Optics and Lasers in Engineering 122 (2019) 294–302

Contents lists available at ScienceDirect

Optics and Lasers in Engineering

journal homepage: www.elsevier.com/locate/optlaseng

Semi-supervised defect classification of steel surface based on multi-training and generative adversarial network

Yu He[a,b], Kechen Song[a,b,*], Hongwen Dong[a,b], Yunhui Yan[a,b,*]

[a] School of Mechanical Engineering & Automation, Northeastern University, Shenyang, Liaoning, China
[b] Key Laboratory of Vibration and Control of Aero-Propulsion Systems Ministry of Education of China, Northeastern University, Shenyang, Liaoning, China

ARTICLE INFO

Keywords:
Defect classification
Steel surface inspection
Semi-supervised learning
Generative adversarial network
Multi-training

ABSTRACT

Defect inspection is very important for guaranteeing the surface quality of industrial steel products, but related methods are based primarily on supervised learning which requires ample labeled samples for training. However, there can be no doubt that inspecting defects on steel surface is always a data-limited task due to difficult sample collection and expensive expert labeling. Unlike the previous works in which only labeled samples are treated using supervised classifiers, we propose a semi-supervised learning (SSL) defect classification approach based on multi-training of two different networks: a categorized generative adversarial network (GAN) and a residual network. This method uses the GAN to generate a large number of unlabeled samples. And then the multi-training algorithm that uses two classifiers based on different learning strategies is proposed to integrate both labeled and unlabeled into SSL process. Finally, through the multiple training process, our SSL method can acquire higher accuracy and better robustness than the supervised one using only limited labeled samples. Experimental results clearly demonstrate that the effectiveness of our proposed method, achieving the classification accuracy of 99.56%.

1. Introduction

Defect classification is a fundamental industrial inspection task, the aim of which is to identify the category of a defect to ensure the surface quality of steel products [1–5]. It is commonly performed manually in practice, which is very unreliable and time-consuming.

As the deep learning (DL) technology has exhibited remarkable performance in many vision tasks, with the aim of replacing manual inspection, there have existed many DL-based methods for automatic defect classification on the steel surface. For example, Zhou et al. [26] train a sequential structured CNN for defect classification. Li et al. [6] simplify the AlexNet [11] on parameters and kernels to reduce the demand for training data. Huang et al. [7] design a small FCN [12] to classify defects. Although these methods adopt various DL models for defect classification, their good results highly depend on a healthy supervised learning process. They have to use small but not very strong DL networks because of the lack of labeled defect samples. A deep network seems to be less robustness and even have the risk to be overfitting if training data is insufficient. Therefore, the main point of a supervised learning scenario is that the number of labeled samples is too small to train deep networks that have strong representation ability. The DL model without an adequate training can inaccurately represent the data distribution which may lead to defect misclassification in test time.

Since it is hard to gain enough labeled samples to support a supervised training, in this paper we establish a SSL defect classification framework, attempting to make use of unlabeled samples. Usually when labeled samples are limited, a SSL algorithm that exploits both labeled and unlabeled samples simultaneously can achieve good learning performance [8]. Thus, it is essential that applying SSL in defect classification to eliminate the expensive sample collection and manual labeling. Moreover with enough defect samples, DL networks can be better used in industrial inspection tasks and achieve high classification accuracy. However, in order to build such a SSL defect classification system, we have to address two major challenges as follows.

One is how to get new samples without additional collection. In general, a SSL system processes fewer labeled samples and larger unlabeled samples, which can alleviate the single demand of DL models for labeled ones. The effectiveness of the SSL methods as in [9–10] attributes to their easy access to real images from the Internet. Unlike these fields, defect images are usually captured on the steel surface by CCD cameras, which is too complex and time-consuming. And after a selection and cropping, only a few defect samples are available due to a defect rarely occur on the steel surface. So we prefer to find a simple way to generate realistic samples, rather than trying to collect more real ones. The recent successful DL framework generative adversarial network (GAN) offers a new way for sample generation. Through an unsupervised learning pro-

* Corresponding authors.
E-mail addresses: songkc@me.neu.edu.cn (K. Song), yanyh@mail.neu.edu.cn (Y. Yan).

https://doi.org/10.1016/j.optlaseng.2019.06.020
Received 8 May 2019; Received in revised form 15 June 2019; Accepted 21 June 2019

cess, a GAN receives real samples and random noise to generate fake samples similar to the real ones. In this SSL system, we adopt the deep convolutional GAN (DCGAN) [13] to generate defect samples. Based on a certain number of original samples, the DCGAN can generate considerable new ones through a competitive training process involving a pair of convolutional neuron networks (CNN). By the DCGAN, our SSL system can use enough samples to train powerful DL models for defect classification.

The other is how to include unlabeled samples into training. The samples generated by DCGAN are unlabeled, so a label assignment process is required before training with labeled samples. In other words, each GAN sample should be assigned the class label, usually a soft or pseudo one [9]. There are two common SSL algorithms on the processing of unlabeled samples, which can be used in DL classifiers. The first method is self-training as in [19], in which a classifier learns on its own and assigns the class label only according to its own predictions. The self-training is simple and easy to apply but highly depends on the initial training. If original samples cannot afford to make a DL network converge, the self-training method tends to offer incorrect label information for GAN samples and finally leads to misclassification. The second method uses co-training as in [20], which trains two classifiers on labeled samples and then uses the most confidence unlabeled samples to enhance the labeled ones. However, the co-training requires both classifiers to be trained on two different views, which is not easy to be satisfied for small-scale defect datasets. Therefore, combining the advantages of the self-training and co-training, we propose the multi-training algorithm for the SSL defect classification. This algorithm uses the multi-training of two different DL models, a categorized GAN and a residual network, which can not only refine by each other but also avoid the view partition.

To overcome the above-mentioned two challenges, this paper presents a SSL defect classification method, which employs GAN for sample generation and uses the multi-training of two classifiers for handling both labeled and unlabeled samples. In detail, we first train the cDCGAN on original labeled samples to produce enough unlabeled samples and meanwhile gain the class predictions. Next, a residual network is also trained initially on the original samples and makes predictions on the GAN samples. And then according to the class confidence predicted by each classifier, the high-confidence samples are labeled with the estimated classes and added in the training set. Finally, the residual network is retrained on the new training set until all the GAN samples are added into training. We evaluate the proposed method on the NEU-CLS defect dataset and the experimental results show that our method can achieve competitive classification performance even if the original samples are limited.

The main contributions of this work are summarized as follows:

1) We introduce a SSL framework for defect classification of steel surface. This novel framework adopts the GAN to generate samples, so avoiding the large collection of defect images;
2) We propose the multi-training algorithm of the cDCGAN and the residual network for SSL, which includes the unlabeled samples into training process and hence the improvement of classification performance;
3) We conduct extensive experiments on the NEU-CLS defect dataset and show that the effectiveness of the proposed method when original samples are limited

2. Related works

In this section, we briefly introduce the generative adversarial network and the SSL algorithms that can be used in DL networks.

2.1. GAN

GAN is the recent emerging DL architecture for semi-supervised or unsupervised learning [16]. Unlike other types of DL networks, the GAN learns around two sub-networks: a generator G and a discriminator D, which are different in network architecture. Thus, a GAN can be characterized by training these two networks in competition with each other. There is a classical analogy, apt for image data, to consider the G as a forger, and the D as the police. The forger produces fake images and disguises them as real ones. The police receives both real and fake images with the aim to distinguish the real images from the fake ones. The standard GAN is introduced in Fig. 1. Both networks are trained jointly and play against each other to reach their goals.

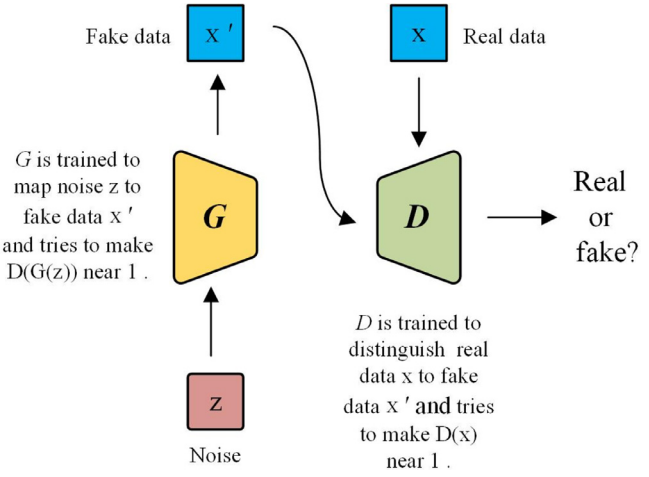

Fig. 1. The structure of a standard GAN.

In each training step, G produces fake data x' from a random noise z and D receives x' as well as the real data x to divide them into "real" or "fake". Both networks are updated repeatedly and the iteration stops until they reach a Nash equilibrium. In more detail, D and G are competitors in a minimax game with the following function:

$$\min_G \max_D V(G,D) = \mathbb{E}_{p_{data}(x)} \log D(x) + \mathbb{E}_{p_z(z)} \log [1 - D(G(z))] \quad (1)$$

where $\mathbb{E}$ is the empirical estimate of the expected value of the probability. G transforms z into $G(z)$, which is sampled from a noise distribution p_z, and the ideal p_z should converge to the real data distribution p_{data}.

2.2. SSL algorithms

Semi-supervised learning is useful for improving model performances when the target domain lacks of labeled samples. The labeled samples are always few in defect inspection tasks, but we use GAN to generate a lot of unlabeled ones. Therefore, in the case of unlabeled samples available, the key issue is how to include them into labeled ones for training DL networks. The self-training and co-training algorithms are quite suitable for integrating with the CNNs. The self-training trains a single classifier on labeled samples and predicts on unlabeled samples. The unlabeled samples that assigned labels according to the previous predictions are added in labeled samples (see Fig. 2a). The co-training trains two classifiers in different view and predicts on unlabeled samples respectively. Based on their predictions, a designed voting method is always used for label assignment (see Fig. 2b).

While the self-training and co-training algorithms are quite suitable for the DL classifiers, there are problems with all of them. The self-training a simple semi-supervised algorithm that can use one DL model for handling unlabeled samples. He et al. [14] use an auto-encoder to predict class information for unlabeled samples in the self-training manner. However, self-training requires a good initial training process that means the labeled samples are not too few. The co-training is a multi-view SSL algorithm that uses labeled samples to train two weak classifiers and then applies them in unlabeled samples to get high confident predictions by each view [15]. This strategy requires two sufficient and

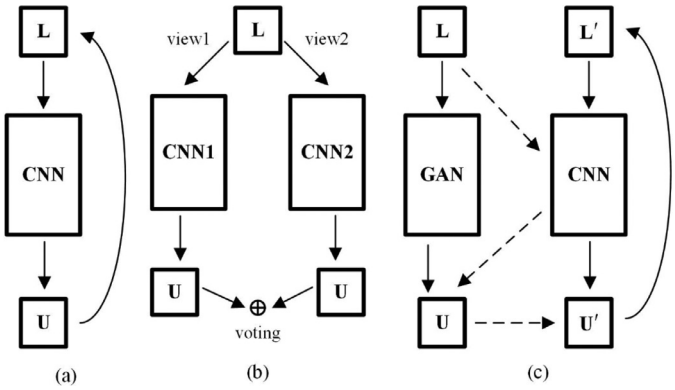

Fig. 2. The different SSL methods based on CNNs. (a) Self-training; (b) Co-training; (c) Multi-training.

redundant views, each of which is conditionally independent. However, these conditions are not easy to be satisfied in practice due to most datasets are described by one view. To extend the usage of co-training, Goldman and Zhou [17] propose a method that trains two classifiers in different supervised learning algorithms. Unfortunately, their method needs a cross validation process for unlabeled samples, which is too time-consuming. Zhou and Li [18] presents the tri-training algorithm that can be efficiently applied in DL models. The tri-training does not require the view partition but instead three classifiers. Inspired by these SSL methods, the multi-training algorithm can not only abandon the complex view partition for dataset, but also achieve strong classification ability through the multiple training process. Furthermore, we also integrate the sample generation into the multi-training, which makes the defect inspection system more compact and efficient (see Fig. 2c).

3. Network architecture

The SSL defect classification system consists of two DL networks: the cDCGAN and residual network that are regarded as classifiers for the multi-training process. In this section, we introduce the network architecture of these two classifiers.

3.1. cDCGAN classifier

The DCGAN is the deep convolutional GAN, which has been succeeded in sample generation [13]. In a standard DCGAN, the D and G are all the CNNs for representation learning. Through an unsupervised learning process, the DCGAN can generate a large number of unlabeled samples in a relatively small number of labeled samples. In this work, we not only use the DCGAN to produce defect samples but also modify its structure as one classifier for the multi-training. D can be seen as a binary classifier when the DCGAN finishes representation learning. Therefore, we replace the last layer of D with a softmax layer that outputs a $C+1$ prediction for each sample, where $C+1$ represents the C defect classes plus the "fake" class. It means that the "real" GAN samples are classified by defect classes after they have been distinguished from the "fake" ones.

The structure of the cDCGAN is shown in Fig. 3 and the detailed architecture parameters are summarized in Table 1. For each past, a 100-d random noise vector is fed into G and reshaped to $4\times4\times8$ using a linear function. The size of each input tensor is changed to $4\times4\times512$ by multiplying the mini-batch size. And then the tensors are enlarged by four deconvolutional layers, each of which has a kernel size of 5×5 and a stride of 2. Each deconvolutional layer is followed with a BN layer and a ReLU function except the last one that uses the tanh function. Finally, the GAN sample which has the same size as the input image is generated. Meanwhile, D receives the generated samples together with real ones as the input. Similarly as G, there are four convolutional layers in D for processing the tensors. These layers are also 5×5 in size with a stride of 2, each of which is equipped with a BN layer and a LeakyReLU function. To the end, D outputs a discrete probability distribution, $P = \{p_1, \ldots, p_C, p_{Fake}\}$, for each GAN sample.

3.2. Residual network classifier

Besides the cDCGAN, a residual network, regarded as the second classifier, is used in multi-training. The residual network is a famous CNN for image classification [23], whose structure differs from the CNNs used in cDCGAN. As described above, the CNNs of the cDCGAN are the sequential pipeline which only has one path for gradient propagation. But the residual network has multiple paths by the skip connection, so it is appropriate as another classifier of the multi-training. Considering that the original defect samples are limited, we cannot use too deep networks to get a poor initial training. The model resnet18 is selected as the second classifier in the SSL defect classification system.

The structure of the resnet18 is shown in Fig. 4 and the detailed architecture parameters are summarized in Table 2. This network consists of five residual blocks, which are connected by the skip connection. Expect for the first block which uses a single convolutional layer with large filter, the other blocks all have two convolutional layers with small filters. At the end of the last residual block, a global average pooling (GAP) layer is used to reshape the tensors into $1\times1\times512$ and then feeds them into the output layer. For each input image, the resnet18 makes a prediction probability, $P = \{p_1, \ldots, p_C\}$, where C represents the number of defect classes. This probability is similar to that made by the cDCGAN for real samples.

3.3. Motivation for architecture design

The co-training algorithm, as in [17], uses two classifiers based on different supervised learning algorithms. Similarly, it is better to use different DL networks in our SSL system, which can represent data distribution in distinct views. The cDCGAN and resnet18 have very different characteristics on network structure and learning manner.

From the architecture details described in Tables 1 and 2, we can observe that these two classifiers are different in structure. The kernel size equipped in the convolutional layers of the cDCGAN is 5×5, and that of the resnet18 is relatively smaller 3×3. Given a same defect image, the two networks can produce different feature maps at each stage due to the difference in receptive field (see Fig. 5). Besides, their down-sampling manners are also different. The cDCGAN mainly uses the large kernel convolutional layer without the edge pixel padding, and the resnet18 uses max-pooling layer and stride convolutional layer with the edge pixel padding. Using the convolutional layer to reduce the size of feature map can make network learn how to down-sample but requires more computation. On the contrary, using the pooling layer is a cost-free operation but has the possibility to filter out key pixels (see Fig. 6). On the whole structure, the cDCGAN has only one path for gradient propagation. But the resnet18 has 2^n paths due to the equipment of residual block, where n represents the number of skip connections (see Fig. 7).

Besides the network structure, they are also different in learning algorithms. The cDCGAN uses the Adam algorithm to optimize the zero-sum game loss function defined in Eq. (1). The weight parameters are obtained from the adversarial learning process. But the renet18 uses the SGD algorithm to optimize the cross-entropy loss function defined in Eq. (5). The weight parameters are computed in the common representation learning process. Therefore, the multi-training gets different data representations, which can improve the generalization and robustness of the SSL defect classification system.

4. The proposed SSL defect classification method

In this section, we introduce the SSL defect classification system and the multi-training algorithm of two DL classifiers in detail. The defect classification system mainly consists of two parts: sample generation

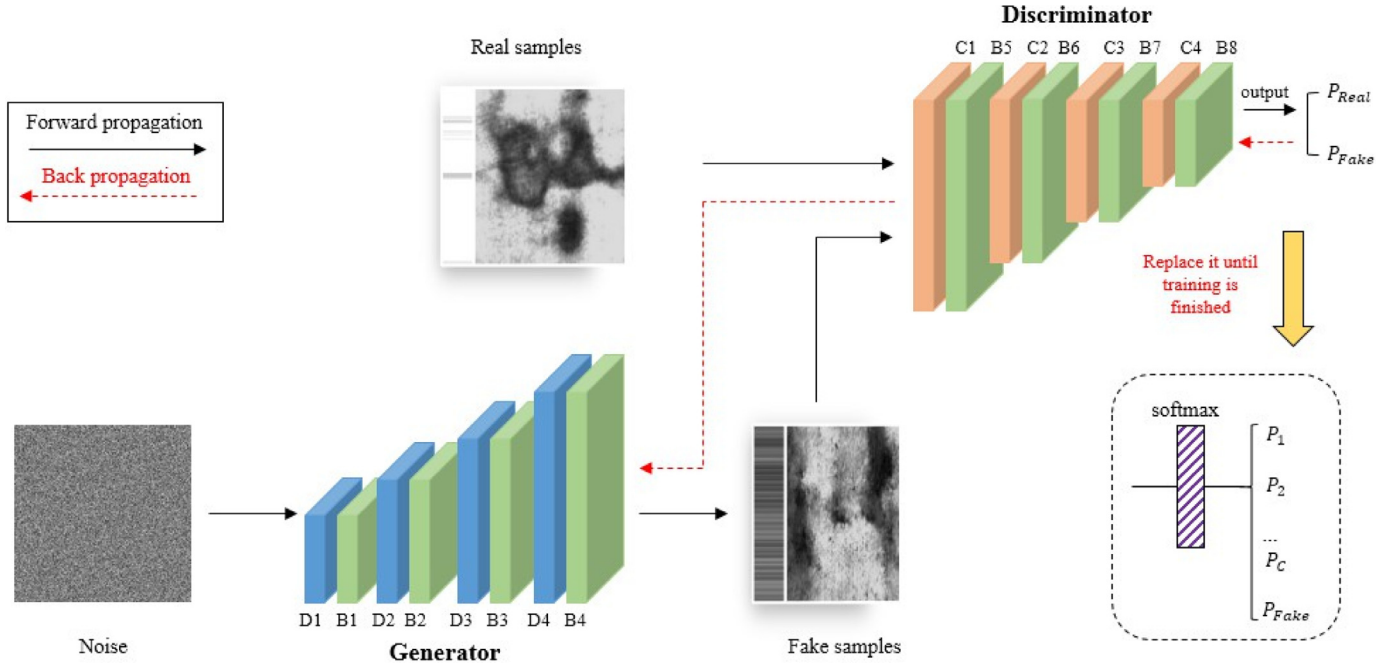

Fig. 3. The structure of the cDCGAN.

Table 1
The network architecture of the cDCGAN classifier.

Generator				Discriminator			
Name	Type	Filters	Size/stride	Name	Type	Filters	Size/stride
D1	deconv	512	5×5/2	C1	conv	64	5×5/2
B1	BN	–	–	B5	BN	–	–
D2	deconv	256	5×5/2	C2	conv	128	5×5/2
B2	BN	–	–	B6	BN	–	–
D3	deconv	128	5×5/2	C3	conv	256	5×5/2
B3	BN	–	–	B7	BN	–	–
D4	deconv	64	5×5/2	C4	conv	512	5×5/2
B4	BN	–	–	B8	BN	–	–

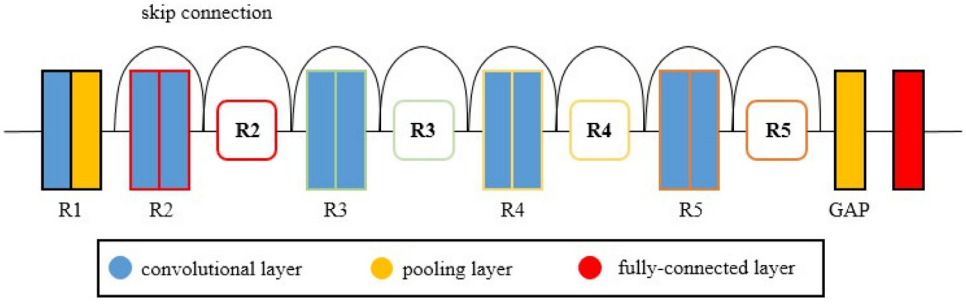

Fig. 4. The structure of the resnet18.

Table 2
The network architecture of the resnet18 classifier.

Name	Number	Type	Filters	Size/stride
R1	1	conv	64	7×7/2
		pool	–	2×2/2
R2	2	conv	64	3×3/1
		conv	64	3×3/1
R3	2	conv	128	3×3/2, 1
		conv	128	3×3/1
R4	2	conv	256	3×3/2, 1
		conv	256	3×3/1
R5	2	conv	512	3×3/2, 1
		conv	512	3×3/1
GAP	1	pool	–	7×7/1

and semi-supervised learning. The labeled samples and random noise are fed into the cDCGAN and a large number of unlabeled samples are generated after an unsupervised learning process. And then the discriminator of the cDCGAN can be used as a classifier to predict the C + 1 class probability for the unlabeled samples, where C represents the number of defect classes. The other classifier resnet18 is trained on the labeled samples and makes C predictions on the unlabeled samples as well. According to both predictions, the unlabeled samples that have the same prediction will be assigned the class label, which likes a simple voting process. Finally, the system uses the multi-training algorithm for SSL on the original samples and the selected GAN samples. The SSL system can generate large defect samples and handle both labeled and unlabeled samples simultaneously to improve the overall classification performance. The SSL defect classification pipeline is described in Fig. 8.

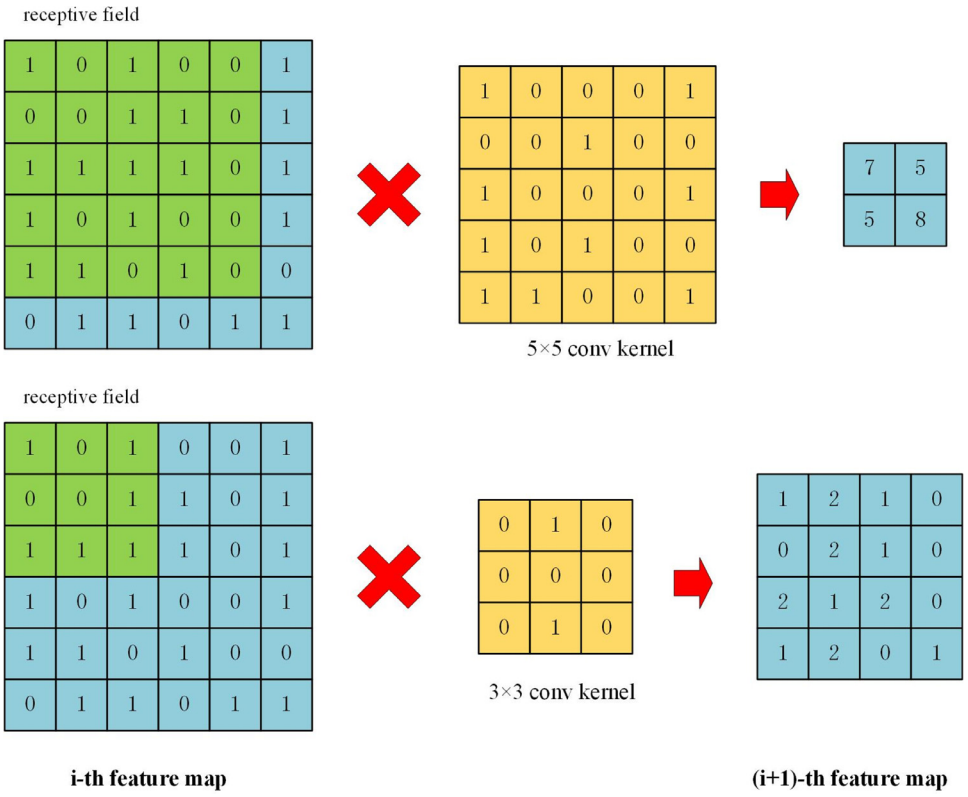

Fig. 5. The different convolutional kernels process on the feature maps.

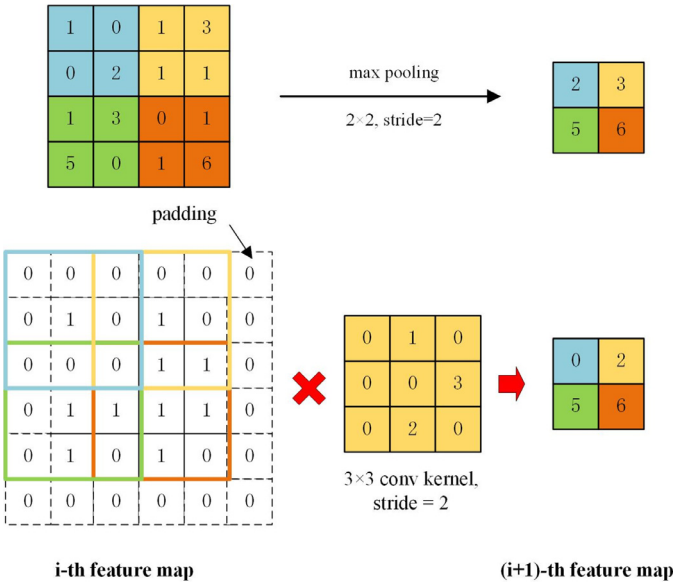

Fig. 6. The different down-sampling manners process on the feature maps.

4.1. Multi-training of two DL classifiers

After the sample generation, we present the multi-training algorithm to handle the original labeled samples and the unlabeled GAN samples for the SSL. Two different DL models, the cDCGAN and resnet18, are used in this algorithm as the classifiers. Each classifier is initially trained on the labeled samples and makes predictions on the unlabeled samples. Although there are not many labeled samples in defect dataset initially, a large number of unlabeled samples are added that regularizes the learning process, which can improve the generalization ability and reduce the risk of overfitting. After the initial training, both classifiers make predictions for each GAN sample respectively. The GAN sample that has the same predictions of two classifiers is assigned the corresponding class label and added in training set for a next training process. Based on the new training set, the classifiers are then updated and this process is repeated until adding enough GAN samples into the training.

In our SSL system, the loss function of the cDCGAN is defined in Eq. (1) and the resnet18 uses the cross-entropy loss. Let k∈{1, 2, ..., C} be the predicted class, where C is the number of defect classes. The cross-entropy loss function can be formulated as:

$$L = -\sum_{k=1}^{C} \log(p(k))q(k) \tag{2}$$

where $p(k)$ is the prediction probability which an input sample belongs to the class k. $q(k)$ is the ground-truth class distribution. Let y be the corresponding ground-truth class label, $q(k)$ can be defined as:

$$q(k) = \begin{cases} 0 & k \neq y \\ 1 & k = y \end{cases} \tag{3}$$

Therefore, if we ignore the zero term, the Eq. (4) can be equivalent to:

$$L = -\log(p(y)) \tag{4}$$

The samples generated by cDCGAN are in fact fake images, which cannot have the same weight as the real ones in learning process. Combining Eq. (2) and Eq. (4), the cross-entropy loss for resnet18 used in the SSL framework can be rewritten as:

$$L = -(1-\omega)\log(p(y)) - \alpha(n)\omega\log(p(y)) \tag{5}$$

If the input sample is real, ω=0. If the input sample is fake, ω=1. The α(n) is the penalty function for the unlabeled samples, which can be written as:

$$\alpha(n) = \begin{cases} 0 & n < N_1 \\ \frac{n-N_1}{N_2-N_1}\alpha_t & N_1 \le n < N_2 \\ \alpha_t & N_2 \le n \end{cases} \tag{6}$$

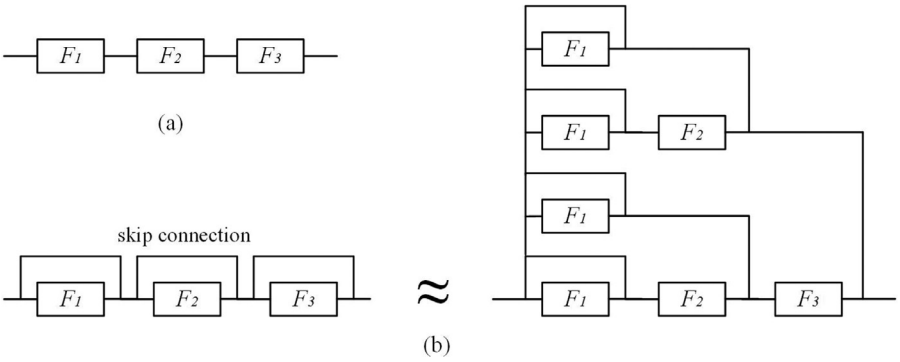

Fig. 7. The different path structure. (a) Single-path; (b) Multi-path. F_i represents the group of convolutional layers.

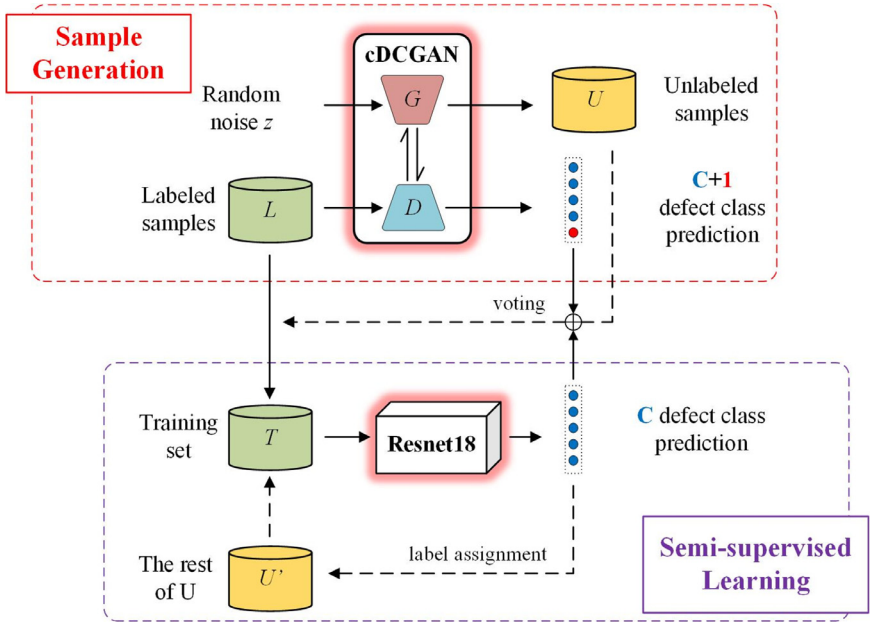

Fig. 8. The proposed SSL defect classification pipeline.

where n is the number of generated samples and α_t is the threshold (In this paper, α_t is set to 0.8). For SSL, the number of unlabeled training samples should be not less than that of the labeled ones. Let N_{real} be the number of real training samples, N_1 is equal to N_{real} and N_2 is as five times as N_{real}. In summary, the loss function of our SSL system has two terms: one for real samples and one for fake ones.

In order to include unlabeled samples into training, we use the cDCGAN and resnet18 for the multi-training. Through their predictions, a part of the GAN samples are assigned the class labels and added into training. Then the classifiers will be refined by the new training set and make predictions on the rest of the unlabeled samples. In this way, the unlabeled samples, far more than labeled ones, can be included into the SSL process.

Usually, the original labeled samples L are partitioned into two subsets: training set L_1 and test set. Then, we perform the multi-training algorithm which includes three main steps:

Step 1. Initial training. The two classifiers are trained on L_1 to generate initial models M_c and M_r, respectively. At the same time the cDCGAN can produce unlabeled samples U and make a prediction, $P_c = \{p_1, \ldots, p_C, p_{Fake}\}$, on each GAN sample. And then U are fed into M_r to obtain the predictions, $P_r = \{p_1, \ldots, p_C\}$.

Step 2. Label assignment. According to P_c and P_r, the sample in U has the same class prediction is assigned the corresponding class label and then added into L_1. Next, M_r is updated to M_r' by training on the new labeled set L_2. Based on M_r', the resnet18 makes predictions P_r' on the rest GAN samples U'. According to P_r', U' get the class labels and are added into L_2.

Step 3. Re-training. After Step 2, a new training set, defined as L^*, have already had all the required training samples which consist of the initial labeled samples L_1 and the generated unlabeled samples U. And then M_r' is retrained on L^* until max iterations reached. Finally, the model M_r^* is produced which is used to classify on the test set.

4.2. Implements

We train the models, the cDCGAN and resnet18, used in the SSL system is to minimize the loss function in Eqs. (1) and (5), respectively. All the input images are reshaped into 256 × 256 × 1 and randomly flipped before processing. The mini-batch size of each past is 128 for both models. We train the cDCGAN with the Adam [21] algorithm with exponential decay parameters β_1 and β_2 are set to 0.9 and 0.99, respectively. The cDCGAN is trained for 600 epochs with a learning rate of 0.0001. For the resnet18, we use the stochastic gradient descent (SGD) algorithm with a learning rate of 0.001. We train this model for 400 epochs in the initial step and then for 800 epochs in the retraining step.

5. Experiments

In this section, we perform extensive experiments to verify the efficiency of the proposed SSL method. We assess the availability of the

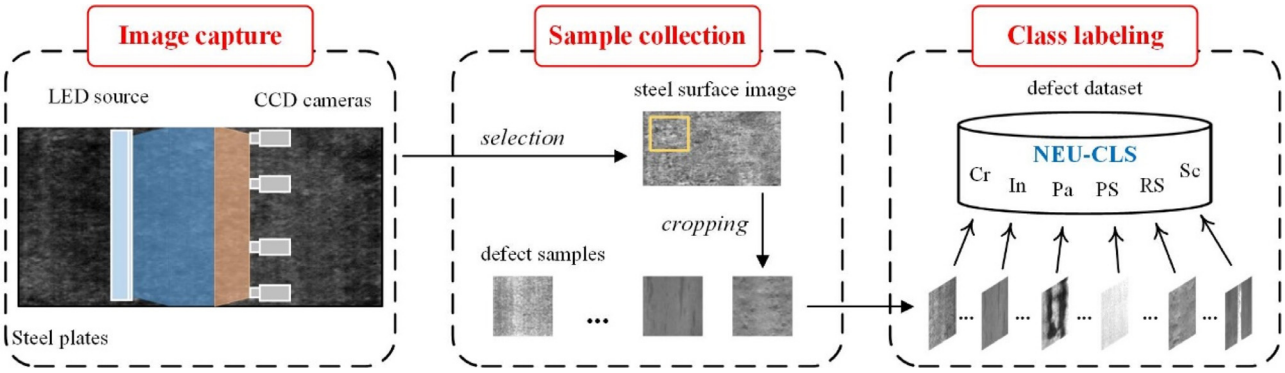

Fig. 9. The establishment of the NEU-CLS dataset.

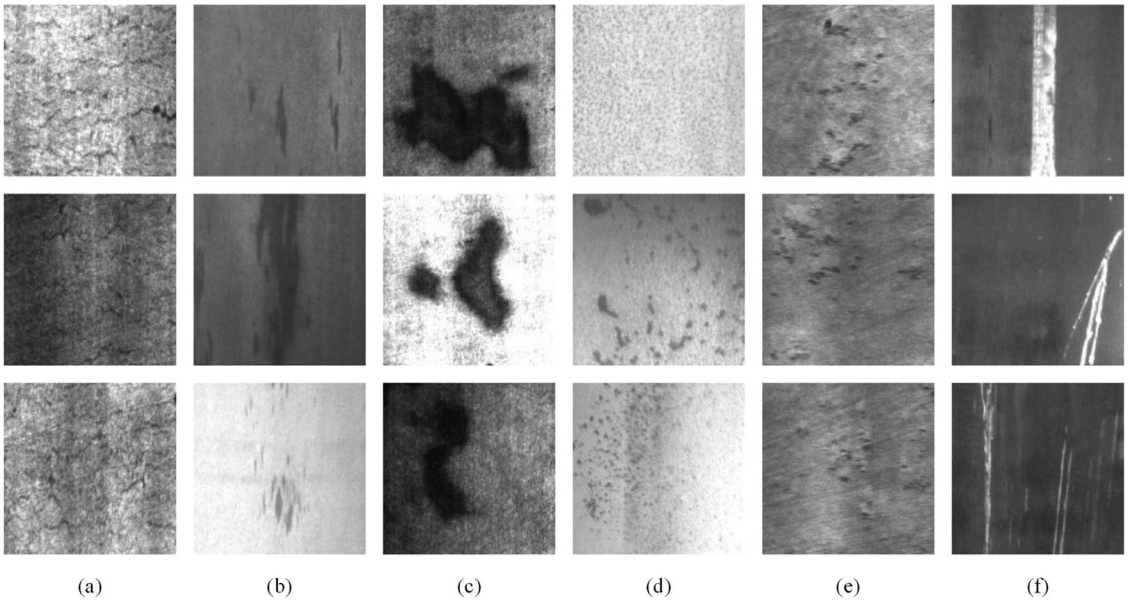

Fig. 10. Examples of defect samples in the NEU-CLS. (a) Cr. (b) In. (c) Pa. (d) PS. (e) RS. (f) Sc.

defect samples generated by the cDCGAN and then demonstrate the effectiveness and robustness of the SSL system under the circumstances of limited original samples. Our method is also compared with other types of defect classifiers and achieve the best results. All the experiments are carried out on the NEU-CLS defect dataset, in which the defect samples are collected from the actual steel surface. The proportion of training set to test set of this dataset is 6:4 and all the experiments are conducted on a single NVIDIA TITAN Xp GPU.

5.1. Defect dataset

The NEU-CLS is a defect dataset that we opened several years ago. We captured the defect images from the surface of hot-rolled steel plates by four area scan CCD cameras under the LED light. Defect samples in the same size are obtained through manual selection and cropping. Finally, the samples are assigned class labels and added into defect dataset. The process of building the NEU-CLS is shown in Fig. 9 and more details of sample collection can be found in our previous work [27]. In the NEU-CLS, there are 1800 defect samples, which belong to six kinds of typical defect of the steel surface, i.e., rolled-in scale (RS), patches (Pa), crazing (Cr), pitted surface (PS), inclusion (In) and scratches (Sc). Each class has 300 defect images that are all 200 × 200 in size. The examples are shown in Fig. 10.

According to the mentioned train/test ratio of 6:4, 60% of all the samples are selected as the training sample and the rest as the test sample. Comparing with the large-scale datasets commonly used in DL methods, such as the ImageNet which has thousands of samples per class [22], the NEU-CLS is an extremely small-scale one. In this defect dataset, we only have 180 training samples per class. Out of this reason, we attempt to generate a large number of unlabeled samples for assisting the labeled samples in learning process.

5.2. GAN results

We need to ensure most of the samples generated by the cDCGAN are available due to there is still instability in GAN architecture. In other words, the GAN samples look slightly fuzzy but can assist the real samples in training. With the real samples in NEU-CLS, we use the cDCGAN to produce a large number of fake defect samples after an unsupervised learning process. To be specific, we gain 6400 fake images based on 1080 real images in the training set of NEU-CLS, and examples of the GAN samples are shown in Fig. 11. For most types of defects, such as the In, Pa, and Sc classes, their GAN samples cannot be intuitively distinguished from the real images by human eyes. Unsurprisingly, not all classes generate well, such as the Cr class that has a little fuzzy in GAN samples (see the first column in Fig. 11).

By the cDCGAN, lots of unlabeled samples can be obtained in a simple way and the number of which is more than five times as many as the labeled ones in NEU-CLS. As mentioned above, the SSL needs the ability to handle both labeled and unlabeled samples, where the number of the

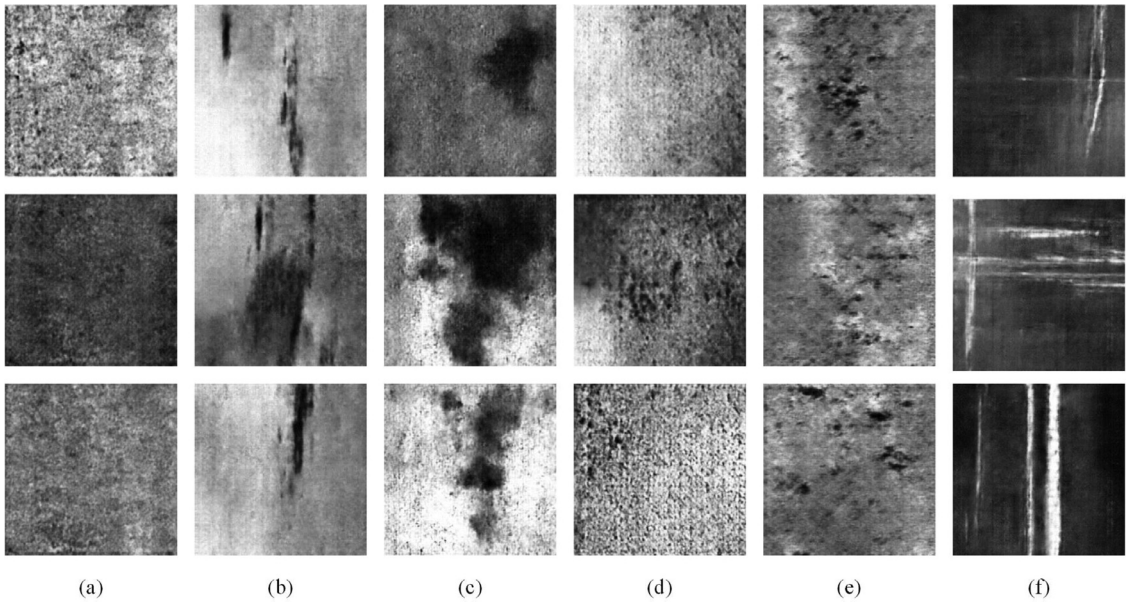

Fig. 11. Examples of the defect samples generated by the cDCGAN. (a) Cr. (b) In. (c) Pa. (d) PS. (e) RS. (f) Sc.

Table 3
Classification results under different numbers of GAN samples.

GAN samples (k-ratio)	Overall accuracy (%)	
	self-training	multi-training
0	95.60	95.60
1×	96.65	97.91
2×	97.19	98.44
3×	98.35	99.56
4×	98.19	98.35
5×	96.61	98.33

Table 4
Classification results under different numbers of original and GAN samples.

Training set original (num.)	GAN (k-ratio)	Overall accuracy (%) initial	final	improv.
54 (5%)	3×	65.62	88.12	22.50
	5×	65.62	91.96	26.34
270 (25%)	3×	86.72	89.58	2.86
	5×	86.72	95.59	8.87
540 (50%)	3×	93.75	96.06	2.49
	5×	93.75	97.92	4.17
810 (75%)	3×	94.92	98.30	3.38
	5×	94.92	97.91	2.99
1080 (100%)	3×	95.60	99.56	3.96
	5×	95.60	98.33	2.73

latter is far more than that of the former. Since the unlabeled samples are easily accessible in our system, we hope that these samples have similar semantic information with the real ones, and thereby can assist the labeled samples for boosting classification performance. So we carry out a series of experiments with the addition of different numbers of GAN samples, where the number is the multiple of that of labeled samples used in training. By this way, we can verify the effects of the unlabeled GAN samples and their number on the classification performance.

Under different numbers of GAN samples, the classification results based on two different SSL algorithms are included in Table 3. From this table, we can observe that accuracies achieved by multi-training of two classifiers outperform those achieved by self-training of single classier by approximately 1%. There is a significant improvement in classification accuracy when GAN samples are added into SSL. With the addition of GAN samples, the accuracy rises gradually and reaches the peak when 3× GAN samples are added (3× represents the number of added GAN samples is three times the original samples). When more samples are added, the accuracy drops slightly that indicates too many fake samples have begun to deteriorate the learning process. Even so, the accuracy at 5× GAN exceeds that at 0 GAN by approximately 3%. Therefore, we can safely conclude that the unlabeled samples produced by cDCGAN share the common knowledge with the real defect samples.

5.3. Results on the NEU-CLS dataset

As mentioned above, lack of defect samples often occurs in surface inspection tasks, which causes DL networks cannot be widely used in industrial fields. Whether the DL classifiers used in our SSL system can still work well when the original samples are very few. To prove it, we evaluate our method under different numbers of the original and GAN samples, and the results are summarized in Table 4. From this table, it is safe to conclude that the proposed SSL method can achieve high classification accuracy even if the original defect samples are limited. When the original samples are quite few (5% original samples), the initial accuracy is very low and then greatly improved by more than 20% with the addition of GAN samples. Although there is no longer significant improvement in accuracy as the number of original samples, our method can still achieve 2~4% improvement in accuracy. Moreover, the best results are achieved at 5× GAN when 5%, 25%, and 50% original samples are added, which means the model requires more data for an adequate training. When the 75% and 100% original samples are added, the peak accuracy is still achieved at 3× GAN.

In order to compare the classification results, we select the related methods which all use DL networks for defect classification. Specifically, methods of Jeffrey Kuo et al. [5] and Li et al. [6] use small CNNs based on fully-supervised learning. The methods as in Ren et al. [24] and Simonyan and Zisserman [25] use large networks based on transfer learning. The SSL method like [14] employs an auto-encoder for classification in a self-training manner. For SSL methods, we all use 3× GAN samples to include into learning. To fair comparison, these methods are reconstructed by the Tensorflow package. The classification results of different methods are shown in Table 5. From this table, we can find that SSL methods are more suited to data-limited defect inspection tasks,

Table 5
Comparisons with other defect classifiers.

Methods	Overall accuracy (%)		
	25% orig.	50% orig.	100% orig.
Fully-supervised learning			
Zhou et.al. [26]	78.09	80.00	86.64
Li et.al. [6]	82.81	85.39	95.00
Transfer learning			
Ren et al. [24].	–	90.88	92.04.
VGG16 [25]	–	92.22	93.18
Semi-supervised learning			
He et al. [14]	85.83	94.87	98.96
Ours	89.58	96.06	99.56

and our method achieves the best result by 99.56% in accuracy. The fully-supervised learning methods highly depend on original samples and therefore are not robust enough when the number of the original ones is too few. The transfer learning-based methods tend to be overfitting when there are only 25% of the original samples. These methods use large models pretrained on the ImageNet, which may require more defect samples for learning due to the difference between the ImageNet and defect dataset.

6. Conclusion

In this paper, we propose a SSL method using multi-training of the cDCGAN and resnet18 applied to the steel surface defect classification. Considering that defect samples are always insufficient, this method uses the cDCGAN to generate new samples instead of an extra collection. In this way, a large number of unlabeled samples can be generated based on the original labeled samples. To exploit the unlabeled samples, the multi-training algorithm of two DL classifiers is proposed in this work, which gradually refines the models and based on their predictions assigns class labels. Finally, these DL models have strong classification ability by training on enough samples. Unlike the supervised classifiers, the SSL defect method we proposed which has the ability to process the labeled and unlabeled samples. Therefore, our method is more powerful and robust than the ones only trained on labeled samples. Extensive experiments on the NEU-CLS defect dataset have shown that our method is extremely effective for defect classification even if the original samples are limited.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (51805078, 51374063), the National Key Research and Development Program of China (2017YFB0304200), the Fundamental Research Funds for the Central Universities (N170304014) and the China Scholarship Council (201806085007).

References

[1] Zhang H, Jin X, Wu J, et al. Automatic visual detection system of railway surface defects with curvature filter and improved Gaussian mixture model. IEEE Trans Instrum Meas 2018;67(7):1593–608.
[2] Chu M, Zhao J, Liu X. Multi-class classification for steel surface defects based on machine learning with quantile hyper-spheres. Chemom Intell Lab Syst 2017;168:15–27.
[3] Luo Q, Sun Y, Li P. Generalized completed local binary patterns for time-efficient steel surface defect classification. IEEE Trans Instrum Meas 2019;68(3):667–79.
[4] Yu H, Li Q, Tan Y, et al. A coarse-to-fine model for rail surface defect detection. IEEE Trans Instrum Meas 2019;68(3):656–66.
[5] Jeffrey Kuo CF, Peng KC, Wu HC, Wang CC. Automated inspection of micro-defect recognition system for color filter. Opt Lasers Eng 2015;70:6–17.
[6] Li Y, Li G, Jiang M. An end-to-end steel strip surface defects recognition system based on convolutional neural networks. Steel Res Int 2017;88(2):176–87.
[7] Huang H, Li Q, Zhang D. Deep learning based image recognition for crack and leakage defects of metro shield tunnel. Tunn Undergr Space Technol 2018;77:166–76.
[8] Bernhard S, Chapelle O, Zien A. Semi-supervised learning. Cambridge, MA: MIT Press; 2006.
[9] Lee DH. Pseudo-label: the simple and efficient semi-supervised learning method for deep neural networks. In: ICML workshop: challenges in representation learning; 2013. p. 1–6.
[10] Papandreou G, Chen LC, Murphy KP, Yuille AL. Weakly- and semi-supervised learning of a deep convolutional network for semantic image segmentation. In: Proc IEEE Int Conf Comput Vis (ICCV); 2015. p. 1742–50.
[11] Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. In: Proc Neural Inf Process Syst (NIPS); 2012. p. 1097–105.
[12] Jonathan L, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. IEEE Trans Pattern Anal Mach Intell 2017;39(4):640–51.
[13] Alec R, Metz L, Chintala S. Unsupervised representation learning with deep convolutional generative adversarial networks; 2014. arXiv:1511.06434..
[14] He D, Xu K, Zhou P, Zhou D. Surface defect classification of steels with a new semi–supervised learning method. Opt Lasers Eng 2019;117:40–8.
[15] Blum A, Mitchell T. Combining labeled and unlabeled data with co-training. In: Proceedings of the 11th annual conference on computational learning theory. ACM; 1998. p. 92–100.
[16] Goodfellow IJ, et al. Generative adversarial nets. In: Proc Neural Inf Process Syst (NIPS); 2014. p. 2672–80.
[17] Goldman SA, Zhou Y. Enhancing supervised learning with unlabeled data. In: Proceedings of the 17th International Conference on Machine Learning (ICML); 2000. p. 327–34.
[18] Zhou ZH, Li M. Tri-training: exploiting unlabeled data using three classifiers. IEEE Trans Knowl Data Eng 2005;17(11):1529–41.
[19] Masood A, Al-Jumaily A, Anam K. Self-supervised learning model for skin cancer diagnosis. In: Proceedings of 7th international IEEE/EMBS conference on Neural Engineering (NER). IEEE; 2015. p. 22–4.
[20] Liu C, Yuen PC. A boosted co-training algorithm for human action recognition. IEEE Trans Circuits Syst Video Technol 2011;21(9):1203–13.
[21] Kingma D, Ba Adam J. A method for stochastic optimization; 2014. arXiv:1412.6980v9..
[22] Deng J, Dong W, Socher R. ImageNet: a large-scale hierarchical image database. In: Proc IEEE Comput Vis Pattern Recognit (CVPR); 2009. p. 248–55.
[23] He K, Zhang X, Ren S. Deep residual learning for image recognition. In: Proc IEEE Comput Vis Pattern Recognit (CVPR); 2015. p. 770–8.
[24] Ren R, Hung T, Tan KC. A generic deep-learning-based approach for automated surface inspection. IEEE Trans Cybern 2018;48(3):929–40.
[25] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. In: Proc Int Conf Learn Represent (ICLR); 2015. p. 1–16.
[26] Zhou S, Chen Y, Zhang D. Classification of surface defects on steel sheet using convolutional neural networks. Materiali in Tehnologije 2017;51(1):123–31.
[27] Song K, Yan Y. A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects. Appl Surf Sci 2013;285(Part B):858–64.