



## AI Ladder-Based Intelligent Enterprise Resource Planning for Predictive Railway Track Maintenance from UAV Imagery

Jakfat Haekal<sup>1\*</sup>, Rizaldi Mu'min<sup>2</sup>, Andi Adriansyah<sup>3</sup>, Paduloh<sup>4</sup>, Didin Sjarifudin<sup>4</sup>, Arif Nuryono<sup>4</sup>, Rifda Ilahy Rosihan<sup>4</sup>, Erwin Barita Maniur Tambunan<sup>4</sup>, Siti Noor Kamariah Yaakop<sup>5</sup>

<sup>1</sup> Department of Industrial Engineering, Universitas Esa Unggul, Jakarta 11560, Indonesia

<sup>2</sup> Department of Management, Universitas Negeri Jakarta, Jakarta 13220, Indonesia

<sup>3</sup> Department of Electrical Engineering, Universitas Mercu Buana, Jakarta 34788, Indonesia

<sup>4</sup> Department of Industrial Engineering, Universitas Bhayangkara Jakarta Raya, Jakarta 12140, Indonesia

<sup>5</sup> Teknoputra Section, Universiti Kuala Lumpur, MIMET, Perak 32200, Malaysia

Corresponding Author Email: [jakfat.haekal@esaunggul.ac.id](mailto:jakfat.haekal@esaunggul.ac.id)

Copyright: ©2025 The authors. This article is published by IETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/mmep.121023>

### ABSTRACT

**Received:** 28 August 2025

**Revised:** 6 October 2025

**Accepted:** 11 October 2025

**Available online:** 31 October 2025

#### Keywords:

UAVs, machine learning, ERP, AI Ladder, predictive maintenance, railway infrastructure

Railway safety relies on the early detection of track defects that can lead to derailments and service disruptions. Traditional inspections are labor-intensive and error-prone, whereas many vision-based studies only focus on detection and fail to link predictions to maintenance execution. This study addresses this perception-to-action gap. In the Collect phase, unmanned aerial vehicles (UAVs) acquire high-resolution images of track segments. In the Organize phase, images are standardized, binary-masked to generate ground truth, and embedded into fixed-length feature vectors. In the Analyze phase, four classifiers including Artificial Neural Networks (ANNs), Support Vector Machines (SVMs), Random Forest (RF), and K-Nearest Neighbors (KNNs) are compared using Area Under the Curve (AUC), accuracy, F1-score, and Matthews Correlation Coefficient (MCC). In the Infuse phase, the optimal model is integrated into an enterprise resource planning (ERP) maintenance module to support real-time defect flagging, automated work orders, and dashboard visualization. ANN model achieves the highest performance (AUC = 0.935; accuracy = 0.884; F1-score = 0.884; MCC = 0.768). The AI Ladder-guided machine-learning (ML)-ERP pipeline demonstrates a practical pathway from aerial sensing to actionable maintenance, aligning with Sustainable Development Goals (SDGs) 9. By directly embedding classification into ERP workflows, operators can transition from periodic, manual inspections to continuous, predictive maintenance, featuring automated scheduling, notifications, and auditable condition histories.

## 1. INTRODUCTION

The railway industry is a vital component of Indonesia's transportation infrastructure, particularly on Java and Sumatra. Investments in both super-and sub-structures, including ballast, slab track, and subgrade, are essential to maintaining operational reliability and long-term capacity of the national rail system [1]. Recent developments, such as the inauguration of Southeast Asia's first high-speed railway connecting Jakarta and Bandung, underscore Indonesia's commitment for the adoption of advanced engineering standards for track construction and system performance [2, 3]. Despite these advancements, safety remains a pressing concern. Historical incidents, highlight persistent vulnerabilities in Indonesia's railway operations. A total of 143 train accidents occurred in Indonesia between 2015 and 2021, causing 132 fatalities and nearly 300 injuries [4], highlighting the continuing instability of railway safety performance. Earlier, around 2004 to 2010, over 700 railway accidents were reported, mostly involving derailments and vehicle collisions. Although train-to-train

crashes made up only about 5% of the total, road-related collisions accounted for nearly 20%, resulting in more than 360 deaths and 1,200 injuries [5]. It reflects long-standing structural and operational safety deficiencies in the national rail system.

Similar incidents have also occurred globally, further underscoring the universal nature of railway safety challenges. Studies have shown that fatalities and injuries among track maintenance workers often occur under high-risk working conditions (especially at night) and can be exacerbated by communication failures and insufficient safety coordination [6]. In South Korea, 240 human errors were found on Korean Railway system [7]. In the United Kingdom, Network Rail was fined £3.75 million after two maintenance workers were fatally struck by a train in Margam, Wales; the investigation revealed systemic safety failures and led to significant operational reforms [8].

Railway defects are a critical issue in transportation safety [9], as they can lead to derailments, service disruptions, and costly repairs [10]. Traditional inspection methods are often

inefficient and prone to human error [11]. In the modern context, human safety and industrial efficiency are global concerns, as emphasized in the Sustainable Development Goals (SDGs). In particular, SDG 9 (Industry, Innovation and Infrastructure) underlines the importance of resilient infrastructure and technological innovation. In the railway sector, this translates into maintaining advanced, safe, and sustainable transportation systems.

Recent studies have explored visual inspection methods for railway defect detection using image processing and computer vision. Zhuang et al. [12] developed an automated visual inspection system using a cascading classifier ensemble trained with LogitBoost to identify cracks and deformations. Infrared thermography (IRT) has also been applied to detect hidden cracks, although it requires specialized equipment and is sensitive to environmental conditions [13]. Classical techniques such as canny edge detection and the Hough transform face challenges in recognizing complex or novel defect types due to reliance on handcrafted features [14]. Collectively, these studies demonstrate promise but also reveal limitations in generalizability, real-time applicability, and integration with maintenance decision-making systems.

To support intelligent and scalable infrastructure maintenance, this study adopts the AI Ladder Model comprising four stages: Collect, Organize, Analyze, and Infuse as a guiding framework for embedding AI into operational workflows [15]. In this framework, unmanned aerial vehicles (UAVs) provide real-time data acquisition (Collect); feature extraction, annotation, and preprocessing tools prepare and structure image data (Organize); machine-learning (ML) algorithms enable pattern recognition and classification (Analyze); and outputs are translated into actionable maintenance strategies (Infuse). This structure ensures end-to-end alignment between data, analytics, and infrastructure decisions, while supporting SDG 9 through digital innovation. Building on this framework, we evaluate four ML algorithms: Artificial Neural Networks (ANNs), Support Vector Machines (SVMs), Random Forest (RF), and K-Nearest Neighbors (KNNs) for railway defect detection. RF is recognized for handling high-dimensional data and mitigating overfitting [16]; KNN is effective for similarity-based classification [17]; ANN excels at learning nonlinear patterns and extracting complex features [18]; and SVM performs well with limited datasets and high-dimensional feature spaces [19]. We further incorporate a deep-learning-based feature embedding to enhance performance and generalizability across diverse track conditions, with all classification outcomes expressed consistently in defective (bad) /non-defective (good) terms.

Unlike most railway inspection studies that do not frame their approach within a systematic framework, this work explicitly employs the AI Ladder (Collect-Organize-Analyze-Infuse) [15] as an end-to-end backbone: In the Collect phase, we use UAVs for scalable and high-frequency RGB image acquisition; The Organize phase arranges annotation/normalization and extracts deep feature embeddings with standardized outputs of good/bad across track conditions; The Analyze phase conducts a controlled comparison of four models (ANN, SVM, RF, KNN) based on AUC, accuracy, F1-score, and MCC, rather than merely reporting a single accuracy number [16, 19]; and the Infuse phase links predictions to enterprise resource planning (ERP)

to trigger work orders, scheduling, and resource prioritization, components that are generally absent in the literature yet crucial to bridge the perception-to-action gap [20]. Thus, our uniqueness lies not only in UAVs, deep embeddings, and multi-model comparison, but especially in the use of the AI Ladder as an operational framework that ensures analytical findings culminate in measurable maintenance impact.

The integration of predictive-maintenance outputs into ERP systems, particularly maintenance modules, is crucial for operational reliability and decision-making in infrastructure-intensive sectors. Prior work shows that combining ML with SAP plant maintenance can improve maintenance scheduling and failure prediction [21], ERP-based modules reduce non-value-added tasks and streamline workflows [22], and integrating Reliability-Centered Maintenance (RCM) with Computerized Maintenance Management Systems (CMMS) improves metrics such as Mean Time Between Failures (MTBF) and Mean Time to Repair (MTTR) [23]. A review further highlights that intelligent ERP optimization using ML improves failure forecasting and resource allocation for preventive and corrective actions [24].

Despite the progress in perception, recent surveys note that most UAV studies stop at reporting detection accuracy without demonstrating how predictions trigger scheduled maintenance in enterprise systems. The absence of end-to-end pipelines that connect computer-vision outputs to asset-management workflows (ERP) remains a key barrier to operational adoption [20]. While prior studies demonstrate effective visual inspection techniques and promising ML-based approaches [12], three key gaps persist:

- (i) limited use of UAVs for scalable, high-frequency data collection directly tied to automated defective/non-defective classification;
- (ii) insufficient generalization across varied track conditions without deep feature embeddings;
- (iii) weak integration between defect-detection outputs and ERP maintenance modules for actionable scheduling and resource optimization [20-22].

This study proposes a comprehensive UAV-based predictive-maintenance framework that integrates UAV data acquisition, deep feature embedding, and comparative evaluation of ANN, SVM, RF, and KNN within the AI Ladder workflow, and infuses predictive insights into ERP maintenance modules.

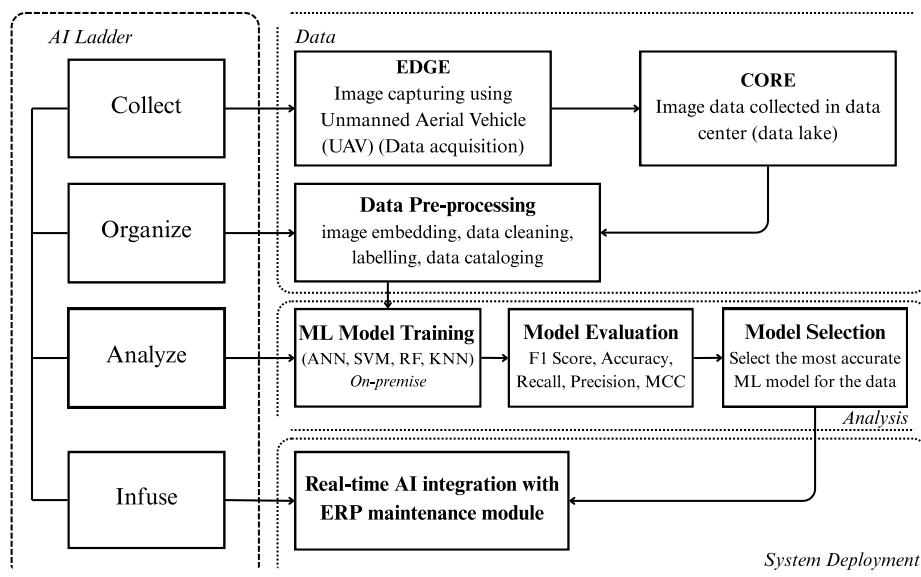
This work makes three key contributions that explicitly bridge the perception-to-action gap. First, it advances the theoretical/methodological frontier by introducing an end-to-end, layered analytics pipeline that couples UAV-captured imagery with deep-learning-based feature embeddings and multiple ML classifiers, enabling robust defective (bad)/non-defective (good) classification across diverse track conditions. Second, it delivers operational system integration by infusing classifier outputs into ERP maintenance modules, converting predictions into scheduled actions automated work orders, prioritization, and optimized resource allocation consistent with prior evidence on ERP-enabled reliability improvements [20, 22]. Third, it provides practical impact through a scalable, future-ready approach that increases inspection frequency and accuracy while reducing reliance on labor-intensive methods, thereby advancing SDG 9 objectives. Collectively, these contributions close the loop from UAV-based perception to enterprise maintenance execution.

## 2. METHODS AND MATERIALS

### 2.1 AI Ladder framework

The proposed framework, grounded in International Business Machines (IBM)'s AI Ladder methodology, provides

a comprehensive and structured approach for implementing machine learning-based railway defect detection using UAVs imagery [25]. Each stage of the AI Ladder: Collect, Organize, Analyze, and Infuse [15], is carefully adapted to reflect the technical and operational realities of railway infrastructure monitoring, as presented below in Figure 1.



**Figure 1.** New scheme for UAV-based railway track on ERP maintenance module using machine learning within the AI Ladder Model framework

In the Collect phase, data acquisition is performed at the edge using UAVs, enabling the capture of high-resolution imagery directly from the railway environment [26]. The captured images are then transmitted to the core infrastructure (data center), where the organize stage takes place. Here, data undergoes preprocessing activities including image embedding, cleaning, labeling, and cataloguing [27], ensuring that the dataset is structured, consistent, and ready for analysis. During the Analyze phase, the framework supports on-premise training of various machine learning models [28]. The training process is guided by a rigorous evaluation pipeline using performance metrics [29]. This enables objective model selection and ensures that the chosen model is well-suited to the characteristics of the dataset. While infuse phase focuses on real-world deployment, wherein the selected model is integrated into an ERP-based railway maintenance module. This integration supports real-time classification of track conditions and provides actionable insights directly to maintenance teams, enhancing operational efficiency and predictive maintenance capabilities.

### 2.2 Phase 1: Collect

Collect phase is the first rung of the AI Ladder. It is a foundational role for collecting and acquiring high-quality and relevant data. It is intended to build essential data management capabilities, ideally by simplifying data access and ensuring availability, regardless of its format or storage location [30]. Within this context, the proposed research introduces a railway inspection system that use UAVs to systematically capture real-time, high-resolution imagery of railway tracks.

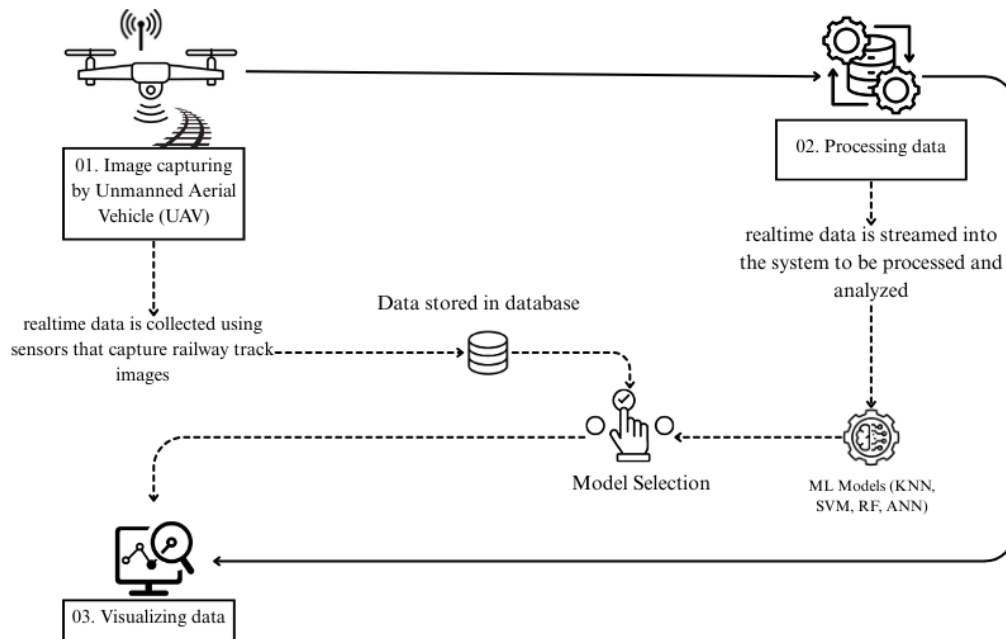
#### 2.2.1 UAV acquired data-based model

To address the growing demand for intelligent and automated infrastructure monitoring, particularly in the

railway sector, this research propose a novel model of UAV acquired data-based. This model encourages recent advancements in Internet of Things (IoT) technology, and machine learning to enable advanced inspection of railway tracks [31]. The system integrates UAV, centralized data processing units, and predictive analytics tools to ensure early detection of defects and proactive maintenance decision-making. Figure 2 illustrates the overall workflow of the UAV acquiring data mechanism, outlining the major components and the flow of data from image acquisition to visualization and analysis.

#### 2.2.2 UAV systematic coverage

Systematic coverage aims to capture imagery of every sleeper, every fastener, and every meter of the rail surface within the target zone. Flight plan is conducted; it incorporates a specific percentage of forward overlap (70%) and side overlap (40%). This overlap ensures that the edges of one image are covered by the next, guaranteeing no gaps. This flight route is designed as a 'flying corridor' that precisely follows the alignment of the railway track, with a constant flying altitude of 15 meters above the rails. This parameter was chosen to obtain the optimal Ground Sampling Distance (GSD), thus, sharp images are provided with efficient area coverage. Moving at a steady speed of approximately 40 km/h, UAV's camera operates continuously. It systematically captures thousands of digital images, photographing every sleeper, every set of fasteners, and every inch of the rail surface. The gimbal system ensures the camera remains focused on the target despite light winds. This is a non-contact inspection process, conducted at the 'edge' or directly in the field environment. Once an area is covered, the UAV performs an initial data upload via a 4G/5G connection to the central data lake ('core').



**Figure 2.** UAV acquiring data mechanism

### 2.2.3 Data transmission to core

In this research, an IoT-enabled UAV is proposed to use wireless transmission, to efficiently send real-time images from the 'edge' to the 'core' infrastructure. For storage, this scenario locates the data in the operator's own data center for control and security. The research suggests using a data lake as the primary repository for raw images due to its ability to handle large volumes of unstructured data, and may be supported by additional databases to manage metadata, ensuring data can be accessed, organized, and analyzed efficiently.

The data acquisition (the 'Collect' phase) is performed at the 'edge', directly within the railway environment. These UAVs are envisioned to act as intelligent, mobile data acquisition points. The raw images captured that represents the real-time visual state of the tracks, are then transmitted from the UAVs to the 'core' infrastructure, typically a central database or data center (data lake).

The proposed railway track inspection system uses a closed-loop architecture with three main components: image capturing, data processing, and data visualization. IoT-enabled UAVs with high-resolution cameras capture real-time images of the tracks for automated, non-contact inspection.

The study collected 201 total images of railway tracks using UAVs, consisting of 52% non-defective (good) and 48% Nb defective (bad) samples. These images are transmitted to a central database and processing system, where advanced image processing and machine learning models (KNN, SVM, RF, ANN) detect and classify defects. The most suitable model is selected based on data characteristics and accuracy needs. Finally, processed data is visualized to support timely interventions and long-term maintenance planning.

## 2.3 Phase 2: Organize

Organize phase is preparing and structuring raw inputs into a consistent and usable format for downstream analysis. It involves standardizing data workflows, ensuring quality and coherence, and enabling traceability across the machine learning pipeline. It emphasizes transforming raw inputs into a well-structured and accessible format suitable and trusted for

further analysis [30]. In the context of this study, organize phase encompasses several key activities including image preprocessing, binary mask generation for ground truth labeling, feature embedding, and structured dataset formation.

### 2.3.1 Image processing and labeling using binary masks

The dataset has undergone a feature extraction and embedding process, converting rail track images into numerical representations. Initially, the dataset consisted of raw images categorized as either non-defective (good) or defective (bad) based on track conditions. After processing, each image was transformed into a structured numerical format, allowing machine learning models to interpret the images mathematically [32], with numerical patterns, computer can observe image distinctive characteristics in detail.

Raw images must be standardized in terms of resolution and format. Each railway track image is resized to a fixed width of 160 pixels, with the height varying (from 333 to 427 pixels) to preserve aspect ratio. Images are also saved in lossless formats such as PNG or BMP.

Each image captured by the UAV is annotated using a corresponding binary mask, an image of the same dimensions consisting solely of black (0) and white (1) pixel values. White pixels indicate regions where visible rail defects such as cracks, deformations, or wear are present, while black pixels represent defect-free areas. They serve as the ground truth labels during model training [33], enabling the machine learning algorithms to learn the visual patterns associated with defective and non-defective track segments.

### 2.3.2 Feature embedding and dataset quality validation

Following the labeling process, each image undergoes feature embedding using a pre-trained deep learning model to numerical representations. This embedding process converts each image into a fixed-length feature vector comprising numerical attributes that capture relevant visual characteristics, such as texture, edges, and structural patterns. Importantly, to prevent data leakage, the embedding process is performed separately for the training and testing subsets after the dataset is split [34]. The dataset was stratified into 80%

training and 20% testing, ensuring class proportions are preserved. These features are then compiled into a structured dataset for model training and validation. Exploratory data analysis (EDA), including distance-based clustering and scatter plots, is conducted to assess class separability and validate dataset quality before feeding it into classification models.

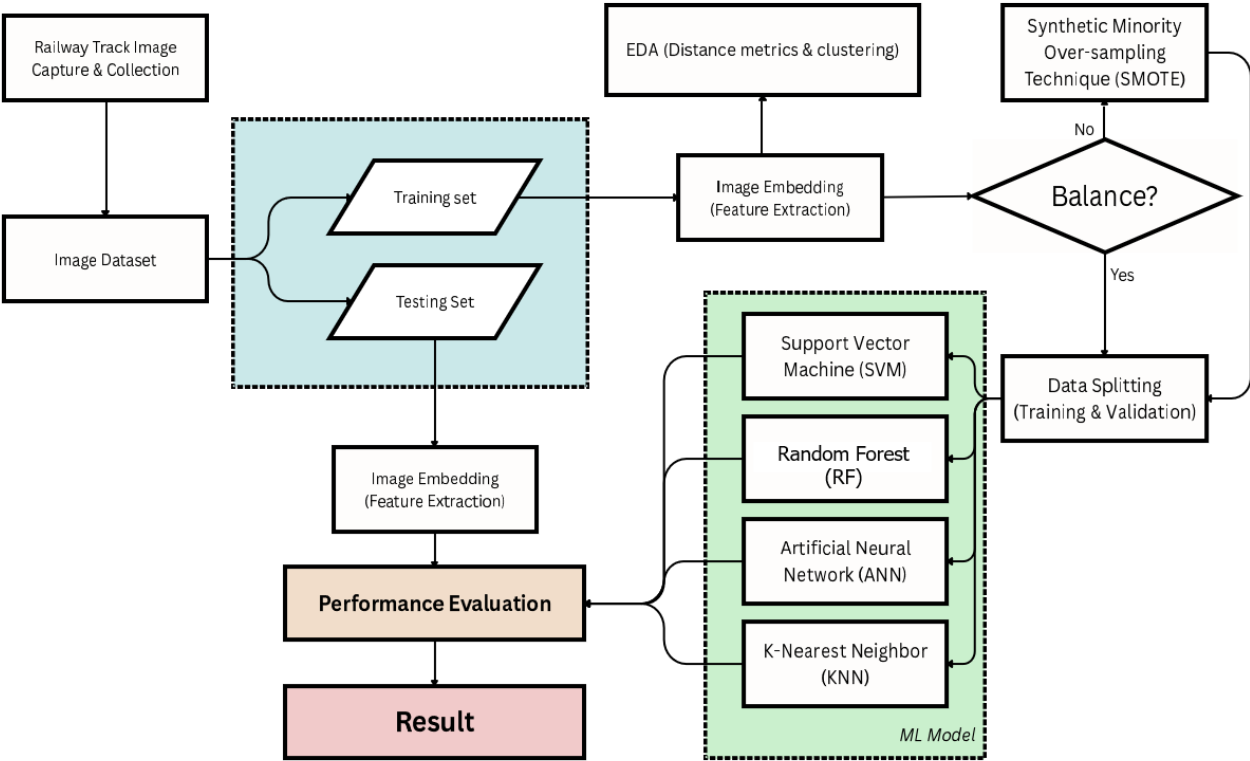
### 2.4 Phase 3: Analyze

Analyze phase is a critical rung in AI Ladder and has a significant role on transforming raw data into actionable insights through advanced analytics and machine learning. It involves not only building predictive models but also understanding their performance and impact to support

informed decision-making and continuous improvement [15]. Aligning with this framework, the study focuses on conducting a comparative evaluation of four prominent machine learning algorithms.

#### 2.4.1 Machine learning model for image-based railways

To systematically evaluate machine learning models for railway track defect detection, a structured research framework was developed to guide the entire process from data acquisition to performance evaluation. This framework ensures a coherent pipeline that integrates image processing, feature extraction, exploratory data analysis, classification, and validation, providing a reproducible and scalable methodology for intelligent defect detection systems. It is illustrated in Figure 3.



**Figure 3.** Framework of machine learning model for image-based railways

The process begins with the capture and collection of railway track images using UAVs, which are compiled into an image dataset representing both defective and non-defective rail segments. This dataset is then divided into training and testing sets. Image embedding is conducted to convert raw images into structured numerical representations. These numerical vectors serve as inputs for subsequent analysis. The embedded data is further processed through EDA using distance-based metrics and hierarchical clustering to assess the natural separability of the dataset and guide model selection and validation. If exploratory checks indicate class imbalance between non-defective and defective images (e.g., skewed class ratios or degraded minority-class recall), the Synthetic Minority Over-sampling Technique (SMOTE) will be applied only to the training partitions [35]. SMOTE generates synthetic minority samples by interpolating between nearest neighbors, creating plausible examples that enrich decision boundaries without simply duplicating data. If no imbalance is detected, SMOTE is skipped and the original class distribution is retained.

The dataset is then split into training and validation sets, which are utilized to train and fine-tune four distinct machine learning classifiers: SVM, RF, ANN, and KNN. These models are systematically evaluated through various performance metrics to determine their effectiveness in classifying railway track defects.

#### 2.4.2 Comparative analysis of machine learning algorithms

This section presents the four machine learning algorithms employed in this study: ANN, SVM, RF, KNN, used for classifying railway track conditions as either “non-defective” or “defective” based on image-derived feature embeddings. The SVM aims to construct an optimal hyperplane that separates two classes with the maximum possible margin. The fundamental prediction function of a linear SVM is expressed as:

$$f(x) = \beta_0 + \sum_{i=1}^n \alpha_i \langle x, x_i \rangle \tag{1}$$

where,  $\alpha_1, \dots, \alpha_n$  and  $\beta_0$  are parameters estimated based on the inner product between pairs of training data points. By replacing the inner product with a kernel function  $K(x_i, x_i)$ , the model applies a kernel-based approach. A linear kernel defined as:

$$K(x_i, x'_i) = \sum_{j=1}^p x_{ij} x'_{ij} \quad (2)$$

While ANN utilizes a multi-layer feedforward architecture to learn non-linear mappings from inputs to outputs. For a network with  $L$  layers, the forward propagation is defined as:

$$\begin{aligned} a^{(1)} &= \sigma(W^{(1)}x + b^{(1)}), \\ a^{(2)} &= \sigma(W^{(2)}a^{(1)} + b^{(2)}), \dots, \\ \hat{y} &= \sigma(W^{(L)}a^{(L-1)} + b^{(L)}) \end{aligned} \quad (3)$$

where,  $W^{(l)}$  and  $b^{(l)}$  denote the weights and biases at layer  $l$ ,  $\sigma$  is a non-linear activation function (ReLU or sigmoid), and  $\hat{y} \in [0,1]$  represents the predicted probability of the track being defective (bad). The ANN model showed strong performance in this study due to its ability to model complex, non-linear relationships in high-dimensional image data.

RF is an ensemble learning method that combines the outputs of multiple decision trees to improve classification robustness. The prediction function of RF is defined as:

$$\hat{y} = \text{majority\_vote}\{h_t(x)\}_{t=1}^T \quad (4)$$

where,  $h_t(x)$  is the prediction from the  $t$ -th decision tree, and  $T$  is the total number of trees in the ensemble. Each tree is trained on a random subset of the training data and features (bagging), allowing the model to reduce variance and improve generalization. RF is particularly effective in noisy datasets, as it aggregates multiple decision boundaries.

KNNs is a non-parametric, instance-based learning algorithm that classifies input samples based on their proximity to training examples. The distance between a test point  $x$  and a training point  $x_i$  is typically calculated using Euclidean distance:

$$d(x, x_i) = \sqrt{\sum_{j=1}^{2047} (x_j - x_{ij})^2} \quad (5)$$

The predicted label  $\hat{y}$  is determined by the majority class among the  $k$  nearest neighbors:

$$\hat{y} = \text{mode}\{x_i \in N_k(x)\} \quad (6)$$

where,  $N_k(x)$  represents the set of the  $k$  nearest training instance to  $x$ . While KNN is simple and intuitive, its effectiveness diminishes in high-dimensional spaces and when the data is not well-separated.

## 2.5 Phase 4: Infuse

This diagram illustrates the development of an ERP maintenance module that integrates AI into business processes. The approach follows the Infuse phase of the IBM AI Ladder, which focuses on embedding AI into workflows to

enable smarter operations and enhanced decision-making. Infusing AI into the core ERP workflows drives more systemic with learning capabilities, enabling them to transform data into predictive insights and smarter decisions [36].

### 2.5.1 Business process re-engineering

The process starts with business process re-engineering activities, including the preparation phase, analysis of the current state (As-Is), and the design of the desired future state (To-Be). These steps are crucial to ensure that existing processes are clearly mapped and improved before introducing AI.

A Focus Group Discussion (FGD) is used to gather expert input on the current maintenance system. This input informs the design of a new maintenance process that is integrated with ERP as we can see in Figure 4.

In practical terms, integration between the ML model and the ERP was exercised through a lightweight REST adaptor. The adaptor receives JSON outputs from the inference service and forwards them to the ERP using its standard remote-procedure API (JSON-RPC depending on version). This pattern ensures the ML pipeline remains vendor-neutral and portable, with the adaptor encapsulating system-specific details. The prototype was validated in a pilot (non-production) ERP environment through user-acceptance tests, rather than in a full live system. To prevent model obsolescence, the classifier is periodically retrained when sufficient new labeled imagery becomes available, keeping predictions calibrated over time. The focus of this study is therefore not on ERP software, but on the integration pattern itself within the AI Ladder's Infuse phase, embedding AI predictions into maintenance workflows for automated work orders, scheduling, and dashboard visualization.

ERP V15 is deployed on a Droplet Server running Ubuntu 20 LTS. The configuration is defined through a blueprinting process, followed by User Acceptance Testing (UAT) to validate system readiness. AI is infused into the maintenance process by embedding intelligent decision support within ERP functionalities. Development and customization are carried out using ERP V15 Studio, resulting in a fully integrated and AI-enhanced ERP maintenance module ready for live deployment.

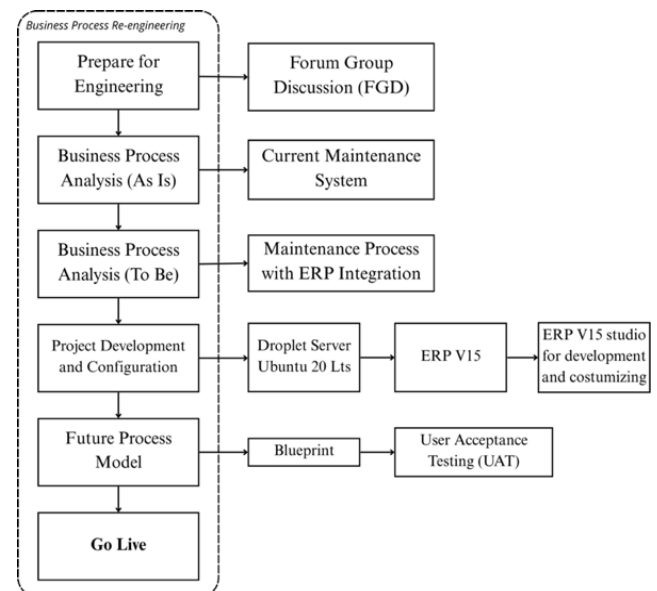


Figure 4. Business process re-engineering scheme

**Table 1.** Experts involved in FGD

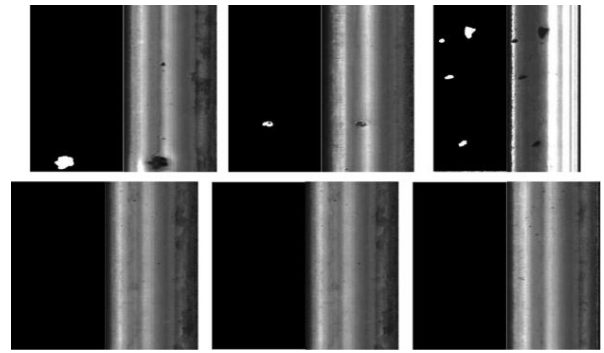
| No. | Company Nature                     | Job Nature                             | Experience (in years) |
|-----|------------------------------------|----------------------------------------|-----------------------|
| 1   | Railway Infrastructure Contractor  | Senior Track Engineer                  | 10                    |
| 2   | National Railway Operator          | Maintenance Planning Specialist        | 8                     |
| 3   | ERP Consulting Firm                | Lead ERP Functional Consultant         | 12                    |
| 4   | UAV and Sensor Integration Company | UAV Systems Engineer                   | 6                     |
| 5   | Software Development Firm          | ERP Module Developer                   | 7                     |
| 6   | University / Research Institute    | AI and Predictive Analytics Researcher | 5                     |

To ensure that the development of ERP features aligns with user needs and strategic priorities, the FGD method was subsequently applied. The following table presents the experts involved in the FGD process for determining the priority of ERP features and modules.

In the preparation engineering phase of an ERP project, the FGD method can be effectively applied to support complex decision-making involving multiple stakeholders, as shown in Table 1. One of its applications is in determining which ERP features and modules should be prioritized. By involving a panel of experts from various related departments, the company can gather in-depth opinions on the most critical and relevant features for the business needs. This approach aligns with findings from a study by Ifinedo and Nahar [37], which emphasizes the importance of incorporating diverse stakeholder perspectives to effectively prioritize ERP system features and ensure successful implementation.

### 3. RESULT

This part will demonstrate the outcomes obtained from each stage of the AI Ladder, which includes collect, organize, analyze, and infuse data. These stages serve as the foundational framework for implementing AI within the company. By thoroughly examining each phase, the study aims to provide a clear understanding of how data is transformed into actionable insights and integrated into business processes, ultimately driving more informed decision-making and enhancing organizational performance.

**Figure 5.** Image data and its binary mask (0 (black), 1 (white)) as ground truth labels

### 3.1 Collect

#### 3.1.1 Dataset collected from UAVs

The dataset utilized in this study comprises rail track images captured by the UAVs. The images in the dataset exhibit various forms of defects, such as cracks, deformations, and surface wear, which could compromise the operational integrity of railway infrastructure.

Each image captured by the UAV is annotated using a corresponding binary mask, an image of the same dimensions consisting solely of black (0) and white (1) pixel values (Figure 5). White pixels indicate regions where visible rail defects such as cracks, deformations, or wear are present, while black pixels represent defect-free areas. They serve as the ground truth labels during model training [33], enabling the machine learning algorithms to learn the visual patterns associated with defective and non-defective track segments.

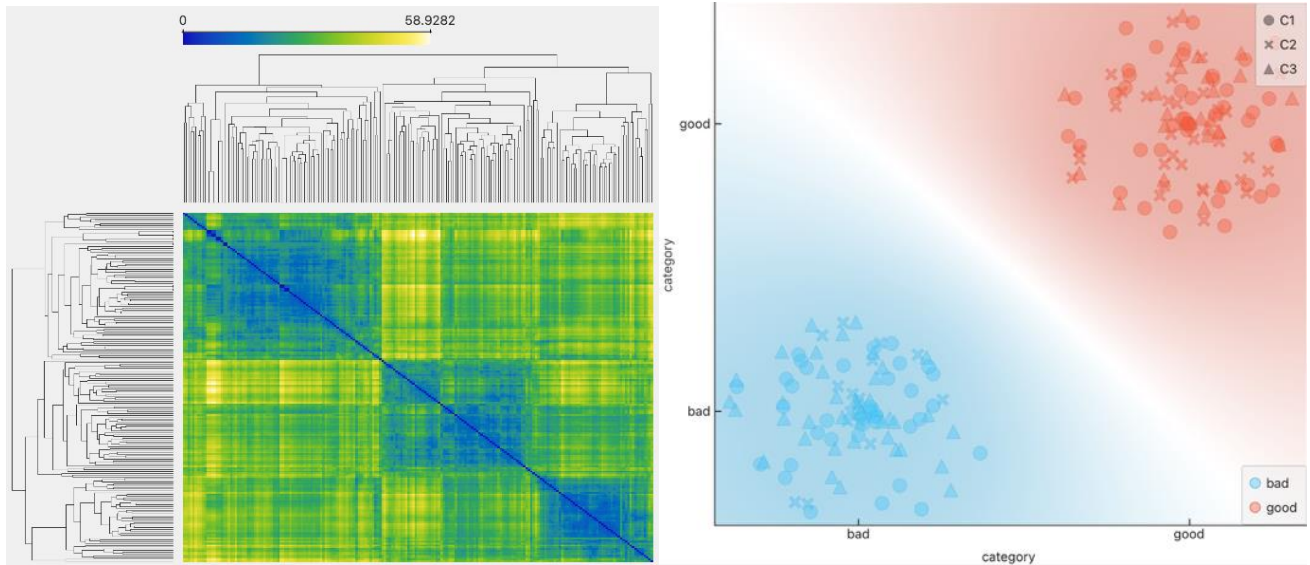
### 3.2 Organize

#### 3.2.1 Feature embedding and extraction

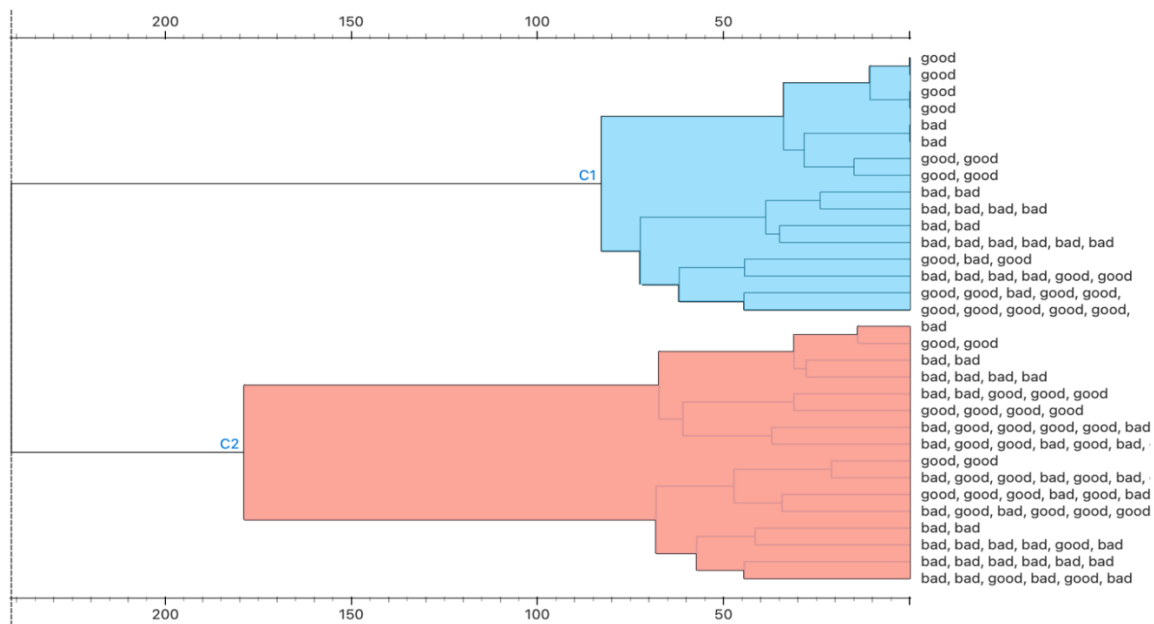
After transferring data to the core, all data is processed and embedded to adjust with the model easily, it is the next step of AI Ladder: organize. The size column represents the file size of each image in bytes. Larger file sizes generally indicate more detailed images [32], which could result from higher resolution or more complex visual content. Meanwhile, the width and height columns specify the image dimensions in pixels. In this dataset, the width is consistently 160 pixels, while the height varies across images, ranging from 333 to 427 pixels. It means that the images may have different aspect ratios, possibly due to variations in how they were captured or processed, as shown in Table 2.

**Table 2.** Images after embedded and converted into numerical values

| Category | Image name   | Image            | Size   | Width | Height | n0       | n1       | n2       | n3       | n4       |
|----------|--------------|------------------|--------|-------|--------|----------|----------|----------|----------|----------|
| bad      | rail_63_top  | bad/rail_63_     | 112755 | 160   | 427    | 0.025431 | 0.040893 | 0        | 0.247499 | 0.655503 |
| bad      | rail_9_top   | bad/rail_9_t...  | 89094  | 160   | 333    | 0.050763 | 0        | 0.000776 | 0.062975 | 0.316339 |
| bad      | rail_26_down | bad/rail_26_     | 107820 | 160   | 420    | 0.249109 | 0.004444 | 0        | 0.108371 | 0.238146 |
| bad      | rail_50_down | bad/rail_50_     | 122348 | 160   | 428    | 0.107908 | 0.008462 | 0.002033 | 0.583412 | 0.499811 |
| bad      | rail_3_down  | bad/rail_3_d...  | 87257  | 160   | 334    | 0.218566 | 0        | 0.01038  | 0.027712 | 0.32916  |
| bad      | rail_2_down  | bad/rail_2_d...  | 89057  | 160   | 334    | 0.131713 | 0.000146 | 0.001676 | 0.050996 | 0.485504 |
| bad      | rail_60_mid  | bad/rail_60_     | 126045 | 160   | 427    | 0.112627 | 0.367399 | 0        | 0.406943 | 0.556458 |
| bad      | rail_12_mid  | bad/rail_12_     | 88780  | 160   | 333    | 0.191149 | 0        | 0.021315 | 0.181009 | 0.336767 |
| bad      | rail_19_down | bad/rail_19_     | 120354 | 160   | 420    | 0.438061 | 0.00319  | 0.004395 | 0.236694 | 0.598159 |
| bad      | rail_18_top  | bad/rail_18_t... | 113431 | 160   | 420    | 0.279714 | 0.000176 | 0.008263 | 0.268504 | 0.41349  |
| bad      | rail_34_mid  | bad/rail_34_     | 121197 | 160   | 420    | 0.280254 | 0        | 0        | 0.145305 | 0.132375 |
| bad      | rail_24_mid  | bad/rail_24_     | 115051 | 160   | 420    | 0.212019 | 0        | 0.027149 | 0.280935 | 0.124773 |



**Figure 6.** Distance and clustering of data through heatmap and scatter plot



**Figure 7.** The dendrogram of hierarchical clustering

While the numerical columns labeled n0, n1, n2, n3, n4, and extending up to n2047, correspond to extracted feature values obtained through an image embedding process. These values are derived using a feature extraction technique, likely based on a deep learning model, where pre-trained data converted high-dimensional image data into lower-dimensional vectors [38]. These vectors capture essential features of the images, such as texture, shape, and patterns, while discarding unnecessary information. Each floating-point number encodes specific visual characteristics of the image, such as texture, structural patterns, or edges.

### 3.2.2 EDA

Before classification, this research uses distance metrics, scatter plots, and hierarchical clustering (Figure 6) to verify data suitability, cleanliness, and the distribution of "non-defective" and "defective" railway track features.

The analysis, particularly the distance matrix and dendrogram in Figure 6, reveals distinct clusters and well-

defined separability. This confirms that the chosen features effectively differentiate between track conditions, validating the dataset's readiness and providing guidance for supervised learning models [39].

A hierarchical clustering heatmap (Figure 6), ranging from blue (high similarity) to yellow (high variability), alongside dendrograms, reveals strong intra-cluster similarity and distinct groupings within rail track image data. This indicates that the images naturally cluster, suggesting good potential for separating non-defective and defective tracks using unsupervised methods. This visualization helps in evaluating how well the data points cluster together [40].

Furthermore, a scatter plot clearly distinguishes non-defective (red) and defective (blue) classes, demonstrating high separability in the feature space. This strong visual distinction reinforces the idea that classification models should perform well on this dataset.

Normalization using Euclidean distance is crucial [40]. It standardizes feature scales, preventing larger-magnitude

features from skewing similarity calculations and ensuring that distance-based models operate effectively and without bias. This preprocessing step helps maintain meaningful relationships and supports accurate clustering and classification [41].

Visual analysis, employing both a hierarchical clustering heatmap and a scatter plot (Figure 7), strongly supports the feasibility of classifying rail track images. The heatmap (blue = similar, yellow = different) and its accompanying dendrograms reveal significant intra-cluster similarity, identifying two primary groups: C1 (largely "non-defective" tracks) and C2 (mainly "defective" tracks), while also highlighting potential transitional states through its hierarchical structure. This clear separability, further emphasized by distinct red ("non-defective") and blue ("defective") groups in the scatter plot, is achieved through crucial Euclidean distance normalization, which ensures unbiased feature representation. These findings confirm that extracted features effectively differentiate track conditions and provide practical guidance for classification: they help determine the appropriate number of classes, inform hyperparameter tuning, aid in selecting balanced training data, and suggest that different defect types may require distinct handling approaches.

### 3.3 Analyze

#### 3.3.1 Model evaluation

Model evaluation is part of Analyze phase in AI Ladder, where machine learning models are applied to interpret and extract meaningful insights from collected and organized data, using performance metrics and visualizations to facilitate understanding and guide decision-making.

##### Precision-recall curve

Precision–recall curves for RF, ANNs, KNNs, and SVMs demonstrate strong classification performance for the “non-defective (good)” class (left panel), it can be seen in Figure 8. All models maintain high precision across a wide range of recall, indicating accurate identification of non-defective tracks with minimal false positives. The curves’ proximity to the top-right region reflects robust predictive power and a favorable precision-recall trade-off for this class [42].

In contrast, the precision-recall curves for the defective class reveal markedly weaker performance. Across most recall levels, precision is low, implying a higher rate of false positives when predicting defective tracks. The curves’ closeness to the baseline suggests the classifiers struggle to effectively separate defective instances [43]. This highlights a clear disparity: models excel on the non-defective class but face substantial challenges on the defective class.

##### ROC curve diagram

For the ROC curves, the non-defective class (left) shows stronger separability than the defective class (right). ANN (orange) and SVM (pink) trace the closest trajectories to the top-left corner, achieving higher True Positive Rates (TPR) at lower False Positive Rates (FPR). RF attains mid-tier performance, while KNN consistently yields the weakest curves as shown in Figure 9.

KNN shows the weakest performance, struggling to differentiate classes. While models handle non-defective tracks well, their sensitivity drops when predicting the defective class, increasing the risk of false negatives (missed defects), particularly for RF and KNN.

The confusion matrices provide a detailed breakdown of the

classification performance of each model by illustrating the distribution of true positives, true negatives, false positives, and false negatives [43].

The confusion matrices in Table 3 provide a detailed comparison of the four classifiers in identifying rail tracks as defective or non-defective. ANN achieves the most balanced results, correctly classifying 84 defective and 89 non-defective tracks, with 37 misclassifications (21 FN, 16 FP). RF follows closely (85/86 correct; 39 errors: 20 FN, 19 FP), showing moderate, well-balanced performance. SVM attains 84 correct defective and 83 correct non-defective predictions, with 43 errors (21 FN, 22 FP). KNN performs the weakest, correctly identifying 69 defective and 95 non-defective tracks but incurring 46 errors, driven largely by 36 false negatives. Using SMOTE, we obtained a class distribution that is approximately even across categories [35]. Overall, ANN emerged as the most reliable models for rail defect detection.

The ANN achieved the best performance due to its natural ability to exploit the complexity of feature representations produced through embedding. The embeddings generated in the Organize stage are nonlinear and high-dimensional, enabling models with greater representational capacity to form more flexible decision boundaries compared to other algorithms. This explains why RF, SVM, and KNN lagged behind: while RF provides stability, SVM offers margin-based robustness, and KNN delivers local simplicity, none of these possess the layered learning mechanism required to capture subtle patterns and complex nonlinear interactions within the embeddings [44].

Although overfitting is a potential concern for ANN, this issue was mitigated through model regularization strategies, including cross-validation and careful monitoring of performance. The outcomes across Accuracy, F1, MCC, and AUC indicate that the model did not merely “memorize” the training data but truly generalized. We also observed that ANN maintained stability under small input perturbations, reflecting robustness and an indication of generalization, consistent with the findings of Novak et al. [45], who demonstrated that the model can generalize effectively.

The sample prediction results (Table 4) provide a detailed comparison of how different machine learning models classify rail track images as either defective or non-defective. Observing the classification trends, the Neural Network and SVM models show higher consistency in labeling the images correctly, aligning with their previously reported higher accuracy, F1-score, and MCC values. The Neural Network, in particular, consistently outperforms others by correctly classifying more images as non-defective or defective without frequent misclassification.

Conversely, the KNN model appears to be the least reliable, as it often disagrees with the more accurate models, frequently misclassifying defective tracks as non-defective. This reflects its lower recall and precision scores, making it less suitable for defect detection. The RF model performs moderately well but still exhibits occasional misclassifications. These results reinforce the conclusion that neural network models are the best choices for rail defect detection due to their superior classification accuracy, reducing the risk of safety hazards caused by undetected rail defects.

Figure 10 displays a set of randomly selected, unlabeled test images used for classification by the models, serving as a basis to evaluate prediction accuracy. Accurate railway track defect detection is vital for safety and structural integrity. Early identification allows timely maintenance, reducing derailment

risks and operational costs. Machine learning models like RF, SVM, ANN, and KNN enhance detection precision and efficiency. Their implementation supports safer railways and

more effective maintenance, contributing to a more reliable and efficient rail network.

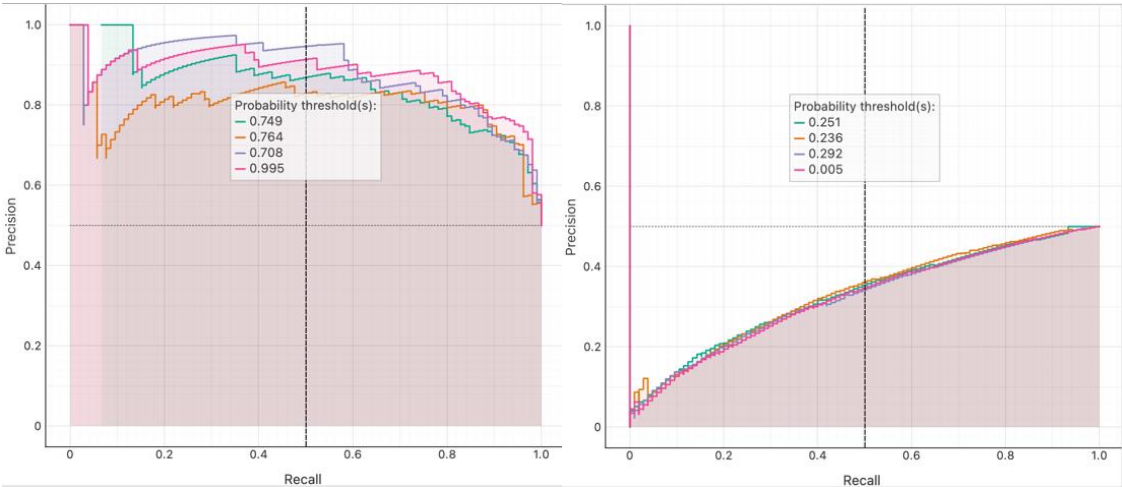


Figure 8. Precision-recall curve (“non-defective (good)” on the left, “defective (bad)” on the right)

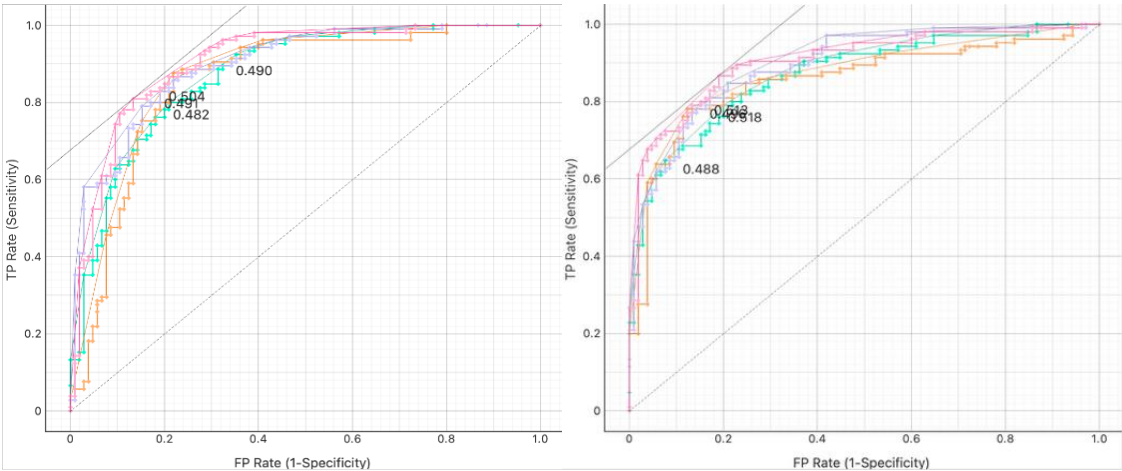


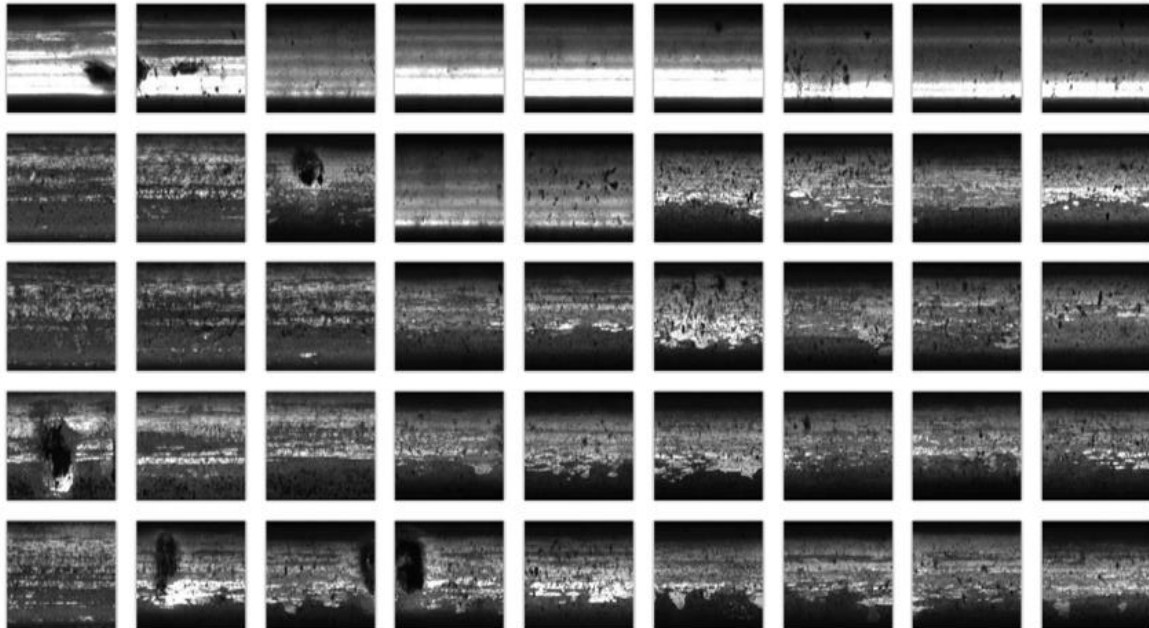
Figure 9. ROC curve and comparison between models (“non-defective (good)” on the left, “defective (bad)” on the right)

Table 3. Confusion matrices of four models

| Predicted (RF)  |      |     |      |     | Predicted (KNN) |      |     |      |     |
|-----------------|------|-----|------|-----|-----------------|------|-----|------|-----|
| Actual (RF)     |      | bad | good | Σ   | Actual (KNN)    |      | bad | good | Σ   |
|                 | bad  | 85  | 20   | 105 |                 | bad  | 69  | 36   | 105 |
|                 | good | 19  | 86   | 105 |                 | good | 10  | 95   | 105 |
|                 | Σ    | 104 | 106  | 210 |                 | Σ    | 79  | 131  | 210 |
| Predicted (ANN) |      |     |      |     | Predicted (SVM) |      |     |      |     |
| Actual (ANN)    |      | bad | good | Σ   | Actual (SVM)    |      | bad | good | Σ   |
|                 | bad  | 84  | 21   | 105 |                 | bad  | 84  | 21   | 105 |
|                 | good | 16  | 89   | 105 |                 | good | 22  | 83   | 105 |
|                 | Σ    | 100 | 110  | 210 |                 | Σ    | 106 | 104  | 210 |

Table 4. Prediction result with implementing each model

| RF (1) | KNN (1) | SVM (1) | Neural Network (1) | Image Name   | Image          | Size   | Width | Height |
|--------|---------|---------|--------------------|--------------|----------------|--------|-------|--------|
| bad    | bad     | bad     | bad                | rail_63_top  | rail_63_top... | 112755 | 160   | 427    |
| good   | good    | bad     | bad                | rail_50_down | rail_50_dow... | 122348 | 160   | 428    |
| bad    | good    | bad     | bad                | rail_60_mid  | rail_60_mid... | 126045 | 160   | 427    |
| good   | good    | good    | good               | rail_65_down | rail_65_dow... | 113633 | 160   | 428    |
| good   | good    | good    | good               | rail_64_down | rail_64_dow... | 120172 | 160   | 428    |
| bad    | bad     | bad     | bad                | rail_56_mid  | rail_56_mid... | 94058  | 160   | 427    |
| bad    | bad     | bad     | bad                | rail_55_top  | rail_55_top... | 109177 | 160   | 427    |
| bad    | bad     | bad     | bad                | rail_57_mid  | rail_57_mid... | 135654 | 160   | 427    |
| good   | good    | good    | good               | rail_62_top  | rail_62_top... | 117386 | 160   | 427    |
| good   | good    | good    | good               | rail_61_mid  | rail_61_mid... | 126217 | 160   | 427    |



**Figure 10.** Sample image test data

**Table 5.** Performance metrics of four models

| Model                     | AUC          | CA           | F1-score     | Precision    | Recall       | MCC          |
|---------------------------|--------------|--------------|--------------|--------------|--------------|--------------|
| Random Forest             | 0.895        | 0.814        | 0.814        | 0.814        | 0.814        | 0.629        |
| Artificial Neural Network | <b>0.903</b> | <b>0.824</b> | <b>0.824</b> | <b>0.825</b> | <b>0.824</b> | <b>0.648</b> |
| K-Nearest Neighbor        | 0.869        | 0.781        | 0.778        | 0.799        | 0.781        | 0.580        |
| Support Vector Machine    | 0.856        | 0.795        | 0.795        | 0.795        | 0.795        | 0.591        |

#### Model performance

To assess the effectiveness in classifying railway track conditions, four machine learning models (ANN, SVM, RF, and KNN) were compared. This evaluation used key metrics like Area Under the Curve (AUC) (distinguishing ability), Classification Accuracy (CA) (overall accuracy), F1-score (balancing precision and recall), Precision, Recall, and Matthews Correlation Coefficient (MCC).

As shown in Table 5, the ANN attains the strongest results, with AUC 0.903, CA 0.824, F1-score 0.824, Precision 0.825, Recall 0.824, and MCC 0.648, indicating robust discrimination and a balanced precision-recall trade-off. The RF follows closely, achieving AUC 0.895, CA 0.814, F1-score 0.814, Precision 0.814, Recall 0.814, and MCC 0.629, reflecting stable and well-balanced predictions. SVM records AUC 0.856, CA 0.795, F1-score 0.795, Precision 0.795, Recall 0.795, and MCC 0.591, showing competent discrimination while trailing ANN and RF. KNN model yields the weakest scores, with AUC 0.869, CA 0.781, F1-score

0.778, Precision 0.799, Recall 0.781, and MCC 0.580, suggesting higher sensitivity to local noise and class overlap in this context. Taken together, the results point to ANN as the most effective and balanced classifier for this task, with RF a close second. SVM delivers solid but lower scores, while kNN is least suited here. If prioritizing missed defects, consider threshold tuning or recall-oriented settings for the selected model.

#### 3.4 Infuse

##### 3.4.1 Focus group discussion

##### Recommended ERP dashboard features

Experts agreed that the dashboard should prioritize clarity and quick decision-making [46]. Since the classification is binary (defective/non-defective), the interface can remain simple yet informative. Recommended features are included in Table 6.

**Table 6.** Recommended ERP dashboard elements

| Dashboard Element              | Description                                                                          | Purpose                                               |
|--------------------------------|--------------------------------------------------------------------------------------|-------------------------------------------------------|
| Track Map Visualization        | Railway line displayed with segments marked green (non-defective) or red (defective) | Enables quick identification of damaged areas         |
| Track Segment List             | Tabular view of segment IDs and their status                                         | For data logging, inspection reference, and filtering |
| Condition Statistics           | Pie chart or bar graph showing percentage of non-defective vs defective segments     | Provides an overview of overall infrastructure health |
| Defective Status Notifications | Real-time alert when a segment is detected as defective                              | Facilitates immediate response from maintenance teams |
| Condition Change Log           | Historical view of changes in segment status                                         | Useful for trend analysis and planning re-inspections |

**Table 7.** Required ERP integration components

| ERP Module             | Integration Function                                             | Explanation                                                  |
|------------------------|------------------------------------------------------------------|--------------------------------------------------------------|
| Maintenance Management | Auto-generates work orders for defective segments                | Ensures immediate action is assigned and tracked             |
| Asset Management       | Stores segment status and metadata                               | Treats each track segment as an asset with condition history |
| Scheduling             | Allocates inspection or repair tasks based on defective segments | Helps prioritize and organize fieldwork                      |
| Notification System    | Sends alerts to relevant personnel                               | Reduces delay in information flow                            |
| Reporting              | Periodic export of condition summaries and work logs             | Supports documentation, audit, and performance tracking      |

**Required ERP integration components**

The discussion emphasized smooth and automated integration between UAVs-ML outputs and ERP modules, ensuring that detected defective segments trigger relevant business processes without manual intervention [47]. The key ERP modules and their roles are shown in Table 7.

The discussion concluded that a minimalist yet actionable dashboard is suitable for this project, as the system only classifies track segments into non-defective or defective conditions. Key emphasis was placed on real-time visual feedback (via map and segment list), automatic ERP-triggered responses (such as maintenance work orders), and alert mechanisms to ensure operational agility [47]. The required integration aligns with the Analyze and Infuse stages of the AI Ladder Model, where simple AI outputs are effectively embedded into core business processes to improve decision-making and responsiveness [48]. The simplicity of the classification model is not a limitation; rather, it supports faster deployment and higher user acceptance in field operations.

**Business process re-engineering flow**

The following section presents the reengineering of the railway track maintenance process, transitioning from a conventional manual approach to a data-driven, AI-integrated workflow.

**3.4.2 As-Is process (Before AI and ERP integration)**

Before digital transformation, railway track inspection and maintenance relied heavily on manual visual checks and periodic patrols. This process posed several limitations in speed, accuracy, consistency, and safety.

The manual workflow often resulted in delayed detection of track issues, inefficient use of maintenance resources, and safety. This reactive model (Table 8) lacked the agility and intelligence needed for modern railway infrastructure management [49].

**To-be process (After AI, UAVs and ERP integration)**

After reengineering, the system integrates UAVs for automated data collection, machine learning for defect detection, and ERP for seamless action execution (Table 9).

In this proposed model, each component plays a strategic role, UAVs offer wide and efficient scanning, ML ensures fast and consistent classification of track conditions, and ERP handles operational execution from asset status updates to

automated work orders. This end-to-end pipeline enables predictive and intelligent maintenance. It significantly enhances predictive maintenance by enabling real-time asset monitoring and automated decision-making [47].

**Table 8.** As-Is process

| Step | Description                                                                 |
|------|-----------------------------------------------------------------------------|
| 1    | Manual field inspections are conducted periodically.                        |
| 2    | Engineers identify track issues visually and record them manually.          |
| 3    | Maintenance teams rely on reports or technician feedback to schedule tasks. |
| 4    | Limited coordination between inspection, scheduling, and asset records.     |

**Table 9.** To-Be process

| Step | Description                                                                    |
|------|--------------------------------------------------------------------------------|
| 1    | UAVs capture high-resolution images of railway tracks.                         |
| 2    | ML models classify segments as non-defective or defective from image datasets. |
| 3    | Results are sent to ERP through middleware.                                    |
| 4    | ERP updates asset condition, triggers work orders automatically.               |
| 5    | Maintenance is scheduled and teams are notified instantly.                     |
| 6    | Dashboard visualizes current status, and logs are updated in real time.        |

**3.4.3 ERP system blueprint**

This section defines the system blueprint, which outlines how data captured from UAVs inspections is processed and utilized within the ERP ecosystem to enable predictive maintenance.

**Input pipeline**

The input pipeline begins with UAVs capturing aerial images of the track, which are then analyzed by machine learning models and converted into structured data inputs (Table 10). It supports efficient transformation of unstructured visual data into actionable insights for railway maintenance [20].

This pipeline ensures that only processed, meaningful data enters the ERP system. The use of an integration layer standardizes the format, enabling consistent updates of track segment conditions and activation of business logic.

**Table 10.** Input pipeline

| Source      | Type                              | Processing                                                      |
|-------------|-----------------------------------|-----------------------------------------------------------------|
| UAVs System | Image Dataset (.jpg, .png, .tiff) | ML classifies condition per segment (non-defective / defective) |
| ML Output   | Structured JSON                   | Image                                                           |
| Integration | API / Middleware                  | Sends data to ERP and links with asset IDs                      |

**Table 11.** ERP module used

| Module                 | Function                        | Purpose                                        |
|------------------------|---------------------------------|------------------------------------------------|
| Asset Management       | Stores railway segment metadata | Tracks segment ID, GPS, length, condition      |
| Condition Monitoring   | Receives ML classification      | Updates asset condition status automatically   |
| Maintenance Management | Generates work orders           | Automates WO creation for defective segments   |
| Scheduling             | Assigns maintenance tasks       | Technician calendars, job queues               |
| Notification System    | Alerts team members             | Sends ERP/email alerts on new defective status |
| Reporting              | Tracks KPIs and logs            | Provides maintenance history, SLA performance  |

**Table 12.** Dashboard elements

| Component        | Description                        | Visual Type             |
|------------------|------------------------------------|-------------------------|
| Track Map Viewer | Map with green/red segment status  | Interactive Map         |
| Segment Table    | List of segment metadata & status  | Table with filters      |
| Condition Stats  | Non-defective vs defective summary | Pie/bar charts          |
| Alerts Panel     | Real-time issue notifications      | Notification cards/logs |
| History Log      | Condition change tracking          | Timeline per segment    |

**Table 13.** Dashboard elements of UAT

| No. | Component     | Test Case                          | Expected Result                                                | Status |
|-----|---------------|------------------------------------|----------------------------------------------------------------|--------|
| 1   | Map Viewer    | Load segment condition map         | Segments shown in green (non-defective) or red (non-defective) | Passed |
| 2   | Segment Table | List track segments & statuses     | Correct data, sortable & filterable                            | Passed |
| 3   | Stats Summary | Display condition ratio            | Accurate chart from ML inputs                                  | Passed |
| 4   | Alerts Panel  | Trigger new defective status alert | ERP and email notifications received                           | Passed |
| 5   | History Log   | Track segment status updates       | Logs show correct timestamps and actions                       | Passed |

**Table 14.** ERP module used for UAT

| No. | Module           | Test Case                              | Expected Result                               | Status |
|-----|------------------|----------------------------------------|-----------------------------------------------|--------|
| 1   | Asset Management | Register/update asset condition        | Status reflects latest ML classification      | Passed |
| 2   | Maintenance      | Trigger work order on defective status | WO is created and linked to asset             | Passed |
| 3   | Scheduling       | Assign jobs to crew                    | Task appears in technician dashboard          | Passed |
| 4   | Notification     | Alert for defective segments           | Email/ERP alert reaches correct personnel     | Passed |
| 5   | Reporting        | Export condition log                   | PDF/Excel report includes timestamped updates | Passed |

### ERP module used

The ERP system comprises several interconnected modules, each with a specific function in the maintenance cycle as shown in Table 11.

Together, these modules allow for automated and intelligent workflows. For example, a "Bad" classification triggers a new maintenance request, which is then scheduled, tracked, and reported, all within the ERP environment. This closed-loop system enhances operational efficiency and accountability [12].

### Dashboard elements

The dashboard is the visual layer of the system, enabling users to monitor track conditions, review alerts, and access historical records.

By combining geospatial visualization (map viewer), data analytics (charts and stats), and operational logs, the dashboard empowers decision-makers with real-time insights. This helps prioritize responses and streamline coordination across departments [50].

### 3.4.4 UAT matrix

UAT was conducted to evaluate whether the developed system functions according to its intended design and meets user requirements.

### 3.4.5 Dashboard element

Table 12 lists each core component of the dashboard and summarizes its test results as shown in Table 13. All dashboard elements performed as expected during UAT. Users were able

to interact with the map, access filtered tables, receive condition alerts, and trace updates.

This confirmed that the dashboard is both user-friendly and functionally reliable. It significantly enhanced usability and reliability in operational contexts [51].

### ERP modules

Each ERP module was tested individually to ensure proper response to AI input and seamless integration across workflows.

The system successfully automated key actions such as condition updates and work order creation. Notifications and reports were generated accurately, confirming that the ERP system can reliably manage predictive maintenance workflows triggered by ML-based track assessments. Accurate generation of notifications and reports confirms that ERP systems integrated with machine learning assessments can effectively support and automate predictive maintenance workflows [52] as shown in Table 14.

## 4. CONCLUSION

Ensuring the reliability of railway infrastructure demands not only accurate detection but also a seamless bridge from perception to action a gap that many vision-only studies leave unaddressed. This work operationalizes the AI Ladder end-to-end Collect, Organize, Analyze, infuse by deploying UAVs for systematic image capture; standardizing imagery; producing binary-masked ground truth; and embedding samples into

fixed-length feature vectors for learning. Across four classifiers (ANN, SVM, RF, KNN) and metrics suited to class imbalance (AUC, accuracy, F1-score, MCC), ANN emerges as the strongest performer (AUC = 0.935; accuracy = 0.884; F1-score = 0.884; MCC = 0.768). Crucially, we translate model outputs into execution via ERP integration that provides real-time defect flags, dashboard visualization, automated work orders, automated scheduling, and notifications, while maintaining auditable condition histories. By embedding classification directly within maintenance workflows, operators can shift from periodic, manual inspections to continuous, predictive maintenance at scale. Beyond technical accuracy, the implemented ML-ERP pipeline delivers a practical path from aerial sensing to actionable decision-making and aligns with SDG 9, offering a replicable blueprint for intelligent, accountable railway asset management.

We note several scope qualifications. The evaluation was conducted on a subset of corridors and operating configurations; variation across seasons and weather, camera viewpoints, and track characteristics has not been exhaustively assessed. The ERP integration was demonstrated in a limited pilot, so evidence of full-scale operational performance remains pending. In addition, probability calibration and corridor-specific operating thresholds have not yet been finalised, and a comprehensive end-to-end cost-benefit assessment with systematic stress testing under adverse field conditions (e.g., glare, precipitation, vegetation occlusion) is outstanding.

The concrete directions are outlined as follows:

- (i) probability calibration and cost-aware thresholding tailored per corridor;
- (ii) multi-corridor, multi-season field trials with operational metrics (e.g., MTTR, actionable-alarm rate);
- (iii) MLOps for drift monitoring, alerting, A/B testing, and scheduled retraining;
- (iv) edge deployment with model compression to meet latency constraints;
- (v) uncertainty estimation to prioritise human review;
- (vi) human-in-the-loop and active learning within the ERP;
- (vii) exploration of modern detection/segmentation architectures alongside self-supervised pretraining for improved robustness;
- (viii) multi-sensor fusion (UAV imagery with GNSS/IMU/LiDAR) and geospatial hotspot mapping;
- (ix) standardisation of defect taxonomy and severity levels with external benchmarking;
- (x) quantitative economic analysis to substantiate safety and maintenance value at scale.

## REFERENCES

- [1] Powrie, W. (2024). Railway track substructure: Recent research and future directions. *Transportation Geotechnics*, 46: 101234. <https://doi.org/10.1016/j.trgeo.2024.101234>
- [2] Wijaya, T. (2024). Risk is not measured, but contested and compromised: A case study of the Jakarta–Bandung High-Speed Railway. *Journal of Contemporary Asia*, 55: 405–429. <https://doi.org/10.1080/00472336.2024.2378856>
- [3] Zufarihsan, R., Tambusay, A., Suprobo, P., Suryanto, B., Laghrouche, O. (2025). Recent developments in high-speed railway in Indonesia: Superstructure construction and track infrastructure. *Transportation Research Interdisciplinary Perspectives*, 31: 101385. <https://doi.org/10.1016/j.trip.2025.101385>
- [4] Yuantoko, T.D., Djunaedi, Z., Wirawan, M. (2024). A socio-technical systems approach of the accident analysis in Indonesian multiple train accident cases: An application of AcciMap methodology. *Journal of Emergency Management*, 22(2): 155–167. <https://doi.org/10.5055/jem.0830>
- [5] Iridiastadi, H., Ikatrinasari, Z.F. (2012). Indonesian railway accidents – Utilizing Human Factors Analysis and Classification System in determining potential contributing factors. *WORK: A Journal of Prevention, Assessment & Rehabilitation*, 41(S1): 4246–4249. <https://doi.org/10.3233/WOR-2012-0126-4246>
- [6] Calabrese, C., Mejia, B., McInnis, C.A., France, M., Nadler, E., Raslear, T.G. (2017). Time of day effects on railroad roadway worker injury risk. *Journal of Safety Research*, 61: 53–64. <https://doi.org/10.1016/j.jsr.2017.02.007>
- [7] Byeoung-Soo, Y., Tae-Yoon, K., Sun-Haeng C., Won-Mo, G. (2024). A study on the cause analysis of human error accidents by railway job. *Korea Science*, 7(1): 27–33. <https://doi.org/10.13106/jwmap.2024.Vol7.no1.27>
- [8] Walker, G., Mallett, X. (2011). Rail incidents. In *Disaster Victim Identification*. CRC Press, pp. 197–212.
- [9] Alahakoon, S., Sun, Y.Q., Spiryagin, M., Cole, C. (2018). Rail flaw detection technologies for safer, reliable transportation: A review. *Journal of Dynamic Systems, Measurement, and Control*, 140(2): 020801. <https://doi.org/10.1115/1.4037295>
- [10] Papathanasiou, N., Adey, B.T., Burkhalter, M. (2020). Defining and quantifying railway service to plan infrastructure interventions. *Infrastructure Asset Management*, 7(3): 146–166. <https://doi.org/10.1680/jinam.18.00044>
- [11] Agnisarman, S., Lopes, S., Madathil, K.C., Piratla, K., Gramopadhye, A. (2019). A survey of automation-enabled human-in-the-loop systems for infrastructure visual inspection. *Automation in Construction*, 97: 52–76. <https://doi.org/10.1016/j.autcon.2018.10.019>
- [12] Zhuang, L., Wang, L., Zhang, Z., Tsui, K.L. (2018). Automated vision inspection of rail surface cracks: A double-layer data-driven framework. *Transportation Research Part C: Emerging Technologies*, 92: 258–277. <https://doi.org/10.1016/j.trc.2018.05.007>
- [13] Li, Z.W., Liu, X.Z., Lu, H.Y., He, Y.L., Zhou, Y.L. (2020). Surface crack detection in precast slab track in high-speed rail via infrared thermography. *Materials*, 13(21): 4837. <https://doi.org/10.3390/ma13214837>
- [14] Ni, X., Liu, H., Ma, Z., Wang, C., Liu, J. (2021). Detection for rail surface defects via partitioned edge feature. *IEEE Transactions on Intelligent Transportation Systems*, 23(6): 5806–5822. <https://doi.org/10.1109/TITS.2021.3058635>
- [15] Thomas, R., Zikopoulos, P. (2020). The AI Ladder: Accelerate Your Journey to AI. O'Reilly Media.
- [16] Zhou, H., Yu, K.M., Chen, Y.C., Hsu, H.P. (2021). A hybrid feature selection method RFSTL for manufacturing quality prediction based on a high dimensional imbalanced dataset. *IEEE Access*, 9: 29719–29735. <https://doi.org/10.1109/ACCESS.2021.3059298>
- [17] Cunningham, P., Delany, S.J. (2021). K-nearest neighbour classifiers-a tutorial. *ACM Computing*

- Surveys (CSUR), 54(6): 1-25. <https://doi.org/10.1145/3459665>
- [18] Chenariyan Nakhace, M., Hiemstra, D., Stoelinga, M., van Noort, M. (2019). The recent applications of machine learning in rail track maintenance: A survey. In Third International Conference on Reliability, Safety, and Security of Railway Systems, Lille, France, pp. 91-105. [https://doi.org/10.1007/978-3-030-18744-6\\_6](https://doi.org/10.1007/978-3-030-18744-6_6)
- [19] Ghaddar, B., Naoum-Sawaya, J. (2018). High dimensional data classification and feature selection using support vector machines. *European Journal of Operational Research*, 265(3): 993-1004. <https://doi.org/10.1016/j.ejor.2017.08.040>
- [20] Kumar, A., Harsha, S.P. (2024). A systematic literature review of defect detection in railways using machine vision-based inspection methods. *International Journal of Transportation Science and Technology*, 18: 207-226. <https://doi.org/10.1016/j.ijtst.2024.06.006>
- [21] Tirupathi, R., Ramachandran, R., Khan, I., Goel, O., Jain, P.A., Kumar, D.L. (2024). Leveraging machine learning for predictive maintenance in SAP plant maintenance (PM). *Journal of Quantum Science and Technology (JQST)*, 1(2): 18-55.
- [22] Pai, A., Rane, S. (2014). Development and implementation of maintenance management module of enterprise resource planning in maintenance of power plant. *International Journal of System Assurance Engineering and Management*, 5(4): 534-543. <https://doi.org/10.1007/s13198-013-0203-4>
- [23] Panchal, D., Garg, D. (2026). Application of reliability-centered maintenance with a computerized maintenance management system for the wheelset in rolling stock. *Spectrum of Decision Making and Applications*, 3(1): 70-84. <https://doi.org/10.31181/sdmap31202636>
- [24] Jawad, Z.N., Balázs, V. (2024). Machine learning-driven optimization of enterprise resource planning (ERP) systems: A comprehensive review. *Beni-Suef University Journal of Basic and Applied Sciences*, 13(1): 4. <https://doi.org/10.1186/s43088-023-00460-y>
- [25] Bojarczak, P., Lesiak, P. (2021). UAVs in rail damage image diagnostics supported by deep-learning networks. *Open Engineering*, 11(1): 339-348. <https://doi.org/10.1515/eng-2021-0033>
- [26] Aela, P., Chi, H.L., Fares, A., Zayed, T., Kim, M. (2024). UAV-based studies in railway infrastructure monitoring. *Automation in Construction*, 167: 105714. <https://doi.org/10.1016/j.autcon.2024.105714>
- [27] Jung, G., Shin, J., Lee, S. (2023). Impact of preprocessing and word embedding on extreme multi-label patent classification tasks. *Applied Intelligence*, 53(4): 4047-4062. <https://doi.org/10.1007/s10489-022-03655-5>
- [28] Butt, U.A., Mehmood, M., Shah, S.B.H., Amin, R., et al. (2020). A review of machine learning algorithms for cloud computing security. *Electronics*, 9(9): 1379. <https://doi.org/10.3390/electronics9091379>
- [29] Rainio, O., Teuho, J., Klén, R. (2024). Evaluation metrics and statistical tests for machine learning. *Scientific Reports*, 14(1): 6086. <https://doi.org/10.1038/s41598-024-56706-x>
- [30] Fishman, N., Stryker, C. (2020). Climbing the AI ladder. In *Smarter Data Science: Succeeding with Enterprise-Grade Data and AI Projects*. John Wiley & Sons, pp. 1-16. <https://doi.org/10.1002/9781119697985.ch1>
- [31] Broday, E.E. (2022). The evolution of quality: From inspection to quality 4.0. *International Journal of Quality and Service Sciences*, 14(3): 368-382. <https://doi.org/10.1108/IJQSS-09-2021-0121>
- [32] Cui, P., Wang, X., Pei, J., Zhu, W. (2018). A survey on network embedding. *IEEE Transactions on Knowledge and Data Engineering*, 31(5): 833-852. <https://doi.org/10.1109/TKDE.2018.2849727>
- [33] Ronneberger, O., Fischer, P., Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. In 18th International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, pp. 234-241. [https://doi.org/10.1007/978-3-319-24574-4\\_28](https://doi.org/10.1007/978-3-319-24574-4_28)
- [34] Varga, D. (2024). Mitigating data leakage in a WIFI CSI benchmark for human action recognition. *Sensors*, 24(24): 8201. <https://doi.org/10.3390/s24248201>
- [35] Haekal, J., Mu'min, R., Fauzan, Hamid, A., Sjarifudin, D., Turseno, A., Paduloh, Adriansyah, A. (2025). Hybrid feature-based critical success factors in cloud enterprise resource planning through artificial neural networks and random forest. *Mathematical Modelling of Engineering Problems*, 12(6): 1911-1924. <https://doi.org/10.18280/mmep.120608>
- [36] Nyathani, R., Allam, K. (2024). Synergizing AI, cloud computing, and big data for enhanced enterprise resource planning (ERP) systems. *International Journal of Computer Techniques*, 11: 1-6. <https://doi.org/10.5281/zenodo.10600959>
- [37] Ifinedo, P., Nahar, N. (2007). ERP systems success: an empirical analysis of how two organizational stakeholder groups prioritize and evaluate relevant measures. *Enterprise Information Systems*, 1(1): 25-48. <https://doi.org/10.1080/17517570601088539>
- [38] Boppana, A. (2024). Enhancing semantic search in high-dimensional vector spaces: An analysis of sentence embeddings and indexing strategies. Master's thesis, State University of New York at Stony Brook.
- [39] Jiang, T., Gradus, J.L., Rosellini, A.J. (2020). Supervised machine learning: A brief primer. *Behavior Therapy*, 51(5): 675-687. <https://doi.org/10.1016/j.beth.2020.05.002>
- [40] Ding, K., Ma, K., Wang, S., Simoncelli, E.P. (2020). Image quality assessment: Unifying structure and texture similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(5): 2567-2581. <https://doi.org/10.1109/TPAMI.2020.3045810>
- [41] Kang, B., Kim, S., Jung, H., Choe, J., Lee, K. (2019). Efficient assessment of reservoir uncertainty using distance-based clustering: A review. *Energies*, 12(10): 1859. <https://doi.org/10.3390/en12101859>
- [42] Miao, J., Zhu, W. (2022). Precision-recall curve (PRC) classification trees. *Evolutionary Intelligence*, 15(3): 1545-1569. <https://doi.org/10.1007/s12065-021-00565-2>
- [43] Williams, C.K. (2021). The effect of class imbalance on precision-recall curves. *Neural Computation*, 33(4): 853-857. [https://doi.org/10.1162/neco\\_a\\_01362](https://doi.org/10.1162/neco_a_01362)
- [44] Notley, S., Magdon-Ismael, M. (2018). Examining the use of neural networks for feature extraction: A comparative analysis using deep learning, support vector machines, and k-nearest neighbor classifiers. *arXiv preprint arXiv:1805.02294*. <https://doi.org/10.48550/arXiv.1805.02294>
- [45] Novak, R., Bahri, Y., Abolafia, D. A., Pennington, J.,

- Sohl-Dickstein, J. (2018). Sensitivity and generalization in neural networks: An empirical study. arXiv preprint arXiv:1802.08760. <https://doi.org/10.48550/arXiv.1802.08760>
- [46] Hjelle, S., Mikalef, P., Altwaijry, N., Parida, V. (2024). Organizational decision making and analytics: An experimental study on dashboard visualizations. *Information & Management*, 61(6): 104011. <https://doi.org/https://doi.org/10.1016/j.im.2024.104011>
- [47] Fernández-Caramés, T.M., Blanco-Novoa, O., Froiz-Míguez, I., Fraga-Lamas, P. (2019). Towards an autonomous industry 4.0 warehouse: A UAV and blockchain-based system for inventory and traceability applications in big data-driven supply chain management. *Sensors*, 19(10): 2394. <https://doi.org/10.3390/s19102394>
- [48] Enholm, I.M., Papagiannidis, E., Mikalef, P., Krogstie, J. (2022). Artificial intelligence and business value: A literature review. *Information Systems Frontiers*, 24(5): 1709-1734. <https://doi.org/10.1007/s10796-021-10186-w>
- [49] Aikhuele, D.O., Sorooshian, S. (2024). A proactive decision-making model for evaluating the reliability of infrastructure assets of a railway system. *Information*, 15(4): 219. <https://doi.org/10.3390/info15040219>
- [50] Ren, R., Zhang, J. (2021). Semantic rule-based construction procedural information extraction to guide jobsite sensing and monitoring. *Journal of Computing in Civil Engineering*, 35(6): 04021026. [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000971](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000971)
- [51] Kang, Y. A., Gorg, C., Stasko, J. (2009). Evaluating visual analytics systems for investigative analysis: Deriving design principles from a case study. In 2009 IEEE Symposium on Visual Analytics Science and Technology, Atlantic City, NJ, USA, pp. 139-146. <https://doi.org/10.1109/VAST.2009.5333878>
- [52] Bagheri, B., Yang, S., Kao, H.A., Lee, J. (2015). Cyber-physical systems architecture for self-aware machines in industry 4.0 environment. *IFAC-PapersOnLine*, 48(3): 1622-1627. <https://doi.org/10.1016/j.ifacol.2015.06.318>