Proceedings of the 5th International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/121/2025.19844

Melt Pool Monitoring in Arc Additive Manufacturing Using Multi-branch Residual Networks

Long Shu[1,a,*], Jiale Wang[1,b]

[1]*School of Marine Engineering Equipment, Zhejiang Ocean University, Zhoushan, Zhejiang, China*
a. sl13966948200@163.com, b. wangjiale@zjou.edu.cn
**corresponding author*

Abstract: Wire arc additive manufacturing (WAAM) is a promising technology in aerospace, automotive, and marine industries due to its high efficiency and cost-effectiveness. However, the dynamic behavior of the melt pool significantly affects the forming quality and requires real-time and accurate monitoring. In this paper, a novel multi-branch residual network for melt pool monitoring is proposed, which addresses the limitations of traditional methods and single-branch networks. The multi-branch structure effectively extracts multi-scale features, while the residual module enhances feature fusion and training stability. The lightweight architecture ensures real-time processing in industrial scenarios. Experiments on a self-built dataset of 6,000 images of three melt pool states demonstrate the superior performance of the proposed method, achieving 95.1% accuracy, 94.2% precision, and 13.1 milliseconds inference time. Comparative and ablation studies confirm the significant contributions of the multi-branch structure and residual module, enabling robust and efficient monitoring under complex conditions.

Keywords: wire arc additive manufacturing, molten pool monitoring, multi-branch residual network, deep learning.

1. Introduction

Wire Arc Additive Manufacturing (WAAM) is an emerging metal additive manufacturing technology. It has been widely used in aerospace, shipbuilding and automotive industries due to its low cost, high material utilization rate and short manufacturing cycle[1][2]. Compared with traditional additive manufacturing processes such as powder bed fusion, WAAM is characterized by its higher forming efficiency and adaptability to large and complex components. However, this technology still faces significant process challenges in practical applications, especially the dynamic behavior of the molten pool has a decisive influence on the forming quality. The morphology, size and thermodynamic properties of the molten pool directly affect the forming accuracy, surface quality and mechanical properties of the parts[3]. Therefore, it is of great significance to realize real-time and accurate monitoring of the dynamic characteristics of the molten pool to improve process stability and part quality.

Melt pool monitoring technology has been widely studied in recent years. The monitoring methods mainly include visual monitoring, infrared thermal imaging and multi-sensor fusion technology[4]. Ebrahimi et al. studied the oscillation behavior of the molten pool and its relationship with process parameters and material properties through high-fidelity three-dimensional numerical simulation,

© 2025 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/121/2025.19844

providing theoretical support for process optimization[5]. Auccot et al. used synchrotron radiation experiments to directly observe the flow in the steel molten pool, revealing the influence of surface tension on flow velocity and turbulence, and providing new ideas for optimizing the flow inside the molten pool[6]. Xia et al. used deep learning methods to detect molten pool anomalies in the WAAM process, including humps and spatter[7]. Studies have shown that the convolutional neural network (CNN) model has an accuracy rate of more than 97% in classifying molten pool images, which can effectively improve monitoring capabilities.

With the development of artificial intelligence and deep learning technology, monitoring methods based on convolutional neural networks (CNN) have gradually attracted attention[8]. Deep learning models have shown excellent performance in complex pattern recognition tasks due to their powerful feature extraction capabilities. In particular, deep models such as residual networks (ResNet) have significantly improved the training efficiency and feature extraction capabilities of deep networks by introducing residual connection mechanisms[9]. However, most of the current deep learning research on molten pool monitoring is limited to a single-branch network structure, which has insufficient ability to capture multi-scale features and is difficult to fully characterize the complex dynamic behavior of the molten pool. In addition, the real-time and robustness of existing models are also difficult to fully meet the industrial needs of arc additive manufacturing.

In response to the above problems, this paper aims to propose a melt pool monitoring method based on a multi-branch residual network to overcome the shortcomings of traditional methods and single-branch networks. This method decomposes the melt pool monitoring task into multiple feature extraction subtasks by introducing a multi-branch structure, and each branch focuses on capturing features of different scales or dimensions (such as texture, boundaries or dynamic changes). At the same time, the introduction of the residual module can effectively enhance the feature fusion ability and training stability of the network, and significantly improve the ability to characterize the complex dynamic behavior of the melt pool. On this basis, this paper designs a lightweight network architecture to ensure the real-time and anti-interference capabilities of the model in actual industrial scenarios.

2. Melt Pool Monitoring Network Based on Multi-branch Residual

2.1. Overview of Network Design

The core of melt pool monitoring lies in accurately capturing multi-scale and multi-type features in images, such as texture details, boundary changes, dynamic behaviors, etc. This paper designs a deep learning network based on a multi-branch residual structure. The overall network architecture is shown in Figure 1. After the melt pool image is input, the basic features are extracted after preliminary convolution, and then enter the multi-branch module. Each branch extracts features of different scales or specific types, and then performs comprehensive processing in the feature fusion layer, and finally outputs the key feature map for monitoring.

2.2. Multi-branch module structure

In deep learning, the multi-branch structure processes different feature extraction tasks through parallel sub-networks, which can effectively improve the representation capability of complex input data. Assuming that the input melt pool image is X, its feature extraction process can be expressed as multiple independent functions f_i, namely:

$$F(X) = \sum_{i=1}^{n} f_i(X) \tag{1}$$

DOI: 10.54254/2755-2721/121/2025.19844

Wherein, $f_i(X)$ represents the feature extracted by the i th branch. Through this branching design, each branch network can focus on the feature extraction of a specific scale or type, thereby improving the robustness and expression ability of the overall network.

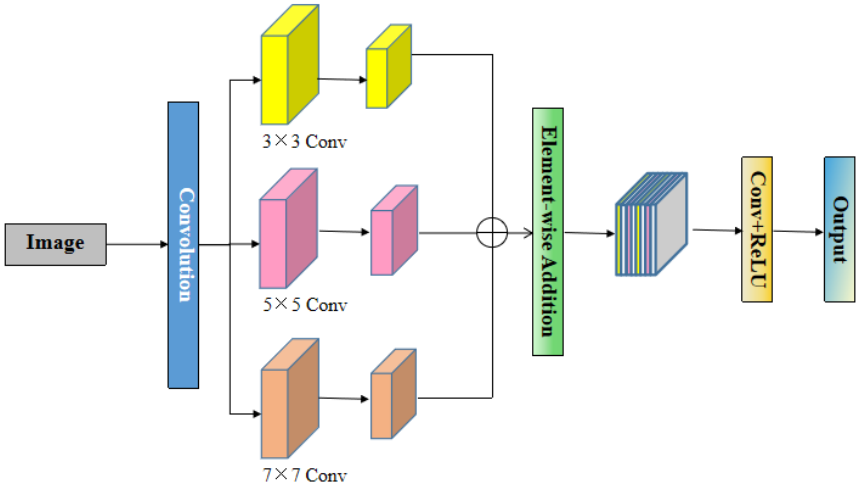

Figure 1: Melting pool state detection network based on multi-branch residual network

The melt pool image contains global features (such as overall contour, heat affected zone distribution) and local features (such as boundary texture, dynamic fluctuation), and a single network branch is often difficult to take into account these features at the same time. Therefore, this paper designs a multi-branch structure to achieve effective extraction and fusion of information of different scales through parallel branch networks in response to the multi-scale and multi-feature requirements in melt pool monitoring.

The global branch focuses on capturing the macro information of the melt pool image, such as the overall shape, the distribution of the heat affected zone, etc. This branch uses a large-size convolution kernel (7×7), whose receptive field covers most of the entire image, which helps to extract overall features. In addition, this branch uses a larger stride to further reduce the size of the feature map to reduce computational complexity. Specifically, for the input feature X, the feature $f_1(X)$ extracted by the global branch is expressed as:

$$f_1(X) = \mathrm{ReLU}\left(\mathrm{Conv}_{7\times7}(X)\right) \tag{2}$$

Where, $\mathrm{Conv}_{7\times7}$ represents a convolution operation with a convolution kernel size of 7×7, and ReLU is the activation function.

The goal of the local branch is to capture medium-scale features, such as texture and medium dynamic changes in the local area of the melt pool. This branch uses medium-sized convolution kernels (5×5), which can balance the details and range of feature extraction while retaining more local information. Compared with the global branch, the convolution kernel of the local branch has a smaller coverage area and is suitable for medium-scale feature extraction.

$$f_2(X) = \mathrm{ReLU}\left(\mathrm{Conv}_{5\times5}(X)\right) \tag{3}$$

The local branch captures features more finely and is an important supplement to the global branch information. The detail branch is mainly used to capture subtle features in the image, such as fine dynamics, small fluctuations or texture details at the edge of the melt pool. This branch uses small-sized convolution kernels (3×3) to capture local changes with a finer receptive field.

DOI: 10.54254/2755-2721/121/2025.19844

$$f_3(X) = \mathrm{ReLU}\left(\mathrm{Conv}_{3\times3}(X)\right) \tag{4}$$

Then the multi-branch module is fused, and the final output integrates the features extracted by different branches through the feature fusion layer. The fusion strategy adopts the element-by-element addition method to achieve the effective combination of global, local and detail features:

$$F_{\mathrm{multi}}(X) = f_1(X) + f_2(X) + f_3(X) \tag{5}$$

The design of the feature fusion layer ensures the full integration of multi-branch features while avoiding feature loss or redundancy. In order to reduce the computational cost of the multi-branch module, batch normalization and activation function (ReLU) are configured after each branch convolution layer to improve training efficiency. In addition, this paper uses depthwise separable convolution to replace traditional convolution operations to further reduce the parameter scale and improve the network inference speed. For each convolution operation, the computational complexity of depthwise separable convolution is reduced from $O(k^2 \cdot C_{in} \cdot C_{out})$ to $O(k^2 \cdot C_{in} + C_{in} \cdot C_{out})$, where k is the convolution kernel size, C_{in} and C_{out} are the number of input and output channels, respectively.

2.3. Residual Module Design

The multi-branch module extracts multi-scale features of the melt pool image through a parallel structure, but the multi-branch design may cause information loss or redundancy problems when the features extracted by each branch are deeply fused. In addition, as the network depth increases, the gradient vanishing and network degradation problems may affect the further optimization of features. Therefore, this paper introduces a residual module after the multi-branch module to enhance the feature fusion capability and improve the training efficiency and expression ability of the deep network. In the traditional convolution structure, the input feature X is directly mapped to the output feature $H(X)$. In the residual module, the network does not directly fit $H(X)$, but learns the residual $R(X)$, and the output is:

$$H(X) = R(X) + X \tag{6}$$

Wherein, $R(X)$ represents the residual feature extracted by two layers of convolution operations. The specific structure includes two layers of 3×3 convolution, each layer is followed by batch normalization and activation function (ReLU) for nonlinear feature extraction. The jump connection directly adds the input features to the residual features to form the module output:

$$Y = \mathrm{ReLU}\left(\mathrm{BN}\left(\mathrm{Conv}_{3\times3}\left(\mathrm{BN}\left(\mathrm{Conv}_{3\times3}(\mathrm{X})\right)\right)\right) + X\right) \tag{7}$$

When the input and output feature dimensions do not match, the jump connection introduces 1×1 convolution to adjust the dimension, thereby ensuring the effectiveness of the addition operation.

2.4. Loss function

In order to improve the accuracy of melt pool monitoring, this paper adopts a multi-task joint loss function, including classification loss and regression loss. The total loss function is defined as:

$$\mathrm{L} = \alpha \mathrm{L}_{\mathrm{cls}} + \beta \mathrm{L}_{\mathrm{reg}} \tag{8}$$

DOI: 10.54254/2755-2721/121/2025.19844

Where, L_{cls} represents the cross entropy loss of the classification task, L_{reg} represents the regression loss of the bounding box or shape feature, and α and β are balance coefficients. The cross entropy loss L_{cls} is expressed as:

$$L_{cls} = -\sum_{i=1}^{N} y_i \log(p_i) \tag{9}$$

Where, y_i is the true category label, and p_i is the predicted probability. The regression loss L_{reg} is defined as the mean square error (MSE):

$$L_{reg} = \frac{1}{N}\sum_{i=1}^{N} \| t_i - \hat{t}_i \|^2 \tag{10}$$

3. Experimental analysis

3.1. Dataset and Experiment settings

To evaluate the performance of the proposed multi-branch residual network, a molten pool image dataset was constructed in a laboratory environment, encompassing three states: "stable molten pool," "overheated molten pool," and "abnormal cooling." Data were acquired using arc additive manufacturing equipment with a high-resolution industrial camera, combined with an infrared filter and fill light to minimize arc interference, resulting in 6,000 images. Annotation was completed through a combination of manual and automated methods, based on experimental records and molten pool state parameters such as temperature and shape. The dataset was divided into training (70%), validation (15%), and test sets (15%).

The experiments were conducted on hardware featuring an NVIDIA RTX 3090 GPU (24GB), 128GB of RAM, and 2TB SSD storage, using Ubuntu 20.04 and PyTorch 1.12.1 with CUDA 11.6. The Adam optimizer was employed with an initial learning rate of 0.001, dynamically reduced via the ReduceLROnPlateau strategy. The minimum learning rate was set to 1×10^{-6}, with a batch size of 32, cross-entropy loss, and a maximum of 100 training epochs, employing early stopping if validation loss showed no improvement for 10 consecutive epochs. The evaluation metrics used in this study include accuracy, precision, recall, and F1 score, which together evaluate the classification performance of the model.

3.2. Comparative Analysis

In order to verify the effectiveness of the proposed multi-branch residual network, we compared the classification performance and inference efficiency of the proposed method with other network models, including traditional convolutional neural network (CNN), ResNet-50 and single-branch residual network. The experimental results are shown in the Table 1.

Table 1: Performance of different methods in Melt Pool Monitoring

Model	Accuracy(%)	Precision(%)	Recall(%)	F1(%)	Inference time(ms)
CNN	84.7	82.5	80.4	81.4	12.3
ResNet-50	90.3	89.2	87.8	88.4	15.0
Single-branch Residual Network	91.6	90.4	89.6	90.6	14.6
Ours	95.1	94.2	93.8	94.2	13.1

DOI: 10.54254/2755-2721/121/2025.19844

The experimental results show that the proposed method is significantly better than other comparison models in accuracy, precision, recall and F1 score, with an accuracy of 95.1%. At the same time, the inference time is only 13.1ms, which is close to the traditional CNN, which fully proves that it has real-time processing capabilities in industrial applications.

3.3. Ablation Analysis

In order to explore the impact of multi-branch structure and residual module on network performance, this paper conducted an ablation experiment to gradually remove the multi-branch structure and residual module in the network. The experimental results are shown in the Table 2:

Table 2: Performance of ablation experiment

Model	Multi-Branch	Residual Module	Accuracy(%)	F1(%)
Full Network	√	√	95.1	94.2
w/o Residual Module	√	×	92.4	91.0
w/o Multi-branch Structure	×	√	89.9	88.6
w/o Both Modules	×	×	85.7	83.8

Experiments show that after removing the multi-branch structure or the residual module, the classification performance of the network is significantly reduced. Among them, after removing the residual module, the accuracy rate dropped from 95.1% to 92.4%, and the F1 score dropped from 94.0% to 91.0%, indicating that the residual module plays an important role in improving feature fusion capabilities and deep network optimization. After removing the multi-branch structure, the accuracy rate further dropped to 89.9%, and the F1 score dropped to 88.6%, indicating that the multi-branch structure is crucial for multi-scale feature extraction. In addition, when the multi-branch structure and the residual module are removed at the same time, the accuracy rate and F1 score drop to 85.7% and 83.8%, respectively, which is a significant drop compared with the complete network.

4. Conclusion

This paper presents a multi-branch residual network for real-time and accurate molten pool monitoring in WAAM. The proposed method effectively extracts and fuses multi-scale features, achieving high classification performance with an accuracy of 95.1% and maintaining a low inference time of 13.1ms. Comparative and ablation studies demonstrate the essential roles of the multi-branch structure and residual module in improving feature extraction and network optimization. This study provides a robust solution for industrial molten pool monitoring and offers a foundation for future research in enhancing model robustness and multi-modal data integration.

References

[1] *Williams, S. W., Martina, F., Addison, A. C., Ding, J., Pardal, G., & Colegrove, P. (2016). Wire+ arc additive manufacturing. Materials science and technology, 32(7), 641-647.*

[2] *Rodrigues, T. A., Duarte, V., Miranda, R. M., Santos, T. G., & Oliveira, J. P. (2019). Current status and perspectives on wire and arc additive manufacturing (WAAM). Materials, 12(7), 1121.*

[3] *Ji, F., Qin, X., Hu, Z., Xiong, X., Ni, M., & Wu, M. (2022). Influence of ultrasonic vibration on molten pool behavior and deposition layer forming morphology for wire and arc additive manufacturing. International Communications in Heat and Mass Transfer, 130, 105789.*

[4] *Sheng, J., Cai, Y., Li, F., & Hua, X. (2017). Online detection method of weld penetration based on molten pool morphology and metallic vapor radiation for fiber laser welding. The International Journal of Advanced Manufacturing Technology, 92, 231-245.*

DOI: 10.54254/2755-2721/121/2025.19844

[5] Ebrahimi, A., Kleijn, C. R., & Richardson, I. M. (2021). A simulation-based approach to characterise melt-pool oscillations during gas tungsten arc welding. International Journal of Heat and Mass Transfer, 164, 120535.

[6] Aucott, L., Dong, H., Mirihanage, W., Atwood, R., Kidess, A., Gao, S., ... & Atkinson, H. V. (2018). Revealing internal flow behaviour in arc welding and additive manufacturing of metals. Nature communications, 9(1), 5414.

[7] Wang, Y., Huang, J., Wang, Y., Feng, S., Peng, T., Yang, H., & Zou, J. (2020). A CNN-based adaptive surface monitoring system for fused deposition modeling. IEEE/ASME Transactions On Mechatronics, 25(5), 2287-2296.

[8] Xia, C., Pan, Z., Li, Y., Chen, J., & Li, H. (2022). Vision-based melt pool monitoring for wire-arc additive manufacturing using deep learning method. The International Journal of Advanced Manufacturing Technology, 120(1), 551-562.

[9] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... & Fei-Fei, L. (2015). Imagenet large scale visual recognition challenge. International journal of computer vision, 115, 211-252.