05-02

16th International Conference on Machine Vision Applications (MVA)
National Olympics Memorial Youth Center, Tokyo, Japan, May 27-31, 2019.

Welding Joints Inspection via Residual Attention Network

Jinguo Zhu[1], Zejian Yuan[1], Tie Liu[2]
[1] Institute of Artificial Intelligence and Robotics
Xi'an Jiaotong University, China
lechatelia@stu.xjtu.edu.cn, yuan.ze.jian@xjtu.edu.cn
[2] Information Engineering College
Capital Normal University, China
liutiel@163.com

Abstract

Welding joints inspection for surface quality evaluation remains a lot of challenging work because of the difficulty in extracting suitable features. We propose a novel residual attention network for automatic inspection of the quality of welding joints. Our network has the ability to extract more useful features while maintaining a compact structure. Owing to the regularization effect of the alpha robust loss we design, our model has enough generality capabilities with limited training samples. In the end, we evaluate the performance of our network on a dataset consisting of welding joints images with score-labelled imperfections, and our proposed method achieves satisfying results in terms of welding joints inspection by predicting quality scores accurately.

1 Introduction

Welding is one of the most important and most often used methods for joining pieces together. In order to ensure that the end-product is defect-free, visual inspection system plays a vital role in industrial inspection due to its convenience and low cost [2]. However, the welding joints inspection is a critical process but not easy to address. The quality of welding should be evaluated by a score based on the degree of the imperfections on the welding joints. Just as shown in the Fig 1, various imperfections can lead to the disqualifications of examples, like welding wrinkles, mixtures between joints, insufficient welding, and excessive welding.

Traditional methods in visual inspection system like statistical pattern recognition, expert systems and human monitoring, are constrained by the crucial inspector training, the need for expert experience, the lack of the adaptability to the working conditions and so on [11].

Convolutional neural network (CNN) [7] solves this problem by extracting problem-specific features automatically directly from the data, which makes it more popular in industrial inspection due to the high production rates of automated production lines [12]. Furthermore, lots of new ideas and improved model architectures in the domain of CNN have been proposed in the past few years. Particularly, with a deep residual learning framework, a "very deep" CNN can be optimized easily and higher accuracy could be gained from the increased model size. Another remarkable thing is the successful application of the attention mechanism of human perception [9, 5] in various image prediction tasks such as images caption[8] and visual question answering [6].

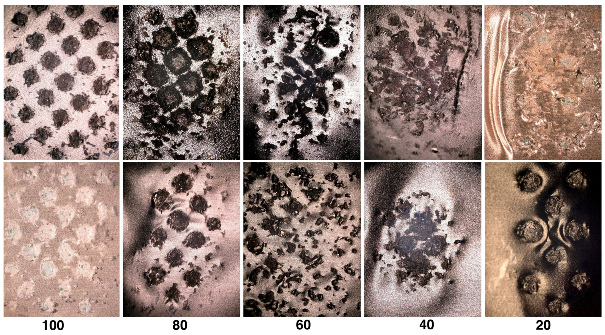


Figure 1: Examples of welding joints with different quality score. From the leftmost column to right, the quality of the welding joints deteriorates as the score below the images decreases, while the quality has the same score in one column.

However, a normal CNN model requires large data sets for training on account of the thousands of or even more parameters in a normal network. In the industrial inspection, however, not all the collected data are ideal sample data sets for the need for training. Consequently, efficient learning from such limited data sets depends on the more elaborate structural design of the network and more optimum training implementation.

This paper proposes a novel Residual Attention Network for automatic inspection of welding joints. We approach to solve this inspection task as an graded score regression based on the gradual changes in quality level of welding joints rather than an imperfections classification whether the sample is qualified. With the architecture of residual attention mechanism, our network can extract sufficiently powerful features on a limited set of labelled training examples while maintaining a compact structure. By using robust alpha loss function, our network is regularized and thus can be trained with less over-fitting. For a given image of welding joints captured by a digital microscope, the trained network outputs an accurate score proportional to its quality. The benefit of the proposed structure and training strategy is verified on the dataset we construct, on which it gains satisfying performance.

2 Residual attention network

Generally, CNN has standard structure that the output of the features extractor, which consists of stacked

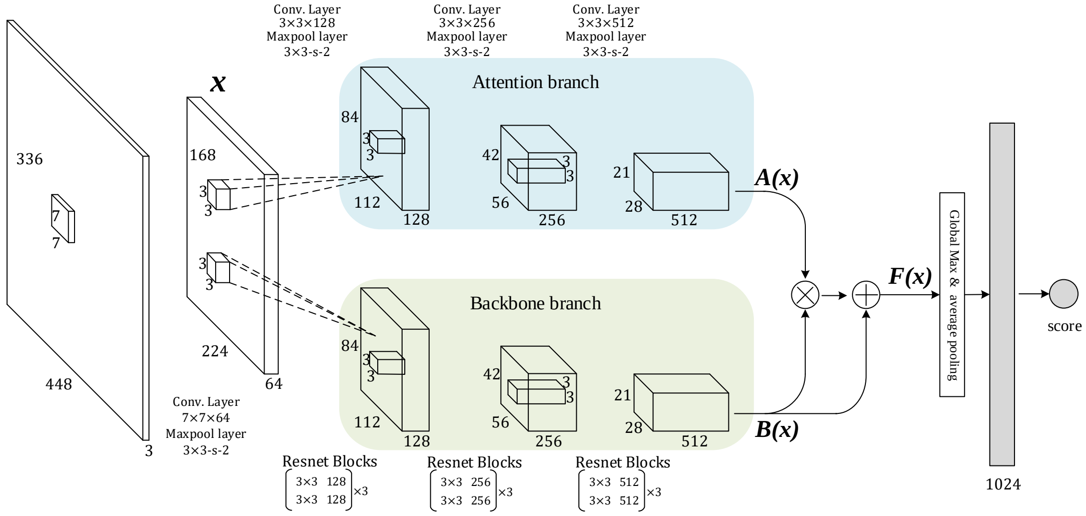


Figure 2: **An illustration of our residual attention network.** Our model has two branches followed by 1 fully connected layer. A 1024-dimensional vector pooled from features map via global average and max pooling layer is fed into the classifier to assign the input examples a score.

convolutional and pooling layers, can be directly input into the classifier. And this typical structure without any additional branch can be called a *plain network*.

Inspired by residual connection [7], our novel residual attention network as shown in Fig 2 has done some improvements on the basic design of plain network, which is comprised of two specific branches: one for extracting feature named backbone branch, the other used as a feature selector named attention branch. We expect that the adding of our soft attention module is similar to the idea of generic shortcut connections in residual learning, where the performance should be no worse than plain network.

The backbone branch outputs $B(x)$ given input x, and the attention branch outputs the weight mask $A(x)$ with the same size of $B(x)$. Thus the output of the residual attention branch in our network is utilized as

$$F_{i,j,c}(x) = (1 + A_{i,j,c}(x)) * B_{i,j,c}(x) \qquad (1)$$

where (i, j) represents the spatial position of a feature map and c is the index of the channel of the output. $F(x)$ is the network's real output after the combination of backbone and attention branch. The element-wise multiples are performed on two feature maps, channel by channel.

It's obvious that the $F(x)$ will approximate the original features maps $B(x)$ extracted by the backbone branch when the $A(x)$ approximates 0. We assume that it's easier to optimize the structure with attention branch than plain one only with backbone branch. Even in the worst case, if the backbone branch learning were optimal and any disturbance in the parameters would result in a loss in the optimal result, our attention branch can avoid this loss by simply leaning all the response activations to zero.

Backbone Branch: The backbone branch of the network consists of three resnet stages as shown in Fig 2, and the task of this branch is extracting valid features for final score regression, which can be interpreted as the network's confidence that the given example's quality is up to standard. One resnet stage is made up of several resnet blocks and we use two convolutions before the addition in each residual block same as [7]. So we use $\begin{bmatrix} k \times k & f \\ k \times k & f \end{bmatrix} \times num$ to denote a resnet stage in Figure 2.

Attention branch: The attention branch will improve the backbone branch features instead of learning the desired but complicated functions directly. We insert the attention module as shown in Fig 2 in the plain network by making the output of first convolutional features be the input of attention module. After three convolutional layers, we combine two branches like equation **1** before the FC layer. In this way, attention module can enhance the good features and suppress the less useful ones from backbone branch, and the network achieves spatial and channel attention with little computation cost.

Global spatial pooling: Instead of adopting the traditional fully connected layers in the end of network, we output the spatial pooling of features maps to the FC layer via a global max and average pooling layer. The usage of global max and average global pooling will give better robustness to the network while maintaining high sensitivity to texture information in images [4]. Besides, the global spatial pooling summing out the activations of every features will make the network learn to be invariant to those spatial transformation.

We should avoid representational bottlenecks when designing a convolutional network [3]. Therefore, if the feature map size is halved, we double the number of filters in order to preserve the network's ability of representation. At the end of the network, the classification score of a single example will be regressed via 1024-d

FC layer followed with a *sigmoid* activation function and a magnification of 100 which will remap the quality score to a regression value from either $[0, 100]$.

3 Alpha robust loss function

We usually use squared loss $L(a) = a^2$ and the absolute loss $L(a) = |a|$ in regression, where a is the difference between the labelled and predicted value. Considering that our score is a random number distributed from $[0, 100]$ but the annotated scores are only multiples of 20, we design a more robust loss function with parametric variable α as:

$$L(\hat{y}, y) = \begin{cases} 0.1\,|(\hat{y}-y)|^{\alpha} & |\hat{y}-y| \le 10 \\ 0.1\alpha\,|(\hat{y}-y)| - \alpha + 1 & \textit{otherwise.} \end{cases} \tag{2}$$

where $\hat{y}$ is the predicted score while y is the labelled one. And parametric variable α is a hyper-parameter depending on a specific problem. This robust loss is always continuous and differentiable to ensure the training of network and also inherits the advantages of two loss functions as shown in Fig 3. Note that, as the parametric variable increases, the loss function gradually transitions from L1 loss when $\alpha = 1$ to huber loss when $\alpha = 2$, followed by more robust functions when $\alpha > 2$. The specific loss is less sensitive to outliers compared to the squared loss when the distribution is heavy tailed, but also can minimize the impact of small score difference. It can be explained that the special loss function will penalize predicted error more strictly when score difference $|\hat{y} - y|$ is closed to *10* and the loss value is compressed when $|\hat{y} - y|$ is closed to zero due to the high-order power, so that the error tolerance of the network is improved which can reduce the over-fitting.

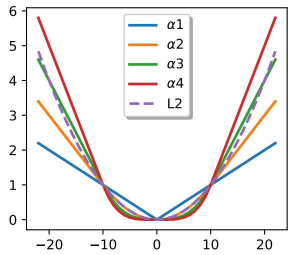


Figure 3: Alpha loss function

In other words, we want our network to be less confident to avoid assigning full probability to the score just as we labelled. the quality of welding joints should not be distributed only to those scores we labelled, but should have small change, depending on its degree of imperfections. Besides, the actual difference between the two example with a score difference of less than *10* is not very large which means that a slight change in imperfection will make the score of the example fine-tuned. Moreover, this loss has a minimum gradient in the compressed area which can regularize our model and make its training more stable similar to the function of learning rate decay which can be regarded as a coarse-to-fine training strategy.

We notice that there are also some other well-known robust loss functions, such as *log-cosh* loss and quantile loss. The *log-cosh* loss has the advantages of the square loss, but will not be strongly affected by the occasional wildly incorrect prediction, which is similar to our alpha loss function. And the quantile loss function turns out to be useful when we are interested in predicting an interval instead of an accurate point. This special loss tries to give different penalties to overestimation and underestimation based on the value of chosen quantile value. We will discuss these robust loss functions with their performance in this specific task later.

4 Experiment

4.1 The dataset generation

We construct a dataset to evaluate the network. Images were captured by a digital microscope VHX-5000, and labelled scores of these images are annotated by professional experts in order to ensure the correctness of the dataset. As you can see in Figure 1, the examples' quality are diverse depending on the different deteriorations like welding wrinkles, mixtures between joints, insufficient welding and excessive welding. The labelling task is to assign a score between 0 and 100 to an example of welding joints based on its quality. We refer to a given example with a score close to 100 as positive if the example is standard while one labelled with a score close to 0 will be classified as a substandard example.

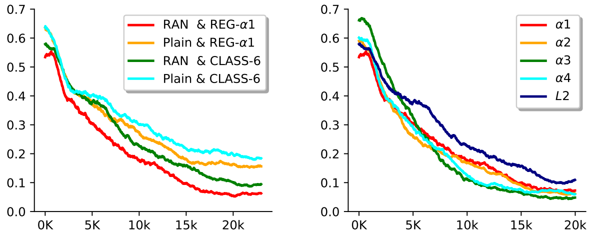
The entire dataset consists of 2000 training examples and 500 testing examples. Every example image of $448 \times 336 \times 3$ pixels is labelled with a graded score which is one of multiples of 20 from 0 to 100, due to fact that the precise score labelling works of the welding solders are costly and can be different from people to people. Therefore, we have adopted the labelling method of only using multiples of 20 as score values and roughly divide these examples into 6 categories based on the scores labelled, which can increase the feasibility of labelling task and its correctness to a certain degree.

The actual distribution of the quality of welding joints is not discrete over the six score values as labelled, but should be continuously distributed between *0.00* and *100*. Considering this fact, we recommend solving this inspection task as score regression problem rather than a binary or multi-class classification to gain better performance.

4.2 Implementation details

Data Augmentation: To reduce the overfitting and learn features efficiently from limited image data, we employ two distinct forms of data augmentation. The first is generating image by horizontal reflection, vertical reflection and rotation with a random small angle. This method significantly increases the diversity of our training dataset and prevents our network from potential overfitting. The second form of data augmentation is changing the sequence of the RGB channels of original images, which forces the nerwork to extract features from the structure and texture information of one example, rather than from color information.

Hyper-parameter Setting: In order to maintain the generality of our network, we define the hyper-parameters as follows. The learning rate is initially set to 1×10^{-3} with exponential decay of 0.95 every 1000 steps to lower the learning rate as the training progresses. The batch size of SGD in training is 8 examples, while we train our network for 200



(a) (b)

Figure 5: Testing error during training procedure. (a) with different structures; (b) with different losses.

epochs. Additionally we set the factor of *L2* regularization $\lambda = 1 \times 10^{-5}$ to avoid overfitting. Training this compact network took around 6 hours on our system equipped with one NVIDIA 1080Ti GPU.

4.3 Results

We evaluate the performance of our network in terms of the true predicted examples, which with the difference between their predicted scores and labelled scores does not exceed 10 according to the compressed area of our loss function. And the accuracy of a model is measured by the proportion of the true predicted examples in the test dataset.

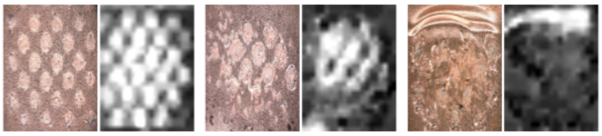


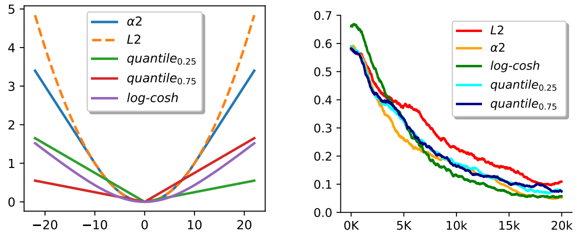
Figure 4: Three Examples illustrating that different images have different corresponding activations of the attention branch in our network.

Analysis of Attention: Fig 4 verifies the validity of our attention branch. The left side of the stitched image is the input image, and the right is the visual output of the 147th channel of the attention branch, where the greater the brightness, the greater the weight to attention the features. It's obvious that our feature branch will pay attention to the characteristic structures in the examples, such as the deteriorating areas, thus will greatly accelerate the converge of the network as a feature selector.

Ablation Experiments: Our experiments mainly aim to verify the validity of the attention branch and the training strategy we proposed. The analysis of these factors in the network is represented in the Fig 5. The structure of the network can be discussed in general from the plain network and our residual attention network (*RAN*). On the basis of the either structure we can model the inspection task as either a 6-class classification problem denoted *CLASS-6* or a score regression task for the examples denoted *REG*. We also explore the impacts of different loss functions, such as the *L2* and four kinds of α losses.

In order to verify that our alpha loss is robust enough, we also do some comparison experiments with other robust loss functions, whose curves shown in Fig 6 (a). Under the premise of keeping the network structure and score regression mechanism unchanged, we replaced our alpha function with *log-cosh* and quantile loss function. Unlike the *log-cosh* is free of hyperparameter , the quantile loss function must choose a quantile value based on whether we want to give more value to positive errors or negative errors. Considering the symmetry of the distribution of dataset samples, we set the quantile vale γ to 0.25 and 0.75. When $\gamma = 0.75$, for concreteness, the over-guessing prediction by a certain amount will give penalties three times as much as under-guessing one by the same amount.

Main Results: Consistent with our prior assumptions, Fig 5 (a) indicates that the additional attention branch and the score regression mechanism manage overcoming the optimization difficulty and gain better performance), while any α loss function in the RAN contributes towards better robust performance compared with the *L2* loss and makes the network learning more efficient, shown in Fig 5 (b).



(a) (b)

Figure 6: (a) Different robust loss functions; (b) Testing error during training procedure.

Table 1: Accuracy of various versions of the network

	plain network	*RAN*
CLASS-6	83.1%	87.3%
REG-L2	90.3%	91.5%
Log-Cosh	93.1%	94.6%
Quantile$_{0.25}$	91.3%	92.7%
Quantile$_{0.75}$	91.2%	92.7%
REG-$\alpha1$	92.9%	93.7%
REG-$\alpha2$	93.2%	94.6%
REG-$\alpha3$	93.8%	95.7%
REG-$\alpha4$	93.9%	94.3%

In Table 1 we show the performance of different options of the network structure with different loss functions in detail. From this we can see the score regression solution is considerably better than the classification solution with an about 5% improving margin. While the attention network is minimally better than plain network with improving the accuracy by about $1 \sim 2\%$. Moreover, The *alpha* loss is slightly better than traditional *L2* loss but this is still a saturation phenomenon that the effect of improving accuracy is not obvious when the α increases. Owing to these options, our network reaches the better performance with accuracy of 95.7%.

As can be seen from these function curves in Fig 6 (a), other well-known loss functions are also more robust to outliers than *L2* loss. The comparison between the error rate of these models, between which only the chosen loss functions are different, reveals the alpha loss we designed can acquire similar performance of other well-known robust loss functions, and even better, shown in Fig 6 (b). More specifically, our alpha loss achieves the strong performance similar to *log-cosh* loss, while maintains a simpler calculation. At the same time, our alpha loss is obviously better than the quantile loss, with a higher precision advantage of about 2% .

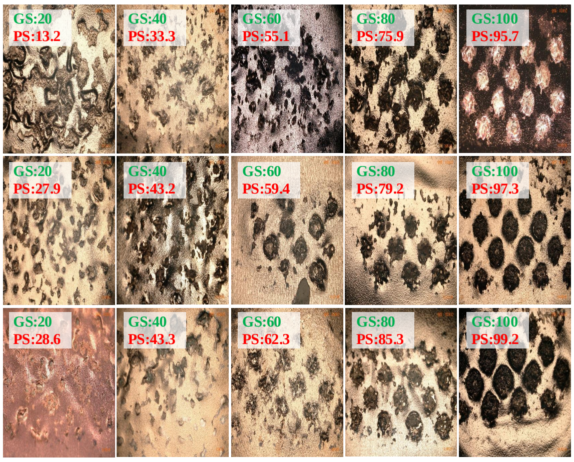


Figure 7: The testing results obtained by our proposed model. The ground truth score (GS) labelled by experts and the prediction score (PS) are posted in the upper corner of the sample images.

Figure 7 shows some typical results predicted on the test dataset by our proposed residual attention network. It demonstrates that our approach can evaluate the quality of welding joints with satisfying performance, robust to variations in brightness, texture and shape. Promising prediction results are achieved even when the imperfections of joints is cluttered.

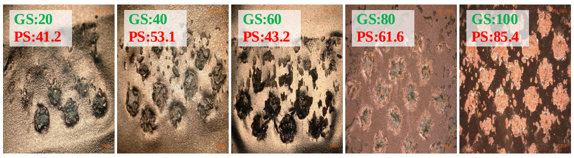


Figure 8: Failed cases, whose difference between their predicted scores and labelled values exceeds 10.

However, there are still a small amount of failure cases. Typical failures are presents in Figure 8. We speculate that these errors are due to the lack of the similar training samples and the useful features are not extracted. Because of this, the network only notices the texture of joints and lacks global information. We think these failure cases might be solved by increasing the similar training samples in the future.

5 Conclusion

In this work, a novel CNN-based network for automatic welding joints inspection is proposed. The proposed method is practical and can learn features efficiently from limited dataset due to the resultful residual attention network, which has two dedicated branches. Benefitting from the attention branch and the α loss function, our network converges faster with lower risk of over-fitting during training. The performance of the network we design is evaluated on a dataset consisting of images of welding joints with their score labelled by experts manually. We achieve satisfying results on visual inspection for the welding joints by predicting quality scores. Besides, we believe that the approach of automatic inspection with deep learning techniques also has potential application prospects in other different industrial automation domains.

6 Acknowledgments

This work was supported by the National Key R&D Program of China (No.2016YFB1001001), and the National Natural Science Foundation of China (No.91648121, No.61573280, No.61603022).

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton: "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems.* 2012.

[2] C. Simion: "Assessing student capability to visually inspect welded joints using attributive MSA technique," *Academic Journal of Manufacturing Engineering* 15.4 (2017).

[3] C. Szegedy, V. Vanhoucke, S. Ioffe, et al.: "Rethinking the inception architecture for computer vision," *In CVPR,* 2016.

[4] D. Racki, D. Tomazevic, and D. Skocaj: "A compact convolutional neural network for textured surface anomaly detection," *2018 IEEE Winter Conference on Applications of Computer Vision.* 2018.

[5] F. Wang, M. Jiang, C. Qian, et al.: "Residual attention network for image classification," arXiv preprint arXiv:1704.06904 (2017).

[6] K. Chen, J. Wang, L. C. Chen, et al.: "ABC-CNN: An attention based convolutional neural network for visual question answering," *arXiv preprint arXiv:1511.05960* (2015).

[7] K. He, X. Zhang, S. Ren, et al.: "Deep residual learning for image recognition," *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2016.

[8] K. Xu, J. Ba, R. Kiros, et al.: "Show, attend and tell: Neural image caption generation with visual attention," *International conference on machine learning.* 2015.

[9] L. Chen, H. Zhang, J. Xiao, et al.: "Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning," *In CVPR,* 2017.

[10] S. Faghih-Roohi, S. Hajizadeh, A. Núñez, et al.: "Deep convolutional neural networks for detection of rail surface defects," *IJCNN.* 2016.

[11] W. Chen, W. Xiong, J. Cheng, et al.: "Robotic Vision Inspection of Complex Joints for Automatic Welding," *17th International Conference on Computer and Information Science.* 2018.

[12] Z. Song, Z. Yuan, and T. Liu: "Residual Squeeze-and-Excitation Network for Battery Cell Surface Inspection," *In MVA*, Tokyo, Japan, 2019.