Journal FORTEI-JEERI, Vol. 6, No. 2, November 2025: 85-94
https://jeeri.fortei.org/index.php/fortei-jeeri

Comparative Analysis of Lightweight CNN Architectures for Railway Track Fault Detection

Avip Zain Haq[1*], Cries Avian[2], Darma Arif Wicaksono[3], Rizki Mendung Ariefianto[4], Mahdin Rohmatillah[5]

Department of Electrical Engineering, Faculty of Engineering, Universitas Jember, Jember, Indonesia [1]
Department of Electrical Engineering, Faculty of Engineering, Universitas Brawijaya, Jember, Indonesia [2,5]
Department of Automation Technology Engineering, Politeknik Negeri Madiun, Madiun, Indonesia [3]
Department of Electrical Engineering, Faculty of Engineering, Universiti Malaya, Kuala Lumpur, Malaysia [4]

*Corresponding Author Email: avipzainhaq@gmail.com

Abstract -- The application of artificial intelligence is widely used in various industrial sectors, including the transportation sector, one of which is the use of Image Processing to detect damage to railway lines. Railway conventional inspections involve visually examining and measuring railway infrastructure to identify potential problems. These inspections are an important aspect of ensuring the safety and efficiency of the railway network. In some places, railway track inspections still use conventional methods with electricity and vision. The use of artificial intelligence is expected to minimize errors, increase efficiency, reduce time and costs in train damage inspections. This research aims to find the best architecture and its development is expected to be used Accelerate the process of locating damage so that railroads can be repaired immediately. In this study, we evaluate the CNN model with the lightweight model to classify the image of the condition of the train track. Several types of lightweight models chosen are EfficientNetB0, EfficientNetB3, MobileNetV2, NasNetMobile. From the results of the evaluation carried out, it was found that EfficientNetB0 was 0.875, EfficientNetB3 was 0.958, MobileNetV2 was 0.917, and NasNetMobile was 0.8333.

Keywords:
Computer Vision
Lightweight
CNN
Artificial Intelligent
Railway Fault Detection
Transfer Learning

Article History:
Received: May 22, 2025
Revised: November 27, 2025
Accepted: December 1, 2025
Published: December 1, 2025

DOI: 10.46962/forteijeeri.v6i2.35

I. Introduction

Image-based railway inspection system (IRIS) is widely used lately to infect damage to railroads in general, the use of these railroad images will be processed by computer vision and also image processing which can detect damage to fixed railway tracks on the surface of the train connection errors, rust, and other damage. Previously In railway inspection, conventional methods have long been used to detect damage and ensure operational safety, namely manual visual inspection, which is the most basic method, where officers directly observe the physical condition of the rail to detect damage such as cracks, corrosion, or deformation. Although simple, this method is highly dependent on the expertise and experience of the inspector and can be affected by environmental conditions such as lighting and weather. In addition, manual inspection takes a long time and can endanger the safety of officers, especially on high-traffic lines.

Conventional railway track inspections utilize various non-destructive testing (NDT) methods to ensure track integrity and safety. Ultrasonic Testing (UT) employs high-frequency sound waves to detect internal defects like cracks in the rail head or weld joints, requiring clean surfaces and skilled operators for accurate results. Magnetic Particle Testing (MPT) identifies surface and near-surface discontinuities in ferromagnetic materials by magnetizing the rail and applying magnetic particles, which accumulate around defects due to disruptions in the magnetic field. Eddy Current Testing (ECT) uses electromagnetic induction to detect surface and near-surface defects in conductive materials, measuring disturbances in eddy currents as changes in impedance. Track Geometry Inspection involves measuring physical parameters such as alignment, curvature, and elevation using specialized inspection vehicles to identify irregularities that could compromise ride comfort and safety. While these conventional methods are effective, they often require significant manual effort, specialized equipment, and can be time-consuming. Advancements in technology, such as the application of Lightweight Convolutional Neural Networks (LCNNs), offer the potential for more efficient, automated, and real-time railway track inspections, particularly when integrated with edge computing devices like drones.

CNN utilizes the convolution process by moving a convolution kernel (filter) of a certain size to an image, the computer gets new representative information from the result of multiplying the part of the image with the filter used. Traditional CNNs, though powerful, come with high computational costs due to Deep architecture, large datasets and High energy usage. Multiple layers with millions of parameters increase memory and processing requirements. Training these models requires vast amounts of data and computational resources. Running such models on smaller devices drains batteries quickly and can cause overheating. The lightweight model on CNN is a type of CNN whose light weight makes the model smaller in size but if we minimize the parameters in Traditional CNN then we also achieve it from Traditional CNN. Recently came across this article explaining how Lightweight CNN (LCNN) performance is similar to CNN but with a 50% reduction in model parameters.

The function of the lightweight model is to speed up the inference process. The new AI/ ML-based techniques, such as computer vision and image recognition can be applied to inspect defects in railways for safety and maintenance, and it is called an image-based railway inspection system (IRIS). Here we will list some significant applications of machine vision (object recognition) in railway transport:

Kim and Cohn (2004) set up a camera in front of a locomotive to investigate level-crossing traffic accidents. They developed a computer vision system that automatically detects possible after-accident scenes by detecting the shape of the vehicles passing in front of the train. Jinbeum Jang and colleague (2019) present a novel railway inspection system using facility detection based on deep convolutional neural network and computer vision-based image comparison approach. The proposed system aims to automatically detect wears and cracks by comparing a pair of corresponding image sets acquired at different times. For visual inspection Rubinsztejn and Chen (2011) used cameras to acquire real images. In order to achieve the automatic detection of parts of interest, missing elements, or defects, they processed the captured images with pattern recognition algorithms. Deutschl et al. (2004) used convolution filters and morphological image analysis to detect rail surface defects.

Lightweight CNN (LCNN) models have been widely used in various applications to overcome the limitations of computing resources, especially on edge devices such as drones, surveillance cameras, and other IoT devices. Some applications of LCNN include real-time object detection on mobile devices: Models such as MobileNet-SSD can be run on smartphones to identify objects in live video streams, enabling applications such as augmented reality and smart photography. Edge-based AI for IoT devices using LCNN enables IoT devices to process data locally, such as in smart home automation and industrial monitoring, without relying on cloud connectivity. Health diagnostics at the Edge using AI-powered wearable devices can monitor vital signs or detect abnormalities such as arrhythmia in real-time, providing immediate feedback to users and healthcare providers. In the context of railway inspections, the application of LCNN to edge devices such as drones is particularly relevant. Drones equipped with cameras and LCNN models can perform automated, real-time railway inspections, even in hard-to-reach areas, without disrupting railway operations. This approach enables early detection of rail defects, so that repairs can be carried out promptly to prevent accidents.

In this study, it is necessary to use the lightweight model as a model that can be used as a reference in the evaluation process, especially for the purpose of direct model inference for the user. This study will evaluate the CNN model. Among them are EfficientNetB0, EfficientNetB3, MobileNetV2, and NasNetMobile. This study uses an open dataset on the types of errors found in rail tracks.

II. METHODS

A. Dataset

The railroad image that we use comes from the Dataset at kaggle in the form of three types of image data for test, train, and validation. Rail images have each type of data category, there are two data classes, namely defective and non-defective image data. However, in this study, we focused exclusively on the train and test datasets to streamline the experimentation and evaluation process. The images are categorized into two distinct classes: defective and non-defective railway track conditions. The non-defective class consists of railway track images that represent normal, undamaged conditions, with no apparent visual anomalies. These images serve as a baseline to help the model learn what a healthy rail looks like.

On the other hand, the defective class includes images that depict various types of damage or irregularities on the railway tracks. These defects include but are not limited to: Rail expansion or deformation (often caused by excessive heat), Loose or missing track fasteners (which may compromise rail stability), Cracks or surface wear along the rail head, Misalignment or displacement of the rail. These defective conditions are critical to detect in real-world railway inspections, as they can pose serious safety risks if left unaddressed.

Figure 1. Dataset of Railway

In this study, we utilized railway images categorized into two classes: defective and non-defective. The dataset originally consisted of three subsets: training, validation, and testing. The dataset employed in this study consists of a total of 384 rail surface images, distributed across training, validation, and testing subsets. The training set comprises 300 images, with an equal class distribution of 150 defective and 150 non-defective samples. For validation, 62 images are used, including 31 defective and 31 non-defective samples, ensuring balanced representation across both categories. The test set contains 22 images, equally divided into 11 defective and 11 non-defective cases. This balanced distribution of both classes ensured that the CNN models could learn discriminative features effectively.

B. CNN Architecture

Deep Learning algorithms, such as Convolutional Neural Networks (ConvNet/CNNs), are able to distinguish one aspect of an image from another based on weights and biases that are learned. Compared to other classification algorithms, ConvNet requires less pre-processing. Contrary to primitive methods, which require hand-engineering of filters, ConvNets are capable of learning these filters/characteristics through training. A ConvNet is structured similarly to how neurons in the Human Brain connect and is inspired by the Visual Cortex. The receptive Field refers to the region of the visual field in which neurons respond to stimuli. The entire visual field is covered by such fields. ConvNets are mathematical constructs consisting of three types of layers: convolution, pooling, and fully connected. It consists of a convolution layer for extracting features, a pooling layer for pooling those features, then a fully connected layer for mapping those features into final outputs, such as classification. Among the many mathematical operations incorporated into CNNs is the convolution layer, which is a linear operation specialized for convolution.

Some of the architecture has proven to be effective in classifying various applications concerning the application of CNN to image recognition. For example, VGG16, ResNet, and InceptionV3 are the top architecture that users widely use. However, the following architecture has problems with huge parameters created. It brought a drawback to the massive architecture calculation and needed a specific computer specification. Therefore, various researchers have already created some architecture known as lightweight models. This model is designed to handle a solution to reduce parameters and calculations to achieve a better speed-accuracy trade-off and make deploying deep neural networks on small devices feasible. Therefore, several models were used for the CNN lightweight model used in this study.

These models include EfficientNetB0, EfficientNetB3, MobileNetV2, and NasNetMobile. Many researchers chose this model because of its ability to extract image features with good performance, so this model is often chosen to solve several problems. We use the model to use a transfer learning technique where the models are loaded using a pre-trained layer. Then, we freeze the model and add a fully connected layer to the model's output. This fully connected layer has a function to learn the global feature and do the classification process on features extracted by a pre-trained model used. This technique is considered more effective than having to do training from scratch and proven to increase accuracy performance. For more details related to the fully connected layer that we designed, it can be seen in Figure 2.

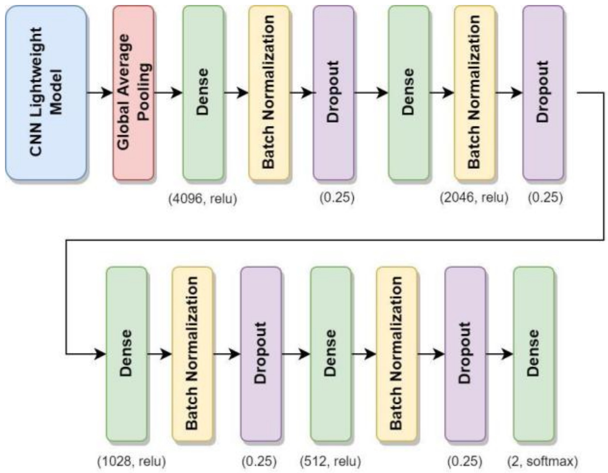

Figure 2. Fully Connected Layer Design Used

Concerning parameter number assembled due to combination model between the pre-trained model and fully connected layer, shown in Table I. The total parameter indicates the total parameter created and the trainable parameter indicates the entire parameter that the user can train.

TABLE I MODEL PARAMETERS CREATED

Model	**Total Parameter**	**Trainable Parameter**
EfficientNetB0	20 M	16 M
EfficientNetB3	28 M	17 M
MobileNetV2	18 M	16 M
NASNetMobile	19 M	15 M
Model	**Total Parameter**	**Trainable Parameter**

C. Evaluation Scheme

In evaluating the selected CNN model, the dataset will be divided using the train split validation technique, that is by dividing the dataset with a ratio of 70:30. This technique is commonly used in evaluating CNN models. However, due to limited data, we also applied the augmentation process supposed to generate a good generalization to this case. As it is known, the convolutional neural network (CNN) can perform amazing things when there is enough data available. Choosing the appropriate amount of training data for all the features that need to be trained is a challenging task. The network may overfit the training data if the user does not have enough. Various sizes, poses, zooms, lighting, noise, and lighting effects are present in realistic images. Using the data Augmentation method, the network becomes robust to these factors. Adding artificially derived data to existing training data is known as data augmentation. A variety of techniques may be used, including resizing, flipping, rotating, cropping, padding, etc. By addressing issues like overfitting and data scarcity, the model becomes more robust and performs better. For large models, data augmentation can help add enough data to alter the original image. For the augmentation parameter used, this work applied some of the techniques as shown in Table II.

To enhance the generalization capability of the CNN model and address potential overfitting due to limited dataset size, various image augmentation techniques were applied. First, a rotation range of 20 degrees was used, allowing the model to learn rotational invariance by simulating slight tilts in the image. In addition, a zoom range of 0.10 enabled the model to recognize features at varying scales, effectively handling distance-related variations. The brightness range was adjusted between 0.6 and 1.4, which allowed the model to adapt to different lighting conditions commonly encountered in outdoor rail inspections. Furthermore, a channel shift of 0.7 introduced variation in the image's color channels, promoting robustness against color distortion. To improve spatial invariance, width and height shifts of 0.15 were implemented, mimicking horizontal and vertical displacements of the railway in the frame. The inclusion of a shear range of 0.15 further distorted the image geometry slightly, encouraging the model to generalize better to perspective variations. Finally, horizontal and vertical flipping were both enabled, doubling the training data by mirroring the images and helping the model learn symmetrical patterns. Together, these augmentation strategies significantly increased the diversity of training samples, strengthening the CNN’s ability to perform reliable classification under various real-world conditions.

Another essential key in training CNN model is to set a hyperparameter for the start general. Another essential key to training the CNN model is to set a hyperparameter for starting generalizing model. Therefore, for the loss function, categorial cross-entropy was chosen. Then for the optimizer, stochastic gradient descent is used with a learning rate of 0.001, a momentum of 0.95, and a batch size is 8. In the training process, we use a python environment and personal computers by utilizing a GPU with the NVIDIA GeForce RTX 2080 Ti brand to speed up the computing process, 128GB RAM, and an intel Xeon CPU E5-2630 V4 @2.2GHz for the processor device. The parameter for the evaluation process is to use classification accuracy. Accurate classification results determine a ratio of the number of correct predictions to the total number of input samples.

TABLE II IMAGE AUGMENTATION TECHNIQUE USED

Parameter	Value
Rotation Range	20
Zoom Range	0.10
Brightness Range	0.6, 1.4
Channel Shift	0.7
Width Shift	0.15
Height Shift	0.15
Shear Range	0.15
Horizontal Flip	True
Vertical Flip	True

D. Accuracy Equations

The ROC curve is another parameter that is used to evaluate the following model. For classification problems, ROC curves are used to measure performance at various threshold settings. A ROC curve is a probability curve. It indicates how well the model can distinguish between classes. An increase in the ROC indicates that the model is better at predicting 0 classes as 0s and 1 classes as 1. Analogously, the higher the ROC, the better the model is at distinguishing whether the railway has a fault problem.

E. Accuracy Equations

The ROC curve shows the performance of a binary classifier with different decision thresholds. It plots the True Positive rate (TPR) against the False Positive rate (FPR) which,

$$TPR = \frac{TP}{TP + PN} \qquad (1)$$

$$FPR = \frac{FP}{FP + PN} \qquad (2)$$

where, *TP*, *FN*, *FP*, and *TN* are true positives, false negatives, false positives, and true negatives, respectively. The ROC curve plots TPR against FPR for different classification thresholds. A perfect model will have a point in the top-left corner (TPR = 1, FPR = 0), while a random guess model plots along the diagonal line (TPR = FPR). To understand which models that have good performance, we are using Area Under Curve (AUC):

$$AUC = \int TPR\ (x)dx \qquad (3)$$

The ROC AUC score is the area under the ROC curve. It sums up how well a model can produce relative scores to discriminate between positive or negative instances across all classification thresholds. The ROC AUC score ranges from 0 to 1, where 0.5 indicates random guessing, and 1 indicates perfect performance.

III. METHODS

All chosen lightweight models are evaluated by using train split validation. It was enriching images using image augmentation and training the model with 100 epochs to get the final result. The result is analyzed by using an accuracy comparison for the first. This result can be seen in Figure 3. Figure 3 illustrates the classification accuracy achieved by the four lightweight CNN models. All models surpassed 75% accuracy, indicating strong discriminative capability even with limited training data. Among them, EfficientNetB3 attained the highest accuracy at **95.9%,** MobileNetV2 followed with **79.2%**, EfficientNetB0 achieved **83.3%**, NASNetMobile reached **87.5%**, which, although the lowest, still reflects acceptable classification capability considering its smaller parameter size. These results suggest that deeper or more optimized architectures can better capture the fine-grained details of railway defects.

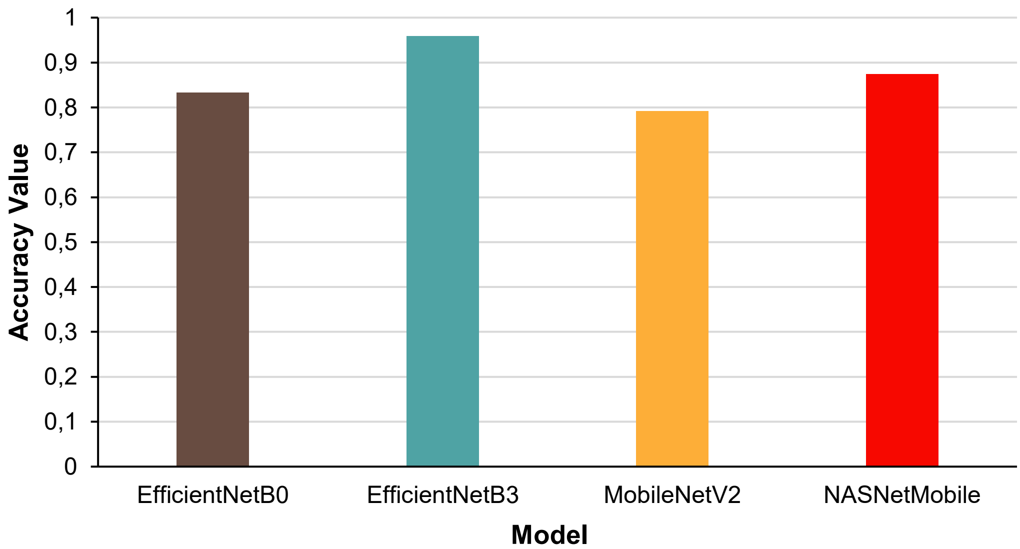

Figure 3. Accuracy Comparison to The Lightweight Model

Figure 4 shows that all models attained more than 0.75 or 75% accuracy in the railway's images. However, the best lightweight model is EfficientNetB3 which achieved a 0.95 accuracy result. Although, the EfficientNetB3 has a higher parameter, either in total parameter or trainable parameter. In the second place, the NasNetMobile achieved 0.875 or 87.5% of accuracy. With fewer total parameters and trainable parameters compared to EfficientNetB3 or EfficientNetB0. Figure 4 presents the ROC curves for all models, showing the trade-off between True Positive Rate (TPR) and False Positive Rate (FPR). EfficientNetB3 displayed the largest area under the curve (AUC ≈ 0.96), indicating excellent ability to distinguish between defective and non-defective classes across different thresholds. MobileNetV2 and EfficientNetB0 achieved AUC values around 0.90 and 0.88, respectively, reflecting strong performance though slightly less robust. NASNetMobile's AUC was about 0.85, showing it is more prone to misclassifications compared to the others. The high AUC scores across all models confirm that lightweight CNNs are capable of reliable defect detection, with EfficientNetB3 standing out as the most effective.

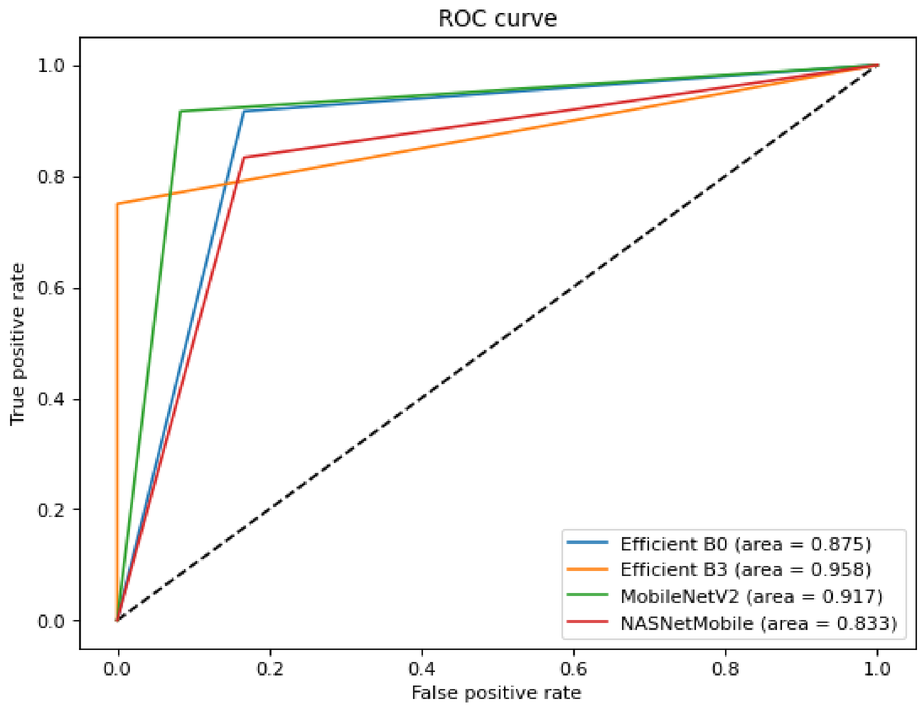

Figure 4. ROG Curve

Broadly, the literature on railway defect inspection can be divided into two main categories: (i) binary/multiclass classification of track conditions from images, similar to the setup of this study, and (ii) object detection or segmentation of specific defects (e.g., cracks, squats, or missing fasteners) using metrics such as mean Average Precision (mAP). Within the first category, An evaluative study by Passos et al. (2022) explored ten popular CNN architectures for detecting surface defects on rails that report classification accuracies in the range of 83–92%, depending on the defect type. For example, InceptionV3 achieved up to 92% accuracy for detecting severe squats, but its performance dropped to around 83.7% for more subtle defects due to the fine-grained nature of visual cues. These findings underline the importance of architectures capable of capturing subtle surface-level features.

In the second category, Choi and Han (2024) evaluated a large number of works adopt R-CNN/YOLO families for defect localization. For instance, Fast R-CNN reported a mAP of 94.9% for railway surface defect detection, reflecting strong recognition ability but with a heavier two-stage inference process. Similarly, an enhanced YOLOv5s achieved 96.9% mAP for surface defect detection in real time, demonstrating that optimized single-stage detection can achieve state-of-the-art precision. By contrast, MobileNetV2 combined with YOLOv3 (a lightweight backbone) obtained 87.4% accuracy across three defect types an attractive option for resource-constrained devices, though still less accurate compared to deeper detection models. Another emerging trend is Min and Li (2022) proposed a self-supervised model detection designed to address the scarcity of defect samples. For example, a two-stage RC-Net combined with DR-VAE achieved an AUC-PR of 0.922 for surface anomaly segmentation, while requiring fewer defect labels. This approach is promising for datasets where defect cases are rare, though its evaluation metrics (AUC/PR) are not directly comparable to classification accuracy.

Other studies have incorporated attention mechanisms or feature-fusion strategies to enhance sensitivity to small-scale defects. For instance, Faster R-CNN augmented with attention modules improved recall on subtle defect areas, while Dual-Path Feature Fusion (DPF) achieved superior performance compared to conventional algorithms. Although evaluation metrics vary across datasets, these studies consistently show that attention and fusion strategies boost detection of fine-grained anomalies. Relative to these findings, our experiments which focused on binary classification of defective versus non-defective tracks show that EfficientNetB3 achieved ~95–96% accuracy, thus performing at the upper bound of CNN-based classification studies (80–92%) and comparable to leading detection methods reporting mAP above 95%. It is important to note, however, that classification accuracy and detection mAP measure different aspects of performance. Nonetheless, the results demonstrate that lightweight CNNs with transfer learning can match or surpass heavier models in discriminative ability, while offering significantly lower computational cost and enabling deployment in edge computing environments.

This study supports the existing body of knowledge by highlighting the feasibility of employing lightweight convolutional neural network architectures for railway surface defect detection, particularly in the context of inference efficiency. While many previous studies have prioritized achieving high accuracy with complex and computationally intensive models, our findings demonstrate that competitive performance can still be achieved with models optimized for real-time deployment. Such an evaluation provides valuable insights into balancing accuracy and computational cost, thereby contributing to ongoing research efforts on edge-based defect detection systems. By emphasizing inference efficiency, this work extends the applicability of computer vision methods in railway safety monitoring where hardware resources are often constrained.

The proposed system offers several advantages, such as achieving high accuracy with lightweight models that can be deployed on edge devices without requiring large computational resources. Its robustness is further enhanced through data augmentation, which improves resistance to variations in lighting, orientation, and scale. Additionally, the use of transfer learning significantly reduces training time and enhances performance compared to models trained from scratch. However, the system also presents certain limitations. The relatively small dataset size may restrict the model's ability to generalize to all real-world conditions. Moreover, the reliance on static image inputs, rather than continuous video data, may limit its applicability in real-time inspection scenarios. Finally, while transfer learning contributes to improved accuracy, it may not fully capture defect patterns that are unique to railway tracks.

IV. CONCLUSION

Several types of lightweight models chosen are EfficientNetB0, EfficientNetB3, MobileNetV2, NasNetMobile. From the results of the evaluation carried out, it was found that EfficientNetB0 was 0.875, EfficientNetB3 was 0.958, MobileNetV2 was 0.917, and NasNetMobile was 0.8333. all models attained more than 0.75 or 75% accuracy in the railway's images. However, the best lightweight model is EfficientNetB3 which achieved a 0.95 accuracy result. The development for inspection of railroads using lightweight models using edge computing is expected to speed up the process of finding damage so that damage to railroads can be repaired immediately.

The development for inspection of railroads using lightweight models with edge computing is expected to speed up the process of detecting damage so that repairs can be conducted immediately. For future research, it is recommended to expand the dataset with larger and more diverse samples that better represent real-world conditions. Exploring real-time video-based defect detection and hybrid deep learning approaches such as attention mechanisms could further improve robustness. Finally, testing on embedded and edge computing platforms is necessary to validate practical deployment in real railway monitoring systems.

V. REFERENCE

[1] C. Tastimur, M. Karakose, and E. Akin, "Image Processing Based Level Crossing Detection and Foreign Objects Recognition Approach in Railways," *International Journal of Applied Mathematics Electronics and Computers*, no. Special Issue-1, pp. 19–23, Sep. 2017.

[2] J. Jang, M. Shin, S. Lim, J. Park, J. Kim, and J. Paik, "Intelligent Image-Based Railway Inspection System Using Deep Learning-Based Object Detection and Weber Contrast-Based Image Comparison," *Sensors*, vol. 19, no. 21, p. 4738, Oct. 2019.

[3] Y. Rubinsztejn and K. Chen, "Automatic Detection of Objects of Interest from Rail Track Images," M.S. thesis, School of Computer Science, University of Manchester, Manchester, U.K., 2011

[4] E. Deutschl, C. Gasser, A. Niel, and J. Werschonig, "Defect Detection on Rail Surfaces by a Vision Based System," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, Parma, Italy, Jun. 2004, pp. 507–510.

[5] M. Deutschl, T. Heiler, and S. Schmid, "Defect Detection on Rail Surfaces by a Vision Based System," in *Proceedings of the 2004 IEEE International Conference on Systems, Man and Cybernetics*, vol. 3, pp. 2271–2276, 2004.

[6] M. Al Rabbani dan M. Hussain, "Lightweight Convolutional Network with Integrated Attention Mechanism for Missing Bolt Detection in Railways," *Metrology*, vol. 4, no. 2, pp. 254–278, 2024.

[7] Z. Zhao, J. Kang, Z. Sun, T. Ye, dan B. Wu, "A Real-Time and High-Accuracy Railway Obstacle Detection Method Using Lightweight CNN and Improved Transformer," *Measurement*, vol. 229, p. 115380, 2024.

[8] J. Li, Y. Fu, D. Yan, S. L. Ma, and C.-W. Sham, "An Edge AI System Based on FPGA Platform for Railway Fault Detection," in *Proceedings of the 2024 IEEE 13th Global Conference on Consumer Electronics (GCCE)*, Auckland, New Zealand, Oct. 2024, pp. 1387–1389.

[9] Z. Zheng, Y. Zhang, and H. Li, "A Real-Time Railway Fastener Inspection Method Using the Lightweight Depth Estimation Network," *Measurement*, vol. 183, p. 109780, 2021.

[10] T. Wang, Z. Zhang, F. Yang, and K. L. Tsui, "Automatic Rail Component Detection Based on AttnConv-Net," *IEEE Sensors Journal*, vol. 22, no. 3, pp. 2379–2388, 2022.

[11] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in *Proceedings of the 36th International Conference on Machine Learning (ICML)*, Long Beach, CA, USA, 2019, pp. 6105–6114.

[12] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, 2018, pp. 4510–4520.

[13] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning Transferable Architectures for Scalable Image Recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, 2018, pp. 8697–8710.

[14] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 2017

[15] H. Cui, J. Li, Q. Hu, dan Q. Mao, "Real-Time Inspection System for Ballast Railway Fasteners Based on Point Cloud Deep Learning," *IEEE Access*, vol. 8, pp. 17038–17050, 2020.

[16] R. S. D. Silva and J. P. Oliveira, "An in-depth assessment of convolutional neural networks for rail surface defect detection," Research, Society and Development, vol. 11, no. 8, p. e12211830252, Jun. 2022, doi: 10.33448/rsd-v11i8.30252.

[17] J.-Y. Choi and J.-M. Han, "Deep learning (Fast R-CNN)-based evaluation of rail surface defects," Applied Sciences, vol. 14, no. 5, p. 1874, Feb. 2024, doi: 10.3390/app14051874.

[18] Y. Min and Y. Li, "Self-supervised railway surface defect detection with defect removal variational autoencoders," Energies, vol. 15, no. 10, p. 3592, May 2022, doi: 10.3390/en15103592.

[19] Y. Mao, S. Zheng, L. Li, R. Shi, and X. An, "Research on rail surface defect detection based on improved CenterNet," Electronics, vol. 13, no. 17, p. 3580, Sep. 2024, doi: 10.3390/electronics13173580.

[20] S. I. Eunus, S. Hossain, A. Adnan, and A. E. M. Ridwan, "Railway Track Fault Detection Dataset," Kaggle, Oct. 2021. [Online]. Available: https://www.kaggle.com/datasets/salmaneunus/railway-track-fault-detection

VI. BIOGRAPHIES

Avip Zain Haq was born in Indonesia. He received a B.Eng. degree in Electrical Engineering with a minor in Power Systems from Universitas Jember, Indonesia. Previously, he completed studies in Applied Informatics and Computer Education (PIKTI) at the Sepuluh Nopember Institute of Technology (ITS). His research interests include artificial intelligence and the Internet of Things (AIoT), renewable energy, generator design, power system analysis, computer vision, halal research, and biomedical informatics.

Cries Avian received his B.Eng and M.Eng degrees in electrical engineering from Universitas Jember, Indonesia. He is currently pursuing a Ph.D. with the Department of Electronic and Computer Engineering, National Taiwan University of Science and Technology. From 2022 to 2023, he was a visiting researcher at the School of Environment and Society, Department of Transdisciplinary Science and Engineering, Tokyo Institute of Technology (Now Institute of Science Tokyo), Japan. His research interests include intelligent control systems, biomedical, signal and image processing, pattern recognition and pervasive computing.

Darma Arif Wicaksono was born in Indonesia. He is pursuing a B.Eng. in Electrical Engineering at Universitas Jember and a Master's degree in Electrical Engineering at Institut Teknologi Sepuluh Nopember. He is a lecturer at Politeknik Negeri Madiun, focusing on power electronics, control systems, and the Internet of Things (IoT).

Rizki Mendung Ariefianto was born in Indonesia. He is received a B.Eng. degree in electrical engineering from Institut Teknologi Sepuluh Nopember, Indonesia, in 2018, the M.Eng. degree in electrical power system from Universitas Brawijaya, in 2022. Currently, he is a Lecturer at Department of Electrical Engineering, Universitas Brawijaya, Indonesia since March 2023. In March 2025, he continues his study in Department of Electrical Engineering Universiti Malaya, Malaysia. His research interests include power electronics, optimization in power systems, solar power systems, turbine study, feasibility study of renewable energy, and power system control.

Mahdin Rohmatillah received the B.Eng. degree majoring in electrical engineering from Brawijaya University, Malang, Indonesia, in 2016, and the M.S. degree majoring in electrical engineering from the National Sun Yatsen Univesity, Kaohsiung, Taiwan, in 2018. He received the Ph.D. degree from National Yang Ming Chiao Tung University, Hsinchu, Taiwan, in 2024. He is right now become an Assistant Professor in Universitas Brawijaya, Malang, Indonesia. His research interests include machine learning, reinforcement learning, and dialogue system