An Appearance Inspection Method for Resistance Spot Welding Based on Semantic Segmentation

Tianyu Zheng¹, Ye Yang¹, Peng Zheng¹, Lukas Benz² and Lei Wang¹,*

¹School of Aerospace Engineering, Xiamen University, 361005 Xiamen, China
²Faculty of Mechanics and Electronics, Heilbronn University, 74081 Heilbronn, Germany

*Corresponding author

Abstract. Resistance spot welding (RSW) plays an important role in manufacturing. The quality of the welding can be efficiently assessed by its appearance. Image segmentation is an important part of the RSW appearance inspection. However, the classical image segmentation algorithms cannot work very well because of the various RSW appearances. In this study, a novel inspection method is proposed based on semantic segmentation. We choose MobileNetV2 as the backbone for the semantic segmentation. After modification and optimization of the network, our model achieves an accuracy of 89% mean intersection-of-union (mIOU), which is averagely 30% higher than the classical image segmentation algorithms. A classifier further evaluates the quality of the RSW according to some geometric features of the segmented regions, and the classification accuracy is improved by 0.79%. This research is of great importance for the high accuracy quality control of the massive production to reduce the producing cost and improve the efficiency of the RSW pipeline.

Keywords: Resistance spot welding; Semantic segmentation; Appearance inspection.

1. Introduction

Resistance spot welding (RSW) is widely used in manufacturing. This technology melds two metal components together by melting them with resistance heat. The welding quality of RSW products suffers from different weld conditions like electric current, processing time, and pressure[1]. The quality control of the RSW products is of great importance for the massive production of the RSW pipeline. Considerable efforts have been devoted to evaluating the quality of RSW automatically. The methods based on destructive testing[2, 3] are time-consuming, while the ultrasonic testing[4, 5] highly depends on the experience of the operator.

Another important method is based on the appearance. Some researches tried to inspect the appearance based on the classical image processing [6, 7] like edge detection, they were only suitable for simple shapes such as circle or line. It’s difficult to transfer the algorithm to complex shapes because of the various RSW appearances. As the rapid development of deep learning, neural networks have been used in the RSW field[8, 9]. Ye et al.[10] showed that the neural network yields 95.82% accuracy in image classification. However, it’s difficult to modify the inference result after the network model is trained. It should be pointed out that changing the evaluation standard according to different customers is common and essential in the factory.

As the Convolutional Neural Network (CNN) develops, the semantic segmentation becomes a vital method for scene understanding[11]. However, it has a high computational cost normally. In the real industrial application, many networks should be carried out on a computationally limited platform to reduce the cost. There are considerable research efforts which have been devoted to the trade-off...
between accuracy and speed like AlexNet, VGGNet, and ResNet. Howard proposed an efficient network called MobileNet for mobile vision applications[12]. Mark upgraded MobileNet to MobileNetV2[13] by introducing “linear bottleneck”. MobileNetV2 has much fewer parameters, meanwhile enables fast training and inferencing with a low computational cost.

In this study, we proposed a method combining semantic segmentation and classification together. As shown in figure 1, we segment the image into different regions based on MobileNetV2. MobileNetV2 yields 89% mean intersection-of-union (mIOU) in the segmentation which is 30% averagely higher compared to the classical algorithms. Then some geometric features like distance, area, or coordinates are extracted. With the same classifier, our method improved the classification accuracy by 0.79%.

2. Methodology

2.1. Sample Preparations

In our study, all the images were obtained by a Complementary Metal-Oxide Semiconductor (CMOS) camera from the RSW pipeline. We cropped the original images into 871×261 pixels using a fixed region of interest (ROI) as shown in figure 2 (a). We divided the image into 6 regions: background (BG) region, molten pool (MP) region, welding squeezed (WS) region, welding un-squeezed (WUS) region, lead wire (LW) region, and dross (DR) region as shown in figure 2 (b).

As mentioned above, many factors can influence the appearance. For example, the DR region has different sizes, colour, and shapes in different images. Moreover, as a result of the welding head ageing and industrial process improvements, the texture of WS region changes as well. Figure 3. (a) shows some typical good (OK) samples. Figure 3. (b) and figure 3. (c) show some not-good (NG) samples for different reasons. There are many types of unqualified samples and here only demonstrate two basic types of them.
Figure 3. Some OK and NG images. (a) OK images with different textures. (b) WS region has the wrong length. (c) DR region is too large. The texture, colour, and size are different in each image.

There are 2084 images with manual annotations in total. We divided them into train set, validation set, and test set. Specifically, there are 1380 images in the train set, 460 images in the validation set, and 244 images in the test set. We also prepared another 379 images for classification evaluation. Each image was labelled as OK or NG.

2.2. MobileNetV2
As mentioned, we chose MobileNetV2 for fast training and inferencing. Conclusively, MobileNetV2 has three outstanding features: depthwise separable convolution, linear bottleneck, and inverted residual. Depthwise separable convolution is commonly known to reduce the parameters significantly by splitting a normal convolution into two parts: a depthwise convolution and a pointwise convolution. Since information may be discarded after Rectified Linear Unit (ReLU) function in a low-dimensional subspace, a linear bottleneck with a linear activation function is inserted into a convolutional layer to capture more information. Therefore, the linear bottleneck layer contains all the necessary information intuitively. Next, an inverted residual module is applied between linear bottlenecks as a shortcut connection. As illustrated in table 1, we employed the same network architecture as MobileNetV2 in this study. The pooling layer was removed to obtain dense segmentation. The normal convolution was also replaced by atrous convolution[13, 14], which enables us to control the output resolution. The channel in the last layer was fixed to 6 because we defined 6 regions above.

| Layer       | Expansion factor | Channel | Repeat times | Stride |
|-------------|------------------|---------|--------------|--------|
| Conv2d      | -                | 32      | 1            | 2      |
| Bottleneck  | 1                | 16      | 1            | 1      |
| Bottleneck  | 6                | 24      | 2            | 2      |
| Bottleneck  | 6                | 32      | 3            | 2      |
| Bottleneck  | 6                | 64      | 4            | 2      |
| Bottleneck  | 6                | 96      | 3            | 1      |
| Bottleneck  | 6                | 160     | 3            | 2      |
| Bottleneck  | 6                | 320     | 1            | 1      |
| Conv2d 1×1 | -                | 1280    | 1            | 1      |
| Conv2d 1×1 | -                | 6       | 1            | 1      |

2.3. Metric
There are at least 4 possible metrics: pixel accuracy (PA), mean PA, mIOU and frequency weighted IOU[15]. However, mIOU is most used to evaluate the average segmentation performance as most researchers like Chen[16], Howard[17], and Wu[18] use it as the main metric in their papers. Therefore, we calculate the IOU of each class and take the mean value across all the classes as mIOU.
3. Results and Analysis

3.1. Experiment Details
For program implementation, we adopted Tensorflow-Slim as our coding tool which has concise and efficient interfaces. All the models were trained on a single TITAN Xp graphics card which has 12GB memory.

We followed a similar training protocol as MobileNetV2. Specifically, we employed output stride (OS) = 8 as default during training and evaluation for better segmentation. All the weights parameters were initialized by the truncated normal method with standard deviation (stddev) = 0.09. A batch norm function with decay = 0.997 was inserted to each convolution layer. Learning power was set to 0.9 in polynomial strategy. Meanwhile, the base learning rate would be decayed by 0.01 every 2000 steps. We adopted momentum optimizer and the momentum parameter was set to 0.9.

3.2. Training

3.2.1. Pre-training. The network was pretrained on MS-COCO[19], VOC 2012 ‘train_aug’ set and ‘trainval’ set[20]. We downloaded the pretrained checkpoint from github[21]. Specifically, we employed base learning rate = 0.005, batch size = 8, training steps = 60000. We observed 1.1% performance degradation for mIOU on the validation set when the pretrained checkpoint is not employed. The mIOU curves are shown in figure 4.

3.2.2. Training steps. Figure 4 also indicates that the curve of the pre-training group converges faster than the group without pre-training. The mIOU is stable after 20000 steps. Therefore, we set training steps = 25000 with the pretrained checkpoint in the following experiments to prevent over-fitting. Particularly, IOU value indicates an average value between 20000 steps and 25000 steps in the following content.

3.2.3. Batch size. Because of the limited GPU memory, only small batch size less than 8 can be employed when the input size is 871×261 pixels and OS = 8. As shown in table 2, the average mIOU drops by around 0.7% when batch size = 4. Due to a similar performance between batch size = 8 and batch size = 6, we employed batch size = 8.

Table 2. Batch size strategy when evaluating on the validation set.

| Batch size | mIOU    |
|------------|---------|
| 8          | 89.01%  |
| 6          | 89.03%  |
| 4          | 88.34%  |
3.2.4. Optimizer. We also experimented different optimizers at a fixed learning rate (0.005). The mIOU curve on the validation set is illustrated in Figure 5. We observed that employing Momentum optimizer is better than Gradient optimizer by around 3.1% on average. While RMSprop optimizer even cannot converge.

![Figure 5. The mIOU curve of different optimizers.](image)

3.3. Comparison with the Classical Algorithm

To further evaluate the segmentation performance of our method, we compared our method with a classical image processing algorithm on the test set. The classical image processing algorithm is based on threshold segmentation, edge detection, and morphology. Note that the classical algorithm does not extract DR region in this study because it is unnecessary for the current project. We employed a complex procedure because the algorithm should adapt to possible colour, light, or texture as more as possible.

As illustrated in table 3, when we employed OS = 8 during the evaluation, the proposed method improves the segmentation performance significantly in WS region, MP region, and LW region by over 30% While in WUS region, our method achieves only 55.39% IOU because this region is small and less in our dataset.

| IOU (%) | Time (ms) |
|---------|-----------|
| WS      | WUS       | MP       | LW       | mean     | Time (ms) |
| Our method (OS = 8) | 93.39 | 55.39 | 96.36 | 91.82 | 84.24 | 938.45 |
| Our method (OS = 16) | 93.04 | 52.41 | 96.17 | 90.83 | 83.11 | 454.63 |
| Our method (OS = 32) | 74.21 | 28.84 | 88.65 | 68.55 | 65.06 | 369.69 |
| Classical algorithm | 63.13 | 6.12 | 66.99 | 38.49 | 43.68 | 422.99 |

We also tried a different OS during the evaluation. The mIOU decreases by about 20% when OS = 32. However, considering the processing time, the classical algorithm costs 423ms on average when the program is performed on an i7 CPU, while our method costs 938ms, 455ms, and 369ms when OS = 8, 16, and 32 respectively. OS = 16 yields the best trade-off between accuracy and efficiency. Figure 6 provides a visual comparison of some test images between our method (OS = 16) and the classical algorithm.
We further experimented our method with OS = 16 in the classification of the 379 images. Each image has an OK or NG label. We use the same classifier to process the segmentation from our method and the classical method respectively. As Table 4 shows, our method classifies all the images correctly, while the classical algorithm achieves 99.65% accuracy for OK products, and 97.87% accuracy for NG products.

| Table 4. Comparison of classification accuracy between our method and the classical algorithm. |
|-----------------------------------------------|
|                                      | Classical Algorithm | Our method (OS = 16) |
| OK                                | 99.65%              | 100%                  |
| NG                                | 97.87%              | 100%                  |

4. Conclusion
In this study, we proposed a novel inspection method for the RSW appearances based on semantic segmentation. Our work optimized the MobileNetV2 to achieve a significant improvement in the segmentation performance, which further improved the classification accuracy.

The next work will focus on the optimization of the classifier. We are also interested in key point matching instead of semantic segmentation because the former is faster.

References
[1] Zhong, J., et al., Process Research on Pressure Resistance Spot Welding Parameters for Nuclear Grade 316L Stainless Steel QD. RECHULI JISHU YU ZHUANGBEI, 2019. 40(05): p. 12-16.
[2] Khodabakhshi, F., M. Kazeminezhad, and A. Kokabi, On the failure behavior of highly cold worked low carbon steel resistance spot welds. Metallurgical and Materials Transactions A, 2014. 45(3): p. 1376-1389.
[3] Khodabakhshi, F., M. Kazeminezhad, and A. Kokabi, Metallurgical characteristics and failure mode transition for dissimilar resistance spot welds between ultra-fine grained and coarse-grained low carbon steel sheets. Materials Science and Engineering: A, 2015. 637: p. 12-22.
[4] Liu, J., et al., Ultrasonic C-scan detection for stainless steel spot welding based on wavelet package analysis. Journal of Wuhan University of Technology-Mater. Sci. Ed., 2015. 30(3): p. 580-585.
[5] Summerville, C., et al., Nugget diameter in resistance spot welding: a comparison between a dynamic resistance based approach and ultrasound C-scan. Procedia Engineering, 2017. 183: p. 257-263.
[6] Hou, Z., et al., Application of image processing in examination of resistance spot welding quality. Welding Technology, 2012. 41(02): p. 45-47+70.

[7] Alghannam, E., et al., A Novel Method of Using Vision System and Fuzzy Logic for Quality Estimation of Resistance Spot Welding. Symmetry, 2019. 11(8): p. 990.

[8] Wan, X., et al., Weld quality monitoring research in small scale resistance spot welding by dynamic resistance and neural network. Measurement, 2017. 99: p. 120-127.

[9] Martín, Ó., M. López, and F. Martín, Artificial neural networks for quality control by ultrasonic testing in resistance spot welding. Journal of Materials Processing Technology, 2007. 183(2-3): p. 226-233.

[10] Ye, S., et al. A Vision Inspection System for the Defects of Resistance Spot Welding Based on Neural Network. in International Conference on Computer Vision Systems. 2017. Springer.

[11] Garcia-Garcia, A., et al., A review on deep learning techniques applied to semantic segmentation. arXiv preprint arXiv:1704.06857, 2017.

[12] Howard, A.G., et al., Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861, 2017.

[13] Sandler, M., et al. Mobilenetv2: Inverted residuals and linear bottlenecks. in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.

[14] Chen, L.-C., et al., Semantic image segmentation with deep convolutional nets and fully connected crfs. arXiv preprint arXiv:1412.7062, 2014.

[15] Long, J., E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. in Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.

[16] Chen, L.-C., et al. Encoder-decoder with atrous separable convolution for semantic image segmentation. in Proceedings of the European conference on computer vision (ECCV). 2018.

[17] Howard, A., et al., Searching for mobilenetv3. arXiv preprint arXiv:1905.02244, 2019.

[18] Wu, H., et al., FastFCN: Rethinking Dilated Convolution in the Backbone for Semantic Segmentation. arXiv preprint arXiv:1903.11816, 2019.

[19] Lin, T.-Y., et al. Microsoft coco: Common objects in context. in European conference on computer vision. 2014. Springer.

[20] Everingham, M., et al., The pascal visual object classes challenge: A retrospective. International journal of computer vision, 2015. 111(1): p. 98-136.

[21] Tensorflow. TensorFlow DeepLab Model Zoo. 2019; Available from: https://github.com/tensorflow/models/blob/master/research/deeplab/g3doc/model_zoo.md.