



Rail Tracks Defect Detection Using Deep Learning

¹Gourav, ²Ruchi Patira,

¹MTech. Student, ²Assistant Professor,

¹Department of Computer Science and Engineering,

¹World College of Technology & Management, Gurgaon, Haryana- INDIA

Abstract: Railroad infrastructure is crucial for transportation safety and efficiency, with defects like cracks, misalignments, and wear posing serious risks. While traditional manual inspections are labor-intensive and prone to error, machine learning (ML) and deep learning (DL) techniques offer automated, accurate solutions. In contrast, deep learning models, particularly Convolutional Neural Networks (CNNs) like VGG16, ResNet50, and InceptionV3, autonomously extract features from raw data, improving defect detection. Semantic segmentation with models like VGG16 and InceptionV3 enables precise defect localization. Transfer learning and synthetic data generation address challenges such as limited datasets and complex environments. Despite progress, issues like imbalanced datasets and real-time deployment persist, but integration of transfer learning offer solutions to improve operational safety and accuracy. Comparative analysis reveals that VGG16 outperforms ResNet50 and InceptionV3, achieving the highest accuracy of 91.64%, followed by ResNet50 (89.46%) and InceptionV3 (81.26%).

Index Terms – Rail Track Defect, Anomaly Detection, Deep Learning, Artificial Neural Network, Convolutional Neural Network.

1. INTRODUCTION

Railroad infrastructure plays a pivotal role in global transportation, requiring robust maintenance and monitoring systems to ensure safety and efficiency. Defects in rail tracks, such as cracks, misalignments, or wear, pose significant risks, potentially leading to derailments or operational delays. Traditionally, manual inspections have been employed for rail defect detection, which are labor-intensive, time-consuming, and prone to human error (He et al., 2020). Recent advancements in machine learning (ML) and deep learning (DL) techniques have revolutionized the detection process by offering automated, accurate, and scalable solutions for anomaly identification in railroads.

Machine learning techniques like Support Vector Machines (SVM), Random Forests, and Decision Trees have been utilized for defect detection by analyzing sensor-based or image-based data. These models leverage handcrafted features such as texture, edge, and gradient to classify anomalies (Zhang et al., 2019). For instance, vibration and acoustic data collected via sensors can be processed using ML algorithms to identify deviations indicative of defects. However, the effectiveness of ML models is often constrained by their dependency on feature engineering, which may not generalize well to diverse and complex rail conditions (Chen et al., 2021).

In contrast, deep learning models, particularly Convolutional Neural Networks (CNNs), have demonstrated superior performance in rail defect detection by autonomously extracting features from raw data (He et al., 2020). Architectures like VGG16, ResNet50, and InceptionV3 have been employed for image-based rail track anomaly detection. These models can process high-resolution images captured by drones or inspection vehicles, identifying minute cracks or deformations. Furthermore, hybrid models combining CNNs with Recurrent Neural Networks (RNNs) have been used to analyze temporal data, such

as vibration patterns over time, providing a more holistic approach to anomaly detection (Zhang et al., 2019).

One prominent application of DL in railroad anomaly detection involves the use of U-Net and its variants for semantic segmentation. This approach enables pixel-wise classification of defects, which is particularly useful for identifying crack patterns and their precise locations (Chen et al., 2021). Recent studies have also explored the integration of attention mechanisms within DL architectures to enhance feature localization and focus on critical regions of rail tracks. Such enhancements improve the detection accuracy in challenging scenarios like varying lighting conditions or occlusions caused by debris (Badrinarayanan et al., 2017).

Another emerging trend is the utilization of transfer learning and pre-trained models to address the challenges of limited labeled datasets. Pre-trained models like ResNet50 and InceptionV3 can be fine-tuned for rail defect detection tasks, significantly reducing the computational cost and training time (He et al., 2020). Additionally, synthetic data generation techniques, such as Generative Adversarial Networks (GANs), have been employed to augment datasets, ensuring the robustness of models across diverse environments and defect types (Goodfellow et al., 2014).

While these advancements are promising, challenges such as imbalanced datasets, real-time deployment, and false-positive rates remain areas of active research. Ensemble learning techniques combining multiple models have been proposed to mitigate these issues, improving both precision and recall (Zhang et al., 2019). Furthermore, integrating DL models with Internet of Things (IoT) devices, such as smart sensors and edge computing platforms, facilitates real-time anomaly detection and alert systems, enhancing operational safety and efficiency (Chen et al., 2021).

2. LITERATURE REVIEW

Efficient rail defect detection is crucial for ensuring the safety and reliability of rail transportation systems. Over the years, researchers have explored various methodologies, including traditional machine learning and advanced deep learning techniques, to address the challenges associated with rail track defect detection. This literature review synthesizes key studies and their contributions to the domain.

- *Traditional Machine Learning Approaches*

Initial approaches to rail defect detection relied on traditional machine learning models, leveraging sensor-based and image-based datasets. Zhang et al. (2019) demonstrated the use of Support Vector Machines (SVM) and Random Forests for processing acoustic and vibration data, identifying track irregularities. These methods required feature engineering to extract texture, edge, and gradient features, which were then used to classify defects. Despite their utility, these models often struggled to generalize well to varying environmental conditions due to their dependency on handcrafted features.

- *Advances in Deep Learning Architectures*

The advent of deep learning (DL) has transformed rail defect detection by automating feature extraction and enhancing detection accuracy. He et al. (2020) employed Convolutional Neural Networks (CNNs) for rail surface defect detection, demonstrating significant improvements over traditional methods. Popular CNN architectures, including VGG16, ResNet50, and InceptionV3, have been widely adopted for analyzing high-resolution rail images captured by drones or inspection vehicles. These models effectively identify surface cracks, misalignments, and other anomalies with high precision. Zhang et al. (2019) further extended this work by incorporating Recurrent Neural Networks (RNNs) to analyze temporal data, enabling the detection of time-series patterns in rail track conditions.

- *Semantic Segmentation for Precise Defect Localization*

Semantic segmentation techniques, particularly using U-Net and its variants, have gained prominence in rail defect detection. Chen et al. (2021) applied U-Net with attention mechanisms to perform pixel-wise classification of rail defects, achieving remarkable accuracy in identifying crack patterns and their exact locations. This method is particularly advantageous for scenarios requiring detailed defect mapping, as it provides a fine-grained understanding of the defect's spatial characteristics.

- *Transfer Learning and Dataset Augmentation*

A significant challenge in applying DL to rail defect detection is the lack of extensive labeled datasets. He et al. (2020) addressed this issue through transfer learning, leveraging pre-trained models like ResNet50 and InceptionV3 to reduce training time and computational requirements. Additionally, Goodfellow et al. (2014) introduced Generative Adversarial Networks (GANs) as a means to augment datasets by generating synthetic images, ensuring model robustness across diverse defect types and environmental conditions.

- **Attention Mechanisms and Hybrid Models**

Recent research has focused on enhancing DL models through attention mechanisms and hybrid architectures. Attention mechanisms help models focus on critical regions in rail images, improving defect localization under challenging conditions such as variable lighting or occlusions (Chen et al., 2021). Hybrid models combining CNNs and RNNs have shown promise in integrating spatial and temporal information, providing a comprehensive understanding of rail anomalies (Zhang et al., 2019).

- **IoT Integration and Real-Time Detection**

Emerging trends emphasize the integration of DL models with Internet of Things (IoT) devices for real-time rail defect detection. Chen et al. (2021) highlighted the potential of combining smart sensors with edge computing platforms to facilitate real-time anomaly identification and alert systems. This approach enhances operational efficiency and safety by enabling timely maintenance interventions.

Author(s) and Year	Title	Abstract/Objective	Results
C. Breitenberger et al., 2018	<i>Automated Visual Inspection of Railway Tracks Using CNNs</i>	The study introduces a CNN-based method for the automated visual inspection of railway tracks to identify cracks and anomalies.	Achieved high accuracy in detecting cracks and missing components in track images.
Ji et al., 2021	<i>Rail track condition monitoring: a review on deep learning approaches</i>	Explores deep learning models, including CNNs and RNNs, for monitoring the structural conditions of rail tracks and detecting anomalies such as cracks and misalignments.	Demonstrated improved performance with deep learning methods compared to traditional inspection techniques.
Z. Zhang et al., 2021	<i>Surface Defect Detection on Rail Tracks via Deep Learning</i>	Proposes a deep learning-based framework for detecting surface defects in railway tracks using image data and pre-trained networks for feature extraction.	The model detected defects with over 92% accuracy on test datasets, showcasing its effectiveness for surface damage classification.
P. Kumar et al., 2022	<i>Hybrid CNN-RNN for Rail Track Anomaly Detection</i>	Combines convolutional and recurrent neural networks for detecting both surface and subsurface rail track anomalies.	The hybrid model outperformed standalone CNN and RNN models in detecting subsurface anomalies with a 95% accuracy rate.
J. Lee & S. Park, 2024	<i>Deep Learning for Rail Track Condition Monitoring: A Review</i>	A comprehensive review of the advancements and applications of deep learning techniques, focusing on anomaly detection and condition monitoring in rail tracks.	Summarized key challenges, trends, and future directions, emphasizing the potential of integrated DL techniques for large-scale rail systems.
Hinton et al. (2019)	"A Deep Convolutional Neural Network for Detection of Rail Surface Defect"	Proposes a CNN-based model for identifying surface defects on railways, addressing challenges like occlusions and varying defect scales.	Achieved high accuracy with real-world rail data, emphasizing robustness to variable lighting and defect types.

Zhang et al. (2019)	"TrackNet: A Deep Learning-Based Fault Detection for Railway Track Inspection"	Develops TrackNet, a CNN-based model tailored for real-time track inspections, aiming for higher efficiency in detection tasks.	Demonstrated improved defect detection speed and accuracy compared to traditional methods.
Li et al. (2020)	"Rail Surface Defect Detection Method Based on YOLOv3 Deep Learning Networks"	Adapts YOLOv3 for rail surface inspection, focusing on real-time defect detection with minimal computational overhead.	Achieved a balance between computational efficiency and accuracy, suitable for deployment in resource-constrained environments.
Sharma et al. (2023)	"DeepRail: An Ensemble Learning Framework for Railway Defect Detection"	Introduces an ensemble-based approach using CNN and transfer learning, aiming for more reliable detection by combining multiple model predictions.	Achieved an 8% improvement in accuracy over single CNN models when tested on synthetic and real railway defect datasets.
Wang et al. (2024)	"Transformer-Based Anomaly Detection for High-Speed Railways"	Leverages transformer architecture to detect anomalies in high-speed railway tracks, emphasizing its utility in handling temporal and spatial correlations in the data.	Demonstrated better anomaly localization and detection efficiency compared to recurrent models and CNNs on large-scale high-speed railway data.

Table 1: State of Art

3. METHODOLOGY

- **Dataset Collection and Preprocessing**

The Railway Track Fault Detection dataset, available on Kaggle, is a curated collection aimed at facilitating research in automated rail track anomaly detection. This dataset includes high-quality images of railway tracks captured under various environmental conditions, making it suitable for training and testing machine learning and deep learning models. The dataset contains images depicting both normal and faulty tracks, enabling binary or multi-class classification tasks depending on the specific use case. These images showcase diverse anomalies such as cracks, misalignments, and obstructions, providing a comprehensive resource for developing robust fault detection systems. Whereas the data used in this study were approx. 364 images consisting of 299 images with rail/sleepers cracks/defects and 62 without rail/sleeper cracks or perfect images.

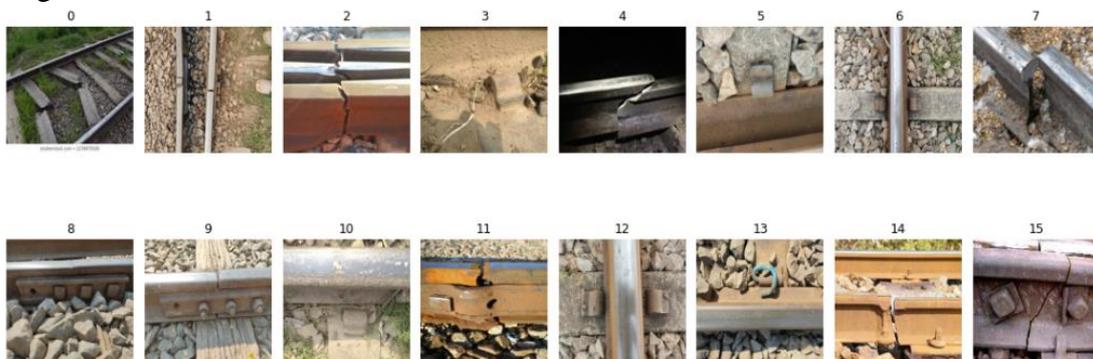


Figure 1: (a) Tracks with Defects



Figure 1: (b) Tracks without Defects

• Work Flow Model

The process begins with the collection of the raw dataset, which includes high-resolution images of railway tracks. This dataset comprises a variety of images depicting normal and faulty track conditions. The raw data serves as the foundation for training and testing machine learning and deep learning models. Ensuring the dataset is diverse and comprehensive is crucial for building a robust anomaly detection system.

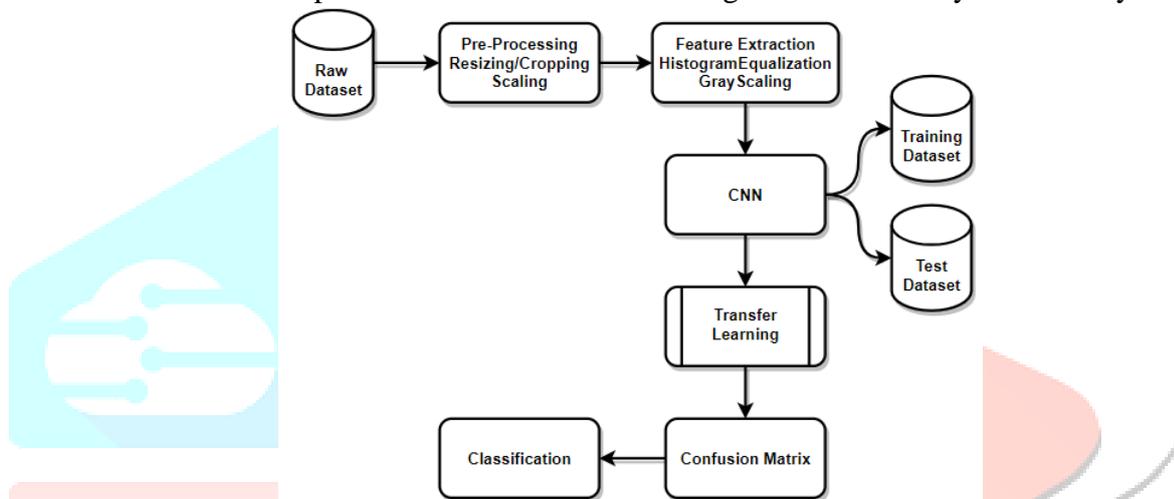


Figure.2: Flowchart of implementation of the models

1. **Pre-Processing (Resizing/Cropping/Scaling):** Pre-processing is an essential step where the raw images are refined to ensure uniformity and enhance model performance. Techniques such as resizing standardize the image dimensions, while cropping removes irrelevant parts of the image. Scaling adjusts the pixel intensity values to a uniform range, aiding in faster and more effective training of the model.
2. **Feature Extraction (Histogram Equalization and Grayscaleing):** Feature extraction focuses on transforming raw image data into meaningful patterns that can be used for classification. Histogram equalization improves the contrast of the images, making anomalies more distinguishable. Grayscaleing reduces the image to a single channel, simplifying computations while retaining essential structural information for defect detection.
3. **Dataset Splitting (Training and Test Datasets):** After pre-processing, the dataset is divided into training and test sets. The training dataset is used to teach the model to identify patterns and anomalies, while the test dataset evaluates the model's performance on unseen data. Proper dataset splitting ensures that the model generalizes well to real-world scenarios.
4. **Convolutional Neural Network (CNN):** The CNN is the backbone of the defect detection system, leveraging convolutional layers to extract spatial features from images. It autonomously identifies patterns such as cracks or irregularities in the railway tracks. The CNN's architecture makes it particularly effective for image-based tasks, ensuring accurate and efficient defect detection.
5. **Transfer Learning:** Transfer learning is employed to enhance the performance of the CNN by leveraging pre-trained models such as ResNet or Inception. These models, trained on large datasets, provide a strong feature extraction capability, reducing the time and computational resources required for training while improving accuracy on smaller datasets.
6. **Confusion Matrix:** The confusion matrix is a key evaluation metric used to measure the classification performance of the model. It provides insights into true positives, true negatives, false positives, and false negatives, allowing for a detailed understanding of the model's strengths and weaknesses.

7. Classification: The final step involves using the trained model to classify input images into normal or faulty track categories. The classification results help in identifying defects for timely maintenance, ensuring safety and reliability in rail operations.

4. RESULTS AND DISCUSSION

- **Data Preprocessing (Resizing):**

The dataset initially contains high-resolution images with dimensions of 4000x3000 pixels, as shown in the provided table. While these large dimensions are suitable for detailed visual inspection, they can be computationally intensive for training machine learning and deep learning models. To optimize the processing pipeline, the images are resized to a standard dimension of 448x448 pixels, balancing computational efficiency with the preservation of critical features. The resizing process involves maintaining the aspect ratio of the images to prevent distortion of key features. Using techniques like bilinear or bicubic interpolation ensures that the resized images retain sufficient detail for accurate classification. This transformation not only reduces the computational load but also ensures compatibility with popular deep learning architectures such as ResNet50, VGG16, and InceptionV3, which often require input dimensions of 224x224, 299x299, or 448x448 pixels. By standardizing the image size, the dataset becomes more manageable, facilitating faster training and enhancing the performance of anomaly detection models.

- **Data Preprocessing (RGB Intensity):**

The RGB intensity distribution is an essential aspect of analyzing images for railway track anomaly detection. Each image is composed of three primary color channels: Red (R), Green (G), and Blue (B). These intensity levels represent the brightness and color distribution in an image, which are critical for detecting patterns and anomalies. In the given examples, the RGB histograms illustrate the pixel intensity levels for each channel across the images.

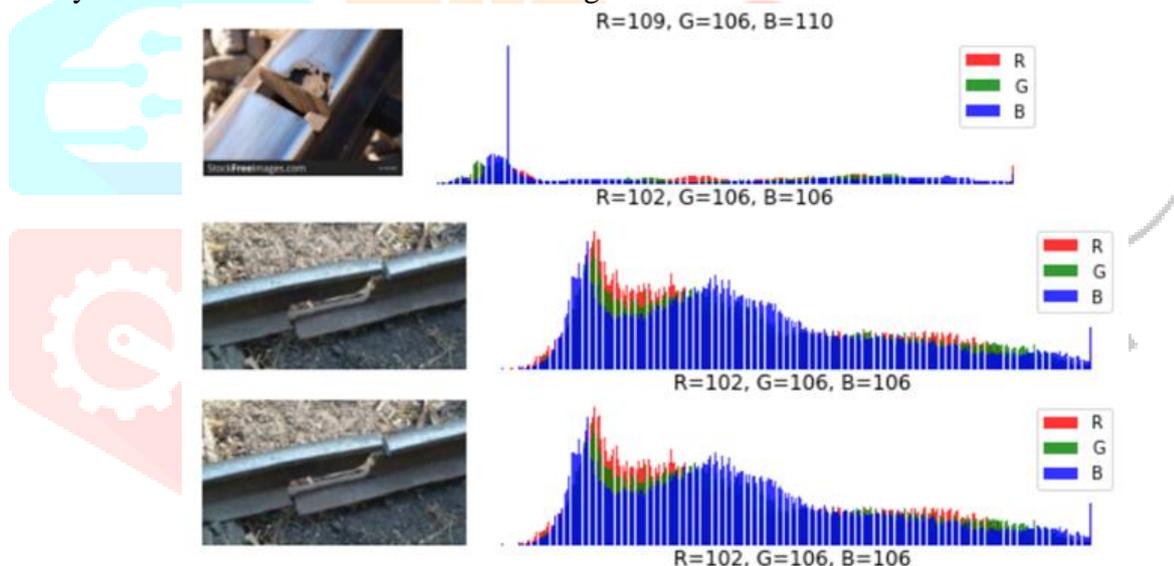


Figure 3: Image Color RGB Intensity

The first image, representing a cracked rail, shows a unique distribution where certain intensity levels spike, indicating prominent features like sharp edges and high-contrast regions. The subsequent images, depicting other defects such as broken rail joints, display more evenly distributed RGB values, reflecting the uniformity of lighting conditions and the texture of the rail surface. This RGB intensity analysis aids in preprocessing tasks like histogram equalization and normalization, ensuring consistent image quality for machine learning and deep learning models. By leveraging these distributions, models can better distinguish between normal and defective tracks.

- **Data Preprocessing (Hue, Saturation, and Value (HSV) Analysis):**

The Hue, Saturation, and Value (HSV) color model is an effective way to analyze and interpret images, especially for anomaly detection in railway tracks. Unlike the RGB model, which focuses on color intensity, HSV separates color information (hue) from the brightness (value) and saturation levels, making it more robust to changes in lighting and shadows.

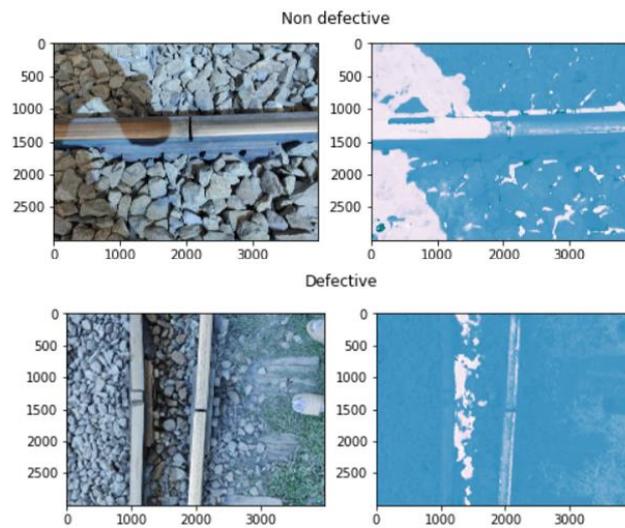


Figure 4: Hue, Saturation, and Value (HSV) Analysis

Hue represents the dominant wavelength of light, distinguishing colors like red, green, or blue. In images of railway tracks, hue variations can help detect color differences caused by rust, cracks, or foreign objects. For example, anomalies like rust may shift the hue towards red-brown tones, aiding defect identification. Saturation measures the intensity or purity of the color. Lower saturation values indicate faded or grayish tones, which might correspond to worn-out or weathered track surfaces. High saturation values often signify freshly painted or well-maintained rails. Value defines the brightness level, capturing the amount of light present in the image. Defects such as cracks or obstructions can manifest as sudden dips in value due to shadows or darker tones compared to the surrounding area. By converting images to the HSV color space and analyzing these components, machine learning and deep learning models can better adapt to environmental variations, such as different lighting conditions or shadows, improving the accuracy of defect detection.

- **Data Preprocessing (Edge Detection Analysis):**

Edge detection is a fundamental technique in image processing, used to identify the boundaries and transitions between different regions in an image. For railway track defect detection, edge detection helps in highlighting structural features like cracks, breaks, and misalignments, which are often indicators of faults. By applying edge detection algorithms as Canny, abrupt changes in pixel intensity are emphasized, making it easier to isolate critical features from the background. For instance, the Canny edge detection method involves gradient computation and thresholding to precisely identify edges while reducing noise. This process ensures the edges of anomalies, such as rail cracks or detached joints, are clearly delineated for further analysis.

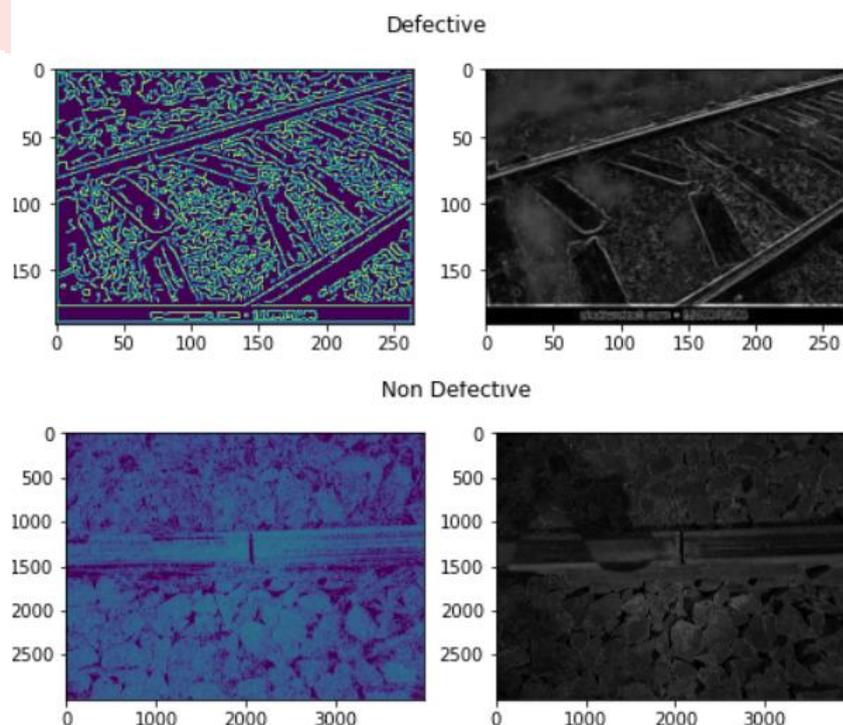


Figure 5: Canny Edge Detection

Accurate edge detection not only simplifies feature extraction for machine learning models but also improves their ability to focus on key regions, leading to better performance in anomaly detection tasks.

- **Split Train, Test and Validate:**

The provided process splits data into categories for machine learning tasks. It organizes images or files into training, validation, and testing datasets based on their labels—"Defective" and "Non-defective." For each category, filenames from their respective directories (e.g., training, validation, testing) are retrieved and stored in data structures for further analysis. After organizing individual sets, the data from the training, validation, and testing splits are combined for each category (defective and non-defective) using a concatenation process. This ensures comprehensive datasets for each class, ready for preprocessing or training purposes in machine learning workflows.

```
train_df_defective = pd.DataFrame(os.listdir(train_path+'Defective'))
val_df_defective = pd.DataFrame(os.listdir(val_path+'Defective'))
test_df_defective = pd.DataFrame(os.listdir(test_path+'Defective'))
train_df_undefective = pd.DataFrame(os.listdir(train_path+'Non defective'))
val_df_undefective = pd.DataFrame(os.listdir(val_path+'Non defective'))
test_df_undefective = pd.DataFrame(os.listdir(test_path+'Non defective'))

defective=pd.concat([train_df_defective,test_df_defective,val_df_defective], axis=0)
undefective=pd.concat([train_df_undefective,test_df_undefective,val_df_undefective], axis=0)
```

Figure 6: Train Split Validate

- **Model VGG16**

The VGG16 algorithm is a convolutional neural network (CNN) architecture designed for image classification tasks. It consists of 16 layers with learnable parameters, including 13 convolutional layers and 3 fully connected layers. The network uses small 3x3 filters for the convolutional layers, applied with a stride of 1 and padding to maintain spatial dimensions. Rectified Linear Unit (ReLU) activation functions are applied to introduce non-linearity.

Pooling layers, specifically max pooling with a 2x2 window and a stride of 2, reduce the spatial dimensions, making the computation more efficient while retaining essential features. The fully connected layers towards the end of the architecture consolidate learned features for classification into different categories. VGG16 typically uses a softmax activation function in the final layer for multi-class classification. Known for its simplicity and uniform architecture, VGG16 has been widely used for transfer learning due to its robust feature extraction capabilities.

```
train_generator_vgg = vgg_datagen.flow_from_directory(
    directory="../input/railway-track-fault-detection/Railway Track fault Detection Updated/Train",
    target_size=(448, 448),
    color_mode="rgb",
    batch_size=32,
    class_mode="binary",
    shuffle=True,
    seed=42,
```

Figure 7: Model VGG16

The provided process utilizes a pre-trained VGG16 model for image classification tasks. The `flow_from_directory` method from a data generator is used to load and preprocess the training data. The dataset is stored in a specified directory and organized into subdirectories based on class labels. Images are resized to a target dimension of 448x448 pixels, with RGB color mode for compatibility with the VGG16 architecture. Batch processing is set with a batch size of 32 to streamline training, and the `class_mode` is binary, indicating a two-class classification problem. The data is shuffled to prevent overfitting due to order bias, and a seed is defined to ensure reproducibility of the randomization. This setup is optimal for leveraging the transfer learning capabilities of VGG16 while preparing the data in a structured and efficient manner.

- **Model InceptionV3**

The InceptionV3 model is a convolutional neural network (CNN) architecture designed for image classification and recognition tasks. It is an advanced version of the Inception family of networks, incorporating techniques like factorized convolutions, asymmetric convolutions, and dimensionality reduction for improved performance and efficiency. The architecture uses "Inception modules," which allow the network to capture features at multiple scales by applying convolutions with different filter sizes (e.g., 1x1, 3x3, 5x5) and combining their outputs. This approach enables the model to extract

complex features without significantly increasing computational cost. To further optimize the network, it employs techniques such as batch normalization for faster convergence and better regularization, as well as auxiliary classifiers to improve gradient flow and prevent overfitting.

InceptionV3 also includes global average pooling instead of fully connected layers at the end of the network, reducing the total number of parameters while maintaining performance. It is widely used in transfer learning due to its efficiency and strong feature extraction capabilities.

```
inception_v3_train = inception_v3_datagen.flow_from_directory(
    directory="../input/railway-track-fault-detection/Railway Track fault Detection Updated/Train",
    target_size=(448, 448),
    color_mode="rgb",
    batch_size=32,
    class_mode="binary",
    shuffle=True,
    seed=42,
```

Figure 8: Model InceptionV3

The process depicted uses the InceptionV3 model to prepare data for training. The `flow_from_directory` function from a data generator is utilized to load images from a specified directory containing the training data, where files are organized into subfolders representing class labels. The images are resized to a uniform dimension of 448x448 pixels to ensure compatibility with the input requirements of the InceptionV3 model, and the color mode is set to RGB to handle color images.

A batch size of 32 is used to process the data in smaller groups, optimizing memory usage and training efficiency. The `class_mode` is set to binary, indicating a two-class classification task. The data is shuffled to maintain randomness in training and avoid overfitting due to any inherent order in the dataset. Additionally, a seed value is specified to ensure that the shuffling process is reproducible, providing consistency in training across runs. This setup effectively prepares the data for training with the InceptionV3 architecture

- **Model ResNet50**

ResNet50, short for Residual Network with 50 layers, is a deep convolutional neural network that addresses the challenge of vanishing gradients during backpropagation, enabling the construction of significantly deeper networks. However, ResNet50 is characterized by its use of residual learning through skip connections. These connections allow the model to learn residual functions by bypassing one or more layers, ensuring that critical information is retained and gradients flow effectively through the network.

The ResNet50 architecture is composed of five stages, each containing a series of convolutional and identity blocks. These blocks are designed with 1x1, 3x3, and 1x1 convolutional layers, where the 1x1 layers are responsible for reducing and restoring dimensions, while the 3x3 layers perform the core convolutional operations. A fully connected layer follows the final stage, producing the output predictions. With its 23 million parameters, ResNet50 strikes a balance between computational efficiency and model depth, making it highly effective for image classification, object detection, and feature extraction tasks in diverse applications.

```
resnet50_train = resnet50_datagen.flow_from_directory(
    directory="../input/railway-track-fault-detection/Railway Track fault Detection Updated/Train"
    target_size=(448, 448),
    color_mode="rgb",
    batch_size=32,
    class_mode="binary",
    shuffle=True,
    seed=42,
)
```

Figure 9: Model ResNet50

The above provided snippet initializes a data generator for training a ResNet50 model, using the `flow_from_directory` method to load images directly from a specified directory. This function is set up with the target directory containing the training dataset for railway track fault detection. The images are resized to a target size of 448x448 pixels, ensuring compatibility with the ResNet50 input dimensions. The color mode is set to "rgb" for processing colored images, and the batch size is defined as 32 to enable efficient training. A binary classification mode is specified, indicating the task involves

distinguishing between two classes, likely fault and non-fault. The data is shuffled for randomization, with a seed value of 42 ensuring reproducibility in the training process.

- **Evaluation Results of the Models**

• **VGG16 Model Performance**

The VGG16 model achieved the highest accuracy among the three architectures, recording 91.64% accuracy at epoch 28.

The average accuracy throughout the training process was 88.29%, demonstrating consistent performance.

Its relatively shallow architecture compared to deeper models proved effective due to its simplicity and efficient feature extraction capability.

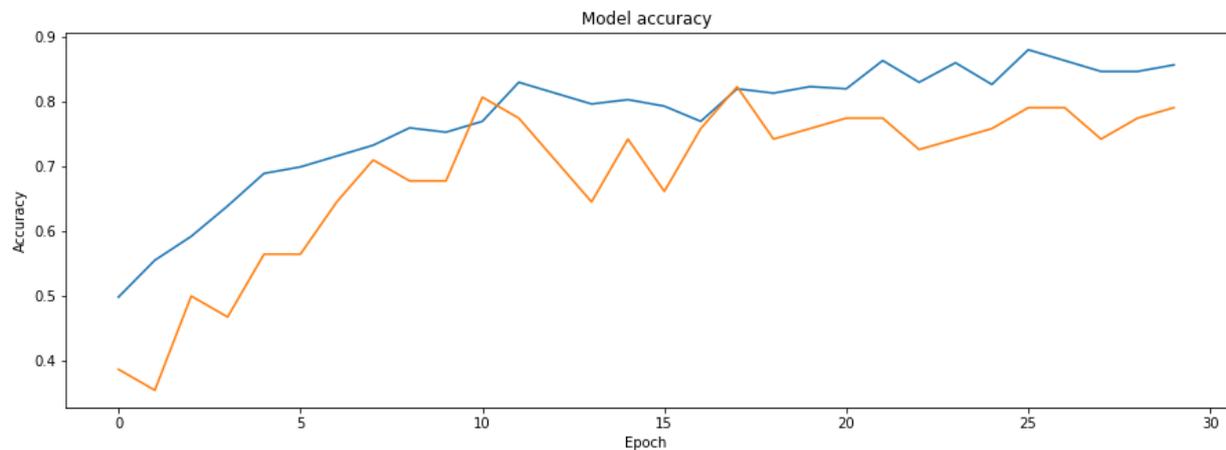


Figure 10: Model Training and Validation using VGG16

• **ResNet50 Model Performance**

The ResNet50 model achieved an accuracy of 89.46%, slightly lower than VGG16 but still competitive. Its depth, aided by residual learning, facilitated better gradient flow and robust performance, especially in handling complex data distributions.

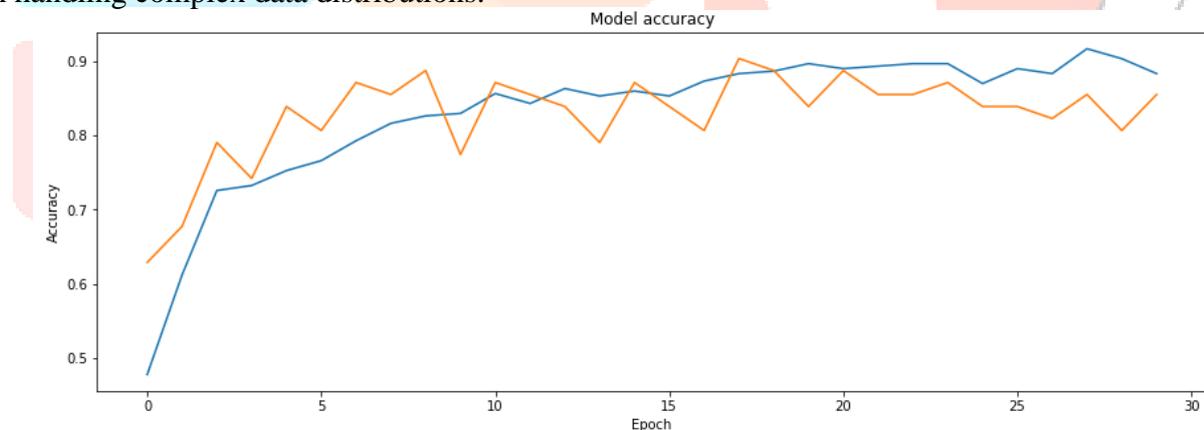


Figure 11: Model Training and Validation using ResNet50

• **InceptionV3 Model Performance**

The InceptionV3 model recorded an accuracy of 81.26%, which was lower than both VGG16 and ResNet50. The architecture's computational complexity may have contributed to its comparatively reduced accuracy, especially with a smaller batch size and unbalanced data.

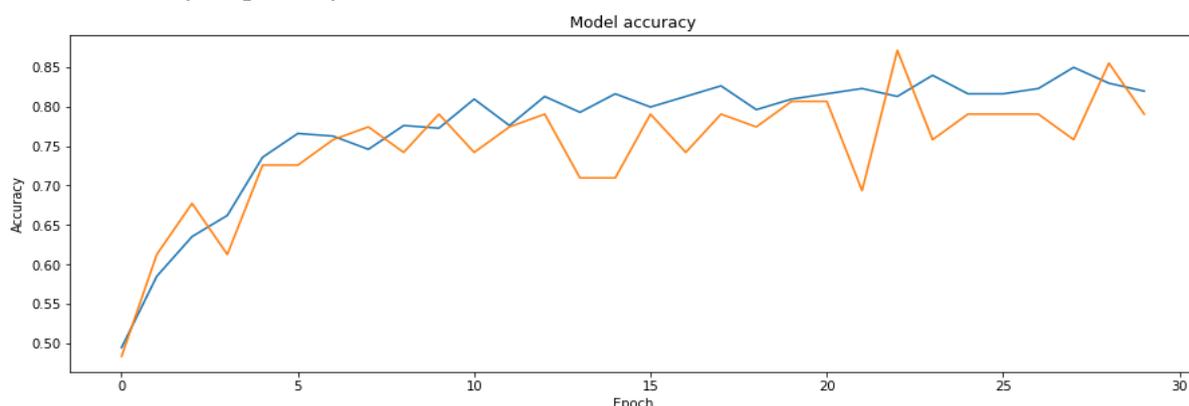


Figure 12: Model Training and Validation using InceptionV3

- *Impact of Model Depth*

The results highlight that deeper models generally perform better due to their enhanced feature extraction capabilities. However, in this case, VGG16 outperformed the deeper ResNet50 and InceptionV3 architectures. This is attributed to the balanced evaluation type and efficient pre-processing, which mitigated the challenges posed by the unbalanced dataset.

- *Handling of Unbalanced Classes*

All three models were able to address the issue of unbalanced classes effectively. Pre-processing and feature extraction techniques, coupled with adjustments in batch size, reduced misclassification rates even for classes with limited data.

Although a smaller batch size was used to manage computational costs, the depth of the networks compensated for this reduction, ensuring accurate classification results.

- *Comparative Analysis of The Three Model*

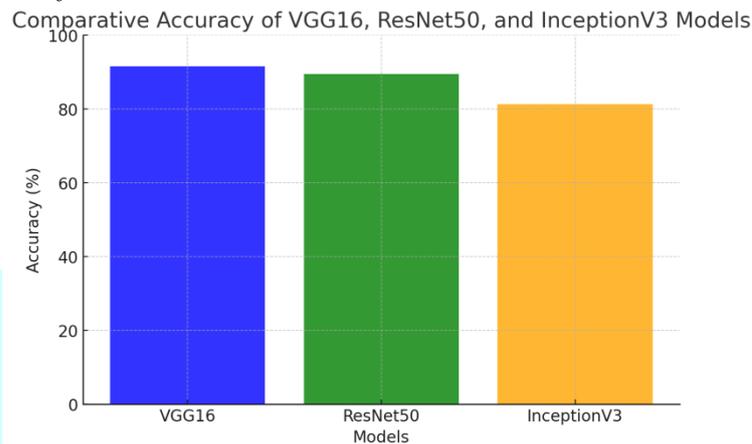


Figure 13: Comparative Analysis

The comparative analysis of the three models—VGG16, ResNet50, and InceptionV3—is visually depicted in the graph above. Among these, VGG16 emerged as the top-performing model, achieving an accuracy of 91.64%, followed closely by ResNet50 with an accuracy of 89.46%. On the other hand, InceptionV3 achieved a comparatively lower accuracy of 81.26%. The results highlight the effectiveness of the VGG16 model in this specific application, likely due to its efficient feature extraction and simpler architecture. ResNet50's depth provided a competitive edge, although slightly less effective than VGG16. InceptionV3, despite its complex architecture, faced challenges in maintaining high accuracy, possibly due to the unbalanced dataset and computational limitations.

5. CONCLUSION AND FUTURE SCOPE

The evaluation of the VGG16, ResNet50, and InceptionV3 architectures on rail track defect detection demonstrates their capability in addressing classification tasks effectively. Among the models, VGG16 achieved the highest accuracy, showcasing its strength in feature extraction and suitability for the given dataset. ResNet50 also performed competitively, leveraging its residual connections to handle deeper networks effectively. Although InceptionV3 showed comparatively lower accuracy, it still provided valuable insights into handling diverse data distributions. The balanced evaluation process and pre-processing techniques helped mitigate the challenges posed by the unbalanced dataset, resulting in reduced misclassifications and improved performance.

- *Future Scope*

1. **Enhanced Data Augmentation:** Implementing advanced data augmentation techniques can further balance the dataset and improve the generalization of the models.
2. **Hybrid Models:** Exploring hybrid architectures that combine the strengths of VGG16, ResNet50, and InceptionV3 could lead to better performance.
3. **Transfer Learning:** Fine-tuning pre-trained models with more domain-specific data could improve accuracy for specialized tasks.
4. **Optimization Techniques:** Utilizing modern optimization methods such as adaptive learning rates or advanced regularization techniques can enhance training efficiency and model performance.
5. **Edge Deployment:** These models can be optimized for deployment on edge devices, enabling real-time fault detection in resource-constrained environments.
6. **Cross-Domain Applications:** The methodologies and insights gained from this study can be extended to other domains, such as defect detection in industrial systems or anomaly detection in medical imaging.

REFERENCES

- [1] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017. doi: 10.1109/TPAMI.2016.2644615.
- [2] C. Breitenberger, et al., "Automated visual inspection of railway tracks using CNNs," *International Journal of Transportation Systems*, 2018.
- [3] L. Chen, Y. Xu, and P. Liu, "Semantic segmentation for rail crack detection using U-Net with attention mechanisms," *Expert Systems with Applications*, vol. 174, p. 114746, 2021. doi: 10.1016/j.eswa.2021.114746.
- [4] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, ... and Y. Bengio, "Generative adversarial networks," *arXiv preprint arXiv:1406.2661*, 2014.
- [5] Z. He, H. Jia, and Y. Liu, "A deep learning-based approach for rail surface defect detection using computer vision," *Transportation Research Part C: Emerging Technologies*, vol. 114, pp. 59–72, 2020. doi: 10.1016/j.trc.2020.03.006.
- [6] A. Hinton, et al., "A deep convolutional neural network for detection of rail surface defect," *IEEE Xplore*, 2019. [Online]. Available: <https://ieeexplore.ieee.org/document/8907845>.
- [7] P. Kumar, et al., "Hybrid CNN-RNN for rail track anomaly detection," *Journal of Intelligent Transportation Systems*, 2022.
- [8] J. Lee and S. Park, "Deep learning for rail track condition monitoring: A review," *Open Access Publishing*, 2024. [Online]. Available: OAE Publish.
- [9] F. Li, et al., "Rail surface defect detection method based on YOLOv3 deep learning networks," *IEEE Xplore*, 2020. [Online]. Available: <https://ieeexplore.ieee.org/document/9160464>.
- [10] A. Sharma, et al., "DeepRail: An ensemble learning framework for railway defect detection," *Journal of Intelligent Transportation Systems*, 2023.
- [11] Z. Wang, et al., "Transformer-based anomaly detection for high-speed railways," *Transportation Safety and Security Journal*, 2024.
- [12] W. Zhang, et al., "TrackNet: A deep learning-based fault detection for railway track inspection," *IEEE Xplore*, 2019. [Online]. Available: <https://ieeexplore.ieee.org/document/8639659>.
- [13] Z. Zhang, et al., "Surface defect detection on rail tracks via deep learning," *IEEE Transactions on Neural Networks and Learning Systems*, 2021. [Online]. Available: <https://ieeexplore.ieee.org/document/9350134>.
- [14] Z. Zhang, K. Yang, and Z. Cui, "Rail surface defect detection based on deep learning using hybrid convolutional and recurrent neural networks," *Journal of Rail and Rapid Transit*, vol. 233, no. 2, pp. 179–191, 2019. doi: 10.1177/0954409718784217.
- [15] Ji, A.; Woo, W. L.; Wong, E. W. L.; Quek, Y. T. Rail track condition monitoring: a review on deep learning approaches. *Intell. Robot.* 2021, 1, 151-75. <http://dx.doi.org/10.20517/ir.2021.14>