Sensors and Materials, Vol. 37, No. 8 (2025) 3755–3771
MYU Tokyo

S & M 4150

YOLOv3-based Detection Method for Sensing Railway Fastener Defect with Training Data Generated by Generative Adversarial Network Models

Ming-An Chung,[1,2*] Chia-Wei Lin,[1] Jia-Wei Lin,[2] Chun-Chia Lin,[2]
Chen-You Gao,[2] Pu-Chun Chen,[2] and Yi-Xuan Ma[2]

[1]Department of Electronic Engineering, National Taipei University of Technology,
Taipei city 10608, Taiwan, R.O.C.
[2]Master Program in Smart Railway, National Taipei University of Technology,
Taipei city 10608, Taiwan, R.O.C.

(Received November 22, 2024; accepted August 14, 2025)

Keywords: YOLOv3, railway fastener, defect detection, rail transport, GAN, WGAN

Manual railway inspection, being both time-consuming and labor-intensive, no longer meets the demands of modern railway maintenance, where efficiency and precision are essential. To address this issue, an automatic rail fastener detection system is proposed based on the lightweight You Only Look Once version 3 (YOLOv3)-tiny architecture is proposed. This approach not only leverages the speed advantages of YOLOv3-tiny but also incorporates generative adversarial networks (GANs), along with three of its variants, Wasserstein Generative Adversarial Network (WGAN), Divergence Generative Adversarial Network (WGAN-div), and Wasserstein Generative Adversarial Network with Gradient Penalty (WGAN-GP), to augment the dataset and alleviate the problem of limited defective fastener samples. The quality of the generated images is quantitatively evaluated using peak signal-to-noise ratio (*PSNR*), structural similarity (*SSIM*), Fréchet inception distance (*FID*), and inception score (*IS*) metrics. These synthetic samples are then integrated into the original dataset to train the YOLO models in a joint learning process. Experimental results show that, after GAN-based augmentation, YOLOv3-tiny, YOLOv3, YOLOv7-tiny, and YOLOv7 achieve *mAP*0.5 scores of 97.3, 98.7, 98.6, and 98.5% respectively, with particularly significant improvements observed in *mAP*95. These results demonstrate the effectiveness of the proposed method in addressing data imbalance and enhancing both model accuracy and generalization. In addition, analysis of computational complexity and inference speed indicates that YOLOv3-tiny, with only 14.3 Giga Floating Point Operations (GFLOPs) of computational load, achieves an inference speed of 138.9 Frames Per Second (FPS). This high level of real-time performance, combined with high accuracy, makes YOLOv3-tiny a highly suitable choice for deployment on edge devices in practical railway inspection applications.

*Corresponding author: e-mail: mingannchung@ntut.edu.tw
https://doi.org/10.18494/SAM5487

ISSN 0914-4935
https://myukk.org/

1. Introduction

Track fasteners are one of the important components of the track system. Their main function is to securely fix the rails to the sleepers. When trains are running at high speed, the rails are subject to vibration from the wheels for a long time, and the track fasteners are prone to defects such as damage and loss. These defective track fasteners will seriously affect the safety of the railway transportation system. Therefore, regular inspection of track fasteners and replacement of damaged track fasteners are important tasks.

The traditional inspection method of track fasteners is to have inspectors walk along the railway and visually identify whether there are abnormalities. This method not only consumes much labor cost but is also time-consuming. In addition, inspectors can be affected by factors such as bad weather and physical fatigue, resulting in misjudgments or omissions. Therefore, it is necessary to develop an accurate and fast inspection method for tracking fasteners.

In recent years, computer vision and deep learning technology have developed rapidly. Deep learning technology has been widely used in railway inspection, such as track fastener and rail surface defect detection. However, insufficient defect samples or imbalanced data will seriously affect the accuracy and robustness of deep learning networks.[1–3]

To solve the above problems, we conduct track fastener detection based on You Only Look Once version 3 (YOLOv3) and YOLOv3-tiny. Owing to the scarcity of track fastener defect samples, we use a generative adversarial network (GAN) to generate track fastener defect samples to overcome the problem of scarcity of track fastener defect samples. Furthermore, the image recognition system can be seamlessly integrated into edge-computing-enabled cameras, enabling real-time detection and facilitating the transmission of detection results via network communication for advanced processing and analysis.

The main contributions of this article are as follows.

(1) A track fastener detection method based on YOLOv3-tiny is proposed.
(2) The problem of insufficient samples of track fastener defects is solved through GANs.
(3) Multiple sets of experiments on the track fastener dataset are conducted. Experimental results prove that the proposed method effectively solves the problem of insufficient track fastener defect samples and helps improve the accuracy of model detection.

2. Related Work

2.1 YOLO

In recent years, many researchers have developed detection models based on deep learning in various fields. Wei *et al.* proposed an improved model, TLMDDNet, based on YOLOv3 to address the complexity issues in traditional image processing methods. By reducing the feature scale in the convolutional layers, the model enhances detection accuracy. Experimental results show that the mean average precision (*mAP*) reaches 99.2%.[4] Feng *et al.* proposed a railway track defect detection method based on the YOLOv8 model. This method enhances the model's ability to recognize features of different sizes by incorporating the SPD-Conv module into the

backbone network and integrating the EMA attention mechanism into the neck part. The improved model achieved a 5.7% increase in mean accuracy compared with the original YOLOv8n.[5] Li *et al.* proposed a multilevel, end-to-end rail surface defect detection method. The method first performs precise rail extraction based on the standard deviation of edge pixels, then combines differential box counting (DBC) and the GrabCut algorithm for defect segmentation, improving the speed and accuracy of detecting complex defects. Finally, YOLOv2 is used for defect recognition. The method achieves a mean accuracy of 97.11%.[6] Cheng *et al.* proposed a lightweight network based on YOLOv5-lite for railway track damage detection. The network reduces the number of parameters and the amount of computation by using BF_MBConv, enhances feature extraction with the SE mechanism, suppresses redundant features with DropBlock, and improves feature fusion with Shuffle convolution. By introducing the Focal-EIoU loss function and incremental data processing techniques, the detection accuracy and efficiency are significantly improved, achieving an 8.13% performance increase compared with the original YOLOv5-lite.[7] Wang *et al.* proposed the RSG-YOLO model, which was specifically designed for crack detection of the track slab. The model introduces a reparameterized dual-fusion feature pyramid structure, which effectively enhances feature extraction capabilities and reduces data loss. Additionally, the SIoU loss function is used to replace the traditional CIoU, which not only reduces the degree of freedom of the loss function but also significantly accelerates the model's convergence speed. Experimental results show that the *mAP* of RSG-YOLO reaches 94.66%, an improvement of 3.08% over that of YOLOv7.[8]

2.2 GAN

Deep learning models require a large amount of data for training. However, in the detection of track fastener defects, defect samples are scarce, whereas normal samples are abundant, causing data imbalance. This will cause the model to be biased towards normal samples and to ignore defects, thus affecting detection performance. Goodfellow *et al.* proposed a GAN in 2014, whose architecture includes a generator and a discriminator.[13] During the training process, the generator continuously generates fake images in an attempt to make the discriminator misjudge them as real images; while the discriminator continues to learn to improve its ability to distinguish real images from fake ones. This process of competing with each other is like a confrontation. GANs have become a popular method for generating photorealistic images. GANs have spawned many variations and are widely used in a variety of tasks.

Zhao *et al.* proposed a GAN-based transP2P network for supervised image data enhancement to solve the problem of insufficient data in the diamond tool sharpening process. This method can quickly generate a large number of reliable samples containing target defects, effectively combine the distribution structure of training data, and generate a large amount of valuable data for a small number of defect images, overcoming the challenge of insufficient data.[9] Liu *et al.* proposed a synthetic dataset generation method based on GAN and the 3-D CRH380A model for electric multiple unit defect detection, which solved the problem of lack of training data. This method improved the detection accuracy by generating synthetic data, especially when it is difficult to collect low-frequency defect data.[10] Zhang *et al.* proposed a bolt defect image

generation method (DP-GAN) based on a dual discriminator architecture and a pseudo-enhancement strategy to solve the problem of scarce bolt defect samples in transmission lines. In this method, a residual discriminator network was introduced and combined with a dual discriminator GAN architecture to preserve image features and enhance the diversity of generated images. An image fidelity assessment method was designed to improve image quality by screening high-quality fake samples. In addition, the proposed pseudo-enhancement training strategy used fake samples to enhance the few-sample dataset, solving the problem of insufficient generation quality caused by the scarcity of bolt defect images.[11] Geng *et al.* proposed a method that combines a deep convolutional GAN (DCGAN) and a seam carving algorithm to solve the problem of small-sample water-cooled wall defect detection. This method uses a seam carving algorithm to prevent DCGAN from overfitting, generates high-quality images, and utilizes a convolutional neural network for intelligent detection.[12]

3. Methods

To address the substantial labor and time required for traditional manual inspection of railway fasteners, we propose a method that integrates GANs with deep learning techniques to develop an automated rail fastener detection system, as shown in Fig. 1. A key challenge arises from the scarcity and difficulty of obtaining defective fastener samples, which leads to a data imbalance problem that negatively affects model training and detection accuracy. In this research, a dataset comprising both normal and defective rail fastener images was first collected

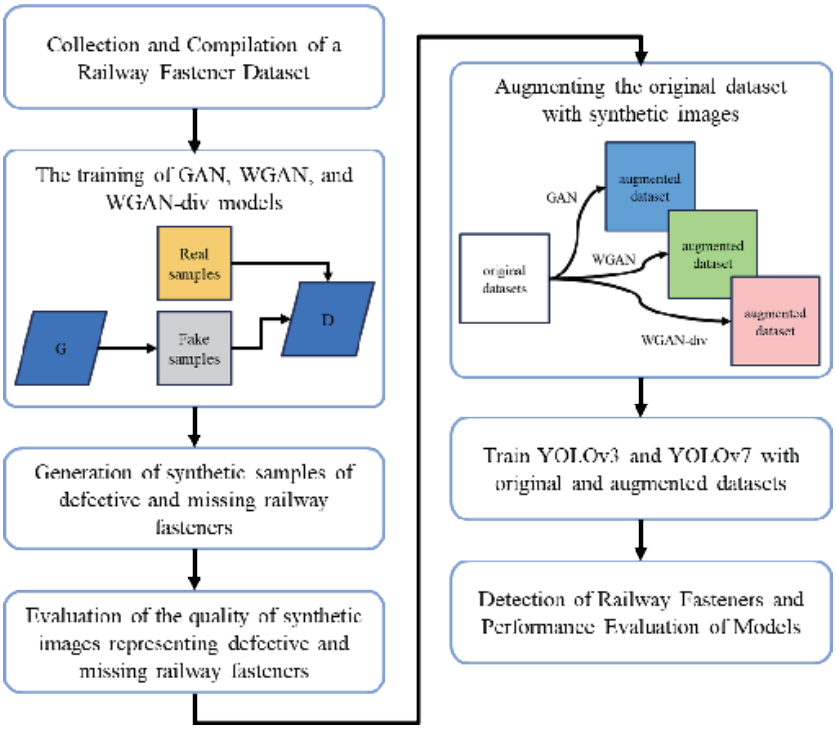


Fig. 1. (Color online) Experiment flow chart.

and curated. Three GAN-based models, GAN, Wasserstein Generative Adversarial Network (WGAN), and Divergence Generative Adversarial Network (WGAN-div), were then trained to generate representative images of defective and missing fasteners to augment the data. By combining synthetic and original images, multiple enhanced datasets were constructed with the goal of alleviating data imbalance. In the model training stage, both the original and augmented datasets were used to train YOLOv3 and YOLOv7 object detection models, followed by a performance comparison and analysis. Final validation was conducted on a separate test set to evaluate detection effectiveness in real-world scenarios. The experimental results show that data augmentation through GANs significantly improves the ability of the models to recognize defective and missing fasteners while also enhancing overall detection accuracy and stability. These outcomes demonstrate the potential for implementing a highly efficient and practical automated inspection system for railway fasteners.

3.1 GANs

GANs, proposed by Goodfellow *et al.* in 2014, represent an innovative training framework for generative models,[13] as illustrated in Fig. 2. GANs utilize an adversarial learning approach by simultaneously training two neural networks, the generator and the discriminator. The generator is responsible for producing samples that resemble the real data distribution, while the discriminator determines whether the input data is real or generated by the generator. These two networks compete during training, and the system reaches an optimal balance when the samples produced by the generator are convincing enough to fool the discriminator.

In practical implementation, both the generator and discriminator are constructed using fully connected layers. The generator takes random noise as input and processes it through multiple fully connected layers combined with batch mormalization and leaky ReLU activation functions, ultimately producing simulated images via a Tanh activation function. The discriminator, on the other hand, flattens the input image into a one-dimensional vector and passes it through several fully connected layers with leaky ReLU activations, outputting a probability between 0 and 1 using a sigmoid function to indicate whether the input is real or generated.

During training, the discriminator's loss function is composed of the classification errors for both real and generated samples and is optimized through backpropagation. The generator

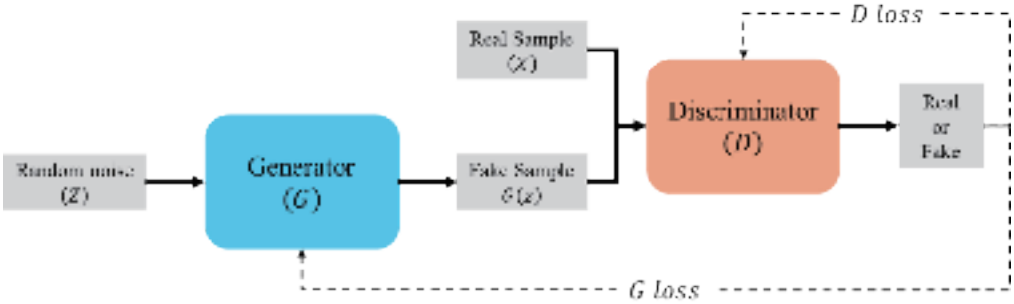


Fig. 2. (Color online) GAN architecture diagram.

updates its weights based on the discriminator's feedback to maximize the probability that the discriminator classifies its outputs as real, i.e., $D\left(G\left(z\right)\right)$ approaches 1. The overall training process can be formulated as a minimax optimization problem:[14]

$$\min_G \max_D V\left(D,G\right) = E_{x\sim Pdata(x)}\left[\log D\left(x\right)\right] + E_{z\sim Pz(z)}\left[\log\left(1-D\left(G\left(z\right)\right)\right)\right], \tag{1}$$

where the goal of the discriminator is to make the predicted value $D\left(x\right)$ for real data xxx approach 1, indicating a high probability that the input is authentic. At the same time, for the samples generated by the generator $G\left(z\right)$, the discriminator aims to make $D\left(G\left(z\right)\right)$ approach 0, meaning it can correctly identify them as fake. Conversely, the generator seeks to make $D\left(G\left(z\right)\right)$ approach 1, so that its generated samples are more likely to be misclassified as real by the discriminator. Through this adversarial training process, the generator continually improves the realism of its outputs, while the discriminator enhances its ability to distinguish between real and synthetic data. Eventually, both models reach a dynamic equilibrium through this competitive interaction.

Accordingly, the loss functions for the generator and the discriminator respectively can be expressed as

$$L_G = E_{z\sim p_z}\left[\log\left(1-D\left(G\left(z\right)\right)\right)\right], \tag{2}$$

$$L_D = -E_{x\sim p_r}\left[\log\left(D\left(x\right)\right)\right] - E_{z\sim p_z}\left[\log\left(1-D\left(z\right)\right)\right], \tag{3}$$

where E denotes the expected value. The generator minimizes the expectation of $\log\left(1-D\left(G\left(z\right)\right)\right)$, aiming to make $D\left(G\left(z\right)\right)$ as close to 1 as possible so that the generated samples are misclassified as real. The discriminator, in contrast, seeks to maximize the probability of correctly identifying real samples while minimizing the likelihood of incorrectly classifying generated samples as authentic.

3.2 YOLOv3

We use YOLOv3 and YOLOv3-tiny to conduct track fastener defect detection experiments. YOLOv3 is a single-stage target detection algorithm that does not need to generate candidate areas, thus improving the speed of object detection. The network architecture of YOLOv3, which can be divided into three parts, backbone, neck, and head, is shown in Fig. 3.

(a) Backbone

The backbone network of YOLOv3 introduces the Darknet53 network structure, which uses a residual network and multiple convolution operations to enhance the ability to extract image features. The function of the residual network is to superimpose features of different dimensions, which helps the network model detect targets of different sizes. Not only that, the

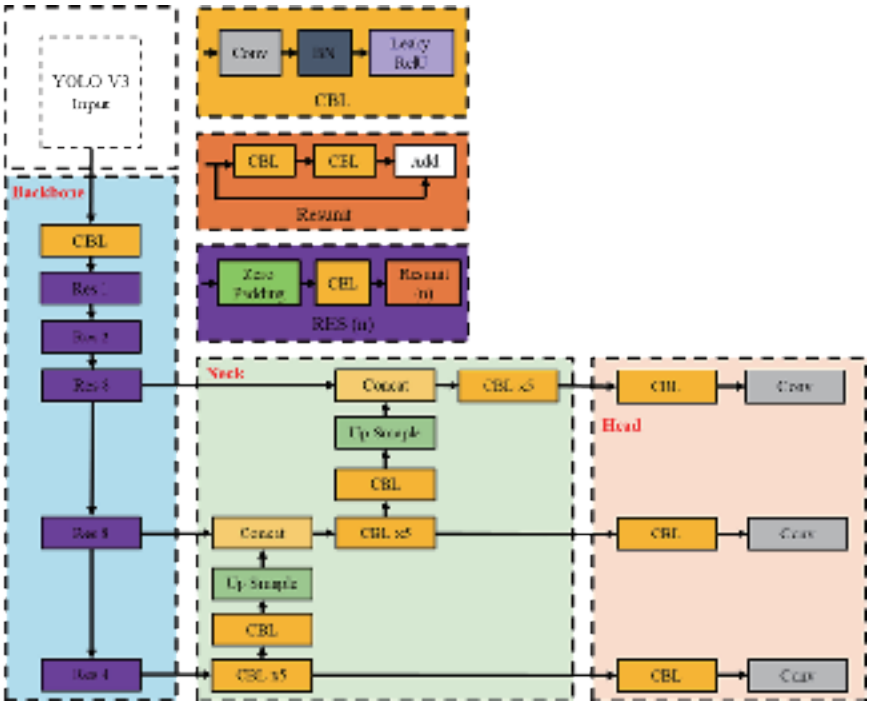


Fig. 3. (Color online) YOLOv3 architecture diagram.

residual network uses skip connections, which effectively alleviates the vanishing gradient problem caused by adding convolutional base layers.

(b) Neck

In the neck network of YOLOv3, the feature pyramid network (FPN) is used for feature fusion. FPN uses horizontal connections and upsampling operations to fuse high-dimensional features with low-dimensional features to obtain global features.

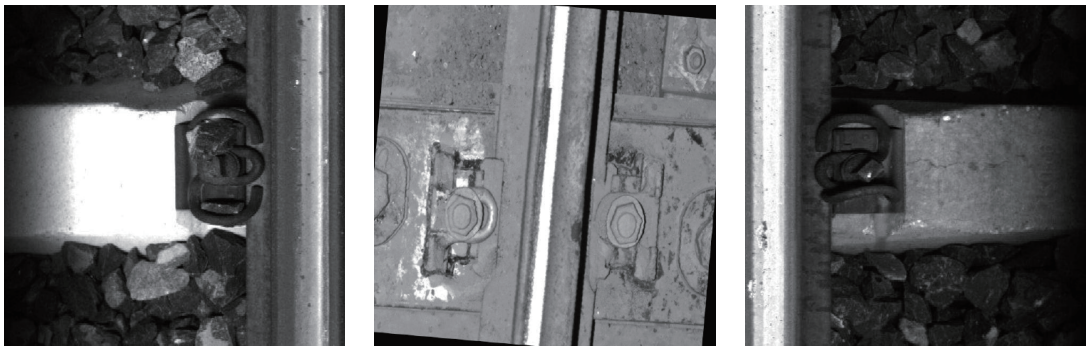
(c) Head

The head network consists of a CBL module and a convolutional layer. The CBL module is used to integrate the prediction information, and the convolutional layer is used to adjust the number of channels. After feature fusion, the neck network passes three different sizes of features to the head network for prediction and selects the best bounding box through nonmaximum suppression to obtain the final prediction result.

4. Experiments

4.1 Dataset

The dataset used in this study was obtained from the open-source platform Roboflow.[15] It is categorized into three classes: normal fasteners, missing fasteners, and damaged fasteners, with 320, 48, and 41 images, respectively. All images have a resolution of 40 × 640 pixels. Figure 4 shows sample images from each category.



(a) (b) (c)

Fig. 4. Sample images of the dataset used in this experiment for three categories: (a) normal fasteners, (b) missing fasteners, and (c) damaged fasteners.

4.2 Experiment platform

All experiments in this study were conducted using the same hardware equipment and under the same environmental conditions, as shown in Table 1.

4.3 Evaluation metrics

In this study, precision, recall, *mAP*0.5, and *mAP*0.95 were adopted as evaluation metrics for assessing the performance of the YOLO algorithm. The corresponding equations for calculation are

$$P = \frac{TP}{TP + FP}, \tag{4}$$

$$R = \frac{TP}{TP + FN}, \tag{5}$$

$$AP = \int_0^1 (P)\, d(R), \tag{6}$$

$$mAP = \frac{1}{n} \sum_{i=1}^{n} AP_i . \tag{7}$$

P stands for precision, R for recall, and AP for average accuracy. *mAP*0.5 represents the average accuracy of each category when the IoU threshold is 0.5. *mAP*0.95 represents the average accuracy of each category when the IoU threshold is 0.95.[16]

To generate high-quality images of rail fastener defects, we use the evaluation metrics peak signal-to-noise ratio (*PSNR*), structural similarity (*SSIM*), Fréchet inception distance (*FID*), and inception score (*IS*) to evaluate the images generated by each GAN.

Table 1
Hardware and software.

Operating System	Windows 11
CPU	13th Gen Intel(R) Core(TM) i5-13500H
GPU	Nvidia RTX4060
RAM	64 GB
Python	3.8
Pytorch	1.12
CUDA	11.3

The *FID* is a metric that measures the difference between the distributions of generated and real data obtained by calculating the covariance matrix and mean values. A lower *FID* score indicates that the distribution of generated data is closer to that of real data, resulting in better generation quality.[17] The *FID* calculation formula is[18]

$$FID = \left\| \mu_r - \mu_g \right\|^2 + T_r\left(\sum\nolimits_r + \sum\nolimits_g - 2\left(\sum\nolimits_r \sum\nolimits_g \right)^{0.5} \right). \tag{8}$$

u_r represents the average value of real image features, u_g the average value of generated image features, Σ_r the covariance value of real image features, and Σ_g the covariance value of generated image features.

The *PSNR* function is an evaluation metric for the reconstruction quality in image processing. The *PSNR* is calculated as[19]

$$PSNR = 10\log\frac{\left(2^n - 1\right)^2}{MSE}, \tag{9}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(g_i - p_i\right)^2, \tag{10}$$

where *MSE* is the mean square error between *g* and *p*. Here, *g* represents the generated image and *p* represents the real image.

The *SSIM* function is used for measuring the similarity between two images. Assuming that *x* is a real image and *y* is a generated image, the *SSIM* between *x* and *y* can be expressed as[20]

$$SSIM\left(x, y\right) = \frac{\left(2\mu_x\mu_y + c_1\right)\left(2\sigma_{xy} + c_2\right)}{\left(\mu_x^2\mu_y^2 + c_1\right)\left(\sigma_x^2 + \sigma_y^2 + c_2\right)}, \tag{11}$$

where u_x represents the mean of x, u_y the mean of y, σ_x^2 the variability of the x data, and σ_y^2 the correlation of the y data. u_{xy} is the degree of correlation between x and y. The constants c_1 and c_2 are used to prevent instability during the calculation process.

The *IS* is a metric used to evaluate synthetic images. The higher the *IS* is, the better the synthesized image is. The calculation formula is [21]

$$IS = \exp\left(E_{x\sim p_{data}(x)}\left[DKL\left(p\left(y\,|\,x\right)\|\,p\left(y\right)\right)\right]\right), \tag{12}$$

where $x \sim p_{data}$ means that x is an image sample from p_{data}. $p(y|x)$ represents the category distribution predicted by the generative model for the image x. $p(y)$ represents the image distribution generated by the generative model. *DKL* represents the Kullback–Leibler divergence of the generated image, which is used to measure the difference between the category distribution $p(y|x)$ of a single image and the overall distribution $p(y)$.

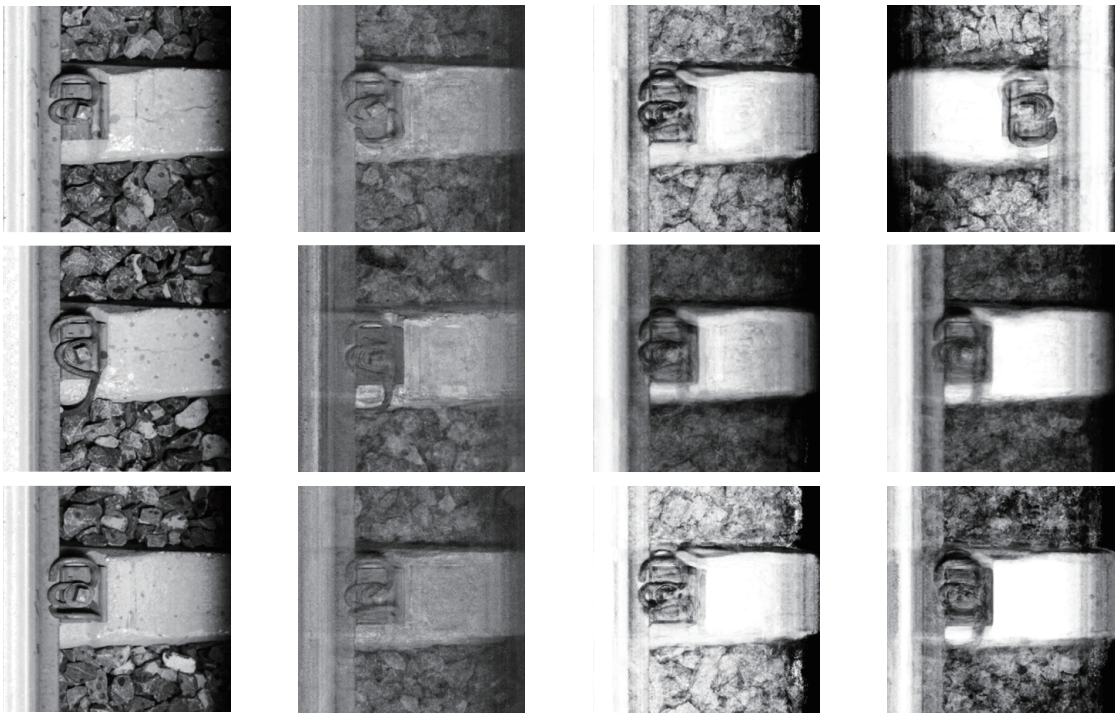
4.4 Experimental results

Owing to the significantly lower number of missing and damaged fastener samples than of normal fasteners, the dataset suffers from an imbalanced distribution that affects the overall performance of deep learning models. To overcome this problem, we use 48 images of missing fasteners and 41 images of damaged fasteners as training data for three types of GANs, namely, GAN, WGAN, and WGAN-div. The training images are resized to 640 × 640 pixels, and each model is trained for a total of 5000 epochs. After training is completed, the models are used to generate additional defective fastener images, as shown in Figs. 4 and 5.

Figures 5 and 6 respectively show the original dataset and the defective and missing fastener images generated by GAN, WGAN, and WGAN-div. A visual comparison reveals noticeable differences in image quality among the three types of GANs. Overall, GAN demonstrates superior performance in both types of generation tasks, particularly in terms of structural clarity and texture preservation. Details such as the edge contours of fasteners, the background elements of tracks and ballast, as well as the visual transitions and contrast in missing fastener regions, are rendered with high fidelity. The generated images exhibit a natural appearance and closely resemble real-world images, offering strong visual distinguishability and potential for practical applications.

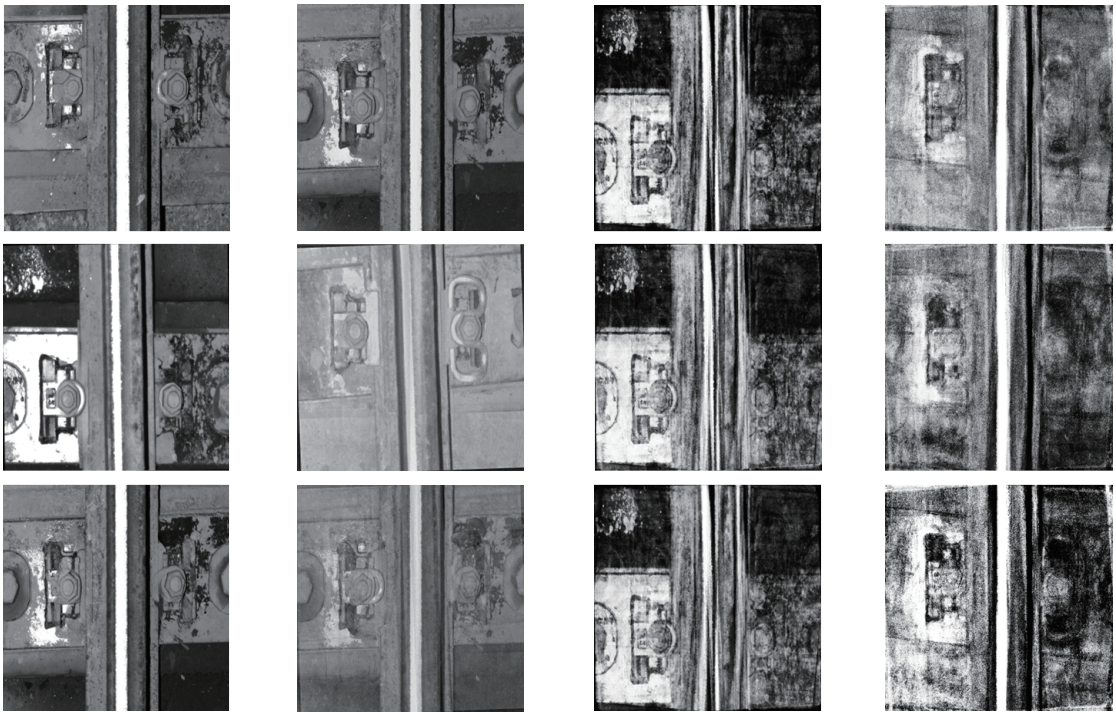
In contrast, although WGAN enhances image contrast in certain cases, it frequently suffers from issues such as local darkness, blurred details, and structural distortions, which hinder the identification and interpretation of defect areas. Images generated by WGAN-div often exhibit excessive blurriness and signal distortion, resulting in poorly defined boundaries between fasteners and their background. These deficiencies significantly reduce the overall usability of the images.

The results of a comprehensive analysis indicate that GAN produces the highest-quality images within the context of this study. Not only does it enrich the dataset with greater diversity and realism, but it also enables the detection models to learn more representative features for identifying abnormalities. These findings highlight the practicality and robustness of GAN for data augmentation in scenarios with limited sample availability.



(a) (b) (c) (d)

Fig. 5. Defective fastener images generated using a GAN: (a) original dataset image, (b) GAN, (c) WGAN, and (d) WGAN_div.



(a) (b) (c) (d)

Fig. 6. Missing fastener images generated using GANs: (a) original dataset image, (b) GAN, (c) WGAN, and (d) WGAN_div.

To objectively evaluate the quality and performance of different GANs in generating images of defective rail fasteners, we adopt multiple evaluation metrics. The experimental results are presented in Table 2. In terms of *PSNR*, the GAN model achieved the best performance with a score of 22.16, indicating that its generated images are more similar to real images at the pixel level. Regarding *SSIM*, the WGAN model obtained the highest score of 0.627, suggesting a better ability to preserve *SSIM*. For the *FID* metric, the GAN model recorded the lowest value of 196.499, demonstrating that its generated images are the closest to real images in terms of feature distribution and thus exhibit greater realism. As for the *IS* metric, the WGAN-div model achieved the highest score of 2.1048, indicating a relative advantage in image diversity and recognizability. Overall, each model shows strengths across different evaluation dimensions.

Table 3 presents the evaluation results for the generation of missing fastener images. The GAN model achieved a *PSNR* of 23.25 and an *SSIM* of 0.707, outperforming the other two models in both pixel-level reconstruction and *SSIM*. It also recorded the lowest *FID* score of 132.978, indicating that its generated images are most similar to real images in terms of feature distribution, and therefore exhibit higher realism. Although the WGAN-div model obtained the highest *IS* score of 2.323, reflecting a slight advantage in terms of image diversity and recognizability, the GAN model demonstrated the most consistent performance across most evaluation metrics. These results indicate that the GAN model holds a relative advantage in the task of generating images of missing rail fasteners.

To verify whether the synthetic defect samples of rail fasteners generated using the proposed image generation methods can effectively improve the performance of detection models, we first utilized GAN, WGAN, and WGAN-div to generate 272 images of missing fasteners and 279 images of damaged fasteners. These generated images were then combined with the original dataset to form three augmented datasets. Each augmented dataset consisted of three categories, with 320 images per category, resulting in a total of 960 images per dataset.

Subsequently, YOLOv3, YOLOv7, and their lightweight variants, YOLOv3-tiny and YOLOv7-tiny, were trained using both the original dataset and three augmented datasets. The results obtained from the original dataset were used as the baseline for comparative analysis. All

Table 2
Evaluation metric scores for damaged fastener images generated by various GAN algorithms (GAN, WGAN, and WGAN_div).

	PSNR	*SSIM*	*FID*	*IS*
GAN	22.16	0.447	196.499	2.033
WGAN	19.49	0.627	224.852	2.089
WGAN_div	14.76	0.429	230.349	2.105

Table 3
Evaluation metric scores for missing fastener images generated by various GAN algorithms (GAN, WGAN, and WGAN_div).

	PSNR	*SSIM*	*FID*	*IS*
GAN	23.25	0.707	132.978	1.862
WGAN	16.65	0.375	230.823	1.655
WGAN_div	15.65	0.223	312.880	2.323

models were trained under identical settings, including 1000 training epochs, a batch size of 16, an input image resolution of 640, and an initial learning rate of 0.01. The performance of each model across different datasets was then analyzed and compared. The experimental results are presented in Table 4.

In general, the inclusion of GAN-generated samples significantly improved the detection performance across all models, with the most prominent enhancement observed in the *mAP*0.95 metric. Specifically, *mAP*0.95 increased from 62.2 to 73.0% for YOLOv3-tiny, from 62.3 to 73.6% for YOLOv3, from 62.9 to 73.3% for YOLOv7-tiny, and from 63.3 to 72.6% for YOLOv7. In addition, most models also showed improvement in precision, recall, and *mAP*0.5, indicating that GAN-based augmentation effectively enhances both model accuracy and generalization.

In contrast, WGAN-based augmentation resulted in moderate improvements in *mAP*0.5 and recall in some models, but often led to decreased precision, which limited its overall effectiveness. WGAN-div showed better performance in terms of recall, reaching over 97% in some models. However, its precision remained lower, and the improvement in *mAP*0.95 was not as significant as that achieved using GAN.

In summary, among the three augmentation methods, the GAN-based approach yielded the most notable enhancement in overall detection performance, particularly in *mAP*0.95. These findings demonstrate the strong potential of GAN-generated samples in addressing data imbalance and improving model generalization for the task of rail fastener defect detection.

Since we use GAN to expand the original rail fastener dataset, it can effectively improve the training performance of each version of the YOLO model, so that they all have good detection accuracy. On the premise that the accuracy rate meets the standard, the computational complexity (giga floating point operations, GFLOPs) and inference speed (frames per second, FPS) of the model are further compared to evaluate its real-time performance in practical

Table 4
Performances of YOLOv3-tiny, YOLOv3, YOLOv7-tiny, and YOLOv7 models trained on the original dataset and three augmented datasets.

Model	Dataset	Precision (%)	Recall (%)	*mA*P0.5 (%)	*mAP*0.95 (%)
YOLOv3-tiny	original dataset	93.6	97.0	96.9	62.2
	original dataset + GAN	95.4	96.5	97.3	73.0
	original dataset + WGAN	92.2	88.5	94.6	63.2
	original dataset + WGAN-div	91.2	94.6	96.2	63.9
YOLOv3	original dataset	90.6	88.8	93.4	62.3
	original dataset + GAN	95.2	97.3	98.7	73.6
	original dataset + WGAN	92.2	90.9	95.8	64.2
	original dataset + WGAN-div	90.3	97.1	96.4	66.2
YOLOv7-tiny	original dataset	87.2	93.1	93.5	62.9
	original dataset + GAN	97.3	97.1	98.6	73.3
	original dataset + WGAN	93.8	90.2	96.5	64.4
	original dataset + WGAN-div	89.9	96.7	96.8	64.8
YOLOv7	original dataset	96.1	85.5	94.0	63.3
	original dataset + GAN	95.7	98.6	98.5	72.6
	original dataset + WGAN	90.2	91.4	94.8	62.4
	original dataset + WGAN-div	89.5	97.7	95.8	64.1

Table 5
GFLOPs and FPS performance of different YOLO models with the GAN augmented dataset.

	GFLOPs	FPS
YOLOv3-tiny	14.3	138.9
YOLOv3	261.8	24.1
YOLOv7-tiny	13.0	101.1
YOLOv7	103.2	31.2

(a) (b) (c)

Fig. 7. (Color online) Test results of YOLOv3-tiny with the test dataset (original dataset + GAN): (a) fasteners are normal, (b) fasteners are missing, and (c) fasteners are damaged.

applications. According to the experimental results in Table 5, YOLOv3-tiny achieves an inference speed of up to 138.9 FPS with only 14.3 GFLOPs of computation, which is significantly better than those of the other models, showing that it is lightweight while still having a high degree of real-time performance, making it particularly suitable for scenarios with strict speed requirements such as rail fastener detection.

Finally, Fig. 7 shows the results of detection using GAN to generate images to expand the original dataset and train YOLOv3-tiny. According to the results, the proposed rail fastener detection method can effectively detect rail fastener defects.

5. Conclusions

We proposed an automated rail fastener inspection method based on YOLOv3-tiny, which integrates edge computing devices with cameras for practical deployment in real-world rail fastener inspection tasks. In the experiments, a self-collected dataset containing damaged and missing rail fasteners was used to train GAN, WGAN, WGAN-div, and Wasserstein Generative Adversarial Network with Gradient Penalty (WGAN-GP) models in order to generate defect images and address the issue of data imbalance. To evaluate the quality of the generated images, four metrics were employed for analysis and comparison: *PSNR*, *SSIM*, *FID*, and *IS*. Subsequently, the defect images generated by the three image synthesis methods were used to augment the original dataset. The augmented datasets were then used to train four object detection models, namely, YOLOv3-tiny, YOLOv3, YOLOv7-tiny, and YOLOv7. The detection performances of these models were compared against their counterparts trained on the original

dataset without augmentation. Experimental results indicated that the defect images generated by GAN achieved the best performance across all image quality evaluation metrics and effectively enhanced the detection performance of the various YOLO models. Among them, the YOLOv3-tiny model not only achieved a high accuracy of 97.3% in terms of *mAP* at 0.5 but also demonstrated superior real-time performance with an inference speed of 138.9 FPS, significantly outperforming the other models. Overall, the proposed method effectively addresses the data imbalance problem while maintaining both detection accuracy and inference speed. This demonstrates its high feasibility and practical value for real-time rail fastener inspection tasks.

Acknowledgments

This work was supported by the National Science and Technology Council, Taiwan, R.O.C. under Grant NSTC 113-2221-E-027-068.

References

1 S. Su, S. Du, and X. Lu: IEEE Trans. Intelligent Transp. Syst. **23** (2022) 23883. https://doi.org/10.1109/TITS.2022.3207490
2 Z. Jian, S. He, S. Liu, J. Liu, and Y. Fang: IEEE Trans. Instrum. Meas. (2024). https://doi.org/10.1109/TIM.2024.3370748
3 J. Liu, Y. Teng, X. Ni, and H. Liu: IEEE Sens. J. **21** (2021) 12179. https://doi.org/10.1109/JSEN.2021.3062021
4 X. Wei, D. Wei, D. Suo, L. Jia, and Y. Li: IEEE Access **8** (2020) 61973. https://doi.org/10.1109/ACCESS.2020.2984264
5 J. Feng, H. Shi, J. Qiu, Z. Yu, and C. He: IEEE Sens. J. (2024). https://doi.org/10.1109/JSEN.2024.3448308
6 C. Li, A. Xu, Q. Zhang, and Y. Cai: IEEE Access (2024). https://doi.org/10.1109/ACCESS.2024.3374869
7 Z. Cheng, L. Gao, Y. Wang, Z. Deng, and Y. Tao: IEEE Access (2024). https://doi.org/10.1109/ACCESS.2024.3391353
8 Y. Wang, K. Zhang, L. Wang, and L. Wu: IEEE Access (2024). https://doi.org/10.1109/ACCESS.2024.3380009
9 C. Zhao, W. Xue, W.-P. Fu, Z.-Q. Li, and X. Fang: IEEE Trans. Instrum. Meas. **72** (2023) 1. https://doi.org/10.1109/TIM.2023.3284139
10 S. Liu, H. Ni, C. Li, Y. Zou, and Y. Luo: IEEE Sens. J. (2024). https://doi.org/10.1109/JSEN.2024.3386711
11 K. Zhang, Y. Xiao, J. Wang, M. Du, X. Guo, R. Zhou, C. Shi, and Z. Zhao: IEEE Trans. Power Delivery (2024). https://doi.org/10.1109/TPWRD.2024.3373130
12 Z. Geng, C. Shi, and Y. Han: IEEE Trans. Ind. Inf. **19** (2022) 7489. https://doi.org/10.1109/TII.2022.3159817
13 I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio: Commun. ACM **63** (2020) 139. https://doi.org/10.1145/3422622
14 L. Wang, W. Chen, W. Yang, F. Bi, and F. R. Yu: IEEE Access **8** (2020) 63514. https://doi.org/10.1109/ACCESS.2020.2982224
15 Roboflow Universe (Online) Available: https://universe.roboflow.com/
16 J. Xu, H. Yang, Z. Wan, H. Mu, D. Qi, and S. Han: IEEE Access **11** (2023) 105281. https://doi.org/10.1109/ACCESS.2023.3303890
17 H. Wang and L. Ma: IEEE Access **11** (2023) 61855. https://doi.org/10.1109/ACCESS.2023.3287854
18 A. Manassakorn, S. Auethavekiat, V. Sa-Ing, S. Chansangpetch, K. Ratanawongphaibul, N. Uramphorn, and V. Tantisevi: IEEE Access **12** (2024) 145845. https://doi.org/10.1109/ACCESS.2024.3464682
19 X. Wang, D. Xu, and F. Gu: IEEE Access **8** (2020) 170355. https://doi.org/10.1109/ACCESS.2020.3024288
20 W. Wei, J. Yan, X. Wu, C. Wang, and G. Zhang: IEEE Wireless Commun. Lett. **11** (2022) 1137. https://doi.org/10.1109/LWC.2022.3159104
21 M. A. Habib, M. A. H. Wadud, L. Y. Pinky, M. M. H. Talukder, M. M. Rahman, M. F. Mridha, Y. Okuyama, and J. Shin: IEEE Access (2023). https://doi.org/10.1109/ACCESS.2023.3342866

About the Authors

Ming-An Chung received his B.Eng. and M.Eng. degrees in electronic engineering from the Chang Gung University, Taoyuan, Taiwan, and his D.Eng. degree in electrical engineering from the National Taiwan University of Science and Technology (NTUST), Taipei, Taiwan, in 2003, 2005, and 2016, respectively. He is currently an associate professor at the Department of Electronic Engineering, National Taipei University of Technology (NTUT), where he also serves as the leader of the Innovation Wireless Communication and Electromagnetic Applications Laboratory. His research interests include wireless communication propagation, intelligent robotics, self-driving vehicles, antenna design for various mobile and wireless communications, and electromagnetic theory and applications. He is also a reviewer of many scientific journals, including the IEEE Transactions on Antennas and Propagation, IEEE Transactions on Industrial Informatics, Journal of Intelligent & Robotic Systems, IET Microwaves, Antennas and Propagation, IEEE Antennas and Wireless Propagation Letters, International Review of Electrical Engineering, International Journal on Communications Antenna and Propagation and AEÜ - International Journal of Electronics and Communications, and many international conferences, including ICRA, ICCE-TW, RFIT, ICBEB, EMCAR and SNSP. (mingannchung@ntut.edu.tw)

Chia-Wei Lin received his B.S. and M.S. degrees from Chung Yuan Christian University in 2007 and is currently pursuing his Ph.D. degree in electrical engineering at the National Taipei University of Technology. His research interests include wireless communication propagation research, antenna design, intelligent robotics research, self-driving vehicle research, embedded systems, and deep learning. (t109369011@ntut.edu.tw)

Jia-Wei Lin received his B.S. degree in technology for smart living from the Huafan University, New Taipei City, Taiwan, in 2023. He is currently pursuing his M.S. degree in smart railways with the National Taipei University of Technology (NTUT), Taipei, Taiwan. His research interests are in smart railways, image recognition, and sensors. (t112c02011@ntut.edu.tw)

Chun-Chia Lin received his B.S. degree in mechanical engineering from the National Chin-Yi University of Technology, Taichung, Taiwan, in 2022. He is currently pursuing his M.S. degree in smart railways with the National Taipei University of Technology (NTUT), Taipei, Taiwan. His research interests are in smart railways, image recognition, and sensors. (t112c02014@ntut.edu.tw)

Chen-You Gao received his B.S. degree in technology for smart living from the National Huafan University, New Taipei City, Taiwan, in 2023. He is currently pursuing his M.S. degree in smart railways with the National Taipei University of Technology (NTUT), Taipei, Taiwan. His research interests are in smart railways, image recognition, and sensors. (t112c02012@ntut.edu.tw)

Pu-Chun Chen received his B.S. degree in electronic engineering from National Ilan University, Ilan, Taiwan, in 2023. He is currently pursuing his M.S. degree in smart railways with the National Taipei University of Technology (NTUT), Taipei, Taiwan. His research interests are in smart railways, image recognition, and sensors. (t112c02015@ntut.edu.tw)

Yi-Xuan Ma received his B.S. degree in electronic engineering from National Ilan University, Ilan, Taiwan, in 2023. He is currently pursuing his M.S. degree in smart railways with the National Taipei University of Technology (NTUT), Taipei, Taiwan. His research interests are in smart railways, image recognition, and sensors. (t112c02013@ntut.edu.tw)