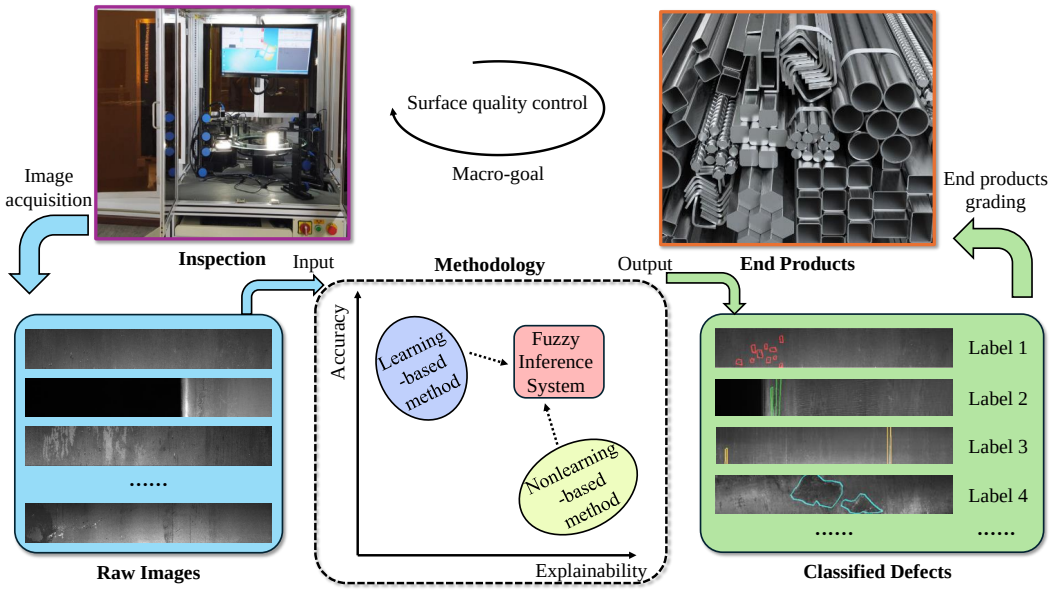
Graphical Abstract

Integrating Large Language Models with Explainable Fuzzy Inference Systems for Trusty Steel Defect Detection

Kening Zhang, Yung Po Tsang, Carman K. M. Lee, C.H. WU



Highlights

Integrating Large Language Models with Explainable Fuzzy Inference Systems for Trusty Steel Defect Detection

Kening Zhang, Yung Po Tsang, Carman K. M. Lee, C.H. WU

- LE-FIS detects steel defect.
- LLMs support LE-FIS.
- Give reliable explanations.

Integrating Large Language Models with Explainable Fuzzy Inference Systems for Trusty Steel Defect Detection

Kening Zhang[a], Yung Po Tsang[a], Carman K. M. Lee[a] and C.H. WU[b,*]

[a]Department of Industrial and Systems Engineering, Research Institute for Advanced Manufacturing, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong,

[b]Department of Supply Chain and Information Management, The Hang Seng University of Hong Kong, Siu Lek Yuen, N.T, Hong Kong,

ARTICLE INFO

Keywords:
Fuzzy interfence system (FIS)
Steel defect detection
Explainable artificial intelligence (XAI)
Black-box model

ABSTRACT

In high-risk industrial applications, the complexity of machine learning models often makes their decision-making processes difficult to interpret and lack transparency, particularly in the steel manufacturing sector. Understanding these processes is crucial for ensuring quality control, regulatory compliance, and gaining the trust of stakeholders. To address this issue, this paper proposes LE-FIS, an explainable fuzzy inference system based on large language models (LLMs) to interpret black-box models for steel defect detection. The method introduces a locally trained, globally predicted deep detection approach (LTGP), which segments the image into small parts for local training and then tests on the entire image for steel defect detection. Then, LE-FIS is designed to explain the LTGP by automatically generating rules and membership functions, with a genetic algorithm (GA) used to optimize parameters. Furthermore, state-of-the-art LLMs are employed to interpret the results of LE-FIS, and evaluation metrics are established for comparison and analysis. Experimental results demonstrate that LTGP performs well in defect detection tasks, and LE-FIS supported by LLMs provides a trustworthy and interpretable model for steel defect detection, which enhances transparency and reliability in industrial environments.

1. Introduction

The steel manufacturing industry is a cornerstone of modern infrastructure, playing a pivotal role in the construction, automotive, and machinery sectors [1]. Ensuring the quality and integrity of steel products is paramount, as defects can lead to significant economic losses and safety hazards. Traditional methods of defect detection, such as visual inspection [2] and manual testing [3], are labor-intensive, time-consuming, and often prone to human error. These methods typically involve trained inspectors examining steel surfaces for visible defects or using non-destructive testing (NDT) techniques such as ultrasonic testing, magnetic particle testing, and radiographic testing to detect internal flaws [4]. While these approaches have been the industry standard for decades, they come with several limitations.

Visual inspection, despite being straightforward, heavily relies on the inspector's experience and can be inconsistent due to human fatigue and subjective judgment. Non-destructive testing methods, although more reliable, require specialized equipment and skilled operators, leading to increased operational costs and potential bottlenecks in the production process. Moreover, these traditional methods may not always detect subtle or emerging defects, which can compromise the quality of the final product [5]. As the demand for higher quality and more complex steel products grows, there is an urgent need for more efficient and accurate defect detection methods.

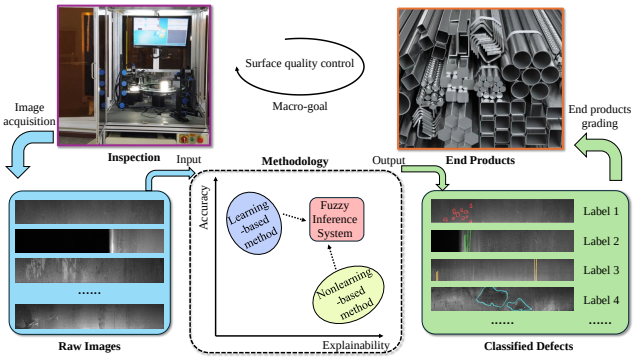


Figure 1: Differences in explainability and accuracy of machine learning methods and non-machine learning methods to steel defect detection.

To overcome these limitations, the industry has increasingly turned to visual intelligence (VI) approaches, which offer the promise of higher accuracy and efficiency in identifying defects. These machine learning-based VI models, often referred to as "black-box" models due to their complex and opaque nature, leverage vast amounts of data to learn intricate patterns and make predictions [6]. Techniques such as convolutional neural networks (CNNs) and deep learning have shown remarkable success in detecting various types of defects in steel products, ranging from surface cracks to internal inconsistencies [7]. For instance, CNNs are particularly effective in image-based defect detection tasks due to their ability to automatically learn hierarchical features from raw image data. Other deep learning architectures, such

*Corresponding author
keningcs.zhang@connect.polyu.hk (K. Zhang); yungpo.tsang@polyu.edu.hk (Y.P. Tsang); ckm.lee@polyu.edu.hk (C.K.M. Lee); jackwu@hsu.edu.hk (C.H. WU)
ORCID(s): 0009-0006-2680-5247 (K. Zhang); 0000-0002-6128-345X (Y.P. Tsang); 0000-0001-8577-4547 (C.K.M. Lee); 0000-0003-1259-4048 (C.H. WU)

as recurrent neural networks (RNNs) and long short-term memory networks (LSTMs), have been employed to analyze time-series data from sensors and predict potential defects based on historical patterns [8]. In addition to deep learning, other machine learning techniques such as support vector machines (SVMs), random forests, and gradient boosting machines have also been applied to defect detection [9]. These models can handle a variety of data types and are often used in combination with feature engineering techniques to improve their performance. For example, SVMs have been used to classify defects based on extracted features from ultrasonic testing data, while random forests have been applied to identify patterns in magnetic particle testing results. Despite their success, these models still suffer from the "black-box" problem, where the decision-making process is not easily interpretable by humans. The process of machine learning and non-machine learning algorithms to steel defect detection is compared in Figure 1.

The inherent complexity of these models makes it difficult to understand how they arrive at their decisions, posing significant challenges in terms of interpretability and transparency. This lack of insight is particularly problematic in high-stakes industrial applications, where understanding the decision-making process is crucial for several reasons. Firstly, quality control processes require clear explanations to ensure that detected defects are accurately identified and appropriately addressed. Secondly, regulatory compliance often mandates that the methods used for defect detection are transparent and justifiable. Thirdly, gaining the trust of stakeholders, including engineers, managers, and customers, necessitates a clear understanding of how these advanced models operate and make decisions. Without this transparency, the adoption of machine learning models in the steel manufacturing industry may be hindered, limiting the potential benefits these technologies can offer.

Therefore, it becomes particularly important to propose an inspection method with explainability and trustworthiness. Such a model can not only provide accurate defect detection, but also allow users to understand the decision-making process of the model, enhance trust in the model results, and quickly make optimizations and adjustments when needed. Such an approach will not only improve inspection efficiency, but also reduce the potential risks caused by misjudgments, meeting the need for transparency and reliability in industrial environments. Then, we propose the large language model (LLM)-based explainable fuzzy inference system (LE-FIS) for black-box model of steel defect detection. The main contributions of this paper can be concluded as follows:

1. We propose a locally trained, globally predicted deep detection method (LTGP) to address the steel defect detection, where the image is first segmented into small images, trained using them, and finally tested on the complete image.
2. We propose the LE-FIS to explain the LTGP. LE-FIS can automatically generate rules and membership functions, and a genetic algorithm (GA) is used to optimize the parameters based on the prediction results of the black-box model.
3. We utilize state-of-the-art LLMs to interpret the results of LE-FIS, establishing evaluation metrics for comparison and analysis, ultimately selecting the best-performing model for explanation.
4. The results show that our proposed LTGP can perform the detection task well, and LE-FIS can be reasonably explained with the support of the LLMs, which provides a trustworthy model for the steel defect detection in industry.

2. Related Works

Defect detection methods for steel surfaces can be broadly classified into distinct categories based on technical roadmaps [10], models [11], groups[12], and so on [5]. Here, we mainly focus on the differences between non-machine learning methods and machine learning-based methods. Non-machine learning methods perform detection by extracting distinct texture features [13]. Luo et al. [14] presented a modular and cost-effective automated optical inspection (AOI) system for real-time hot-rolled flat steel surface inspection with the statistical method. Neogi et al. [15] introduced a global adaptive percentile thresholding technique for gradient images, which dynamically adjusted the threshold based on pixel intensity distribution to effectively segment defects of varying sizes in steel strips, outperforming local adaptive thresholding methods. When tiny defects and lack of light are plainly present, spectral methods become even more remarkable [10]. Yazdchi et al. [16] proposed a novel defect detection algorithm based on multifractal analysis with temporal Fourier transform (FT), which thereby enhanced the speed and precision of real-time inline surface defect inspection in steel manufacturing. Similarly, Ai et al. [17] effectively detected longitudinal cracks in continuous casting slabs by utilizing Curvelet transform and kernel locality preserving projections (KLPP) for feature extraction and dimensionality reduction. Both approaches struggle with handling diverse defects and background variations, while model-based methods excel by using specialized models to project texture distributions into a low-dimensional space, improving defect detection [18]. Cross et al. [19] used Markov random fields (MRF) and a binomial model for texture generation and analysis, demonstrating realistic microtextures. Timm et al. [12] introduced a non-parametric defect detection method using Weibull fit parameters and a novelty detection algorithm, achieving high accuracy across various textures. Song et al. [20] developed a saliency-based convex active contour model for detecting micro surface defects in silicon steel strips, effectively suppressing background clutter and achieving high accuracy. In these research lines, it is essential to construct noise robust, computationally simple and mathematically explanatory models for steel defect detection.

The application of machine learning to steel defect detection has greatly improved accuracy and efficiency. Liu

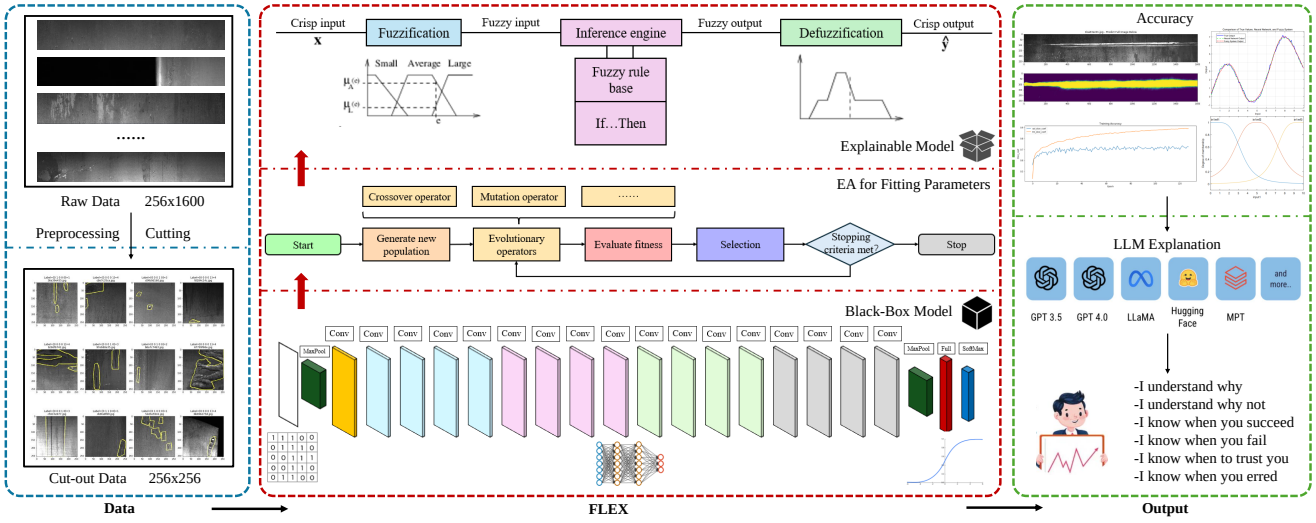


Figure 2: Framework of the proposed LE-FIS.

et al. [21] proposed a two-layer feed-forward neural network for surface defect detection on steel strips, showing its effectiveness. CNNs, particularly deep CNNs, have been successful in identifying various defects, as demonstrated by Soukup et al. [22] and Zhang et al. [23]. DCGANs and autoencoders have been used for unsupervised defect detection, such as in Zhao et al.'s [24] work combining GAN and autoencoder methods. Unsupervised models like Youkachen et al.'s [11] convolutional auto-encoder (CAE) handle defects without labeling. Reinforcement learning also enhances defect detection, as Ren et al. [25] showed using pretrained network features.

In industries like steel manufacturing, where defect detection is crucial for maintaining product quality and safety, it is essential for methods to replicate human expertise in annotation. This capability ensures that humans can trust the models, as it allows for transparency in decision-making processes. While complex models like deep neural networks may perform well in specific scenarios, their lack of explainability can hinder trust and make it challenging to trace errors or make timely adjustments under new conditions.

3. Method

In this section, we delineate the comprehensive methodology employed for the Steel Defect Detection task, encapsulated in the LE-FIS framework. The proposed approach integrates image preprocessing, deep learning, fuzzy inference, and LLM-based explainability. The methodology is structured into four pivotal steps: image preprocessing and segmentation, defect detection using ResNet-18, training and fitting with GA, and explainability through LLM.

3.1. Preprocessing and Segmentation

To ensure experiment controllability and scalability, we systematically preprocessed the steel defect images. This included resizing for consistent input dimensions, normalizing pixel values to reduce brightness and contrast variations, and applying segmentation to extract key regions while minimizing background noise. These steps enhanced data quality, reduced noise, and improved the overall efficiency and accuracy of the experiment, providing a solid foundation for subsequent analysis.

Let the original input image be denoted as I_{raw}, with dimensions $H \times W$. The goal is to crop the image into smaller patches of size $h_c \times w_c$. We employ a sliding window approach to extract patches from the raw image. Let s_h and s_w be the vertical and horizontal stride lengths, respectively. Each patch $I^{i,j}_{crop}$ can be expressed as: $I^{i,j}_{crop} = I_{raw}(i \cdot s_h : i \cdot s_h + h_c, j \cdot s_w : j \cdot s_w + w_c)$, where i and j are the indices of the sliding window position. The stride values s_h and s_w can be set to equal h_c and w_c, or overlap can be introduced. Then, patches extracted are $N_h = \left\lfloor \frac{H-h_c}{s_h} \right\rfloor + 1$ and $N_w = \left\lfloor \frac{W-w_c}{s_w} \right\rfloor + 1$. This process produces $N_h \times N_w$ patches from the original image. If H or W is not divisible by the stride, some regions near the image borders may not be fully covered by the cropping window. To address this, we apply zero-padding to ensure all regions are included in the final set of patches. Each patch can then be treated as an independent input for model training.

Then, normalization is performed to scale pixel values to the range [0, 1]. Assuming that the cropped images are $I_r(l)$, where $l \in \{1, 2, \ldots, N_h \times N_w\}$, the normalized images can be expressed as $I_n(l) = \frac{I_r(l)-\mu}{\sigma}$, where μ and σ are the mean and standard deviation of the pixel values in the dataset.

3.2. Defect Detection

3.2.1. Deep learning-based module

After preprocessing and segmentation, we proposed a LTGP method for steel defect detection based on [26], which consists of a series of residual blocks designed to mitigate the vanishing gradient problem in deep networks through shortcut connections. Each residual block can be expressed as $y = F(x, \{W_i\}) + x$, where x is the input feature

map, and $F(x, \{W_i\})$ is the convolutional transformation with weights $\{W_i\}$. The residual block learns the residual mapping $F(x)$, and the addition of x allows gradients to flow more effectively during backpropagation.

The convolution operation inside the residual block is given by $F(x, \{W_i\}) = W_2 \cdot \sigma(W_1 \cdot x)$, where W_1 and W_2 are convolutional weights, and $\sigma(\cdot)$ is the ReLU activation function.

After passing through the residual blocks, the final feature map undergoes global average pooling (GAP) to reduce its dimensionality. Let the dimensions of the final feature map be $H_f \times W_f$. The GAP operation is defined as $f_{gap} = \frac{1}{H_f \times W_f} \sum_{i=1}^{H_f} \sum_{j=1}^{W_f} f_{rn}(i, j)$, where $f_{rn}(i, j)$ denotes the pixel value at position (i, j) in the final feature map.

The pooled feature vector f_{gap} is then fed into a fully connected (FC) layer for classification. The output of the FC layer is given by $z = W_{fc} \cdot f_{gap} + b_{fc}$, where W_{fc} is the weight matrix of the FC layer, and b_{fc} is the bias term. Finally, the softmax function is applied to produce the class probabilities by $\hat{y}_i = \frac{\exp(z_i)}{\sum_{j=1}^{C} \exp(z_j)}, \quad i = 1, 2, \ldots, C$, where C is the number of classes, and z_i is the unnormalized logit for class i.

3.2.2. FIS module for black-box fitting

To interpret the black-box model, we employ an FIS to fit the outputs. FIS combines the learning capabilities of neural networks with the interpretability of fuzzy logic by generating membership functions and fuzzy rules.

For each input feature x_i, FIS maps it to a membership degree using a membership function $\mu_{A_i}(x_i)$. A commonly used membership function is the generalized bell-shaped function, which is $\mu_{A_i}(x_i) = 1/(1 + \left(\frac{x_i - c_i}{\sigma_i}\right)^2)$, where c_i is the center and σ_i is the width of the membership function.

Then, we use a set of fuzzy rules to map the input features to the output. A typical rule can be expressed as:

If x_1 *is* A_1 *and* x_2 *is* A_2, *then* $z = f(x_1, x_2)$.

The firing strength of the i-th rule is computed as $w_i = \prod_{j=1}^{n} \mu_{A_j}(x_j)$. Therefore, the final output is a weighted average of all rule outputs, which is defined as $z = \frac{\sum_{i=1}^{N} w_i f_i(x_1, x_2)}{\sum_{i=1}^{N} w_i}$, where N is the number of rules, and $f_i(x_1, x_2)$ is the output of the i-th rule.

The proposed FIS module optimizes the parameters of the membership functions c_i and σ_i as well as the rule weights w_i using GA. The loss function is the mean-squared error (MSE) between the FIS output z_i and the prediction $\hat{y}_i$ of the black-box model, which can be expressed as $\mathcal{L} = \frac{1}{m} \sum_{i=1}^{m} \left(z_i - \hat{y}_i\right)^2$, where m is the number of samples.

3.3. Prompt Design of LLMs

With the wide application of LLMs, prompts are transformed into a kind of engineering, the so-called prompt engineering (PE) [27]. PE is an LLM usage technique to improve the performance of LLM by designing and improving the prompts of LLM. Here, we design the cue words through six sections so that LLMs can interpret the results of the LE-FIS.

1. Task: The task involves using a fuzzy logic system to detect steel defects, followed by an explanation of the detection process and reasoning results using a large language model (LLM).
2. Context: The Severstal dataset contains images of steel surfaces annotated with different types of defects. The fuzzy logic system will classify defects based on image features such as color and texture. The LLM will be responsible for explaining the reasoning behind the fuzzy logic system's decisions, ensuring human users can understand the underlying logic.
3. Example: Based on the fuzzy logic rules, the color variation and irregularities in texture in the image suggest the presence of a scratch.
4. Roles: In this task, the LE-FIS acts as the "detector," analyzing images and detecting defects, while the LLM serves as the "explainer," converting the detection results into understandable natural language.
5. Format: The output should be in structured text format, including: detection result, explanation and possible recommendation (e.g., "further inspection of this area is recommended").
6. Tone: The tone should be professional yet accessible, ensuring that technical terminology is properly explained and that the language is clear enough for non-expert users to comprehend.

4. Performance and Analysis

This section presents the performance evaluation of the proposed LE-FIS framework on the steel defect detection task. We first describe the dataset used for training and evaluation, followed by a detailed analysis of the algorithm results, including the performance metrics and interpretability aspects.

4.1. Dataset Description

The Kaggle Steel Defect Detection dataset is used in this study as the primary dataset for training and evaluating the proposed LE-FIS framework. This dataset is publicly available and widely used for benchmarking defect detection models in industrial applications. It contains images of steel surfaces annotated with four types of surface defects. These defects are localized by pixel-wise masks, making it possible to train models for both detection and segmentation tasks.

- Number of images: 12,568 training images, each with a resolution of 1600×256 pixels.
- Defect types: The dataset is labeled with four defect types, namely:
  - Type 1 (Scratches): Linear, narrow surface markings.
  - Type 2 (Patches): Irregular, larger surface defects with rough textures.

  - Type 3 (Dents): Depressions or surface distortions with varied shapes and sizes.
  - Type 4 (Cracks): Thin, elongated fractures that extend across the surface.

- Pixel-wise annotations: Each image comes with pixel-wise defect annotations in the form of masks, indicating the exact regions affected by the defects.

- Class distribution: The dataset suffers from a highly imbalanced distribution of defects, with certain defect types, such as cracks and scratches, appearing much more frequently than others like dents and patches. This imbalance poses a significant challenge for model training, as the model may become biased toward the majority classes.

This dataset encompasses primary defect types, accompanied by pixel-wise annotations that effectively support the training and evaluation of our LE-FIS. Leveraging this information while implementing strategies to address class imbalance is crucial for enhancing performance. It is anticipated that this approach can enable the proposed system to achieve higher accuracy in defect detection and segmentation within industrial applications.

4.2. Experimental Performance

In this study, we utilize the *Dice coefficient* as the evaluation metric to assess the segmentation performance of the model in detecting steel surface defects. The Dice coefficient measures the similarity between the predicted segmentation and the ground truth, and is defined as:

$$\text{Dice} = \frac{2 \times |X \cap Y|}{|X| + |Y|}$$

where X represents the set of predicted pixels, and Y represents the set of ground truth pixels. The Dice coefficient ranges from 0 to 1, where a value of 1 indicates perfect overlap between the prediction and the ground truth, while a value of 0 indicates no overlap. When both X and Y are empty, the Dice coefficient is defined to be 1. This metric is particularly suitable for handling imbalanced datasets and small-area defect segmentation.

Furthermore, for quantitative assessment, precision, recall, and F-measure are used as key metrics to evaluate the performance of the model. Precision measures the proportion of true positive predictions among all positive predictions, defined as:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

where TP represents true positives and FP denotes false positives. Recall measures the proportion of true positives correctly identified out of all actual positives, defined as:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

where FN represents false negatives. F1-score is the harmonic mean of Precision and Recall, given by:

$$\text{F1-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

The experiments are conducted on a 64-bit Windows 11 Enterprise operating system (23H2), powered by a 12th Gen Intel(R) Core(TM) i7-12700KF 3.60 GHz and 32 GB of RAM. MATLAB R2024B is used as the primary software environment, with the Fuzzy Logic Toolbox 24.2 and Optimization Toolbox employed for implementing and optimizing the LE-FIS. The GA is configured through the '*tunefisOptions*' function with the following key parameters: the maximum number of generations is set to 500 with '*MethodOptions.MaxGenerations*', the function tolerance ('*MethodOptions.TolFun*') is defined as 1×10^{-6}, and the population size is 50 based on '*MethodOptions.PopulationSize*'. These settings are used to optimize the membership functions and rule base of the LE-FIS.

The results of the above indicators can be obtained according to Table 1. Although LE-FIS is slightly inferior to machine learning models in terms of performance, it is able to provide more explainability for complex black-box models. For industrial application scenarios such as steel defect detection, LE-FIS helps to improve model transparency and reliability, which may be important for the practical operation of production lines. Therefore, LE-FIS provides an effective way to trade-off between performance and interpretability, especially when the accuracy requirement is relatively low and the explainability requirement is high.

In Figure 3a, the proposed learning-based model demonstrates a strong learning capability, with validation accuracy improving rapidly in the early epochs and maintaining a high level throughout the training process, which indicates robust adaptability in identifying steel defects. Additionally, the Dice coefficient in Figure 3b shows a steady upward trend, suggesting that the model is progressively refining its segmentation accuracy and effectively capturing the characteristics of various defect types. These trends highlight the segmentation and classification performance, achieving rapid and stable success in this challenging task.

Figure 4 shows the process of optimizing LE-FIS parameters using a GA. In Figure 4a, the rapid decrease in both the best and mean fitness values in the early generations suggests that the algorithm efficiently identified optimal parameter combinations in the initial phase. As the generations progress, the fitness values stabilize, indicating successful convergence. Figure 4b illustrates that while some stalling occurs in later stages, the algorithm avoids significant local optima, maintaining its global search capability. Overall, the GA effectively optimizes the parameters of LE-FIS, which makes it well-suited for the steel defect detection task.

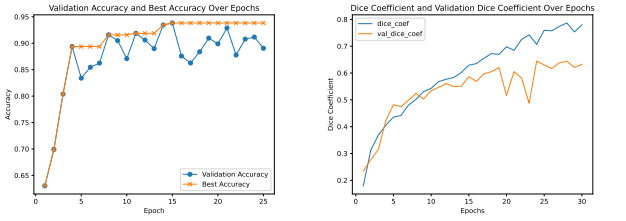
Figure 5a gives clear separation between selected different steel defect categories, which suggests that the features effectively distinguish between defect types. The membership functions in Figure 5b smoothly transition between fuzzy sets, which presents the system's ability to handle uncertainty in the inputs.

Table 1
Performance comparison between machine learning model and LE-FIS

Class	Machine Learning Model			LE-FIS		
	Precision (%)	Recall (%)	F1-score (%)	Precision (%)	Recall (%)	F1-score (%)
Scratches	92.75	91.92	92.33	82.18	80.45	81.30
Patches	90.63	89.74	90.18	79.58	78.22	78.89
Dents	88.92	87.88	88.40	77.45	75.22	76.32
Cracks	93.85	92.96	93.40	83.12	81.24	82.17
Macro Avg	91.54	90.62	91.08	80.58	78.78	79.67
Weighted Avg	91.97	91.75	91.86	81.03	79.18	80.09
Support (Scratches)		100			100	
Support (Patches)		30			30	
Support (Dents)		15			15	
Support (Cracks)		50			50	

Table 2
Explanation performance of different LLMs on the LE-FIS

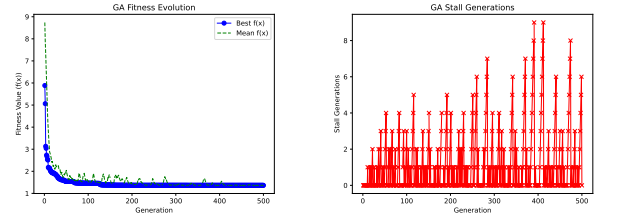
		LLM Model							
		Copilot	ChatGPT-4o	ChatGPT-4o mini	ChatGPT-o1 mini	ChatGPT-4	Qwen	Mistral	Llama
Words	IQ	700	928	714	1125	619	754	572	598
	DQ	1036	1924	789	1357	788	738	1033	1253
Membership function explanation		Yes	Yes	No	No	No	Yes	Yes	No
Rules explanation		Yes	Yes	No	Yes	Yes	Yes	No	Yes
Feature explanation		No	Yes	No	Yes	No	Yes	No	No



(a) Validation accuracy and best accuracy over epochs

(b) Dice coefficient and validation dice coefficient over epochs

Figure 3: Accuracy and dice coefficient comparison



(a) Fitness comparision over generations

(b) Distribution of GA stall generations

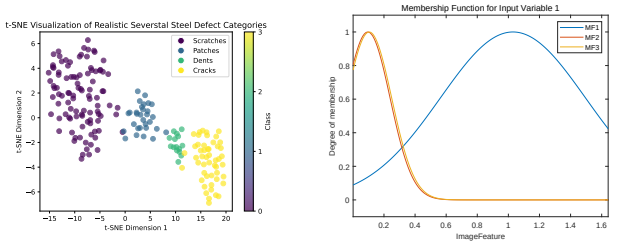
Figure 4: The process of parameter change in evolution

4.3. LLM Explanation

4.3.1. Model Comparisons

To improve the explainability of the system, we give the output of LE-FIS to LLMs for rule naming and explanation. We first give the model an initial question (IQ) and secondly a detailed question (DQ). The specific results are shown in Table 2.

During the IQ stage, ChatGPT-4 with 928 words and ChatGPT-o1 mini with 1125 words gave the most explanations, indicating that these models provided more detailed explanations at the first response. In contrast, Mistral and Llama provided relatively brief explanations. In the DQ phase, ChatGPT-4 and Copilot once again lead in terms of word count, especially ChatGPT-4 (1924 words), suggesting that it gives more detailed explanations at the time



(a) t-SNE visulization for selected dataset

(b) Distribution of GA Stall Generations

Figure 5: Relationship between the membership function and image features

of the follow-up question. Comparatively, Llama and Mistral continued to respond with fewer words and relatively concise explanations. Copilot and ChatGPT-4o are able to interpret the membership function, but ChatGPT-4o mini, ChatGPT-o1 mini, and Llama fail to provide an explanation for this. This suggests that some LLMs are limited in their ability to deal with affiliation, while ChatGPT-4o has a stronger understanding in this respect. Most models are capable of providing rule-based explanations, which implies their ability to elucidate the logical rules underlying fuzzy reasoning systems. However, models like ChatGPT-4o mini and Mistral fail to explain these rules, suggesting a weaker capability in handling rule-based reasoning. On the other hand, ChatGPT-4o and Qwen are able to provide feature-level explanations, while other models, such as Copilot and ChatGPT-4o mini, do not demonstrate this ability. This indicates significant differences in how these models handle input features, due to varying levels of attention to features during the reasoning process.

4.3.2. The Best Explanation

By synthesizing the comparisons, we chose the GPT-4o responses as the explanations to present. Here, we have eliminated additional redundant introductory and intonational sentences by selecting only useful information. The explanation is expressed as follows:

In the steel defect detection task, image features like edge detection, texture, brightness, and shape analysis help identify different defect types. These features are fuzzified using membership functions, which assess the likelihood of defects. Low, medium, and high defect possibilities guide the classification and detection process. For example, if edge detection shows strong signals, it suggests a high possibility of cracks, prompting the system to mark the area for repair. Simple rules are applied: smooth surfaces indicate no defects, small scratches suggest medium defects, and deep cracks or mixed signals point to severe or complex issues requiring immediate attention.

5. Conclusion

In this paper, we proposed a novel approach for addressing the challenges of explainability and trustworthiness in

steel defect detection using a LTGP method combined with the LE-FIS based on LLMs. The LTGP method demonstrated effectiveness in detecting steel defects by leveraging localized training on segmented images, while LE-FIS provided a transparent explanation for the detection process through rule generation and parameter optimization. The integration of state-of-the-art LLMs further enhanced the interpretability of the model. Our results showed that this approach not only performs well in defect detection but also addresses the need for transparency and reliability in industrial applications.

In future work, we aim to extend this approach to other defect types and materials, optimize the fuzzy inference system, and integrate real-time feedback for improved adaptability.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to thank the Research and Innovation Office of the Hong Kong Polytechnic University for supporting the project (Project Code: RMGT)

References

[1] Hongming Na, Jingchao Sun, Ziyang Qiu, Jianfei He, Yuxing Yuan, Tianyi Yan, and Tao Du. A novel evaluation method for energy efficiency of process industry—a case study of typical iron and steel manufacturing process. *Energy*, 233:121081, 2021.

[2] Roland T Chin and Charles A Harlow. Automated visual inspection: A survey. *IEEE transactions on pattern analysis and machine intelligence*, (6):557–573, 1982.

[3] Henry YT Ngan, Grantham KH Pang, and Nelson HC Yung. Automated fabric defect detection—a review. *Image and vision computing*, 29(7):442–458, 2011.

[4] Mark R Jolly, Arun Prabhakar, Bogdan Sturzu, K Hollstein, Rajendra Singh, S Thomas, Peter Foote, and Andy Shaw. Review of non-destructive testing (ndt) techniques and their applicability to thick walled composites. *Procedia CIRP*, 38:129–136, 2015.

[5] Qiwu Luo, Xiaoxin Fang, Li Liu, Chunhua Yang, and Yichuang Sun. Automated visual defect detection for flat steel surface: A survey. *IEEE Transactions on Instrumentation and Measurement*, 69(3):626–644, 2020.

[6] Qingyuan Zhao and Trevor Hastie. Causal interpretations of black-box models. *Journal of Business & Economic Statistics*, 39(1):272–281, 2021.

[7] Prahar M Bhatt, Rishi K Malhan, Pradeep Rajendran, Brual C Shah, Shantanu Thakar, Yeo Jung Yoon, and Satyandra K Gupta. Image-based surface defect detection using deep learning: A review. *Journal of Computing and Information Science in Engineering*, 21(4):040801, 2021.

[8] K Sharath Kumar and M Rama Bai. Lstm based texture classification and defect detection in a fabric. *Measurement: Sensors*, 26:100603, 2023.

[9] Halimi Abdellah, Roukhe Ahmed, and Ouhmad Slimane. Defect detection and identification in textile fabric by svm method. *IOSR Journal of Engineering*, 4(12):69–77, 2014.

[10] Heying Wang, Jiawei Zhang, Ying Tian, Haiyong Chen, Hexu Sun, and Kun Liu. A simple guidance template-based defect detection method for strip steel surfaces. *IEEE Transactions on Industrial Informatics*, 15(5):2798–2809, 2018.

[11] Sanyapong Youkachen, Miti Ruchanurucks, Teera Phatrapomnant, and Hirohiko Kaneko. Defect segmentation of hot-rolled steel strip surface by using convolutional auto-encoder and conventional image processing. In *2019 10th International conference of information and communication technology for embedded systems (IC-ICTES)*, pages 1–5. IEEE, 2019.

[12] Fabian Timm and Erhardt Barth. Non-parametric texture defect detection using weibull features. In *Image Processing: Machine Vision Applications IV*, volume 7877, pages 150–161. SPIE, 2011.

[13] Jiawei Zhang, Heying Wang, Ying Tian, and Kun Liu. An accurate fuzzy measure-based detection method for various types of defects on strip steel surfaces. *Computers in Industry*, 122:103231, 2020.

[14] Qiwu Luo and Yigang He. A cost-effective and automatic surface defect inspection system for hot-rolled flat steel. *Robotics and Computer-Integrated Manufacturing*, 38:16–30, 2016.

[15] Nirbhar Neogi, Dusmanta K Mohanta, and Pranab K Dutta. Defect detection of steel surfaces with global adaptive percentile thresholding of gradient image. *Journal of the Institution of Engineers (india): Series B*, 98:557–565, 2017.

[16] Mohammadreza Yazdchi, Mehran Yazdi, and Arash Golibagh Mahyari. Steel surface defect detection using texture segmentation based on multifractal dimension. In *2009 International Conference on Digital Image Processing*, pages 346–350. IEEE, 2009.

[17] Yong-hao Ai and Ke Xu. Surface detection of continuous casting slabs based on curvelet transform and kernel locality preserving projections. *Journal of Iron and Steel Research International*, 20(5):80–86, 2013.

[18] Shuangdong Hua, Bin Li, Leshi Shu, Ping Jiang, and Si Cheng. Defect detection method using laser vision with model-based segmentation for laser brazing welds on car body surface. *Measurement*, 178:109370, 2021.

[19] George R Cross and Anil K Jain. Markov random field texture models. *IEEE Transactions on pattern analysis and machine intelligence*, (1):25–39, 1983.

[20] Kechen Song and Yunhui Yan. Micro surface defect detection method for silicon steel strip based on saliency convex active contour model. *Mathematical Problems in Engineering*, 2013(1):429094, 2013.

[21] Ge-Wen Kang and Hong-Bing Liu. Surface defects inspection of cold rolled strips based on neural network. In *2005 international conference on machine learning and cybernetics*, volume 8, pages 5034–5037. IEEE, 2005.

[22] Daniel Soukup and Reinhold Huber-Mörk. Convolutional neural networks for steel surface defect detection from photometric stereo images. In *International symposium on visual computing*, pages 668–677. Springer, 2014.

[23] Hongkai Zhang, Suqiang Li, Qiqi Miao, Ruidi Fang, Song Xue, Qianchuan Hu, Jie Hu, and Sixian Chan. Surface defect detection of hot rolled steel based on multi-scale feature fusion and attention mechanism residual block. *Scientific Reports*, 14(1):7671, 2024.

[24] Zhixuan Zhao, Bo Li, Rong Dong, and Peng Zhao. A surface defect detection method based on positive samples. In *PRICAI 2018: Trends in Artificial Intelligence: 15th Pacific Rim International Conference on Artificial Intelligence, Nanjing, China, August 28–31, 2018, Proceedings, Part II 15*, pages 473–481. Springer, 2018.

[25] Ruoxu Ren, Terence Hung, and Kay Chen Tan. A generic deep-learning-based approach for automated surface inspection. *IEEE transactions on cybernetics*, 48(3):929–940, 2017.

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[27] Louie Giray. Prompt engineering with chatgpt: a guide for academic writers. *Annals of biomedical engineering*, 51(12):2629–2633, 2023.