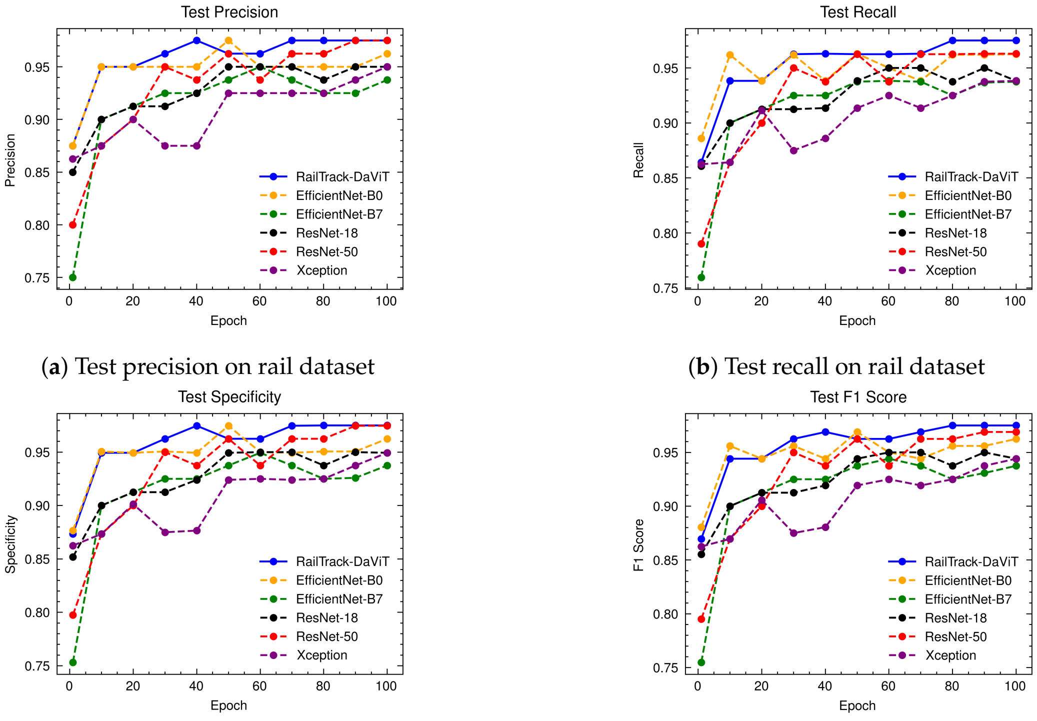
(**b**) Training recall on rail dataset

(**c**) Training specificity on rail dataset

(**d**) Training F1 score on rail dataset

Figure A1. Comparative analysis of training precision, recall, specificity, and F1 score for different classification models on rail dataset over 100 training epochs.

Figure A2 illustrates test curves for precision, recall, specificity, and F1 score for our RailTrack-DaViT model and baseline models on rail dataset. The results demonstrate that RailTrack-DaViT, EfficientNet-B0, and ResNet-50 perform comparably; however, RailTrack-DaViT achieves the highest scores across all four metrics from the early training epochs onward.



(**a**) Test precision on rail dataset

(**b**) Test recall on rail dataset

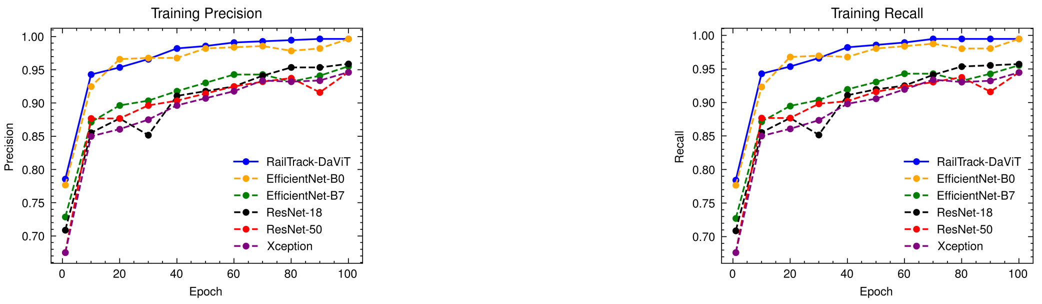
(**c**) Test specificity on rail dataset

(**d**) Test F1 score on rail dataset

Figure A2. Comparative analysis of test precision, recall, specificity, and F1 score for different classification models on rail dataset over 100 training epochs.

Appendix B. The Curves of Precision, Recall, Specificity, and F1 Score for Baselines and RailTrack-DaViT on Fastener and Fishplate Dataset

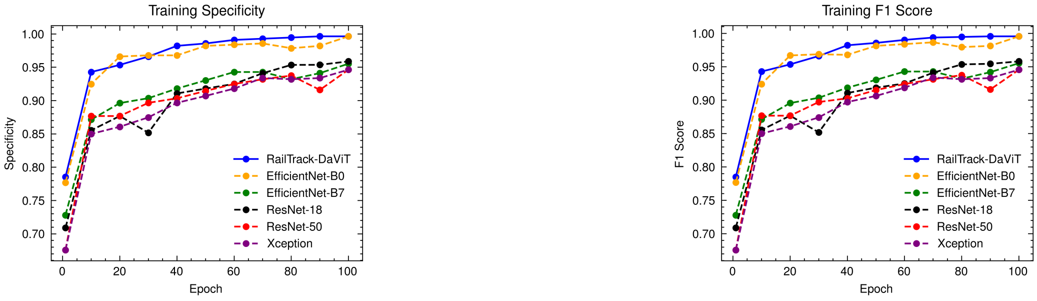
Figure A3 illustrates training curves for precision, recall, specificity, and F1 score for our RailTrack-DaViT model and baseline models on fastener and fishplate dataset. The results demonstrate that RailTrack-DaViT and EfficientNet-B0 perform comparably; however, RailTrack-DaViT achieves the highest scores across all four metrics from the early training epochs onward.



(**a**) Training precision on fastener and fishplate dataset

(**b**) Training recall on fastener and fishplate dataset

Figure A3. *Cont.*

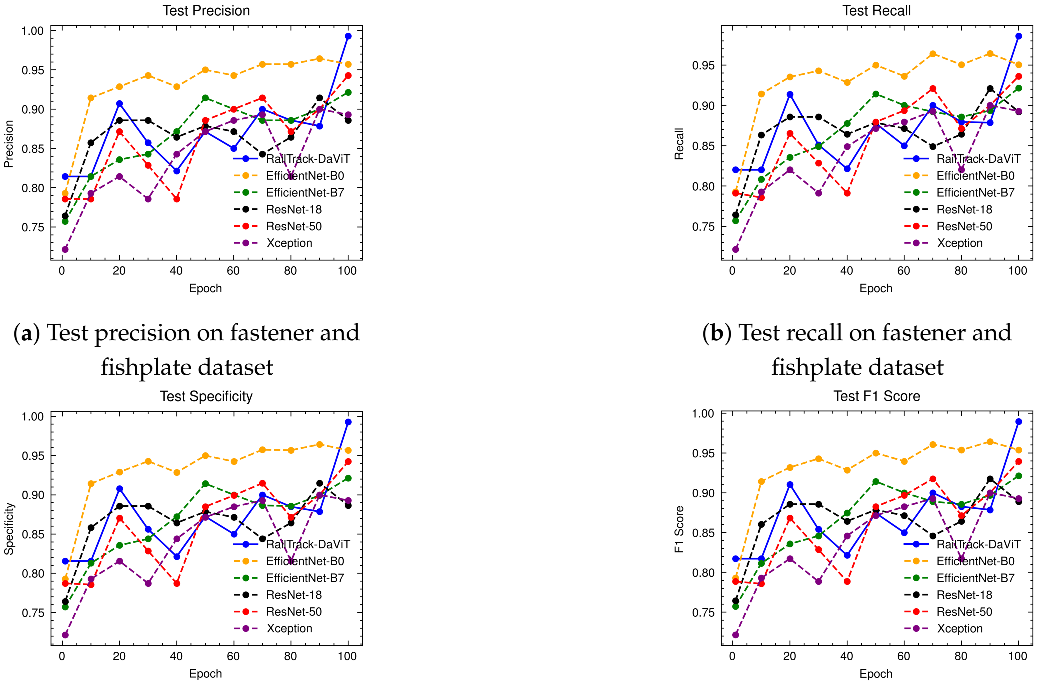


(**c**) Training specificity on fastener and fishplate dataset

(**d**) Training F1 score on fastener and fishplate dataset

Figure A3. Comparative analysis of training precision, recall, specificity, and F1 score for different classification models on fastener and fishplate dataset over 100 training epochs.

Figure A4 illustrates test curves for precision, recall, specificity, and F1 score of our RailTrack-DaViT model and baseline models on fastener and fishplate dataset. The results demonstrate that EfficientNet-B0 outperforms all models from the initial training state until the 90th epoch. However, starting from epoch 90, when all layers in RailTrack-DaViT were unfrozen, the performance curves of RailTrack-DaViT for all four metrics improved significantly.



(**a**) Test precision on fastener and fishplate dataset

(**b**) Test recall on fastener and fishplate dataset

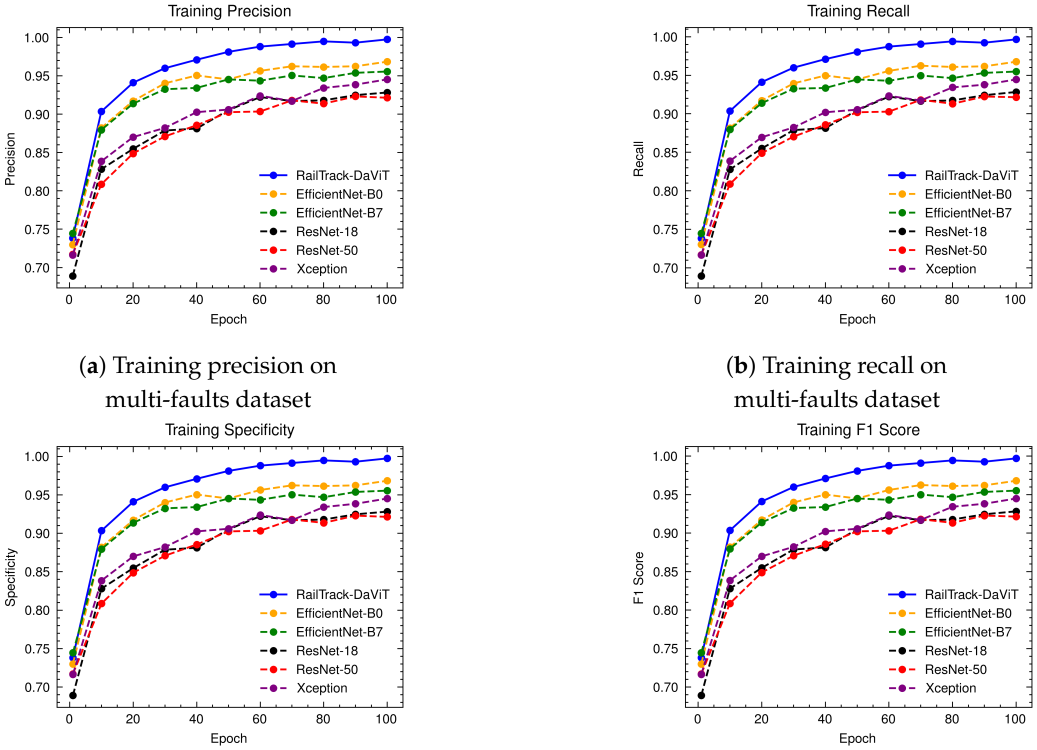
(**c**) Test specificity on fastener and fishplate dataset

(**d**) Test F1 score on fastener and fishplate dataset

Figure A4. Comparative analysis of test precision, recall, specificity, and F1 score for different classification models on fastener and fishplate dataset over 100 training epochs.

Appendix C. The Curves of Precision, Recall, Specificity, and F1 Score for Baselines and RailTrack-DaViT on Multi-Faults Dataset

Figure A5 illustrates training curves for precision, recall, specificity, and F1 score for our RailTrack-DaViT model and baseline models on multi-faults dataset. The results demonstrate that RailTrack-DaViT achieves the highest scores across all four metrics from the early training epochs onward. EfficientNet-B0 and EfficientNet-B7 rank second and third, respectively, in terms of performance.



(**a**) Training precision on multi-faults dataset

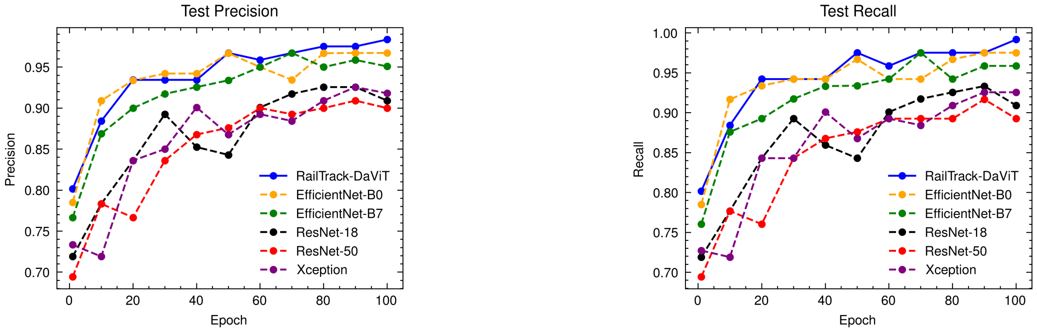
(**b**) Training recall on multi-faults dataset

(**c**) Training specificity on multi-faults dataset

(**d**) Training F1 score on multi-faults dataset

Figure A5. Comparative analysis of training precision, recall, specificity, and F1 score for different classification models on multi-faults dataset over 100 training epochs.

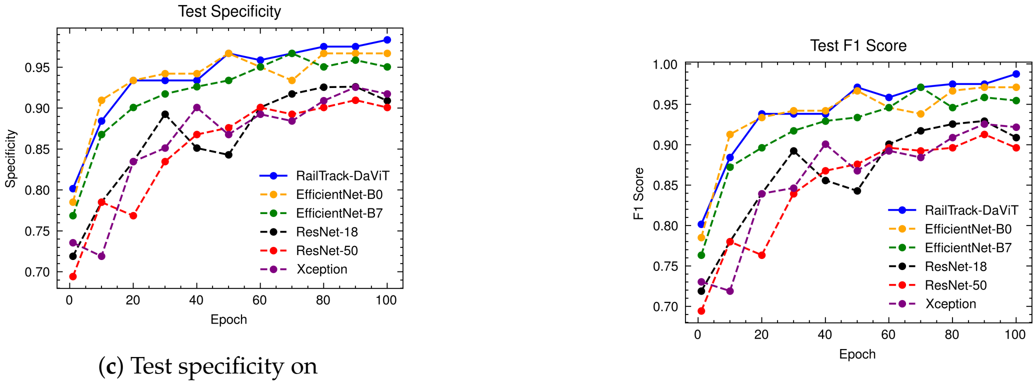
Figure A6 illustrates test curves for precision, recall, specificity, and F1 score of our RailTrack-DaViT model and baseline models on the multi-faults dataset. The results demonstrate that RailTrack-DaViT and EfficientNet-B0 perform comparably the middle period of model training; however, RailTrack-DaViT achieves the highest scores across all four metrics in the later epochs of training.



(**a**) Test precision on multi-faults dataset

(**b**) Test recall on multi-faults dataset

Figure A6. *Cont*.



(**c**) Test specificity on multi-faults dataset

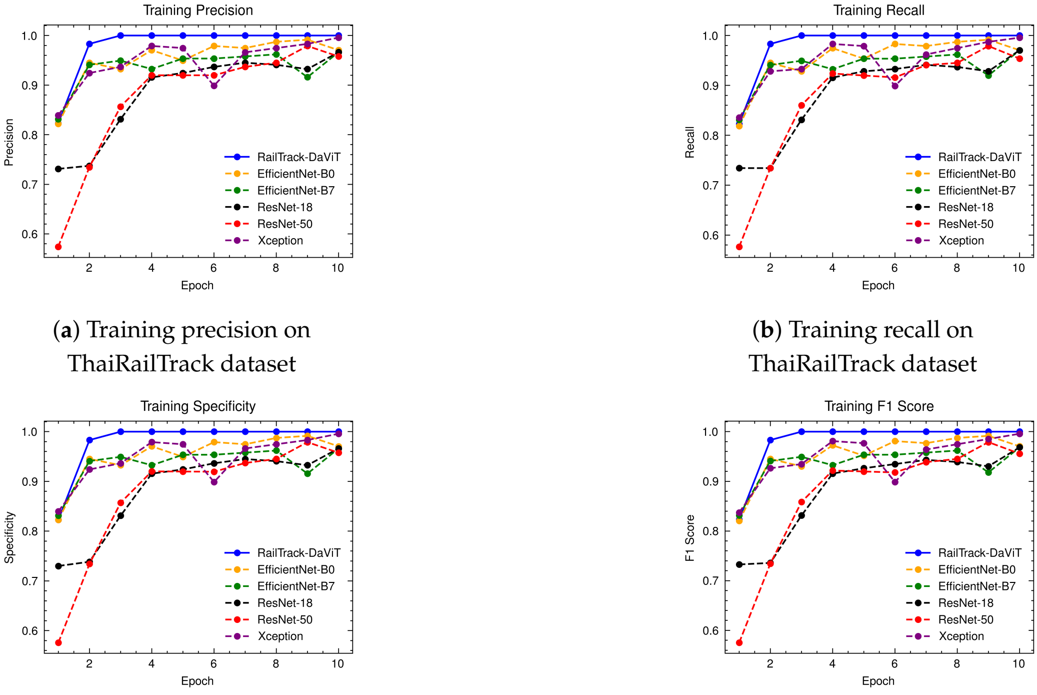
(**d**) Test F1 score on multi-faults dataset

Figure A6. Comparative analysis of test precision, recall, specificity, and F1 score for different classification models on multi-faults dataset over 100 training epochs.

Appendix D. The Curves of Precision, Recall, Specificity, and F1 Score for Baselines and RailTrack-DaViT on ThaiRailTrack Dataset

Figure A7 illustrates training curves for precision, recall, specificity, and F1 score for our RailTrack-DaViT model and baseline models on ThaiRailTrack dataset. The results demonstrates that RailTrack-DaViT achieves the highest scores across all four metrics from the early training epochs onward.

Figure A8 illustrates test curves for precision, recall, specificity, and F1 score of our RailTrack-DaViT model and baseline models on ThaiRailTrack dataset. The results demonstrates that RailTrack-DaViT achieves the highest scores across all four metrics from the beginning of model training and shows slightly improvement subsequent epochs. In contrast to other CNN-based models, whose performance fluctuates during fine-tuning process on this dataset, RailTrack-DaViT exhibits more stable performance.



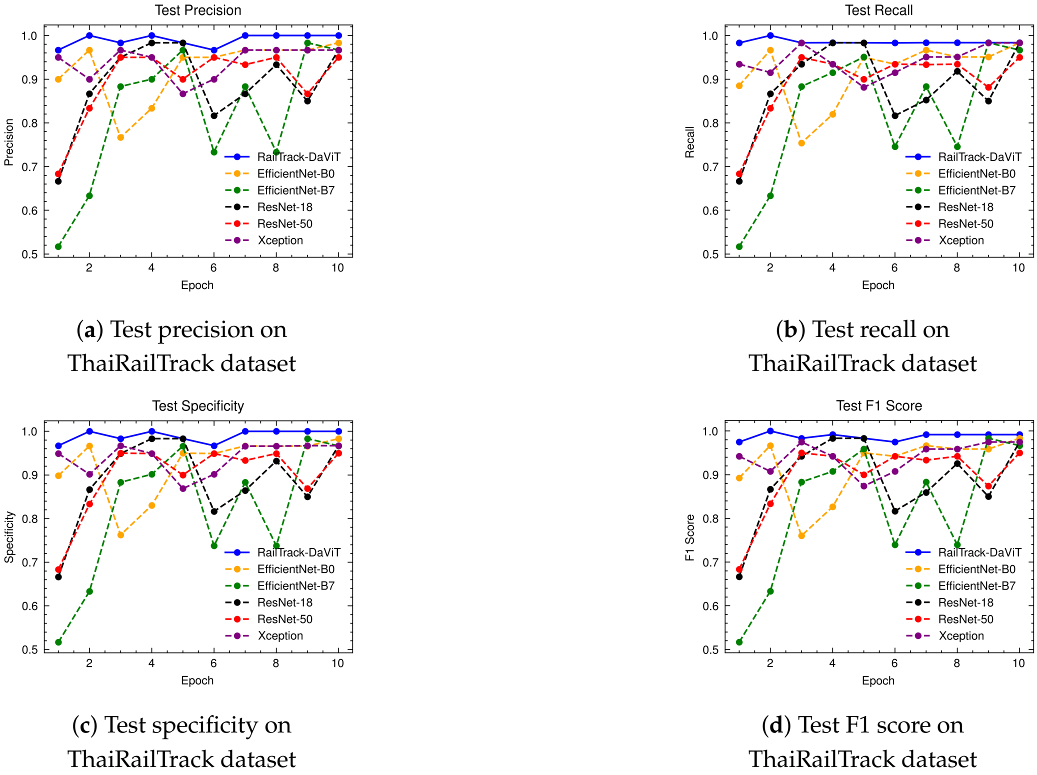
(**a**) Training precision on ThaiRailTrack dataset

(**b**) Training recall on ThaiRailTrack dataset

(**c**) Training specificity on ThaiRailTrack dataset

(**d**) Training F1 score on ThaiRailTrack dataset

Figure A7. Comparative analysis of training precision, recall, specificity, and F1 score for different classification models on ThaiRailTrack dataset over 100 training epochs.



(**a**) Test precision on ThaiRailTrack dataset

(**b**) Test recall on ThaiRailTrack dataset

(**c**) Test specificity on ThaiRailTrack dataset

(**d**) Test F1 score on ThaiRailTrack dataset

Figure A8. Comparative analysis of test precision, recall, specificity, and F1 score for different classification models on ThaiRailTrack dataset over 100 training epochs.

References

1. Li, H.; Strauss, J.; Shunxiang, H.; Lui, L. Do high-speed railways lead to urban economic growth in China? A panel data study of China's cities. *Q. Rev. Econ. Financ.* **2018**, *69*, 70–89. [CrossRef]
2. Hassan, S.T.; Zhu, B.; Lee, C.C.; Ahmad, P.; Sadiq, M. Asymmetric impacts of public service "transportation" on the environmental pollution in China. *Environ. Impact Assess. Rev.* **2021**, *91*, 106660. [CrossRef]
3. Köllő, S.A.; Faur, A.; Köllő, G.; Puskás, A. Environmental impacts of railway transportation systems. *Earth Sci. Hum. Constr.* **2021**, *1*, 1–5. [CrossRef]
4. Eunus, S.I.; Hossain, S.; Ridwan, A.; Adnan, A.; Islam, M.S.; Karim, D.Z.; Alam, G.R.; Uddin, J. ECARRNet: An Efficient LSTM-Based Ensembled Deep Neural Network Architecture for Railway Fault Detection. *AI* **2024**, *5*, 482–503. [CrossRef]
5. Sen, P.K.; Bhiwapurkar, M.; Harsha, S.P. Analysis of Causes of Rail Derailment in India and Corrective Measures. In *Reliability and Risk Assessment in Engineering: Proceedings of INCRS 2018*; Springer: Singapore, 2020; pp. 305–313.
6. Zheng, D.; Li, L.; Zheng, S.; Chai, X.; Zhao, S.; Tong, Q.; Wang, J.; Guo, L. A defect detection method for rail surface and fasteners based on deep convolutional neural network. *Comput. Intell. Neurosci.* **2021**, *2021*, 2565500. [CrossRef] [PubMed]
7. Yang, T.L.; Altabey, W.A. Modern methods of railway track safety inspection. *Int. J. Sustain. Mater. Struct. Syst.* **2018**, *3*, 99–122. [CrossRef]
8. Loveday, P.W. Guided wave inspection and monitoring of railway track. *J. Nondestruct. Eval.* **2012**, *31*, 303–309. [CrossRef]
9. Hashmi, M.S.A.; Ibrahim, M.; Bajwa, I.S.; Siddiqui, H.U.R.; Rustam, F.; Lee, E.; Ashraf, I. Railway track inspection using deep learning based on audio to spectrogram conversion: An on-the-fly approach. *Sensors* **2022**, *22*, 1983. [CrossRef] [PubMed]
10. De Ruvo, P.; Ruvo, G.D.; Distante, A.; Nitti, M.; Stella, E.; Marino, F. A visual inspection system for rail detection & tracking in real time railway maintenance. *Open Cybern. Syst. J.* **2008**, *2*, 57–67.
11. Ritika, S.; Rao, D. Data augmentation of railway images for track inspection. *arXiv* **2018**, arXiv:1802.01286.
12. Gasparini, R.; Pini, S.; Borghi, G.; Scaglione, G.; Calderara, S.; Fedeli, E.; Cucchiara, R. Anomaly detection for vision-based railway inspection. In Proceedings of the Dependable Computing-EDCC 2020 Workshops: AI4RAILS, DREAMS, DSOGRI, SERENE 2020, Munich, Germany, 7 September 2020; Proceedings 16; Springer: Berlin/Heidelberg, Germany, 2020; pp. 56–67.
13. Gasparini, R.; D'Eusanio, A.; Borghi, G.; Pini, S.; Scaglione, G.; Calderara, S.; Fedeli, E.; Cucchiara, R. Anomaly detection, localization and classification for railway inspection. In Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2021; pp. 3419–3426.
14. Gibert, X.; Patel, V.M.; Chellappa, R. Robust fastener detection for autonomous visual railway track inspection. In Proceedings of the 2015 IEEE winter conference on applications of computer vision, Waikoloa, HI, USA, 5–9 January 2015; pp. 694–701.

15. Gibert, X.; Patel, V.M.; Chellappa, R. Deep multitask learning for railway track inspection. *IEEE Trans. Intell. Transp. Syst.* **2016**, *18*, 153–164. [CrossRef]
16. Liu, J.; Huang, Y.; Wang, S.; Zhao, X.; Zou, Q.; Zhang, X. Rail fastener defect inspection method for multi railways based on machine vision. *Railw. Sci.* **2022**, *1*, 210–223. [CrossRef]
17. Baek, S.; Park, J.; Vepakomma, P.; Raskar, R.; Bennis, M.; Kim, S.L. Visual transformer meets cutmix for improved accuracy, communication efficiency, and data privacy in split learning. *arXiv* **2022**, arXiv:2207.00234.
18. Zhu, J.; Peng, B.; Li, W.; Shen, H.; Huang, Q.; Lei, J. Modeling Long-range Dependencies and Epipolar Geometry for Multi-view Stereo. *ACM Trans. Multimed. Comput. Commun. Appl.* **2023**, *19*, 1–17. [CrossRef]
19. Pang, D.; Wang, H.; Ma, J.; Liang, D. DCTN: A dense parallel network combining CNN and transformer for identifying plant disease in field. *Soft Comput.* **2023**, *27*, 15549–15561. [CrossRef]
20. Khan, A.; Rauf, Z.; Khan, A.R.; Rathore, S.; Khan, S.H.; Shah, N.S.; Farooq, U.; Asif, H.; Asif, A.; Zahoora, U.; et al. A Recent Survey of Vision Transformers for Medical Image Segmentation. *arXiv* **2023**, arXiv:2312.00634.
21. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In Proceedings of the International Conference on Learning Representations, Virtual Event, Austria, 3–7 May 2021.
22. Rosso, M.M.; Marasco, G.; Aiello, S.; Aloisio, A.; Chiaia, B.; Marano, G.C. Convolutional networks and transformers for intelligent road tunnel investigations. *Comput. Struct.* **2023**, *275*, 106918. [CrossRef]
23. Wang, R.; Shao, Y.; Li, Q.; Li, L.; Li, J.; Hao, H. A novel transformer-based semantic segmentation framework for structural condition assessment. *Struct. Health Monit.* **2024**, *23*, 1170–1183. [CrossRef]
24. Zhang, H.; Goodfellow, I.; Metaxas, D.; Odena, A. Self-attention generative adversarial networks. In Proceedings of the International Conference on Machine Learning (PMLR), Long Beach, CA, USA, 9–15 June 2019; pp. 7354–7363.
25. Feng, X.; Wang, T.; Yang, X.; Zhang, M.; Guo, W.; Wang, W. ConvWin-UNet: UNet-like hierarchical vision Transformer combined with convolution for medical image segmentation. *Math. Biosci. Eng.* **2023**, *20*, 128–144. [CrossRef]
26. Xu, T.; Jiang, T.; Xing, H.; Li, X. Multi-Resolution Diffeomorphic Image Registration with Convolutional Vision Transformer Network. In Proceedings of the 2023 4th International Conference on Computing, Networks and Internet of Things, Xiamen, China, 26–28 May 2023; pp. 388–397.
27. Rouabhi, S.; Azerine, A.; Tlemsani, R.; Essaid, M.; Idoumghar, L. Conv-ViT fusion for improved handwritten Arabic character classification. *Signal Image Video Process.* **2024**, *18*, 355–372. [CrossRef]
28. Aslan, M.F. Comparison of vision transformers and convolutional neural networks for skin disease classification. In Proceedings of the International Conference on New Trends in Applied Sciences, Online, 1–3 December 2023; Volume 1, pp. 31–39.
29. Ding, M.; Xiao, B.; Codella, N.; Luo, P.; Wang, J.; Yuan, L. Davit: Dual attention vision transformers. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 74–92.
30. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE conference on computer vision and pattern recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
31. Clevert, D.; Unterthiner, T.; Hochreiter, S. Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). In Proceedings of the International Conference on Learning Representations (ICLR), San Juan, Puerto Rico, 2–4 May 2016.
32. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the International Conference on Learning Representations (ICLR), San Diega, CA, USA, 7–9 May 2015.
33. Smith, L.N.; Topin, N. Super-convergence: Very fast training of neural networks using large learning rates. In Proceedings of the Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications (SPIE 2019), San Diego, CA, USA, 2–7 February 2019; Volume 11006, pp. 369–386.
34. Zhang, J.; He, T.; Sra, S.; Jadbabaie, A. Why Gradient Clipping Accelerates Training: A Theoretical Justification for Adaptivity. In Proceedings of the International Conference on Learning Representations (ICLR), Addis Ababa, Ethiopia, 26–30 April 2020.
35. Adnan, A. Railway Track Fault Detection: Dataset1 (Rail). 2021. Available online: https://www.kaggle.com/datasets/salmaneunus/railway-track-fault-detection (accessed on 19 April 2024).
36. Adnan, A. Railway Track Fault Detection: Dataset2 (Fastener). 2021. Available online: https://www.kaggle.com/datasets/ashikadnan/railway-track-fault-detection-dataset2fastener/data (accessed on 19 April 2024).
37. Minguell, M.G.; Pandit, R. TrackSafe: A comparative study of data-driven techniques for automated railway track fault detection using image datasets. *Eng. Appl. Artif. Intell.* **2023**, *125*, 106622. [CrossRef]
38. Nayan, M.M.R.; Al Sufi, S.; Abedin, A.K.; Ahamed, R.; Hossain, M.F. An IoT based real-time railway fishplate monitoring system for early warning. In Proceedings of the 2020 11th International Conference on Electrical and Computer Engineering (ICECE), Dhaka, Bangladesh, 17–19 December 2020; pp. 310–313.
39. Eunus, S.I. Railway Track Fault Detection. 2021. Available online: https://www.kaggle.com/datasets/salmaneunus/railway-track-fault-detection (accessed on 19 April 2024).
40. Amin, M.F. Confusion Matrix in Binary Classification Problems: A Step-by-Step Tutorial. *J. Eng. Res.* **2022**, *6*, 1.
41. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 26 July 2017; pp. 1251–1258.

42. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 30 June 2016; pp. 770–778.
43. Tan, M.; Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In Proceedings of the International Conference on Machine Learning (PMLR 2019), Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114.
44. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 30 June 2016; pp. 2818–2826.

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.