1D Segmentation Network for 3D Seam Weld Grinding

Chunyu Dong¹, Tiankai Shi¹, Qunfei Zhao¹, Xiaolin Huang¹,² and Chao Liu²

¹School of Electronical Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, 200030, China
²Vitex, Shanghai, 200000, China

Corresponding author Email: xiaolinhuang@sjtu.edu.cn

Abstract. With the development of deep learning, convolutional neural networks provide new solutions for obstacles in the industry. For seam weld grinding, traditional algorithms are neither convenient nor efficient. Meanwhile, previous methods are not robust for various shapes of the weld seam. In this paper, we propose a new algorithm based 1D convolutional neural network for 3D weld seam grinding. We test different loss functions for the 1D segmentation network and picked the best one for model training. Besides, we design various feature extracting blocks and make extensive experiments on the cloud point data set of the weld seam. The best combination of the loss function and feature extractor is generated for weld seam prediction. With the open operation on the 3D map and the elimination of abnormal points, we obtain a robust prediction grinding trail for the robot controller.

1. Introduction

Due to the diverse positions of the welds and the different shapes of the workpieces, most existing methods of grinding the welds are manually controlling the weld seam grinder. During the grinding process, the engineer observes the workpiece being polished to correct the direction of machine drilling and feeding. The manually grinding method is inefficient because most of the time is wasted on observation and posture correction. Meanwhile, sparks aroused during polishing make it dangerous for engineers to observe the feeding angle closely, resulting in an inadequate polishing effect. Automatically polishing a metal workpiece with seam welds like [1,2] avoids the defects mentioned above. [2] exploited a binocular vision inspection system that extracts feature points according to the difference between the original image and the image with a median filter. In [1], an SVM-based classifier was employed to predict the welding state. However, these methods are limited by the shape of the weld seam. It is hard to form a universal and effective solution based on traditional machine vision methods.

The development of image segmentation methods based on deep learning makes it possible to handle this task via convolutional neural networks. During the past few years, the emergence of U-Net [3], Segnet [4], and FCN [5] has led the research of 2D image segmentation based on fully convolutional neural networks. Afterward, V-Net [6] architecture was proposed for volumetric medical image segmentation. All these models treat segmentation as a pixel-wise classification task via encoder-decoder architecture. Researchers tend to improve the semantic segmentation performance of these neural networks through better feature extractors, more efficient feature fusion ways, and more brilliant loss functions.
In this paper, we propose a one-dimensional segmentation network based on U-Net architecture for 3D weld seam segmentation. The proposed model accepts one-dimensional distance information. After encoding and decoding procedures, the input distance information is classified into different categories. The training and validation scan data are collected from different seam welds by the laser sensor. We design multiple loss functions and feature extractors and make extensive experiments on the collected data set. Finally, the best combination of them is generated to segment weld seam online. The main contribution of this paper can be summarized as:

- The adaptation of the U-Net architecture to the weld seam grinding, which suits welds of different shapes.
- A comparison of the effects of different loss functions and distinct feature extractors.

The rest of the paper is organized as follows. In Section 2, we illustrate the proposed method. In Section 3, we give a detailed description of our experiments. Section 4 is the conclusion.

![Figure 1. The workflow.](image)

2. Methodology
In this part, we first give a brief introduction to what our workflow is. Then we give a depiction of our presented segmentation network. Afterward, we introduce loss functions used during training.

2.1. Workflow
The overall workflow of the robot grinding weld seam is depicted in Fig. 1. In our weld grinding method, the data set is a must for training the segmentation network. So, we first collect enough 3D point cloud data with the help of a robot and a laser sensor. After cropping and padding, the data is unified to the same size. Then we feed the model with these data for optimization. When testing, the robot with a laser sensor first moves and scans to gain 3D point cloud data of the workpiece. Then, the collected data is segmented with the trained model. The predicted mask is post-processed before we form a trail. All the track points are sent to the robot controller, and the robot will grind along this track.

2.2. Model
Our 1D segmentation network, as is depicted in Fig. 2, is based on the U-Net architecture. In the schematic diagram, the curve in the upper left corner represents the input of the model. The colorized curve right below is the output of the model. The green part refers to the mask of the weld seam while...
the red part means the background. It accepts input of $n \times 1 \times 1536$, in which $n$ denotes the batch size, 1 represents the channel number, and 1536 means the width of weld seam after pre-processing. Feature extraction blocks are mainly composed of one-dimensional convolution, group normalization, and rectified linear unit (ReLU) activation function. Details will be explained in section 3. The downscale block in our work is one-dimensional convolution with kernel size 3, stride 2, and padding 1. Through a sequence of feature extraction block and downscale block, original $n \times 1 \times 1536$ weld seam data is encoded to the shape of $n \times 128 \times 48$. For decoding, we apply the same strategy from U-Net: concatenating different feature maps to resist information loss during forwarding, especially downscale block. Finally, a $n \times 24 \times 1536$ feature map produced by the last concatenation is squeezed to the original input data shape via a 1D convolution and a sigmoid activation.

![Figure 2. The proposed 1D segmentation network.](image)

### 2.3. Loss functions

In our work, the segmentation is treated as a point-wise classification. We adapt the most common loss functions in semantic segmentation: the cross-entropy loss. Apart from that, the Dice loss function [6] is also widely used, which has better performance in the category imbalanced data set. Because the classes of our seam weld segmentation data set are highly imbalanced, we assume the 1D dice loss function may have better training performance. We adapt these two loss functions and their combination to the one-dimensional task as follows:

\[
L_{BCE} = - \frac{1}{w} \sum_{i=1}^{w} (y_i \log \hat{y}_i + (1 - y_i) \log (1 - \hat{y}_i))
\]

\[
L_{Dice} = 1 - \frac{2 \sum_{i=1}^{w} y_i \hat{y}_i + \epsilon}{\sum_{i=1}^{w} y_i + \sum_{i=1}^{w} \hat{y}_i + \epsilon}
\]

\[
L_{Combined} = (1 - \alpha)L_{BCE} + \alpha L_{Dice}
\]

where $w$ represents the width of a seam weld. $y_i$ is the ground truth class of a point and $\hat{y}_i$ is the prediction category of the point. In the Eq. (3), $L_{Combined}$ denotes a combination of binary cross-
entropy (BCE) loss and Dice loss via a ratio $\alpha$ which ranges from 0 to 1. The combined loss function is most widely used in nowadays U-Net like neural networks due to its superior effects. For all experiments, $\alpha$ is set as 0.5 to keep fair in comparison.

3. Experiments

In this part, we make a description of the data set and the data pre-processing. Then we show various comparisons of the loss function and feature extractor. Finally, we visualize our 3D segmentation performance.

3.1. Experiment setting and data pre-processing

All the implementation of this work is based on Pytorch architecture. The training and testing settings are under windows 10 system with 1 Nvidia Geforce 2070 super graphics card. During training, the data is randomly flipped horizontally for augmentation. We adopt an SGD optimizer with bath size 8, initial learning rate 0.0001, and decay it every 10 epochs until 30 epochs finished.

We collect 3700 groups of one-dimensional data, which is not a large data set but enough for training a 1D segmentation network as a 1D segmentation network has fewer parameters than networks like 2D or 3D networks. From the data set, 2433 groups of them are used for training and the rest for validation. Note that, there is no workpiece providing data for both training and validation, avoiding data leakage. For better training effects, all the seam data is normalized to the range of $-1$ and $1$ through $\frac{x - x_{\text{mean}}}{x_{\text{max}} - x_{\text{min}}}$, in which $x_{\text{mean}}$, $x_{\text{max}}$, and $x_{\text{min}}$ are mean, maximum and minimum values of this group of data respectively. Afterward, the too-wide seam data are pruned while too narrow seam data are padded to the same width (1536 in our experiment).

![Figure 3. Different feature extractors.](image)

3.2. Better loss function

We choose the basic feature extractor in Fig. 3(a) and binary cross-entropy loss function for our baseline model. The basic feature extractor is composed of a sequence of a 1D convolution with filter size 1 and a 1D convolution with filter size 3, each followed by a group normalization [7] and a rectified linear unit (ReLU) activation function. The 1D network converges after 30 epochs of training. The evaluation metrics in our experiments include the intersection over union ($IOU$) and soft dice coefficient. $IOU = \frac{TP}{TP + FP + FN}$, where $TP$, $TN$, $FP$, and $FN$ denote the number of true positive, true negative, false positive, and false negative respectively.

We compare three different loss functions for the segmentation network and the combined loss function shows the best training effects. As we can see in Fig. 4, the training loss of binary cross-entropy loss function is more turbulent than the training loss of the Dice loss function. Because the
positive and negative samples are not balanced, the cross-entropy loss function makes the network more easily to fall into sub-optimal convergence. The combined loss function combines the advantages of BCE loss and Dice loss, generating better training results. From Tab. 1, compared with pure BCE loss function and dice loss function, combined loss function improves the IOU metric dice coefficient of validation by 14.6% and 5.5% respectively.

3.3. Better feature extractor

![Image: Validation performance with different loss functions](image1)

**Figure 4.** Validation performance with three loss functions: BCE loss, dice loss and the combined loss. Model trained with the combined loss performs best.

![Image: Validation losses with distinct feature extractors](image2)

**Figure 5.** Different performance of various feature extractors. Model with triple densely sum extractor performs best.

The emergence of Resnet [8] solves the problem of vanishing gradient caused by increasing the depth of the neural networks through skip connection. Since then, Inception V4 [9], Densenet [10], and VoVNet [11] explored how to encode features in an energy and computation efficient way. Just like MultiResUNet [12], we explore building an efficient feature extracting block of a 1D segmentation network in this paper.

Original 2D U-Net demonstrates that two $3 \times 3$ convolutional operations give the best medical image segmentation. For the 1D segmentation network, we make a similar experiment on how many convolutional operations are best for training. Moreover, as is explained in [9], a series of two convolutional operations with filter size 3 resembles a convolutional operation with filter size 5. We factorize the bigger 1D convolutional operation by a sequence of smaller and lightweight convolutional operations with filter size 3. As is shown in Fig. 3, each 1D convolutional operation follows a group normalization and a ReLu activation function. Group normalization [7] proves to be not sensitive to batch size, which is more robust than batch normalization. Fig. 3(a) is a basic 1D convolution block. (b) and (c) are double and triple 1D convolution blocks respectively.

Fig. 3(d) and Fig. 3(f) utilize skip connection between 1D convolutional operations with kernel size 3, enhancing the capacity of perceiving subtle gradient change of weld seam data, which is inspired by Densenet [10] and VoVNet [11]. Besides, we also test replacing residual sum operation with residual concatenation in Fig. 3(e) and Fig. 3(g). We make experiments with these feature extractors, and compare their segmentation ability under the same environment. The results are shown in Fig. 5.

| Table 1. Performances of different loss functions and feature extractors on validation set. |
|---|---|---|
| loss function w/ basic block | settings | IOU | Dice |
| BCE Loss | 62.6 | 68.3 |
| dice loss | 71.7 | 82.1 |
| combined loss | **77.2** | **85.0** |
| feature extractor w/ combined loss | basic | 77.2 | 85.0 |
| double | 77.9 | 86.5 |
In Fig. 5, the validation losses of a simple sequence of 1D convolution like basic, double, and triple conv1d fluctuate more fiercely. Generally, feature blocks with skip connections are more stable and their losses are smaller than without. The best feature block is the triple 1D convolution with densely sum operation, which is depicted in Fig. 3(f). Meanwhile, its IOU and Dice coefficient metrics are highest among these seven blocks. From Tab. 1, compared with basic feature extractor, the best feature extractor improves the IOU metric dice coefficient of validation by 5.8% and 4.3% respectively.

|                | Value 1 | Value 2 |
|----------------|---------|---------|
| triple          | 78.7    | 86.4    |
| triple densely concate | 77.2    | 84.5    |
| double densely sum   | 80.5    | 87.4    |
| double densely concate | 82.0    | 87.9    |
| triple densely sum   | 83.0    | 89.3    |

3.4. Post-processing and visualization
When the robot grinds the weld seam, a prediction trail is sent to the robot controller based on the segmentation result of the 1D segmentation network. To generate the prediction trail, we have to post-process the predictions of the trained network because there are subtle glitch predictions. For the generated 3D prediction map, we perform a simple open operation, which is similar to the usage in a 2D image, to eliminate small error predictions. After the open operation, the coarse prediction trails

Figure 6. From top to bottom: 3D point cloud, the label of the weld seam, prediction of the basic extractor with BCE loss, prediction of the basic extractor with the best loss function, prediction of the best feature extractor with the best loss function, prediction after post-processing, and the predicted trail map.
are just the middle points of the predicted masks. Then, outliers of these points are dismissed, replaced by interpolation values. The final performance is shown in Fig. 6.

4. Conclusion

In this paper, we propose a 1D segmentation network to segment the 3D weld seam. We test different loss functions and different feature extracting blocks, pursuing the best combination of them. With the best 1D segmentation model, we perform post-procession to gain the prediction trail of the weld seam. With this workflow, we can grind the 3D weld seam more efficiently.

References

[1] Pandiyan V, Tjahjowidodo T. In-process endpoint detection of weld seam removal in robotic abrasive belt grinding process[J]. The International Journal of Advanced Manufacturing Technology, 2017, 93(5-8): 1699-1714.
[2] Dinham M, Fang G. Autonomous weld seam identification and localisation using eye-in-hand stereo vision for robotic arc welding[J]. Robotics and Computer-Integrated Manufacturing, 2013, 29(5): 288-301.
[3] Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation[C]. International Conference on Medical image computing and computer-assisted intervention. Springer, Cham, 2015: 234-241.
[4] Badrinarayanan V, Kendall A, Cipolla R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation[J]. IEEE transactions on pattern analysis and machine intelligence, 2017, 39(12): 2481-2495.
[5] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation[C]. Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 3431-3440.
[6] Milletari F, Navab N, Ahmadi S A. V-net: Fully convolutional neural networks for volumetric medical image segmentation[C]. 2016 fourth international conference on 3D vision (3DV). IEEE, 2016: 565-571.
[7] Wu Y, He K. Group normalization[C]. Proceedings of the European conference on computer vision (ECCV). 2018: 3-19.
[8] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]. Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.
[9] Szegedy C, Ioffe S, Vanhoucke V, et al. Inception-v4, inception-resnet and the impact of residual connections on learning[J]. arXiv preprint arXiv:1602.07261, 2016.
[10] Huang G, Liu Z, Van Der Maaten L, et al. Densely connected convolutional networks[C]. Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 4700-4708.
[11] Lee Y, Hwang J, Lee S, et al. An energy and gpu-computation efficient backbone network for real-time object detection[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2019: 0-0.
[12] Ibtehaz N, Rahman M S. MultiResUNet: Rethinking the U-Net architecture for multimodal biomedical image segmentation[J]. Neural Networks, 2020, 121: 74-87.