Welding in the World
https://doi.org/10.1007/s40194-024-01759-9

RESEARCH PAPER

WeldNet: a lightweight deep learning model for welding defect recognition

Rongdi Wang[1] · Hao Wang[1] · Zhenhao He[1] · Jianchao Zhu[1] · Haiqiang Zuo[1]

Received: 1 January 2024 / Accepted: 18 March 2024

Abstract

Weld defect detection is an important task in the welding process. Although there are many excellent weld defect detection models, there is still much room for improvement in stability and accuracy. In this study, a lightweight deep learning model called WeldNet is proposed to improve the existing weld defect recognition network for its poor generalization performance, overfitting, and large memory occupation, using a design with a small number of parameters but with better performance. We also proposed an ensemble-distillation strategy in the training process, which effectively improved the accuracy rate and proposed an improved model ensemble scheme. The experimental results show that the final designed WeldNet model performs well in detecting weld defects and achieves state-of-the-art performance. Its number of parameters is only 26.8% of that of ResNet18, but the accuracy is 8.9% higher, while achieving a 24.2 ms inference time on CPU to meet the demand of real-time operation. The study is of guiding significance for solving practical problems in weld defect detection, and provides new ideas for the application of deep learning in industry. The code used in this article is available at https://github.com/Wanglaoban3/WeldNet.git.

Keywords Convolutional neural networks · Weld defect · Model ensemble · Knowledge distillation

Nomenclature

TIG	Tungsten inert gas
CNN	Convolutional neural network
SVM	Support vector machine
CPU	Central processing unit
GPU	Graph processing unit
FCN	Fully connected network
HDR	High dynamic range

1 Introduction

Welding is a technique to join two or more metal workpieces together and is widely used in industry. There are different types of welding such as manual welding, arc welding, laser welding, and plasma welding [7, 23, 34]. The classification and selection of welding depend on factors such as material, application scenario, and cost. Although welding is a common technology, the welding process and the generation of welding defects are complex and diverse, and the generation of defects may lead to weak and easily fractured welded connections, which can lead to serious accidents and serious human and material losses. Therefore, the detection and evaluation of welding defects are of great importance, and the development of a portable and rapid defect detection method is even more important.

Welding defects are defects that may occur during the welding process, including unfused, porosity, slag, overburning, cracks, and burrs [30, 31], and the presence of these defects may lead to accidents and cause great losses. Therefore, for welded parts, welding quality evaluation and welding defects detection become critical.

Traditional methods of weld defect detection include visual inspection, X-ray inspection, ultrasonic inspection, eddy current inspection, and magnetic particle inspection [8, 24]. In industrial weld defect detection, visual inspection is widely applied by technicians due to its low cost, high reliability, and minimal need for additional equipment. However,

Recommended for publication by Commission V - NDT and Quality Assurance of Welded Products.

✉ Haiqiang Zuo
zhqupc@upc.edu.cn

[1] College of New Energy, China University of Petroleum (East China), Qingdao, Shandong 266580, People's Republic of China

Published online: 04 April 2024

visual inspection is heavily reliant on the experience of the technicians. As work intensity increases, the efficiency of manual inspection decreases significantly. Therefore, the automation of weld defect detection is imperative. In 2015, Gao et al. [9] proposed a reliance on active visual systems to obtain the characteristics of the molten pool morphology during the welding process, by analyzing the relationship between the molten pool morphology and the stability of the welding process to judge the classification of welding defects, while Chen et al. [4] used a high-speed infrared camera to photograph the molten pool during the welding process and analyze the thermal imaging images obtained. Huang et al. [17] designed an eddy current orthogonal axial probe for eddy current detection of carbon steel plate welds, and the experiment proved that the probe can effectively detect the effect of unevenness of the weld surface on the lifting-off effect. Silva et al. proposed a segmented analysis of time-of-flight diffraction ultrasound for flaw model, which is able to identify the type of defects in the welding process by analyzing ultrasonic signal fragments [27]. Du et al. [6] proposed a fusion of background subtraction and grayscale wave analysis for defect classification by analyzing weld X-ray.

Although most of the above methods achieve a better detection effect, there are still many problems with these methods. For example, they require specialized equipment and skills, are difficult to implant, do not generalize well, require a lot of time and labor costs for on-site debugging, are not very intelligent, etc. To overcome these problems, more and more researchers are using deep learning techniques for welding defect detection [13, 16].

Deep learning techniques are a neural network-based machine learning method with a high level of automation and the ability to process complex data, and they excel in image and video processing. Convolutional neural network is a commonly used neural network structure in deep learning, which is mainly used in computer vision fields such as image recognition classification and target detection. It achieves feature extraction, abstraction, and classification of images through multiple layers of computation and learning such as convolutional layer, pooling layer, and fully connected layer, and relies on the backpropagation algorithm [25] to update the parameters in the model. Common CNN models include AlexNet, VGG, ResNet, and MobileNet [11, 14, 18, 28], all of which are widely used for various vision tasks including defect detection. Compared to relying on traditional image processing, where features are extracted manually for image preprocessing and then entered into SVM classification, deep learning-based models can achieve better performance by simple training without human intervention and without complex tuning of parameters. Bacioiu et al. [2, 3] proposed a series of CNN models for TIG welding of SS304 and aluminum 5083 materials, respectively, which are capable of classifying images by simply inputting images taken by industrial cameras of the weld seams during the welding process. On their publicly available SS304 dataset, a maximum accuracy of 71% was achieved for a six-class defect classification task. Ma et al. [22] developed a CNN network for detecting and classifying defects generated during fusion welding of galvanized steel sheets, and he achieved 100% recognition accuracy using AlexNet, VGG networks combined with data enhancement and migration learning. Golodov [10], Yu [35], and others also achieved automatic classification and recognition of defective parts by optimizing existing CNNs for weld image segmentation. Xia et al. [33] designed a CNN model for defect classification during TIG welding, which was optimized for the interpretation of the CNN model. In addition to classification using images alone, a multi-sensor combined with a deep learning approach has also been used to accomplish defect classification. Li et al. [21] designed a triple pseudo-twin network by simultaneously inputting images, sound, and current–voltage signals of the molten pool during the welding process and using the network to automatically fuse the information collected by different sensors to finally output the classification results.

Although the remarkable performance achieved by these deep learning-based weld defect detection models, they still encounter some common issues. Firstly, the models are susceptible to overfitting. Due to the typically small scale of the weld defect dataset, deep learning models trained on it tend to overfit, resulting in low robustness. Secondly, the speed of model inference needs improvement. Most existing models heavily rely on specific GPU devices for real-time operation due to their large parameter sizes and computational complexity. However, such specialized GPU equipment is often unavailable in practical industrial environments where CPUs or other devices are more commonly utilized. The former issue presents a prevalent challenge in the industrial application of deep learning models, as specific production environments make it difficult to collect data on a large scale, leading to neural networks that are prone to overfitting and have poor generalization performance. Model ensemble serves as a common approach to mitigate overfitting. For instance, in the study conducted by [5], multiple feature extractors were employed for feature extraction, and the features from various branches were weighted to enrich the feature information, thereby enhancing the model's robustness. Similar approaches can be found in [20, 26]. However, this practice often results in a significant increase in model parameters and computational requirements. Therefore, inspired by these algorithms [5, 12], we have designed a training strategy that combines model ensemble and distillation. This strategy involves training multiple models together during training and then performing knowledge distillation after the completion of multi-model training. As a result, we obtain a compact neural network with high performance. Additionally, we propose a focal ensemble optimization strategy in this process to enhance the effectiveness of model ensembling. To improve model inference speed, we have also developed a lightweight deep neural network named WeldNet. In order to validate the effectiveness of our algorithms and models, extensive experiments were conducted on publicly available datasets. The experimental results demonstrate that our proposed model achieves a smaller number of parameters and higher performance at a faster speed compared to state-of-the-art methods. The contributions of this paper are as follows:

(1) We design a lightweight network called WeldNet, which exhibits superior performance and faster inference speed in the field of weld defect recognition compared to mainstream models.
(2) We propose a novel model training method for weld defect recognition models, called the ensemble-distillation strategy, which significantly improves model performance without introducing any additional computational overhead during model deployment. This approach allows for more efficient operation of the model during deployment while achieving remarkable performance enhancements.
(3) Our proposed model exhibits significantly higher accuracy and faster inference speed compared to other models in performance comparisons on public datasets. Furthermore, it is capable of real-time execution on CPU devices, which holds great significance for the future development of weld defect recognition models.

Our steps are as follows: first, we introduce the implementation algorithms and ideas involved in Section 2, design the relevant experiments and reveal the specific experimental details in Section 3, analyze the experimental results and discuss them in Section 4, and conclude in the last section.

2 Methodology

This section first introduces the dataset in this paper, followed by a description of the algorithms and principles used.

2.1 Dataset

This paper uses the TIG Al5083 weld defect public dataset by Bacioiu et al. [2] in 2019. The dataset was obtained by tracking the TIG Al5083 welding process using an HDR camera facing directly towards the weld pool to acquire real-time images. During the data collection process, weld pool images depicting “good welds” were initially captured. Subsequently, welding parameters were systematically adjusted to generate various defect types, introducing diversity in defects. For instance, in a series of welding experiments, the welding current was progressively reduced until a “lack of fusion” was achieved. Finally, the collected dataset was divided into training and testing sets at a ratio of 4:1. It is important to note that while both the training and testing sets contain the same types of defects, they originate from welding processes with different parameter combinations to assess the model’s robustness and generalization capabilities comprehensively. The dataset comprises six categories, namely “good welds,” “Burn through,” “Contamination,” “Lack of fusion,” “Misalignment,” and “Lack of penetration,” totaling 33,254 images, with the specific number of images per category detailed in Table 1. Some samples are shown in Fig. 1. The dataset is currently available at https://www.kaggle.com/datasets/danielbacioiu/tig-aluminium-5083.

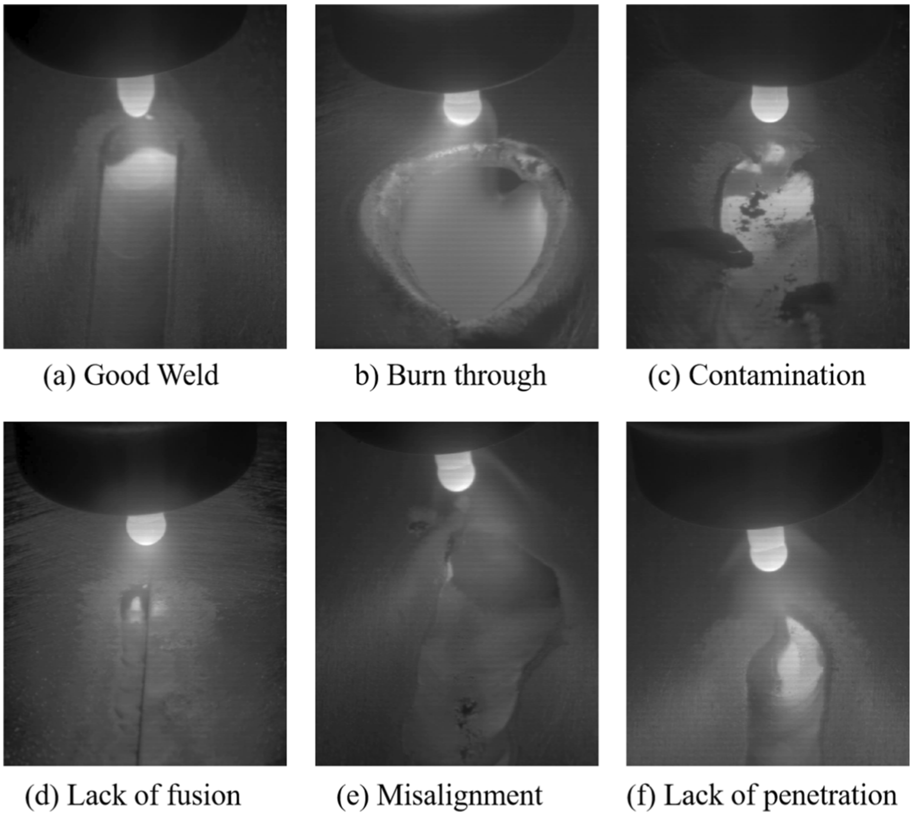
2.2 Convolutional neural networks (CNN)

In recent years, due to the continuous development of artificial intelligence technology, CNN is widely used in the image field as a neural network with excellent performance. For example, it was featured in the early handwritten digit recognition [19] and the ImageNet competition [18] in 2012. The main features of CNN are its ability to automatically learn useful features from the original data and its good hierarchical structure to extract higher-level features layer by layer. The most important ones in CNN are convolutional layer and pooling layer. Convolutional layer can effectively extract local features in images, and pooling layer can reduce the number of fused features and computation, highlight important features while filtering minor features, and increase the overall generalization ability of the model, by setting a convolution kernel of fixed size and using the convolution kernel to slide over the image to calculate the product sum of the convolution kernel and the corresponding region, where stride represents the step size of each convolution kernel move.

In image classification, CNN and FCN (fully connected networks) are usually used in combination. After a large

Table 1 Number of categories in the training and test sets

Label	Number of samples	
	Train	Test
Good weld	8758	2189
Burn through	1783	351
Contamination	6325	2078
Lack of fusion	4028	1007
Misalignment	2953	729
Lack of penetration	2819	234
Total	26,666	6588



(a) Good Weld

b) Burn through

(c) Contamination

(d) Lack of fusion

(e) Misalignment

(f) Lack of penetration

Fig. 1 Samples of six-classes defect

number of convolution and pooling layers, the size of the feature map gradually becomes smaller from the original map. It is transformed into vector form by the flatten operation, and finally, the probability of each class is output after one or more fully connected layers. FCN uses the input vector to multiply with the weight matrix, adds a bias term, and finally outputs a result through the activation function. The calculation formula is as follows: $\mathbf{y} = f(\mathbf{wx} + \mathbf{b})$. Take the example of a_n in the figure (bias omitted): $\mathbf{a_n} = (\mathbf{w_{n1}x_1} + \mathbf{w_{n2}x_2} + \mathbf{w_{n3}x_3} + \mathbf{w_{n4}x_4}) + \mathbf{b}$.

The parameter updating process of the neural network can be expressed as Eq. 1.

$$w_{ij} = w_{ij} - \alpha \frac{\partial L(w)}{\partial w_{ij}} \tag{1}$$

where w_{ij} denotes the weight connecting the ith neuron to the jth 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. In backpropagation, the partial derivatives of each parameter can be calculated by the chain rule, and then updated in this way when updating the parameters. In this way, the above process is repeated until the loss function converges, and a better network parameter is obtained.

Figure 2 shows a classical CNN called ResNet18, which consists of several convolutional and pooling layers and a fully connected layer. An input RGB image of size 224×224 is continuously convolved and pooled to obtain a 512×1×1 feature map, and finally, a flatten operation and a fully connected layer are used to obtain the probability of each category. In the convolutional layer, the parameter after conv is the number of output channels, *k* represents the convolutional kernel size, and *s* represents stride. Here, the padding parameter setting is omitted, and the padding of each convolutional layer is *k*/2 and rounded down. The batch normalization layer and activation function layers are omitted after each convolutional layer. AdapativeAvgpool is an adaptive average pooling layer that converts the input feature map to the target size; here, we change 7×7 after the pooling layer to 1×1.

2.3 WeldNet

The style of WeldNet proposed in this paper is slightly similar to ResNet, which is also a neural network composed of CNN+FCN, but the modules in it are optimized. The details are shown in Fig. 3. To ensure fairness in subsequent experimental comparisons, the model and ResNet18 have an equivalent number of model layers and similar channel configurations. Differing from ResNet18, the model employs consecutive convolutional layers with a stride of 2 in the first three layers to reduce the size of feature maps, thereby

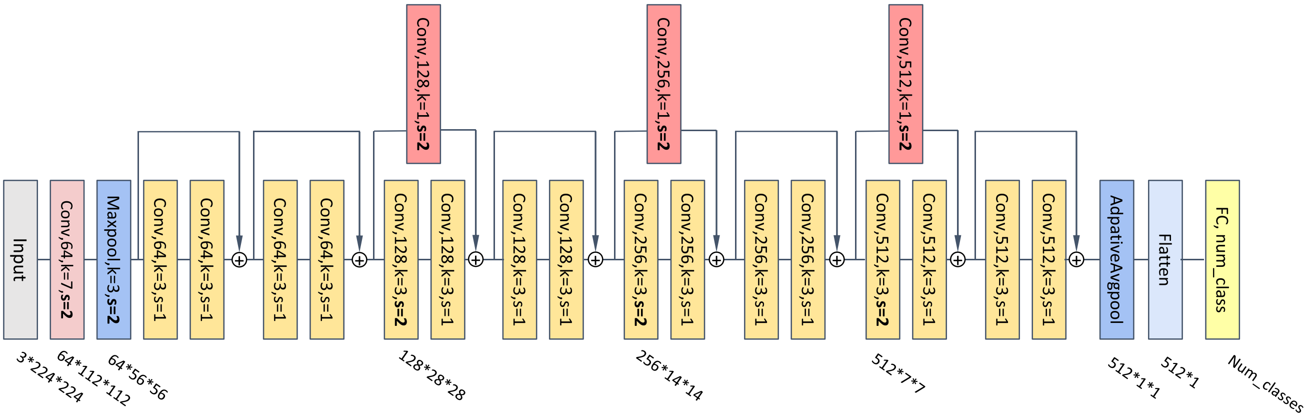


Fig. 2 ResNet18

decreasing computational complexity. Additionally, the specially designed WeldBlock replaces the BasicBlock in the model. Within the WeldBlock, the input feature map undergoes two branches: one consisting of a 3 × 3 convolutional layer followed by a 1 × 1 convolutional layer and the other comprising a 3×3 average pooling layer followed by a 1×1 convolutional layer, with the results of the two branches being added together. In the pooling branch, the input features first pass through a 3×3 average pooling layer and then through a 1×1 convolutional layer. In comparison to the BasicBlock in ResNet18, the 3×3 average pooling layer in the WeldBlock lacks learnable parameters and provides feature maps with a larger receptive field for the 1×1 convolutional layer, resulting in fewer parameters and a reduced risk of overfitting. Furthermore, this approach mitigates the issue of detail loss during the convolution process when the stride of the 1×1 convolution layer is set to 2. The 32 after the first convolutional layer represents the number of output 32 channels. The e parameter in WeldBlock is a scaling factor that controls the module in which the output channel of the convolution layer is the input channel × e, and s stands for stride.

2.4 Ensemble-distillation strategy

Model ensemble is a common means to improve model stability and generalization performance. The principle is to use the same dataset to train multiple models simultaneously and average the output results to obtain the final result [15]. Since each model is initialized with different parameters, the order of each sample in the traversed dataset is different, and the form of each data augmentation is different, making each model trained have different degrees of generalization, and also avoiding the problem of unstable training process due to model initialization and dataset order in the process of training a single model. Combining the outputs of all models on average usually results in better performance than a single model [32]. The principle is shown in Fig. 4.

However, the use of model ensemble results in a significant increase in inference time and memory consumption, which is not conducive to practical weld defect assessment in real-world environments. Therefore, we propose the utilization of knowledge distillation to reduce the model size. Knowledge distillation is a technique for transferring knowledge from a large neural network to a small neural network. Unlike model compression, the goal of knowledge distillation is not to reduce the size and computational burden of a model, but to transfer knowledge from a large neural network (called teacher network) to a small neural network (called student network), thereby improving the accuracy, generalization, and robustness of the student model [1]. Usually, a teacher network is first trained on the dataset until convergence, then the teacher network is used to predict values for each sample as labels for the training of the student network instead of the real labels produced in the dataset, and finally,

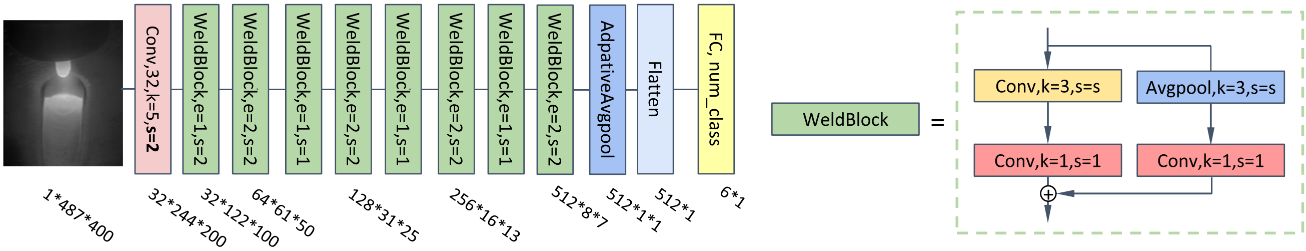
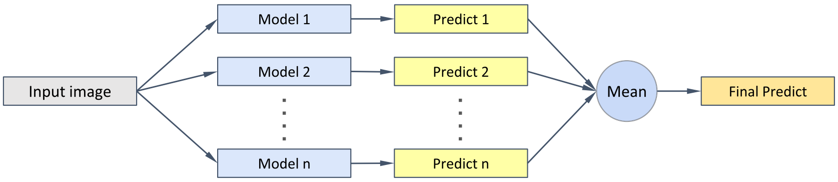


Fig. 3 WeldNet (ours)

Fig. 4 Model ensemble



the student network is trained so that the student network learns the knowledge of the teacher network while avoiding the high memory and computing power overhead of the teacher network [29]. The principle is shown in Fig. 5.

In this study, we propose a novel training method for weld defect detection models, which involves first training multiple single models and then using the trained models as teacher networks for knowledge distillation. This method effectively improves model performance without increasing additional parameters or computational costs. For specific implementation details, please refer to Algorithm 1.

Algorithm 1 Framework of ensemble-distillation strategy for our system.

```
Input: $m_1, m_2, ..., m_n$: number of $n$ single model as teacher model; $m_s$: student model; $k$: training iterations; $F_e$: model ensemble function; $L$: loss function;
Output: Training set $x$, $l_t$
for $i = 1$ to $n$ do                                    ▷ train each teacher model
    for $j = 1$ to $k$ do
        $y_p \leftarrow m_i(x)$
        $l_s \leftarrow L(y_p, y_t)$
        Update Parameters $m_i$
    end for
end for
for j=1 to k do                                          ▷ train student model
    $y_p^{(1)}, y_p^{(2)}, ..., y_p^{(n)} \leftarrow \{m_1, m_2, ..., m_n\}(x)$
    $y_e \leftarrow F_e(y_p^{(1)}, y_p^{(2)}, ..., y_p^{(n)})$        ▷ get model ensemble pesudo label
    $y_s \leftarrow m_s(x)$
    $l_s \leftarrow L(y_s, y_e)$
    Update Parameters $m_s$
end for
return $m_s$
```

2.5 Focal ensemble

Among the existing model ensemble techniques, we think it is too simple and brutal to directly add up the predicted values of each model. Considering the model prediction, when the confidence level of a model prediction is high, we should trust the prediction of that model more. Therefore, we proposed a new combination to amplify the high confidence prediction values, which we call focal ensemble, which is calculated as Eq. 2.

$$y = \frac{1}{N}(\sum_{i=1}^{N} y_i^k)^{\frac{1}{k}} \tag{2}$$

where y represents the final predicted value, y_i represents the individual model predicted value, N represents the number of models, and k represents the exponential weighting factor of ensemble.

3 Experiment design and details

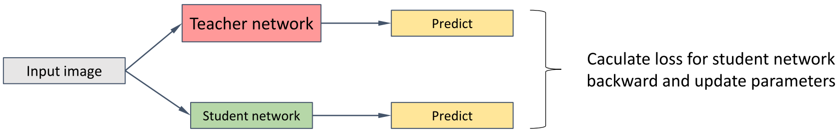
In this section, we describe the details of model training, model evaluation criteria, and the selection of hyperparameters in the experiments.

3.1 Design of experiment

To verify the effectiveness of the proposed WeldNet optimization strategy, the performance metrics of each individual model were first compared. Subsequently, to demonstrate the effectiveness of the proposed model ensemble strategy, comparative experiments were conducted between single models and various ensemble strategies. Finally, the model ensembles and knowledge-distilled models were compared with each individual model for the final assessment. All models are thoroughly trained on the TIG Al5083 training set until convergence, tested after each training epoch, and the best performance metric on the test set is taken as the final result.

Note that since the dataset is a single-channel grayscale map and the final output is a 6 classification probability, we set the first convolutional layer input channel of the above CNN model to 1 and the last fully connected layer output channel to 6. During the training process, we designed a combination of data augmentation means for this dataset, including rotation, random horizontal and random crop, random brightness, and contrast adjustment. For the training set,

Fig. 5 Knowledge distillation


Input image
Teacher network
Student network
Predict
Predict
Caculate loss for student network backward and update parameters

the images were first randomly cropped to 600×731, randomly rotated by −30 to 30°, randomly flipped horizontally, and randomly adjusted brightness and contrast, and finally, the images were scaled down to 400×487 size and input into our model. For the test set, we only scaled the image to 400×487 and input it into the model. Figure 6 shows our means of data augmentation.

We choose cross entropy loss for the loss function of the training model classification, which is calculated as Eq. 3.

$$L(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{ic}\log(p_{ic}) \tag{3}$$

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

3.2 Experiment details

The hyperparameters involved in the experiments are the learning rate of the optimizer and the number of models in the model ensemble. We empirically set the SGD optimizer learning rate to 0.01, and the learning rate becomes 0.001 after running 15 epochs, and the momentum parameter is set to 0.9, while five models are trained simultaneously in the model ensemble for a total of 25 epochs, the weight k in Eq. 2 is set to 2. The language environment we use is Python, and we use the PyTorch library for all model training and testing, and the hardware we use is i7-11700k and rtx-3080ti.

3.3 Performance metric

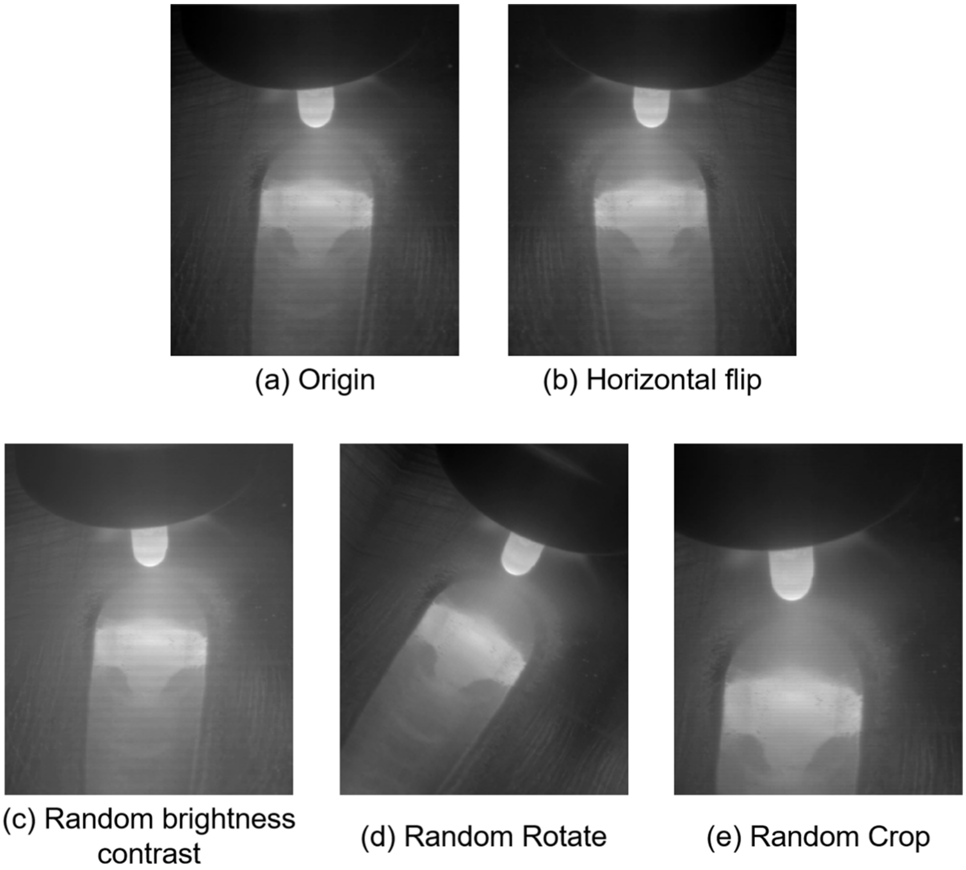
The weld defect classification in this paper is an image multi-classification task, so in this paper, accuracy, precision, recall, and F1-score are used as evaluation metrics for model performance, and the formula is calculated as Eqs. 4, 5, 6, and 7.

$$Accuracy = \frac{1}{n}\sum_{i=1}^{N} TP_i \tag{4}$$

$$Precision = \frac{1}{N}\sum_{i=1}^{N}\frac{TP_i}{TP_i + FP_i} \tag{5}$$

$$Recall = \frac{1}{N}\sum_{i=1}^{N}\frac{TP_i}{TP_i + FN_i} \tag{6}$$

Fig. 6 Data augmentation



$$F1 - score = \frac{1}{N}\sum_{i=1}^{N}\frac{2 \times Precision_i \times Recall_i}{Precision_i + Recall_i} \tag{7}$$

where n denotes the total number of samples, N denotes the total number of classes, TP_i represents the total number of samples in which the model correctly predicts the $i-th$ class as that class, FP_i represents the total number of samples in which the model incorrectly predicts another class as $i-th$ class, and FN represents the number of samples in which the model incorrectly predicts the $i-th$ as another class.

Furthermore, to effectively assess the operational speed of the new model, we employed model inference time as the metric for evaluating its speed performance. This refers to the duration required for the model to process an image. It is important to note that all speed testing experiments were carried out using an Intel i7 11700k processor (CPU).

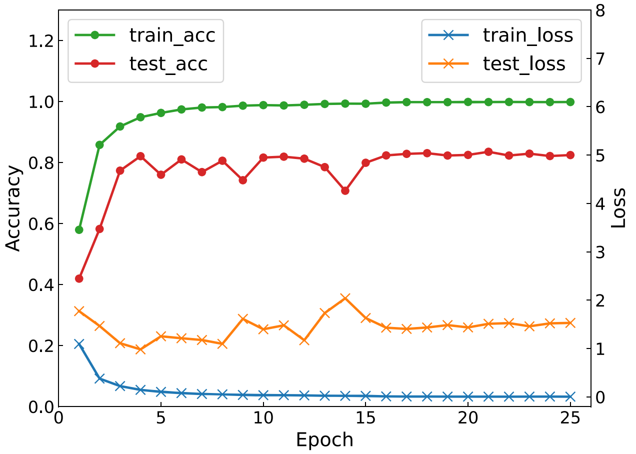


Fig. 8 Accuracy and loss during WeldNet training on single model training

4 Results and discussion

In this section, we first compare the performance of different models, followed by a detailed discussion.

4.1 Single model

The results of single-model training are shown in Figs. 7 and 8. Figure 7 shows that when single-model training, the more the number of model parameters, the worse the performance tends to be instead, which is contrary to our previous knowledge. In Fig. 8, we find that the loss of the model on the training set rapidly decreases and converges to 0 while the loss and accuracy on the test set start to oscillate during training. This indicates that the model quickly learns the classification task on the training set, but as the training loss gets smaller, the model starts to overfit and the performance instead decreases slightly and gradually stabilizes. At the same time, we found that models with more number of parameters are more prone to overfitting, and models with moderate parameters perform better with the number of parameters instead. Our carefully designed WeldNet is able to outperform ResNet18 and other networks. The detailed experimental results are shown in Table 2.

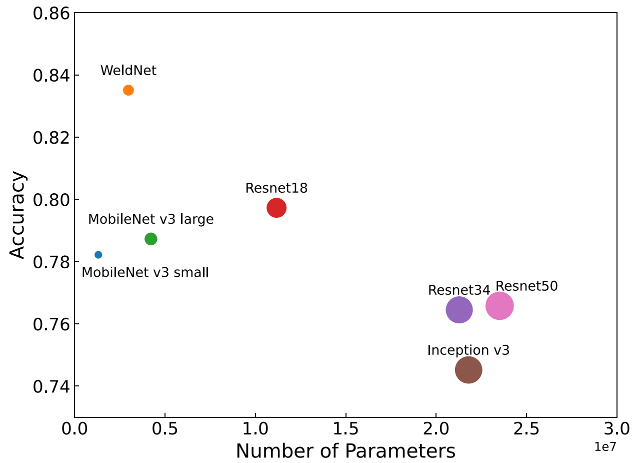


Fig. 7 Accuracy and number of parameters of different models on single model training

4.2 Model ensemble

The results of model ensemble training are shown in Figs. 9 and 10. When using multiple models integrated training, it can be found that multiple models, due to different initialization during training, different order of datasets, and different data enhancement methods, combine them together to significantly improve generalization ability and stability, and have a good response to the overfitting that occurs during single model training, improving the performance very considerably. This inspires us to improve model performance in weld defect recognition tasks not only by modifying the structure and parameters of the model, but also by optimizing the existing training methods, which can significantly improve the model performance. All of our proposed focal ensemble results are better than the existing mean ensemble, which proves that focal ensemble is effective (Table 3).

4.3 Knowledge distillation

We understand from the above experiments that model ensemble can improve model performance, but it also inevitably increases computing time and memory consumption, so we use knowledge distillation to let the labels generated in model ensemble be learned by a single model. We used the WeldNet network trained in the focal ensemble approach as the teacher network, a single WeldNet as the

Table 2 Accuracy and number of parameters of different models. Bolded values represent the best results

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Parameters(10^7)
MobileNet v3 small	78.2	69.3	73.2	69.7	**0.13**
MobileNet v3 large	78.7	69.8	73.7	70.2	0.42
ResNet18	79.3	70.4	74.4	70.8	1.12
ResNet34	76.5	67.9	71.9	68.4	2.13
Inception v3	74.6	66.0	69.8	66.4	2.18
ResNet50	76.6	68.0	71.9	68.4	2.35
WeldNet (ours)	**83.5**	**75.1**	**78.4**	**74.9**	0.30

student network, and the loss function selected as CE Loss, and finally obtained a WeldNet with an accuracy of 88.2%, which is only 0.9% lower than that of the teacher network.

The experimental results show that the model accuracy hardly decreases when the number of model parameters is reduced to a single model size due to knowledge distillation, confirming that the distilled student network is effectively learning the knowledge of the teacher network. Through Table 4, we can find that the model accuracy can be improved significantly by designing an efficient lightweight network WeldNet, combined with the training strategy of ensemble distillation. Compared with a single ResNet18 model, our WeldNet+FE+KD accuracy is 8.9% higher, and the number of participants is only 26.8% of it.

4.4 Discussion

In a single-model experiment, we observed that models with a large number of parameters tended to exhibit relatively poor performance, contrary to the common belief in deep learning. Typically, a larger number of parameters in a model signifies stronger fitting ability and often leads to better performance. However, this notion is based on the availability of abundant and diverse datasets. When dataset size is limited and sample diversity is restricted, as in the case of the weld defect recognition dataset discussed in this paper, models with a larger number of parameters are more prone to overfitting, resulting in faster convergence during training but poorer performance during testing. The optimally designed WeldNet demonstrated the best performance among single models, validating the effectiveness of the proposed improvements. A comparison with the similarly small-parameter MobileNet v3 indicated that model overfitting is not only related to model parameter count but also to model structure.

In the experiments involving model ensembling, our proposed multi-model ensembling strategy significantly improved model performance, demonstrating that model ensembling is an effective approach for enhancing model robustness, especially in scenarios where single models are prone to overfitting. During single-model training, susceptibility to noise can lead to considerable fluctuations, whereas training and prediction using ensembled models effectively reduce the impact of such bias and noise, resulting in overall higher performance. In our proposed focal ensemble, adjusting the exponential weighting coefficients enables models with better performance to make a greater contribution, further reducing the influence of noise and improving overall prediction results.

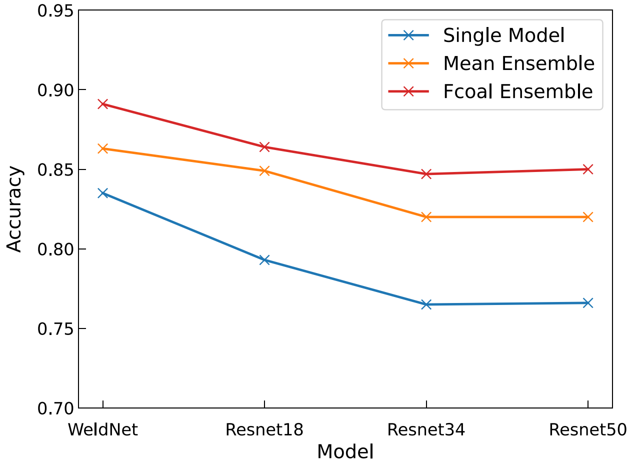


Fig. 9 Comparison of different training methods

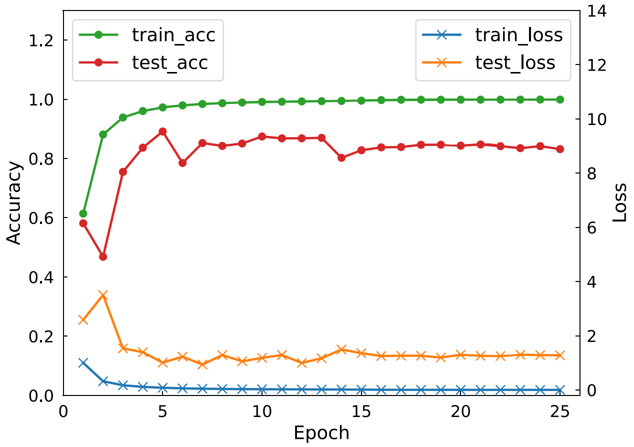


Fig. 10 Accuracy and loss during WeldNet focal ensemble training

Table 3 Effect of different training methods on model accuracy, precision, recall, and F1-score. Bolded values represent the best results

Method	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Single model	83.5	75.1	78.4	74.9
Mean ensemble	87.6	81.2	82.6	79.2
Focal ensemble (ours)	**89.1**	**83.8**	**84.6**	**81.8**

Although model ensembling effectively enhances model performance, it noticeably reduces operational speed, as evidenced by the comparison between WeldNet+FE and WeldNet+FE+KD in Table 4. Therefore, by employing the proposed model ensembling and knowledge distillation strategies, single models can possess the knowledge and detection capabilities of multiple models in situations with lower parameter counts. Furthermore, from Table 4, it can be observed that most models run slowly on the CPU platform and fail to meet the speed requirements in industrial settings, especially in the absence of specific running devices such as GPU devices. Therefore, it is evident that the majority of existing models heavily rely on computational resources for their operations. In industrial inspection scenarios, there are also many tasks where models suffer from overfitting. The proposed WeldNet demonstrates certain advantages in addressing these challenges, suggesting potential extensions of WeldNet to a broader range of tasks in the future.

5 Conclusion

In this paper, we designed WeldNet, a customized lightweight detection network designed specifically for identifying defects in welding operations. This network demonstrates improved robustness on a small-scale dataset. Additionally, to enhance the generalization performance of the model while maintaining low computational and parameter complexity, we propose an ensemble-distillation training method that effectively combines multiple models without introducing additional computational burden during model deployment. This innovative technique not only surpasses the performance of existing models significantly but also addresses the challenge of model dependence on specific equipment in weld defect detection. Experimental results on the TIG AL5083 dataset confirm the superior detection accuracy of our approach compared to all existing networks. Compared to the single ResNet18 model, our WeldNet+FE+KD achieves an accuracy increase of 8.9%, precision increase of 11.8%, recall increase of 8.8%, and F1-score increase of 10.8%. Additionally, the parameter count of our model is only 26.8% of the ResNet18 model. These findings hold immense significance for future research and exploration in this field.

Although our proposed network has achieved relatively high performance and can be practically deployed in industrial scenarios, there are still certain limitations that need to be addressed. One such limitation is its heavy reliance on a large number of manually annotated labels for training. This dependency on labeled data poses challenges in terms of scalability and cost-effectiveness. To overcome this bottleneck, future research directions will focus on exploring semi-supervised or unsupervised learning approaches to further optimize the weld defect detection model. These approaches aim to leverage unlabeled data or utilize limited labeled data more efficiently, reducing the reliance on extensive manual annotation. In addition, the model proposed in this paper has only been tested for defect identification in TIG welding scenarios. Due to limitations such as experimental equipment and time, we have not conducted verification experiments in a wider range of welding scenarios. In the future, we plan to extend defect identification to include a broader range of welding processes. By adopting these methodologies, we aim to improve the scalability, generalization, and cost-efficiency of the model, making it more practical and applicable in real-world industrial settings.

Table 4 Accuracy and the number of parameters of the models obtained by different training methods

Method	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Parameters (10^7)	Inference time (ms)
MobileNet v3 small	78.2	69.3	73.2	69.7	**0.13**	27.4
MobileNet v3 large	78.7	69.8	73.7	70.2	0.42	80.6
ResNet18	79.3	70.4	74.4	70.8	1.12	59.9
WeldNet	83.5	75.1	78.4	74.9	0.30	**24.2**
WeldNet+FE	**89.1**	**83.8**	**84.6**	**81.8**	1.49	75.8
WeldNet+FE+KD (ours)	88.2	82.2	83.2	80.0	0.30	**24.2**

FE represents focal ensemble, and KE represents knowledge distillation. Bolded values represent the best results

Funding This study is supported by Shandong Province Key Research and Development Program, No. 2020CXGC011201.

Data availability Data will be made available on request.

Declarations

Competing interests The authors declare no competing interests.

References

1. Allen-Zhu Z, Li Y (2020) Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. arXiv:2012.09816 (2020)
2. Bacioiu D, Melton G, Papaelias M, Shaw R (2019) Automated defect classification of aluminium 5083 TIG welding using HDR camera and neural networks. J Manufac Process 45:603–613. https://doi.org/10.1016/j.jmapro.2019.07.020
3. Bacioiu D, Melton G, Papaelias M, Shaw R (2019) Automated defect classification of SS304 TIG welding process using visible spectrum camera and machine learning. NDT & E International 107. https://doi.org/10.1016/j.ndteint.2019.102139
4. Chen Z, Gao X (2014) Detection of weld pool width using infrared imaging during high-power fiber laser welding of type 304 austenitic stainless steel. Int J Adv Manufac Technol 74(9–12):1247–1254. https://doi.org/10.1007/s00170-014-6081-3
5. Chiaranai S, Pitakaso R, Sethanan K, Kosacka-Olejnik M, Srichok T, Chokanat P (2023) Ensemble deep learning ultimate tensile strength classification model for weld seam of asymmetric friction stir welding. Processes 11(2):434
6. Du D, Hou R, Shao J, Wang L, Chang B (2008) Real-time Xray image processing based on information fusion for weld defects detection. In: 17th world conference on nondestructive testing, Shanghai, China
7. Ericsson M (2003) Influence of welding speed on the fatigue of friction stir welds, and comparison with MIG and TIG. Int J Fatigue 25(12):1379–1387. https://doi.org/10.1016/s0142-1123(03)00059-8
8. Gao P, Wang C, Li Y, Cong Z (2015) Electromagnetic and eddy current NDT in weld inspection: a review. Insight-Non-Destructive Testing and Condition Monitoring 57(6):337–345
9. Gao X, Zhang Y (2015) Monitoring of welding status by molten pool morphology during high-power disk laser welding. Optik - Int J Light Electron Optics 126(19):1797–1802. https://doi.org/10.1016/j.ijleo.2015.04.060
10. Golodov VA, Maltseva AA (2022) Approach to weld segmentation and defect classification in radiographic images of pipe welds. NDT & E Int 127. https://doi.org/10.1016/j.ndteint.2021.102597
11. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
12. Hinton G, Vinyals O, Dean J (2015) Distilling the knowledge in a neural network. arXiv:1503.02531
13. Hou W, Zhang D, Wei Y, Guo J, Zhang X (2020) Review on computer aided weld defect detection from radiography images. Appl Sci 10(5). https://doi.org/10.3390/app10051878
14. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam HJapa (2017) MobileNets: efficient convolutional neural networks for mobile vision applications. arXiv:1704.04861
15. Huang G, Li Y, Pleiss G, Liu Z, Hopcroft JE, Weinberger KQ (2017) Snapshot ensembles: train 1, get m for free. arXiv:1704.00109
16. Huang J, Zhang Z, Qin R, Yu Y, Li Y, Wen G, Cheng W, Chen X (2023) Residual swin transformer-based weld crack leakage monitoring of pressure pipeline. Welding in the World pp 1–13
17. Huang L, Liao C, Song X, Chen T, Zhang X, Deng Z (2020) Research on detection mechanism of weld defects of carbon steel plate based on orthogonal axial eddy current probe. Sensors (Basel) 20(19). https://doi.org/10.3390/s20195515. https://www.ncbi.nlm.nih.gov/pubmed/32993112. Huang, Linnan Liao, Chunhui Song, Xiaochun Chen, Tao Zhang, Xu Deng, Zhiyang eng 51807052/National Natural Science Foundation of China/ Switzerland Sensors (Basel). 2020 Sep 26;20(19):5515
18. Krizhevsky A, Sutskever I, Hinton GE (2017) ImageNet classification with deep convolutional neural networks. Commun ACM 60(6):84–90
19. Le Cun Y, Bottou L, Bengio Y (1997) Reading checks with multilayer graph transformer networks. In: 1997 IEEE International conference on acoustics, speech, and signal processing, IEEE 1:151–154
20. Li H, Li L, Chen X, Zhou Y, Li Z, Zhao Z (2024) Addressing the inspection selection challenges of in-service pipeline girth weld using ensemble tree models. Eng Failure Anal 156:107852
21. Li Z, Chen H, Ma X, Chen H, Ma Z (2022) Triple pseudo-siamese network with hybrid attention mechanism for welding defect detection. Mater & Design 217. https://doi.org/10.1016/j.matdes.2022.110645
22. Ma G, Yu L, Yuan H, Xiao W, He Y (2021) A vision-based method for lap weld defects monitoring of galvanized steel sheets using convolutional neural network. J Manufac Process 64:130–139. https://doi.org/10.1016/j.jmapro.2020.12.067
23. Mackwood AP, Crafer RC (2005) Thermal modelling of laser welding and related processes: a literature review. Optics & Laser Technol 37(2):99–115. https://doi.org/10.1016/j.optlastec.2004.02.017
24. Madhvacharyula AS, Pavan AVS, Gorthi S, Chitral S, Venkaiah N, Kiran DV (2022) In situ detection of welding defects: a review. Welding in the World 66(4):611–628. https://doi.org/10.1007/s40194-021-01229-6
25. Rumelhart DE, Hinton GE, Williams RJ (1986) Learning representations by back-propagating errors. Nature 323(6088):533–536
26. Say D, Zidi S, Qaisar SM, Krichen M (2023) Automated categorization of multiclass welding defects using the X-ray image augmentation and convolutional neural network. Sensors 23(14):6422
27. Silva LC, Simas Filho EF, Albuquerque MCS, Silva IC, Farias CTT (2020) Segmented analysis of time-of-flight diffraction ultrasound for flaw detection in welded steel plates using extreme learning machines. Ultrasonics 102:106057.https://doi.org/10.1016/j.ultras.2019.106057. https://www.ncbi.nlm.nih.gov/pubmed/31952796. Silva, Lucas C Simas Filho, Eduardo F Albuquerque, Maria C S Silva, Ivan C Farias, Claudia T T eng Netherlands Ultrasonics. 2020 Mar;102:106057. Epub 2019 Dec 11
28. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556
29. Tian Y, Wang Y, Krishnan D, Tenenbaum JB, Isola P (2020) Rethinking few-shot image classification: a good embedding is all you need? In: Computer Vision-ECCV 2020: 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XIV 16, Springer, pp 266–282
30. Tyystjärvi T, Virkkunen I, Fridolf P, Rosell A, Barsoum Z (2022) Automated defect detection in digital radiography of aerospace welds using deep learning. Welding in the World 66(4):643–671
31. Vilar R, Zapata J, Ruiz R (2009) An automatic system of classification of weld defects in radiographic images. NDT & E Int 42(5):467–476. https://doi.org/10.1016/j.ndteint.2009.02.004

32. Wang X, Kondratyuk D, Christiansen E, Kitani KM, Alon Y, Eban E (2020) Wisdom of committees: an overlooked approach to faster and more accurate models. arXiv:2012.01988
33. Xia C, Pan Z, Fei Z, Zhang S, Li H (2020) Vision based defects detection for keyhole TIG welding using deep learning with visual explanation. J Manufac Process 56:845–855. https://doi.org/10.1016/j.jmapro.2020.05.033
34. Yan J, Gao M, Zeng X (2010) Study on microstructure and mechanical properties of 304 stainless steel joints by TIG, laser and laser-TIG hybrid welding. Optics Lasers Eng 48(4):512–517. https://doi.org/10.1016/j.optlaseng.2009.08.009
35. Yu R, Kershaw J, Wang P, Zhang Y (2021) Real-time recognition of arc weld pool using image segmentation network. J Manufac Processes 72:159–167. https://doi.org/10.1016/j.jmapro.2021.10.019

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.