Few-shot Learning Combine Attention Mechanism-Based Defect Detection in Bar Surface

Qianwen LV[1] and Yonghong SONG[2]*

1) School of Software Engineering, Xi'an Jiaotong University, Xi'an, 710049 China.
2) School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an, 710049 China.

(*Received on October 30, 2018; accepted on January 10, 2019*)

Defect detection on bar surface is a challenging task due to the complex and variable bar surface conditions. Traditional pattern recognition methods are widely used to detect defects in the industry, however most of existing methods are not very universal for all kinds of defects. Meanwhile because of the limited number of defective samples, traditional deep learning methods are not very effective in practice. This paper addresses these issues and proposes a novel few-shot learning method which combines with attention mechanism. Our method is built by a Convolutional Neural Network (CNN) which extracts image features, and a Relation Network (RN) which calculates the similarity score between a pair of images, predicts image categories through similarity scores. Firstly, in order to extract more effective and discriminative features, we introduced Squeeze-and-Excitation Networks (SENet) as an attention module into our method which can enhance effective features and weaken invalid features. Secondly, unlike traditional object detection techniques which mainly focus on foreground information, background information is also necessary in our method, because we need to utilize background information to distinguish pseudo and real defects. So in our method, we replaced Max-Pooling with Mean-Pooling. Finally, in order to solve the low efficiency of parameter update caused by sharp dropping of loss function values on our dataset, we use L1Loss and BCELoss to replace Mean square error loss function. Experiment results show that the proposed method can achieve an average accuracy rate of 97.25% on our data set, increased by 7.92% compared with state-of-the-art.

KEY WORDS: few-shot learning; bars; attention mechanism.

1. Introduction

The iron and steel industry is a strategic industry of all industrialized countries in the world.[1] As the largest steel producer and consumer country in the world, the importance of steel industry is self-evident. However, due to various reasons such as the production of raw materials, and production processes, it is inevitable that there will be some defects in production process of bar steel. Due to technical difficulties, high costs and related hardware and software technologies for bar surface defect detection are still not complete, the defect detection of bar still relies mainly on the human eye observation method,[2] which has many drawbacks.

Deep learning is a machine learning theory that is increasingly popular nowadays.[3] Unsupervised or weakly supervised methods don't require a large number of samples, which are more suitable for the scarcity of samples in the industrial field theoretically. *e.g.* based on restricted Boltzmann machines, deep belief network,[4] stacked automatic encoder[5] and so on. Deep learning method has achieved the better results in defect detection on fabric surfaces. In 2017, the SDA[6] network was applied to industrial field which detection accuracy rate reached 96%, and the average classification accuracy rate reached 95.26%. The FCSDA network[7] proposed in the same year is a method that combination of SDA and fisher criterion. The fisher boundary analysis[8] is added to the loss function of the traditional SDA network as a regular term, and the detection accuracy rate reaches 95.83%. The application of FCSDA[9] in radio fingerprint feature extraction has also achieved an average recognition rate of 85.13%, which is 7.49% higher than the traditional SDA. The performance of unsupervised methods and weakly supervision methods are difficult to meet the needs of practical applications.

In addition, it is very difficult to collect enough useful samples on a real steel production line. The traditional supervised deep learning method requires a large number of labeled samples which cannot be directly applied in the field of bar surface defect detection. However, few-shot learning network[10] has low requirements on the amount of samples, which learns to learn a deep distance metric to compare a small number of images within episodes, each of which is designed to simulate the few-shot setting. From the above analysis, few-shot learning[10] are more suitable for our situation than traditional deep learning methods.

Few-shot learning[10] aims to recognize novel visual categories from very few labelled examples. There are four

* Corresponding author: E-mail: songyh@mail.xjtu.edu.cn

DOI: https://doi.org/10.2355/isijinternational.ISIJINT-2018-722

main methods in the field of few shot learning. The first is based on fine tuning, which obtains a certain amount of annotation data and then fine-tunes based on a base network. It requires a small train set, but it is not very generalized. The second is a graph based neural network,[14] which has a high precision but is not very suitable for dataset with only has few categories. It can be regarded as the extension of the three networks, and it is an innovation to solve these problems with the graph. The third is based on meta-learning which has good generalization and can achieve good performance on similar dataset, but its network architecture is very complex, training is time consuming, not suitable for the requirements of our dataset. The Memory-Augmented Neural Networks[15] is an LSTM-based model. The innovation is that the network takes the predicted value of the previous batch as input, and adds external memory to store a previous batch, so that the subsequent input can obtain related images through external memory for better prediction. The goal of the optimizer learning[16] is to learn an update function or update rule of a model parameter to solve the failure of the gradient-based optimization algorithm under a small amount of data. The last Model-Agnostic method[17] is similar to the previous one, but it is not an update function or an update rule for learning model parameters. The fourth is based on Metric, such as Siamese Networks[11] which proposed in 2015, Matching Networks[12] which proposed in 2016, and Prototypical Networks[13] which proposed in 2017. The input of Siamese Networks is a pair of samples, they are mapped to the target space using specific functions, and the Euclidean distance is calculated for similarity comparison in the target space. By minimizing the loss function values of a pair of samples which belonging to the same category, maximizing the loss function values of samples which belonging to different categories to optimize the model continuously, the purpose is to get a set of parameters to make samples belonging to the same class similarity measure smaller and not belong to the same class similarity measure is larger. Matching Networks is similar to Siamese Networks expect that it constructs mapping function using LSTM. Prototypical Networks need to learn a metric space, which regards the mean center of each category as a prototype, which calculates the distance between the test cases and the prototypes to get the category of the test cases. These network architecture are simple relatively, training is not time-consuming, and the precision is high relatively, which is adapting to the requirements of our dataset. So we chose the method in [10] which belongs few-shot learning based metric.

This paper presents few-shot learning method combining attention mechanism for defect detection on bar surface. The paper comprises three main contributions:

1. Firstly, feature maps are very important for our method because our model predicts image labels mainly depends on feature maps. In order to extract more effective and discriminative feature maps, the attention mechanism is introduced into our method, which can obtain more discriminative features for calculating similarity scores in the next step. Specifically, we use SENet as an attention module, by embedding SENet in the convolutional layer of CNN to get more effective feature maps.

2. Secondly, unlike traditional object detection which mainly focus on foreground information of an image, defect detection on the surface of bars also requires background information, because we need utilize background information to distinguish pseudo and real defects. In order to retain more background information, we replaced Max-Pooling with Mean-Pooling. Although Max-Pooling is commonly used in most popular deep network, it only selects the nature of the maximum response and does not apply to our data set.

3. Finally, this paper proposes to use L1Loss and BCELoss to replace Mean square error loss function. L1Loss can control the sharp drop of loss function values. BCELoss can restrain the two-class problem better.

The paper is structured as follows: Section 2 presents our method, introduces our improvement points compared to the previous network architecture. Section 3 proves the significance and effectiveness of our method through experiments. Section 4 summarizes the paper.

2. Method

The base network uses cvpr2018's few-shot learning network,[10] and the original network architecture is shown in **Fig. 1**. The lower part which has four convolutional blocks is a network of CNN for extracting features, the upper half is a Relation Network (RN) for calculating relation scores.

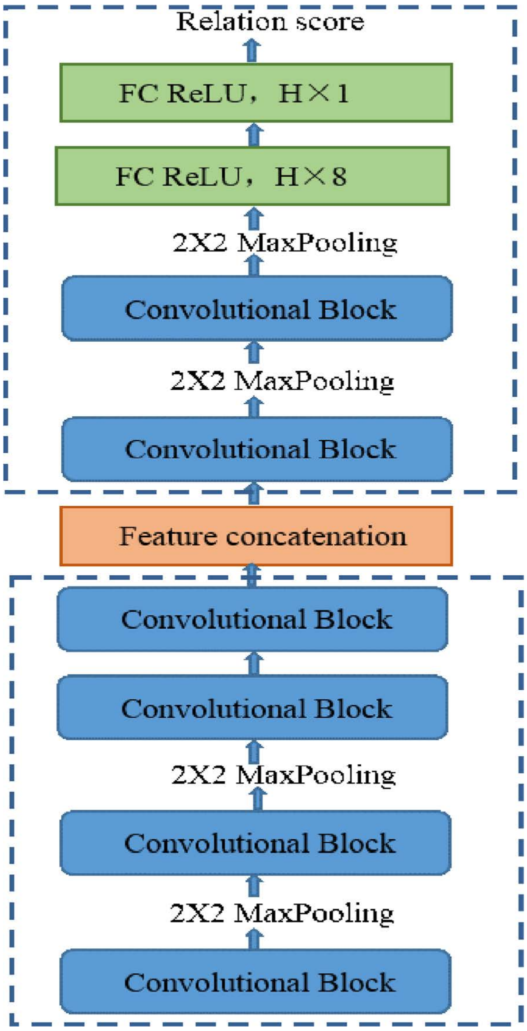

Fig. 1. Architecture of original few-shot learning network. (Online version in color.)

The CNN includes four layers, each layer consists of Rectified Linear Units (ReLU), Batch normalization (BN), 3×3 convolution kernel and a 64 channel filter as shown in **Fig. 2**, of which the first two layers add the Max-Pooling feature pooling operation to reduce the dimension.

The biggest difference between few-shot learning[10] and other traditional method of few-shot learning is that it does not use a fixed measure (such as Euclidean distance, cosine similarity, *etc.*) to measure the similarity of each feature pair, but to use a deep network to learn a similarity measure.

After extracting features by CNN, these features are cascaded, and then these feature pairs are fed into the RN to calculate the relation scores. Predicting the label according to the relation score, minimizing the mean square error between the theoretical label and the predictive label to optimize the parameters of entire network.

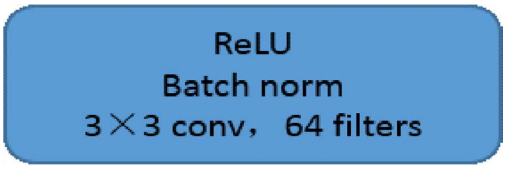

Fig. 2. The consist of each Convolutional Block. (Online version in color.)

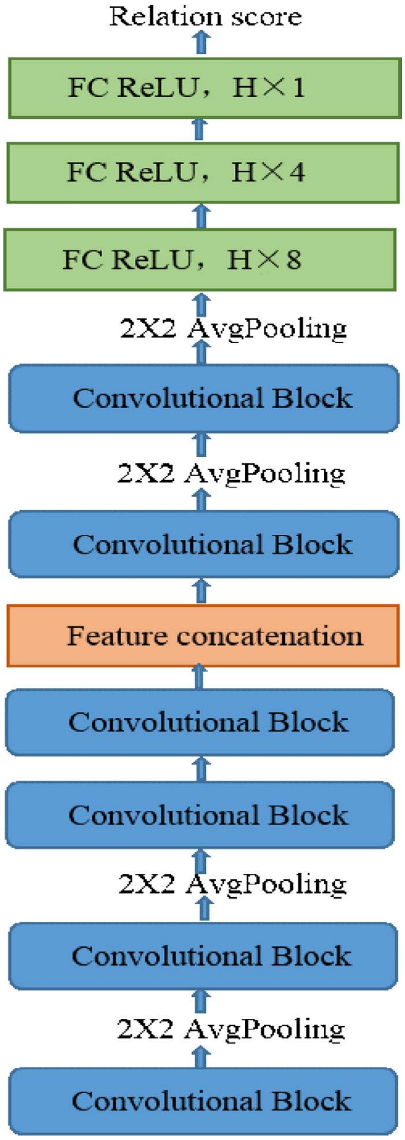

Fig. 3. Architecture of improved network. (Online version in color.)

Inspired by Few-shot learning,[10] our network architecture is similar to it, we first use CNN to extract features, and then use RN to calculate relation scores. In order to extract more effective and more discriminative features, this paper introduce attention mechanism into CNN. Through the attention mechanism, we can get the importance of each dimension feature, which means we can enhance effective features and weaken invalid features. Unlike traditional object detection, defect detection on the surface of bars also requires background information. So we change Max-pooling to Mean-pooling in order to retain more background information at pooling. In Relation Network, we replace the Mean square error loss function with the fusion of L1Loss and BCELoss. L1Loss can represent the differences between different categories very well, it can also prevent the sharp decline in loss function values. At the same time, we add a new layer of full-connection layer to the RN in order to alleviate the problem of information loss caused by large dimension reduction. The improved network architecture is shown in **Fig. 3** and each convolutional block is as shown in **Fig. 4**.

2.1. Attention Mechanism

In the part of extracting feature by CNN, we introduce the attention mechanism to make the feature more efficient. We use Squeeze and Excitation network (SENet)[18] to generate attention map. SENet mainly includes two operations, "Squeeze" and "Excitation" as shown in **Fig. 5**.

The "Squeeze" part means that compresses the feature map of C-dimensional H×W through global pooling to a real sequence of C-dimensional 1×1; "Excitation" is dimension-reduced and dimension-up based on the external parameter R, and finally a new C-dimensional 1×1 is obtained. This sequence of real numbers of 1×1 can be

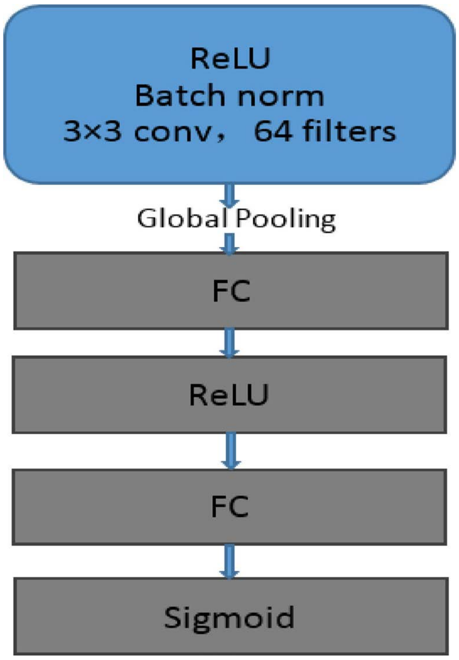

Fig. 4. The consist of each Convolutional Block. (Online version in color.)

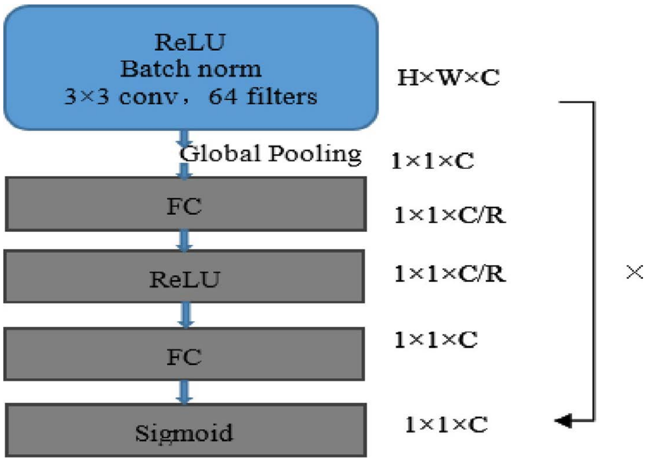

Fig. 5. Architecture of SENet. (Online version in color.)

considered as the degree of importance of each dimension feature. Finally, the C-dimensional 1×1 sequence of numbers is multiplied by the original C-dimensional H×W feature map to obtain a new set of feature maps.

We add the SE module to the end of each layer of CNN convolution block, and apply the SE module to the feature map of each layer to get the attention feature map. We can believe that feature extraction on the attention map has better performance than feature extraction directly on the original image. We regard the attention map as one step of preprocessing, in order to get a more significant image.

2.2. Selection of Loss

The original network[10] use the MSE mean square error loss function as its loss function, although the commonly used loss function is mean square error loss, but obviously, this is unreasonable for our data set. Because our data have only two classes, namely defective and non-defective, the representation in the network usually is 0 and 1, and the predicted relation scores is certainly less than or equal to1. According to the equation of MSE:

$$\text{MSELoss}(x_i, y_i) = (x_i - y_i)^2 \quad \text{(1)}$$

The difference between x_i and y_i must be less than 1. The loss value will be very small after squared. After a large number of iterations, the loss value will decrease sharply, which will not be conducive to updating parameters. So we use L1Loss and BCELoss to replace mean square error loss function, According to the formula of L1Loss:

$$\text{L1Loss}(x_i, y_i) = |x_i - y_i| \quad \text{(2)}$$

We can find that L1Loss not only expresses the difference between x_i and y_i very well, but also does not cause a sharp drop in the value of the loss function because it has no square operation.

BCELoss is the cross entropy used for the two classification. Because there are only positive and negative samples, and the probability sum of them is 1, then only one probability needs to be predicted, so it can be simplified:

$$\text{BCELoss}(x_i, y_i) = -w_i\left[y_i logx_i + (1 - y_i)\log(1 - x_i)\right] \quad \text{(3)}$$

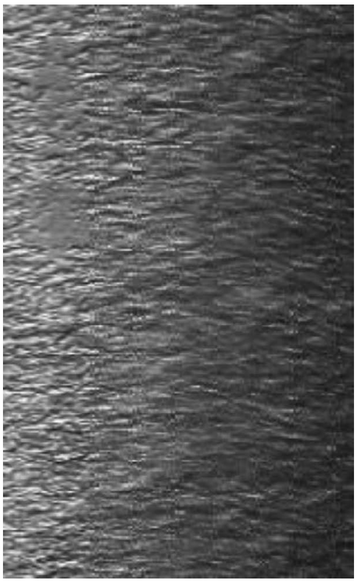

Fig. 6. Iron oxide pseudo-defect.

In which x_i represents the probability that the i sample is predicted to be positive, y_i represents the label of the i sample, and w_i represents the weight. The final loss function can be represented as fellow:

$$\text{Loss}(x_i, y_i) = |x_i - y_i| - w_i\left[y_i logx_i + (1 - y_i)\log(1 - x_i)\right] \quad \text{(4)}$$

2.3. Selection of Pooling

The goal of Max-pooling is to get the largest feature point in the neighborhood. When network forward, it only needs to take the largest value in the window. When network backward, it puts the current value to the previous maximum position, and the other three positions or more are set to 0.

The goal of Mean-pooling is to averages the feature points in the neighborhood, assuming that the window size of the pooling is 2×2. The 2×2 window average is not coincident on the output of the previous convolution when network forward. Finally this value is divided into four equal parts and put it in the front 2×2 grid when network backward.

In our dataset, both the background and the object are equally important, because sometimes we need to judge whether the current sample is a real defect or a pseudo defect based on the background.

As shown in **Fig. 6**, iron oxide pseudo-defect is a pseudo-defect that is easily confused by computers. Sometimes our model finds a local maximum response point and thinks that it is a real defect, but in human eyes, it is easy to find that the local maximum response point is not the global maximum response. Because the human eye receives global pixel information. This local maximum response point is surrounded by such similar points, so it is easy for the human eye to distinguish that it is a pseudo defect.

The Max-Pooling used by the original network[10] to extract feature maps in CNN will lose most of the background information. Although most of the popular networks use Max-Pooling, the background information is lost on our dataset, which leads to inefficient feature extraction. So we replaced the Max-Pooling with Mean-pooling in order to keep more image information.

3. Experiments

3.1. Dataset and Evaluation Metrics

1) Dataset: The datasets used in this paper are collected from the production line through CCD cameras, the collected images are shown in **Fig. 7**. In this image, the black area in the upper right corner is a scar.

However, it is obviously unreasonable that fed the original image into network directly which will not only emerge a large number of redundant images, but will also directly affect the overall accuracy of the network. So we randomly crop the original image into 56 × 56 small blocks. Specifically, in the data preparation stage, we generate a series of small batches by random cropping and edge gradient feature based cropping on the original image which selected from the original image obtained at the production line. This process is very fast. According to statistics, each cutting operation only takes 0.01 ms. In the process of edge gradient-based clipping, because these samples usually have obvious edge features, we use the Sobel operation in order to get pseudo-defective and defective samples. The purpose of random clipping is to get a non-defect samples.

These batches may be non-defective, defective or pseudo defect. The defective batches are shown in **Fig. 8**. (a)–(e) below, the non-defective batches is shown in (f) below and the pseudo defect batches is shown in (g)–(i).

We obtained 1 700 defective batches and 1 800 non-defective batches through random cropping in original images. At the same time, we collect 1 743 serious misdirected black spot pseudo defects and oxide pseudo defects, add them to non-defective samples. The goal of our method is only to mark the defective batches so we divide the dataset into two categories, positive class and negative class. There are defective batches in positive samples, non-defective and pseudo defects in non-defective.

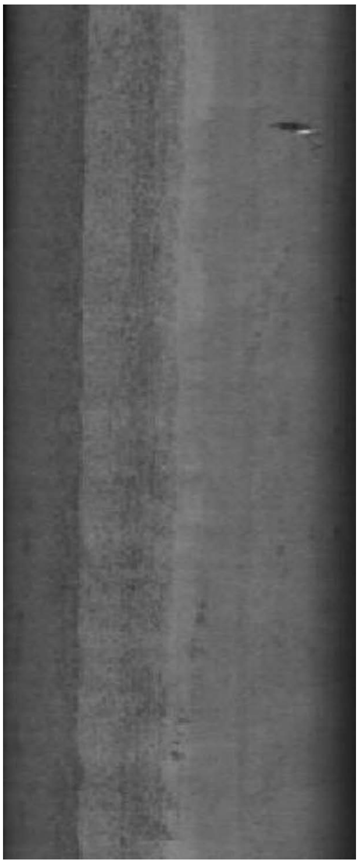

Fig. 7. The original image.

Before the data augmentation, our training set includes 100 defective samples, 150 non-defective samples and 150 pseudo-defect samples. The test set includes the remaining 1 600 defective samples, 1 650 non-defective samples and 1 593 pseudo-defective samples. And then augmentation is performed for the original train set: flipping, adding noise (0.1) and Gaussian filtering ([3, 3]) which make train set increase from 400 to 1 600. Our augmentation operation is based on the fact that each data set has been divided. For example, if the training set includes 100 defective samples before the augmentation, then we flip them, add noise and perform Gaussian filtering. Finally 400 samples after the augmentation were obtained. Our data set is composed as shown in **Table 1**.

In the training set, we divide our data into two categories, which are defective samples and non-defective samples (including both non-defective and pseudo-defective in non-defective samples). Although it seems that the total number of total is enough, but the defective samples include four subclasses, such as scars, roll-marks, scratches and folding. If a shallow two-class categorization network is used, it

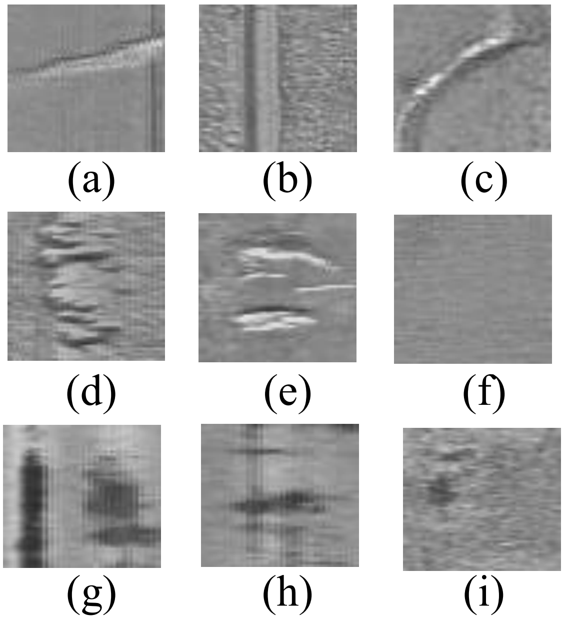

Fig. 8. Image batches with defects and image batch without defect.

Table 1. Composition of the data set.

		Train-set	*Test-set*	*Total*
defective		400	1 600	2 000
non-defective	non-defective	600	1 650	2 250
	pseudo-defective	600	1 593	2 193
Total		1 600	4 843	6 443

may confuse defective and pseudo-defective samples. If a deep two-class categorization network is used, it may cause serious over-fitting because of that the number of samples in each subclass is not enough.

2) Evaluation Metrics: A group of metrics including accuracy (ACC), true positive rate (TPR), and false Error rate, were employed to quantify the detection accuracy. The definition of ACC, TPR, and false Error rate are described as formula (5)–(7).

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (5)$$

$$\text{TPR} = \frac{TP}{TP+FN} \quad (6)$$

$$\text{Error rate} = \frac{FP}{TP+FP} \quad (7)$$

True positive (TP) represents the number of image batches that are detected as defective correctly, true negative (TN) represents the number of image batches that are detected as non-defective correctly, false positive (FP) represents the number of image batches that are detected as defective incorrectly, and false negative (FN) represents the number of image batches that are detected as non-defective by mistake.

Accuracy represents the overall accuracy of the model we have trained. The case where the real defect batches are divided into positive samples and the pseudo-defects and non-defective batches are divided into negative samples are considered correct. TPR can also be called the detection rate, which means that in all positive samples, the proportion of samples that our trained model can correctly classify. The error rate can also be called the false detection rate, which means the proportion of dividing the pseudo-defective and non-defective batches in the positive sample in the classification result of our model.

Accuracy can better describe the overall classification ability of the model. The goal of our model is to increase the detection rate and reduce the false detection rate, so a balance needs to be obtained between the detection rate and the false detection rate. We can not blindly improve the detection rate and ignore the improvement of false detection, because this is not applicable in the real situation.

3.2. Network Architecture and Parameter Setting

Our method mainly consists of two networks, the first is CNN for generating feature maps, and the other is relation network (RN) for obtaining relation scores. There are four convolutional blocks in CNN. Each convolutional block contains a 64-filters 3×3 convolution, a batch normalization, a ReLU nonlinearity layer and a SE model. Each SE model contains one global-pooling, two fully-connected layers, one ReLU nonlinearity layer and one Sigmoid function. The first two blocks of CNN also contains a 2×2 mean-pooling layer. There are two convolutional blocks and three full-connected layers in RN. Each convolutional block is a 3×3 convolution with 64 filters followed by batch normalization, ReLU nonlinearity and 2×2 mean-pooling. The dimension of the three full-connected layers is 8, 4 and 1 respectively. We set learning rate is 0.001, and use R=16 in SE model. Select ten images from the positive and negative sample sets of the training set for each training.

3.3. Component Analysis of the Proposed Model

We investigate the effect of each component of our model by conducting several analytic experiments. In **Table 2**, we list the results of each component in the proposed method.

Baseline is the experiment result of using the network structure in Few-shot learning[10] on our dataset, changing the sum of classes to 2, because our data set only needs to be divided into two categories. Mean-Pooling means that we replaces Max-Pooling in both CNN and RN networks with Mean-Pooling. Loss means that we replaces MSELoss with L1Loss and BCELoss, and SENet means that we adds SE modules to each layer of CNN. ALL means that we add all above operations, at the same time we changes the two-layer fully-connected layer in the RN network to the three-layer fully-connected layer.

From the above table we can see that Mean-pooling retain more information to a certain extent, which improves accuracy by 2.54%. When we replace MSELoss with L1Loss and BCELoss, the accuracy improved from 91.87% to 92.86%, we can believe that L1Loss guarantees the loss function value does not drop sharply, and BCELoss limits the accuracy of the two classification problem. The addition of the SE module has improved the accuracy of 4.34%. Finally although the addition of the new full-connected layer only increases the accuracy from 97.2% to 97.25%, we are more concerned about the true positive rate in the field of bar surface defect detection, that is, how many real defects can be detected. From the above table, we can find that this method is the only one that true positive rate is higher than accuracy in all methods.

3.3.1. Different Loss Function in Relation Network

In this part, we compare the loss functions commonly

Table 2. Component Analysis of the Proposed Model.

	ACC	*TPR*	*Error rate*
Baseline[10]	89.33	89.31	2.01
Mean-Pooling	91.87	91.74	1.66
Mean-Pooling+Loss	92.86	92.61	1.7
Mean-Pooling+Loss+SENet	97.2	97.16	1.41
ALL	**97.25**	**97.26**	**1.35**

Table 3. Loss function compare of the Proposed Model.

	ACC	*TPR*	*Error rate*
L1Loss	93.16	92.22	2.1
SmoothL1Loss	92.85	89.1	2.4
BCELoss	93.65	92.5	1.9
BCEWithLogitsLoss	92.59	91.6	2.1
L1Loss+ BCELoss	**93.9**	**92.55**	**1.8**

used in the two-class problem. In **Table 3**, we list the results of each loss function in the proposed method. L1Loss means absolute value between two numbers which can represent the accuracy of classification. SomoothL1Loss is a special case of L1 loss, the error is square loss on (−1, 1), and the other case is L1 loss. In our dataset, the theoretical label is 0 or 1, and the prediction label is a decimal between [0, 1], so the difference of two numbers must be between [−1, 1]. That means if use SmoothL1Loss function, it's essentially a square error. From the Table 3, we can find that L1Loss function shows the better performance. We can believe that the square operation in the loss function can't bring improvement of performance on our dataset.

The difference between BCELoss function and BCEWithLogitsLoss function is that the latter is composed of a sigmoid layer and a BCELoss function, in order to solve the problem of derivation instability in BCELoss. However in our method, the last layer of the relational network has included a sigmoid layer to get the relation scores distributed between (0, 1), so the sigmoid in the BCEWithLogitsLoss function is not necessary. If we add one more layer of sigmoid in the network, it may lead to an error in the meaning of the relation score. The result of Table 3 also proves our analysis, compared with BCEWithLogitsLoss, BCELoss has better performance.

Among the four loss functions, L1Loss and BCELoss have the top two best performance, so it is easy to think about that if combine the two loss functions, will we achieve better results? Experimental results prove that we speculate, the combination of L1Loss and BCELoss shows the best performance which reaches 93.9%, beyond all the above cases.

It should be noted that the experiment here does not introduce the attention mechanism module, but only the loss function replacement experiment based on the base network. The fully connected layer dimensions of the relational network are 8, 4 and 1 respectively.

3.3.2. Different Value of R in SE Models

In the previous section, it was mentioned that the excitation part of SENet was a dimension reduction operation based on an external parameter R. In order to determine the optimal value of R, we conduct three groups of contrast experiments, R=8, R=16, and R=32, respectively. R is setting 8 means that the 64-dimensional feature map is first reduced to 8 dimensions, and then upgraded to 64 dimensions. R is setting 16 means that the 64-dimensional feature map is first reduced to 16 dimensions, and then upgraded to 64 dimensions. R is setting 32 means that the 64-dimensional feature map is first reduced to 32 dimensions, and then upgraded to 64 dimensions. We list the results of each situation in **Table 4**. From the experimental results, R=16 achieved the best accuracy and true position rate, R=32 achieved the lowest error rate.

Table 4. SENet parameter value.

	ACC	*TPR*	*Error rate*
R=8	93.8	94.1	1.4
R=16	**97.25**	**97.26**	**1.37**
R=32	96.1	95.23	1.22

From Table 4, we can find that the performance does not improve monotonically with the increase of R, this is consistent with the expression in SENet.[18] If R is too small, the dependence between channels is ignored, direct compression to 8 dimensions from 64 dimensions will definitely lose a lot of information. However the value of R is too large, resulting in over-fitting of the dependencies between channels. We found that setting R=16 achieved a good trade-off between accuracy and error rate, we used this value for all experiments.

It should be noted that the experiment here adds the SE module to all four layers of CNN, and the loss function uses L1Loss and BCELoss. Each training iteration selects 10 samples from each of the two classes. The fully connected layer dimensions of the relational network are 8, 4 and 1 respectively.

3.3.3. Different Locations of SE Models

The discriminative and excellent features will directly affect the result of classification, so the location of SE module is very important. SE module is very flexible module which can be added to any positions in our network, and different positions have different effects. Usually, the first layer convolution of CNN is extracted from edges or color feature maps, and as the number of convolution layers increases, it is a semantic combination of high–level features. Considering that our method introduces attention mechanisms for extracting more effective features, so we introduce it to CNN. CNN includes four convolution blocks, in order to verify which convolution layer of SE module added to the CNN will get the best performance, we make seven groups of contrast experiments, the first layer of CNN, the second layer, the third layer, the fourth layer, the first two layers, the second two layers and all four layers. Firstly, we add the SE modules on each layer respectively in order to determine which layer added the attention mechanism alone can get the best performance. Secondly, the first and second layers of CNN we used belong to the lower layer. We can obtain edges, colors or other morphological features. The third and fourth layers of CNN belong to the upper layer, and we can obtain advanced features. Therefore, we add SE modules to the upper and lower layers respectively to determine whether the SE modules will perform better at upper levels or perform better at lower levels. Finally, we not only introduce attention mechanism in the extraction of the low-level features, but also introduce it in the extraction of high-level features to determine whether it will achieve better results if all layers of CNN are added to the SE module. As shown in **Fig. 9**, corresponding to the CNN network architectures in the case of 7 in **Table 5**, respectively.

We list the result of each situation in Table 5. From the experimental results, we can get the best performance by adding SE modules to all four convolution layers of CNN with an accuracy of 97.25%, a true position rate of 97.26%, an error rate of 1.35%, which is much higher than other cases. We can believe that the attention mechanism not only shows great advantages in extracting the low-level features, but also has better performance in extract-

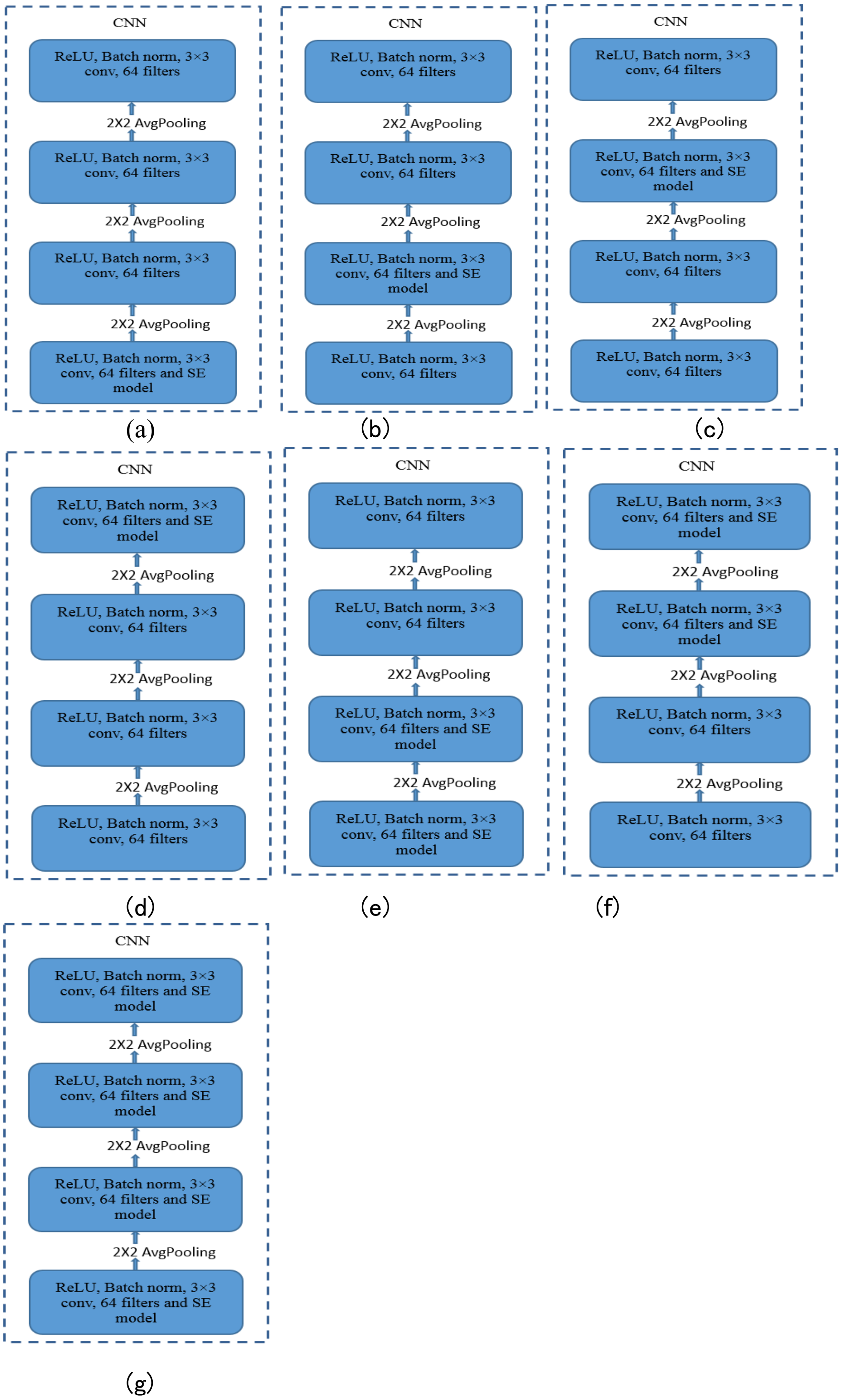
CNN
ReLU, Batch norm, 3×3 conv, 64 filters
2X2 AvgPooling
ReLU, Batch norm, 3×3 conv, 64 filters
2X2 AvgPooling
ReLU, Batch norm, 3×3 conv, 64 filters
2X2 AvgPooling
ReLU, Batch norm, 3×3 conv, 64 filters and SE model
(a)
CNN
ReLU, Batch norm, 3×3 conv, 64 filters
2X2 AvgPooling
ReLU, Batch norm, 3×3 conv, 64 filters
2X2 AvgPooling
ReLU, Batch norm, 3×3 conv, 64 filters and SE model
2X2 AvgPooling
ReLU, Batch norm, 3×3 conv, 64 filters
(b)
CNN
ReLU, Batch norm, 3×3 conv, 64 filters
2X2 AvgPooling
ReLU, Batch norm, 3×3 conv, 64 filters and SE model
2X2 AvgPooling
ReLU, Batch norm, 3×3 conv, 64 filters
2X2 AvgPooling
ReLU, Batch norm, 3×3 conv, 64 filters
(c)
CNN
ReLU, Batch norm, 3×3 conv, 64 filters and SE model
2X2 AvgPooling
ReLU, Batch norm, 3×3 conv, 64 filters
2X2 AvgPooling
ReLU, Batch norm, 3×3 conv, 64 filters
2X2 AvgPooling
ReLU, Batch norm, 3×3 conv, 64 filters
(d)
CNN
ReLU, Batch norm, 3×3 conv, 64 filters
2X2 AvgPooling
ReLU, Batch norm, 3×3 conv, 64 filters
2X2 AvgPooling
ReLU, Batch norm, 3×3 conv, 64 filters and SE model
2X2 AvgPooling
ReLU, Batch norm, 3×3 conv, 64 filters and SE model
(e)
CNN
ReLU, Batch norm, 3×3 conv, 64 filters and SE model
2X2 AvgPooling
ReLU, Batch norm, 3×3 conv, 64 filters and SE model
2X2 AvgPooling
ReLU, Batch norm, 3×3 conv, 64 filters
2X2 AvgPooling
ReLU, Batch norm, 3×3 conv, 64 filters
(f)
CNN
ReLU, Batch norm, 3×3 conv, 64 filters and SE model
2X2 AvgPooling
ReLU, Batch norm, 3×3 conv, 64 filters and SE model
2X2 AvgPooling
ReLU, Batch norm, 3×3 conv, 64 filters and SE model
2X2 AvgPooling
ReLU, Batch norm, 3×3 conv, 64 filters and SE model
(g)

Fig. 9. CNN network architectures. (Online version in color.)

Table 5. Location compare of SENet.

	ACC	*TPR*	*Error rate*
Fig. 9(a)	93.55	93.65	1.57
Fig. 9(b)	94.2	94.02	1.45
Fig. 9(c)	89.9	88.1	1.52
Fig. 9(d)	91.6	90.95	1.6
Fig. 9(e)	94.4	94.32	1.65
Fig. 9(f)	93.4	93.15	1.4
Fig. 9(g)	**97.25**	**97.26**	**1.35**

ing high-level features. From the Table 5, we also can see that if only the SE module is added to the first two layers of CNN, that is, extract the low-level feature, the accuracy is 94.4%, and only add it to the last two layers which extract the high-level, the accuracy rate is 93.4%, the former is 1% higher than the latter. It can be proved that the SE module works better for the extraction of low-level features.

It should be noted that the experiment here sets the external parameter R is 16, and the loss function uses L1Loss and BCELoss. Each training iteration selects 10 samples from each of the two classes. The fully connected layer dimensions of the relational network are 8, 4 and 1 respectively.

4. Conclusions

In this paper, we solve the problem of limitation and imbalance of defective sample in the industry. Better training with few samples can achieve better performance. We embed the SENet as an attention module in CNN. Through the attention module, we can extract more effective and discriminative feature maps. The accuracy of using these features for similarity calculation is higher than that of direct extraction. In order to preserve the background information of the image, we change the pooling layer. At the same time, we determined the loss function through experiments and determined the parameters and position of the attention module through experiments. Finally, we improved the detection accuracy by adding a fully connected layer. In the experimental stage, we conducted a module-by-module experiment to prove that our method is more efficient. At the same time, conducting experiment to determine the optimal value of parameters.

We selected 1 600 samples for training and 4 843 samples for testing. The classification accuracy rate can reach 97.25%, the detection rate can reach 97.26%, and the false detection rate is only 1.35%. The results show that our method has improved accuracy relative to the base network on our dataset.

Acknowledgments

This work is partially supported by the Natural Science Basic Research Plan in Shaanxi Province of China (Program No. 2018JM6104) and the National Key Research and Development Project of China 2017YFB1301101.

REFERENCES

1) F. DuPont, C. Odet and M. Cartont: *NDT&E Int.*, **30** (1997), 3.
2) D. Chen: *Friend Sci. Amat.*, **01** (2013), 25.
3) J. Qu, X. Sun and X. Gao: *Foreign Electron. Meas. Technol.*, **08** (2016), 50.
4) G. E. Hinton, S. Osindero and Y. Teh: *Neural Comput.*, **18** (2006), 1527.
5) B. Schölkopf, J. Platt and T. Hofmann: Int. Conf. on Neural Information Processing Systems, MIT Press, Cambridge, MA, (2006), 153.
6) J. Jing, Y. Dang, Z. Su, P. Li and H. Zhang: *J. Electron. Meas. Instrum.*, **31** (2017), 1321.
7) Y. Li, W. Zhao and J. Pan: *IEEE Trans. Autom. Sci. Eng.*, **14** (2017), 1256.
8) S. Yan, D. Xu, B. Zhang, Q. Yang, H. Zhang and S. Lin: *IEEE Trans. Pattern Anal. Mach. Intell.*, **29** (2007), 40.
9) H. Jianhang and Y. Lei: *Pattern Recognit. Artif. Intell.*, **30** (2017), 1030.
10) F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr and T. M. Hospedales: Conf. on Computer Vision and Pattern Recognition, IEEE, New York, (2018), 1199.
11) G. Koch, R. Zemel and R. Salakhutdinov: Int. Conf. on Machine Learning, ACM, New York, (2015), 2252.
12) O. Vinyals, C. Blundell, T. Lillicrap and D. Wierstra: Advances in Neural Information Processing Systems, NeurlPS, La Jolla, CA, (2016), 3630.
13) J. Snell, K. Swersky and R. Zemel: Advances in Neural Information Processing Systems, NeurlPS, La Jolla, CA, (2017), 4077.
14) V. Garcia and J. Bruna: Int. Conf. on Learning Representations, ICLR, La Jolla, CA, (2018), 1.
15) A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra and T. Lillicrap: Int. Conf. on Machine Learning, ACM, New York, (2016), 1842.
16) S. Ravi and H. Larochelle: Int. Conf. on Learning Representations, ICLR, La Jolla, CA, (2017), 1.
17) C. Finn, P. Abbeel and S. Levine: Int. Conf. on Machine Learning, ACM, New York, (2017), 1126.
18) J. Hu, L. Shen and G. Sun: The IEEE Conf. on Computer Vision and Pattern Recognition, IEEE, New York, (2018), 7132.