mathematics

MDPI

Article

A Noise-Robust Deep-Learning Framework for Weld-Defect Detection in Magnetic Flux Leakage Systems

Junlin Yang and Senxiang Lu *

School of Information Science and Engineering, Northeastern University, Shenyang 110819, China; 20225414@stu.neu.edu.cn

* Correspondence: lusenxiang@ise.neu.edu.cn

Abstract: Magnetic flux leakage (MFL) inspection systems are widely used for detecting pipeline defects in industrial sites. However, the acquired MFL signals are affected by field noise, such as electromagnetic interference and mechanical vibrations, which degrade the performance of the developed models. In addition, the noise type or intensity is unknown or changes dynamically during the test phase in contrast to the training phase. To address the above challenges, this paper introduces a novel noise-robust deep-learning framework to remove the noise component in the original signal and learn its noise-invariant feature representation. This can handle the unseen noise pattern and mitigate the impact of dynamic noises on MFL inspection systems. Specifically, we propose a transformer-based architecture for denoising, which encodes noisy input signals into a latent space and reconstructs them into clean signals. We also devise an up–down sampling denoising block to better filter the noise component and generate a noise-invariant representation for weld-defect detection. Finally, extensive experiments demonstrate that the proposed approach effectively improves detection accuracy under both static and dynamic noise conditions, highlighting its value in real-world industrial applications.

Keywords: magnetic flux leakage (MFL); weld-defect detection; noise robustness

MSC: 68T07



Academic Editor: Jonathan Blackledge

Received: 24 March 2025
Revised: 15 April 2025
Accepted: 22 April 2025
Published: 24 April 2025

Citation: Yang, J.; Lu, S. A Noise-Robust Deep-Learning Framework for Weld-Defect Detection in Magnetic Flux Leakage Systems. *Mathematics* **2025**, *13*, 1382. https://doi.org/10.3390/math13091382

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

As a predominant mode of transportation, pipelines play a crucial role in the land and maritime transport of oil and gas, due to their merits of low transportation cost, stable transportation capacity, and uninterrupted transportation [1]. However, as service time extends, the gradual progression and accumulation of defects increasingly threaten the structural integrity and overall safety of pipelines. In this context, non-destructive pipeline inspection is highly valuable, as it enhances operational safety, prolongs service life, and reduces economic losses. Magnetic flux leakage (MFL) has emerged as a crucial non-destructive inspection technique for various metallic materials [2,3], e.g., steel plates and pipelines, due to its efficiency, high sensitivity to damage, and minimal invasiveness [4]. For assessing metal loss in oil and gas pipelines, this method can operate without relying on a specific couplant (medium) and offers high detection reliability. It is particularly effective in detecting metal loss caused by corrosion, scratches, and external metal deposits [5]. For pipeline inspection, MFL systems (robots) automatically collect MFL signals from the pipeline. Then, the raw MFL signals are stored and processed offline, allowing engineers to analyze and identify defects. Recent advancements in MFL inspection and deep-learning

technology have driven research on MFL-based pipeline inspection, leading to improved data processing efficiency and enhanced defect-detection accuracy in pipeline inspection.

The existing work in MFL pipeline inspection mainly focuses on better feature extraction [6–12], data efficiency [13–17], rapid detection [18,19], and system designs [20,21]. Specifically, Mukherjee et al. [18] develop a fast under-sampling scheme for MFL scan points and use the kriging method to reconstruct MFL data for detecting defective areas. Liu et al. [5] propose an improved deep residual network based on the convolution neural network, which automatically learns the features from the MFL image signals and performs the identification of pipeline defects. Shen et al. [15] devise a parallel feature extraction network with hybrid attention and first introduce a semi-supervised-learning strategy to reduce the decision bias of unlabeled samples. Wang et al. [11] study a novel irregular defect size estimation method based on an expertise-informed collaborative network to establish potential relationships between MFL signals and defect sizes. Liu et al. [19] propose LRNet, an online intelligent detection method for the real-time identification of weld defects.

However, the MFL inspection system often operates in complex environments, where certain types of noise, such as electromagnetic interference and mechanical vibrations, can affect the acquired signal. To mitigate such interference, various noise-reduction methods have been proposed in the literature, including wavelet filtering [22], ensemble empirical mode decomposition (EEMD) combined with smoothing filters [23], and more recently, deep-learning-based denoising approaches [24]. A comprehensive review can be found in [3]. This greatly hinders the performance of the developed or trained model if the detection model is not robust enough to noise.

Furthermore, the noise pattern that appears during the test phase may be inconsistent with it during training or even change dynamically [25,26]. This dynamic has not been studied in previous work. The most relevant work is DWWA-Net [24], a dynamic weights-based wavelet attention neural network, which considers strong background noise but ignores its dynamic characteristics.

The performance degradation of models in the presence of noise can be explained from both training and test perspectives [27,28]. (1) During training, the model may be exposed to noisy data, which introduces random fluctuations or irrelevant information. If the model is not robust enough to handle the noise, it might overfit these noisy samples instead of learning the true defect features [26,27]. In other words, the model memorizes the noises rather than generalizing them to the true data distribution. This overfitting can lead to the model being specific to the noisy training data and losing its generalization ability. (2) During the test, the model is deployed on new data that may have different environments or noise magnitudes compared to the training data, and a distribution shift occurs. This means that the data distribution at test time differs from what the model encountered during training, leading to degraded performance, especially exposure to dissimilar noise patterns.

To address the challenges posed by noises, we propose a noise-robust deep-learning framework to adaptively handle different noise distributions. The core idea is to learn noise-invariant feature representation for mitigating the impact of static and dynamic noises. Specifically, we propose a transformer-based encoder-decoder architecture that compresses the noisy signal and reconstructs the raw signal using a reconstruction loss. Additionally, we design an up–down sampling denoising block that filters signal components and generates a noise-invariant representation for defect detection. In summary, the main contributions of this paper are as follows:

- We analyze the impact of various noise patterns on existing models, revealing their susceptibility to degradation and underscoring the need for enhanced noise robustness across diverse environments.
- We introduce a noise-robust framework that effectively handles both static and dynamic noise, leveraging an encoder–decoder architecture with a specialized denoising block to learn noise-invariant feature representations.
- Extensive experiments validate the effectiveness of our approach, showing consistent performance gains across different noise conditions.

2. MFL Inspection System and Data Acquisition

2.1. Principle of MFL in Pipeline Inspection

The working principle of MFL inspection is to judge the severity of defects in an inspected workpiece by measuring the MFL on the surface of a magnetized ferromagnetic using a strong permanent magnet [2]. As shown in Figure 1a, when the material is free of defects, the magnetic field remains uniform and confined within the material. However, when there is a flaw such as corrosion, cracks, or pits, the magnetic flux "leaks" out of the material due to the disruption in the magnetic path. In this case, a magnetic leakage field is formed outside the tube wall. The magnetic sensor detects the leakage field and converts it into electrical signals, which are then processed to assess the condition of the defect.

After inspecting the pipeline, the collected MFL signals encapsulate comprehensive health information. By analyzing MFL data, we can determine the size and location of welds, defects, and other structural features. MFL data are represented as a three-dimensional vector, consisting of:

- Axial component (along the direction of pipeline movement).
- Radial component (perpendicular to the pipe wall) to capture flux leakage intensity.
- Circumferential component (along the pipeline's circumference), which is equipped with uniformly distributed Hall sensors.

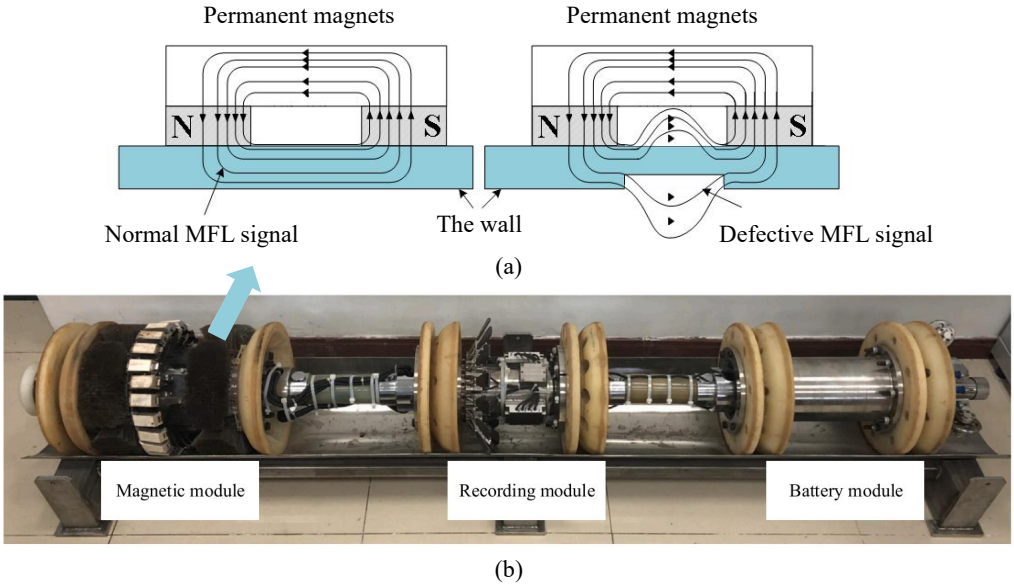


Figure 1. (**a**) Schematic diagram of magnetic flux leakage. (**b**) The MFL inspection system.

Typically, multiple Hall sensors are arranged along the circumferential axis to capture MFL signals as different channels. The values from each sensor channel can be converted into a two-dimensional curve image as shown in Figure 2a, which shows the sample data at specific intervals along the pipeline's forward direction. A smaller sampling

interval increases the number of channels and enhances accuracy, enabling a more detailed inspection. In the curvature image, the horizontal axis represents the travel distance, while the vertical axis corresponds to the magnitude of the MFL signal collected by each channel. In deep-learning methods, curved images often exhibit low contrast, making feature extraction challenging for the network (similar to medical imaging) [13,29]. To address this, pseudo-color image conversion can be applied to enhance contrast and improve feature visibility, as illustrated in Figure 2c,d.

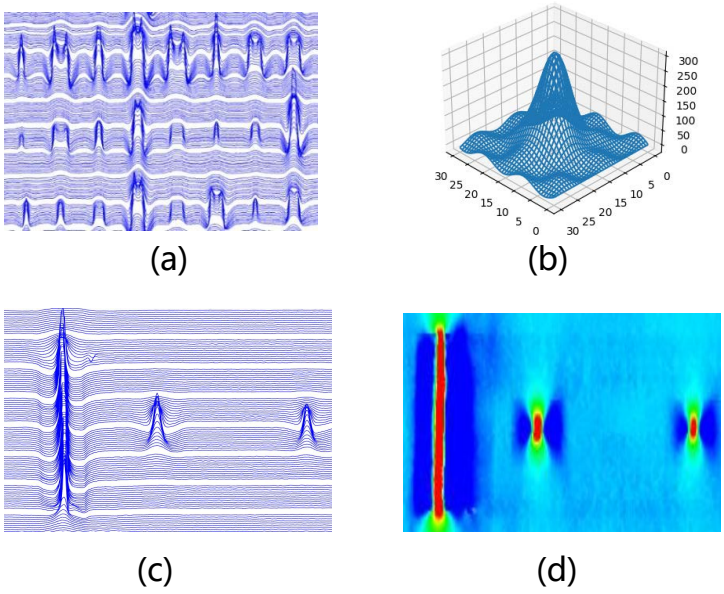


Figure 2. (**a**) Overall defect diagram of a certain pipeline section. (**b**) Single defect 3D view. (**c**) The magnetic flux leakage data (**d**) Converted pseudo-color image.

2.2. MFL Inspection System

The MFL inspection system is illustrated in Figure 1b. The system consists of a magnetic module, a record module, and a battery module [19]. The Hall sensors are arranged continuously in 24 capsules of the magnetic module, and each capsule contains 6 Hall sensors to capture MFL signals. In operational conditions, this robot system is deployed in the pipeline "launching end" and then moves forward at a constant speed of 0.5 m/s under the pressure difference of the medium. During the MFL inspection process, a magnetic field is applied to saturate the tested pipeline. When the pipe wall is uniform and defect-free, the magnetic flux lines induced by the external magnetic field remain entirely confined within the pipeline, and no magnetic field exists outside the pipeline. However, if there is a defect in the pipeline, the magnetic flux lines change direction and follow a path of lower magnetic reluctance, leaking from the pipeline surface to the exterior.

2.3. Data Processing

To analyze pipeline signals using a neural network, the acquired data need to be preprocessed. This process involves noise removal and can be divided into three steps: (1) baseline correction, (2) outlier detection and removal, (3) inter-lobe data interpolation for sensors, and (4) pseudo-color image conversion.

(1) Baseline correction: To address sensor drift or anomalies caused by variations in sensor output under zero-magnetic-field conditions, baseline correction is applied to the leakage data. The average median algorithm is used to calibrate the data for each channel [24]. Specifically, the mean value of the magnetic leakage data collected from each channel is calculated, denoted as S_i, where j represents the channel index, ranging from

1 to N (the total number of channels). If there are K sampling points in the axial direction. The original data are calibrated using their mean value as follows:

$$S_i = \frac{1}{K}\sum_{i=1}^{K} x_{ij} \tag{1}$$

$$x_{ij}^{'} = x_{ij} - S_i + S \tag{2}$$

where x_{ij} represents the magnetic field amplitude at the i-th sampling point for the j-th channel. S represents the median of all channels. x'_{ij} represents the calibrated value. By applying channel mean values for calibration, the influence of sensor drift or anomalies can be eliminated, resulting in more accurate and reliable leakage data.

(2) Outlier detection and removal: This process identifies anomalies arising from sensor failures, environmental disturbances, or human intervention [5,24]. Common abnormal patterns include sudden oscillations and sharp spikes. To effectively mitigate excessive fluctuations, this method classifies a data point as abnormal if its final value surpasses a predefined threshold. The identification criterion is formulated as:

$$Z = \frac{x'_{ij} - \mu}{\sigma} \tag{3}$$

where Z quantifies the deviation of a raw value from the overall mean in terms of standard deviations, and σ denotes the standard deviation. Typically, a threshold of $|Z| > 3$ are considered an outlier and removed.

(3) Inter-lobe data interpolation for sensors: Data interpolation is mainly used to address missing data due to abnormal value removal or other factors. There are three common types of missing data: single-point missing data, entire channel missing data, and randomly missing data [5,24]. Single-point missing data may result from data loss during transmission. Since the amount of missing data is small, its impact on subsequent data analysis is minimal. Therefore, an interpolation method with fast execution speed and moderate precision is required. The cubic spline interpolation algorithm meets this requirement and is adopted to effectively handle missing data. The cubic spline interpolation algorithm is used to handle missing data. The interpolation function is defined as follows:

$$M(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3, \quad x \in [x_i, x_{i+1}] \tag{4}$$

where $M(x)$ is the interpolated function, x_i and x_{i+1} are known data points, and a_i, b_i, c_i, d_i are the coefficients determined by the cubic spline interpolation constraints. These coefficients are obtained by solving a system of equations based on the continuity of the function and its first and second derivatives at the given data points.

(4) Pseudo-color image conversion: After preprocessing the magnetic flux leakage data, it needs to be converted into a pseudo-color image for training the model. First, the preprocessed MFL data are mapped to a grayscale range of [0, 255]. Then, the corresponding grayscale values are further transformed into an RGB three-channel pseudo-color image. Finally, the converted data are shown in Figure 2d.

2.4. Problem Statement

In actual industrial scenarios, the acquired pipeline data usually include both the weld area and the parent material of the welded steel pipe. Due to environmental noise and collection noise, the transformed pipeline images likely contain a large amount of background interference, impairing the model's accuracy. On the other side, the defects and welds have a strong coupling relation (weld defects located in vulnera-

ble areas of the pipeline). These problems affect the accuracy and reliability of the model-detection results [8,10,13,17].

3. Methodology

3.1. Preliminary

3.1.1. Vision Transformer (ViT)

The vision transformer [30] is a pioneering framework that adapts the transformer architecture, originally designed for natural language processing (NLP), to image recognition tasks. Unlike conventional convolutional neural networks (CNNs), ViT processes images through a sequence-based paradigm by treating local image regions as "visual tokens" [30,31].

The core design of ViT involves three key components [30]:

- Image Patch Embedding. An input image $\mathbf{I} \in \mathbb{R}^{H\times W\times C}$ is divided into N non-overlapping patches $\{\mathbf{p}_{i=1}^{N}\}$, where each patch has a resolution of $P \times P$. These patches are flattened into 1D vectors and linearly projected to a D-dimensional embedding space via a trainable matrix $\mathbf{E} \in \mathbb{R}^{(P^2C)\times D}$.
- Positional Encoding. To retain spatial information, learnable positional embeddings $\mathbf{E}_{pos} \in \mathbb{R}^{(N+1)\times D}$ are added to the patch embeddings. An additional [CLS] token is prepended to the sequence to aggregate global features for classification.
- Transformer Encoder. The resulting sequence is fed into a standard transformer encoder comprising L layers. Each layer consists of multi-head self-attention and a feed-forward network (FFN), with layer normalization and residual connections applied to stabilize training. The self-attention mechanism enables global interactions between patches, overcoming the limited receptive fields of CNNs.

For classification, the final state of the $[CLS]$ token is passed through a multi-layer perceptron (MLP) head. ViT demonstrates that pure attention-based models outperform CNNs when trained on large-scale datasets (e.g., ImageNet-21k). However, it lacks inductive biases inherent to CNNs, necessitating careful initialization or pre-training strategies for small datasets.

3.1.2. YOLOS

YOLOS [32] is a modified vision transformer framework designed to extend ViT capabilities from image classification to object detection. Unlike conventional detection pipelines that rely on region proposals or anchor boxes, YOLOS reformulates object detection as a sequence-prediction task, aligning it with the ViT sequence-to-sequence paradigm.

Specifically, YOLOS removes the $[CLS]$ token used in ViT for classification and instead introduces a fixed set of detection tokens (e.g., 100 learnable queries) appended to the patch embeddings. These tokens interact with visual patches via self-attention to capture object-centric features. The input image is split into patches and embedded as in ViT, with positional embeddings retained to encode spatial relationships. Instead of a classification head, YOLOS employs a lightweight feed-forward network on the detection tokens to directly predict bounding box coordinates (x, y, w, h) and class labels. This eliminates the need for handcrafted components like non-maximum suppression. Finally, YOLOS adopts a bipartite matching loss inspired by DETR [33], where Hungarian matching aligns predicted and ground-truth boxes during training. This enforces permutation invariance and avoids redundant predictions.

3.2. Overview

This work follows the architecture of the YOLOS [32], improving its robustness to various noises by integrating a denoising block module. Figure 3 illustrates the overall

framework of our method, which consists of five crucial components: embedding layer, transformer encoder, denoising block, transformer decoder, and detection head. Specifically, we first add random noise into the MFL signal (pseudo-color image) to simulate potential environmental noise. Then, we transform it into a vision token sequence through the embedding layer and extend it with extra learnable detection tokens as input to transformer encoders. After encoding the input tokens through transformer encoders, we adopt a denoising block to eliminate latent noises, obtaining noise-invariant representations of vision and detection tokens. Finally, we reconstruct the raw image from noiseless vision tokens through a decoder and conduct object-detection tasks through a detection head.

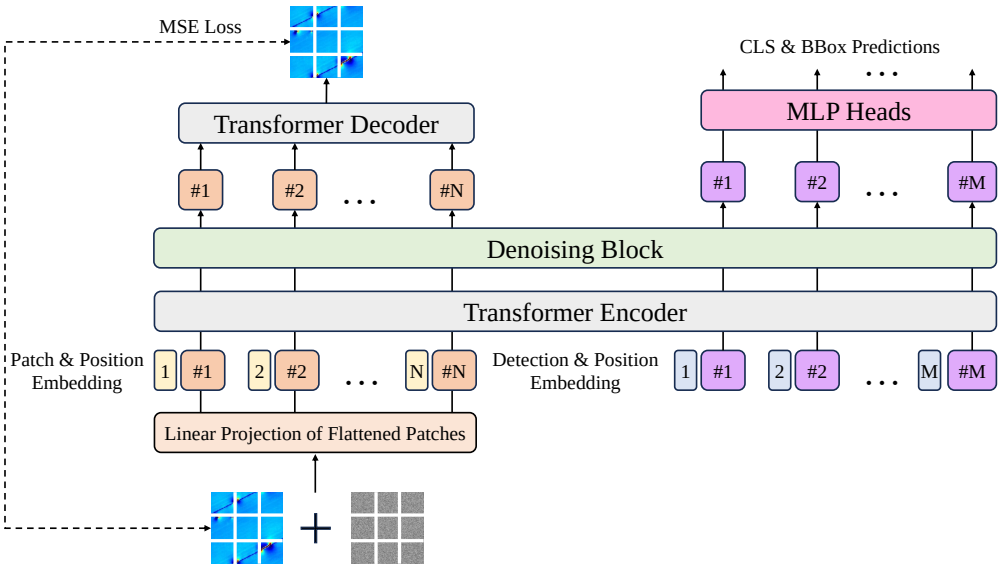


Figure 3. The overall framework of our proposed method.

3.3. Embedding Layer

For an input pipeline image $\mathbf{X} \in \mathbb{R}^{H\times W\times C}$, we first divide it into N patches $\mathbf{X}_p \in \mathbb{R}^{N\times(P^2C)}$, where the resolution of each patch is $P \times P$. Then, these patches are flattened and projected to D dimensional embeddings $\mathbf{X}_{\text{PATCH}} = E(\mathbf{X}_p)$ through an embedding layer E with parameter matrix $\mathbf{E} \in \mathbb{R}^{(P^2C)\times D}$. Meanwhile, following YOLOS, we initialize a series set of learnable detection embeddings $\mathbf{X}_{\text{DET}} \in \mathbb{R}^{K\times D}$ (K is the number of detection embeddings) and append them after the patch embeddings. Finally, we add position embeddings $\mathbf{P} \in \mathbb{R}^{(N+K)\times D}$ to the patch and detection embeddings, obtaining the final input sequence as the input of transformer encoders:

$$\mathbf{Z}_0 = [\mathbf{X}_{\text{PATCH}}, \mathbf{X}_{\text{DET}}] + \mathbf{P}. \tag{5}$$

3.4. Transformer Encoder

The input embeddings $\mathbf{Z}_0$ are then fed into a transformer encoder $\mathcal{T}_{enc}$, which follows the classic architecture of ViT. As shown in Figure 4, the encoder consists of multi-head self-attention and feed-forward layers, which capture global context and semantic relationships within the image patches.

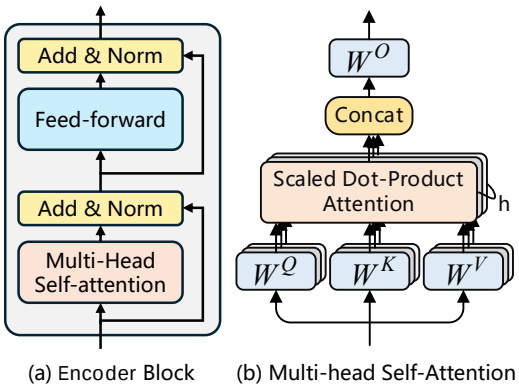


Figure 4. The architecture of the transformer encoder and multi-head self-attention.

The multi-head self-attention module consists of four types of weights: W^Q, W^K, W^V, and $W^O \in \mathbb{R}^{D \times d_{\text{head}}}$, where d_{head} is the dimension of the attention heads, and d_{head}=D/N_{head}. For the input embeddings in ℓ-th layer $\mathbf{Z}_\ell \in \mathbb{R}^{(N+K)\times D}$, the calculation of the multi-head self-attention layer is as follows:

$$\begin{aligned} \mathbf{Z}'_{\ell+1} &= \text{MSA}(\mathbf{Z}_\ell) + \mathbf{Z}_\ell = \left(\text{Concat}(\text{head}_1, \ldots, \text{head}_{N_{\text{head}}})W^O\right) + \mathbf{Z}_\ell, \\ \text{where} \quad \text{head}_i &= \text{Attention}(Q_i, K_i, V_i) = \text{softmax}\left(\frac{(\mathbf{Z}_\ell W_i^Q)(\mathbf{Z}_\ell W_i^K)^\top}{\sqrt{d_{\text{head}}}}\right)\mathbf{Z}_\ell W_i^V \end{aligned} \tag{6}$$

The feed-forward network is an MLP with two linear layers and one non-linear activation function as follows:

$$\begin{aligned} \mathbf{Z}_{\ell+1} &= \text{LN}(\text{FFN}(\text{LN}(\mathbf{Z}'_{\ell+1})) + \mathbf{Z}'_{\ell+1}) \\ &= \text{LN}(\sigma(\text{LN}(\mathbf{Z}'_{\ell+1})W_{\text{up}})W_{\text{down}} + \mathbf{Z}'_{\ell+1}) \end{aligned} \tag{7}$$

where W_{up} and W_{down} are the weights of linear layers, $\text{LN}(\cdot)$ is the layer normalization function, and σ is an activation function (e.g., ReLU, GELU).

3.5. Denoising Block

After L layer encoders, we obtain an intermediate variable $\mathbf{Z}_L = \mathcal{T}_{enc}(\mathbf{Z}_0)$, which is then transformed as a denoised representation, mitigating the impact of noise. The framework of our denoising block is illustrated in Figure 5. It is divided into two stages, namely down-sampling and up-sampling, each consisting of two convolutional layers and one activation layer. Mathematically, for an input embedding $\mathbf{h}_t$, the output $\mathbf{h}_{t+1}$ of the MLP operation at each layer can be calculated as:

$$\mathbf{h}_{t+1} = \text{Conv}_2(\text{ReLU}(\text{Conv}_1(\mathbf{h}_t))), \tag{8}$$

where Conv is the 1D convolution function. The input $\mathbf{Z}_L$ first traverses the down-sampling pathway, which reduces the spatial dimensions of feature maps while extracting hierarchical features. Subsequently, the up-sampling pathway recovers the spatial dimensions of feature maps via transposed convolutions. Finally, the output is generated by integrating the processed features from the up-sampling pathway, producing the denoised representations used for the following detection task.

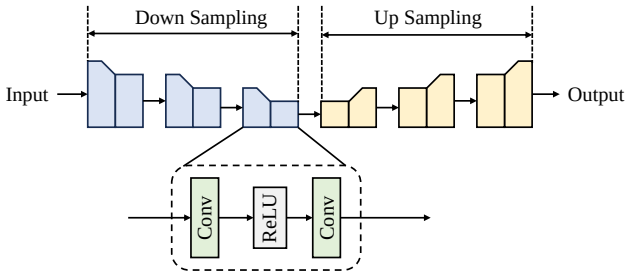


Figure 5. The detailed architecture of the denoising block.

3.6. Noise-Invariant Representation Learning and Object Detection

To obtain an effectively denoised representation of the input embeddings, this model is trained with two main optimization objectives: noise-invariant representation learning and object detection. We first split the denoised intermediate embeddings $\mathbf{Z}'_L$ into patch and detection embeddings. Then, we utilize a lightweight ViT decoder and a linear projection like MAE [31] to reconstruct the raw input image and apply MLP heads to predict classification and bounding box regression. At this point, we expect that regardless of the random noises, this intermediate representation will always remain unaffected.

Particularly, we introduce "noise augmentation" by applying the Gaussian noise at random level to the input during training. Specifically, we perturb the input $\mathbf{X}$ with noise $\mathbf{N}$ sampled from a noise distribution $\mathcal{D}$, such as Gaussian or Poisson noise.

$$\mathbf{X}_{perb} = \mathbf{X} + \mathbf{N}, \quad N \sim \mathcal{D}. \tag{9}$$

The model then learns to recover the raw input $\mathbf{X}$ from the noisy input $\mathbf{X}_{perb}$. This ensures that it remains stable regardless of noise interference.

Afterward, MSE loss, cross-entropy loss, and IoU loss are used to optimize the reconstruction, classification, and bounding box prediction tasks, respectively. The overall loss function can be written as follows:

$$\mathcal{L} = \mathcal{L}_{MSE} + \mathcal{L}_{CLS} + \mathcal{L}_{BBox} + \lambda||\Theta||_2^2. \tag{10}$$

By introducing and optimizing the self-supervised reconstruction task, we force the denoising block to produce representations that can be decoded into the original clean image under random noise interference. Even if the model faces unknown noises in the test phase, we can still mitigate the effect of noise through this denoising block. Finally, the derived representation is used for the defect-detection task.

4. Experiments

4.1. Experimental Setup

To assess the effectiveness of the proposed noise-robust method, we conduct experiments under various noise conditions, including both static and dynamic noise. We conduct experiments using both private and public datasets for the MFL detection and surface defect-detection scenarios, respectively. Additionally, we perform the ablation study and case study to observe.

4.1.1. Datasets

- MFL dataset: The MFL data originate from a pipeline platform in a factory in northern China and includes both artificial and natural corrosion defects. The training data are collected from a pipeline with a diameter of 12 inches, a length of 100 m, and a wall thickness of 12.7 mm. The test data come from a pipeline with a diameter of 16 inches,

a length of 200 m, and a wall thickness of 12.7 mm. The pre-processing pipeline can be divided into four steps: data baseline correction, abnormal data determination and correction, inter-lobe data interpolation of sensors, and data filtering. Then, we converted the collected data into pseudo-color images in four steps: grayscale mapping, smoothing, sharpening, and pseudo-colorization. After converting the collected data into pseudo-color images, we obtain a total of 1405 samples, which then are divided into 70% training set, 15% validation set, and 15% test set. Finally, we add Gaussian noise to the images and convert the images and labels into YOLO format.

- NEU-DET dataset (http://faculty.neu.edu.cn/songkechen/zh_CN/zdylm/263270/list/index.htm, accessed on 18 December 2024): This dataset is a widely used benchmark for surface defect detection in industrial settings. It contains 1800 grayscale images with a 200 × 200 pixel resolution, covering six common types of defects found in hot-rolled steel strips: crazing, inclusion, patches, pitted surface, rolled-in scale, and scratches. Each defect type has 300 images, all captured under consistent conditions to ensure dataset reliability. The dataset is primarily used for tasks such as defect classification, object detection, and segmentation in the field of industrial quality inspection.

4.1.2. Comparison Methods

- Mask R-CNN [34] is a two-stage model for instance segmentation by adding a mask prediction branch, enabling precise object detection and segmentation.
- YOLOv7 [35] is an advanced object-detection model designed to perform real-time object detection with high precision.
- DETR [33] is a transformer-based object-detection mode, which formulates object detection as a direct set prediction problem, eliminating the need for region proposals and post-processing.
- YOLOS [32] a modified vision transformer framework designed to extend ViT capabilities from image classification to object detection.
- DWWA-Net [24] introduces wavelet transform and convolution networks to dynamically filter static noise and improve model convergence for defect detection.

4.1.3. Synthetic Noise in Defect Detection

Static Noise: Noise refers to undesired perturbations that affect acquired signals or images. These perturbations may originate from environmental interference or sensor limitations. Formally, noise is often modeled as an additive component of the observed signal [36]. The noise intensity is typically characterized by its standard deviation σ or power spectral density. If the noise follows a Gaussian distribution, it can be expressed as:

$$n \sim \mathcal{N}(0, \sigma^2) \tag{11}$$

where n represents the noise component, $\mathcal{N}(\mu, \sigma^2)$ denotes a normal distribution with mean μ and variance σ^2, and σ determines the noise magnitude.

In image-based defect detection, the signal-to-noise ratio (SNR) is commonly used to quantify noise levels:

$$SNR = 10 \log_{10}\left(\frac{P_s}{P_n}\right) \tag{12}$$

where P_s and P_n are the power of the signal and noise, respectively.

Dynamic Noise: In many industrial environments, noise is not purely random but exhibits temporal or spatial correlation. Such dynamic noise can be modeled as a combination of Gaussian noise and Sinusoidal interference:

$$n_t = \alpha\epsilon + \beta \sin(\omega t) \tag{13}$$

where:

- n_t is the noise at time step t,
- α is a controlling factor of Gaussian noise,
- $\epsilon \sim \mathcal{N}(0, \sigma^2)$ is independent Gaussian noise,
- β determines the amplitude of the sinusoidal noise,
- ω is the angular frequency of the sinusoidal interference.

This noise model, e.g., Equation (13) captures scenarios where sensor readings are influenced by both random fluctuations (Gaussian noise) and periodic disturbances (sinusoidal component), such as mechanical vibrations or cyclic electromagnetic interference in defect-detection systems.

4.1.4. Implementation Details

Our experiments were conducted on an NVIDIA RTX 3090 GPU (NVIDIA Corporation, Santa Clara, CA, USA), and all methods were implemented using PyTorch 2.1. We trained the model using Adam with a learning rate of 0.001 and a batch size of 32 for 300 epochs. The weight decay is set to 0.0005. Both datasets were split into training/validation/testing sets with a ratio of 7:2:1. We evaluated our model using mAP@0.5 and mAP@0.5:0.95. The best model was selected based on mAP@0.5.

4.2. Results

4.2.1. Performance Without Noise

To validate the overall performance on general surface-defect tasks of the proposed method, we compare defect-detection results of various baselines on the public dataset NEU-DET. As shown in Table 1, our method achieves the highest mAP of 81.2% and mAP@0.5:0.95 of 45.4%, surpassing the second-best baseline (DWWA-Net with 79.0% mAP) by 2.2%. In individual defect categories, our method also excels, reaching 51.6% in crazing, 81.3% in inclusion, 92.4% in patches, 85.8% in pitted surface, 78.2% in rolled in scale, and 96.7% in scratches. The performance improvement can be attributed to the elimination of inherent noise in the dataset. Specifically, compared with other methods, our denoising module integrated into our method effectively removes substantial noise from data collecting. This denoising mechanism alleviates the adverse effects of noise on detection accuracy, enabling our method to achieve superior performance compared with counterparts lacking such a denoising strategy.

Table 1. Performance (%) without noise on the NEU-DET dataset.

Method	mAP	mAP@0.5:0.95	Crazing	Inclusion	Patches	Pitted Surface	Rolled in Scale	Scratches
Mask R-CNN	69.5	32.2	35.5	76.9	88.7	79.2	55.9	80.6
YOLO v7	72.5	35.6	40.3	78.1	91.5	80.7	57.5	86.8
DETR-ResNet18	71.6	34.7	40.6	77.3	92.0	79.4	57.1	83.3
YOLOS	74.3	38.6	41.1	79.2	91.5	82.0	62.7	89.4
DWWA-Net	79.0	42.1	50.1	80.9	90.6	84.8	73.1	94.6
Ours	81.2	45.4	51.6	81.3	92.4	85.8	78.2	96.7

4.2.2. Performance Under Static Noise

In the case of the MFL dataset, Table 2 systematically demonstrates the robustness of the proposed method under diverse levels of static noise for MFL weld-defect detection. In this experiment, we introduce static noise of different levels according to Equation (12) to the test data. Across all specified noise conditions, our approach consistently exhibits superiority over comparative methods. When we do not introduce noise (i.e., N/A), our method achieves the highest mAP of 99.2% and mAP@0.5:0.95 of 59.9%. As noise levels increase (50 dB $\rightarrow$ 20 dB $\rightarrow$ 10 dB), the performance of the baselines decreases significantly, while our method largely maintains its original performance. This is because our method adds random noise to the original image during the training phase, and learns to construct noise-invariant image representations through the reconstruction task. With our denoise block, our method extracts the clean noise-invariant image representations and uses them for the object-detection task during the inference phase, avoiding noise interference. When SNR is 0 dB (i.e., the noise power is the same as the signal power), our technique achieves an mAP of 90.7% and mAP@5:.95 of 50.4%, showcasing a remarkable advantage over the second-best DWWA-Net (70.9% mAP), further validating the robustness of our method to noise.

Table 2. Performance (%) of MFL weld-defect detection under different noise levels.

Noise Level	Mask R-CNN		YOLOv7		DETR		YOLOS		DWWA-Net		Ours	
	mAP	mAP@ 0.5:0.95	mAP	mAP@ 0.5:0.95	mAP	mAP@ 0.5:0.95	mAP	mAP@ 0.5:0.95	mAP	mAP@ 0.5:0.95	mAP	mAP@ 0.5:0.95
0 dB	38.6	8.9	50.2	18.6	48.1	17.9	51.7	20.3	70.9	41.0	90.7	50.4
10 dB	52.7	20.8	68.5	31.9	67.4	32.6	69.1	35.4	84.2	48.3	93.1	55.8
20 dB	65.3	36.2	86.9	43.4	81.4	44.1	84.7	46.3	93.0	52.8	95.6	57.1
50 dB	82.1	49.8	97.1	54.5	94.3	53.9	96.4	55.9	96.8	56.4	98.9	58.6
N/A	97.9	53.4	97.7	55.5	97.5	56.9	97.3	58.1	98.4	58.6	99.2	59.9

4.2.3. Performance Under Dynamic Noise

We compare the performance of our method for defect detection with baselines under dynamic noise on the MFL dataset. The dynamic noises are conducted based on Equation (13). For the independent Gaussian noise in the equation, we control its level at 50 dB. When $\beta = 0$, the noise is static noise at 50 dB. As shown in Table 3, across various $\alpha - \beta$ combinations, our method consistently outperforms competitors. For instance, when $\alpha = 0.8$, $\beta = 0.2$, our method achieves 98.4% mAP and 59.0% mAP@0.5:0.95, exceeding other methods. As α decreases and β increases, this means that static noise decreases while dynamic noise increases, resulting in overall performance gradually declining. However, our proposed method is relatively less affected by noise and has a more robust performance. In the setting of $\alpha = 0.0$ and $\beta = 1.0$, ours leads with 92.1% mAP and 55.3% mAP@0.5:0.95. This excellence originates from the denoising module integrated into our method and paired training scheme. This method effectively addresses dynamic noise controlled by α and β and reduces noise interference, enhancing feature robustness. Compared with other methods lacking such a denoising mechanism, our approach more robustly alleviates the adverse effects of dynamic noise, thus achieving higher detection precision across diverse $\alpha - \beta$ noise configurations.

Table 3. Performance (%) of MFL defect detection under dynamic noise.

α	β	Mask R-CNN		YOLOv7		DETR		YOLOS		DWWA-Net		Ours	
		mAP	mAP@ 0.5:0.95	mAP	mAP@ 0.5:0.95	mAP	mAP@ 0.5:0.95	mAP	mAP@ 0.5:0.95	mAP	mAP@ 0.5:0.95	mAP	mAP@ 0.5:0.95
0.8	0.2	90.4	54.2	96.5	57.9	95.6	57.3	95.8	57.5	97.7	58.6	98.4	59.0
0.6	0.4	84.2	50.5	91.4	54.8	89.8	53.9	90.3	54.2	94.6	56.8	96.8	58.1
0.4	0.6	78.1	46.9	86.2	51.7	84.0	50.4	84.8	50.9	91.6	54.9	95.3	57.2
0.2	0.8	72.0	43.2	81.1	48.6	78.2	46.9	79.2	47.5	88.5	53.1	93.7	56.2
0.0	1.0	65.9	39.5	75.9	45.5	72.4	43.4	73.7	44.2	85.5	51.3	92.1	55.3

4.2.4. Ablation Study

To assess the contributions of different components in our approach, we conduct an ablation study by selectively removing each component under static noise of 0 dB. The results in Table 4 demonstrate that incorporating only the noise augmentation slightly improves performance, suggesting its effectiveness in enhancing model robustness. After introducing the reconstruction loss, the detect performance is further enhanced, indicating its role in preserving feature integrity. The denoising block brings additional improvement, boosting robustness against variations in the input. The full method, integrating all three components, achieves the best performance, confirming that each component contributes to overall accuracy, yielding the highest mAP scores.

Table 4. Ablation study on different components in the MFL dataset.

Denoising Block	Reconstruction Loss	Noise Augmentation	mAP	mAP@0.5:0.95
✗	✗	✗	51.7	20.3
✓	✗	✗	58.6	23.8
✗	✓	✗	67.4	28.7
✗	✗	✓	66.3	28.9
✗	✓	✓	78.4	35.5
✓	✓	✓	90.7	50.4

4.2.5. Visualization

As presented in Figure 6, qualitative validation of our method's denoising efficacy is provided through a representative test image (000476.jpg). The clean column displays the noise-free original image with defect annotations. When exposed to 0dB Gaussian noise, the image undergoes severe degradation. Noise artifacts obscure fine defect structures, particularly in regions with lower confidence scores. In contrast, the denoised reconstruction demonstrates superior noise robustness while preserving critical defect details. The denoised image exhibits sharper defect edges and higher confidence predictions, indicating improved feature fidelity. Notably, our method effectively recovers fine defects indistinguishable from the noisy input. This visual improvement aligns closely with our quantitative results, confirming that the denoising module enhances defect visibility under extreme noise conditions. By prioritizing noise reduction while retaining structural information, our approach ensures the detection operates on high-quality representations, thereby improving both localization accuracy and confidence in defect detection.

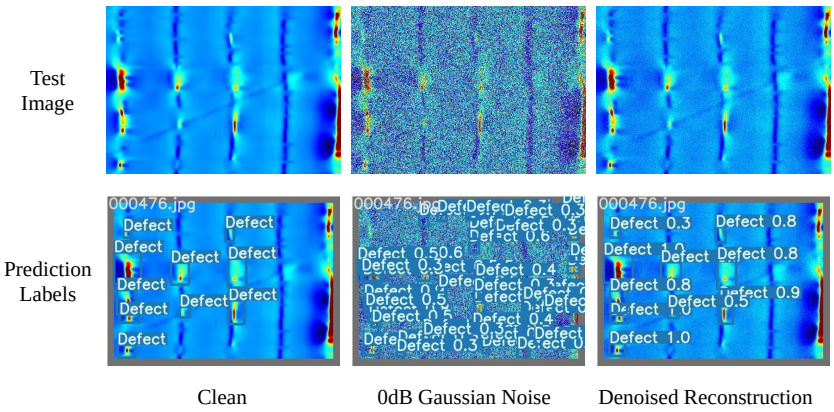


Figure 6. Visualization of a test image and the prediction under 0 dB noise and after denoised by our method.

5. Conclusions

In this paper, we address the challenge of noise robustness in the MFL pipeline inspection system. Due to dynamic noise on defect-detection performance, we design a transformer-based encode–decode architecture that effectively compresses and reconstructs MFL signals while removing noise. Additionally, we introduce a tailored denoising block that enhances noise-invariant feature learning, thereby improving model generalization in varying noise environments. Through extensive experiments, we demonstrate that our proposed framework significantly enhances defect-detection performance under diverse noise conditions. The results confirm the effectiveness of our approach in mitigating both static and dynamic noise. Specifically, under severe static noise (0 dB SNR), our method attains 90.7% mAP and 50.4% mAP@0.5:0.95, outperforming the second-best baseline (DWWA-Net) by 19.8% and 9.4% in absolute terms, respectively. For dynamic noise scenarios, our approach maintains 92.1% mAP and 55.3% mAP@0.5:0.95, surpassing DWWA-Net by 6.6% and 4.0%. Even in noise-free conditions, our method achieves 99.2% mAP on the MFL dataset and 81.2% mAP on the NEU-DET benchmark, representing 0.8% and 2.2% improvements over existing methods. The ablation study confirms the critical role of our denoising block, with its inclusion boosting performance by 39.0% mAP under extreme noise. These results validate our framework's ability to mitigate both static and dynamic noise interference while maintaining high detection accuracy, making it particularly suitable for real-world industrial applications where noise patterns are unpredictable.

Our method depends on the reconstruction task, and its applicability will be challenged on datasets without clean images. We will leave this to future work.

Author Contributions: Conceptualization, methodology, and validation, S.L.; writing—original draft preparation, writing—review and editing, J.Y.; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The MFL dataset used in this study is private and cannot be shared. However, the NEU-DET dataset is publicly available and can be accessed at: https://drive.google.com/file/d/1qrdZlaDi272eA79b0uCwwqPrm2Q_WI3k/view, accessed on 18 December 2024.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Biezma, M.; Andrés, M.; Agudo, D.; Briz, E. Most fatal oil & gas pipeline accidents through history: A lessons learned approach. *Eng. Fail. Anal.* **2020**, *110*, 104446.
2. Shi, Y.; Zhang, C.; Li, R.; Cai, M.; Jia, G. Theory and application of magnetic flux leakage pipeline detection. *Sensors* **2015**, *15*, 31036–31055. [CrossRef]
3. Feng, B.; Wu, J.; Tu, H.; Tang, J.; Kang, Y. A review of magnetic flux leakage nondestructive testing. *Materials* **2022**, *15*, 7362. [CrossRef] [PubMed]
4. Huang, S.; Peng, L.; Sun, H.; Li, S. Deep learning for magnetic flux leakage detection and evaluation of oil & gas pipelines: A review. *Energies* **2023**, *16*, 1372. [CrossRef]
5. Liu, S.; Wang, H.; Li, R. Attention module magnetic flux leakage linked deep residual network for pipeline in-line inspection. *Sensors* **2022**, *22*, 2230. [CrossRef]
6. Yang, L.; Wang, Z.; Gao, S. Pipeline magnetic flux leakage image detection algorithm based on multiscale SSD network. *IEEE Trans. Ind. Inform.* **2019**, *16*, 501–509. [CrossRef]
7. Yang, L.; Wang, Z.; Gao, S.; Shi, M.; Liu, B. Magnetic flux leakage image classification method for pipeline weld based on optimized convolution kernel. *Neurocomputing* **2019**, *365*, 229–238. [CrossRef]
8. Liu, J.; Shen, X.; Wang, J.; Jiang, L.; Zhang, H. An intelligent defect detection approach based on cascade attention network under complex magnetic flux leakage signals. *IEEE Trans. Ind. Electron.* **2022**, *70*, 7417–7427. [CrossRef]
9. Zhao, H.; Liu, J.; Tang, J.; Shen, X.; Lu, S.; Wang, Q. A MFL mechanism-based self-supervised method for defect detection with limited labeled samples. *IEEE Trans. Instrum. Meas.* **2022**, *72*, 1–10. [CrossRef]
10. Shen, X.; Liu, J.; Jiang, L.; Liu, X.; Zhang, H. A Novel Weld Defect Detection Method for Intelligent Magnetic Flux Leakage Detection System via Contextual Relation Network. *IEEE Trans. Ind. Electron.* **2023**, *71*, 6304–6314. [CrossRef]
11. Wang, L.; Zhang, H.; Liu, J.; Shen, X.; Zuo, F. Irregular Defect Size Estimation for the Magnetic Flux Leakage Detection System via Expertise-Informed Collaborative Network. *IEEE Trans. Ind. Electron.* **2024**, *71*, 13189–13200. [CrossRef]
12. Zhao, H.; Liu, J.; Wang, Q.; Shen, X.; Jiang, L. A novel anomaly detection method for magnetic flux leakage signals via a feature-based unsupervised detection network. *Comput. Ind.* **2025**, *164*, 104190. [CrossRef]
13. Liu, X.; Liu, J.; Wang, Z.; Wang, L.; Zhang, H. Basic-class and cross-class hybrid feature learning for class-imbalanced weld defect recognition. *IEEE Trans. Ind. Inform.* **2022**, *19*, 9436–9446. [CrossRef]
14. Shen, X.; Liu, J.; Zhang, H.; Jiang, L.; Zhao, H.; Yang, H. A Novel Incremental Defect Detection Method via Elastic Heterogeneous Distillation Network. *IEEE Trans. Autom. Sci. Eng.* **2024**, *22*, 10149–10161. [CrossRef]
15. Shen, X.; Liu, J.; Sun, J.; Jiang, L.; Zhao, H.; Zhang, H. SSCT-Net: A semisupervised circular teacher network for defect detection with limited labeled multiview MFL samples. *IEEE Trans. Ind. Inform.* **2023**, *19*, 10114–10124. [CrossRef]
16. Wang, L.; Liu, J.; Zhang, H.; Zuo, F. KMSA-Net: A knowledge-mining-based semantic-aware network for cross-domain industrial process fault diagnosis. *IEEE Trans. Ind. Inform.* **2023**, *20*, 2738–2750. [CrossRef]
17. Liu, J.; Li, H.; Zuo, F.; Zhao, Z.; Lu, S. Kd-lightnet: A lightweight network based on knowledge distillation for industrial defect detection. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 1–13. [CrossRef]
18. Mukherjee, S.; Huang, X.; Udpa, L.; Deng, Y. A kriging-based magnetic flux leakage method for fast defect detection in massive pipelines. *J. Nondestruct. Eval. Diagn. Progn. Eng. Syst.* **2022**, *5*, 011002. [CrossRef]
19. Liu, J.; Wen, Z.; Shen, X.; Zuo, F.; Jiang, L.; Zhang, H. Online Pipeline Weld Defect Detection for Magnetic Flux Leakage Inspection System via Lightweight Rotated Network. *IEEE Trans. Ind. Electron.* **2024**, *early access*. [CrossRef]
20. Heo, C.G.; Kim, Y.C.; Bae, J.H.; Park, G.S. A Novel Magnetic Flux Leakage Sensor System for Inspecting Large Diameter Pipeline. *J. Magn.* **2022**, *27*, 100–105. [CrossRef]
21. Cui, W.; Xiao, Z.; Feng, Z.; Yang, J.; Zhang, Q. A magnetic flux leakage detector for ferromagnetic pipeline welds with a magnetization direction perpendicular to the direction of travel. *Sensors* **2024**, *24*, 5158. [CrossRef] [PubMed]
22. Zhang, O.; Wei, X. De-noising of magnetic flux leakage signals based on wavelet filtering method. *Res. Nondestruct. Eval.* **2019**, *30*, 269–286. [CrossRef]
23. Wang, D.; Zhu, L.; Yue, J.; Lu, J.; Li, G. Denoising method of natural gas pipeline leakage signal based on empirical mode decomposition and improved Bhattacharyya distance. *Eng. Res. Express* **2021**, *3*, 035030. [CrossRef]
24. Liu, J.; Zhao, H.; Chen, Z.; Wang, Q.; Shen, X.; Zhang, H. A dynamic weights-based wavelet attention neural network for defect detection. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *35*, 16211–16221. [CrossRef] [PubMed]
25. Zhang, W.; Li, C.; Peng, G.; Chen, Y.; Zhang, Z. A deep convolutional neural network with new training methods for bearing fault diagnosis under noisy environment and different working load. *Mech. Syst. Signal Process.* **2018**, *100*, 439–453. [CrossRef]
26. Zhou, K.; Liu, Z.; Qiao, Y.; Xiang, T.; Loy, C.C. Domain generalization: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 4396–4415. [CrossRef]
27. Zhang, C.; Bengio, S.; Hardt, M.; Recht, B.; Vinyals, O. Understanding deep learning requires rethinking generalization. In Proceedings of the International Conference on Learning Representations, San Juan, Puerto Rico, 2–4 May 2016.

28. Recht, B.; Roelofs, R.; Schmidt, L.; Shankar, V. Do imagenet classifiers generalize to imagenet? In Proceedings of the 36th International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 5389–5400.
29. Zhou, S.K.; Greenspan, H.; Davatzikos, C.; Duncan, J.S.; Van Ginneken, B.; Madabhushi, A.; Prince, J.L.; Rueckert, D.; Summers, R.M. A review of deep learning in medical imaging: Imaging traits, technology trends, case studies with progress highlights, and future promises. *Proc. IEEE* **2021**, *109*, 820–838. [CrossRef]
30. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16 $\times$ 16 Words: Transformers for Image Recognition at Scale. In Proceedings of the International Conference on Learning Representations, Vienna, Austria, 3–7 May 2021.
31. He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; Girshick, R. Masked autoencoders are scalable vision learners. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 16000–16009.
32. Fang, Y.; Liao, B.; Wang, X.; Fang, J.; Qi, J.; Wu, R.; Niu, J.; Liu, W. You Only Look at One Sequence: Rethinking Transformer in Vision through Object Detection. In Proceedings of the Neural Information Processing Systems, Virtual, 6–14 December 2021.
33. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In *Proceedings of the European Conference on Computer Vision*; Springer: Cham, Switzerland, 2020; pp. 213–229.
34. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
35. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 7464–7475.
36. Goodfellow, I.; Bengio, Y.; Courville, A.; Bengio, Y. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016; Volume 1.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.