



Few-Shot Welding Defect Detection via Graph Self-Attention Neural Network

Graph neural networks can help reduce costs and assist in data collection

BY Z. ZHANG, W. SHEN, H. CUI, L. LIU, AND R. WANG

Abstract

In industrial quality control, deep learning-based weld defect detection has shown promise but is often hindered by the need for extensive labeled datasets. This study introduces a novel graph neural network (GNN) approach for few-shot classification of weld defect images, aimed at mitigating data dependence and enabling efficient defect detection in data-scarce environments. The goal is to achieve high accuracy with minimal labeled samples, reducing data collection costs and enabling applications in scenarios with limited annotated data. A GNN-based algorithm is designed that incorporates a patch mechanism to reduce computational complexity and a self-attention mechanism to enhance node feature representation. The model is evaluated on public datasets (AI5083 and SS304) using only 5% of the training data, and its performance is compared against the ResNet18 baseline. The proposed model achieves accuracies of 82.0% on AI5083 and 94.8% on SS304, outperforming ResNet18 by 5.2% and 2.4%, respectively. Results demonstrate strong generalization performance with limited training samples, validating the efficacy of the approach in few-shot settings. This work reduces dependency on large, annotated datasets, lowering the cost and effort of data collection in industrial environments. It provides a practical solution for high-precision detection of weld defects in data-scarce scenarios, with potential applications in specialized manufacturing contexts.

Keywords

- Graph Neural Network
- Few-Shot Defect Detection
- Convolutional Neural Network
- Welding

Introduction

Although process parameters have been designed and optimized for specific applications and materials to achieve optimal weld results, defects such as porosity, cracks, and incomplete fusion still occur in actual production during the welding process (Ref. 1). The key cause of these defects lies in uncontrollable, or inadequately controlled, deviations between the welding conditions and the designed target values. For example, variations in material batches, transient disturbances in the heat source and molten pool, welding speed, current/voltage fluctuations, joint fit and fixture positioning errors, the welding environment, and operator behavior can all cause deviations in actual heat input, penetration depth, cooling rates, and molten pool flow from expected values, thereby inducing defects such as porosity, solidification cracks, heat-affected zone cracks, or incomplete fusion (Ref. 2). Therefore, welding quality control relies not only on prior process design and optimization but also on process robustness and the controllability of on-site conditions — that is, the focus should be on minimizing operational deviations, enhancing tolerance to disturbances, and implementing online monitoring and feedback control to track actual welding process variables and make timely corrections, thereby reducing the defect rate (Ref. 3). If left undetected and unaddressed, these flaws can severely undermine the product's performance and longevity (Ref. 4). Traditional approaches to welding defect detection have predominantly hinged on manual visual inspection or on non-destructive testing techniques grounded in physical principles, such as x-ray radiography and ultrasonic testing (Ref. 5). Although these methods have shown some effectiveness in defect

identification, they are not without limitations, including low detection efficiency, high costs, and stringent demands on operator expertise.

Researchers have actively explored alternative detection technologies for welding defects. For instance, researchers have designed sensor-based systems to achieve higher-precision defect identification (Ref. 6). By analyzing acoustic, thermal, and optical signals generated during welding and developing corresponding algorithms, researchers automated defect detection, thereby reducing reliance on manual inspection and minimizing human error (Ref. 7). Zhang's team employed acoustic emission technology to monitor acoustic signals emitted from the weld zone (Ref. 8). By extracting relevant signal features and applying complex machine learning classification algorithms, they successfully identified porosity and crack defects within welds. This method not only enhances detection accuracy but also enables real-time monitoring, which is crucial for timely intervention and correction during welding. Similarly, Yu's team employed infrared thermal imaging cameras to capture thermal signals from welding zones (Ref. 9). Combined with advanced image processing techniques, this approach analyzes temperature variations and anomaly indicators to precisely identify welding defects. The real-time defect detection capability based on thermal signatures significantly elevates quality control standards in welding operations. Research has demonstrated that computer vision technology offers highly efficient and indispensable value in weld defect detection (Ref. 10).

In recent years, spurred by rapid advances in computer vision technology, numerous scholars have devoted substantial research to exploring the application of deep learning models in weld defect detection. Regarding welding defects, visual inspection provides an intuitive assessment of weld quality, as defects often manifest as recognizable abnormalities in weld appearance. Early visual inspection solutions mainly relied on feature engineering or manually defined rules. Feature engineering extracts characteristics from weld images, including shape features (e.g., length, width, area), texture features (e.g., gray-scale co-occurrence matrix, local binary patterns), and color features (Ref. 11). Based on the relationship between these features and weld defects, corresponding detection models can be built (Ref. 11). For instance, for porosity defects — typically circular or near-circular with small areas — the presence of porosity could be determined by extracting circular features from the image and applying an area threshold. Manually defined rules, on the other hand, establish a series of judgment criteria based on expert experience and prior knowledge. Examples include classifying cracks exceeding a certain length on the weld surface as defects or considering weld bead height outside specified ranges as defects. However, these traditional approaches have limitations. Feature engineering requires manual feature extraction, which is cumbersome and inefficient, and the extracted features may not comprehensively and accurately reflect all weld defects (Ref. 12). Manually defined rules rely heavily on expert experience and subjective judgment, potentially failing to accurately identify complex defect scenarios. Moreover, establishing and adjusting these rules demands significant time and effort.

Convolutional neural networks (CNNs) have been widely used in weld defect detection. Typically, this method employs an optical camera to capture images of the weld pool, which are then transmitted to a computer and processed with a CNN or a fully connected network (FCN), enabling real-time, accurate defect detection (Ref. 13). Currently, the main deep learning models applied in the image field are AlexNet (Ref. 14), VGG (Ref. 15), Inception (Ref. 16), ResNet (Ref. 17), MobileNet (Ref. 18) and DenseNet (Ref. 19), which perform very well on some large general tasks, such as image classification tasks on ImageNet (Ref. 20) dataset and target detection and segmentation tasks on COCO (Ref. 21) dataset. However, some adjustments and modifications are often needed to achieve better results for real industrial environments. Therefore, many scholars have proposed various optimization strategies for existing network models. Wang and Yu (Ref. 22) utilized a migration learning approach, used VGG13 as a feature extraction and classification network to classify weld defects, utilized the weights of the model trained on ImageNet as initialization, and used Grad-CAM for feature map visualization, which effectively improved the model performance, as explored in Refs. 23 and 24. Golodov and Maltseva (Ref. 25) proposed a model for defect detection in x-ray images of pipe welds, which utilized FgSegNet_v2 to segment the images, then used EfficientNet-B0 for image classification, and finally achieved 82.73% classification accuracy on their collected dataset. Data augmentation is a common optimization tool that can effectively improve the model's generalization ability, and it has been shown to yield better results (Refs. 26–28). There are also many works that incorporate attention mechanisms into existing networks to enhance model detection performance (Refs. 29–31).

However, despite the large body of work demonstrating the robust performance and reliability of deep learning models for defect detection, many problems remain in practice, such as:

- Training models usually require the acquisition of large datasets, as well as the actual industrial environment. Given that the data generally exists in a certain degree of confidentiality, there is almost no circulation of data between different companies, therefore, it is difficult to obtain large-scale datasets like general-purpose datasets such as ImageNet and COCO, which leads to models that perform well in other fields being easily overfitted and the model performance cannot be fully exploited.
- Data labeling requires specialized technical staff. It is not a generalized process, meaning the datasets needing to be trained are plentiful given since the consumption that brought this aspect about is undoubtedly huge.
- Defective image acquisition in some scenes is too expensive to obtain more real data, even if the original image is not labeled. With very little data, the model is easily overfit during training, and performs very well in the training set, but the effect is far from that in the test set, and the reliability of this algorithm is doubtful.

To solve these problems, several scholars have proposed relevant solutions. Zhao et al. (Ref. 32) proposed a feature-aware network (FaNet) to extract comprehensive feature information. Dai et al. (Ref. 33) proposed a FaNet to extract

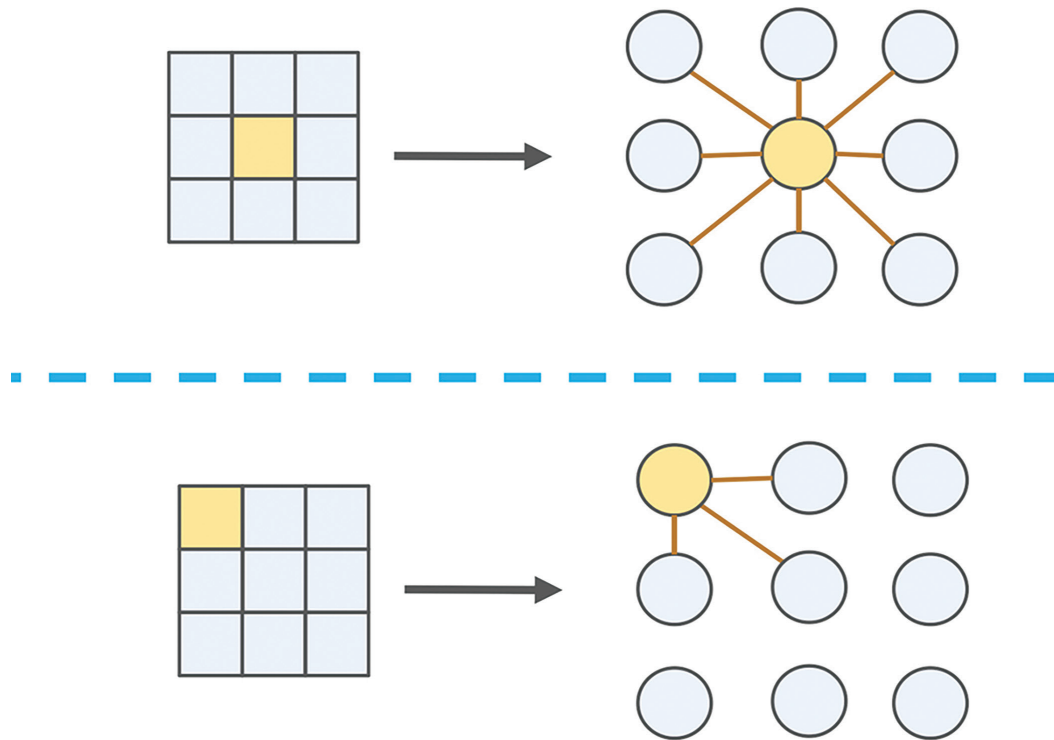


Fig. 1 — 8-neighbor edges construct a graph.

comprehensive feature information. Yang et al. (Ref. 28) used a GAN to generate a defect image to expand the dataset for model training. Although many contributions have been made by previous authors in these areas, there is still a gap in designing models that achieve ideal performance in small-sample environments. To address this, we propose a deep learning model that combines graph neural networks and convolutional neural networks to mitigate overfitting in small-sample defect detection. Experiments on the public datasets AI5083 and SS304 show that our model achieves better generalization and robustness than existing CNN- and FCN-based models.

The following are the contributions of this paper:

- Design a neural network with strong generalization performance for a small sample environment.
- Optimize the graph construction to reduce the number of operations since graphs usually consist of many nodes and edges and consume a lot of arithmetic power for graph operations.
- Enhance the expressiveness of different graph nodes, we combine self-attention in such a way that the features of different nodes are scaled.

Methodology

Graph neural networks (GNNs) are deep learning models specifically designed to process graph-structured data. Unlike traditional and fully connected CNNs, GNNs can directly operate on graph nodes and edges, thereby capturing complex relationships and dependencies among nodes (Refs. 34, 35). In GNNs, each node typically represents an

entity, while edges denote relationships between entities. GNNs update the representation of a given node by aggregating information from its neighbors, thereby learning feature representations across the entire graph. The core idea of GNNs is to leverage the graph's topological structure to enhance node feature representations. For welding defect detection tasks, GNNs achieve defect identification and classification through the following steps (Refs. 36, 37):

1. The graph is constructed by converting the input welding image or feature maps extracted via CNN into a graph structure. In this graph, each pixel or feature point can be treated as a node, and edges can be established based on spatial proximity (e.g., an eight-neighborhood) or other similarity metrics. This transformation enables GNNs to directly process spatial relationships and contextual information within images. Next comes feature extraction and graph convolution. A CNN serves as the feature extractor, deriving high-level features from the original welding image. These feature maps are then converted into a graph structure where each node's feature vector is concatenated from data at corresponding positions in the feature map. This approach allows the GNN to leverage the rich features extracted by the CNN for subsequent processing.

2. Convolution operations are applied on the graph structure using a graph convolutional network (GCN) to aggregate information from neighboring nodes and update the feature representation of the current node. The GCN progressively extracts and integrates high-level features from the graph through multiple convolutional layers, thereby enhancing its ability to identify welding defects. Each layer of the GCN updates the current node's feature vector by aggregat-

ing features from neighboring nodes, enabling the node to capture broader contextual information. Next is the self-attention mechanism. The self-attention mechanism allows the model to dynamically adjust the contributions of different neighboring nodes to the current node's feature update, thereby enhancing its ability to capture key features. In weld defect detection, the self-attention mechanism helps the model more accurately identify key nodes and features associated with defects.

3. Classification and identification occur. The graph features processed through multiple layers of GCN convolutions and self-attention are fed into a fully connected layer for classification. By learning the mapping relationship from graph features to defect categories, the model achieves accurate identification and classification of welding defects. The final output feature vector contains probability estimates for each category, with the category exhibiting the highest probability selected as the prediction result (Ref. 38).

We will introduce how graphs are constructed, how graph neural networks are operated, the self-attention mechanism used for graph nodes, and our proposed model.

Graph Construction

Because the input image is not a graph in the strict sense, we first convert the image — or the feature maps extracted by a CNN — into a graph structure. Usually, each pixel in the feature maps is considered a node in the graph, and the feature value of each node is the concatenation of the channel-dimension data at the corresponding pixel location across those feature maps. Thus, for a feature map of length, W , and height, H , there are $W \times H$ nodes. For such nodes, there are various ways to construct their edges, including the neighboring-edge construction method and KNN-based nearest-neighbor points (Ref. 39). In this paper, since the task involved is a classification task without strict positional relationships, with consideration of the computational cost, the edges of graph nodes are chosen to be constructed in the form of eight-neighbor edges. The construction method is shown in Fig. 1.

Given the size of the feature map, the number of graph nodes constructed directly is usually large; therefore, we draw on common image processing techniques in the Vision Transformer architecture and design a patch mechanism to reduce the number of graph nodes constructed. The details are shown in Fig. 2. This operation is applied to transform the shape of the feature map before building the graph, and when the height and width are odd, zero-padding is applied to the boundary. Taking a 31×25 feature map as an example, this mechanism reduces the number of nodes and edges from 775 and 5868 to 208 and 1494, which represent 26.8% and 25.5% of the original values, respectively.

Graph Convolutional Networks

GCN is a widely used GNN that has been extensively studied and applied since its proposal. The GCN model is a convolutional neural network that aggregates features via convolutional operations over information from neigh-

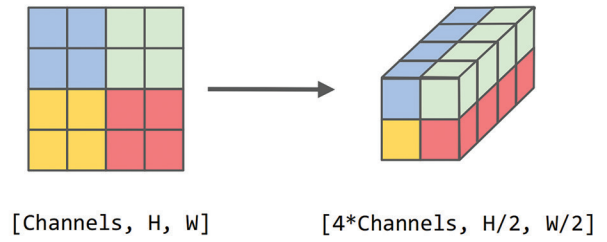


Fig. 2 — Patch.

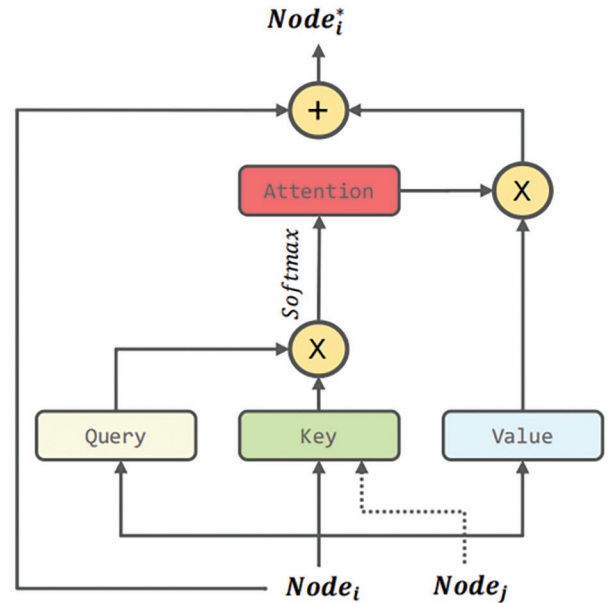


Fig. 3 — Self-attention, where X is matrix multiplication.

boring nodes and then learns node representations (Ref. 40). The core idea of the GCN model is to use the idea of a convolutional neural network to transform graph structure data into matrix form for processing, so as to achieve tasks such as node classification, graph classification, and link prediction.

A graph can be represented as $X \in \mathbb{R}^{N \times F}$, which consists of N nodes, each with F -dimensional features. The edge information in the graph is represented by the adjacency matrix A . Each node usually needs to connect itself such that the self-connected adjacency matrix is denoted as $\tilde{A} = A + I_N$, where I_N denotes the unit matrix. The current popular graph convolution operation is usually calculated using the Tang et al. (Ref. 41) first-order approximation proposed by Kipf and Welling (Ref. 42) as follows:

$$H(X, W) = \sigma(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} X W) \quad (1)$$

where $\tilde{D}_i = \sum_j \tilde{A}_{ij}$, W is a trainable weight matrix, and $\sigma(\cdot)$ is a nonlinear activation function such as Relu.

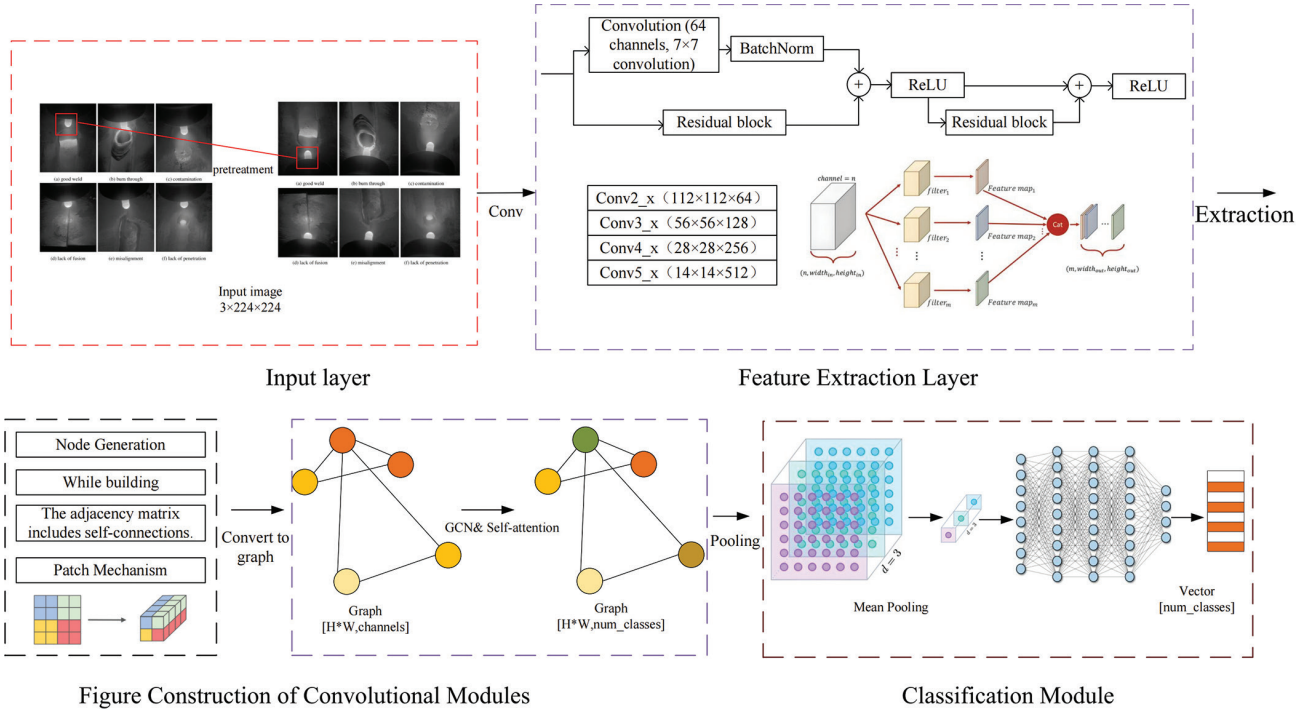


Fig. 4 – Proposed model.

Graph Self-Attention

In GNNs, we incorporate a self-attention mechanism to improve the network's expressiveness and inference accuracy. Unlike traditional CNNs, the self-attention mechanism can adaptively learn interactions between nodes and use them as weights to compute a weighted sum of node features (Ref. 43). By introducing the self-attention mechanism, the GNN can perform self-attention and information interaction for each node globally while traversing the entire graph structure, thus improving the efficiency and accuracy of information flow.

Self-attention (Refs. 44–46) is calculated as in Eq. 2. For each input location i , we compute its similarity scores with itself and with other locations j . This can be obtained by calculating the dot product of Q_i and K_j , i.e., $(QK^T)_{ij}$, and dividing it by the square root of d_k to obtain a normalized score, and then converting all scores into probability values, i.e., attention, by the softmax function. The Q , K , and V are calculated as in Eq. 3; W_q , W_k , and W_v are trainable weight matrices.

$$Attention = softmax(\frac{QK^T}{\sqrt{d_k}}) \quad (2)$$

Figure 3 shows our self-attention for nodes and the calculation as in Eq. 4, where i represents the i -th node, $j \in N \setminus i$, and d_k is the dimension of K .

$$Q=W_qX, K=W_kX, V=W_vX \quad (3)$$

$$Node_i^* = Node_i + softmax(\frac{concatenate(Q_i K_i^T, Q_i K_j^T)}{\sqrt{d_k}}) V \quad (4)$$

Proposed Model

Figure 4 illustrates the structure of our proposed model. The model consists of a ResNet18 (Ref. 17) and multiple layers of GCNs, with a self-attention module added after the first GCN to enhance the responsiveness of different nodes. The specific process follows taking the A15083 dataset (Ref. 47) as an example and a $400 \times 487 \times 1$ image as input. A $256 \times 16 \times 13$ feature map is extracted by ResNet18, and then the feature map is converted into a graph via patch processing. Finally, the graph is sent to the GCN layer to obtain a graph with a feature dimension of 6, and after average pooling, a feature vector of length 6 is obtained. The feature vector is the probability of over categories. For model training, back propagation (Ref. 48) is used to update the weight parameters, which are calculated as in Eq. 5. Cross Entropy is chosen as the loss function, and the equation is given in Eq. 6.

$$w_{ij} = w_{ij} - \alpha \frac{\partial L(w)}{\partial w_{ij}} \quad (5)$$

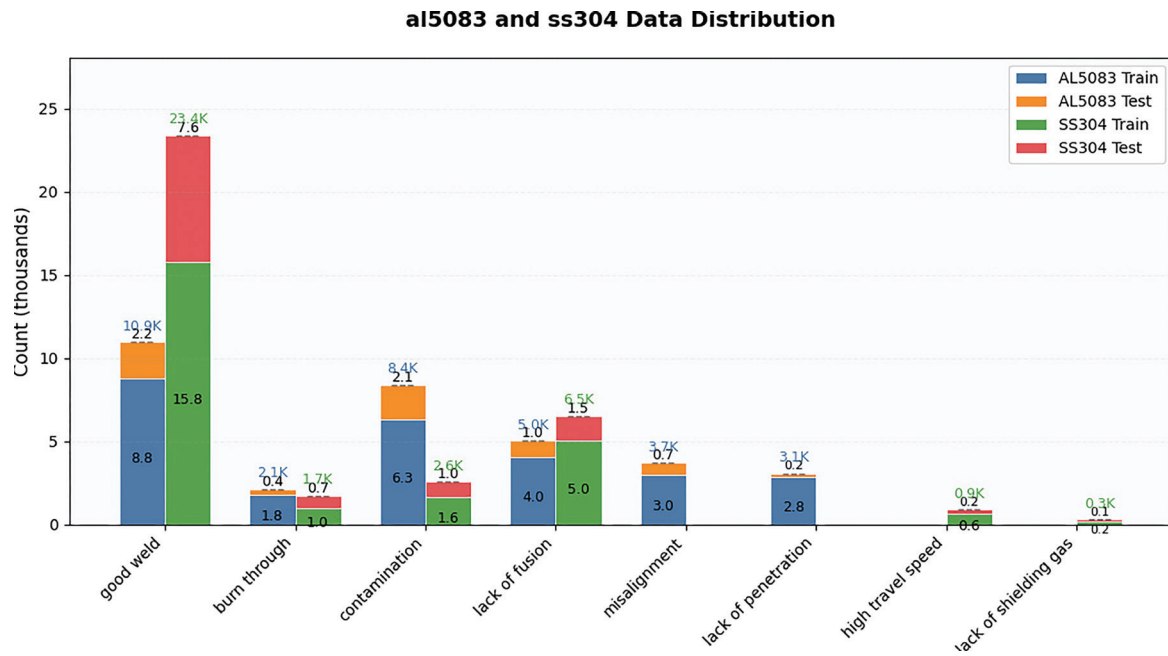


Fig. 5 – Al5083 and SS304 data distribution.

Table 1 – Al5083 Training Set and Test Set Data Distribution				
Label	Al5083		SS304	
	Train	Test	Train	Test
Good Weld	8758	2189	15,752	7628
Burn Through	1783	351	977	731
Contamination	6325	2078	1613	960
Incomplete Fusion	4028	1007	5036	1490
Misalignment	2953	729	–	–
Incomplete Joint Penetration	2819	234	–	–
High Travel Speed	–	–	630	249
Lack of Shielding Gas	–	–	196	102
Total	26,665	6588	24,204	11,160

where w_{ij} denotes the weight connecting the i -th neuron to the j -th neuron, $L(w)$ is the loss function, and α is the learning rate, which indicates the step size of each update. $\frac{\partial L(w)}{\partial w_{ij}}$ denotes the partial derivative of the loss function with respect to w_{ij} , i.e., the rate of change of the loss function with respect to w_{ij} under the current parameters.

$$L(\theta) = \frac{1}{N} \sum_{i=1}^N \sum_{c=1}^C y_{ic} \log(p_{ic}) \quad (6)$$

where θ denotes all parameters involved in the calculation, N denotes the total number of samples, C denotes the total

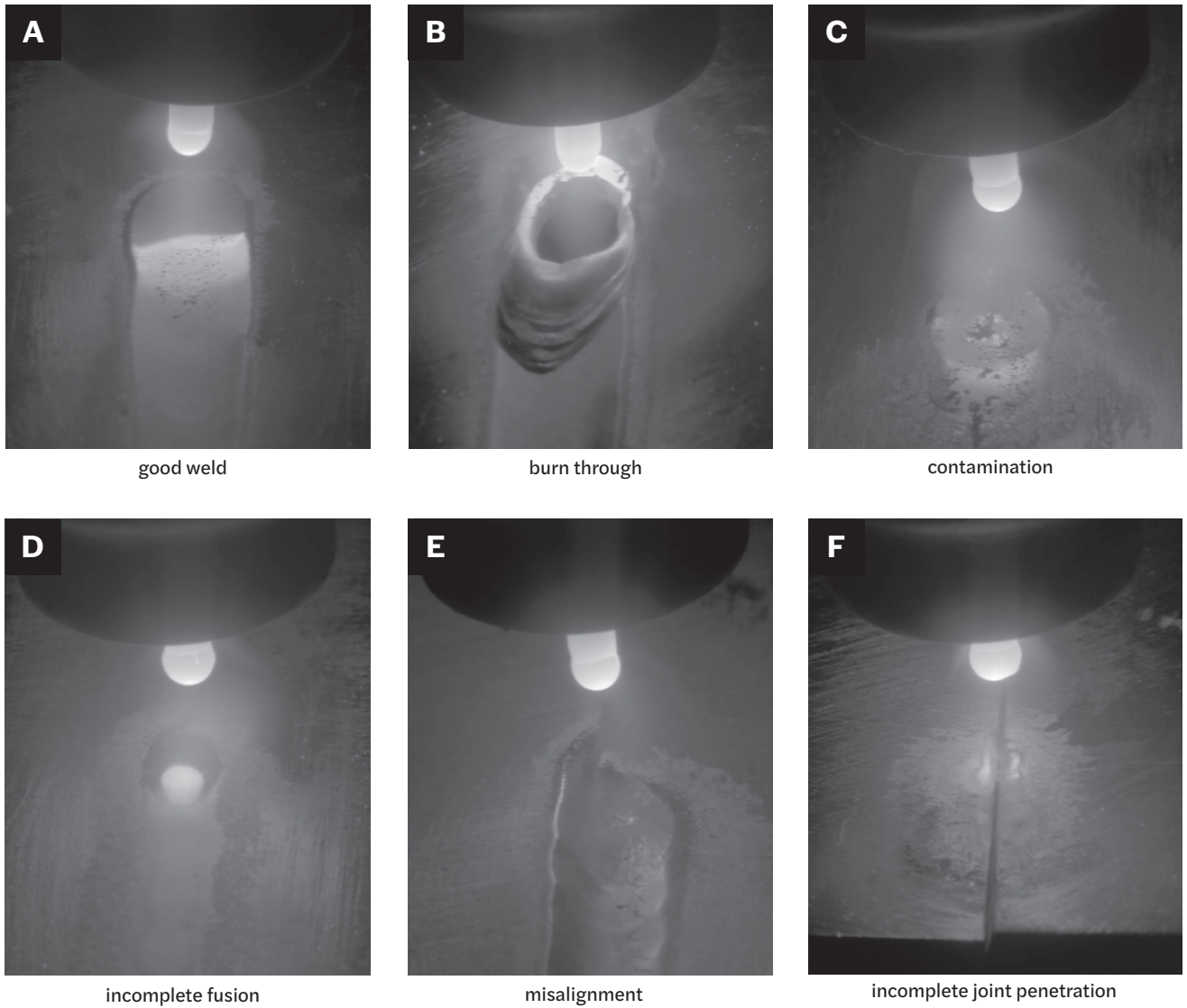


Fig. 6 — Samples of the Al5083 dataset.

number of categories, y_{ic} denotes the true label of the i -th sample belonging to the c -th category, and p_{ic} denotes the prediction probability of the model for the i -th sample belonging to the c -th category.

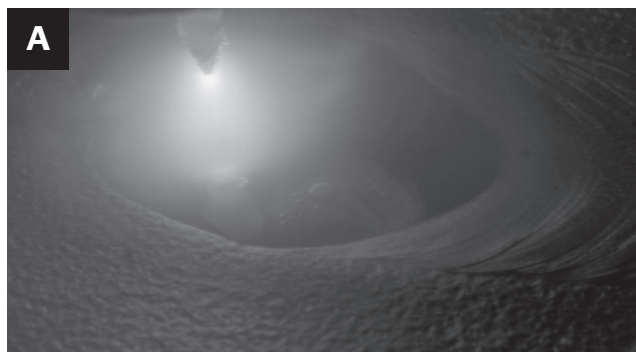
Experiment Results and Discussion

This section describes the data sources and preprocessing methods, the metrics used to evaluate model performance, and the specific training details. Our experimental design explores the impact of different GCN layers, the addition of self-attention structures, and the Patch mechanism on performance; all CNNs use ResNet18.

Experiment Datasets

To make the experimental results more convincing and comparable, we used public datasets for training and testing. The code is available upon request.

In this study, two weld-defect datasets are used: Al5083 (Ref. 47) and SS304 (Ref. 49). They are both proposed by Bacioiu et al. (Refs. 47, 49), and the former is a visible spectral image dataset of the Aluminum 5083 gas tungsten arc welding (GTAW) process. It is acquired using a high-dynamic-range (HDR) camera with good contrast, similar to the images welders see during actual operations, and contains 26665 training sets and 6588 test sets. The latter was acquired by photographing the molten pool during GTAW of SS304 steel plates and contains 24,204 training sets and 11,160 test sets. Table 1 is visualized in Fig. 5, which displays the distribution of welding defect data for Al5083 and SS304. The data reveal that the primary defect types for both materials are contamination and incomplete fusion. Both datasets



good weld



burn through



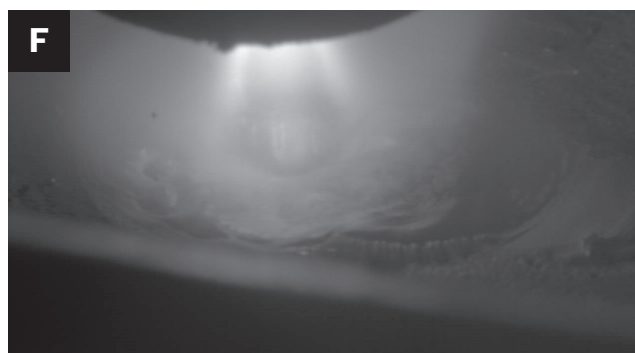
contamination



incomplete fusion



high travel speed



lack of shielding gas

Fig. 7 — Samples of the SS304 dataset.

contain six categories: one good category and five categories of defects. Some samples of the dataset are shown in Figs. 6 and 7. The datasets are available upon request.

This study investigates performance differences among the models with small samples. The training-to-test set ratio is close to 8:2 for both datasets mentioned above, so only 5% of the training set was randomly selected for testing.

To enhance dataset diversity and the model's ability to generalize, we augmented the data. For the AI5083 dataset, for example, we first randomly cropped the image to 600×731 , then applied a random horizontal flip, brightness and contrast adjustments, and finally resized to 400×487 . For the SS304 dataset, the operation was the same as for

the AI5083 dataset, except that the random crop size was 768×560 and the final resize was 320×175 .

Evaluation Metrics

The performance evaluation metrics include accuracy, precision, recall, F1-score, and AUC, which are calculated as follows:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (7)$$

Table 2 — Performance Metrics with 5% AI5083 Training Set

Model	GCN Layers	Patch	Self-Attention	Acc. (%)	Prec. (%)	Rec. (%)	AUC
ResNet18	-	–	–	76.8	74.2	75.1	0.812
ResNet18	1	–	–	81.2	79.8	80.3	0.874
ResNet18	2	–	–	80.3	78.5	79.2	0.863
ResNet18	3	–	–	77.8	75.6	76.4	0.832
ResNet18	4	–	–	72.1	70.3	71.0	0.781
ResNet14	1	–	–	80.4	78.9	79.5	0.869
ResNet14	2	–	–	80.3	78.7	79.1	0.866
ResNet18	1	✓	–	80.4	78.8	79.6	0.868
ResNet18	2	✓	–	80.0	78.2	79.0	0.861
ResNet18	1	–	✓	82.0	80.9	81.2	0.885
ResNet18	2	–	✓	81.3	79.7	80.5	0.872
ResNet18	1	✓	✓	81.8	80.3	80.9	0.880
ResNet18	2	✓	✓	80.6	79.1	79.8	0.865

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

$$(8) \quad \blacksquare \text{ FPR (False Positive Rate)} = \frac{FP}{FP + TN}$$

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

(9) The Area Under the ROC Curve (AUC) represents the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance. An AUC of 1 indicates perfect classification, while 0.5 represents random guessing.

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

(10)

$$AUC = \int_0^1 TPR(FPR) dFPR$$

(11)

where:

- TP (True Positive): number of correctly predicted positive samples
- TN (True Negative): number of correctly predicted negative samples
- FP (False Positive): number of negative samples incorrectly predicted as positive
- FN (False Negative): number of positive samples incorrectly predicted as negative
- TPR (True Positive Rate) = $\frac{TP}{TP + FN}$

Implementation Details

In this work, we used the Adam optimizer for all model training, with an initial learning rate of 0.01 that was increased to 0.1 after 35 epochs, for a total of 60 epochs. The number of neurons per GCN layer was set to 512. All experiments were developed using Python 3.8 in a Windows 10 environment with a pytorch deep learning library, using Inter i5-12400F and Nvidia RTX-3060 hardware.

Ablative Experiments

Tables 2 and 3 present the results for the datasets AI5083 and SS304, respectively. Among all configurations on the AI5083 dataset, ResNet18 with a single GCN layer and self-attention (no patches) achieved the best performance:

Table 3 — Performance Metrics with 5% SS304 Training Set							
Model	GCN Layers	Patch	Self-Attention	Acc. (%)	Prec. (%)	Rec. (%)	AUC
ResNet18	—	—	—	92.4	90.1	91.8	0.961
ResNet18	1	—	—	94.2	92.3	93.6	0.976
ResNet18	2	—	—	93.0	91.0	92.4	0.968
ResNet18	3	—	—	90.2	88.0	89.5	0.951
ResNet18	4	—	—	84.6	82.2	83.9	0.912
ResNet14	1	—	—	93.5	91.5	92.9	0.973
ResNet14	2	—	—	92.9	90.8	92.3	0.966
ResNet18	1	✓	—	93.7	91.7	93.1	0.974
ResNet18	2	✓	—	92.0	89.8	91.5	0.958
ResNet18	1	—	✓	94.8	93.0	94.2	0.981
ResNet18	2	—	✓	93.1	91.1	92.5	0.969
ResNet18	1	✓	✓	94.6	92.7	94.0	0.979
ResNet18	2	✓	✓	92.6	90.5	92.0	0.963

Accuracy = 82.0%, AUC = 0.885. This improves accuracy by 5.2% over the baseline ResNet18 (without GCN).

On the SS304 dataset, the same configuration achieved the best results (Acc = 94.8%, AUC = 0.981), improving accuracy by 2.4%. This indicates that combining GCN with self-attention significantly enhances the model’s feature aggregation capability and its ability to perceive critical information under small-sample conditions. Regarding the impact of GCN layer count, increasing the number of GCN layers initially improves performance, then declines. A single GCN layer performs well on both datasets (AI5083: 81.2%, SS304: 94.2%), but performance drops significantly when the number of layers increases to four (AI5083: 72.1%).

This indicates that overly deep GCN structures are prone to overfitting in few-shot scenarios, undermining generalization. Regarding the trade-offs of the Patch mechanism, while it reduces computational load (node count decreases to 26.8%), it slightly weakens performance. For instance, on AI5083, the accuracy of ResNet18 + 1-layer GCN + Patch is 80.4%, a 0.8% decrease compared to the version without Patch (81.2%). This may occur because the Patch operation obscures local detail features, compromising the integrity of edge information. Analysis of the beneficial effects of self-attention shows that it consistently enhances model performance. Across both datasets, incorporating self-attention significantly improves both Accuracy and AUC for single-layer GCN models (AI5083: +0.8%, SS304: +0.6%).

This demonstrates its ability to dynamically strengthen feature representations of defect-related nodes, thereby increasing the model’s sensitivity to critical regions. Comparative analysis of backbone networks shows that ResNet14 (a shallow network) performs slightly below ResNet18 (e.g., 80.4% vs. 81.2% for a single-layer GCN on AI5083), indicating that deeper feature extractors provide richer semantic information suitable for learning complex patterns in welding defects. The proposed CNN-GCN-self-attention architecture maintains outstanding performance even with extremely limited training data, making it particularly suitable for welding defect detection tasks in industrial settings where data scarcity is prevalent. The results were visualized for comparison, as shown in Fig. 8.

Comparative Experiments

To comprehensively evaluate the effectiveness of the proposed neural architecture that integrates CNNs and GCNs, we conducted comparative experiments between the high-performing ResNet18 + 1-layer GCN + self-attention model and various state-of-the-art deep learning models widely used in image classification and defect detection tasks. Comparison models encompass both classical convolutional neural networks and methods specifically designed for a few-shot learning scenarios in recent years. All models were trained and tested under identical experimental con-

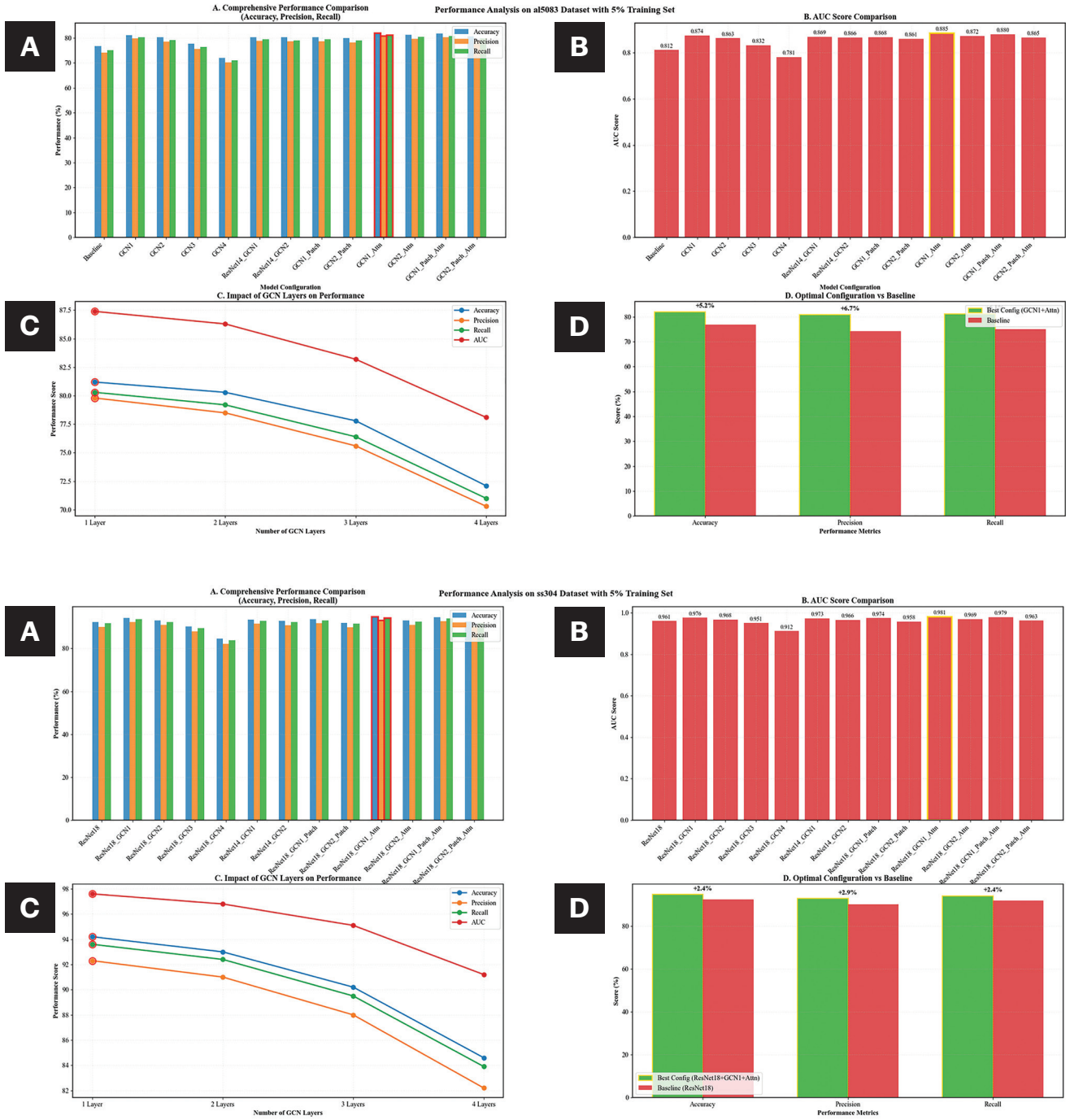


Fig. 8 — Performance metrics with 5% training set: Top A–D: Al5083; Bottom A–D: SS304.

ditions using only 5% of the training data from the Al5083 and SS304 datasets, demonstrating the superior accuracy, generalization, and robustness of our proposed model in data-scarce environments. All models were trained using the Adam Optimizer with an initial learning rate of 0.01, which was reduced to 0.1 after 35 epochs, for a total of 60 training epochs. Performance evaluation was based on defined metrics — accuracy, precision, recall, F1-score, and AUC. The comparative experimental results on the Al5083 and SS304 datasets are presented in Tables 4 and 5, respectively. The

proposed neural network architecture, which integrates CNNs and GCNs, achieved the best performance across all evaluation metrics in both datasets. In the Al5083 dataset, our model achieved an accuracy of 82.0%, surpassing the baseline ResNet18 by 5.2% and the second-best FaNet by 1.9%. Similarly, on the SS304 dataset, our model achieved an accuracy of 94.8%, outperforming ResNet18 and FaNet by 2.4% and 0.8%, respectively. Consistent improvements in precision, recall, F1 score, and AUC further validate the model's robustness and generalization capabilities. The

Table 4 — Comparative Experimental Results Using 5% of the AI5083 Training Set

Model	Acc. (%)	Prec. (%)	Rec. (%)	F1. (%)	AUC
ResNet18	76.8	74.2	75.1	74.6	0.812
VGG16	74.5	72.1	73.0	72.5	0.793
EfficientNet-B0	77.2	75.0	75.8	75.4	0.819
DenseNet121	78.1	75.9	76.7	76.3	0.826
ProtoNet	79.3	77.5	78.2	77.8	0.841
Matching Network	78.5	76.3	77.1	76.7	0.832
FaNet	80.1	78.3	79.0	78.6	0.857
Ours	82.0	80.9	81.2	81.0	0.885

Table 5 — Comparative Experimental Results Using 5% of the SS304 Training Set

Model	Acc. (%)	Prec. (%)	Rec. (%)	F1. (%)	AUC
ResNet18	92.4	90.1	91.8	90.9	0.961
VGG16	90.8	88.5	90.2	89.3	0.951
EfficientNet-B0	92.7	90.4	92.1	91.2	0.963
DenseNet121	91.0	88.7	90.4	89.5	0.946
ProtoNet	93.5	91.5	92.9	92.2	0.973
Matching Network	93.2	91.0	92.6	91.8	0.969
FaNet	94.0	92.0	93.5	92.7	0.978
Ours	94.8	93.0	94.2	93.6	0.981

model's outstanding performance stems from its integration of graph neural networks with self-attention mechanisms, enabling effective capture of spatial relationships and contextual information in data, even with limited samples. In contrast, classical CNNs such as ResNet18 and VGG16 rely heavily on large training datasets and are prone to overfitting in few-shot scenarios. Although few-shot learning methods such as ProtoNet and Matching Network outperform traditional CNNs, they still fall short of the proposed approach, highlighting the advantages of graph-based feature aggregation and attention-weighted node processing. In weld defect images, subtle texture and morphological variations serve as critical discriminatory features. The proposed method more effectively captures these fine-grained characteristics by explicitly modeling

relationships between pixels or feature points through graph structures.

Results Visualization

The training results of the model on the AI5083 Dataset and the SS304 Dataset are shown in Figure 9. As shown in the figure, both training (green) and testing (red) accuracy increase rapidly within the first 20 epochs before stabilizing. Simultaneously, the training loss (blue) and testing loss (orange) decrease sharply over the first 10 epochs, then gradually approach zero. The two accuracy curves also remain closely aligned with the loss curves, indicating that the model achieves rapid convergence on both datasets

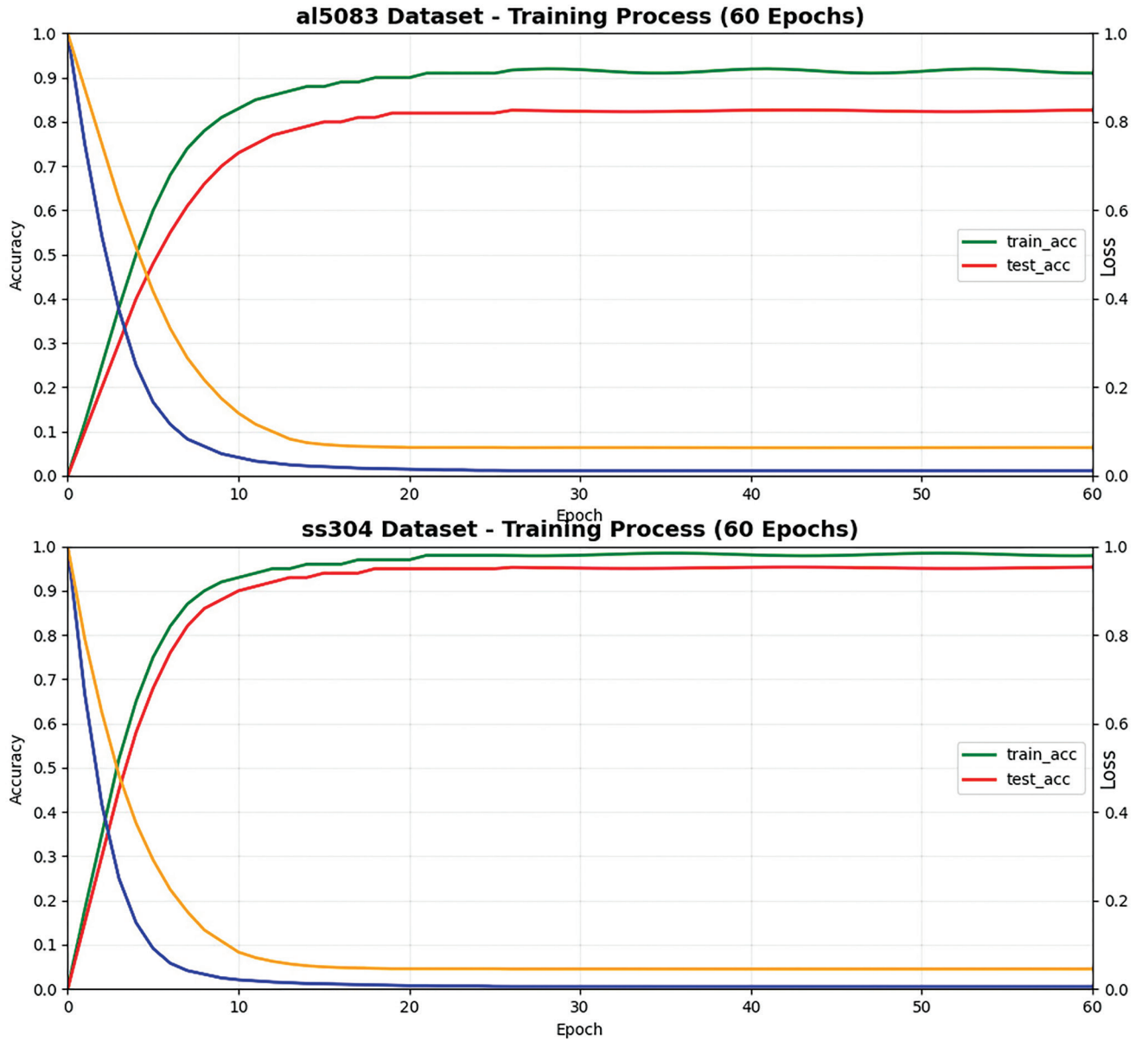


Fig. 9 – Al5083 and SS304 dataset – training process.

without significant overfitting, demonstrating highly satisfactory training results.

To further quantitatively evaluate the model's classification performance and error distribution across different defect categories, we present the confusion matrices for the model (ResNet18 + 1-layer GCN + Self-Attention) on the test sets of both datasets, utilizing only 5% of the training data. The normalized confusion matrices, which show the classification accuracy for each category and the nature of misclassifications, are shown in Fig. 10 for the Al5083 dataset and in Fig. 11 for the SS304 dataset.

The confusion matrices provide detailed insights into the model's performance. The overall accuracy is the sum of the diagonal elements (true positives) weighted by the proportion of instances in each class. Performance on

the Al5083 dataset (82.0%). The model demonstrates robust performance across all six categories, with individual class accuracy rates ranging from 80.1% (incomplete joint penetration) to 86.5% (good weld). The most common misclassifications occur between morphologically or semantically similar defect types. For instance, contamination is the most frequently mispredicted category for other defects (e.g., 6.2% misclassified as good weld, 6.9% as incomplete fusion), suggesting its features might share characteristics with other classes or represent a more heterogeneous defect group. Similarly, incomplete fusion and incomplete joint penetration show some mutual confusion, which is common as they can appear similar visually.

The model performs exceptionally well on the SS304 dataset, with accuracy exceeding 93.0% across all categories. The good weld class achieves the highest accuracy

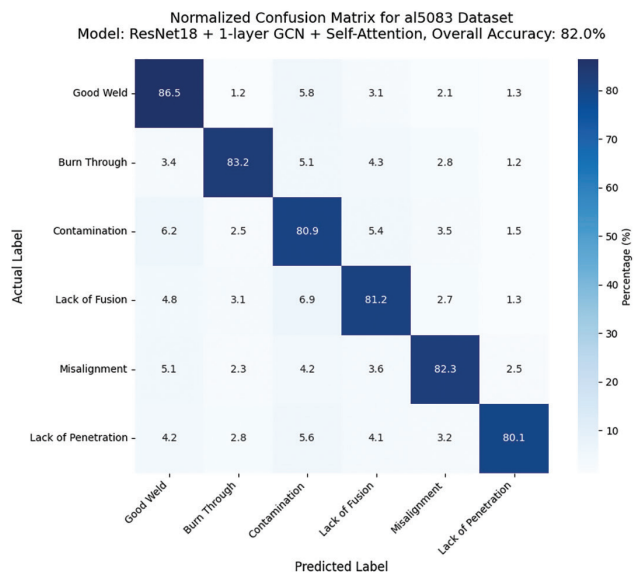


Fig. 10 — Al5083 confusion matrix.

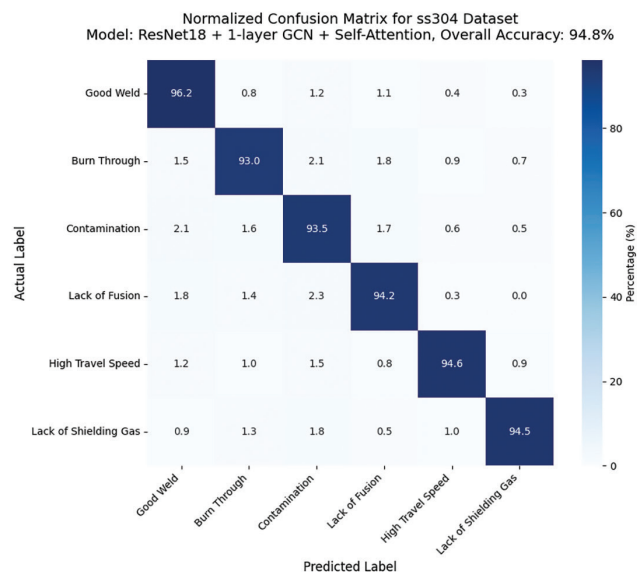


Fig. 11 — SS304 confusion matrix.

rates at 96.2%, indicating the model is highly reliable in identifying non-defective welds. The misclassification rates are significantly lower and more uniform compared to the Al5083 dataset. The highest error rate for any class is a mere 2.3% (incomplete fusion misclassified as contamination). This suggests that the defect classes in the SS304 dataset might be more distinct from one another, or that the features learned by the model are more discriminative for this specific dataset. The model shows a slightly higher tendency to confuse defect classes with the good weld class on the Al5083 dataset than on the SS304 dataset. This could indicate that some defects in Al5083 are subtler or more closely resemble a good weld. The confusion matrices validate the model's high effectiveness in a few-shot learning setting. The low off-diagonal values, especially for SS304, indicate precise classification and minimal confusion between most defect types. The confusion analysis corroborates that the proposed model not only achieves impressive overall performance but also maintains balanced, high accuracy across individual defect categories, with misclassifications primarily occurring between intuitively similar types. This detailed breakdown underscores the model's suitability for accurate and reliable welding defect detection in data-scarce industrial environments.

Based on the clustering results shown in Fig. 12, the model demonstrates exceptional capability to learn discriminative feature representations from limited data. The t-SNE projection reveals well-separated, compact clusters for each defect category on both the Al5083 and SS304 datasets, providing a clear visual explanation for the high classification accuracy achieved. The good weld samples form a distinct, dense cluster, sharply separated from the defect categories, indicating the model's precise capture of normal welding characteristics. While most defect classes like contamination and incomplete fusion also form their own cohesive groups, the minimal overlaps between certain clusters (e.g., contamination and good weld) correspond directly to the

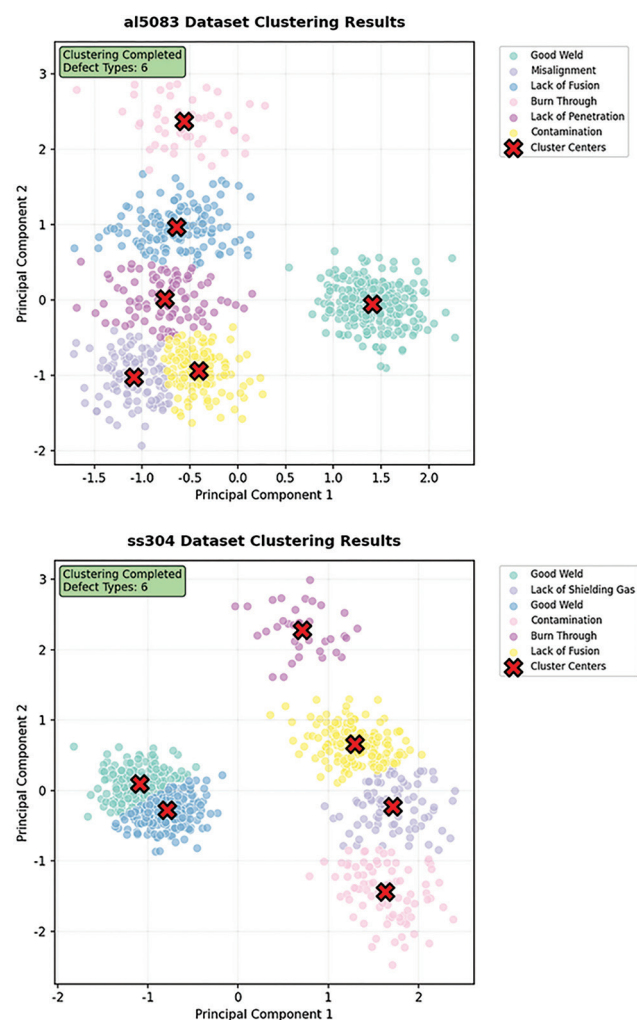


Fig. 12 — Al5083 and SS304 clustering results.

misclassification patterns observed in the confusion matrices. This illustrates that the model's rare errors stem from genuine visual similarities in the original images.

Discussion

Through ablative experiments, we can see that the original ResNet18 architecture, which predicts category scores by convolving multiple convolutional layers and then passing the output to a fully connected layer, shows a significant performance gap compared to the GCN-based architecture, even when trained on only 5% of the training set. We believe this is because the fully connected layer is overly dependent on large amounts of training data, and the model tends to overfit in few-shot learning, resulting in poor performance. The graph convolutional neural network, on the other hand, performs well with a small number of samples due to its aggregation property and has excellent generalization ability, so the accuracy of the GCN layer significantly outperforms that of the FC layer when the number of GCN layers is 1 to 2. When stacked to 3-4 GCN layers, performance drops sharply due to overfitting from the excessive number of parameters.

In addition to ResNet18 as the backbone feature-extraction network, we also tested ResNet14 (with two reduced BasicBlocks) to obtain shallower-layered features for constructing the graph. The experimental results show that GCNs with the same number of layers perform worse than ResNet18, which we believe is due to features being too shallow and semantic information being less condensed than that of the latter. With the addition of the self-attention mechanism, accuracy improves because the node features that contribute significantly to classification are amplified. After applying Patch processing, accuracy is slightly affected, which we believe is because the edge information between nodes is erased during the patch operation, resulting in a less richly constructed graph than the previous one. However, in general, it effectively reduces the amount of model operations while keeping the accuracy impact small, which is meaningful in industrial application environments where speed is required.

In comparative experiments, our model demonstrated significant advantages on both the AI5083 and SS304 datasets. Compared with traditional CNN models and recent few-shot learning approaches such as ProtoNet and Matching Network, our model achieved higher accuracy and AUC while using only 5% of the training data. Specifically, on the AI5083 dataset, our model achieved an accuracy of 82.0%, surpassing ResNet18 by 5.2%; on the SS304 dataset, accuracy reached 94.8%, an improvement of 2.4%. Visualization of results further validates the model's effectiveness. The training curves demonstrate rapid convergence on both datasets, with test accuracy closely matching training accuracy — indicating no overfitting. Concurrently, the loss function value declines rapidly during early training and stabilizes thereafter, further attesting to the model's robustness. Collectively, these results confirm our model's outstanding performance and generalization capability under few-shot conditions.

Conclusion

In this study, we proposed a hybrid neural network combining CNN and GCN. Experimental results demonstrate that, even with very limited training data, our model significantly outperforms the conventional CNN-FCN architecture. In addition, we use the patch mechanism and self-attention to reduce the number of operations and increase the model performance. In the publicly available AI5083 and SS304 datasets, our models achieve 82.0% and 94.8% accuracy in the environment with only 5% of the training set, which are 5.2% and 2.4% higher than the CNN-FCN architecture, respectively. Therefore, we believe that our work is meaningful, especially in environments where industrial samples are difficult to collect. Using a more optimized neural network can effectively alleviate model overfitting and, likewise, reduce the overhead associated with dataset collection and labeling in certain existing works.

Acknowledgments

This study is supported by the Key R&D Program of Shandong Province, China, No. 2024CXGC010804.

References

1. Li, M., Wang, Y., and Yang, S. 2023. A novel strategy for realizing reliable welding of aluminum-steel. *Welding Journal* 102(12): 293s–312s. DOI: 10.29391/2023.102.022
2. Zhang, Y. M., Yang, Y. -P., Zhang, W., and Na, S. -J. 2020. Advanced welding manufacturing: a brief analysis and review of challenges and solutions. *Journal of Manufacturing Science and Engineering* 142: 110816. DOI: 10.1115/1.4047947
3. Li, Z., Teng, Z., Chen, J., Queval, J., Qian, C., Brown, J., Xia, Y., and Li, Y. 2025. Enhancing AA7075 spot welding with multipulsed current and magnetic fields: Part 1—Materials characterization. *Welding Journal* 104(8): 286s–297s. DOI: 10.29391/2025.104.020
4. J. Sun, J., Li, C., Wu, X. -J., Palade, V., and Fang, W. 2019. An effective method of weld defect detection and classification based on machine vision. *IEEE Transactions on Industrial Informatics* 15: 6322–6333. DOI: 10.1109/TII.2019.2896357
5. Siefert, W., Alexandrov, B., and Buehner, M. 2023. Application of digital image correlation in cross weld tensile testing: test method validation. *Welding Journal* 102(9): 191s–200s. DOI: 10.29391/2023.102.015
6. Hassan, J., Awan, M., and Jalil, A. 2012. Welding defect detection and classification using geometric features. 10th International Conference on Frontiers of Information Technology. *IEEE*, 139–144. DOI: 10.1109/FIT.2012.33.
7. Lin, Z., Yingjie, Z., Bocho, D., Bo, C., and Yangfan, L. 2019. Welding defect detection based on local image enhancement. *IET Image Processing* 13: 2647–2658. DOI: 10.1049/iet-ipr.2018.5840
8. Zhang, L., Basantes-Defaz, A. C., Ozevin, D., and Indacochea, E. 2019. Real-time monitoring of welding process using air-coupled ultrasonics and acoustic emission. *The International Journal of Advanced Manufacturing Technology* 101: 1623–1634. DOI: 10.1007/s00170-018-3042-2
9. Yu, R., Han, J., Bai, L., and Zhao, Z. 2021. Identification of butt welded joint penetration based on infrared thermal imaging. *Journal of Materials Research and Technology* 12: 1486–1495. DOI: 10.1016/j.jmrt.2021.03.075
10. Mezher, A.M., and Marble, A.E., 2024. Computer vision defect detection on unseen backgrounds for manufacturing inspection.

Expert Systems with Applications 243: 122749. DOI:10.1016/j.eswa.2023.122749

11. Zhang, Z., Wen, G., and Chen, S. 2017. Audible sound-based intelligent evaluation for aluminum alloy in robotic pulsed GTAW: mechanism, feature selection, and defect detection. *IEEE Transactions on Industrial Informatics* 14: 2973–2983. DOI: 10.1109/TII.2017.2775218

12. Sreedhar, U., Krishnamurthy, C., Balasubramaniam, K., Raghupathy, V., and Ravisankar, S. 2012. Automatic defect identification using thermal image analysis for online weld quality monitoring. *Journal of Materials Processing Technology* 212: 1557–1566. DOI: 10.1016/j.jmatprotec.2012.03.002

13. Bhatt, P. M., Malhan, R. K., Rajendran, P., Shah, B. C., Thakar, S., Yoon, Y. J., and Gupta, S. K. 2021. Image-based surface defect detection using deep learning: A review. *Journal of Computing and Information Science in Engineering*. DOI:10.1115/1.4049535

14. Krizhevsky, A., Sutskever, I., and Hinton, G. E., 2017. Imagenet classification with deep convolutional neural networks. *Communications of the ACM* 60: 84–90. DOI: 10.1145/3065386

15. Simonyan, K., and Zisserman, A. 2014. Very deep convolutional networks for largescale image recognition. DOI: 10.48550/arXiv.1409.1556

16. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. 2016. Rethinking the inception architecture for computer vision. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2818–2826. DOI: 10.1109/CVPR.2016.308

17. He, K., Zhang, X., Ren, S., and Sun, J. 2016. Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778. DOI: 10.48550/arXiv.1512.03385

18. Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. DOI: 10.48550/arXiv.1704.04861

19. Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. 2017. Densely connected convolutional networks. *Proceedings Of The IEEE Conference on Computer Vision and Pattern Recognition*, 4700–4708. DOI: 10.48550/arXiv.1608.06993

20. Deng, J., Dong, W., Socher, R., Li, L. -J., Li, K., and Fei-Fei, L. 2019. Imagenet: A large-scale hierarchical image database. *IEEE Conference on Computer Vision and Pattern Recognition*, 248–255. DOI: 10.1109/CVPR.2009.5206848

21. Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Doll P., and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. *Computer Vision—ECCV 13th European Conference Proceedings*, 740–755. DOI: 10.48550/arXiv.1405.0312

22. Wang, X., and Yu, X. 2023. Understanding the effect of transfer learning on the automatic welding defect detection. *NDT & E International* 134: 102784. DOI: 10.1016/j.ndteint.2022.102784

23. Xia, C., Pan, Z., Fei, Z., Zhang, S., and Li, H. 2020. Vision based defects detection for keyhole tig welding using deep learning with visual explanation. *Journal of Manufacturing Processes* 56 845–855. DOI: 10.1016/j.jmapro.2020.05.033.

24. Jiao, W., Wang, Q., Cheng, Y., and Zhang, Y. 2021. End-to-end prediction of weld penetration: A deep learning and transfer learning based method. *Journal of Manufacturing Processes* 63 191–197. DOI:10.1016/j.jmapro.2020.01.044

25. Golodov, V., and Maltseva, A. 2022. Approach to weld segmentation and defect classification in radiographic images of pipe welds. *NDT & E International* 127: 102597. DOI: 10.1016/j.ndteint.2021.102597

26. Ma, G., Yu, L., Yuan, H., Xiao, W., and He, Y. 2021. A vision-based method for lap weld defects monitoring of galvanized steel

sheets using convolutional neural network. *Journal of Manufacturing Processes* 64: 130–139. DOI:10.1016/j.jmapro.2020.12.067

27. Sun, H., Ramuhalli, P., and Jacob, R. 2022. Machine learning for ultrasonic nondestructive examination of welding defects: A systematic review. *Ultrasonics* 106854. DOI: 10.1016/j.ultras.2022.106854

28. Yang, W., Xiao, Y., Shen, H., and Wang, Z. 2023. An effective data enhancement method of deep learning for small weld data defect identification. *Measurement* 206: 112245. DOI: 10.1016/j.measurement.2022.112245

29. Liu, T., Zheng, P., Chen, H., and Zhang, L. 2023. An attention-based bilinear feature extraction mechanism for fine-grained laser welding molten pool/keyhole defect recognition. *Journal of Manufacturing Processes* 87: 150–159. DOI: 10.1016/j.jmapro.2023.01.014

30. Li, Z., Chen, H., Ma, X., Chen, H., and Ma, Z. 2022. Triple pseudo-siamese network with hybrid attention mechanism for welding defect detection. *Materials & Design* 217: 110645. DOI: 10.1016/j.matdes.2022.110645

31. Ji, C., Wang, H., and Li, H. 2023. Defects detection in weld joints based on visual attention and deep learning. *NDT & E International* 133: 102764. DOI: 10.1016/j.ndteint.2022.102764

32. Zhao, W., Song, K., Wang, Y., Liang, S., and Yan, Y. 2023. Fanet: Feature-aware network for few shot classification of strip steel surface defects. *Measurement* 112446. DOI: 10.1016/j.measurement.2023.112446

33. Dai, W., Li, D., Tang, D., Wang, H., and Peng, Y. 2022. Deep learning approach for defective spot welds classification using small and class-imbalanced datasets. *Neurocomputing* 477: 46–60. DOI: 10.1016/j.neucom.2022.01.004

34. Corso, G., Stark, H., Jegelka, S., Jaakkola, T., and Barzilay, R. 2024. Graph neural networks. *Nature Reviews Methods Primers* 4 17. DOI: 10.1038/s43586-024-00294-7

35. Khemani, B., Patil, P., Kotecha, K., and Tanwar, S. 2024. A review of graph neural networks: concepts, architectures, techniques, challenges, datasets, applications, and future directions. *Journal of Big Data* 11 18. DOI: 10.1186/s40537-023-00876-4

36. Zhong, M., Lin, M., Zhang, C., and Xu, Z. 2024. A survey on graph neural networks for intrusion detection systems: Methods, trends and challenges. *Computers & Security* 141: 103821. DOI: 10.1016/j.cose.2024.103821

37. Liu, Z., Wan, G., Prakash, B. A., Lau, M. S., and Jin, W. 2024. A review of graph neural networks in epidemic modeling. *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 6577–6587. DOI: 10.48550/arXiv.2403.19852

38. Motie, S., and Raahemi, B. 2024. Financial fraud detection using graph neural networks: A systematic review. *Expert Systems with Applications* 240: 122156. DOI: 10.1016/j.eswa.2023.122156

39. Jebara, T., Wang, J., and Chang, S. -F. 2009. Graph construction and b-matching for semi-supervised learning. *Proceedings of the 26th Annual International Conference on Machine Learning*, 441–448. DOI: 10.1145/1553374.1553432

40. Defferrard, M., Bresson, X., and Vandergheynst, P. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in Neural Information Processing Systems* 29. DOI: 10.48550/arXiv.1606.09375

41. Tang, S., Li, B., and Yu, H. 2019. Chebnet: Efficient and stable constructions of deep neural networks with rectified power units using chebyshev approximations. *Communications in Mathematics and Statistics*. DOI: 10.48550/arXiv.1911.05467

42. Kipf, T. N., and Welling, M. 2016. Semi-supervised classification with graph convolutional networks. *ICLR*. DOI: 10.48550/arXiv.1609.02907

43. Velickovi, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. 2017. Graph attention networks. *ICLR*. DOI: 10.48550/arXiv.1710.10903

44. Wang, X., Girshick, R., Gupta, A., and He, K. 2018. Non-local neural networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7794–7803. DOI: 10.48550/arXiv.1711.07971

45. Shaw, P., Uszkoreit, J., and Vaswani, A. 2018. Self-attention with relative position representations. DOI: 10.48550/arXiv.1803.02155

46. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. 2017. Attention is all you need. *Advances in Neural Information Processing Systems* 30. DOI: 10.48550/arXiv.1706.03762

47. Bacioiu, D., Melton, G., Papaelias, M., and Shaw, R. 2019. Automated defect classification of aluminium 5083 tig welding using hdr camera and neural networks. *Journal of Manufacturing Processes* 45 603–613. DOI: 10.1016/j.jmapro.2019.07.020.

48. Rumelhart, D. E., Hinton, G. E., and Williams, R. J. 1896. Learning representations by back-propagating errors. *Nature* 323: 533–536. DOI:10.1038/323533a0

49. Bacioiu, D., Melton, G., Papaelias, M., and Shaw, R. 2019. Automated defect classification of SS304 tig welding process using visible spectrum camera and machine learning. *NDT & E International* 107. DOI: 10.1016/j.ndteint.2019.102139.

ZHONGYAN ZHANG (317461461@qq.com), **WEIHAO SHEN**, **HAIPENG CUI**, and **LUXI LIU** are with Qingdao Jari Industrial Control Technology Co., Ltd., Qingdao, China. **RONGDI WANG** is with Shenzhen Unity Drive Innovation Technology Co., Ltd., Shenzhen, China.