ISSN: 2093-5374 / E-ISSN: 2093-5382

Entropy Stacked Autoencoder based Diffusion Model (ESADM) and Fuzzy Clustering with Semi Supervised Fuzzy Graph Convolutional Network (FC-SSFGCN) for Rail Surface Defect Detection

Samyuktha Sasi Sekaran[1], and Dr.M. Subaji[2*]

[1]Ph.D. Research Scholar (EPT), School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, India. samyuktha.ss2015@vit.ac.in, https://orcid.org/0000-0002-5464-1511

[2*]Professor & Director, Institute for Industry and International Programmes, Vellore Institute of Technology, Vellore, India. msubaji@vit.ac.in, https://orcid.org/0000-0001-5332-4656

Received: March 07, 2024; Revised: May 25, 2024; Accepted: July 04, 2024; Published: September 30, 2024

Abstract

Rails, fasteners, and other parts of railway track lines eventually develop flaws due to continuous strain from train operations and direct exposure to the environment; these faults directly affect the safety of train operations. Detecting defects on rail surfaces presents a formidable challenge due to the wide array of possible flaws and their unpredictable nature. However, Defect identification errors, massive variances, inadequate training sample availability, and weak contrast between faults and the surrounding background all contribute to the complexity of this procedure. In this work, rail surface and fastener flaws may be detected non-destructively using a multi-crack detection approach based on the Entropy Stacked Autoencoder Diffusion Model (ESADM). A novel approach, ESAFM, has been developed, combining a rail surface image encoder with a multi-layer, Stacked Autoencoder to extract latent materials from images showcasing different types of cracks. This integration reduces the need for significant processing resources by integrating easily into the conventional physically-based image workflow. Additionally, the Zero Shot-Semi Supervised Fuzzy Class Knowledge Graph (ZS-SSFCKG) method proposes Class Knowledge Graph Construction (CKGC), which constructs a CKG to elucidate the connection between defects and non-defects. Class features are learned by using a Fuzzy Clustering with Semi Supervised Fuzzy Graph Convolutional Network (FC-SSFGCN). The method employs a transformer encoder to capture distant relationships, allowing for the extraction of features from defect samples. This facilitates the acquisition of distinct defect image characteristics, as industrial defects vary in shape and size. The experimental results on the public Railway Track Fault Detection (RTFD)and Rail Surface Defect Datasets (RSDDs) with rail surface defects are collected from rail tracks surface defect detection.

Keywords: Rail Surface Defect Detection, Entropy Stacked Autoencoder Based Diffusion Model (ESADM), Fuzzy Clustering with Semi Supervised Fuzzy Graph Convolutional Network (FC-SSFGCN), and Class Knowledge Graph (CKG).

Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA), volume: 15, number: 3 (September), pp. 17-35. DOI: *10.58346/JOWUA.2024.I3.002*

*Corresponding author: Professor & Director, Institute for Industry and International Programmes, Vellore Institute of Technology, Vellore, India.

1 Introduction

As train speeds and capacities continue to increase rapidly, the safety standards for railway operations are escalating (Madhan & Shanmugapriya, 2024; Jiang et al., 2018). Rails, as a critical component of train operations and railroad system maintenance, undergo irreversible defects over time. Once these defects form, they tend to escalate quickly, posing a significant threat to train safety. Failure to detect and address these defects promptly can result in severe incidents such as broken rails, endangering both passengers and causing substantial economic losses. Therefore, timely identification and warning of defects by railway personnel are crucial to mitigate potential hazards and ensure safe train operations (Jessop et al., 2016).

Rail and its fasteners undergo various pressures during service, including those from train wheel-rail contact, environmental factors, and material aging (Wang et al., 2012; Usamentiaga et al., 2022). Traditional manual inspection methods dominate railway inspection, but they suffer from subjectivity, inefficiency, prolonged duration, high expenses, and susceptibility to harsh external conditions. While manual inspection offers simplicity and affordability, it is plagued by low efficiency, high error rates, and limited real-time capabilities. Consequently, non-destructive testing methods have gained popularity in railway systems, employing techniques such as three-dimensional scanning, current testing, and acoustic wave analysis. Thus, there is great practical value and research significance in developing automated rail surface flaw identification.

There are two primary categories into which machine vision methods for rail surface fault detection may be divided: Deep learning techniques as well as conventional machine learning techniques (Chenariyan Nakhaee et al., 2019). Traditional machine learning techniques involve using cameras to automatically identify defects on rail surfaces. Rail defect images must be manually analysed since these approaches usually depend on manually created or predetermined attributes. Subsequently, feature learning algorithms are proposed for classification purposes. While traditional machine learning approaches have advanced rail surface defect detection to a certain degree, they often fall short in adequately extracting defect features, especially for small-sized targets where detection accuracy tends to be low (Yang et al., 2020). In contrast, deep learning has seen rapid advancements in recent years and is increasingly recognized as a preferable option for rail surface defect detection due to its inherent advantages in feature representation and modelling capabilities (Varshavardhini & Rajesh, 2023; Rampriya et al., 2021).

However, in real-world scenarios, new defect types in products often emerge since the production environment is complex and constantly changing. The pre-trained model is anticipated to function with excellent accuracy on these new classes of defects as well since these unique classes of defects were not encountered during model training (Varshavardhini & Rajesh, 2023; He et al., 2022). This introduces a far more difficult commercial problem: the model has to generalize to these new classes, which is called zero-shot recognition (Suleiman, 2023). While there is a wealth of literature on zero-shot recognition, primarily focusing on natural images, railway track surface defect images have been largely overlooked (Levchenko et al., 2020). Additionally, the surface defect dataset often has an excessive number of replicate nodes created by current techniques to construct knowledge graphs for the classes in these investigations. Railway track detection involves the identification and segmentation of several cracks, making it a crucial aspect of analysing railway images (Rathi et al., 2024; Wei et al., 2019). Given the high speeds of trains, the research emphasis is biased toward real-time detection of rail surface defects, necessitating rapid network detection speeds. Consequently, a single-stage target detection algorithm is chosen to address this need (Samyuktha & Subaji, 2024; Kocbek & Gabrys, 2019).

This study introduces a multi-object detection approach called the Entropy Stacked Autoencoder Based Diffusion Model (ESADM) for non-destructive detection of defects on rail surfaces and fasteners. ESAFM is used to extract latent characteristics from images showing various sorts of faults, which combines a multi-layered Stacked Autoencoder with a rail surface image encoder. Furthermore, Correlations are built utilizing the Class Knowledge Graph Construction (CKGC) technique between issues and non-defects. The trials' findings demonstrate that the recommended model works better and finds defects more precisely (Popov et al., 2022).

2 Literature Review

Yanan et al., (2018) introduced a deep learning method for identifying flaws in rail surfaces that makes use of the YOLOv3 algorithm. The technique determines the width, height, and center point coordinates of defects according to their location inside cells using dimensional clustering, then normalizes the coordinates appropriately. Next, bounding box item scores are predicted using logistic regression, and possible bounding box categories are predicted with the help of binary cross-entropy loss. Next, confidence is calculated to make predictions easier. For the purpose of identifying flaws in rail surfaces, this approach has several advantages. According to (Yu et al., 2018) a coarse-to-fine model (CTFM) was proposed to discover faults at various sizes. Within the CTF framework, this model functions on three different levels: subimage, region, and pixel levels. First, a background subtraction approach uses longitudinal row consistency at the subimage level to efficiently separate possible subimages with defects by filtering out defect-free regions. Then, drawing inspiration from the ideas of visual saliency, the region extraction approach employs phase-only Fourier transforms to pinpoint definitive fault locations. Tests performed at the pixel-level and defect-level indices show that CTFM performs better than the state-of-the-art techniques.

Jin et al., (2019) presented a deep multimodal RIS (DM-RIS) for surface defect detection that use the Faster Region-Based Convolutional Neural Network (RCNN) for defect localization and a quick and robust spatially limited Gaussian mixture model for segmentation (Kutlu & Camgözlü, 2021). These elements function in parallel inside a parallel structure. A kind of Deep Convolutional Neural Network (DCNN) called SegNet architecture was introduced (Liang et al., 2018) and is used to identify surface flaws on rails. The technology takes images of surface flaws in rails and feeds them into a 59-layer training network that is trained on 120 rail images. This network outperforms conventional image threshold segmentation techniques in terms of effectiveness, precision, and interference-free defect detection.

James et al., (2018) suggested a multi-stage deep learning algorithm first to concentrate on the area of interest and then a segment images. Subsequently, this cropped image is passed to a binary image classifier, which determines which warnings are accurate and which are not. This method has shown improved detection performance by lowering the false alarm rate. By building on the DeepLab v3+ deep learning system, (Xia-Ting et al., 2019) produced DeeperLab, a flexible and lightweight Bayesian alternative. This modification is intended especially for the purpose of identifying flaws on complex and diverse rail surfaces. In particular, Dropout is integrated into the enhanced Xception network for posterior distribution-based Monte Carlo sampling. Multiple scales and speeds of extraction of the dense features are achieved by use of Atrous Spatial Pyramid Pooling (ASPP). Moreover, a more straightforward and effective decoder is suggested to enhance the defect edges, and it generates segmentations and uncertainty of the Softmax probability mean and variances.

Feng et al., (2020) provided a unique object recognition method for rail defect detection. The algorithm's network architecture incorporates a MobileNet backbone network with distinct detection layers with multi-scale feature maps, drawing inspiration from both YOLO and feature pyramid networks. The effectiveness of the algorithm in identifying defects is assessed using two distinct MobileNet architectures. The experiment findings show how promising the proposed technique is; it shows good accuracy and quick inference rates, making it very applicable in industrial settings. In order to extract contextual information (Zhang et al., 2020) proposed using a Long Short-Term Memory (LSTM) network by an OC-IAN and OC-TD, while a One-Dimensional Convolutional Neural Network (ODCNN) is utilized for feature extraction (Camgözlü & Kutlu, 2023). The primary distinction is in the design: OC-TD has a double-branch architecture instead of the attention module seen in OC-IAN.

Processing the Gabor magnitude image is the initial step for extracting first-order statistical data in the study (Thendral & Ranjeeth, 2021). In order to discern between track images that have been cracked and those that have not, these attributes are then fed into a deep learning neural network. A deep classifier network is used to recognize items ahead of a train on a railway track, according to (Kapoor et al., 2022). 2-D Singular Spectrum Analysis (SSA) is a decomposition approach that breaks down the image into useful components to help with this procedure. The deep classifier network then receives these inputs. The deep network's identification ability for identifying complications is greatly enhanced by the combination of 2D-SSA. This approach also presents a new metric for railroad track identification. A system like this might help lower the financial load and number of railway accidents. The approach's results show that obstructions on the railway track may be recognized with high accuracy and effectiveness, which improves railway safety.

Wang et al., (2022) introduced a novel rail defect detection network leveraging on Mask Region-based Convolutional Neural Network (Mask R-CNN). This detection network is equipped with a fresh feature pyramid for merging multiple scales. Furthermore, to overcome the inadequacies of IOU in specific circumstances, a new assessment measure named Complete Intersection over Union (CIoU) is added into the region proposal network. In order to deal with the problem of limited faulty datasets, both data augmentation and transfer learning approaches are used during the training phase. A novel Rail Boundary Guidance Network (RBGNet) is presented (Wu et al., 2022) in order to identify notable Rail Surfaces (RS). At first, In order to fully use the synergy between Rail Edges (RE) and RS, a unique design is proposed that facilitates the accurate identification of RS with different boundaries. In order to steer the network and comprehend the shift between the input and ground truth, RBGNet is then equipped with a novel hybrid loss that combines Binary Cross Entropy (BCE), Structural Similarity Index Measure (SSIM), and IoU. Ultimately, tests carried out on a demanding Unmanned Aerial Vehicle (UAV) rail dataset show how effectively the system detects objects and adjusts to difficult settings.

Liu et al., (2022) presented A pyramid feature called Convolutional Neural Net (CNN) is employed to detect surface flaws on railroads. Just 40% of the dataset is used to train the network utilizing IoU loss functions in addition to binary cross-entropy. Experiments on the RSDD dataset are used to evaluate this approach's efficacy in comparison to other approaches. Ye et al., (2023) proposed a novel approach to identify rail surface problems using laser technology and 3-D pixel-level analysis, combining deep semantic segmentation principles with accurate laser measurements. Initially, the train surface is scanned in three dimensions using a moderately priced two-dimensional laser triangulation sensor. Next, a newly developed deep semantic segmentation network with fully convolutional segmentation and dual symmetric mapping is demonstrated. This network can handle 3-D laser measurement data as input and output pixel-level defect detection findings from beginning to end with ease.

Luo et al., (2023) stated that the rail surface fault detection system is based on an enhanced version of YOLOv5s. To improve the dataset of photos of rail surface flaws, random cropping, flip transformations, and brightness adjustments are first applied. After that, a Conv2D and Dilated Convolution (CDConv) module is made to lessen network processing. Additionally, by integrating the Swin Transformer with the Neck and Backbone ends, the original network's C3 module is enhanced. Next, the Global Attention Mechanism (GAM) and the Path Aggregation Network (PANet) are integrated to build a new prediction head called the Swin Transformer and GAM Prediction Head (SGPH). Finally, the Soft-SIoUNMS (Scale-sensitive Intersection over Union Non-Maximum Suppression) loss is used in place of the original CIoU loss, accelerating algorithm convergences while minimizing regression errors. Zero-Shot Class Knowledge Graph (ZS-CKG), which was suggested (Li et al., 2022), is a technique that is provided for surface defect diagnosis in real-world scenarios. This approach identifies associations between defect classes that have previously been recognized and those that haven't by using the CKGC methodology to generate a class knowledge graph. A graph convolutional neural network is then used to learn the class characteristics. Because industrial faults come in a variety of shapes and sizes, this method uses a transformer encoder to capture broad dependencies that allow for the extraction of discriminative characteristics from defect samples.

3 Proposed Methodology

The Entropy Stacked Autoencoder Based Diffusion Model (ESADM) is a multi-crack detection technique that this work proposes for the non-destructive diagnosis of fastener and rail surface flaws. Using a multi-layer Stacked Autoencoder and a rail surface image encoder, ESADM extracts latent representations from photos with different kinds of fractures. Furthermore, the ZS-SSFCKG approach is put out, which integrates Class Knowledge Graph Construction (CKGC) to create relationships between fault classes. Class characteristics are learnt using a semi-supervised fuzzy graph convolutional network (FC-SSFGCN) in conjunction with a fuzzy clustering technique. The overall framework of the proposed system is illustrated in Figure 1.

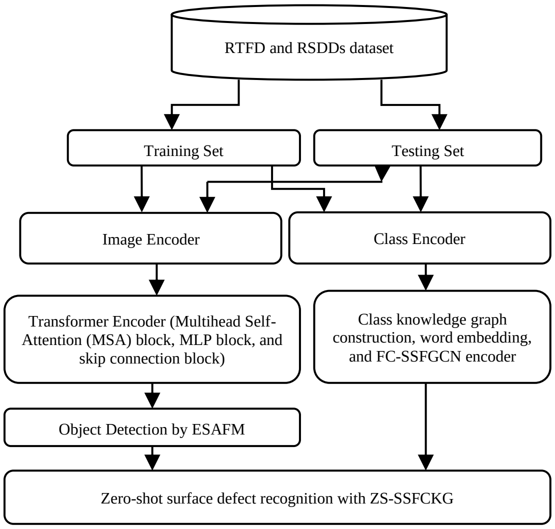
RTFD and RSDDs dataset
Training Set
Testing Set
Image Encoder
Class Encoder
Transformer Encoder (Multihead Self-Attention (MSA) block, MLP block, and skip connection block)
Class knowledge graph construction, word embedding, and FC-SSFGCN encoder
Object Detection by ESAFM
Zero-shot surface defect recognition with ZS-SSFCKG

Figure 1: Architecture Diagram of Surface Defect Recognition

1) Dataset Description

Two benchmark datasets like RTFD and RSDDs has been introduced for surface defect detection.

RTFD: Railway track problem detection data was gathered from https://www.kaggle.com/datasets/salmaneunus/data. With 384 images for training, testing, and validation, it includes all incorrect and non-defective images. Both classes have equal number of samples.

RSDDs: The RSDDs dataset contains two categories of rail defect data: Types I and II. 67 fault images from high-speed rail tracks are the source of Type-I flaws, while 128 defect images from conventional or heavy-duty transportation tracks are the source of Type-II problems. Furthermore, a finite number of randomly chosen defect samples from the test set are needed by the model for the training phase. Type-I and Type-II rail images are downsized to 64 × 64 and 160 × 160, respectively, after dividing and resizing. This dataset is accessible at https://github.com/neu-rail-rsdds/rsdds.

2) Proposed Model for Surface Defect Detection

Figure 2 illustrates the components of the proposed model, which include an image encoder, a class encoder, and a classifier. Firstly, the image encoder function encodes an input image into an image attribute. Secondly, the class encoder function encodes classes into class attributes. The classification function then utilizes the image feature and class value to determine the image label. The proposed architecture consists of two phases: training and testing, illustrated in Figure 2.

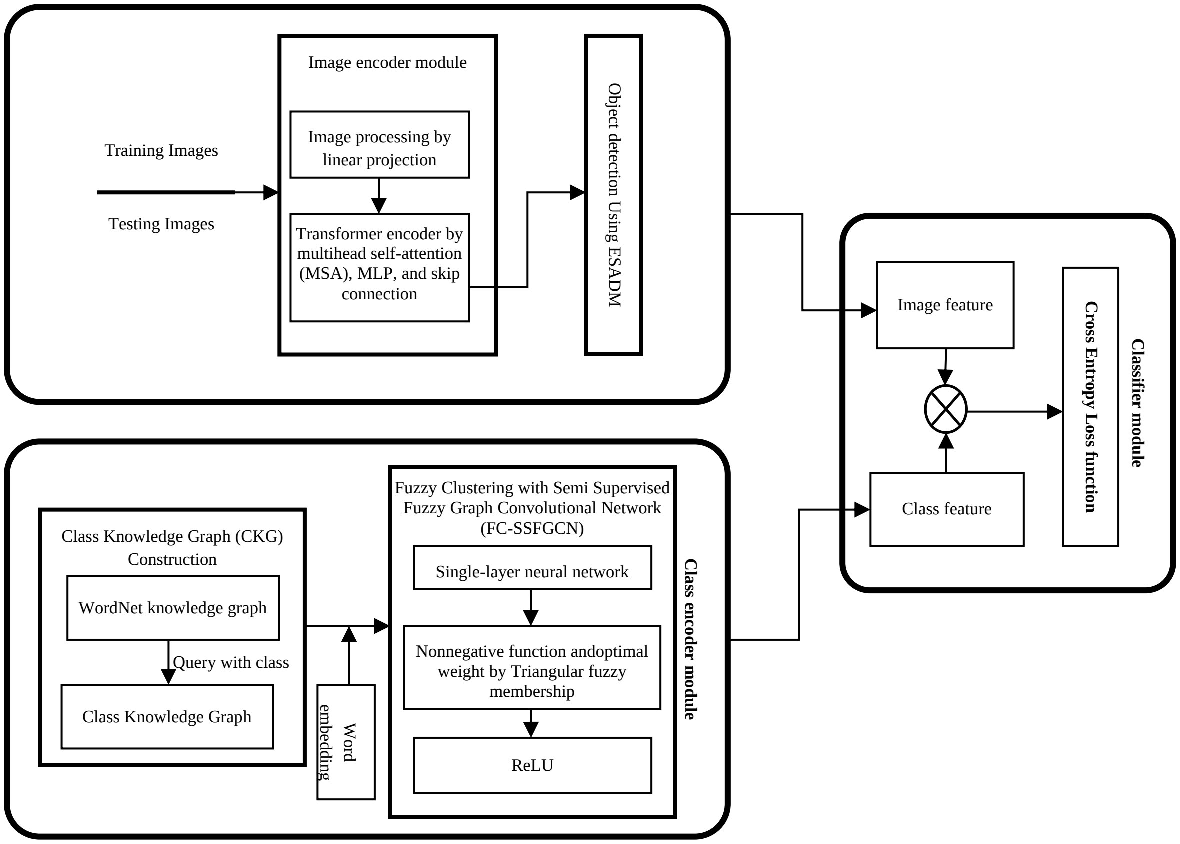
Training Images
Testing Images
Image encoder module
Image processing by linear projection
Transformer encoder by multihead self-attention (MSA), MLP, and skip connection
Object detection Using ESADM
Image feature
Class feature
Cross Entropy Loss function
Classifier module
Class Knowledge Graph (CKG) Construction
WordNet knowledge graph
Query with class
Class Knowledge Graph
Word embedding
Fuzzy Clustering with Semi Supervised Fuzzy Graph Convolutional Network (FC-SSFGCN)
Single-layer neural network
Nonnegative function andoptimal weight by Triangular fuzzy membership
ReLU
Class encoder module

Figure 2: Proposed Multi-Crack Detection Model

The image encoder module is responsible for extracting image features, using components for transformer encoding and image processing. Meanwhile, the class encoder module uses word embedding to initialize node features, CKGC to build a class knowledge graph, and the GCN encoder module to learn node features. Lastly, the classifier module predicts the class data based on the class score, which it computes by integrating the characteristics of the input image with the class features.

- **Image Encoder Module**

Figure 2 (Kampffmeyer et al., 2019) the transformer encoder component and the image processing component make up the image encoder module. The transformer encoder needs the image processed into a certain input form, which is handled by the image processing component. During the training and testing phases, there is no cross-section amongst the input image classes. To extract individuality, the transformer encoder transmits the previously analyzed image.

Image Processing Module

The image is divided into N fixed-size series patches by the image processing element, and a linear block embeds the input patches into a 1D latent variable. Subsequently, every patch embedding and the specific token have position embeddings, and all patch embeddings have specific tokens added to the front. Thus, the image processing function processes conversion of the image to the transformer encoder's required input format.

Transformer Encoder

Transformer blocks are arranged in L layers within the transformer encoder system. The transformer block consists of the skip connection block, MLP block, Multi head Self-Attention (MSA), and normalization (Norm). le_0 is presently employed to denote an input image $x \in \mathbb{R}^{\mathcal{H}\times\mathcal{W}\times\mathcal{C}}$ that is supplied into the transformer encoder element. First, by using the connection blocks for Norm, MSA, and skip, le_0 is denoted as le'_l, and the method is given by equation (1). There, le'_l denotes the l^{th} layer of the transformer block. Secondly, employing MLP, Norm, and skip connection blocks to denote le_l, the procedure may be expressed as equation (2). In image feature, le_0 is ultimately incorporated as le_lover the [class] token and transformer encoderle $^{0}_{\mathrm{tb}}$ is assigned as x' utilizing the Norm block; this procedure may be summed up as equation (3). The transformer encoder module operation stated as follows.

$$\mathrm{le}_\mathrm{l}^{'} = \mathrm{MSA}(\mathrm{Norm}(\mathrm{le}_{\mathrm{l}-1})) + \mathrm{le}_{\mathrm{l}-1}, \forall \mathrm{l} = 1 \cdots \mathrm{tb} \quad (1)$$

$$\mathrm{le}_\mathrm{l} = \mathrm{MLP}(\mathrm{Norm}(\mathrm{le}_\mathrm{l}^{'})) + \mathrm{le}_\mathrm{l}^{'}, \forall \mathrm{l} = 1 \cdots \mathrm{tb} \quad (2)$$

$$\mathrm{x}' = \mathrm{Norm}(\mathrm{le}_{\mathrm{tb}}^{0}) \quad (3)$$

Where tb is the quantity of transformer blocks. The input image is extracted during the phases of the transformer encoder and image processing$x \in \mathbb{R}^{\mathcal{H}\times\mathcal{W}\times\mathcal{C}}$as a 1D vector $\mathrm{x}' \in \mathbb{R}^{\mathcal{D}}$.

Object Detection Using ESADM

The Diffusion Model (DM) aims to understand target defect objects, referred to as p(x), using iterative noise removal on a conventional normal distribution variable. By repeatedly adding noise $\epsilon \sim N(0, 1)$ to an initial input image x, this approach interprets diffusion as a preset Markov chain with a length of T. The equation (4) is given as,

$$x_t = \alpha_t x + \beta_t \epsilon \quad (4)$$

t is equally sampled from 1...T noise region is defined by hyperparameters α_t and β_t. The learning purpose of widespread score-matching DM and the equation (5) is given as,

$$L_{DM} = \mathbb{E}_{x,\epsilon\sim\mathcal{N}(0,1),t}[||\epsilon - \epsilon_\theta(x_t, t)||_2^2] \quad (5)$$

Where an equally weighted denoising autoencoder trained to anticipate a solution of x_{t-1} from x_t is indicated by the symbol $\epsilon_\theta(x_t, t)$. With x representing the image representation and c the condition, the goal is to construct a conditional distribution, p(x|c). Such conditional distributions may be modeled by DM by using the following learning goal to train a conditional denoising backbone, $\epsilon_\theta(x_t, t, \tau_\theta(c))$. The equation (6) is given as,

$$L_{DM} = \mathbb{E}_{x,c,\epsilon\sim\mathcal{N}(0,1),t}[||\epsilon - \epsilon_\theta(x_t, t, \tau_\theta(c))||_2^2] \quad (6)$$

Where c is transformed into an intermediate representation by the domain-specific encoder τ_θ.

The amount of x has a major impact on the computing cost of the denoising backbone of the Diffusion Model (DM), which may represent a conditional distribution p(x|c). On the other hand, diffusion and denoising are carried out inside a perceptually compressed latent space by the Latent Diffusion Model (LDM), which usually uses a pre-trained encoder E and decoder D. Accordingly, LDM operates in the latent domain p(z), where D may be used to decode z to its matching x.The input, hidden, and output layers make up the three layers of an autoencoder, a form of unsupervised learning architecture. An encoder and a decoder are the two halves of an autoencoder's training process. Based on the encoder's hidden representation, the decoder reconstructs the input data. With the railway track dataset input that is not labeled $\{x_n\}_{n=1}^N$, where $x_n \in \mathbb{R}^{m\times 1}$, The hidden encoder vector, denoted by hn, is determined usingx_n, and $\hat{x}_n$represents the output layer's decoder vector. Therefore, the following is the encoding process. The equation (7) is given as,

$$h_n = f(W_1 x_n + b_1) \quad (7)$$

Where b_1 is the bias vector, W_1 is the encoder's weight matrix, and f is the encoding function by encoder E. Here is a definition of the decoder process. The equation (8) is given as,

$$\hat{x}_n = g(W_2 h_n + b_2) \quad (8)$$

Where the weight matrix of the decoder is W_2, the bias vector is b_2, and g is the decoding function by decoder D. Weight value of encoder and decoder is computed using the entropy. It is defined as follows. The equation (9,10) is given as,

$$E_k = -\frac{1}{\ln{(mn)}}\sum_{i=1}^{m}\sum_{j=1}^{n} p_{ij}^k \ln p_{ij}^k, k \in T \quad (9)$$

$$p_{ij}^k = \frac{u_{ij}^k}{\sum_{i=1}^{m}\sum_{j=1}^{n} u_{ij}^k} \&0 \leq E_k \leq 1 \quad (10)$$

m, n is denoted as the row and column. To reduce the reconstruction error, the auto encoder's parameter settings are improved. The equation (11) is given as,

$$\phi(\Theta) = \underset{\theta,\theta'}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{n} L(x^i, \hat{x}^i) \quad (11)$$

Where a loss function is represented by L. The equation (12) is given as,

$$L(x, \hat{x}) = ||x - \hat{x}||^2 \quad (12)$$

Figure 3 illustrates the configuration of a Stacked Autoencoder (SAE), it uses an unsupervised layer wise learning strategy to stack n autoencoders into n hidden layers and then uses a supervised algorithm to fine-tune.

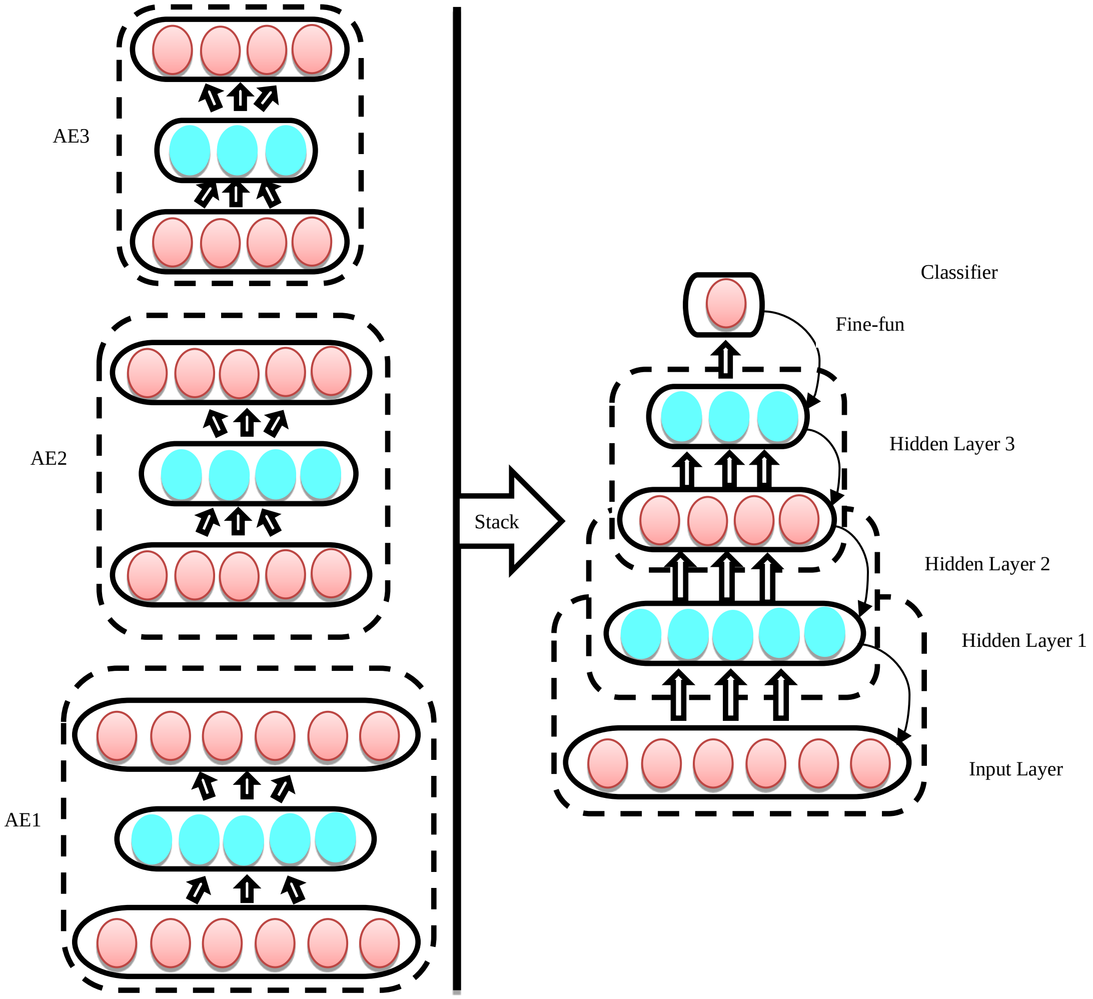

Figure 3: Stacked Autoencoder (SAE) Structure

Three phases make up the SAE-based strategy (Yu et al., 2022; Wan et al., 2019). To obtain the learned feature vector, (1) train the initial autoencoder using input data. (2) Repeating this until training is complete, using the feature vector from the previous layer as input for the next layer. (3) The Backpropagation (BP) approach minimizes the cost function and fine-tunes weights using the labeled training set after training all hidden layers.

- **Class Encoder Module**

Class encoder system includes the word embedding, FC-SSFGCNencoder system, and CKGC. The defect CKG is generated utilizing the surface defect class data provided by the CKGC function. The nodes in the generated CKGC have their attributes prepared by the word embedding function. When node properties and the defect CKG are supplied into the FC-SSFGCNencoder part it may acquire every node visual representation.

Class Knowledge Graph (CKG) Construction

By requesting the nodes in a current knowledge graph that are associated with the defect class, this component creates the defect CKG. The connection amongst the base and novel defect classes is established utilizing the CKG. Utilizing the WordNet knowledge graph, the ZS-CKG will generate a CKG.

Word Embedding

CKG node parameters are to be initialized in this part. Every node in the CKG has its attribute generated utilizing word embeddings that were pre-trained on extensive language datasets. The ZS-CKG uses GloVe, which has been pre-trained on Gigaword 5 and Wikipedia 2014, to map each class into a 300-D vector form. An unsupervised learning system called GloVe is employed to generate vector illustrations for words. After training on consolidated global word-word combination statistics from a corpus, intriguing linear components of the word vector space are displayed in the final visualizations.

Fuzzy Clustering with Semi Supervised Fuzzy Graph Convolutional Network (FC-SSFGCN) Encoder

The class knowledge graph and the node initialization features are the inputs used by the FC-SSFGCN encoder module to extract node features. It is composed of an activation function block and a graph convolutional block. Let $\mathrm{X} = (\mathrm{x}_1, \mathrm{x}_2, \ldots., \mathrm{x}_\mathrm{n}) \in \mathbb{R}^{\mathrm{n}\times\mathrm{p}}$ represent the group of n p-dimensional image vectors. Let G(X,A) be X represented on a graph using $\mathrm{A} \in \mathbb{R}^{\mathrm{n}\times\mathrm{n}}$ representing the pairwise connections (neighborhoods of similarity) between image X. Propagation (hidden) layers consist of many layers, one final perceptron layer, and one input layer in GCN. A unique network architecture called FC-SSFGCN simultaneously combines graph learning and graph convolution. $\mathrm{X} = (\mathrm{x}_1, \mathrm{x}_2, \ldots., \mathrm{x}_\mathrm{n}) \in \mathbb{R}^{\mathrm{n}\times\mathrm{p}}$ to search for a function that is not negative $\mathrm{S}_{\mathrm{ij}} = \mathrm{g}(\mathrm{x}_\mathrm{i}, \mathrm{x}_\mathrm{j})$. It illustrates the link between images pairwise x_i & x_j. $\mathrm{g}(\mathrm{x}_\mathrm{i}, \mathrm{x}_\mathrm{j})$ is specified by a weight vector using a single-layer neural networka $= \left(\mathrm{a}_1, \mathrm{a}_2, \ldots., \mathrm{a}_\mathrm{p}\right)^{\mathrm{T}} \in \mathbb{R}^{\mathrm{p}\times 1}$. Formally, understand the following graph S. The equation (13) is given as,

$$\mathrm{S}_{\mathrm{ij}} = \mathrm{g}(\mathrm{x}_\mathrm{i}, \mathrm{x}_\mathrm{j}) = \frac{\exp\left(\mathrm{ReLU}\left(\mathrm{a}^{\mathrm{T}}\left|\mathrm{x}_\mathrm{i} - \mathrm{x}_\mathrm{j}\right|\right)\right)}{\sum_{\mathrm{j}=1}^{\mathrm{n}} \exp\left(\mathrm{ReLU}\left(\mathrm{a}^{\mathrm{T}}\left|\mathrm{x}_\mathrm{i} - \mathrm{x}_\mathrm{j}\right|\right)\right)} \quad (13)$$

The nonnegativity of S_{ij} is ensured by the activation function $\mathrm{ReLU}(.) = \max(0,.)$. Assuring that the learned graph S is what the softmax operation does on each row of S. The equation (14) is given as,

$$\sum_{\mathrm{j}=1}^{\mathrm{n}} \mathrm{S}_{\mathrm{ij}} = 1\,, \mathrm{S}_{\mathrm{ij}} \geq 0 \quad (14)$$

Reduce the loss function to maximize the weight vector a and the equation (15) is given as,

$$\mathcal{L}_{\mathrm{G}} = \sum_{\mathrm{i,j}=1}^{\mathrm{n}} \left\|\mathrm{x}_\mathrm{i} - \mathrm{x}_\mathrm{j}\right\|_2^2 \mathrm{S}_{\mathrm{ij}} + \gamma\|\mathrm{S}\|_{\mathrm{F}}^2 \quad (15)$$

$\mathrm{S}_{\mathrm{ij}}$is encouraged to be lower when the distance between image points x_i and x_j is larger $\left\|\mathrm{x}_\mathrm{i} - \mathrm{x}_\mathrm{j}\right\|_2^2$. A projection matrix $\mathrm{P} \in \mathbb{R}^{\mathrm{p}\times\mathrm{d}}$, d < p, is used to parameterize the implementation of a single-layer low-dimensional embedding network. Specifically, the following describes final graph learning. The equation (16) is given as,

$$\mathrm{S_{ij}} = \mathrm{g}(\tilde{\mathrm{x}}_\mathrm{i}, \tilde{\mathrm{x}}_\mathrm{j}) = \frac{\mathrm{A_{ij}} \exp\left(\mathrm{ReLU}\left(\mathrm{a^T}\left|\tilde{\mathrm{x}}_\mathrm{i} - \tilde{\mathrm{x}}_\mathrm{j}\right|\right)\right)}{\sum_{\mathrm{j}=1}^{\mathrm{n}} \mathrm{A_{ij}} \exp\left(\mathrm{ReLU}\left(\mathrm{a^T}\left|\tilde{\mathrm{x}}_\mathrm{i} - \tilde{\mathrm{x}}_\mathrm{j}\right|\right)\right)} \quad (16)$$

Where the initial graph is represented by A. $\mathrm{A_{ij}}$= 1 may be set in the aforementioned update rule if it is unavailable. When the loss function is transformed into and the equation (17) is given as,

$$\mathcal{L}_{\mathrm{GL}} = \sum_{\mathrm{i,j}=1}^{\mathrm{n}} \left\|\tilde{\mathrm{x}}_\mathrm{i} - \tilde{\mathrm{x}}_\mathrm{j}\right\|_2^2 \mathrm{S_{ij}} + \gamma\|\mathrm{S}\|_\mathrm{F}^2 \quad (17)$$

The suggested graph, denoted as S, possesses a specific probability attribute where the optimal $\mathrm{S_{ij}}$indicates the likelihood of image $\mathrm{x_j}$ being linked to $\mathrm{x_i}$ as a neighboring node. Graph convolution layers consist of many layers, a final Perceptron layer, and a single graph learning layer in FC-SSFGCN. Using the adaptive neighbor graph S that the graph learning layer provides, it applies a layer-wise propagation concept inside the convolutional layers. The equation (18) is given as,

$$\mathrm{X^{(k+1)}} = \sigma\left(\mathrm{D_s^{-\frac{1}{2}} S D_s^{-\frac{1}{2}} X^{(k)} W^{(k)}}\right) \quad (18)$$

Where $\mathrm{k} = 0, 1 \ldots., \mathrm{K} - 1$. $\mathrm{D_s} = \mathrm{diag}(\mathrm{d_1}, \mathrm{d_2}, \ldots\ldots, \mathrm{d_n})$ is a matrix containing diagonal elements that are diagonal $\mathrm{d_i} = \sum_{\mathrm{j}=1}^{\mathrm{n}} \mathrm{S_{ij}}$. $\mathrm{W^{(k)}} \in \mathbb{R}^{\mathrm{d_k} \times \mathrm{d_{k+1}}}$is each convolution layer's unique trainable weight matrix.(.) indicates a function that activates $\mathrm{ReLU}(,) = \max(0,.)$, & $\mathrm{X^{(k+1)}} \in \mathbb{R}^{\mathrm{n} \times \mathrm{d_{k+1}}}$ is represented as the result of the kth layer's activations. Graph S that was learned satisfy $\Sigma_\mathrm{j}\mathrm{S_{ij}} = 1, \mathrm{S_{ij}} \geq 0$, here is a simplified version of equation (19).

$$\mathrm{X^{(k+1)}} = \sigma\left(\mathrm{S X^{(k)} W^{(k)}}\right) \quad (19)$$

The last perceptron layer should be defined as follows for semi-supervised classification. The equation (20) is given as,

$$\mathrm{Z} = \mathrm{softmax}\left(\mathrm{S X^{(K)} W^{(K)}}\right) \quad (20)$$

Where the number of classes is indicated by c and $\mathrm{W^{(K)}} \in \mathbb{R}^{\mathrm{d_K} \times \mathrm{c}}$. The label prediction of the FC-SSFGCN network is shown by the final output $\mathrm{Z} \in \mathbb{R}^{\mathrm{n} \times \mathrm{c}}$, where $\mathrm{Z_i}$ is the label prediction for the ith node in each row. By reducing the losses function, all of the network parameters, represented by $\Theta = \{\mathrm{P, a, W^{(0)}}, \ldots\ldots, \mathrm{W^{(K)}}\}$, are jointly trained. The equation (21) is given as,

$$\mathcal{L}_{\mathrm{Semi-FC-SSFGCN}} = \mathcal{L}_{\mathrm{Semi-SSFGCN}} + \lambda\mathcal{L}_{\mathrm{GL}} \quad (21)$$

Where equations (17–20) determine $\mathcal{L}_{\mathrm{GL}}$ and $\mathcal{L}_{\mathrm{Semi-SSFGCN}}$. One trade-off parameter is $\lambda \geq 0$. It is observed that when λ = 0, the optimum graph S is similarly feasible in FC-SSFGCN and is learned exclusively from the labeled image (i.e., cross-entropy loss). GCN specifies the final perceptron layer for semi-supervised classification as follows. The equation (22-24) is given as,

$$\mathrm{Z} = \mathrm{softmax}\left(\mathrm{D^{-\frac{1}{2}} A D^{-\frac{1}{2}} X^{(K)} W^{(K)}}\right) \quad (22)$$

Where $\mathrm{W^{(K)}} \in \mathbb{R}^{\mathrm{d_K} \times \mathrm{c}}$ and the number of classes is indicated by c. The resultant product$\mathrm{Z} \in \mathbb{R}^{\mathrm{n} \times \mathrm{c}}$ is the label prediction for every image X, where the label prediction for the ith node is indicated by each row $\mathrm{Z_i}$. Constrained to a range of 0 to 1, the fuzzy membership function represents the degree of similarity between the location's image value and the ideal image value. The {a, b, c} parameters define the triangular fuzzy membership function, where the membership function is assigned to each value image x.

$$\mu_{\mathrm{triangle}}(\mathrm{x}, \mathrm{a}, \mathrm{b}, \mathrm{c}) = \begin{cases} 0, \mathrm{x} \leq \mathrm{a} \\ \frac{\mathrm{x} - \mathrm{a}}{\mathrm{b} - \mathrm{a}}, \mathrm{a} \leq \mathrm{x} \leq \mathrm{b} \\ \frac{\mathrm{c} - \mathrm{x}}{\mathrm{c} - \mathrm{b}}, \mathrm{b} \leq \mathrm{x} \leq \mathrm{c} \\ 0, \mathrm{c} \leq \mathrm{x} \end{cases} \quad (23)$$

$$\mathcal{L}_{\mathrm{Semi-SSFGCN}} = -\sum_{\mathrm{i} \in \mathcal{L}} \sum_{\mathrm{j}=1}^{\mathrm{c}} \mathrm{Y}_{\mathrm{ij}} \ln \mathrm{Z}_{\mathrm{ij}} \quad (24)$$

Fuzzy membership function is introduced to minimize the cross-entropy loss function, the optimum weight matrices $\{\mathrm{W}^{(0)}, \ldots\ldots, \mathrm{W}^{(\mathrm{K})}\}$, $\mathcal{L}$ denoted as the set of nodes that have been labeled (Kipf & Welling, 2016). The process iterates through membership and centroid while minimizing the objective function to provide clustering results. FCM has the following design. The equation (25) is given as,

$$\mathrm{J}_{\mathrm{FCM}}(\mathrm{u}, \mathrm{v}) = \sum_{\mathrm{i}=1}^{\mathrm{c}} \sum_{\mathrm{j}=1}^{\mathrm{n}} \mathrm{u}_{\mathrm{ij}}^{\mathrm{m}} \mathrm{D}_{\mathrm{ij}}^{2}, \mathrm{m} > 1 \quad (25)$$

where m>1 applies to the fuzzy exponent m, and $\mathrm{D}_{\mathrm{ij}}^{2} = \left\|\mathrm{x}_{\mathrm{j}} - \mathrm{v}_{\mathrm{i}}\right\|^{2}$ is the definition of the Euclidean distance. Reduced objective function (25) may lead to membership. Iterative membership and centroid functions make up equations (26–27).

$$\mathrm{u}_{\mathrm{ij}} = \left[\sum_{\mathrm{k}=1}^{\mathrm{c}} \left(\frac{\mathrm{D}(\mathrm{x}_{\mathrm{j}}, \mathrm{v}_{\mathrm{i}})}{\mathrm{D}(\mathrm{x}_{\mathrm{j}}, \mathrm{v}_{\mathrm{k}})}\right)^{\left(\frac{2}{\mathrm{m}-1}\right)}\right]^{-1} \quad (26)$$

$$\mathrm{v}_{\mathrm{i}} = \frac{\sum_{\mathrm{j}=1}^{\mathrm{n}} \mathrm{u}_{\mathrm{ij}}^{\mathrm{m}} \mathrm{x}_{\mathrm{j}}}{\sum_{\mathrm{j}=1}^{\mathrm{n}} \mathrm{u}_{\mathrm{ij}}^{\mathrm{m}}} \quad (27)$$

The approach is subject to the following three restrictions, notwithstanding the improved clustering performance: $\mathrm{u}_{\mathrm{ij}} \in [0, 1], (\mathrm{i} = 1, 2, \ldots, \mathrm{c}, \mathrm{j} = 1, 2, \ldots, \mathrm{n}), \sum_{\mathrm{i}=1}^{\mathrm{c}} \mathrm{u}_{\mathrm{ij}} = 1$ and $0 < \sum_{\mathrm{i}=1}^{\mathrm{c}} \mathrm{u}_{\mathrm{ij}} < n$, $(j = 1, 2, \ldots, \mathrm{n})$.

Classifier Module Using GCN

The image feature obtained from the image encoder system was multiplied by the class feature supplied by the class encoder system by the classifier system. To obtain the label of input image, argmax the score. The transformer encoder function extracts batch images. Graph neural networks that function with rich relational graph data between pieces are known as Graph Convolutional Network (GCN). Label information may migrate from labeled to unlabeled nodes through the GCN, which distributes label and feature data among connected nodes. GCN-based techniques have recently shown impressive results on knowledge graphs. GCN-based techniques have demonstrated notable efficiency and attracted increased attention in knowledge graphs. GCN is employed by the knowledge graph-based Zero-Short recognition to effectively encode categories as matrices.

4 Results and Discussion

A single NVIDIA RTX2080S GPU, 16GB of RAM, Intel Core i7-9700 CPU, Windows 10 OS, and the YOLO technology are all included in the test environment. The simulation involves adjusting the image size to 224 × 224 pixels, employing the Adam optimization technique to improve the framework, setting the learning rate to 0.0001, training the model over 100 epochs utilizing a mini-batch of 10. In this case, consider into view that 70% of the data from Rail-5k and RSDDs is the train set, and 30% of the data is the test set. The study chooses precision, recall, F-measure, accuracy, and additional variables to contrast the algorithm with in order to appropriately assess its impact.

Equation (28) displays the percentage of actual positive samples amongst those determined as positive samples, defines precision.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (28)$$

Equation (29) defines recall, shows the percentage of positive instance in the sample are properly recognized.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (29)$$

It is insufficient to evaluate the model accuracy solely on recall or precision. As a result, the F-measure was created to take recall and precision into account simultaneously. Equation (30), which defines the F-measure, is an example.

$$\text{F} - \text{Measure} = 2 \times \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (30)$$

Typically, accuracy is employed to assess an algorithm's global accuracy, that can't include too much data or provide a complete assessment of the model. Equation (31) provides the explanation.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (31)$$

True Positive (TP) denoted a properly determined positive instance; True Negative (TN) denoted a precisely discovered negative instance; False Positive (FP) denoted a misidentified negative instance; and False Negative (FN) denoted a mistakenly discovered positive instance for a misidentified negative instance. Accuracy, recall, precision, and f-measure for RTFD and RSDDs among defect detection methods are displayed in Figures 4-7. ZS-CKG [26], CTFM [36], DCNN [37], R-CNN [13], and proposed system with their evaluation metrics are discussed in Table 1.

Table 1: Comparative Results Analysis for Datasets

DATASETS	METRICS	ZS-CKG	CTFM	DCNN	R-CNN	ZS-SSFCKG
RTFD	**PRECISION (%)**	75.87	78.55	80.42	82.19	84.74
	RECALL (%)	78.73	79.84	82.15	85.47	87.15
	F-MEASURE (%)	77.27	79.19	81.28	83.79	85.93
	ACCURACY (%)	76.18	78.51	80.86	82.33	84.87
RSDDs	**PRECISION (%)**	74.39	76.18	78.53	82.47	85.25
	RECALL (%)	77.85	78.37	80.31	84.66	86.58
	F-MEASURE (%)	76.08	77.26	79.41	83.55	85.90
	ACCURACY (%)	77.54	79.85	82.59	85.82	87.61

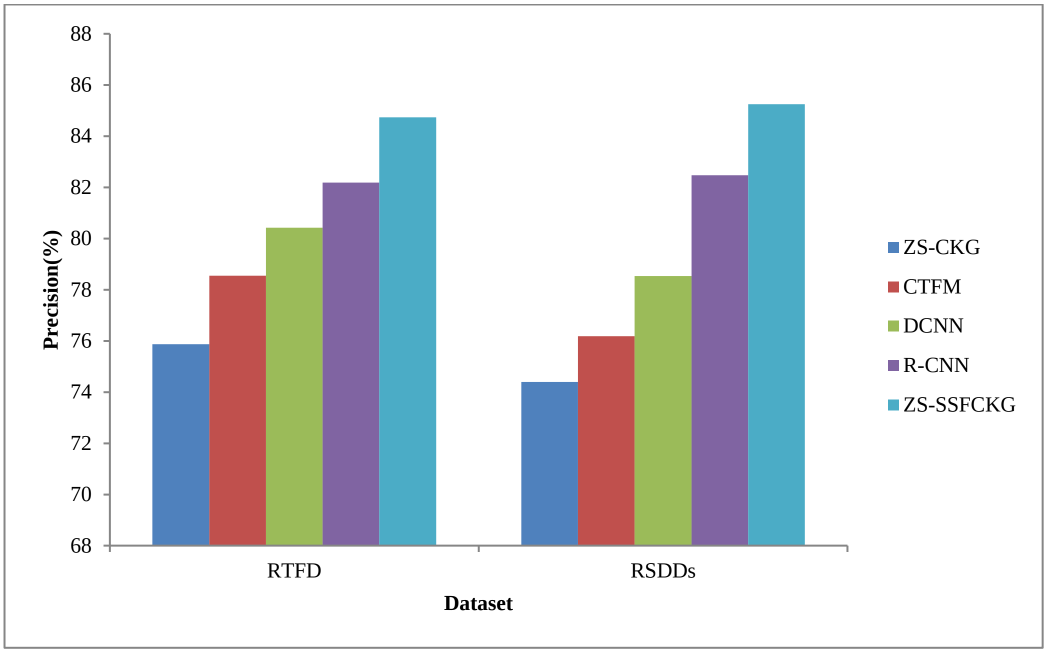

Figure 4: Precision vs. Zero-shot based on Knowledge Graph Methods

Figure 4 shows the precision comparison of methods with 2 datasets. From the graph, it can be observed that the proposed system produces superior surface defect identification of 85.25%, previous approaches is 74.39%, 76.18%, 78.53%, and 82.47% for RSDDs. Other methods like ZS-CKG, CTFM, DCNN, and R-CNN had lesser results of 10.86%, 9.07%, 6.72%, and 2.78% for RSDDs. However, the presented empirical findings lead to the conclusion that the proposed system utilizes the advantages of the rapid intersecting across the GCN, making it a superior choice for the prompt detection of surface defects.

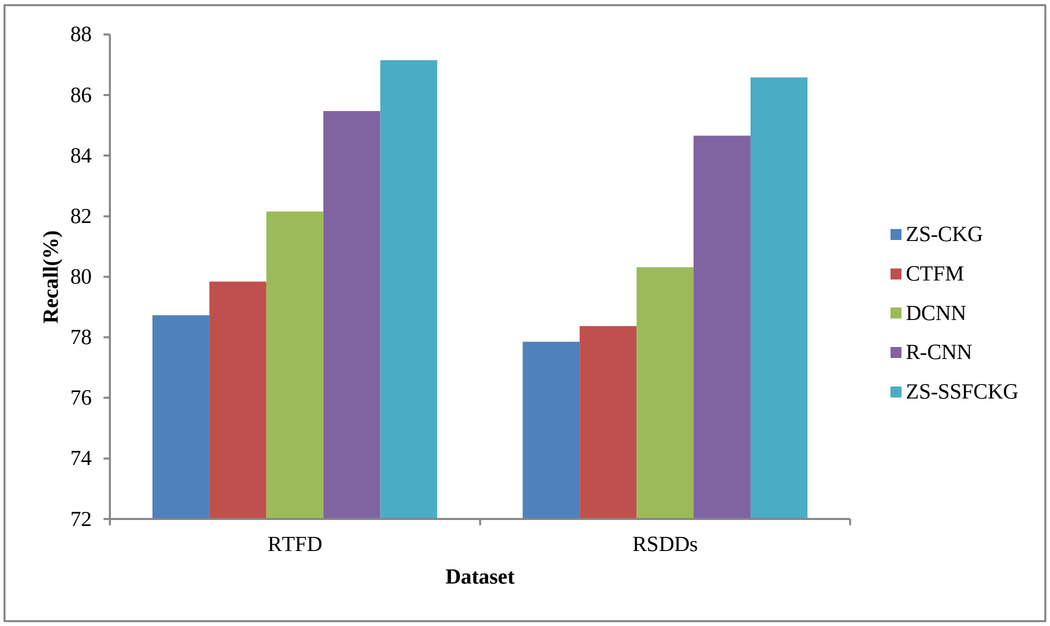

Figure 5: Recall VS. Zero-Shot based on Knowledge Graph Methods

Figure 5, this graph clarifies the recall comparison of methods with 2 datasets. From the graph, it can be observed that the proposed system produces superior surface defect identification is 86.58%, the previous approaches is 77.85%, 78.37%, 80.31%, and 84.66% for RSDDs. Other methods like ZS-CKG, CTFM, DCNN, and R-CNN had lesser results of 8.73%, 8.21%, 6.27%, and 1.92% for RSDDs. ZS-CKG, CTFM, DCNN, and R-CNN had lesser results of 78.73%, 79.84%, 82.15%, and 85.47% for RTFD. From the railway surface defect dataset, the proposed system has the largest recall which was with the improved training strategy with ESAFM.

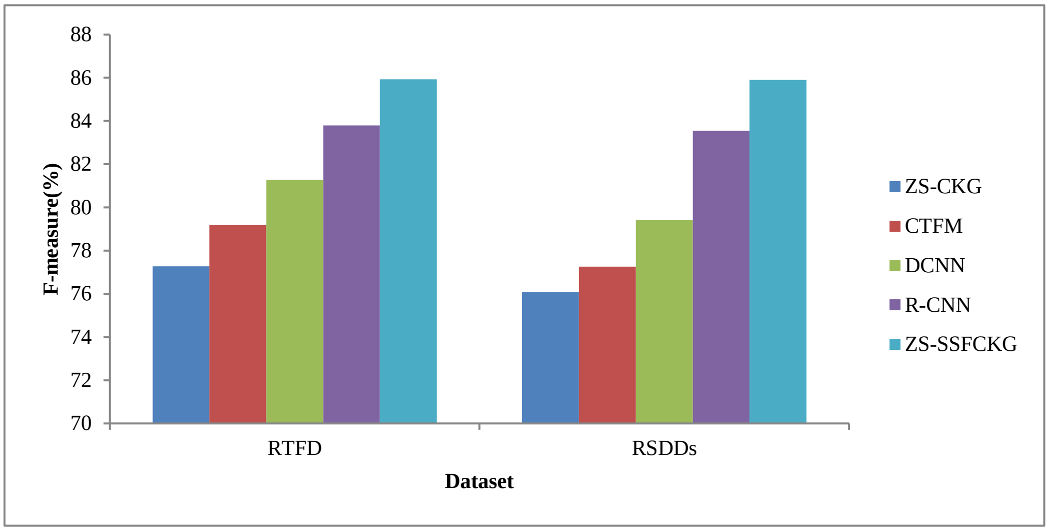

Figure 6: F-Measure VS. Zero-Shot based on Knowledge Graph Methods

Figure 6, this graph clarifies the F-measure comparison of methods with 2 datasets. It shows that the proposed system produces best results of 85.90%, existing methods have given lowest results of 76.08%, 77.26%, 79.41%, and 83.55% for RSDDs. Other methods like SGCN, GCNZ, ZS-CKG, and DCNN had lesser results of 9.82%, 8.64%, 6.49%, and 2.35% for RSDDs. SGCN, GCNZ, ZS-CKG, and DCNN have lesser results of 77.27%, 79.19%, 81.28%, and 83.79% for RTFD. It is found that the results of the proposed system provide a Latent Diffusion Model (LDM) for surface defect detection without any barrier and thus it is more effective than the existing methods.

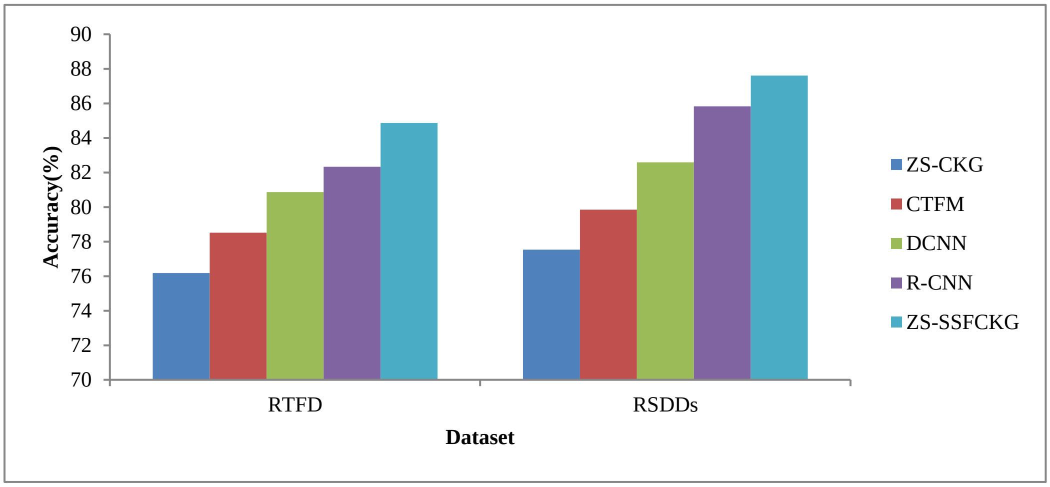

Figure 7: Accuracy VS. Zero-Shot based on Knowledge Graph Methods

Figure 7, this graph clarifies the accuracy comparison of methods with 2 datasets. Proposed system produces superior results of 87.61%, previous approaches have given results of 77.54%, 79.85%, 82.59%, and 85.82% for RSDDs. Other methods like ZS-CKG, CTFM, DCNN, and R-CNN had lesser results of 10.07%, 7.76%, 5.02%, and 1.79% for RSDDs. ZS-CKG, CTFM, DCNN, and R-CNN have lesser results of 76.18%, 78.51%, 80.86%, and 82.33% for RTFD. Proposed system was developed to increase detection rate due to exact detection of rail track cracks via ESAFM which led the detection accuracy gets improved than the other methods.

5 Conclusion and Future Work

Through prolonged and intensive railway usage, various defects tend to arise, typically resulting in mild to moderate surface damage. Such defects can negatively impact the consistent operation of trains and even pose risks to travel safety. In this paper, Entropy Stacked Autoencoder Based Diffusion Model (ESADM) is introduced for multi-crack detection to improve rail surface error-detecting precision. ESADM is designed to learn target defects by progressively denoising an image with normal distribution. Concurrent identification of fasteners and rail surface problems in railway track line images is accomplished by the use of ESADM. This includes layer-wise unsupervised autoencoder stacking and supervised fine-tuning. In ESADM, the weight values of encoders and decoders are optimized using an entropy function. In the ZS-SSFCKG framework, the CKGC creates a class knowledge graph to connect classes. ZS-SSFCKG extracts faulty sample characteristics and captures long-range dependency using a transformer encoder. The ZS-SSFCKG automatically detects rail surface flaws and fastener conditions. CKG follows the procedure of Fuzzy Clustering with Semi Supervised Fuzzy Graph Convolutional Network (FC-SSFGCN) encoder. Thorough experimentation on benchmark datasets such as RTFD and RSDDs showcases the efficacy of the proposed approach. To assess the results, performance assessment criteria such as accuracy, precision, recall, and F-measure are used. In forthcoming endeavors, emphasis will be placed on gathering additional images of rail surface defects, potentially facilitating the identification of various defect types. Moreover, future research will explore the integration of deep learning techniques and optimization strategies into the rail surface defect detection model, aimed at enhancing the accuracy of defect detection.

References

[1] Camgözlü, Y., & Kutlu, Y. (2023). Leaf Image Classification Based on Pre-trained Convolutional Neural Network Models. *Natural and Engineering Sciences, 8*(3), 214-232.

[2] Chenariyan Nakhaee, M., Hiemstra, D., Stoelinga, M., & Van Noort, M. (2019). The recent applications of machine learning in rail track maintenance: A survey. *In Reliability, Safety, and Security of Railway Systems. Modelling, Analysis, Verification, and Certification: Third International Conference, RSSRail 2019,* 91-105.

[3] Feng, J. H., Yuan, H., Hu, Y. Q., Lin, J., Liu, S. W., & Luo, X. (2020). Research on deep learning method for rail surface defect detection. *IET Electrical Systems in Transportation*, *10*(4), 436-442.

[4] He, Y., Wu, J., Zheng, Y., Zhang, Y., & Hong, X. (2022). Track defect detection for high-speed maglev trains via deep learning. *IEEE Transactions on Instrumentation and Measurement*, *71*, 1-8.

[5] James, A., Jie, W., Xulei, Y., Chenghao, Y., Ngan, N. B., Yuxin, L., & Zeng, Z. (2018). Tracknet-a deep learning based fault detection for railway track inspection. *In IEEE International Conference on Intelligent Rail Transportation (ICIRT)*, 1-5.

[6] Jessop, C., Ahlström, J., Hammar, L., Fæster, S., & Danielsen, H. K. (2016). 3D characterization of rolling contact fatigue crack networks. *Wear*, *366*, 392-400.

[7] Jiang, Y., Wang, H., Tian, G., Chen, S., Zhao, J., Liu, Q., & Hu, P. (2018). Non-contact ultrasonic detection of rail surface defects in different depths. *In IEEE Far East NDT New Technology & Application Forum (FENDT)*, 46-49.

[8] Jin, X., Wang, Y., Zhang, H., Zhong, H., Liu, L., Wu, Q.J., & Yang, Y. (2019). DM-RIS: Deep multimodel rail inspection system with improved MRF-GMM and CNN. *IEEE Transactions on Instrumentation and Measurement*, *69*(4), 1051-1065.

[9] Kampffmeyer, M., Chen, Y., Liang, X., Wang, H., Zhang, Y., & Xing, E. P. (2019). Rethinking knowledge graph propagation for zero-shot learning. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11487-11496.

[10] Kapoor, R., Goel, R., & Sharma, A. (2022). An intelligent railway surveillance framework based on recognition of object and railway track using deep learning. *Multimedia Tools and Applications*, *81*(15), 21083-21109.

[11] Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

[12] Kocbek, S., & Gabrys, B. (2019). Automated machine learning techniques in prognostics of railway track defects. *In IEEE International Conference on Data Mining Workshops (ICDMW)*, 777-784.

[13] Kutlu, Y., & Camgözlü, Y. (2021). Detection of coronavirus disease (COVID-19) from X-ray images using deep convolutional neural networks. *Natural and Engineering Sciences*, *6*(1), 60-74.

[14] Levchenko, A., Polikutin, A., & Barabash, D. (2020). Crack Resistance of Reinforced Concreted and Reinforced Rubber Concrete Beams. *Archives for Technical Sciences*, *1*(22), 21–26.

[15] Li, Z., Gao, L., Gao, Y., Li, X., & Li, H. (2022). Zero-shot surface defect recognition with class knowledge graph. *Advanced Engineering Informatics*, *54*, 101813. https://doi.org/10.1016/j.aei.2022.101813

[16] Liang, Z., Zhang, H., Liu, L., He, Z., & Zheng, K. (2018). Defect detection of rail surface with deep convolutional neural networks. *In IEEE 13th World Congress on Intelligent Control and Automation (WCICA)*, 1317-1322.

[17] Liu, Y., Xiao, H., Xu, J., & Zhao, J. (2022). A rail surface defect detection method based on pyramid feature and lightweight convolutional neural network. *IEEE Transactions on Instrumentation and Measurement*, *71*, 1-10..

[18] Luo, H., Cai, L., & Li, C. (2023). Rail surface defect detection based on an improved YOLOv5s. *Applied Sciences*, *13*(12), 7330. https://doi.org/10.3390/app13127330.

[19] Madhan, K., & Shanmugapriya, N. (2024). Efficient Object Detection and Classification Approach Using an Enhanced Moving Object Detection Algorithm in Motion Videos. *Indian Journal of Information Sources and Services*, *14*(1), 9–16.

[20] Popov, K., DeBold, R., Chai, H. K., Forde, M. C., Ho, C. L., Hyslip, J. P., & Hsu, S. S. (2022). Rail Track Monitoring Using AI and Machine Learning. *In International Conference on Trends on Construction in the Post-Digital Era*, 513-523.

[21] Rampriya, R. S., Suganya, R., Sabarinathan, Ganesan, A., Prathiksha, P., & Rakini, B. (2021). Object detection in railway track using deep learning techniques. *In International Conference on Computational Techniques and Applications*, 107-115.

[22] Rathi, S., Mirajkar, O., Shukla, S., Deshmukh, L., & Dangare, L. (2024). Advancing Crack Detection Using Deep Learning Solutions for Automated Inspection of Metallic Surfaces. *Indian Journal of Information Sources and Services*, *14*(1), 93–100.

[23] Samyuktha, S.S., & Subaji, M. (2024). Variational Autoencoder Diffusion Model (VAEDM) and Divergence Asynchronous Reinforcement Learning (DARL) for Rail Surface Defect Detection. *Journal of Internet Services and Information Security*, *14*(2), 298-317.

[24] Suleiman, H. (2023). *Pcεκ*max-Means++: Adapt-*P* Driven by Energy and Distance Quality Probabilities Based on *κ*-Means++ for the Stable Election Protocol (SEP). *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, 14*(4), 128-148.

[25] Thendral, R., & Ranjeeth, A. (2021). Computer vision system for railway track crack detection using deep learning neural network. *In IEEE 3rd International Conference on Signal Processing and Communication (ICPSC)*, 193-196.

[26] Usamentiaga, R., Lema, D. G., Pedrayes, O. D., & Garcia, D. F. (2022). Automated surface defect detection in metals: a comparative review of object detection and semantic segmentation using deep learning. *IEEE Transactions on Industry Applications*, *58*(3), 4203-4213.

[27] Varshavardhini, S., & Rajesh, A. (2023). An Efficient Feature Subset Selection with Fuzzy Wavelet Neural Network for Data Mining in Big Data Environment. *Journal of Internet Services and Information Security, 13*(2), 233-248.

[28] Wan, F., Guo, G., Zhang, C., Guo, Q., & Liu, J. (2019). Outlier detection for monitoring data using stacked autoencoder. *IEEE Access*, *7*, 173827-173837.

[29] Wang, H., Li, M., & Wan, Z. (2022). Rail surface defect detection based on improved Mask R-CNN. *Computers and Electrical Engineering*, *102,* 108269. https://doi.org/10.1016/j.compeleceng.2022.108269

[30] Wang, J., Ma, H., Tang, Q., Li, J., Zhu, H., Ma, S., & Chen, X. (2012). A New Efficient Verifiable Fuzzy Keyword Search Scheme. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, 3*(4), 61-71.

[31] Wei, X., Yang, Z., Liu, Y., Wei, D., Jia, L., & Li, Y. (2019). Railway track fastener defect detection based on image processing and deep learning techniques: A comparative study. *Engineering Applications of Artificial Intelligence*, *80*, 66-81.

[32] Wu, Y., Qin, Y., Qian, Y., Guo, F., Wang, Z., & Jia, L. (2022). Hybrid deep learning architecture for rail surface segmentation and surface defect detection. *Computer‐Aided Civil and Infrastructure Engineering*, *37*(2), 227-244.

[33] Xia-Ting, J., Yao-Nan, W., Hui, Z., Li, L., Hang, Z., & Zhen-Dong, H. (2019). Deep Rail: automatic visual detection system for railway surface defect using bayesian CNN and attention network. *Acta Automatica Sinica*, *45*(12), 2312-2327.

[34] Yanan, S., Hui, Z., Li, L., & Hang, Z. (2018). Rail surface defect detection method based on YOLOv3 deep learning networks. *In IEEE Chinese Automation Congress (CAC)*, 1563-1568.

[35] Yang, C., Sun, Y., Ladubec, C., & Liu, Y. (2020). Developing machine learning-based models for railway inspection. *Applied Sciences*, *11*(1), 13. https://doi.org/10.3390/app11010013

[36] Ye, J., Stewart, E., Chen, Q., Roberts, C., Hajiyavand, A. M., & Lei, Y. (2023). Deep learning and laser-based 3-D pixel-level rail surface defect detection method. *IEEE Transactions on Instrumentation and Measurement*, *72*, 1-12.

[37] Yu, H., Li, Q., Tan, Y., Gan, J., Wang, J., Geng, Y. A., & Jia, L. (2018). A coarse-to-fine model for rail surface defect detection. *IEEE Transactions on Instrumentation and Measurement*, *68*(3), 656-666.

[38] Yu, M., Quan, T., Peng, Q., Yu, X., & Liu, L. (2022). A model-based collaborate filtering algorithm based on stacked Auto Encoder. *Neural Computing and Applications*, 1-9.

[39] Zhang, D., Song, K., Wang, Q., He, Y., Wen, X., & Yan, Y. (2020). Two deep learning networks for rail surface defect inspection of limited samples with line-level label. *IEEE Transactions on Industrial Informatics*, *17*(10), 6731-6741.

Authors Biography

Samyuktha Sasi Sekaran, with 11 years' experience in ML, AI, JavaScript, Node.js, and DBMS, specializes in ML/AI for 7 years. Currently pursuing a Ph.D. at VIT University, India, she's a Data Scientist at SAP India. Holding Bachelor's and Master's degrees in Engineering, she's proficient in Python, PyTorch, Keras, OpenAI, and Generative AI, as well as cloud platforms like Kubernetes, GCP, and Azure. Skilled in JavaScript, Node.js, and React, she specializes in Genetic Algorithms, NLP, and Computer Vision, with expertise in Big Data tech like Apache Spark, SQL, and NoSQL databases. She conducted training sessions at SAP on deep learning and ML techniques, excelling in designing and developing commercial applications, contributing significantly to various sectors, recognized for her adaptability, communication skills, and proactive approach.

Dr. Subaji Mohan, obtained his Bachelor’s degree in Mechanical Engineering, Master’s degree in Business Administration and PhD from Kookmin University, Seoul, South Korea. He is working as Professor & Director, Institute for Industry and International Programmes at Vellore Institute of Technology, Vellore, Tamil Nadu, India. He has produced two PhD students and his research area focus on Image Processing, Data Analytics & Distributed Systems