WEP Warwick Evans Publishing

International Journal of Computer Science and Information Technology

ISSN: 3005-9682 (Print), ISSN: 3005-7140 (Online) | Volume 4, Number 1, Year 2024
DOI: https://doi.org/10.62051/ijcsit.v4n1.19
Journal homepage: https://wepub.org/index.php/IJCSIT/index

# Research on Welding Defect Classification Methods Based on Convolutional Neural Networks

Wei Tang, Yan Shi *, Yinghua Liao, Qi Li, Yi Luo

College of Light and Chemical Technology, Mechanical Engineering University, Yibin Sichuan 644000, China

## ABSTRACT

To address the issues of low detection and identification rates and inaccurate classification of welding defects, this paper proposes a welding defect classification method based on convolutional neural networks. By modifying the ResNet101 backbone network and incorporating a multi-head attention mechanism (MHA), the method enhances the focus on regions of interest in welding defect images, thereby improving the ability to extract defect features. This approach reduces the chances of missed detections and false positives. By analyzing the effects of different optimizers and dynamic factors on recall (M) and precision (N), the optimal optimizer and dynamic factor are identified. The algorithm is validated on the JPEGWD dataset. Experimental results show that the recall (M) reaches 0.9634, representing a 4.88% improvement, and the precision (N) reaches 0.9875, representing a 3.81% improvement. The mean average precision (mAP) is 0.9817, reflecting a 5.98% improvement. These results effectively enhance the recognition rate of welding defects and reduce the probability of missed detections and false positives.

## KEYWORDS

Convolutional neural networks; Multi-head attention mechanism; Welding defects; ResNet101

## 1. INTRODUCTION

Most steel structures require welding, and the fusion zone is a weak link in the weld joint. This area is characterized by uneven melting, and defects in the weld, such as cold cracks and reheat cracks, often originate here. In welded structures, besides visible external defects, there are many defects that are difficult to detect visually, such as cracks, porosity, slag inclusion, incomplete fusion, depressions, and holes, all of which have a significant impact on brittle fracture [1-2]. Welding is widely used in fields such as aerospace, machinery manufacturing, nuclear industry, energy transportation, and petrochemical industries [3]. It is a process that permanently joins welded parts to workpieces. Due to internal factors like process variations and external factors like humidity, defects may occur in welded components, significantly affecting product quality. Different welding defects require different solutions. Accurate classification of welding defects not only improves the efficiency of addressing these defects but also enhances the quality of welded parts. Therefore, effectively detecting and classifying defects that occur during welding is crucial for both product quality and manufacturing processes [4].

Traditional non-destructive testing (NDT) techniques for welding defects mainly include ultrasonic testing [5], radiographic testing [6], penetrant testing [7], and magnetic particle testing [8]. However, these methods have drawbacks, such as susceptibility to interference, high costs, and strong radiation. In recent years, with the rapid development of digital imaging technology, many scholars have proposed solutions for defect detection based on machine vision. For example, Pan Haihong et al. [9]

addressed the issues of low recognition accuracy, poor adaptability, and inefficiency in traditional convolutional neural networks (CNNs) for welding defect identification and classification, by proposing a DG-MobileNet welding defect identification model based on dilated convolution. Hu Jianfeng et al. [10] proposed an automatic identification method for welding defects based on image processing technology to identify defects in PE pipe electrofusion welding and to determine the category and grade of defects in ultrasonic images. Liu Mengxi et al. [11], addressing the challenge of high difficulty in defect recognition and classification for X-ray welds, improved and designed a deep CNN structure based on the typical CUDA-CONVNET convolutional neural network (CNN). Similarly, Gu Jing et al. [12], in response to the complexity and inefficiency of traditional algorithms in defect classification and recognition of X-ray welds, introduced the DenseNet algorithm based on deep learning, and dynamically enhanced the data.

To address the issues of low detection and identification rates, inaccurate classification, and missed or false detections in welding defect detection, this paper proposes a welding defect classification method based on convolutional neural networks. By modifying the ResNet101 backbone network, this method enhances the model's focus on input features by adding attention mechanism modules before the 2nd, 11th, 23rd, and 92nd layers, allowing the model to automatically learn regions of interest, effectively reducing the influence of redundant information, improving feature representation capabilities, and enhancing model performance on complex datasets, thereby increasing the identification rate of welding defects. Additionally, the Bottleneck structure unit is improved by replacing the 3x3 convolution kernel with a 3x1 and a 1x3 convolution kernel while maintaining the same receptive field, reducing the number of parameters. Finally, a Dropout layer is added before the fully connected layer to improve model accuracy and prevent overfitting.

## 2. CONVOLUTIONAL NEURAL NETWORKS

A Convolutional Neural Network (CNN) is a type of deep learning algorithm, classified as a feedforward neural network, and it performs exceptionally well in image processing, computer vision, and pattern recognition [13]. Its distinctive feature is the use of convolution operations to capture local features of the input data, characterized by parameter sharing and spatial invariance. CNNs are composed of multiple layers, including convolutional layers, pooling layers, and fully connected layers. The convolutional layer is the core part of a CNN. Each convolutional layer consists of several convolutional kernels, each performing convolution operations on a local region of the input data. These operations allow the kernels to extract local features from the input data and generate feature maps. Multiple kernels in the convolutional layer can capture different features such as edges, textures, and corners. Pooling layers are used to reduce the spatial dimensions of feature maps, thereby decreasing the number of parameters in the model. The most common pooling operation is max pooling, which selects the maximum value from a local region as the pooling result for that area. Pooling helps to retain the primary features while reducing detailed information, improving the model's robustness and computational efficiency. Fully connected layers convert the output feature maps from the convolutional and pooling layers into final classification or prediction results. Each neuron in a fully connected layer is connected to all neurons in the previous layer, integrating and classifying features. A simple structural diagram of a Convolutional Neural Network is shown in Figure 1.

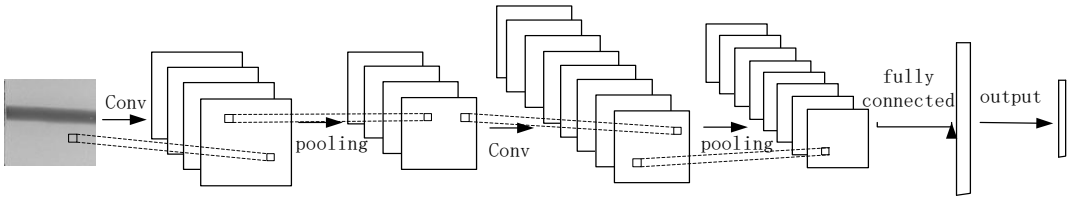



**Figure 1.** Simple Structural Diagram of a Convolutional Neural Network

# 3. CNN NETWORK MODEL CONSTRUCTION

## 3.1. ResNet101 Convolutional Neural Network Architecture

The traditional ResNet101 network consists of 5 convolutional groups, each containing one or more convolution operations, and performs downsampling in each group to reduce the size of the feature maps by half. In the 2nd convolutional group, max pooling with a stride of 2 is used; while in the other 4 convolutional groups, downsampling is achieved through convolutions with a stride of 2. Stage 1 includes 3 residual units and 3 Bottleneck modules, with a total of 256 feature maps. These modules use 64 kernels of size 1x1, 64 kernels of size 3x3, and 256 kernels of size 1x1. Stage 2 includes 4 residual units and 4 Bottleneck modules, with a total of 512 feature maps. These modules use 128 kernels of size 1x1, 128 kernels of size 3x3, and 512 kernels of size 1x1. Stage 3 includes 23 residual units and 23 Bottleneck modules, with a total of 1024 feature maps. These modules use 256 kernels of size 1x1, 256 kernels of size 3x3, and 1024 kernels of size 1x1. Stage 4 includes 3 residual units and 3 Bottleneck modules, with a total of 2048 feature maps. These modules use 512 kernels of size 1x1, 512 kernels of size 3x3, and 2048 kernels of size 1x1. A simple structural diagram of ResNet101 is shown in Figure 2.

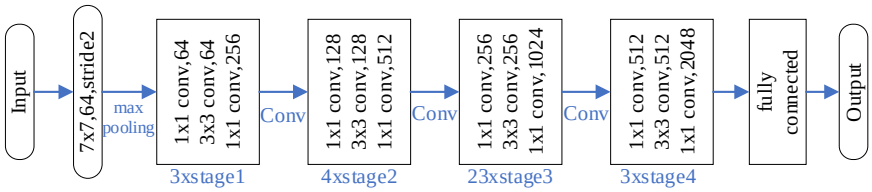



**Figure 2.** Simple Structural Diagram of ResNet101

## 3.2. Improved ResNet101 Convolutional Neural Network Structure

An attention mechanism module is added before each convolutional group in the traditional ResNet101 network. This enhances the model's focus on input features, allowing it to automatically learn the regions of interest, reduce the impact of redundant information, improve feature representation, and enhance the model's performance on complex datasets. Additionally, the inclusion of the attention mechanism can improve the model's precision and perceptual range. In the traditional ResNet101 model, features are extracted at different levels, but each feature map is added or connected with equal weight. By utilizing the attention mechanism, the weight of each feature map can be dynamically adjusted, enabling the model to focus more on the feature maps that contribute more to the decision-making process for a specific task. This enhances the model's perceptual range and representation capability. In addition to adding the attention mechanism, the 3x3 convolutional kernels in Stage 1, Stage 2, Stage 3, and Stage 4 are replaced with a 3x1 convolutional kernel and a 1x3 convolutional kernel. These have the same receptive field but reduce the number of parameters by approximately 33%, thus lowering the computational load. To avoid overfitting, a Dropout layer is added before the fully connected layer in ResNet101. During training, the Dropout layer randomly sets the output of some neurons to zero with a certain probability, reducing the dependency between neurons. Specifically, some neurons are randomly deactivated during training, while all neurons' outputs are retained during prediction. By randomly deactivating neurons, the model does not overly rely on specific neurons, thereby reducing overfitting. Finally, the fully connected layer is modified to meet the category number required by this experiment. The improved ResNet101 network structure is shown in Figure 3.

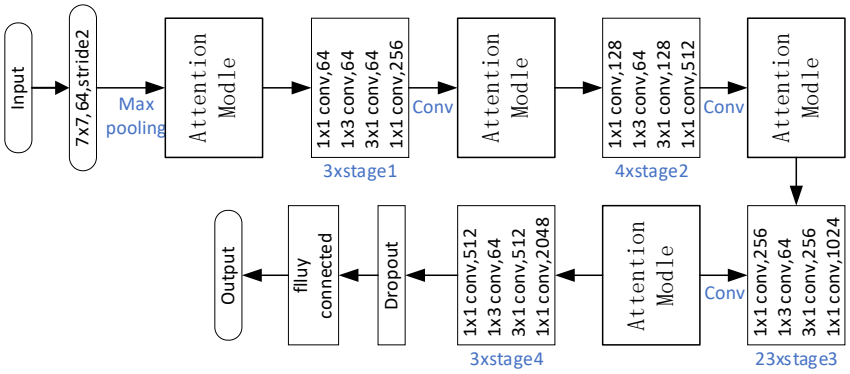



**Figure 3.** Improved ResNet101 Network Structure

The structure of the improved Bottleneck unit is shown in Figure 4. As can be seen from the figure, the input data has two branches to choose from. In one branch, the input data first undergoes a 1x1 convolutional operation to reduce the number of channels. Feature normalization helps to normalize each batch of input data, standardizing the data to a distribution with a mean of 0 and a variance of 1, which aids in reducing the issues of vanishing and exploding gradients, and increases the model's adaptability to data of different scales. The *ReLU* activation function can effectively alleviate the vanishing gradient problem and help the network converge more quickly. The middle layer uses a 1x3 convolutional kernel followed by a 3x1 convolutional kernel without changing the number of channels. Finally, a 1x1 convolutional kernel is used to increase the number of channels. The other branch does not perform any data processing, only normalization, which enhances the model's expressive capability, and then the data is output.

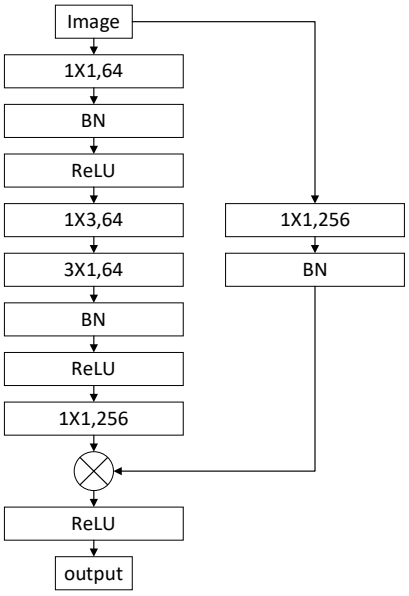



**Figure 4.** Improved Bottleneck Unit Structure

The mathematical expression of the *ReLU* activation function is shown in Equation (1). The graph of the *ReLU* activation function is shown in Figure 5.

$$f(x) = \max(0, x) \tag{1}$$

Where $x$ is the input, and $f(x)$ is the output.

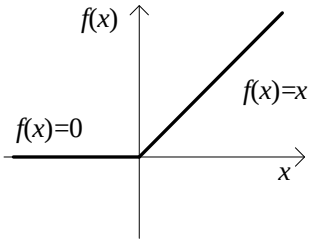



**Figure 5.** Graph of the ReLu Activation Function

$$\mathrm{Re}\,Lu = \begin{cases} x & x > 0 \\ 0 & x < 0 \end{cases} \tag{2}$$

### 3.3. Attention Module

The attention mechanism [14] can focus more on task-relevant features among the input information, reducing the focus on irrelevant aspects, thereby improving task efficiency and accuracy [15]. Based on different Q, K, and V, there are various types of attention mechanisms, such as soft attention [16], hard attention [17], and multi-head attention, among others. In this work, we employ Multi-Head Attention (MHA), a type of multi-head attention model, also known as the multi-head attention mechanism, which enhances attention to different features by extending the attention mechanism to multiple heads. MHA takes three vectors as input: query (Q), key (K), and value (V). For a given query, MHA performs a weighted sum of the keys, with the weights determined by the similarity between the query and the keys. The resulting weighted sum is then multiplied by the value vector to produce the output. The computation of MHA can be represented by the following formula:

$$MultiHead(Q,K,V) = Concat(h_1,...,h_n)W^O \tag{3}$$

In the formula, $n$ represents the number of heads, $h_i$ represents the output of the $i$ head, and $W^O$ is the output transformation matrix. The output of each head can be expressed as:

$$h_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \tag{4}$$

In the formula, $W_i^Q$, $W_i^K$, $W_i^V$ are the query, key, and value transformation matrices for the $i$-th head, respectively. The Attention function is the attention computation function. The structure of the Multi-Head Attention mechanism is shown in Figure 6.

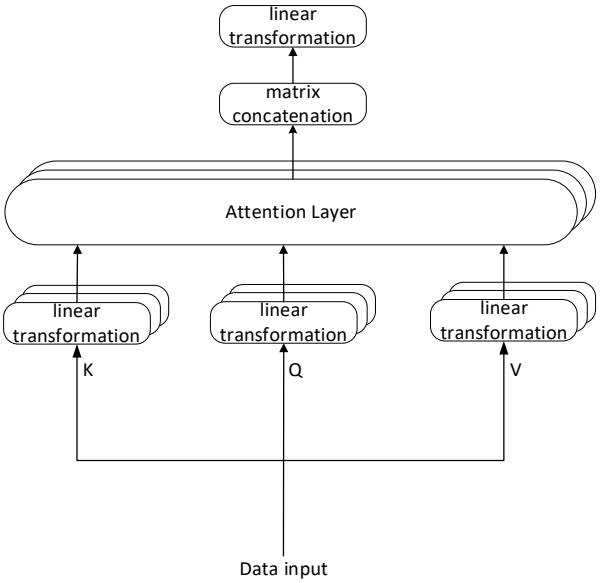



**Figure 6.** Structure of the Multi-Head Attention Mechanism

# 4. RESULTS AND ANALYSIS

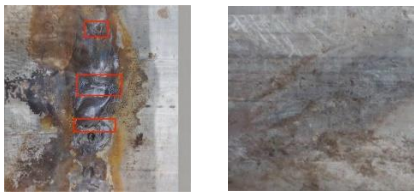
## 4.1. Experimental Environment Configuration

The experiment was conducted on hardware with an Nvidia Tesla T4 GPU, which has a power consumption TDP of 75W and supports DirectX version 12. The system is equipped with 16GB of RAM. The deployment environment was Python 3.9, running on Windows 10. The deep learning framework used was PyTorch, with CUDA version 12.0.

## 4.2. Experimental Sample Data and Hyperparameter Settings

The dataset used in this study is the JPEGWD, a specialized dataset for welding defect detection. The defects are categorized into four classes: concavity, holes, burrs, and no defects, with a resolution of 416×416. Data augmentation on the existing data produced 2,052 images. Among them, 1,368 images were selected as the training set, including 1,368 images of concavity, 328 images of holes, 400 images of burrs, and 240 images without defects. A validation set of 342 images was selected, including 100 images of concavity, 82 images of holes, 100 images of burrs, and 60 images without defects. A test set of 342 images was also selected, including 100 images of concavity, 82 images of holes, 100 images of burrs, and 60 images without defects. The welding defect data were divided into batches of 8 images for preprocessing. The preprocessed input welding defect images were resized to 224x224 and eventually passed into a 1x1x2048 fully connected layer. The learning rate (lr_scheduler) was set to 0.001, momentum to 0.9, and weight decay coefficient (weight_decay) to 1e-5.

Concavity Holes



Burrs No Defects

**Figure 7.** Sample Images of Welding Defects

### 4.3. Evaluation Metrics

To evaluate the performance of the classification model and better assess the detection results and recognition effectiveness, this study introduces Recall (M) and Precision (N) [18]. Recall refers to the ratio of the number of samples correctly predicted as positive by the model to the actual number of positive samples. It measures the model's ability to identify positive cases. The higher the recall, the more accurately the model can identify positive cases. The mathematical formula for recall is as follows:

$$M = \frac{TP}{(TP + FN)} \tag{5}$$

In this context, TP (True Positive) represents the number of samples correctly detected as having welding defects, while FN (False Negative) represents the number of samples that actually do not have welding defects but were incorrectly detected as having defects.

Precision also known as positive predictive value, is one of the metrics used to evaluate the performance of a classification model. It measures the proportion of samples predicted as positive that are truly positive. The mathematical formula for precision is as follows:

$$N = \frac{TP}{(TP + FP)} \tag{6}$$

In the formula, FP (False Positive) represents the number of samples that actually have welding defects but were incorrectly detected as defect-free.

The mathematical formula for the mean Average Precision (mAP) is as follows:

$$AP = \int_0^1 P(R)dR \tag{7}$$

$$\mathrm{m}AP = \frac{\sum_{i=1}^{n} AP_i}{C} \tag{8}$$

In the formula, AP represents the average precision for each category, and C is the number of categories.

The mathematical formula for the cross-entropy loss function is as follows:

$$H = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij}\log\left(\hat{y}_{ij}+\delta\right) \tag{9}$$

In the formula, N is the number of samples, C is the number of categories, $y_{ij}$ is 1 if the i-th sample belongs to the j-th category, and 0 otherwise. $y_{ij}$ is the predicted probability of the j-th category in the model's probability distribution. δ is a small positive number used to avoid taking the logarithm of zero.

### 4.4. The Impact of Different Optimizers and Dynamic Factors on Recall and Precision

To avoid missing welding defects and achieve relatively high recall and precision, we observed the performance of the ResNet101 network while keeping hyperparameters like learning rate and weight parameters constant. In order to achieve high recall and precision, we experimented with different optimizers and dynamic factors during the training process of the network model. Figure 8 shows the impact of different optimizers and dynamic factors sss on recall, while Figure 9 shows their impact on precision. From the results of these two figures, it can be seen that using different optimizers and dynamic factors leads to different trends in precision and recall. When the dynamic factor is equal to 3 and 4, the Stochastic Gradient Descent (SGD) optimizer achieves the highest recall, outperforming the Adamax (Adaptive Moment Estimation with Infinity Norm) and Adagrad (Adaptive Gradient) optimizers. When the dynamic factor is equal to 4, the SGD optimizer achieves the highest precision, outperforming the Adamax and Adagrad optimizers. Therefore, considering overall performance, SGD with a dynamic factor of 3 is selected as the optimal optimizer.

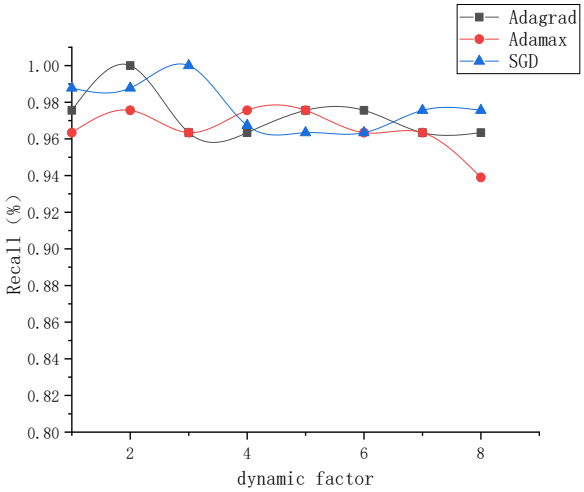



**Figure 8.** The impact of different optimizers and dynamic factors on recall

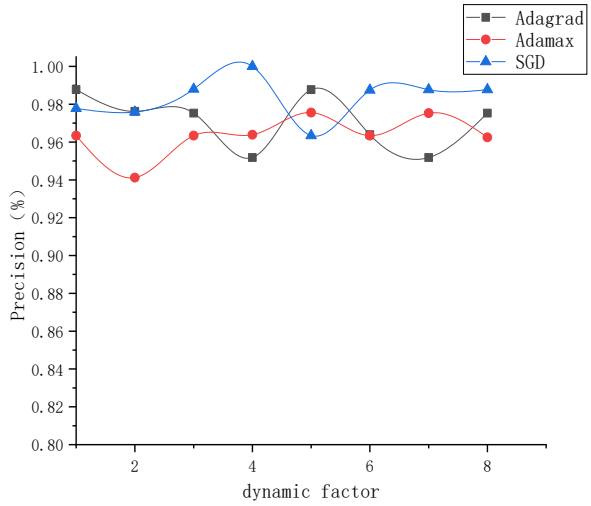



**Figure 9.** The impact of different optimizers and dynamic factors on precision

To better observe whether the model exhibits overfitting, Figure 10 presents the accuracy of the training and validation sets. As can be seen from the figure, the trends of the training and validation sets are generally consistent, indicating that overfitting did not occur.

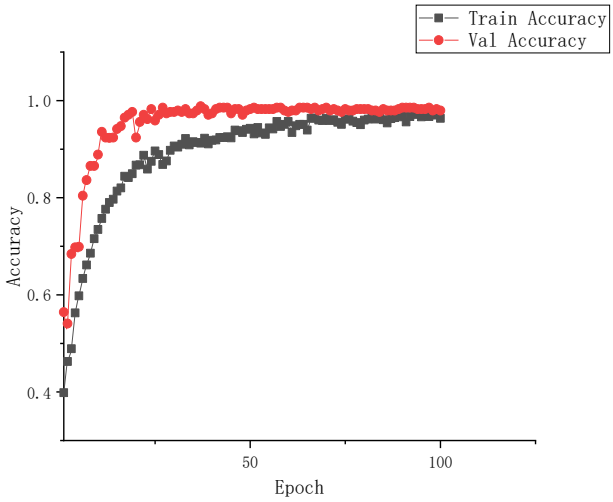



**Figure 10.** Accuracy of the training set and validation set

The training loss values of the improved algorithm compared to other classifier models are shown in Figure 11. As indicated by the figure, the improved model has a lower loss value on the training set compared to the original classifier models. Training loss is the target that the model aims to minimize during training through optimization algorithms (such as gradient descent). A lower loss value indicates that the model's predictions are closer to the actual labels on the training set, reflecting better fit during the training phase.

Figure 12 shows the validation loss values of the improved algorithm compared to other classifier models. As indicated by the figure, the improved model's loss value decreases more rapidly and is lower compared to the original classifier models. The loss value is an indicator of the discrepancy between the model's predictions and the actual labels; a lower loss value signifies that the model's predictions are closer to the actual labels on the validation set, indicating better model performance. Therefore, it is concluded that the improved model performs better on the validation set.

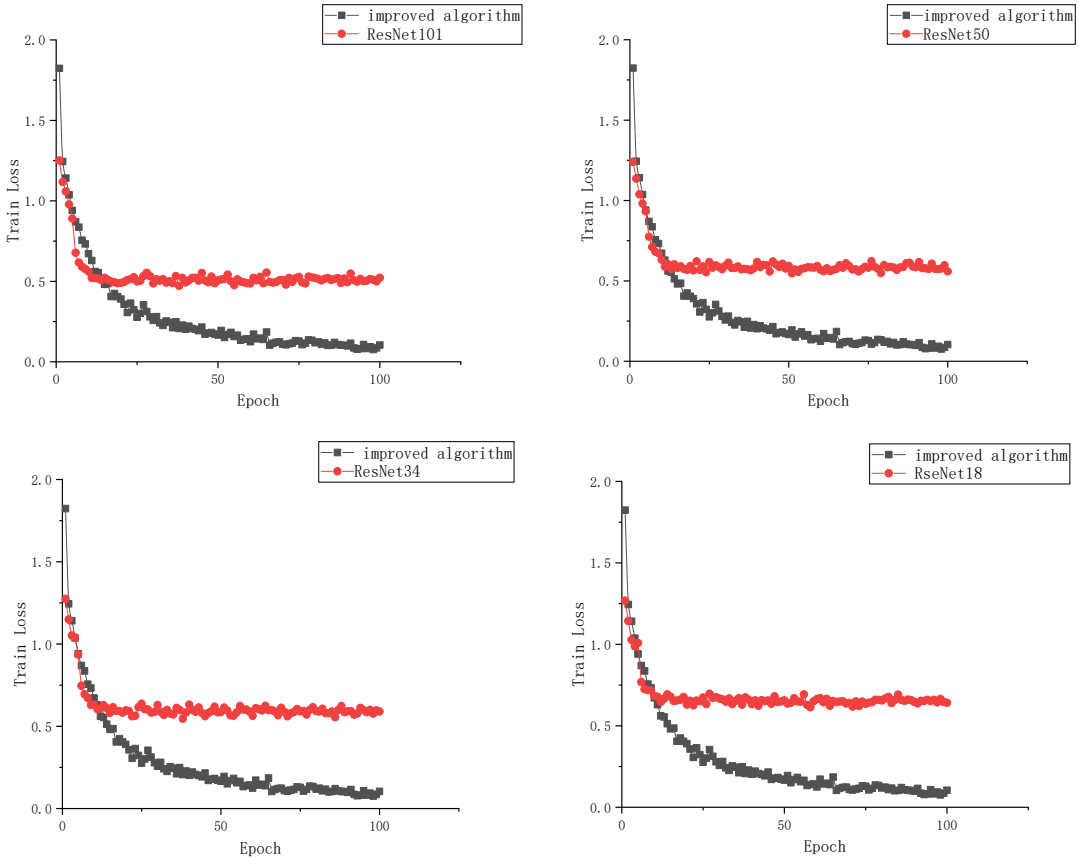



**Figure 11.** Comparison of training loss values between the improved model and different models

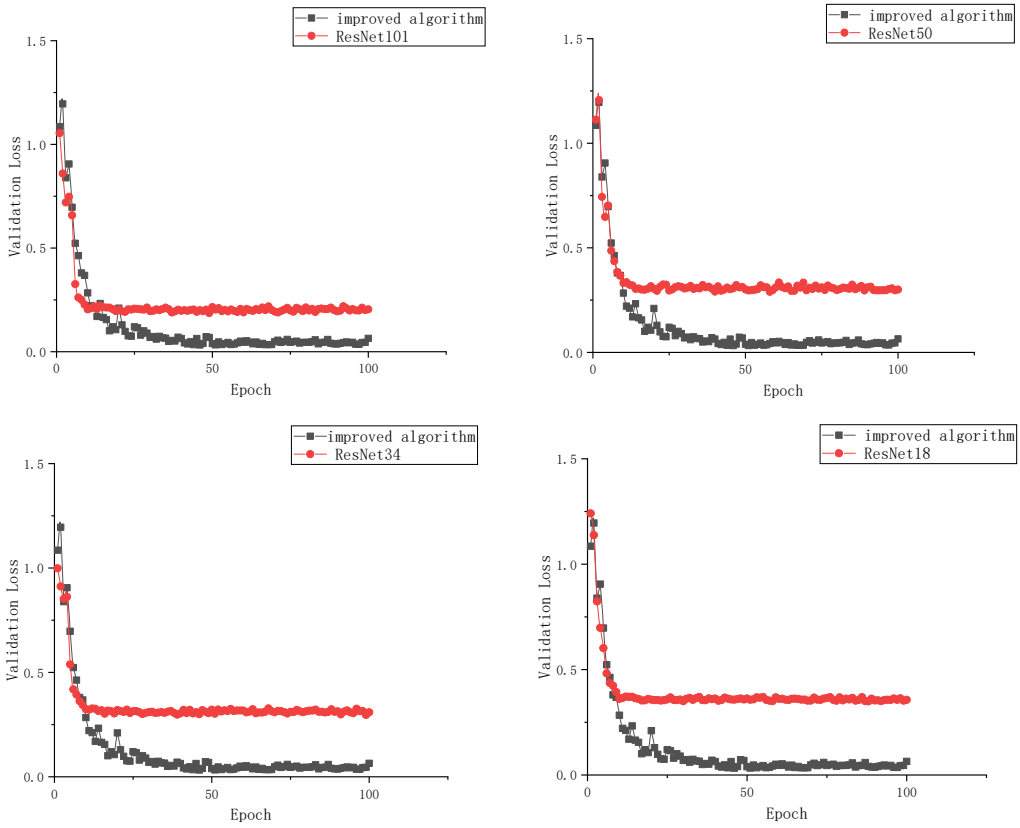



**Figure 12.** Comparison of validation loss functions between the improved model and different models

### 4.5. Comparison Experiments of Different Network Models

The results of comparing the improved ResNet101 network model classifier with the original ResNet18, ResNet34, ResNet50, and ResNet101 network model classifiers are shown in Table 1. From Table 1, it can be seen that the improved algorithm achieves the highest recall, precision, and mAP values. The recall and precision are 13.34% and 10.73% higher than those of the original ResNet18, 10.97% and 5.42% higher than ResNet34, 19.05% and 10.61% higher than the original ResNet50, and 4.88% and 3.81% higher than the original ResNet101. Additionally, the accuracy AP values for each class are the highest, and the mAP is 5.98% higher than that of the original ResNet101 classifier. Therefore, Table 1 shows that the improved ResNet101 network model exhibits better overall performance in weld defect detection compared to the other four models.

**Table 1.** Comparative experimental results

| network model | AP | | | | Recall | Precision | mAP |
|---|---|---|---|---|---|---|---|
| | Concavity | Holes | Burrs | No Defects | | | |
| ResNet18 | 0.8312 | 0.8187 | 0.7728 | 0.9287 | 0.7927 | 0.8667 | 0.8378 |
| ResNet34 | 0.8581 | 0.8318 | 0.8334 | 0.9238 | 0.8537 | 0.9333 | 0.8618 |
| ResNet50 | 0.8243 | 0.8387 | 0.8095 | 0.9677 | 0.8902 | 0.8690 | 0.86 |
| ResNet101 | 0.9152 | 0.9351 | 0.8811 | 0.9561 | 0.9146 | 0.9494 | 0.9219 |
| study improves | 0.9929 | 0.9722 | 0.9615 | 1 | 0.9634 | 0.9875 | 0.9817 |

## 5. CONCLUSION

This paper proposes an improved ResNet101 algorithm to address issues such as low weld defect recognition rates, inaccurate classification, and missed and false detections. By modifying the ResNet101 backbone network and adding an MHA (Multi-Head Attention) mechanism at the beginning of each stage, the model focuses more on task-relevant features in the input information and reduces attention to irrelevant features, thereby improving weld defect recognition rates. The Bottleneck structure unit has been improved by replacing the 3x3 convolutional kernel with a 3x1 and a 1x3 convolutional kernel, which reduces the number of parameters while maintaining the same receptive field. Additionally, a Dropout layer has been added before the fully connected layer to enhance model accuracy and prevent overfitting. Experimental results show that compared to the original ResNet101 model, the improved ResNet101 model achieves a recall rate of 0.9634, an increase of 4.88%; a precision rate of 0.9875, an increase of 3.81%; and a mAP value of 0.9817, an increase of 5.98%. However, for online real-time detection of weld defects, further learning about the internal structure of weld defects and other classifiers is needed, and deeper research is required on precise defect localization and related issues.

## REFERENCES


[1] Li Yajuan, Wang Juan, Jiang Qinglei, et al. *Analysis and Countermeasures of Welding Defects* [M]. 3rd ed. Beijing: Chemical Industry Press, 2018: 3-81.

[2] Zhang Fulu, Wang Runzhi, Dong Ce, et al. Quantitative characterization of internal initiation fatigue crack growth rate [J]. Heat Treatment of Metals, 2019(S01): 611-614.

[3] Ye Zhilong. Research on Automatic Detection Methods of Welding Defects Based on Image Processing [D]. Nanjing University of Aeronautics and Astronautics, 2013.

[4] Wang Fei., Zhu Jianjiang. Application of Machine Vision in Welding Defect Detection Under Complex Environments [J]. Welding Technology, 2017, 46(05): 127-133.

[5] Liang Hongbao, Wang Lixun, Liu Lei. Current Status and Development of Non-Destructive Testing Technology for Pressure Vessels [J]. Petroleum Machinery, 2007, No.336(02): 54-57.

[6] Peng Shuaijun., Wang Kai, Wu Haifen, et al. Application of Radiographic Testing Technology in Dimensional Measurement [J]. Nondestructive Testing, 2007, No.272(11): 679-681.

[7] Liu Bin. Inspection of Surface Defects in Artificial Joint Metal Parts Using Penetrant Testing Technology [J]. Materials Engineering, 2009, No.312(05): 73-75.

[8] Kang Ziwen, Duan Yixiong, Wang Long. Research on Intelligent Magnetic Particle Testing Technology for Integrated Wheelsets [J]. Locomotive & Rolling Stock Technology, 2023, No.359(01): 47-50.

[9] Pan Haihong, Li Songting, Chen Lin, et al. Welding Seam Defect Recognition Method Based on Improved DG-MobileNet Model [J]. Modular Machine Tools and Automatic Manufacturing Techniques, 2023(08):127-130.

[10] Hu Jianfeng, Zhou Jiale, Wang Huifeng, et al. Research and Application of an Automatic Recognition Method for Electrofusion Welding Defects [J]. Journal of East China University of Science and Technology (Natural Science Edition), 2022, 48(04):526-533.

[11] Liu Mengxi, Ju Yongfeng, Gao Weixin, et al. Research on X-ray Welding Seam Defects Using Deep Convolutional Neural Networks [J]. Sensors and Microsystems, 2018, 37(05):37-39+43.

[12] Gu Jing, Wang Qiwen, Zhang Min, et al. Welding Seam Defect Detection and Recognition Based on DenseNet Network [J]. Sensors and Microsystems, 2020, 39(09):129-131.

[13] Qiao Li, Zhao Erdun, Liu Junjie, et al. Research on Workpiece Defect Detection Method Based on CNN [J]. Computer Science, 2017, 44(S2):238-243.

[14] Luo Shunhua, Wang Zhenlei, Wang Xin. Soft Measurement Modeling Based on Attention Mechanism with Multi-head CNN-LSTM [J]. Control Engineering, 2022, 29(10):1821-1828.

[15] Jin Guoqiang. Research on End-to-End Rolling Bearing Fault Diagnosis Algorithm Under Complex Working Conditions Based on Deep Learning [D]. University of Science and Technology of China, 2021.

[16] TAO SHEN, TIANYI ZHOU, GUODONG LONG, et al. Reinforced Self-Attention Network: a Hybrid of Hard and Soft Attention for Sequence Modeling[C]. //27th International Joint Conference on Artificial Intelligence and 23rd European Conference on Artificial Intelligence: IJCAI-ECAI 2018, Stockholm, Sweden, 13-19 July 2018, Volume 6 of 8: Curran Associates, Inc., 2018:4345-4352.

[17] JOAN SERRA, DIDAC SURIS, MARIUS MIRON, et al. Overcoming Catastrophic Forgetting with Hard Attention to the Task[C]. //35th International Conference on Machine Learning: ICML 2018, Stockholm, Sweden, 10-15 July 2018, volume 10 of 13: Curran Associates, Inc., 2018:7225-7242.

[18] Yao Zongwei, Yang Hongfei, Hu Jiyong, et al. Rail Surface Defect Detection Method Based on Machine Vision and Convolutional Neural Networks [J]. Journal of the China Railway Society, 2021, 43(04):101-107.