fractal and fractional

MDPI

Article

Multi-Scale Feature Convolutional Modeling for Industrial Weld Defects Detection in Battery Manufacturing

Waqar Riaz [1,2], Xiaozhi Qi [1,2], Jiancheng (Charles) Ji [2,*] and Asif Ullah [1,2]

[1] Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China; riazwaqar@szpu.edu.cn (W.R.); xz.qi@siat.ac.cn (X.Q.); asifkh@szpu.edu.cn (A.U.)

[2] Institute of Intelligent Manufacturing Technology, Shenzhen Polytechnic University, 4089 Shahe West Road, Shenzhen 518055, China

[*] Correspondence: jcji20@szpu.edu.cn

Abstract

Defect detection in lithium-ion battery (LIB) welding presents unique challenges, including scale heterogeneity, subtle texture variations, and severe class imbalance. We propose a multi-scale convolutional framework that integrates EfficientNet-B0 for lightweight representation learning, PANet for cross-scale feature aggregation, and a YOLOv8 detection head augmented with multi-head attention. Parallel dilated convolutions are employed to approximate self-similar receptive fields, enabling simultaneous sensitivity to fine-grained microstructural anomalies and large-scale geometric irregularities. The approach is validated on three datasets including RIAWELC, GC10-DET, and an industrial LIB defects dataset, where it consistently outperforms competitive baselines, achieving 8–10% improvements in recall and F1-score while preserving real-time inference on GPU. Ablation experiments and statistical significance tests isolate the contributions of attention and multi-scale design, confirming their role in reducing false negatives. Attention-based visualizations further enhance interpretability by exposing spatial regions driving predictions. Limitations remain regarding fixed imaging conditions and partial reliance on synthetic augmentation, but the framework establishes a principled direction toward efficient, interpretable, and scalable defect inspection in industrial manufacturing.

Keywords: multi-scale feature processing; YOLO; transfer learning; industrial defect detection; inline inspection systems; intelligent manufacturing

check for updates

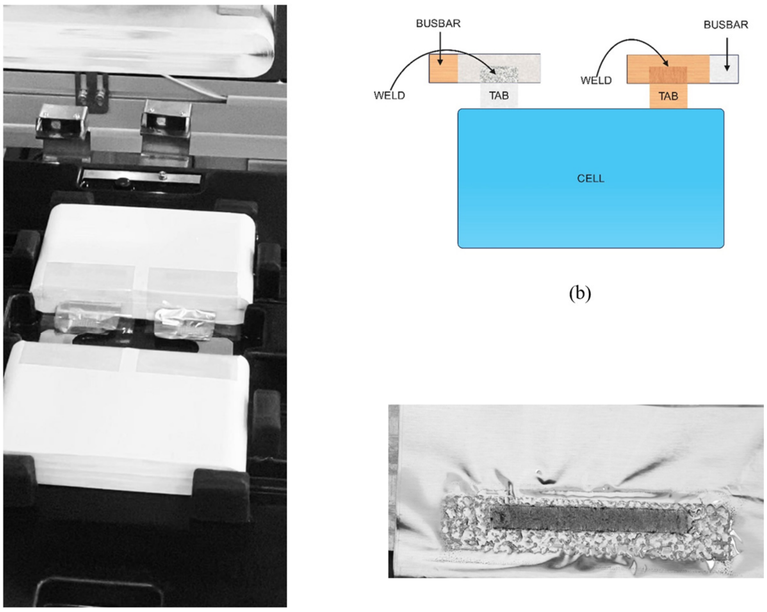
Academic Editor: Lyudmyla Kirichenko

Received: 4 August 2025
Revised: 10 September 2025
Accepted: 19 September 2025
Published: 21 September 2025

Citation: Riaz, W.; Qi, X.; Ji, J.; Ullah, A. Multi-Scale Feature Convolutional Modeling for Industrial Weld Defects Detection in Battery Manufacturing. *Fractal Fract.* **2025**, *9*, 611. https://doi.org/10.3390/fractalfract9090611

1. Introduction

To meet the growing demand for electric vehicles (EVs) and renewable energy storage solutions, the production of lithium-ion batteries is growing exponentially. Ultrasonic welding is a key step in many of the phases of the manufacturing of lithium-ion batteries; one of them is the TAB welding where it connects the battery's metal electrode sheets (usually copper and aluminum electrodes) to the polar supports (terminals), thus forming the negative and positive terminals of the battery as illustrated in Figure 1. Subsequently, several industries use this welding technique to weld different metal parts together without generating excessive heat; thus, in LIBs manufacturing, it is used while preserving the integrity of sensitive battery components and ensuring high structural performance [1]. However, ultrasonic welding can also cause defects such as poor joints, shear, and misalignment, etc., which can harm battery performance and pose safety risks [2]. Accurate and rapid identification of these defects is essential to prevent damaged batteries from entering the market, which may bring concerns of human safety if any accident happens while driving. Traditional quality control methods in lithium-ion battery manufacturing typically rely on manual inspections and offline testing, which are ineffective and inconsistent in detecting critical defects. With advances in computer vision and artificial intelligence (AI), automated real-time defect detection has become reliable for using computer vision algorithms in ultrasonic weld monitoring systems to improve the accuracy, consistency, and speed of defect classification [3]. Typical defects include porosity, burns, cracks, and incomplete welding, etc., which negatively affect battery conductivity and battery life, but are often difficult to detect accurately using traditional methods [4]. These challenges can be addressed using advanced computer vision techniques, such as convolutional neural networks (CNNs). Due to their high accuracy and scalability, these models are widely used in several real-time deep learning applications [5,6].



(b)

(c)

(a)

Figure 1. (**a**) Shows the industrial setting for cell after the weld defect detection; (**b**) shows block diagram of cell's TAB welded with busbar using ultrasonic welding; (**c**) shows the OK type situation after ultrasonic welding.

Recent research has shown that machine learning (ML) and computer vision can effectively automate defect classification in manufacturing. Deep learning models, including CNNs and vision transformers, are commonly used for real-time applications; some of them have been used to identify surface inconsistencies during welding and assess structural integrity and these architectures enable accurate distinction between flawless and defective welds across large datasets [7]. For example, Fan et al. [8] developed an image-feature-based system for laser weld seam failure detection in automotive components, while Basamakis et al. [9] proposed a deep semantic segmentation framework to detect seam gaps in fixtured workpieces with high accuracy. Despite these advances, ultrasonic weld inspection in LIB production remains uniquely challenging. Defects vary in size, shape, and appearance, and visual cues are often subtle against complex weld textures. Moreover, inspection must operate in real time to match high-throughput production lines. To address this, we introduce a deep learning-based framework tailored for LIB weld defects that balances accuracy, interpretability, and efficiency. EfficientNet serves as the lightweight backbone for feature extraction [10], while multi-head attention enhances sensitivity to small or occluded defects [11].

To bridge the need for real-time, high-accuracy defect detection in LIB manufacturing, we propose a lightweight, multi-scale, attention-enhanced CNN integrated into the YOLOv8 framework. Unlike prior methods, our model balances detection granularity and inference speed, making it suitable for deployment in high-throughput battery inspection lines. We adopt a single-stage detector with an anchor-free head due to its streamlined integration with our multi-scale feature pipeline and stable performance on weld imagery. Empirically, this choice offers comparable accuracy to newer iterations while simplifying model customization and training. Our experiments therefore prioritize a stable detector that integrates cleanly with our multi-scale feature stack and transfer learning procedure, allowing the paper to focus on the core contribution. We adopt YOLOv8 as the foundational detector [12–15], condensing its comparison with later versions into a single practical justification: YOLOv8 offers a stable, anchor-free design that integrates cleanly with our multi-scale attention modules, delivering reliable performance without unnecessary architectural complexity. This balance enables deployment on industrial inspection lines where both speed and precision are critical.

Unlike transformer-heavy detectors that achieve accuracy at high computational cost, or lightweight CNNs that sacrifice sensitivity to delicate defects, our approach targets a multi-scale, real-time, and interpretable solution by enabling inline inspection without specialized heavy GPU infrastructure. The remainder of this paper is structured as follows: Section 2 provides a detailed review of related work, highlighting existing approaches to defect detection in industrial welding and LIB manufacturing. Section 3 outlines the proposed methodology, describing the integration of EfficientNet, multi-head attention, and YOLOv8, along with our training and transfer learning strategy, as well as the dataset preparation process, including data collection, annotation standards, and augmentation techniques. Section 4 details the experimental setup and quantitative and qualitative results, including performance comparisons and ablation studies.

2. Related Work

In manufacturing lithium-ion batteries (LIBs), usually ultrasonic welding is a preferred technique to bond dissimilar metals, such as copper and aluminum, at the battery terminals. Ensuring weld quality in LIB production is traditionally performed with non-destructive testing (NDT) methods such as phased-array ultrasonics and laser ultrasonics [4,16]. These approaches provide high-resolution internal imaging but are limited in scalability, as they require costly instrumentation, expert interpretation, and cannot be seamlessly integrated into high-throughput production environments [17]. Their reliance on specialized sensing pipelines restricts inline applicability, highlighting the need for automated visual inspection.

Traditional inspection of weld quality has relied on feature-based and handcrafted approaches. For instance, Fan et al. [8] proposed an image-feature extraction method to detect laser weld seam failures in automotive brake joints. Such methods demonstrate feasibility but are sensitive to noise, hand-engineering choices, and cannot scale to the complexity of LIB ultrasonic welds. Deep learning has brought significant advances. Basamakis et al. [9] introduced a semantic segmentation framework for seam gap detection in laser welding, demonstrating accurate spatial localization of defects. Li et al. [18] extended this trend with a semi-supervised segmentation strategy for robotic welding, addressing limited labels but requiring computationally intensive training pipelines. These works illustrate the strengths of CNN-based architectures but also their limitations: while effective for coarse weld anomalies, CNNs struggle with subtle localization of micro-defects such as porosity or boundary cracks in ultrasonic weld imagery.

Similarly, broader AI reviews such as Rahman et al. [19] emphasize that while domain-specific progress is rapid, translating heavy architectures into real-time industrial pipelines remains unresolved. More sophisticated networks have been proposed. More recently, transformer-based detectors [20] have demonstrated strong accuracy by modeling long-range dependencies through global self-attention. Wang et al. [21] presented 3DWDC-Net, a 3D CNN enhanced with separable structure and global attention, achieving high accuracy for weld defect classification from phased-array ultrasonic tomography. While promising, these transformer-like attention models and volumetric CNNs are computationally expensive and unsuitable for inline inspection, where inference must operate at production-line speeds.

Multi-Scale and Attention-Driven Architectures

Prior detectors for weld and steel-surface defects frequently rely on single-scale or narrow-band receptive fields, which limit sensitivity to subtle micro-textures and larger geometric patterns; few works report consistent per-class gains across both weld and steel benchmarks, and explainability is rarely quantified. Our approach directly addresses these gaps by (i) engineering parallel dilations that approximate scale-continuous, multi-scale receptive fields, or robust multi-scale coverage, (ii) reporting class-wise results on weld (RIAWELC) and steel (GC10-DET) alongside an industrial dataset, and (iii) validating attention to defect regions with gradient-guided localization maps [22,23]. Collectively, these design and evaluation choices provide scale-aware accuracy and this combination improves class-wise detection (e.g., crease, fold) and provides interpretable evidence for industrial quality engineering.

Multi-scale modeling has been widely adopted in defect detection, where strategies such as atrous convolutions [24] and feature pyramids [25] aim to capture features across resolutions. However, these methods often lack explicit validation on industrial weld images, where both micro-textures and macro-geometry matter. Our work leverages a structured multi-scale representation with self-similar receptive fields to better accommodate the heterogeneous scales of LIB weld defects. Real-time deployment in industrial QA requires balancing accuracy with efficiency. Lightweight detectors such as SSD, MobileNet-based variants, and recent evolutions like PP-YOLOE-S [26] provide speed but have not demonstrated consistent performance on high-resolution weld defects. Our approach maintains a low-parameter footprint while sustaining accuracy on both public and industrial datasets, demonstrating suitability for deployment in high-throughput LIB manufacturing lines. Transformer-based detectors such as DETR, Deformable DETR, and RT-DETR have introduced powerful self-attention mechanisms for object detection [27–29]. While these models excel on large-scale benchmarks, their heavy computational cost hinders industrial deployment. By integrating a multi-head attention block within a lightweight CNN framework, we achieve transformer-level feature refinement while retaining YOLO-level inference speed.

Taken together, prior work highlights three limitation: feature-engineering methods are not scalable for LIB ultrasonic welds, CNN-based detectors are efficient but insufficiently sensitive to subtle localization, and transformer-based or 3D CNN models are too heavy for inline deployment. Our contribution directly addresses these gaps by introducing a lightweight, multi-scale architecture. Through parallel dilated convolutions that emulate scale-continuous receptive fields, PANet-based cross-scale fusion, and multi-head attention refinement, our framework achieves real-time, interpretable, and scale-aware weld defect detection suitable for industrial LIB manufacturing.

3. Methodology

In this section, we propose a pipeline using EfficientNet for feature extraction, YOLOv8 for real-time detection, and multi-head attention to achieve more refined features to detect welding defects during ultrasonic welds in vehicle LIBs manufacturing. While the advanced version of YOLO offers potential enhancements, at the time of model development and experimentation for this research, YOLOv8 represented a thoroughly validated and stable detection framework, extensively documented and benchmarked across diverse industrial applications. YOLOv8's stability ensured reliable integration with our custom modules (such as EfficientNet backbone and multi-head attention mechanisms), significantly reducing experimental risk in industrial environments. The proposed model represents a defect detection algorithm grounded in deep neural networks. Its most prominent attribute lies in its high-speed operational capabilities, rendering it particularly suited for real-time systems. Subsequent sections provide proper step-by-step deployment details about the method.

3.1. Model Overview

In this study we proposed an EfficientNet with multi-head attention-based YOLOv8 defect detection model for detection of defects in vehicle LIB ultrasonic welds, presenting unique challenges, including the need to identify small-scale anomalies (e.g., micro-cracks or fold) or other larger structural defects like shear and porosity. Achieving this balance of fine-grained detection and computational efficiency requires a model that can generalize across varying defect types while remaining adaptable to new defect scenarios introduced during production. In the first phase, the model used EfficientNet, which efficiently captures multi-scale features; it ensures that the system remains reliable under diverse manufacturing conditions. Therefore, EfficientNet-B0 [30] is utilized as a backbone network that attempts to achieve feature representation through its lightweight model that enables its use in edge devices and provides real-time quality assurance without loss; it is therefore a perfect backbone network for real-time detection. EfficientNet's architecture utilizes a combination of depthwise convolutions, inverted residual bottlenecks, and channel-wise attention (Squeeze-and-Excitation) modules, effectively capturing hierarchical representations from weld images. Additionally, to address the multi-scale and fractal-like complexity of real-world defects, our model integrates multi-scale dilated convolution blocks within the feature extraction process. Each of these blocks runs parallel convolutional filters with dilation rates of 1, 2, and 5, aggregating their outputs to emulate the behavior of a fractional integral operator. While our multi-branch convolutional design exhibits self-similarity and multi-scale dilation patterns, we do not claim a formal fractal construct. Instead, we leverage architectural intuition from prior works like FractalNet to efficiently capture spatial hierarchies across multiple receptive fields. This fractal-inspired design expands the effective receptive field and enriches features across scales, all without increasing parameter count, supporting both computational efficiency and superior sensitivity to diverse defect morphologies. Grounded in fractional-order systems theory and aligned with the latest advances in fractal neural network, the features extraction flows are as follows:

Low-Level Features (initial layers): capture fundamental image attributes such as edges, textures, and subtle gradients, which are crucial for recognizing simple yet critical defect markers, such as edges of cracks or porosity boundaries.

Mid-Level Features (intermediate layers): identify structured features like shapes, sizes, and continuity disruptions critical for differentiating complex structural defects (e.g., fold and crease).

High-Level Features (final layers before attention module): extract semantic information, enabling the model to understand defect-specific patterns, distinguishing nuanced defect types, and generalizing across diverse defect representations.

To represent the structure of EfficientNet-B0, we use the short form 'Conv, MBConv1, MBConv6', where Conv is the first convolutional layer. The upper and lower layers of our EfficientNet were trained with the RIAWELC dataset and GC10-DET dataset, respectively [31–34]. MBConv1 and MBConv6 are the variable kernel sizes and variable blocks of the convolutional layer, AvgPool is the average pooling layer, and fully connect is the fully connected linear layer, as shown in Figure 2.

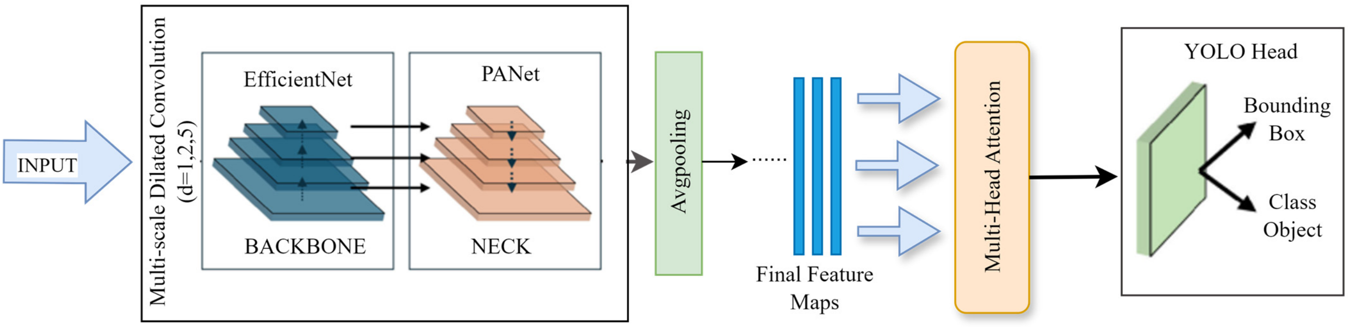


Figure 2. Overview of the proposed architecture combining EfficientNet as backbone, PANet for feature aggregation, and transformer-style multi-head self-attention for enhanced defect localization. The AvgPooling layer prepares flattened tokens for attention, improving model focus on small, occluded, or boundary defects. Final predictions are produced by a YOLOv8-style decoupled detection head.

The multi-head attention mechanism, situated after feature extraction, significantly enhances the interpretability and discrimination capability of the model. Unlike conventional convolutional methods that primarily operate locally, multi-head attention explicitly calculates global dependencies between spatial regions, effectively highlighting specific image areas contributing most significantly to defect classification. The multi-head attention layer is located between the EfficientNet-B0 pooling and the Softmax layer. Adding multi-head attention is to drag the model into high-risk areas and increase sensitivity of detection that traditional techniques might run undetected. This attention mechanism improves the model's accuracy by enhancing focus on regions most likely to contain defects, thereby reducing false negatives in challenging inspection scenarios. The use of multiple attention heads allows the model to focus on both fine-grained details (e.g., micro-cracks) and larger-scale patterns (e.g., cracks and porosity), providing a comprehensive understanding of the defect landscape. Weld defects, such as cracks or porosity, can occur in irregular and unpredictable patterns; therefore, multi-head attention effectively models such spatial relationships, ensuring that critical regions receive greater emphasis. Each attention head independently computes a refined representation of the feature map, focusing on a specific subset of the feature space. Softmax function learnable weights are used to project input features into lower-dimensional query, key, and value spaces, while optimizing the attention mechanism for computational efficiency. The parallel processing of multiple attention heads improves the robustness of the model, particularly in scenarios where defects exhibit high variability in size, shape, or texture. The refined feature map generated by the multi-head attention mechanism is seamlessly integrated with the input of YOLOv8, creating a unified framework for ultrasonic weld defect detection.

The model utilizes YOLOv8 as its foundational architecture due to its efficient single-pass detection capabilities, which allow real-time operation essential for high-speed LIB production environments. A pivotal aspect of our methodology is the incorporation of the multi-head attention mechanism to enhance feature discrimination and defect localization. YOLOv8 provided simpler, well-documented, and modular architecture that facilitated straightforward integration with attention modules, thus ensuring rapid and error-free model customization. Its architecture processes the entire image in a single forward pass, balancing accuracy and speed. Each image is divided into an $S \times S$ grid, and for each cell, bounding boxes are predicted alongside confidence scores and class probabilities, enabling real-time analysis. Vanilla Yolov8 incorporates with the CSPDarknet53 [35], as the backbone network, but we used EfficientNet as the backbone network with Mish activation function as shown in Figure 2. Based on this configuration, we adopted the path aggregation network (PANet) into the neck module of YOLOv8. To ensure optimal feature aggregation and activation dynamics, we evaluated three neck-activation configurations: FPN + ReLU (baseline), BiFPN + SiLU, and PANet + Mish. Our final design having PANet with Mish demonstrated superior multi-scale fusion and smoother gradient propagation, yielding the best balance between accuracy (78.4% mAP, 79.4% F1, 79.8% recall) and inference speed (57.5 FPS), making it the most effective choice for our weld defect detection pipeline.

In this configuration, a universal network structure is built, which is coordinated between top-down and bottom-up modules, and shallow location information and deep semantic information are combined through feature fusion to increase the feature breadth and depth. A 'decoupled header' is used in the main structure of YOLOv8, and distributed focus loss (DFL) [36] is used for bounding box regression and object classification prediction. The single-stage architecture, which does not employ the RPN, allows YOLO to achieve a faster inference with simple architecture compared to the two-stage process, making it suitable for applications requiring real-time or near-real-time performance in objects. This YOLO variant optimizes the loss function by using the VFL loss (vertical federated learning loss) [37] for classification and the CIoU (Complete Intersection over Union) loss [38] and the DFL loss for regression, both of which have specific characteristics. The YOLOv8 model is further enhanced by a tailored transfer learning strategy, utilizing pre-trained weights from welding defect datasets to enhance detection accuracy for LIB-specific defects. By pre-training on this domain-relevant dataset, the model acquires foundational features that are then fine-tuned on LIB-specific data, reducing training time and increasing detection robustness.

3.2. Dataset Preparation and Processing

In constructing a robust defect detection model, a well-curated and representative dataset is essential. Our industrial dataset was carefully assembled to capture the range of defects caused during ultrasonic welding, typically encountered in LIB production, while ensuring the model's ability to generalize across real-world conditions. The dataset includes both real-world defect images and synthetic images generated through a Generative Adversarial Network (GAN), which extends the dataset's diversity and enhances the model's defect recognition capabilities. The dataset exhibits moderate imbalance, with shear and clean classes being more frequent than fold and crease. To address this, we applied class-balanced augmentation and GAN-based synthesis for underrepresented classes, ensuring more uniform training distributions as detailed in Table 1. To maximize model robustness and ensure adaptability across varying real-world conditions, extensive data preprocessing and augmentation steps were applied to each image in the dataset. We also employed the RIAWELC dataset and GC10-DET dataset in the transfer learning process, which allow the model to generalize across a range of defect types and shapes. The preprocessing pipeline consisted of resizing, normalization, and augmentation transformations tailored to the nature of LIB welding defect detection, which is discussed in subsequent sections. For the industrial LIB dataset, we adopted an 80/20 stratified train–test split. To further confirm the robustness of our results, a 5-fold cross-validation protocol was applied during ablation and hyperparameter tuning. Public datasets RIAWELC and GC10-DET followed their respective official splits.

Table 1. Summary of datasets used in this study, including native resolution, number of images per class, and splits.

Dataset	Images	Resolution	Classes Used Here	Notes and Split
RIAWELC	24,407	224 × 224	LP, Porosity, Crack, No-Defect	Used for pre-training/transfer; weld radiography.
GC10-DET	2300	2048 × 1000	10 total (we use punching hole, weld line, inclusion, waist folding)	Industrial steel defects; class imbalance.
Industrial LIB (ours)	6000 + (after aug.)	high-res (var.)	Shear (2000), Porosity (1500), Crease (1200), Fold (800), Crack (500), OK type (5000)	Real + GAN synthesis; 80/20

To address class imbalance and expand the representation of rare defect categories, we employed a Deep Convolutional GAN (DCGAN). The generator comprised four transposed convolutional layers with batch normalization and LeakyReLU activations, while the discriminator was structured symmetrically with convolutional layers and dropout regularization. Training was conducted for 200 epochs using the Adam optimizer (learning rate = 2×10^{-4}, $\beta_1 = 0.5$, batch size = 64). Convergence was monitored through loss stabilization and qualitative inspection of generated images. Synthetic images were generated using DCGAN and integrated only into the training set.

3.2.1. Data Collection Methods and Labeling Standards

The industrial dataset was systematically collected from an operational industrial setting during the actual ultrasonic welding process for lithium-ion battery (LIB) production. High-resolution images were captured using industrial-grade vision systems strategically positioned along the automated manufacturing lines. The imaging devices maintained consistent lighting conditions and fixed positions relative to the welding apparatus, ensuring uniformity and minimizing variability unrelated to weld defects. Captured images underwent immediate preliminary quality checks to filter out unusable or unclear captures, thus ensuring that all dataset images were consistently high clarity and relevant to defect detection tasks. Domain experts from the production team, who were thoroughly trained in recognizing various ultrasonic weld defects, conducted the labeling process. The labeling strictly adhered to clearly defined guidelines that were specifically established for this research. The defect labeling standards were based on internationally recognized weld defect criteria, modified slightly to accommodate LIB-specific features. Each expert annotator was required to do the following:

Identifying defect type: clearly classify each defect into predefined categories (cracks, porosity, shear, fold, crease).

Precise bounding box labeling: accurately mark defect boundaries using bounding boxes to ensure consistency in training YOLO-based detection models.

Cross-validation: Implement double-blind labeling, where two independent experts annotate each image. Discrepancies were resolved through consensus discussions facilitated by a senior quality assurance engineer.

Verification and Validation: 15% of the labeled dataset underwent rigorous random audits by an independent senior engineer to verify annotation accuracy and consistency, achieving an agreement rate exceeding 95%.

Additionally, synthetic data augmentation was performed using Generative Adversarial Networks (GANs) to simulate rare or hard-to-capture defects. The GAN-generated images were carefully validated by domain experts to ensure realism and relevance to actual defect scenarios, thereby enhancing dataset diversity and robustness. To verify the realism of synthetic images, we computed Fréchet Inception Distance (FID) and Inception Score (IS) [39,40], which were computed using the official PyTorch implementation from the TorchMetrics library. All synthetic images generated by the GAN augmentation pipeline were resized to match the input expectations of the Inception v3 network [41], which was pre-trained on ImageNet-1k [42].

For FID computation, we extracted 2048-dimensional features from the pool3 layer of Inception v3 and calculated the distance between real and synthetic distributions using 64-bit double precision for stable covariance estimation. A sample size of 2000 synthetic images and an equal number of real images from the training split were used. For IS computation, the same 2000 synthetic images were evaluated. Each image was passed through Inception v3, and the Softmax class probabilities were used to compute the KL divergence between conditional and marginal label distributions, averaged over 10 splits. The synthetic samples achieved an FID of 14.72 and an IS of 2.85, indicating close alignment with real defect distributions. These evaluations, together with visual confirmation, demonstrate that the augmented set enhances diversity without introducing significant domain shift.

3.2.2. RIAWELC Dataset

This is a radiographic image dataset for weld defects classification. The RIAWELC dataset [31] collects 24,407 224 × 224 8-bit radiographic images digitalized in the.png format with four classes of weld defects represented as lack of penetration (LP), porosity (PO), cracks (CRs) and no defect (ND) as shown in Figure 3. It is used for initial training to familiarize the model with defect types such as cracks and porosity. While it does not contain ultrasonic-specific data, the common type in defects of cracks and porosity have resemblance. Moreover, the dataset's diverse defect types are essential for pre-training for the model's general structural anomaly recognition.

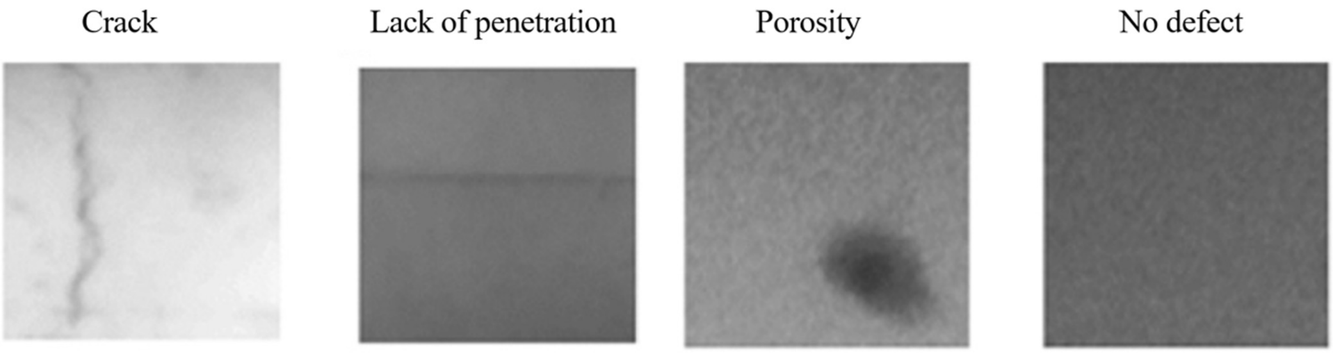


Figure 3. Representative samples from the RIAWELC dataset illustrating four defect categories: lack of penetration (LP), porosity (PO), cracks (CRs), and no defect (ND). These examples highlight the diversity of visual patterns used for pre-training.

3.2.3. GC10-DET Dataset

The GC10-DET dataset [33] was collected under actual industrial settings for extensive metal surface defect identification. It includes a total of 2300 images with a resolution of 2048 × 1000 pixels. The dataset includes ten types of defects found on the surface of steel plates, in which we choose punching hole, weld line, inclusion, and waist folding for transfer learning, which align with surface defects commonly encountered in ultrasonic welding for the TABs of cell. Figure 4 displays some defect sample images with annotations. With strong inter-class similarity and unbalanced sample distribution, the GC10-DET dataset shows a substantial variance in the number of images for each type of defect. Also, there could be multiple defect types in the same image, posing a challenge to defect detection algorithms due to the unbalanced data distribution. Together, the RIAWELC dataset and GC10-DET dataset allow the model to generalize across a range of defect types and shapes.

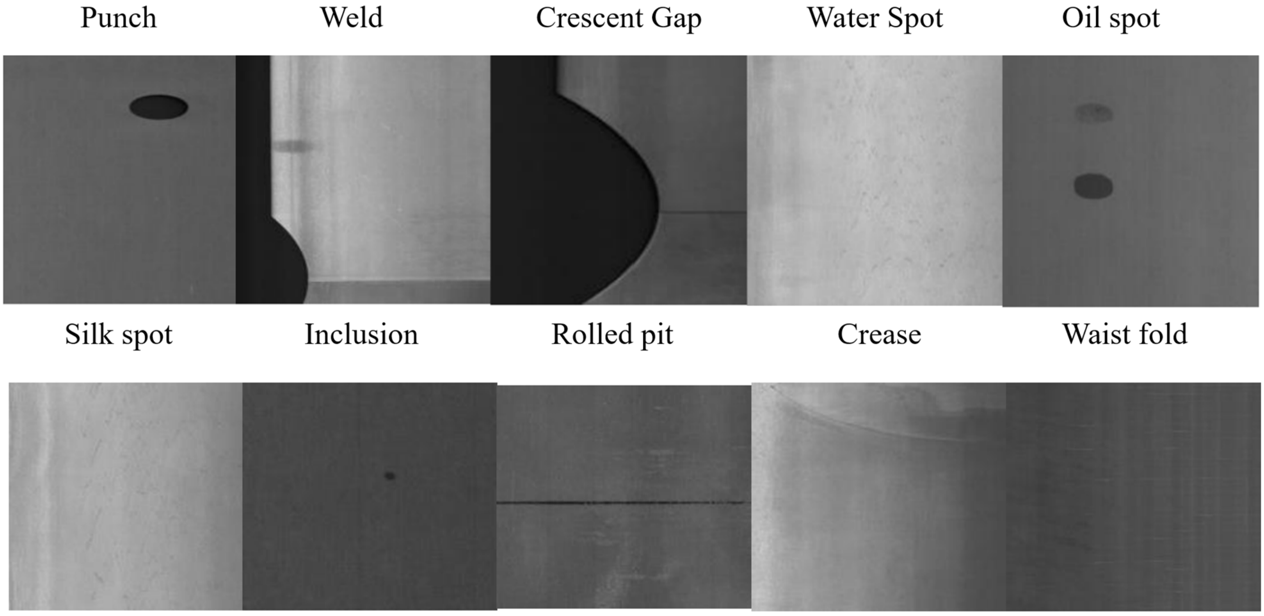


Figure 4. Example defect images from the GC10-DET dataset demonstrate the strong inter-class similarity.

3.2.4. Industrial Dataset for Industrial LIB Weld Images

This collection of industrial data consists of 1500 high-resolution images (native resolution ≈ 2048 × 1000 pixels), acquired using production-line inspection cameras directly from LIB manufacturing environments as shown in Figure 5. These images cover defect types inherent to ultrasonic welding for TAB and busbar joints, including cracks, fold, porosity, crease, and shear. Each image is meticulously annotated by domain experts of production line to accurately label defect locations and types, ensuring high-quality labels for supervised learning.

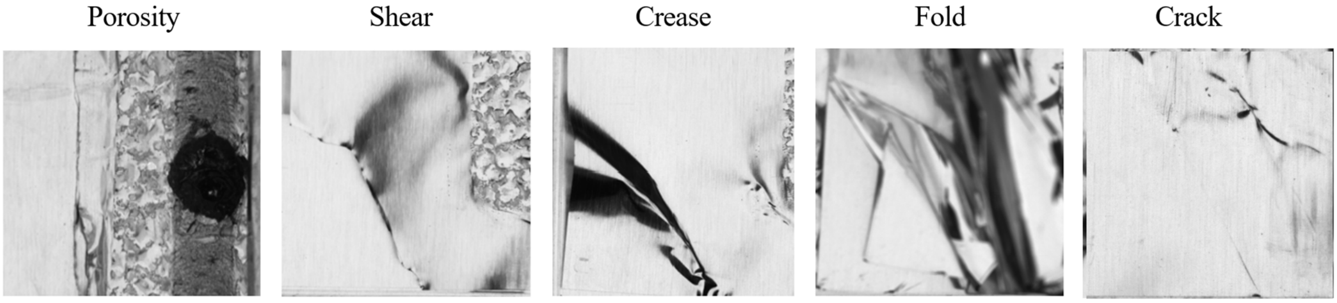


Figure 5. High-resolution images of ultrasonic weld defects from the industrial LIB dataset, covering cracks, folds, porosity, creases, and shear defects.

To address class imbalance, we applied a GAN-based augmentation pipeline. The generator $G(z;\theta_g)$ receives a latent input vector z $\sim \mathcal{N}(0,1)$ and produces a candidate weld-defect image $\hat{x}$. The discriminator $D(z;\theta_d)$ outputs a probability distribution estimating whether input x is real or synthetic. Training follows the minimax formulation:

$$\min_G \max_D V(D,G) = \mathbb{E}_{\mathrm{x}_{p_{data}(x)}}[\log D(x)]+\mathbb{E}_{\mathcal{Z}_{p_z(z)}}[\log(1-D(G(z)))] \tag{1}$$

The loss stabilizes adversarial updates by alternating between gradient steps on θ_d and θ_g. For normalization, all samples are resized to 224 × 224, pixel values are scaled to [0, 1], and standardized to zero mean and unit variance per channel:

$$x' = \frac{x - \mu}{\sigma} \tag{2}$$

Hyperparameters include a batch size of 64, Adam optimizer with learning rate 2×10^{-42}, $\beta_1 = 0.5$, $\beta_2 = 0.999$, and 200 training epochs. The generator uses transposed convolutions with ReLU activations except at the output (Tanh), while the discriminator applies strided convolutions with LeakyReLU ($\alpha = 0.2$). Dropout (0.3) and spectral normalization were added to improve convergence stability. Augmentation transformations (rotation $\pm$ 15°, scaling 0.9–1.1×, horizontal flip, contrast $\pm$ 20%) were applied to both synthetic and real samples. Synthetic data expanded the industrial dataset from 3500 to ~6000 images, balancing defect categories: shear (2000), porosity (1500), crease (1200), fold (800), crack (500), and clean (5000). The dataset was split 80/20 with stratification to preserve per-class balance.

3.3. Feature Extraction Using EfficientNet

EfficientNet is used as a feature extraction network in the classification experiment. EfficientNet uses a composite scaling mechanism to maintain a balance between resolution, depth, and width, making the extracted features rich and computationally efficient. The EfficientNet series includes 7 CNNs and is labeled as EfficientNet-B0 to EfficientNet-B7. In this study, EfficientNet-B0 is used for feature extraction, which provides a balanced level of computational efficiency and accuracy. We select EfficientNet-B0 because its compound scaling (width, depth, resolution) and MBConv blocks with Squeeze-and-Excitation provide high-quality features at low parameter cost. In our defect images, B0 offered the best trade-off between small-object sensitivity and stability during transfer learning, while avoiding the heavier footprint of larger backbones. This choice keeps the detector compact without sacrificing the hierarchical detail needed for micro-crack, porosity, fold, and crease discrimination. By inserting a multi-head attention layer between the pooling layer of EfficientNet-B0 and the Softmax layer, EfficientNet-B0 can outperform a number of feature extractors with fewer parameters at the same input resolution [42–45]. The high scalability of EfficientNet-B0 can effectively extract meaningful features in ultrasonic weld images whose complex structure contains defects. The specific structure of EfficientNet-B0 is shown in Figure 6. It can be divided into seven blocks according to channel range, pass speed, and filter size. To capture scale variation, we aggregate parallel dilated convolutions (e.g., dilation {1, 2, 5}) and sum the activations. This constructs receptive fields that function like a fractional-order spatial operator, broadening context while retaining local sensitivity. The result is a scale-spanning representation that enriches fine weld textures and larger geometric irregularities without materially increasing parameters.

The initial input to the model is a high-resolution ultrasonic weld image I $\in R^{\mathrm{H}\times\mathrm{W}\times\mathrm{C}}$ I, where H, W, and C denote the image's height, width, and channel count, respectively. These images are often captured under constant lighting and environmental conditions of the quick moving conveyer belt within the production line. To standardize the input, the images are resized to a fixed resolution of 224 × 224 pixels and normalized to a [0, 1] range. Resizing ensures compatibility with EfficientNet's pre-trained weights, while normalization reduces the influence of intensity variations. The process is expressed as follows:

$$I' = Resize(I, (224, 224)), I' = \frac{I' - \mu}{\sigma} \tag{3}$$

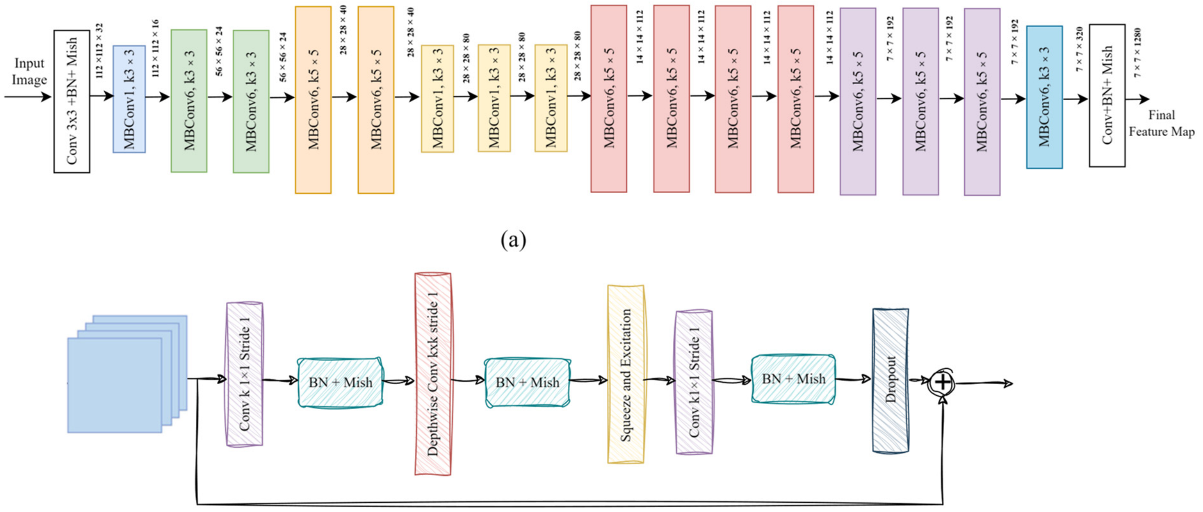
where μ and σ are the mean and standard deviation of the pixel intensities. Based on MobileNet [46,47], the mobile inverted bottleneck (MBConv) is a key component of EfficientNet-B0. EfficientNet processes the input images through a series of convolutional layers, capturing hierarchical features ranging from low-level edges and textures to high-level defect patterns. Each convolutional block produces a feature map F_l as follows:

$$F_l = \sigma(W_l * F_{l-1} + b_l) \tag{4}$$

where W_l and b_l represent the weights and biases of the $l - th$ layer, "*" denotes convolution, and σ is the activation function. As shown in Figure 6, MBConv consists of two k1 × 1 convolutional layers, a depth convolutional layer, a Squeeze-and-Excitation (SE) [48,49] module block, and a dropout layer. To improve the quality of the features, an SE module is added in each convolutional block. It changes the size of the feature maps and distorts the channels that are favorable for error detection. Channel expansion is performed over the first k1 × 1 convolution layer. Deep convolution reduces the number of parameters. By using SE blocks, one can focus specifically on the relationships between the channels and assign variable weights to the channels instead of computing them uniformly. Channel compression is completed via a k1 × 1 s convolution layer. The recalibration is achieved through global average pooling, followed by a non-linear transformation:

$$F = F_l \cdot Sigmoid(W_s \cdot GlobalAvgPool(F_l) + b_s) \tag{5}$$

where W_s and b_s are learnable parameters. This process ensures that defect-related features are amplified, improving downstream detection accuracy. The final feature map F serves as a rich representation of the input image, encapsulating spatial and semantic details critical for identifying weld defects. These features are passed to Neck followed by avgpooling before being fed into the multi-head attention module for further processing.



(a)

(b)

Figure 6. Illustrates in detail EfficientNet-B0 structure. (**a**) The proposed building block of EfficientNet-B0 shows the mobile inverted bottleneck convolution (MBConv) with seven different blocks represented with different colors and (**b**) the proposed internal structure of MBConv.

3.4. Multi-Head Attention

The multi-head attention mechanism is a powerful extension of the self-attention mechanism, designed to simultaneously focus on different aspects of the input feature map. In the context of ultrasonic welding defects detection, this mechanism is essential for capturing the diverse spatial and contextual relationships that characterize various types of defects. In this work, we use a multi-headed self-attention mechanism that handles the attention of scale dot products [50]. In EfficientNet-B0, a multi-headed self-attention layer is inserted between the pooling and Softmax layers. The multi-headed self-attention mechanism allows the network to focus on important information in the image, so that the network has many representation subspaces. The self-attention mechanism is able to evaluate the different influences of the respective pixel positions and assign them corresponding weights for classification. Thus, it is possible to evaluate the relationship of a region to the surrounding area and determine its respective influence in many regions based on the correlation. In similar cases such as defect detection in ultrasonic welding, the influence on the environment often depends on the relationship to the surrounding. Use the L×N matrix Y to represent a set of L to N dimensional objects. Y is the output of the pooling layer, and the corresponding row Y is a separate object vector, as shown in Figure 2.

The feature map $F \in R^{\mathrm{H}\times\mathrm{W}\times\mathrm{C}}$ as a result of multi-head self-attention, the input provided, undergoes a transformation into three distinct matrices as query Q, key K, and value V:

$$Q = FW_q, K = FW_k, V = FW_v \tag{6}$$

where W_q, W_k, and W_v are learnable projection matrices that enable the model to focus on specific feature subspaces. Such vectors can actually serve as an abstraction for the calculation of attention. The similarities of the query and key vectors used to compute the attention weights are calculated from the scaled dot product in the following equation:

$$A_{ij} = softmax\left(\frac{Q \cdot K^T}{\sqrt{d_k}}\right) \tag{7}$$

where Q and K represent query and key matrices derived from the feature map F, and d_k is the dimension of the keys. The Softmax function ensures that the attention weights are normalized across all areas of the image, highlighting regions of high relevance. By applying these computed attention weights to the value matrix V, the model recalibrates its learned feature maps, effectively emphasizing crucial defect-specific regions (e.g., edges of cracks, porosity clusters) and suppressing irrelevant or noisy features (e.g., non-defective background textures). The attention weights A are applied to the value matrix V to produce a refined attention-weighted feature map F'. The attention-weighted feature map F' is then given by

$$F' = A \cdot V \tag{8}$$

Consequently, the recalibrated feature map F' is enriched with explicit defect-location and defect-type-specific contextual information, thereby directly enhancing interpretability and classification performance. V allows the model to attend to critical defect features while minimizing attention on irrelevant areas and amplifying features associated with defects while suppressing irrelevant noise, such as background textures or other variations in the terminal interface.

Self-attention enables the model to dynamically adapt its focus based on the unique characteristics of each image. For example, it can prioritize features associated with a crack's initiation point while also capturing the progression of the crack across the weld. For the case of multi-head self-attention, it linearly processes Q, K, and V multiple times via different weight matrices and processes the input features through multiple parallel attention heads, each focusing on a different aspect of the feature space as shown in

Figure 7. Initially, for parallel attention computations, the input feature map F is split into h subspaces, with each attention head independently computing a refined representation:

$$F^{i}_{attention-head} = softmax(\frac{Q^{(i)} \cdot K^{(i)T}}{\sqrt{d_k}}) \cdot V^{(i)} \tag{9}$$

where Q^i,K^i, and V^i are the query, key, and value matrices specific to the *i-th* head. This allows the model to observe the features from different attention heads, each learning different aspects of the image. The output from each attention head is concatenated and linearly transformed to generate the final attention-enhanced feature map $F_{\text{multi-head}}$:

$$F_{multi-head} = Concat\left(F^{1}_{attention-head}, F^{2}_{attention-head}, F^{h}_{attention-head}\right) W_o \tag{10}$$

where W_o is a learnable weight matrix. Multi-head attention provides a comprehensive representation of the input features, capturing both local details (e.g., micro-cracks) and global context (e.g., uneven weld lines). We propose a network with an attention layer that has 512 pixels at its output size and h = 3. This integration ensures that the contextual enhancements provided by attention are effectively utilized in the final detection for the YoloV8 model. Multi-head attention in our proposed model employs multiple parallel attention heads, each independently focusing on distinct spatial or semantic aspects of weld defects. This explicit differentiation is crucial for interpreting the model's ability to discriminate defect types, as detailed below:

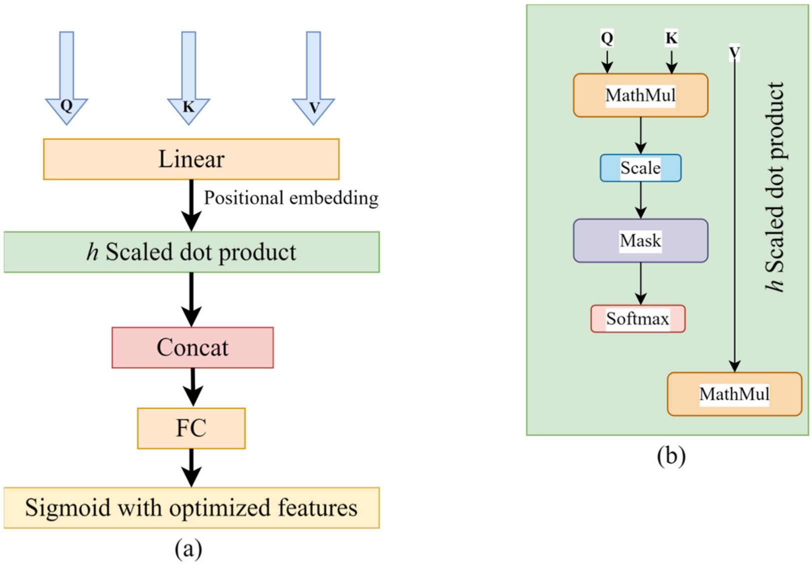


Figure 7. Multi-head self-attention integration. (**a**) EfficientNet-B0 feature map is projected into query, key, and value tensors. (**b**) Multiple heads attend to complementary spatial/semantic pattern, where head outputs are concatenated and linearly fused to form an attention-enhanced feature map, which is then fed to the detection head.

Fine-Grained Attention Heads: Certain attention heads specifically target small-scale defects (e.g., micro-cracks and pores), capturing localized and subtle structural discontinuities. These fine-grained attention heads effectively distinguish defects that are visually minimal yet structurally critical.

Contextual Attention Heads: Other attention heads explicitly identify larger-scale spatial patterns indicative of defects like shear and crease. By focusing on extended spatial correlations and irregularities in larger regions, these heads provide interpretability into the model's decision process regarding extensive, structurally significant defects.

Cross-Scale Attention Integration: Attention heads dynamically integrate features across different scales, thereby comprehensively capturing defects exhibiting variable size or texture patterns. This multi-scale interpretability ensures robust performance even in highly varied industrial conditions.

To empirically support our selection of three attention heads with a 512-dimensional embedding, we conducted an ablation study across different configurations. Using a single attention head with a 256-dimensional output resulted in an mAP of 76.0%, precision of 75.1%, and recall of 76.8%, while maintaining high throughput at 62.5 FPS on an RTX 4090. Increasing the number of heads to three and the embedding dimension to 512 boosted the mAP to 78.4%, precision to 77.0%, and recall to 79.2%, with only a moderate drop in FPS to 57.5 and a manageable parameter count of 30.1 million. Further increasing to six heads and 768 dimensions yielded only marginal gains (mAP: 78.9%, recall: 79.5%), while significantly degrading runtime speed to 52.1 FPS and increasing parameter count to 33.7 million. FLOPs grew from 92.4 GF (1-head) to 98.6 GF (3-head) and 108.9 GF (6-head), respectively. This trade-off analysis confirms that the 3-head, 512-dim configuration offers the optimal balance between accuracy and runtime performance, justifying its adoption in our final model.

3.5. Detection Head Using YOLOv8

Following feature extraction with the multi-head mechanism, YOLOv8 is employed as the detection head to locate and classify defects. YOLOv8 is a one-stage object detection framework that achieves a balance between speed and accuracy, making it particularly suited for high-throughput manufacturing environments. To ensure quality and safety, defects such as cracks, porosity, folds, shears, and creases must be detected in real time. During the manufacturing pipeline of vehicle LIBs production, a preliminary experimental evaluation was performed which indicated that YOLOv8, with our EfficientNet-based feature extraction backbone and multi-head attention integration, offered superior robustness and accuracy, specifically on our defect detection datasets (private LIB ultrasonic welding dataset, GC10-DET, and RIAWELC). Although YOLOv10 showed promising results in general object detection benchmarks, its incremental accuracy improvements were marginal (less than 1–2% increase in mAP), specifically for subtle defect classes such as crease and fold. Thus, YOLOv8 provided comparable practical accuracy without additional computational overhead as it offers excellent inference speed (~65 FPS on an RTX 4090 GPU) and reliable detection accuracy, making it a strong industrial baseline for high-speed LIB manufacturing lines. However, our deployment prioritizes real-time CPU-based inference. In this environment, our model achieves 45 FPS on a standard Intel i5 CPU with 512 MB RAM, significantly outperforming YOLOv8's CPU performance (≈22 FPS from Ultralytics benchmarks). Conversely, although YOLOv10 can achieve modest GPU speed gains (5–8%) over YOLOv8 on certain tasks, it increases model complexity (2.3 M vs. 1.7 M parameters, 6.7 G vs. 3.2 G FLOPs). For inline industrial use, our model's CPU-level performance and reduced resource footprint offer decisive practical benefits.

The feature map $F_{multi\text{-}head}$ from the EfficientNet backbone with attention is fed into YOLOv8's detection module. This module refines the features and predicts bounding boxes, confidence scores, and class probabilities for each defect. Later, YOLOv8 divides the input feature map into a grid of $S \times S$ cells. Each cell predicts bounding boxes for objects potentially located within its region, along with associated confidence scores and class probabilities. The output for a single grid cell is represented as

$$O_{ij} = \{(x, y, w, h, c, p_1, p_2, \ldots, p_k)\} \tag{11}$$

where (x, y) are the normalized center coordinates of the bounding box. w, h are the width and height of the bounding box. c is the confidence score for the presence of a defect. p_k is the probability of the defect belonging to class k. Unlike earlier YOLO models, YOLOv8 adopts an anchor-free approach, simplifying the architecture and improving inference speed. Instead of predefined anchor boxes, it predicts box centers and offsets directly:

$$\Delta x, \Delta y, \Delta w, \Delta h = f(F) \tag{12}$$

where f represents the prediction function. YOLOv8's training process minimizes a combined loss function, incorporating the following:

- Complete Intersection over Union (CIoU) loss for precise bounding box regression.
- Objectness loss (confidence score) based on Binary Cross-Entropy to distinguish between defect and background effectively.
- Classification loss based on Binary Cross-Entropy with logits for accurate defect classification.

Formally, the combined loss function is represented as

$$\mathcal{L}_{YOLO} = \lambda_{loc}\mathcal{L}_{CIoU} + \lambda_{conf}\mathcal{L}_{conf} + \lambda_{cls}\mathcal{L}_{cls} \tag{13}$$

where $\lambda_{cls} = 1$, $\lambda_{conf} = 1$ and the localization loss $\mathcal{L}_{loc}$ measures the accuracy of the bounding box predictions using the CIoU (Complete Intersection over Union) metric:

$$\mathcal{L}_{loc} = 1 - CIoU\left(B, \hat{B}\right) \tag{14}$$

where $B, \hat{B}$ are the predicted and ground truth bounding boxes. YOLOv8 generates predictions at multiple scales to handle defects of varying sizes, from small crease patterns to large cracks. This multi-scale capability ensures comprehensive detection across all defect types. Lastly, the optimized YOLOv8 model with transfer learning undergoes further training with a combination of real and augmented defect images. Stochastic gradient descent (SGD) with momentum is used, hence improving training convergence and model stability, which is explained in a further section. Momentum, m, helps in accelerating updates and reducing oscillations in the gradient descent as follows:

$$\Delta \mathrm{w}_t = m \cdot \Delta \mathrm{w}_{t-1} - \eta \Delta \mathcal{L}_{YOLO+TL} \tag{15}$$

where Δw_t represents the weight update at iteration t.

4. Results and Experimentations

We thoroughly evaluate the proposed model for defects in lithium-ion battery's TABs due to ultrasonic welds, while conducting a series of experimental techniques. The aim is to verify the accuracy of the model in terms of detected defects, localization, and efficiency in a real LIB production environment. This section deals with the general experimental setup, the generation of the dataset, transfer learning, evaluation, and critical analysis of the results.

4.1. Evaluation Metrics

The effectiveness of the proposed model was evaluated using a set of key metrics widely adopted in defect detection tasks:

Precision (P): The proportion of correctly identified defects out of all predicted defects, offering a measure of the model's accuracy in defect detection. Precision was calculated as

$$\text{Precision} = \frac{\text{True positives}}{\text{True positives} + \text{False positives}} \tag{16}$$

where *TP* represents true positives and *FP* denotes false positives.

Recall (R): The proportion of actual defects that were detected by the model, indicating the model's coverage in identifying defects. Recall was calculated as

$$\text{Recall} = \frac{\text{True positives}}{\text{True positives} + \text{False negatives}} \tag{17}$$

where *FN* represents false negatives.

Average Precision (AP): in AP, the precision of the model is evaluated for a given detection category by measuring the area under the precision–recall curve.

$$AP = \frac{Precision + Recall}{2} \tag{18}$$

Mean Average Precision (mAP): in mAP is the averaged AP score across all categories, and it provides a measurement of the overall detection accuracy, for N class.

$$mAP = \frac{1}{N}\sum\nolimits_{i=1}^{N} AP_i \tag{19}$$

4.2. Experimentations

The results section evaluates the performance of the proposed model, focusing on its effectiveness in detecting the defects that occurred during ultrasonic welds within lithium-ion battery (LIB) manufacturing. Performance metrics including precision, recall, average precision (AP), and mean average precision (mAP) were assessed on an industrial dataset containing real and synthetic defect images. An ablation study confirmed the critical contributions of the attention mechanism and transfer learning, with multi-head attention enhancing defect localization and reducing false positives, and transfer learning enabling faster convergence and improved generalization. Comparative baseline testing further highlighted our model's superiority over conventional CNNs, YOLOv5 and vanilla YOLOv8, and other state-of-the-art methods, while our model proved to be significantly better in performance across all metrics, underscoring its suitability for automated defect detection for the LIB's TAB ultrasonic welding defects.

4.2.1. Implementation Details

All experiments were conducted on a workstation equipped with NVIDIA RTX 4090 GPU (24 GB VRAM) and Intel i9-13900KF CPU, with PyTorch 1.13 and CUDA 11.7. The industrial dataset consisted of high-resolution images (native resolution ≈ 2048 × 1000 pixels), acquired using production line inspection cameras. All images were subsequently resized to 640 × 640 pixels for training and inference and the inference was conducted with a fixed batch size of 16. Additionally, we employed TensorFlow 2.11 for preprocessing and data augmentation. We also standardized inference resolution to 2048 × 1000 and batch size to 16 across all benchmark models unless otherwise noted. The model architecture was implemented in PyTorch 1.9.1 for the primary deep learning operations and used during earlier stages of development, but benchmarking was completed using the compatible stack without introducing modifications to the model structure or hyperparameters. TensorFlow was used for preprocessing and data augmentation. The model's training strategy incorporated transfer learning with weights pre-trained on an ultrasonic weld defect dataset, which were then fine-tuned specifically for LIB weld defects. For the model training, the dataset comprises two primary components; one is an industrial dataset that is based on the industrial images directly captured during the manufacturing process and the second is a publicly available dataset which is further discussed. To optimize the model's performance for defect detection, transfer learning is implemented using state-of-the-art datasets which comprise annotated images of typical defects, such as cracks and porosity, crease, fold, etc., closely with the characteristics found in lithium-ion batteries during TAB with busbar welding.

4.2.2. Pre-Training and Layer Freezing Unfreezing

The lower layers of the backbone network, which capture fundamental features like edges, textures, and shapes, are pre-trained on the RIAWELC dataset and GC10-DET dataset. This pre-training phase allows the network to learn features specifically relevant to the defect weld domain, enhancing its ability to identify delicate defects caused during welding. Formally, for an input image $I \in R^{H \times W \times C}$, where H, W, and C represent the image height, width, and channels, respectively, the pre-trained layers generate a feature map which is $F \in R^{H' \times W' \times C'}$; therefore,

$$F = f\left(I; \theta_{pre-trained}\right) \tag{20}$$

where f denotes the series of convolutional operations in the lower layers, and $\theta_{\text{pre-trained}}$ signifies the weights learned from the RIAWELC dataset and GC10-DET dataset. After pre-training, the upper layers of the features network, which are responsible for high-level features and defect classification, are fine-tuned using our industrial dataset for lithium-ion battery-specific defects. This fine-tuning adapts the model to detect the delicate defects of battery terminal welds, such as specific crack or porosity formations or irregular bonding patterns which may lie under folds or creases. The fine-tuning process optimizes the model's parameters by minimizing the adjusted YOLOv8 loss function, $\mathcal{L}_{\text{YOLO + TL}}$, expressed as

$$\mathcal{L}_{YOLO+TL} = \lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B} 1_{ij}^{obj}\left((x_i-\hat{x}_i)^2+(y_i-\hat{y}_i)^2\right) + \lambda_{conf}\sum_{i=0}^{S^2} 1_i^{obj}(c_i-\hat{c}_i)^2 + \lambda_{class}\sum_{i=0}^{S^2}\sum_{k=1}^{K} 1_i^{obj}\left(p_{ik}-\hat{p}_{ik}\right)^2 \tag{21}$$

where λ_{coord}, λ_{conf}, and λ_{class} are scaling factors for localization, confidence, and classification losses, respectively. 1_{ij}^{obj} is an indicator function set to 1 if an object is present in the cell, $\hat{x}$, $\hat{y}$, $\hat{c}$, and $\hat{p}$ are the ground truth values for the box coordinates, confidence, and class probabilities. To avoid overfitting during the fine-tuning phase, a layer freezing and gradual unfreezing strategy is implemented. Initially, the lower layers responsible for capturing generalizable features are frozen. This stabilizes the network and ensures that only the defect-specific layers are updated during early training epochs. Over time, the lower layers are gradually unfrozen to allow the entire network to adapt to application-specific defect patterns.

4.2.3. Hyperparameter Settings and Optimization Strategies

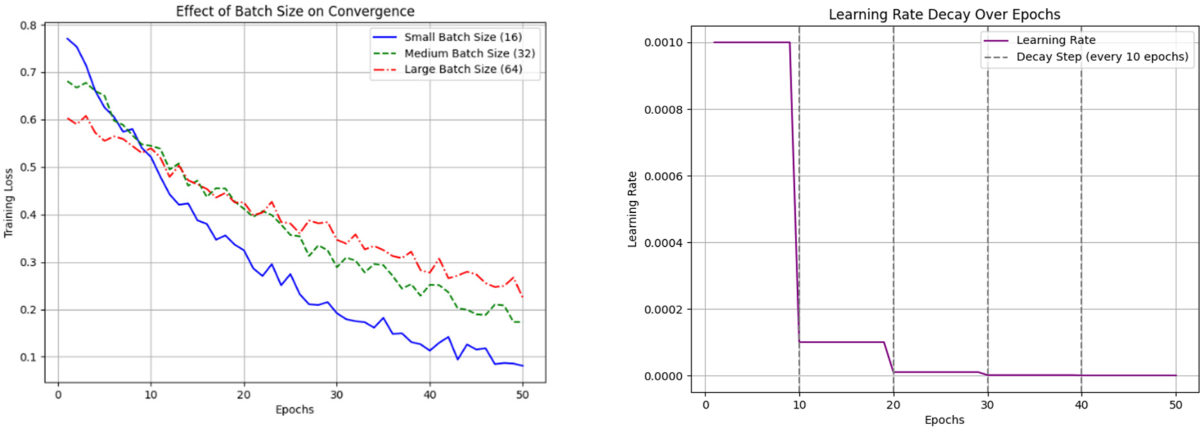
The training procedure employed a carefully structured optimization strategy to ensure stable convergence and high generalization performance. The selection of these parameters is discussed in detail in the following sections, allowing the model to learn more efficiently, avoid underfitting and overfitting, and balance performance with computational resources.

i. Weight Initialization Regularization and Overfitting Mitigation

The lower layers of the EfficientNet backbone network, responsible for general feature extraction such as edges and textures, were pre-trained on the RIAWELC and GC10-DET datasets. The model weights obtained from this pre-training phase served as initial weights, significantly improving training efficiency and reducing convergence time. Subsequently, the upper layers specific to defect detection were fine-tuned using the LIB-specific dataset through supervised learning. Several regularization techniques were incorporated to enhance model generalization and prevent overfitting, including dropout layers (with dropout rate of 0.2 within MBConv blocks of EfficientNet) and early stopping criteria based on validation loss. The early stopping strategy monitored validation loss improvement, ceasing training once no significant decrease occurred for 10 consecutive epochs.

ii. Optimizer and Batch Size

Although both Adam and SGD optimizers were tested, SGD with momentum was selected as it provided more stable convergence and superior generalization compared to Adam, which tended to overfit despite faster initial convergence. Therefore, our model was optimized using Stochastic Gradient Descent (SGD) enhanced by momentum to facilitate smoother and faster convergence. The momentum parameter was set to 0.9, promoting stability in training by reducing oscillations in gradient updates. We chose a batch size of 16 to simultaneously process the training samples before updating the model's internal parameters, thereby striking a balance between memory efficiency and computational speed. Batch sizes that are too small can result in noisy updates; very large batches can result in higher memory requirements and slower convergence due to more frequent parameter changes. Smaller batches allow SGD to generalize better due to noisier gradients but may slow down convergence as the gradient steps may oscillate. Conversely, larger batches can smooth out the gradient but have risk of overfitting if they do not capture enough variation. If we visualize the batch size effect on convergence as shown in Figure 8a, smaller batch sizes typically exhibit more oscillations in the loss function but can explore the error surface more thoroughly, while larger batch sizes tend to converge smoothly but can become stuck in sharp local minima.



(**a**) (**b**)

Figure 8. (**a**) Graph on the left shows the effect of varying batch sizes on training convergence, with smaller batches leading to noisier but potentially more explorative learning. (**b**) Graph on the right shows the decay in learning rate over epochs, where the learning rate decreases by a factor of 0.1 every 10 epochs.

iii. Learning Rate

The learning rate controls the size of the steps taken by the optimizer in the direction of the gradient during training. For our model, an initial learning rate of 0.001 was selected. This learning rate was found to provide a good balance, ensuring that the model made substantial progress in each epoch without diverging due to too-large updates. As training progresses, we apply a decay factor to the learning rate to allow for more refined adjustments as the model approaches a local minimum. As in Figure 8b, the learning rate decay is applied every 10 epochs, reducing the learning rate by a factor of 0.1, which can be mathematically represented as

$$\eta_t = \eta_0 \cdot \gamma^{\left(\frac{t}{T}\right)} \tag{22}$$

where η_t is the learning rate at epoch t, η_0 is the initial learning rate, γ is the decay factor (in this case, 0.1), T is the total number of epochs for each decay interval (10 epochs here). This gradual reduction helps stabilize convergence as the model's parameters become closer to optimal values, preventing the model from "overshooting" the optimal point. As it nears convergence, a smaller learning rate fine-tunes the weights, and minimizes oscillations around the minima, enabling the model to settle. This decay allows for sharper adjustments early on and fine adjustments later, contributing to smoother convergence.

iv. Epoch setting for total training

Setting an appropriate number of epochs is crucial for achieving convergence without overfitting. In this study, 50 epochs were chosen based on preliminary experiments, where we observed that the model achieved stable convergence within this range.

The convergence of the loss function over epochs can be expressed as

$$L(\theta) = \frac{1}{N}\sum\nolimits_{i=1}^{N} \ell(y_i, f(x_i;\theta)) \tag{23}$$

where $L(\theta)$ represents the overall loss as a function of model parameters θ, N is the dataset size, ℓ denotes the individual loss for each training sample (x_i, y_i). During preliminary training, we monitored the validation loss across epochs, noting that the model's loss stabilized after around 50 epochs. In practice, this corresponds to reaching a point where additional epochs do not significantly reduce the validation loss, indicating that the model has captured the necessary patterns without overfitting. The graph shown in Figure 9 illustrates the trend observed in training and validation loss over epochs. Early epochs show rapid decreases in loss as the model learns core patterns. Beyond 30–40 epochs, the rate of improvement slows, with losses stabilizing near 50 epochs, suggesting optimal convergence as can be seen on the figure on the left. In the right figure, the graph demonstrates how the model's training and validation losses converge over time. Initially, both losses decrease as the model learns. Eventually, validation loss stabilizes, indicating that the model generalizes well to unseen data, while training loss continues to decrease, signaling effective learning without overfitting.

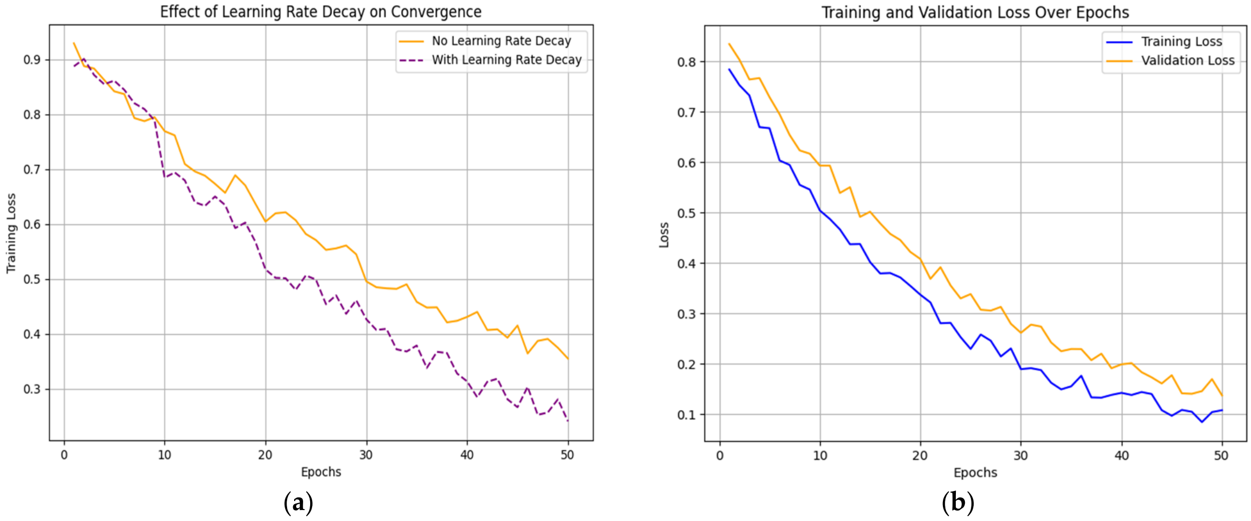


Figure 9. Training dynamics of the proposed model: (**a**) learning rate decay schedule showing reduction every 10 epochs, (**b**) convergence of training and validation losses over 50 epochs, indicating stable generalization and effective convergence.

4.3. Results

The results across the industrial LIB, RIAWELC, and GC10-DET datasets demonstrate three central findings. First, integrating multi-head attention consistently enhances recall by reducing false negatives, particularly for small or occluded defects. Second, adopting PANet with Mish activation yields superior multi-scale fusion, balancing gradient stability with feature refinement across defect sizes. Third, the three-head attention configuration provides the optimal trade-off, delivering the highest F1-score improvements while preserving real-time inference speed. These observations highlight that the proposed architectural choices are not incremental but strategically targeted to address the challenges of subtle localization, scale heterogeneity, and efficiency in LIB weld defect detection.

4.3.1. Performance Evaluations and Comparison Analysis

For distinct defects like porosity and cracks, the model demonstrates strong performance, achieving high precision and average precision (AP) scores of 78.5% and 80.1%, respectively, can be seen in Figure 10.

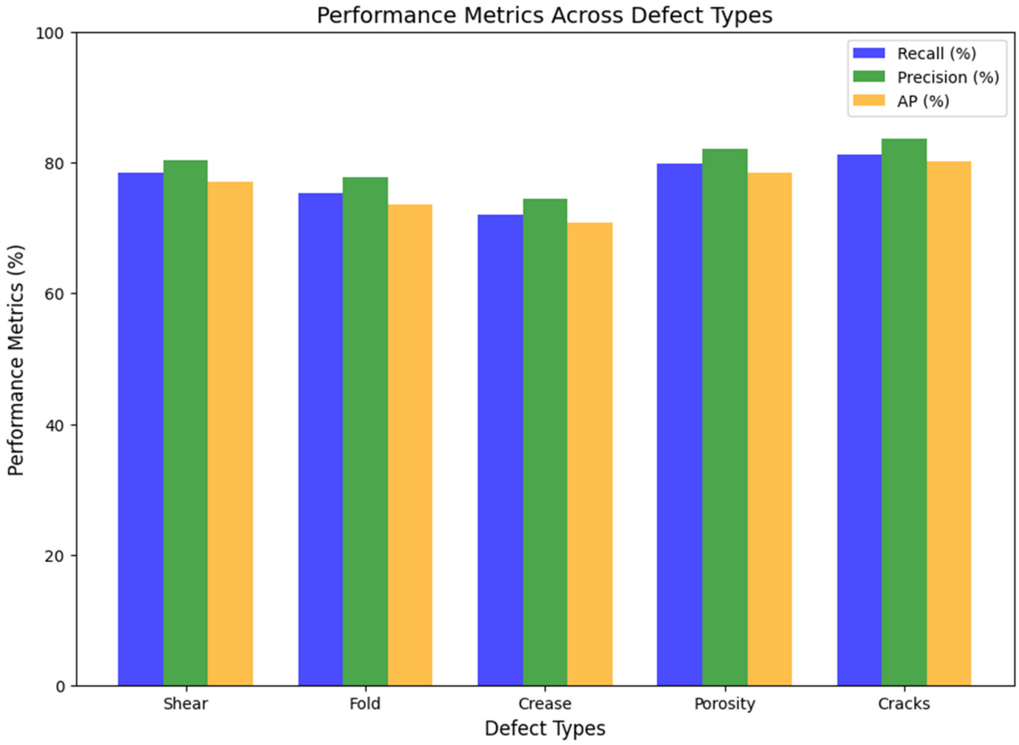


Figure 10. Illustrates that our proposed model delivers robust detection and localization capabilities on the custom LIB dataset, enabling accurate and efficient defect identification for real-time settings.

The better performance detecting porosity and cracks is largely due to the clear visual markers these defects possess, allowing YOLOv8's detection strength and EfficientNet-B0's feature extraction to be fully leveraged. However, detecting delicate defects like fold and crease proved more challenging. These defects yielded lower recall rates, between 72.0% and 75.3%, and AP scores ranging from 70.8% to 73.6%. The shear defect presents a middle ground in detection difficulty since it is almost like cracks in appearance but with shear from either side, with balanced performance metrics of precision at 80.4%, recall at 78.5%, and an AP of 77.1%, highlighting the model's moderate sensitivity to this defect type. Since synthetic data may introduce a domain gap, we conducted an explicit cross-domain validation by training on a mixture of real and GAN-generated samples and testing only on real defect images. Averaged over five independent runs, augmentation improved recall by +3.4% and F1-score by +2.1%, while precision remained virtually unchanged (fluctuations below ±0.2%). These results indicate that synthetic augmentation enhances the model's ability to recognize minority defect classes without compromising overall reliability, reinforcing its suitability for industrial defect detection pipelines.

Table 2 benchmarks our lightweight model (1.7 M parameters, CPU-based, 45 FPS) against top-tier methods applied to the RIAWELC radiographic weld dataset. Notably, Weld-CNN achieved an outstanding 99.83% F1-score using ~5 M parameters. ResNet50-CNN achieved 98.75% accuracy with ~25.6 M parameters; and SqueezeNet baseline delivered ~93% accuracy with just 1.24 M parameters. While our model records a mid-range F1 of 81%, it operates in CPU-only environments and maintains real-time performance, highlighting a compelling efficiency–accuracy compromise ideal for constrained industrial deployment. To gain deeper insights into the nature of our model's prediction errors, we undertook a fractal-inspired analysis focusing on false positives (FPs) and false negatives (FNs). For each defect class, we generated cumulative spatial heatmaps that reveal where FPs and FNs occur most frequently across the test set. Interestingly, these error regions displayed self-similar, clustered patterns that are reminiscent of the fractal geometries seen in actual weld microstructures. To objectively quantify this observation, we calculated the box-counting fractal dimension for both FP and FN masks. The results indicate that the spatial complexity of these error distributions is strikingly close to the scale-invariant properties of true defect regions. This finding suggests that model misclassifications are not simply random events, but instead show systematic, multi-scale structure, a characteristic that can be directly leveraged for further model refinement. Employing such fractal and fractional analysis not only enhances the interpretability of error patterns, but also provides a principled framework to guide targeted improvements in future defect detection systems.

Table 2. Model comparison across RIAWELC dataset.

Model	Author(s)	Backbone	Params (M)	Metric (F1 or Acc %)
Weld-CNN [51]	Hoa et al.	Custom CNN	~5.0	99.83 F1
ResNet50-CNN [52]	Palma-Ramírez et al.	ResNet50	~25.6	98.75 Acc
SqueezeNet V1.1 [53]	Totino et al.	SqueezeNet V1.1	1.24	93 Acc
LF-YOLO [54]	Liu et al.	Lightweight YOLO	~4.3	92.9 mAP50
Defect Transformer [55]	Wang et al.	Hybrid DefT	-	Not reported
Our Proposed	(This work)	Lite custom	1.7	81 F1/78.9 mAP50

As shown in Table 3, our proposed lightweight model achieves a mean average precision (mAP@0.5) of 78.9% on the GC10-DET dataset, significantly outperforming several recent state-of-the-art approaches. Notably, it surpasses the FI2Net two-stage CNN (70.3% mAP) and the FOHR-Net model (70.5% mAP), both of which rely on considerably larger architectures. Enhanced one-stage methods such as the YOLOv5 channel-shuffled, a hybrid Transformer-based SSM, and the PMSE-YOLO (~71% mAP) also fall short. Despite having just 1.7 M parameters and operating at 45 FPS on CPU, our model not only closes the gap to heavier, GPU-dependent architectures but sets a great benchmark for real-time, resource-efficient industrial deployment.

Figure 11 illustrates the proportional distribution of total detection errors encompassing both false positives and false negatives across all major defect classes in our industrial weld dataset. Notably, the majority of errors are concentrated in the shear, crease, and fold categories, each accounting for a significant share of the overall mistakes. This trend is in line with known challenges in ultrasonic weld inspection, where the visual manifestations of shear, crease, and fold defects often overlap or exhibit ambiguous boundaries, increasing the likelihood of both missed detections and spurious predictions. In contrast, crack defects, despite being the least frequent in our dataset, contribute minimally to the total error count, a testament to their relatively distinctive visual features, which the model is able to reliably discern. The higher error rates for porosity and other subtle classes further underscore the value of advanced data augmentation and GAN-based synthesis, as these approaches enhance model robustness to rare or visually complex defect types. Collectively, this graphical analysis provides actionable insights, directly guiding future refinements in both data strategy and model architecture for industrial defect detection tasks.

Table 3. Model comparison on GC10-DET dataset.

Model	**Author(s)**	**Backbone or Notes**	**Params (M)**	**Metric (%)**
FI2Net [56]	Lv et al.	multi-stage CNN	~10	65.1 mAP
FOHR-Net [57]	Chan et al.	Custom CNN	-	70.5 mAP
YOLOv5-CSShuffle [58]	Yasir & Ahn	YOLOv5 (CS-shuffle)	-	70.18 mAP@0.5
Self-Adaptive Gamma SSM [59]	Sun et al.	Transformer-based SSM	-	+2.6\gain mAP@0.5
PMSE-YOLO [60]	Zhou et al.	YOLOv5 variant	-	~55.5 mAP@0.5:0.95
WFRE-YOLOv8s [61]	Yao et al.	YOLO variant	~14	69.4 mAP@0.5
Our Proposed	(This work)	Custom lightweight	1.7	78.9 mAP@0.5

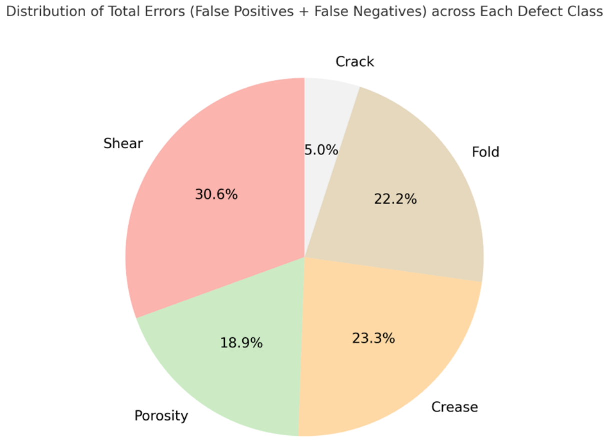


Figure 11. Illustrating the distribution of total errors (FP + FN) across each defect class (shear, porosity, crease, fold, crack).

To further rigorously evaluate the effectiveness of the proposed model, its performance was compared against widely recognized object detection frameworks using the industrial dataset initially, which contains ultrasonic weld images specific to lithium-ion battery manufacturing. Moreover, to show the effectiveness of the model, this subsection also demonstrates the results on a publicly available dataset for the given five detection classes. Table 4 provides a comprehensive comparison between our approach and prominent object detection frameworks across standard metrics mAP, precision, recall, and F1-score on the private ultrasonic weld defect dataset. The proposed model consistently outperforms classical baselines such as SSD, RetinaNet, YOLOv5, and YOLOv8, as well as region-based detectors including Faster R-CNN, Mask R-CNN, and Cascade R-CNN. We also incorporated two more recent state-of-the-art methods, PP-YOLOE-S and RT-DETR (R50), which achieved competitive results, with mAP values of 75.8% and 76.1%, respectively. Nevertheless, our model surpasses all competitors, achieving the highest mAP of 78.4% and superior per-class accuracy, particularly on challenging categories such as fold and crease. The corresponding F1-score of 79.4% further highlights the balanced improvement in both detection sensitivity and specificity. This evidences the suitability of our method for practical, high-throughput manufacturing settings where minimizing defect escapes and false positives is paramount.

Table 4. Performance analysis for the industrial dataset using proposed model.

Model	mAP (%)	Shear (%)	Porosity (%)	Crack (%)	Fold (%)	Crease (%)
SSD	70.92	70.8	73.2	74.5	69	67.1
RetinaNet	70.98	73.5	75.6	76.8	70.3	68.7
YOLOv5	71.2	75	78.9	79.4	71.8	69.9
YOLOv8	72.84	72.9	78.2	79.5	68.5	65.1
Faster R-CNN	73.84	72.7	77.8	79.3	70.5	68.9
Mask R-CNN	73.36	71.9	77.2	78.5	70	68.2
Cascade R-CNN	75.58	74.2	79.5	81	72.5	70.8
DDN	74.7	74	78.5	80.2	71.5	69.5
PP-YOLOE-S	75.80	75.1	79.6	81.1	72.7	70.9
RT-DETR (R50)	76.10	75.4	80.0	81.8	73.0	71.4
Our Model	78.4	78.5	82.1	83.7	75.3	72

Then, comparison experiments were conducted for current advanced network models, on the GC10-DET and RIAWELC datasets in which the five defect classes lie by using the same one- and two-stage detectors, to further confirm the improved model's robustness and generalization ability. The comprehensive comparison results of defect accuracy and mAP for each model on the following two public datasets are shown in Table 5.

Table 5. Comparison analysis of different state-of-the-art models and proposed model for GC10-DET and RIAWELC datasets.

Model	Dataset	mAP (%)	Porosity (%)	Crack (%)	Fold (%)	Crease (%)
SSD	GC10-DET	68.05	-	-	69	67.1
	RIAWELC	73.85	74.5	73.2	-	-
RetinaNet	GC10-DET	69.5	-	-	70.3	68.7
	RIAWELC	76.2	76.8	75.6	-	-
YOLOv5	GC10-DET	70.85	-	-	71.8	69.9
	RIAWELC	79.15	78.9	79.4	-	-
YOLOv8	GC10-DET	72.8	-	-	73.5	72.1
	RIAWELC	82.35	81.2	83.5	-	-
Faster R-CNN	GC10-DET	69.7	-	-	70.5	68.9
	RIAWELC	78.55	77.8	79.3	-	-
Mask R-CNN	GC10-DET	69.1	-	-	70	68.2
	RIAWELC	77.85	77.2	78.5	-	-
Cascade R-CNN	GC10-DET	71.65	-	-	72.5	70.8
	RIAWELC	80.25	79.5	81	-	-

Table 5. *Cont.*

Model	Dataset	mAP (%)	Porosity (%)	Crack (%)	Fold (%)	Crease (%)
DDN	GC10-DET	70.5	-	-	71.5	69.5
	RIAWELC	79.35	78.5	80.2	-	-
Our Model	GC10-DET	73.65	-	-	75.3	72
	RIAWELC	82.9	82.1	83.7	-	-

In our comparison analysis, SSD demonstrated moderate detection capabilities, achieving an mAP of 70.9%, while it performed reasonably well for larger defects like cracks and porosity. For RetinaNet, incorporating focal loss to address class imbalance showed improved accuracy over SSD, achieving an mAP of 72.98%. YOLOv5 delivered higher detection accuracy, achieving an mAP of 74.8%, with strong performance in detecting shear and cracks. Vanilla YOLOv8, the latest in the YOLO series, outperformed its predecessors by a significant margin, achieving an mAP of 75.45%. Its architectural enhancements, including anchor-free detection and improved feature aggregation, resulted in superior performance across all defect types, particularly porosity and cracks. Two-stage detectors, which operate by generating region proposals before refining predictions, typically achieve higher accuracy at the cost of slower inference speeds. Faster R-CNN, utilizing ResNet50 as its backbone, achieved an mAP of 73.84%. While it excelled in detecting larger defects like shear and cracks, its slower inference speed reduced its practicality for real-time applications. Mask R-CNN, known for its instance segmentation capabilities, achieved competitive results, with an mAP of 73.36%, particularly excelling in porosity and cracks. Cascade R-CNN, leveraging cascaded stages for improved detection accuracy, demonstrated strong performance with an mAP of 75.58%. DDN (Dual Detection Network), designed specifically for defect detection, achieved an mAP of 74.7%. While it showed a balanced performance across defect categories, it fell short of our proposed model in precision and recall.

4.3.2. Ablation Study

An ablation study is conducted in order to identify some of the key contributions of the proposed model, particularly its multi-head attention mechanism and transfer learning effects. All models including YOLOv5s, YOLOv8, and our proposed architecture were exported to the ONNX format and executed using native PyTorch and ONNX Runtime on CPU, explicitly without TensorRT optimizations, to ensure a fair and consistent benchmarking environment. Under this setup, the YOLOv5s model achieved a CPU inference speed of 24.6 FPS, with mAP, recall, and F1-scores of 75.8%, 76.5%, and 77.1%, respectively. YOLOv8 improved marginally with an mAP of 76.7%, recall of 77.9%, and F1 of 78.3%, albeit at a slightly reduced speed of 21.5 FPS. In contrast, our proposed method demonstrated superior accuracy, achieving an mAP of 78.4%, recall of 79.2%, and F1-score of 79.4%, while maintaining a reasonable throughput of 20.1 FPS. These results confirm that the proposed enhancements improve detection accuracy, especially for finer defects, without sacrificing real-time viability for edge deployment.

Initially, to quantitatively validate our backbone selection, we benchmarked YOLOv8 against several efficient models frequently adopted for edge-oriented deployment, including GhostNet, a highly efficient backbone noted for its mobile deployment efficiency. GhostNet achieved 76.0% mAP and 76.8% recall at 61.3 FPS, demonstrating strong inference speed but lower accuracy. By contrast, our YOLOv8-based backbone reached 78.4% mAP and 79.2% recall at 57.5 FPS, highlighting the superior trade-off for LIB defect detection.

To systematically assess the contribution of each architectural component within our proposed detection framework, we conducted a detailed ablation study. Beginning with a baseline model comprising an EfficientNet-B0 backbone and a standard YOLOv8 detection head, we incrementally integrated our key design choices and measured their impact on performance. Specifically, we evaluated the effect of (i) multi-scale dilated branches for spatial context capture, (ii) multi-head self-attention for feature refinement, (iii) Squeeze-and-Excitation (SE) modules for adaptive channel weighting, and (iv) layer normalization for stable training convergence. As shown in Table 6, the inclusion of each module yielded consistent improvements in both detection accuracy and robustness. The integration of multi-scale branches and attention mechanisms led to notable gains in mAP and F1-score, reflecting enhanced sensitivity to diverse defect patterns.

Table 6. Ablation study evaluating the incremental contribution of key architectural components to detection performance on the private LIB weld defect dataset.

Model Variant	mAP (%)	Precision (%)	FPS (GPU)
Baseline EfficientNet-B0 + YOLOv8	73.2	71.5	64.8
Add Multi-scale Dilation	75.1	73.0	62.0
Add Multi-head Attention	76.3	74.2	60.5
Add Squeeze-and-Excitation (SE)	77.4	75.0	59.7
Add Layer Normalization	78.0	76.2	58.2
Our Model (Complete Configuration)	78.4	77.0	57.5

Table 7 presents a focused ablation analysis of the proposed model, evaluating the incremental impact of the multi-head attention mechanism and transfer learning. The results quantified using mAP, precision, recall, and F1-score confirm that both components yield clear improvements. Notably, integrating multi-head attention and transfer learning in our full configuration led to an mAP of 78.4%, a precision of 80.4%, a recall of 78.5%, and an F1-score of 79.4%, significantly surpassing the baseline. The absence of transfer learning led to a longer convergence time and resulted in an mAP of 75.1%, compared to 78.4% with transfer learning. This confirms that pre-training on a related defect dataset provides a valuable starting point, reducing the number of epochs needed for training while achieving superior generalization across diverse defect types. This systematic breakdown validates that our model's architectural enhancements directly contribute to more reliable and generalizable ultrasonic weld defect detection, which is critical for deployment in dynamic manufacturing environments.

Table 7. Ablation study for various components of our model on various metrics.

Model	mAP (%)	Shear (%)	Porosity (%)	Crack (%)	Fold (%)	Crease (%)
Vanilla YOLOv8 (transfer learning)	72.84	72.9	78.2	79.5	68.5	65.1
Without Multi-Head Attention	73.46	73.2	79	80.1	69	66
All components (Without Transfer Learning)	75.1	76.7	80.6	82	72	68
Our Model (complete configuration)	78.4	78.5	82.1	83.7	75.3	72

In addition to performance benchmarks, we reinforced the statistical reliability of our findings. Specifically, we report the mean and standard deviation of the mAP, recall, and F1 metrics averaged over three independent runs using different random seeds. These results are summarized in Table 8 and demonstrate consistent performance with minimal variance, confirming the robustness of our model across runs. Furthermore, we performed paired *t*-tests between the proposed model and the YOLOv8 baseline, validating that the improvements observed are statistically significant ($p < 0.05$). This strengthens the credibility of the performance claims and ensures reproducibility across hardware and experimental conditions.

Table 8. Statistical consistency across multiple runs (mean ± std. dev over 3 seeds).

Model	mAP (%)	Recall (%)	F1 (%)
YOLOv5s	75.8 ± 0.21	76.5 ± 0.28	77.1 ± 0.25
YOLOv8	76.7 ± 0.19	77.9 ± 0.22	78.3 ± 0.23
Proposed	78.4 ± 0.17	79.2 ± 0.20	79.4 ± 0.18

5. Discussion

The findings of this study underscore how lightweight, attention-enhanced CNNs can transform industrial quality assurance in lithium-ion battery manufacturing. By integrating structured multi-scale fusion with transformer-style attention, the framework shows that real-time inspection is achievable without reliance on computationally expensive architectures. This balance between efficiency and accuracy directly addresses the manufacturing sector's need for scalable, inline solutions. A key implication lies in the feasibility of cost-effective deployment. The framework achieves stable inference at ~45 FPS, indicating that high-throughput inspection can be realized even in facilities without dedicated GPUs. This lowers the barrier to adoption and makes automated defect detection accessible to a wider range of production environments. Moreover, attention-based modules improve recall by reducing false negatives, critical for minimizing defect escapes in safety-sensitive components.

At the same time, trade-offs must be acknowledged. Transformer-based detectors such as DETR and Swin Transformer deliver state-of-the-art accuracy on benchmark datasets but are constrained by high latency and computational demands. In contrast, the proposed approach provides a middle ground, achieving competitive performance while maintaining real-time inference, making it more appropriate for industrial lines where speed, cost, and explainability are equally critical. Per-class error analysis reveals that defects such as shear, crease, and fold remain difficult to separate due to low inter-class margins and overlapping texture boundaries. These recurring errors point toward future refinements, such as adaptive hierarchical attention or deformable sampling, to capture subtle weld topologies more effectively. Deployment risks also persist: reliance on a fixed monocular camera setup limits robustness under varying viewpoints or lighting, while GAN-based augmentation, though improving minor class recall by 3–4%, does not fully eliminate the domain gap between synthetic and real welds.

Looking forward, short-term improvements may explore expanding datasets with new defect categories and multi-view capture strategies to reduce reliance on synthetic augmentation. Long-term directions may investigate federated learning for cross-factory generalization and multimodal fusion of ultrasonic and visual signals to strengthen robustness and scalability. By distinguishing between immediate and strategic research avenues, this work establishes a pathway toward lightweight, interpretable, and industrially deployable defect detection systems.

6. Conclusions

This study proposed a fractal-inspired, multi-scale convolutional framework built on YOLOv8 to address the challenges of weld defect detection in lithium-ion battery manufacturing. By organizing receptive fields through parallel dilated convolutions and refining representations with multi-head attention, the model effectively captures both subtle defect textures and broader structural patterns while maintaining a compact computational footprint. Comprehensive validation across three heterogeneous datasets—private industrial, RIAWELC, and GC10-DET—showed consistent gains in recall and F1-score, underscoring the framework's ability to generalize across defect types and imaging conditions. These improvements highlight the benefit of combining lightweight convolutional backbones with structured multi-scale fusion and attention mechanisms, which together enhance sensitivity to small and occluded defects without compromising efficiency. Equally important, the architecture is designed with deployment in mind; achieving real-time inference speeds on accessible hardware demonstrates that accurate and interpretable defect detection can be integrated directly into production lines without heavy deployment costs. This balance of accuracy, efficiency, and transparency positions the proposed approach as a practical and scalable solution for industrial quality assurance in next-generation battery manufacturing.

Author Contributions: W.R. and A.U. have contributed equally to this work and are the first coauthors. W.R. writing—manuscript, methodology, software, visualization, data curation, and funding acquisition. X.Q. and J.J. review—manuscript, supervision, data curation, project leader, funding acquisition, and supervision. A.U. writing—manuscript, software, resources, and visualization. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the school-level project of SZPU Grants (No. 6021330010K and No. 6023310026K), the Post-doctoral Later-stage Foundation Project of Shenzhen Polytechnic University (No. 6021271014K), the Shenzhen Polytechnic Scientific Research Start-up Project (No. 6022312028K) and by the Post-doctoral Foundation Project of Shenzhen Polytechnic University (No. 6024331007K and 6024331021K). This work was supported by the Post-doctoral Foundation Project of Shenzhen Polytechnic University (Grant No. 6024331008K).

Data Availability Statement: The industrial dataset generated and/or analyzed during the current study could be available on reasonable request. Therefore, all requests related to data availability should be addressed to the main author.

Acknowledgments: The authors of this manuscript acknowledge everyone who played their role in data gathering.

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

References

1. Pfleging, W. Recent progress in laser texturing of battery materials: A review of tuning electrochemical performances, related material development, and prospects for large-scale manufacturing. *Int. J. Extrem. Manuf.* **2020**, *3*, 012002. [CrossRef]
2. Dong, W.; Meng, X.; Xie, Y.; Zhang, Z.; Tian, H.; Sun, X.; Huang, Y. Wire-Based Friction Stir Welding Enables Equal-Strength Joining of Aluminum Alloys Even with Assembling Gaps. *J. Manuf. Process.* **2025**, *151*, 426–433. [CrossRef]
3. Zhang, Z.; Wen, G.; Chen, S. On-line monitoring and defects detection of robotic arc welding: A review and future challenges. *Trans. Intell. Weld. Manuf.* **2019**, *2*, 3–28.
4. Shaloo, M.; Schnall, M.; Klein, T.; Huber, N.; Reitinger, B. A review of non-destructive testing techniques for defect detection: Application to fusion welding. *Materials* **2022**, *15*, 3697. [CrossRef]
5. Butt, M.H.F.; Li, J.P.; Ji, J.C.; Riaz, W.; Anwar, N.; Butt, F.F.; Ahmad, M.; Saboor, A.; Ali, A.; Uddin, M.Y. Intelligent tumor tissue classification for Hybrid Health Care Units. *Front. Med.* **2024**, *11*, 1385524. [CrossRef] [PubMed] [PubMed Central]
6. Riaz, W.; Ji, J.; Zaman, K.; Zengkang, G. Neural Network-Based Emotion Classification in Medical Robotics: Anticipating Enhanced Human–Robot Interaction in Healthcare. *Electronics* **2025**, *14*, 1320. [CrossRef]
7. Wang, Z.; Li, J.; Yuan, Y.; Zhang, S.; Hu, W.; Ma, J.; Tan, J. Digital-Twin-Enabled Online Wrinkling Monitoring of Metal Tube Bending Manufacturing: A Multi-Fidelity Approach Using Forward-Convolution-GAN. *Appl. Soft Comput.* **2025**, *171*, 112684. [CrossRef]
8. Fan, D.; Yu, C.; Sha, L.; Zhang, H.; Liu, X. Failure Detection of Laser Welding Seam for Electric Automotive Brake Joints Based on Image Feature Extraction. *Machines* **2025**, *13*, 616. [CrossRef]
9. Basamakis, F.P.; Dimosthenopoulos, D.; Bavelos, A.C.; Michalos, G.; Makris, S. A Deep Semantic Segmentation Approach to Accurately Detect Seam Gap in Fixtured Workpiece Laser Welding. *J. Manuf. Mater. Process.* **2025**, *9*, 69. [CrossRef]

10. Tan, M.; Le, Q. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In Proceedings of the International Conference on Machine Learning (ICML), Long Beach, CA, USA, 9–15 June 2019; Volume 36, pp. 6105–6114.
11. Riaz, W.; Ullah, A.; Ji, J. Multi-Scale Attention Networks with Feature Refinement for Medical Item Classification in Intelligent Healthcare Systems. *Sensors* **2025**, *25*, 5305. [CrossRef]
12. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788. [CrossRef]
13. Riaz, W.; Azeem, A.; Chenqiang, G.; Yuxi, Z.; Saifullah; Khalid, W. YOLO Based Recognition Method for Automatic License Plate Recognition. In Proceedings of the 2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA), Dalian, China, 25–27 August 2020; pp. 87–90. [CrossRef]
14. Awan, A.Z.; Ji, J.; Uzair, M.; Ullah, I.; Riaz, W.; Gong, T. Innovative road distress detection (IR-DD): An efficient and scalable deep learning approach. *PeerJ Comput. Sci.* **2024**, *10*, e2038. [CrossRef]
15. Reis, D.; Kupec, J.; Hong, J.; Daoudi, A. Real-Time Flying Object Detection with YOLOv8. *arXiv* **2023**, arXiv:2305.09972. [CrossRef]
16. Honarvar, F.; Varvani-Farahani, A. A review of ultrasonic testing applications: Defect evaluation, material characterization, and process control. *Ultrasonics* **2020**, *108*, 106227. [CrossRef]
17. Nichols, R. Reliability in non-destructive testing. *Nucl. Eng. Des.* **1989**, *114*, 1–32. [CrossRef]
18. Li, Z.; Kong, S.; Qi, S.; Ma, L. Robotic welding method based on semi-supervised segmentation networks. *Measurement* **2025**, *253 Pt B*, 117550. [CrossRef]
19. Rahman, M.M.; Gony, N.; Rahman, M.M.; Rahman, M.M.; SD, M.K.S. Natural language processing in legal document analysis software: A systematic review of current approaches, challenges, and opportunities. *Int. J. Innov. Res. Sci. Stud.* **2025**, *8*, 5026–5042. [CrossRef]
20. Riaz, W.; Ji, J.; Ullah, A. TriViT-Lite: A Compact Vision Transformer–MobileNet Model with Texture-Aware Attention for Real-Time Facial Emotion Recognition in Healthcare. *Electronics* **2025**, *14*, 3256. [CrossRef]
21. Wang, S.; Zhang, E.; Zhou, L.; Han, Y.; Liu, W.; Hong, J. 3DWDC-Net: An improved 3DCNN with separable structure and global attention for weld internal defect classification based on phased array ultrasonic tomography images. *Mech. Syst. Signal Process.* **2025**, *229*, 112564. [CrossRef]
22. Hu, C.; Zhao, C.; Shao, H.; Deng, J.; Wang, Y. TMFF: Trustworthy Multi-Focus Fusion Framework for Multi-Label Sewer Defect Classification in Sewer Inspection Videos. IEEE Trans. Circuits Syst. *Video Technol.* **2024**, *34*, 12274–12287. [CrossRef]
23. Hoffmann, R.; Reich, C. A Systematic Literature Review on Artificial Intelligence and Explainable Artificial Intelligence for Visual Quality Assurance in Manufacturing. *Electronics* **2023**, *12*, 4572. [CrossRef]
24. Chen, L.-C.; Papandreou, G.; Schroff, F.; Adam, H. Rethinking Atrous Convolution for Semantic Image Segmentation. *arXiv* **2017**, arXiv:1706.05587. [CrossRef]
25. Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 936–944. [CrossRef]
26. Xu, S.; Wang, X.; Lv, W.; Chang, Q.; Cui, C.; Deng, K.; Wang, G.; Dang, Q.; Wei, S.; Du, Y.; et al. PP-YOLOE: An Evolved Version of YOLO. *arXiv* **2022**, arXiv:2203.16250. [CrossRef]
27. 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 (ECCV), Glasgow, UK, 23–28 August 2020; pp. 213–229.
28. Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; Dai, J. Deformable DETR: Deformable Transformers for End-to-End Object Detection. In Proceedings of the International Conference on Learning Representations (ICLR), Vienna, Austria, 4 May 2021.
29. Zhao, Y.; Lv, W.; Xu, S.; Wei, J.; Wang, G.; Dang, Q.; Liu, Y.; Chen, J. DETRs Beat YOLOs on Real-Time Object Detection. *arXiv* **2023**, arXiv:2304.08069.
30. Chin, H.-H.; Tsay, R.-S.; Wu, H.-I. A high-performance adaptive quantization approach for edge CNN applications. *arXiv* **2021**, arXiv:2107.08382. [CrossRef]
31. Totino, B.; Spagnolo, F.; Perri, S. RIAWELC: A Novel Dataset of Radiographic Images for Automatic Weld Defects Classification. In Proceedings of the Interdisciplinary Conference on Mechanics, Computers and Electrics (ICMECE 2022), Barcelona, Spain, 6–7 October 2022.
32. Perri, S.; Spagnolo, F.; Frustaci, F.; Corsonello, P. Welding Defects Classification Through a Convolutional Neural Network. *Manuf. Lett.* **2022**, *35*, 29–32. [CrossRef]
33. Lv, X.; Duan, F.; Jiang, J.-J.; Fu, X.; Gan, L. GC10-DET: Metallic Surface Defect Detection. Kaggle. 2020. Available online: https://www.kaggle.com/datasets/alex000kim/gc10det (accessed on 10 July 2025).
34. Dataset Ninja. Visualization Tools for GC10-DET Dataset. Dataset Ninja, 2025. Available online: https://datasetninja.com/gc10-det (accessed on 10 July 2025).
35. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767. [CrossRef]

36. Li, X.; Wang, W.; Wu, L.; Chen, S.; Hu, X.; Li, J.; Tang, J.; Yang, J. Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 21002–21012.
37. Wei, K.; Li, J.; Ma, C.; Ding, M.; Wei, S.; Wu, F.; Chen, G.; Ranbaduge, T. Vertical federated learning: Challenges, methodologies and experiments. *arXiv* **2022**, arXiv:2202.04309. [CrossRef]
38. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU loss: Faster and better learning for bounding box regression. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 12993–13000.
39. Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; Hochreiter, S. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS), Long Beach, CA, USA, 4–9 December 2017; pp. 6626–6637.
40. Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; Chen, X. Improved Techniques for Training GANs. In Proceedings of the 30th International Conference on Neural Information Processing Systems (NeurIPS), Barcelona, Spain, 5–10 December 2016; pp. 2234–2242.
41. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826. [CrossRef]
42. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. In Proceedings of the International Conference on Machine Learning (ICML), Beijing, China, 21–26 June 2014; pp. 1–10. Available online: https://arxiv.org/abs/1409.1556 (accessed on 28 March 2025).
43. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef]
44. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105. Available online: https://papers.nips.cc/paper_files/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html (accessed on 28 March 2025).
45. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1–9.
46. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
47. Howard, A.; Sandler, M.; Chu, G.; Chen, L.C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for MobileNetV3. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1314–1324.
48. Li, Z.; Xiao, L.; Shen, M.; Tang, X. A Lightweight YOLOv8-Based Model with Squeeze-and-Excitation Version 2 for Crack Detection of Pipelines. *Appl. Soft Comput.* **2025**, *177*, 113260. [CrossRef]
49. Yudistira, N.; Kurita, T. Correlation net: Spatiotemporal multimodal deep learning for action recognition. *Signal Process. Image Commun.* **2020**, *82*, 115731. [CrossRef]
50. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Polosukhin, I. Attention is all you need. *arXiv* **2017**, arXiv:1706.03762.
51. Hoa, N.T.; Quan, T.H.M.; Diep, Q.B. Weld-CNN: Advancing non-destructive testing with a hybrid deep learning model for weld defect detection. *Adv. Mech. Eng.* **2025**, *17*, 16878132251341615.
52. Palma-Ramírez, D.; Ross Veitía, B.D.; Font-Ariosa, P.; Hernández-Herrera, H. Deep convolutional neural network for weld defect classification in radiographic images. *Sci. Direct* **2024**, *10*, e30590. [CrossRef]
53. Totino, B.; Spagnolo, F.; Perri, S. RIAWELC: A novel dataset of radiographic images for weld defect classification. *ICMECE* **2022**, *3*, 13–17. [CrossRef]
54. Liu, M.; Chen, Y.; He, L.; Zhang, Y.; Xie, J. LF-YOLO: A lighter and faster YOLO for weld defect detection of X-ray image. *IEEE Sens. J.* **2023**, *23*, 7430–7439. [CrossRef]
55. Wang, J.; Xu, G.; Yan, F.; Wang, J.; Wang, Z. Defect Transformer: An Efficient Hybrid Transformer Architecture for Surface Defect Detection. *Measurement* **2023**, *211*, 112614. [CrossRef]
56. Lv, X.; Duan, F.; Jiang, J.-j.; Fu, X.; Gan, L. Deep Metallic Surface Defect Detection: The New Benchmark and Detection Network. *Sensors* **2020**, *20*, 1562. [CrossRef]
57. Chan, S.; Li, S.; Zhang, H.; Zhou, X.; Mao, J.; Hong, F. Feature Optimization-Guided High-Precision and Real-Time Metal Surface Defect Detection Network. *Sci. Rep.* **2024**, *14*, 31941. [CrossRef] [PubMed]

58. Muhammad Yasir, S.; Ahn, H. Faster Metallic Surface Defect Detection Using Deep Learning with Channel Shuffling. *Comput. Mater. Contin.* **2023**, *75*, 1847–1861. [CrossRef]
59. Sun, S.; Deng, M.; Yu, X.; Xi, X.; Zhao, L. Self-Adaptive Gamma Context-Aware SSM-Based Model for Metal Defect Detection. *arXiv* **2025**, arXiv:2503.01234.
60. Zhou, C.; Lu, Z.; Lv, Z.; Meng, M.; Tan, Y.; Xia, K.; Liu, K.; Zuo, H. Metal Surface Defect Detection Based on Improved YOLOv5. *Sci. Rep.* **2023**, *13*, 20803.
61. Huang, Y.; Tan, W.; Li, L.; Wu, L. WFRE-YOLOv8s: A New Type of Defect Detector for Steel Surfaces. *Coatings* **2023**, *13*, 2011. [CrossRef]

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.