Optics and Lasers in Engineering 117 (2019) 40–48

Contents lists available at ScienceDirect

Optics and Lasers in Engineering

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

Surface defect classification of steels with a new semi-supervised learning method

He Di, Xu Ke*, Zhou Peng, Zhou Dongdong

Collaborative Innovation Center of Steel Technology, University of Science and Technology Beijing, Xueyuan Road 30, Haidian, Beijing 100083, China

ARTICLE INFO

Keywords:
Surface inspection
Defect detection
Semi-supervised learning
Convolutional autoencoder
Generative adversarial networks

ABSTRACT

Defect inspection is extremely crucial to ensure the quality of steel surface. It affects not only the subsequent production, but also the quality of the end-products. However, due to the rare occurrence and appearance variations of defects, surface defect identification of steels has always been a challenging task. Recently, deep learning methods have shown outstanding performance in image classification, especially when there are enough training samples. Since most sample images of steel surface are unlabeled, a new semi-supervised learning method is proposed to classify surface defects of steels. The new method is named CAE-SGAN, as it is based on Convolutional Autoencoder (CAE) and semi-supervised Generative Adversarial Networks (SGAN). CAE-SGAN first trains a stacked CAE through massive unlabeled data. Considering the appearance variations of defects, the passthrough layer is used to help CAE extract fine-grained features. After CAE is trained, the encoder network of CAE is reserved as the feature extractor and fed into a softmax layer to form a new classifier. SGAN is introduced for semi-supervised learning to further improve the generalization ability of the new method. The classifier is trained with images collected from real production lines and images randomly generated by SGAN. Extensive experiments are carried out with samples captured from different steel production lines, and the results indicate that CAE-SGAN had yielded best performances compared with traditional methods. Especially for hot rolled plates, the classification rate is improved by around 16%.

1. Introduction

Surface defect is one of the most important factors affecting the quality of steel products [1]. Some defects on steel surface will not only affect the subsequent production, but also affect the corrosion and wear resistance of the end products. The inspection systems capture images of steel surface with CCD cameras under special illumination. Then, these images are processed by some defect identification algorithms to detect and classify the surface defects.

However, due to the rare occurrence and appearance variations of defects, the design of the detection and classification algorithms has always been a challenging task. Fig. 1 shows two typical defects on hot rolled plates, seams (a)–(d) and scales (e)–(h). Surface defects of steels have large intra-class variations and the image background is very complex.

Recently, various algorithms have been developed for detecting and classifying defects of steel surface. For example, a method based on extreme learning machine (ELM) and genetic algorithm (GA) was proposed to classify the defects of hot rolled plates [2]. GA was introduced to enhance the robustness of ELM in steel surface inspection; The RNAMlet was introduce into surface inspection as a feature exactor to decompose the image asymmetrically [3], which improved the adaptability of the feature extraction process in different steel production lines. Based on scale-invariant feature transform (SIFT) and support vector machine (SVM), an algorithm of surface inspection was proposed to achieved good detection result in production lines with simple and clean image background [4]. The shearlet transform (ST) was introduce in surface inspection to provide efficient multi-scale directional representation of defects [5] which improved the accuracy of defect recognition of steel surface with complex background. An approach based on discrete Fourier transform and artificial neural network was proposed to detect the surface cracks of structural steels [6]. Gabor filters were used to detect thin and corner cracks in raw steel block by minimizing the cost function of energy separation criteria of defect and defect-free regions [7]. A method for online crack detection system based on the 3D contour data of the surface of steel plate was designed [8], and it integrated image processing and statistical classification based on logistic regression in the detection system.

The above methods have achieved high detection and classification rates in some steel production lines. However, there are problems that still remain unsolved.

* Corresponding author.
E-mail address: xuke@ustb.edu.cn (X. Ke).

https://doi.org/10.1016/j.optlaseng.2019.01.011
Received 28 September 2018; Received in revised form 18 December 2018; Accepted 29 January 2019
Available online 5 February 2019

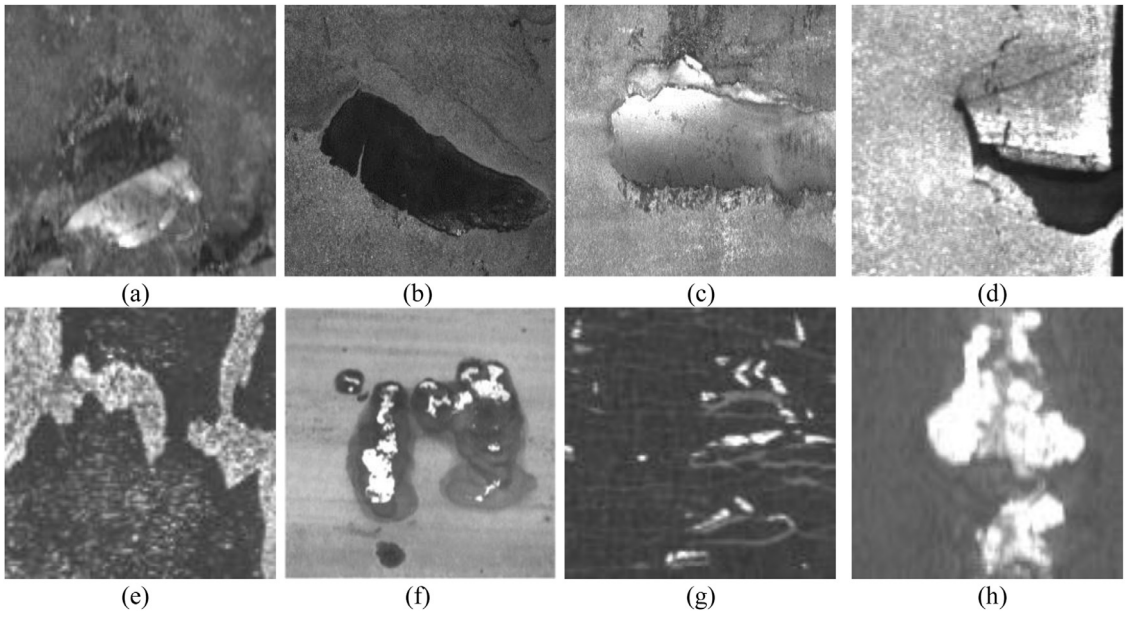

Fig. 1. Two typical defects on hot rolled plates: seams [(a)–(d)] and scales [(e)–(h)].

(1) Most of the above methods are limited to specific steel products or defects, while for defects with complex appearance, the classification rates of the above methods are relatively low and cannot meet industrial requirements. In general, the accuracy of defect classification is closely related to the result of feature extraction. Good feature descriptor can not only improve the classification rate but also improve the generalization ability of the algorithms. However, the feature descriptors used in the above methods are proposed for specific steel products or defects, so the generalization ability of these methods is limited. Especially for some defects with complex background or appearance, it is difficult to represent these defects with some simple feature descriptors, which leads to the low classification rate.
(2) The above methods cannot make full use of sample images of steel surface because all of the above methods belong to supervised learning. In general, due to the limitation of production line conditions and randomness of the defects, it is impossible to collect and label sufficient defect samples, and most of existing samples of surface defects are unlabeled. While the above methods can only learn from a small amount of labeled data, this makes the final detection and classification result unstable.

Therefore, the algorithms of surface inspection should not only be able to classify defects with complex appearance, but also be able to learn from unlabeled data. Recently, deep learning methods have shown outstanding performance in image classification and object detection. As the most typical deep learning method, convolutional neural network (CNN) was first proposed by LeCun [9] and have been proved the most effective method in image classification. Unlike other classification algorithms, there is no explicit feature extraction process in CNN. By minimizing the loss function, the weights of the convolutional layers are automatically learned. This improves the classification performance of images with complex background and appearance variation. Besides, lots of variants of CNN were also proposed, such as Alexnet [10], VGG [11], NIN [12], Inceptions [13], Inception-Resnet [14] and Densenet [15]. All of these works tend to use deeper stacked convolutional layers or asymmetrical structure to extract more non-linear features, thus further improving the performance of CNN in complex images. Some novel works also showed that the CNN can be applied to more difficult tasks, such as object detection [16–22]. However, the performance of CNN-based methods largely depends on sufficient training samples. Training a CNN with small datasets greatly affects the generalization ability of the algorithm, which limits the application of CNN in industrial scenes. For now, the most effective way of applying CNN to small datasets is transfer-learning [23] which is developed on the assumption that the sample images used in our own field share universal features like curves and edges with the images used to train the pre-trained model. Based on transfer learning, CNN can be applied to the training of some small datasets, such as emotion recognition [23] and automatic medical diagnosis [24–26]. However, application of transfer learning in steel surface inspection is not as good as other fields. An important reason is that the image context of the steel surface is quite different from most pre-trained model, which violates the application conditions of transfer learning.

Therefore, to solve the problems above, a new semi-supervised learning method is proposed in this paper to classify steel surface defects, based on Convolutional Autoencoder (CAE) [27] and Semi-supervised Generative Adversarial Networks (SGAN) [28]. As an unsupervised learning method, CAE is widely used to extract features from unlabeled data. Compare with transfer learning, CAE can better preserve the essential aspects of steel surface defects in more robust and discriminative representations. Considering the complex appearance variation of surface defects, the passthrough layer [21] is used to help CAE extract fine-grained features. Besides, SGAN is also introduced in our method for semi-supervised learning, which further improves the performance of defect classification with limited training samples.

The rest of the paper is organized as follows: Section 2 illustrates the architectures of our methods; Section 3 and Section 4 show the image acquisition devices used in this paper and the experimental results of our method achieved from the recognition of the defects of different steel production lines; followed by conclusions drawn in Section 5.

2. Architectures

2.1. Standard autoencoder

An autoencoder is an artificial neural network used for unsupervised learning of feature encoding [27]. Since the number of labeled samples is limited and the image context of steel surface is different from the public dataset, the autoencoder is more suitable for extracting the features of steel surface defects, comparing with other deep learning methods [9,23]. In general, an autoencoder consists of two parts, the encoder

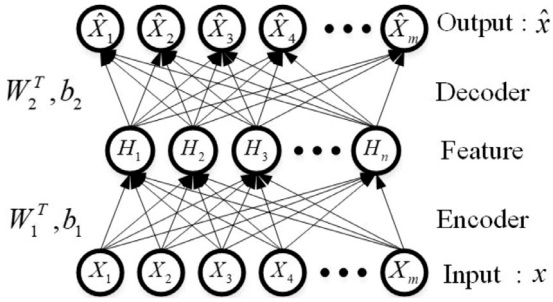

Fig. 2. The basic structure of Autoencoder.

network and the decoder network. The input is first transformed into a typically lower-dimensional space in encoder network, and then expanded to reproduce the initial data in decoder network. The general structure of autoencoder is shown in Fig. 2.

The input data x represents a m-dimension vector $x \in R^m$. The output data *feature* represents n-dimension vector $feature \in R^n$. And the $m > n$. Standard autoencoder includes three main steps.

Step 1: convert the input x into code of encoder network.

$$f = sigmoid(W_1^T x + b_1) \tag{1}$$

where x is the input vector; W_1^T is the weight matrix between the input and hidden layers. b_1 is the bias vector. f is output value of the encoder network.

Step 2: Based on the output of the encoder network, the decoder network reconstructs the input value.

$$\hat{x} = sigmoid(W_2^T f + b_2) \tag{2}$$

where f is the output of encoder network; W_2^T is the weight matrix between the hidden and output layer; b_2 is the bias vector. $\hat{x}$ is output value of the encoder network.

Step3: Minimize the error of the original image and the output of decoder network.

$$\min_{En,De} Loss(En, De) = \|x - \hat{x}\|^2 \tag{3}$$

where *En* represents the encoder network; *De* represents the decoder network;x is the input vector; $\hat{x}$ is the output of the decoder network and $\hat{x} = De(En(x))$. In general, a penalty term α is added to force the network encode more sparse representation of the input image.

$$\alpha = \sum_{i}^{n} KL(\rho||\hat{\rho}_i) \tag{4}$$

where $\hat{\rho}_i$ is the average output of the encoder network; ρ is the target value which is set 0.05 in experiments; *KL* is the Kullback–Leibler divergence which helps the encoder network output more sparse value.

2.2. Convolutional autoencoder with passthrough layer

The standard autoencoder can only be applied to one-dimensional data, so it is not suitable for processing image. One important reason is that there are lots of redundancy in the parameters [27,29]. Considering the great success of CNN in image classification [10–15], we used CAE to extract the features of steel surface defects. Different from the standard autoencoder, CAE uses convolutional layers to encode features. The weights and bias are shared among all locations in the input, which means it can preserve spatial locality. The reconstruction is hence a linear combination of basic image patches based on the output of the encoder.

Fig. 3 shows the architecture of a CAE used in the defect classification. The specific parameters are shown in Table 1 and Table 2. The encoder network takes an image of size 224 × 224 as input, and processed by eight convolutional layers and four max-pooling layers, which are

Table 1
The parameters of the encoder network of CAE used in the steel surface inspection.

Encoder network		
Type	Filters	Size/stride
Convolutional	32	3 × 3/1
Max-pooling		2 × 2/2
Convolutional	64	3 × 3/1
Convolutional	64	3 × 3/1
Max-pooling		2 × 2/2
Convolutional	128	3 × 3/1
Convolutional	128	3 × 3/1
Max-pooling		2 × 2/2
Convolutional	256	3 × 3/1
Convolutional	256	3 × 3/1
Convolutional	128	1 × 1/1
Max-pooling		2 × 2/2

Table 2
The parameters of the decoder network of CAE used in steel surface inspection.

Decoder network		
Type	Filters	Size/stride
Convolutional	128	3 × 3/1
Convolutional	128	3 × 3/1
Upsampling		2 × 2/2
Convolutional	128	3 × 3/1
Convolutional	64	1 × 1/1
Upsampling		2 × 2/2
Convolutional	64	3 × 3/1
Convolutional	64	3 × 3/1
Upsampling		2 × 2/2
Convolutional	32	3 × 3/1
Convolutional	32	3 × 3/1
Upsampling		2 × 2/2
Convolutional	1	1 × 1/1

represented by the blue and green boxes respectively in Fig. 3, to construct a set of discriminative feature maps. The convolutional layers can be described as:

$$y = f\left(\sum_c f_c * k_c + b_c\right) \tag{5}$$

where y denotes the output feature maps of a convolutional layer; f_c and k_c are c th feature maps and convolutional kernel respectively; b_c is the c th bias; $*$ is convolution operation; Both k_c and b_c are continuously learned in the training process. To improve the non-linear expression ability of the network, the rectified linear unit (ReLU) is used at each convolutional layer. It is expressed as:

$$y = \max(x, 0) \tag{6}$$

where x is the output of the convolutional layers; The max-pooling is used to reduce the dimension of the feature maps and is described as:

$$y_{(i,j)} = \max_{(i,j)\in R} \left(x_{(i,j)}\right) \tag{7}$$

where the $x_{(i,j)}$ denotes the (i, j) th pooling region; R is all the different pooling regions; and $y_{(i,j)}$ is the output of (i, j) th pooling regions; The max-pooling increases the translation invariability of CAE and forces it to encode more sparse features.

Since the size of the input image is 224 × 224 and there are four max-pooling layers in encoder network, the size of the output feature maps of encoder network is 14 × 14, which is then feed into the decoder network to reconstruct the original image.

Besides, considering the complex variation in appearance of defects, we also introduce the passthrough layer [21] in the encoder network to extract fine-grained features. With the number of convolutional and max-pooling layer increases, detailed features will be lost, which is bad

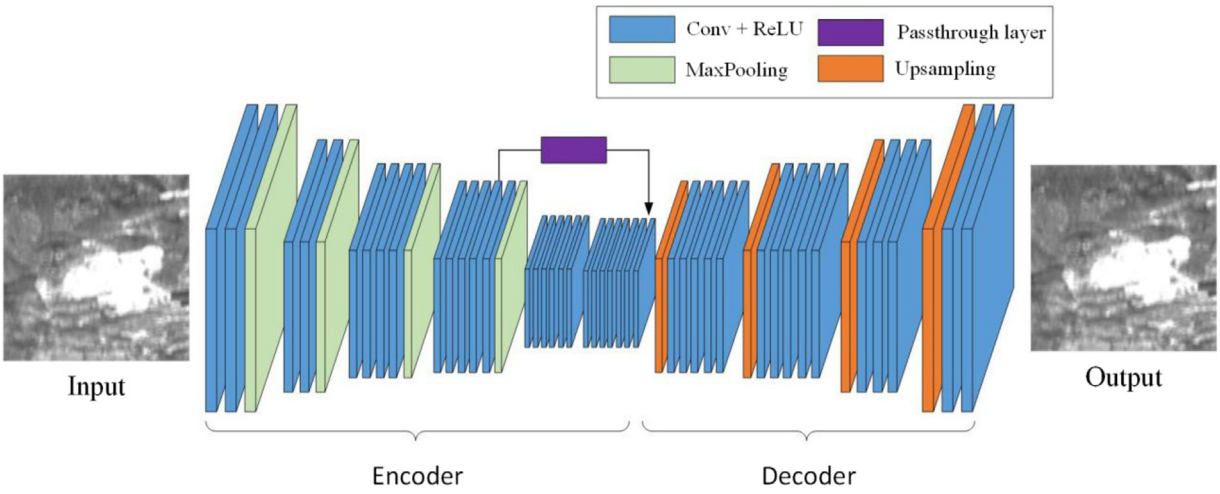

Fig. 3. The structure of a Convolutional Autoencoder. The blue box represents feature maps of convolutional layers; The green box represents max-pooling layers; The purple box represents passthrough layer; The orange box represents upsampling layers. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

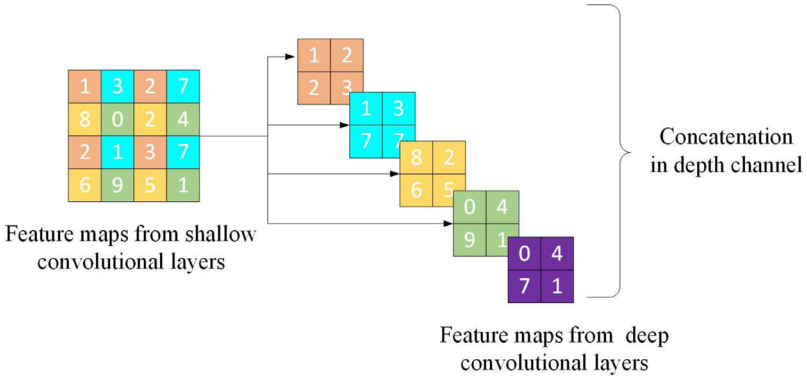

Fig. 4. The process of passthrough layer. The feature map with size of 4×4, output by shallow convolutional layer, is converted into four feature maps with size of 2×2, and concatenated with the feature maps output by deep convolutional layers.

for image reconstruction and classification. While the passthrough layer can solve this problem because it introduces the higher resolution features into the output of the encoder network. Fig. 4 shows a simple example of passthrough layer. The passthrough layer converts a feature map with the size of 4×4, which is output by a shallow convolutional layer, to 4 different feature maps with the size of 2×2 by stacking the adjacent features into different feature map channels. And then, the passthrough layer concatenates these feature maps output by shallow convolutional layer with the feature maps output by deep convolutional layers, represented by purple rectangle in Fig. 4, in depth channels to form the final output of the encoder network.

More specifically, in the encoder network, the passthrough layer is used at the fifth convolutional layer, in which there are 128 different feature maps with size of 28×28. The passthrough layer convert these 128 feature maps to 512 by the method discussed above. And then, the passthrough layer concatenates these 512 feature maps with the 128 feature maps output by the last convolutional layers. Therefore, the final output of the encoder network is $14 \times 14 \times 640$, where the 640 represents 640 different feature maps.

As shown in Fig. 3 and Table 2, the decoder network takes the output of encoder network as inputs, and convolves with nine convolutional layer and up-sampling layers. The up-sampling works opposites to max-pooling to expand the dimensions of the feature maps. Fig. 5 shows how up-sampling works in our method. We follow the work in [30] and up-sample the feature maps by replication.

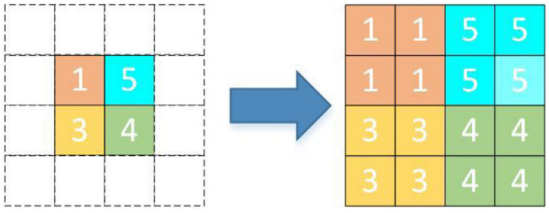

Fig. 5. The process of up-sampling.

The final output of the decoder network is the same size as the input image of the encoder network, which is later used to calculate the reconstruction error.

2.3. Semi-supervised generative adversarial networks

A standard GAN consists two deep neural networks, the generator and the discriminator [28]. The generator is used to generates new data instances, while the other, the discriminator, evaluates them for authenticity. The structure of GAN is shown in Fig. 6.

The discriminator is a CNN that can categorize the images fed to it, a binominal classifier labeling images as real or fake, and the objective of the discriminator is to maximize the chance to recognize real images as real and generated images as fake. The loss function of the discriminator is defined as:

$$\max_{D} Loss(D) = \mathrm{E}_{x \sim p_{data}(x)}\left[\log D(x)\right] + \mathrm{E}_{z \sim p_z(z)}\left[\log\left(1 - D(G(z))\right)\right] \quad (8)$$

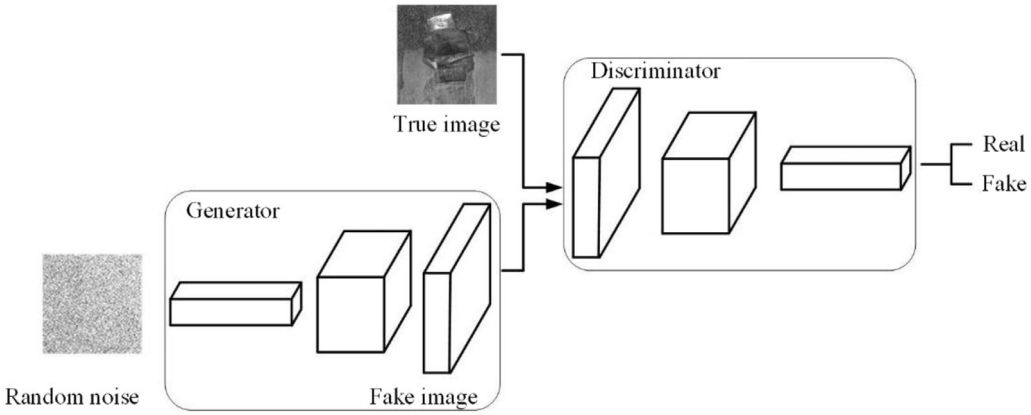

Fig. 6. The structure of GAN.

where the D represents the discriminators and the G represents the generator which takes a random noise z as input. The x represents the real image of the datasets. The objective of the generator is to generate the image with the highest possible value of $D(x)$ to fool the discriminator. The loss function of the discriminator is defined as:

$$\min_G Loss(G) = \mathrm{E}_{z\sim p_z(z)}\left[\log\left(1 - D(G(z))\right)\right] \tag{9}$$

The two networks are trained jointly by the alternating gradient descent.

In CAE-SGAN, we follow the work in [27] and extend GAN for semi-supervised learning to improve the classification performance. In experiments, the encoder network of CAE is reserved and fed into a softmax layer to form the discriminator. Rather than predicting true or fake of the input image, the discriminator is made to predict $N + 1$ classes, where N is the number of the different classes, and the extra class represents if the input image is from the real dataset or the generator. Through this improvement, our method shows better generalization ability especially for hot rolled plates.

Another improvement is that we do not truncate the decoder network as most other methods [26] do when training the classifier (the discriminator). On the contrary, the decoder network still reconstructs the sample images from the real production lines, and propagate the errors back to the encoder network. Therefore, the loss function of training the discriminator is defined as:

$$\begin{aligned}\min_G \max_D Loss(G, D) = {} & \mathrm{E}_{x\sim p_{data}(x)}\left[\log D(y|x, y <= N)\right] \\ & + \mathrm{E}_{x\sim p_{data}(x)}\left[\log\left(1 - D(y = N + 1|x)\right)\right] \\ & + \mathrm{E}_{z\sim p_z(z)}\left[\log\left(D(y = N + 1|G(z))\right)\right] \\ \min_{En,De} Loss(En, De) = {} & \alpha\|x - \hat{x}\|^2_{x\sim p_{data}(x)}\end{aligned} \tag{10}$$

where De represents the decoder network; En represents the encoder network which is also used as the convolutional layers of the discriminator; α is used to down-weighting of the loss from the image reconstruction to ensure the discriminator converge quickly.

The specific structure of the generator of SGAN is shown in Table 3. The input size of the generator is 14×14 which are sampled from a uniform distribution from 0 to 1. The generator of the SGAN consists of nine convolutional layers and four up-sampling layers, as shown in Table 3. The up-sampling layer works the same as it in the decoder network of CAE to expand the feature maps size. After proceed by these up-sampling and convolutions, the noise input of the generator is mapped into a fake image which is used to confuse the discriminator of SGAN. The structure of discriminator of SGAN is basically the same as the encoder network of CAE, and the only difference is that the output of encoder network is fed into a softmax layer to predict $N + 1$ classes.

Table 3
The parameters of the generator of SGAN used in steel surface inspection.

Generator		
Type	Filters	Size/stride
Convolutional	128	3×3/1
Convolutional	128	3×3/1
Upsampling		2×2/2
Convolutional	128	3×3/1
Convolutional	64	1×1/1
Upsampling		2×2/2
Convolutional	128	3×3/1
Convolutional	64	1×1/1
Upsampling		2×2/2
Convolutional	32	3×3/1
Convolutional	32	3×3/1
Upsampling		2×2/2
Convolutional	1	1×1/1

2.4. Trainings

So far, we have described the algorithms that are used in our new method. Next, we will describe how to train the two networks into a unified classification network.

In the experiments, the training process is divided into two stages. In the first stage, CAE is trained with unlabeled data collected from different steel production lines to extract features of steel surface defects. To improve the generalization ability of feature extraction, the training samples used in first stage include not only the common defects, but also some background images of the steel surface. In the second stage, the encoder network of CAE is reserved as feature extractor and is fed into a soft-max layer to form the discriminator. Then, GAN is introduced to generate fake images of steel surface defects to train the discriminator. The training samples used at this stage will only include the sample images of the specific production line that need to be classified, and for image reconstruction, the decoder network only reconstructs the images of the real defects.

We also introduce various data augmentation methods, including scaling and translation of up to 10% of the original image size, gaussian blur, mean filtering and random dropout. The learning rate is set to 0.001 and 0.0001 for the first and second stage, respectively. The momentum is 0.9 and the weight decay is 0.0005. Our implementation uses Keras and Tensorflow.

3. Image acquisition devices

Fig. 7 shows the image acquisition devices of surface inspection system that are used in this paper to collected sample images from different

Fig. 7. The image acquisition device: (a) LED lighting for cold rolled steel strips, (b) green laser for hot rolled steel plates, (c) green laser for continuous casting slabs. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

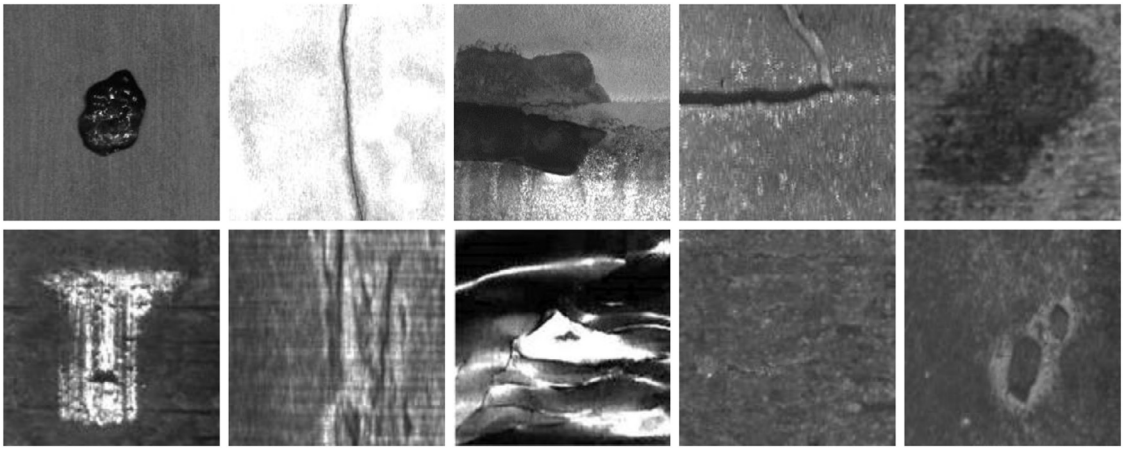

Fig. 8. Samples of the unlabeled dataset collected from different steel production lines.

steel production lines, which are designed by the authors [2,3,5]. The image acquisition device includes cameras and lighting sources. Multiple line-scan CCD cameras are used to capture images of steel surface under different ways of illumination. For the cold and hot rolled strips, the white LED lamps are used as lighting sources. While for the hot rolled plates and continuous casting slabs, the green lasers are used to reduce the influence of high-temperature radiation of the steel surface on CCD camera imaging (The wavelength of laser is 532 nm, which is far away from spectrum of high-temperature radiation of hot rolled plates and slabs). Furthermore, a color filter of narrow band is added to the camera lens, and the central bandwidth of the filter is 532 nm. With the laser lighting and the narrow band filter, most of the high-temperature radiation is filtered out, and only lights of lasers reflected by the steel surface enter into the CCD cameras.

4. Experiments and results

In this section, we show the experimental results of our method applied to the surface defect classification of steels. Since steel products from different types of production lines have different surface conditions and different types of defects. Therefore, the methods were tested with the samples collected from three most typical production lines, including hot rolled plates, hot rolled strips, and cold rolled strips. As the images are collected by the devices with the devices introduced in Section 3. To compare the classification results, other algorithms commonly used in surface inspection of steels were also tested, including ST [5], G-ELM [2], RNAMlet [3] and transfer-learning based on VGG16 [13].

4.1. Unlabeled data

Fig. 8 shows some samples of the dataset. There are around 21,000 images in the dataset, which is used to train CAE of the new method.

Table 4
The specific number of each type of defects in hot rolled plates.

Defects	Sc	Wd	Ws	Se	Vc	Hc	Rm	Total
Training	850	850	650	650	600	700	700	5000
Testing	200	200	200	200	200	200	200	1400
Total	1050	1050	850	850	800	900	900	6400

4.2. Hot rolled plates

Of all the steel production lines, the surface defects of hot rolled plates are the most difficult to classify. An important reason is that the surface of hot rolled plates is covered with a large number of scales, which can be easily misclassified as other defects. Fig. 9 shows seven types of defects of hot rolled plates, including scale(Sc), water drop(Wd), water strain(Ws), seam(Se), vertical crack(Vc), horizontal crack(Hc) and rolling mark(Rm). Horizontal cracks, vertical cracks, seams and rolling marks are true defects, which affect the quality of steels seriously, while the other three types are pseudo defects, which have no effect on the quality of steels. In Fig. 9, scales in Fig. 9(a) and (b) are very similar to seams in Fig. 9(f) and (g). As the number of scales on hot rolled steel plates is usually much more than seams, a lot of scales are incorrectly classified as seams with the traditional methods. The same problem also exists in vertical cracks in Fig. 9(h) and water stains in Fig. 9(d) and (e).

In the experiment, we used 6400 images of hot rolled plates, among which 5000 sample images are selected randomly as training sets, the remaining 1400 sample images are used as test sets. The specific number of each type of defects is shown in Table 4. The samples of scales and water drops are reduced to make the number of each type of defects more balanced.

Table 5 shows the classification result of hot rolled plates obtained by different algorithms. For ST, G-ELM and RNAMlet, the classification

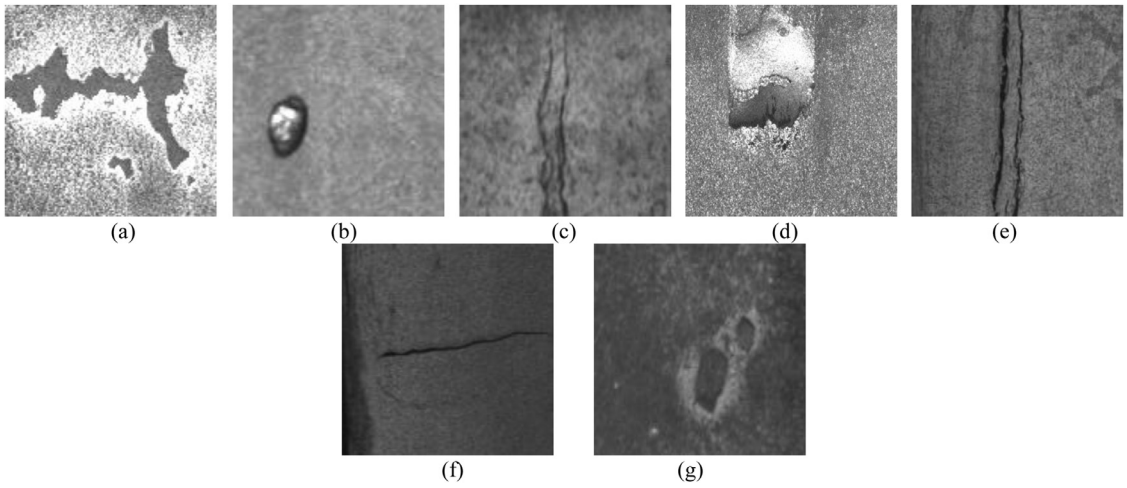

Fig. 9. Seven typical defects on the hot rolled steel plates: (a) scale(Sc); (b) water drop(Wd); (c) water stain(Ws); (d) seam(Se); (e) vertical crack(Vc); (f) horizontal crack(Hc); and (g) rolling mark(Rm).

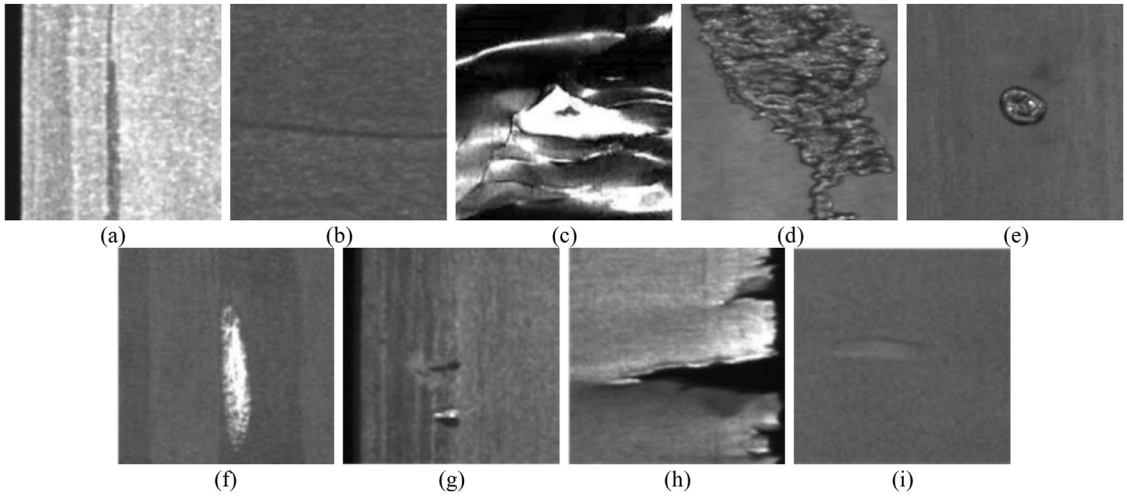

Fig. 10. Nine typical defects of hot rolled strips: (a) longitudinal crack(Lc); (b) transverse crack(Tc); (c) wrinkle(Wr); (d) scar(Sr); (e) water mark(Wm); (f) scale(Sc); (g)seam(Sm); (h) edge crack(Ec); and (i) rolling mark(Rm).

Table 5
The experiment result of hot rolled plates.

Algorithms	Training	Test
ST	82.3%	79.5%
G-ELM	85.6%	80.2%
RNAMlet	87.6%	82.5%
VGG16	94.2%	93.8%
CAE-SGAN	98.3%	97.2%

rate in the test set decreased significantly. An important reason is that the feature extraction methods used by these algorithms are based on prior knowledge, which can easily cause over-fitting when the training samples are insufficient or the image backgrounds are very complex. For the transfer-learning, VGG16 achieved a classification rate of 94.2% in the training set and 92.8% in the test set, which is much better than traditional methods. While for CAE-SGAN, the classification rate of the test set is around 97%, which is the best of all algorithms, 5% higher than transfer learning and 16% higher than other methods.

4.3. Hot rolled steel strips

Fig. 10 shows nine image samples of hot rolled strips, including longitudinal crack(Lc), transverse crack(Tc), wrinkle(Wr), scar(Sr), water mark(Wm), scale(Sc), seam(Sm), edge crack(Ec) and rolling mark(Rm). All samples except scales and water marks are true defects. The surface of hot rolled strips is relatively clean, and the classification of defects is easier than that of hot rolled plates.

In the experiment, we used 10,800 images of hot rolled steel strips, among which 9000 sample images are selected randomly as training sets, the remaining 1800 sample images are used as testing sets. The specific number of each type of defects is shown in Table 6.

Table 7 shows the experiment result of different algorithms used in the defect classification for the hot rolled steel strips. Because the surface of hot rolled steel strip is relative clean, the accuracy of these methods in hot rolled strips is higher than that in hot rolled plates. Among all the methods, CAE-SGAN achieved the best performance in classifying the training and test set. The classification rate of our method is above 98%.

4.4. Cold rolled strips

The surface quality of cold rolled strip is the best among all the steel production lines. However, the number of defect samples of cold rolled strips are much less than that of other production lines, which limits the application of deep learning methods to defect classification of cold rolled strip. Fig. 11 shows eight image samples of cold rolled strips,

Table 6
The specific number of each type of defects in hot rolled plates.

Defects	Lc	Tc	Wr	Sr	Wm	Sc	Sm	Ec	Rm	Total
Training	1000	1000	900	900	900	1200	1000	1100	1000	9000
Testing	200	200	200	200	200	200	200	200	200	1800
Total	1200	1200	1100	1100	1100	1400	1200	1300	1200	10,800

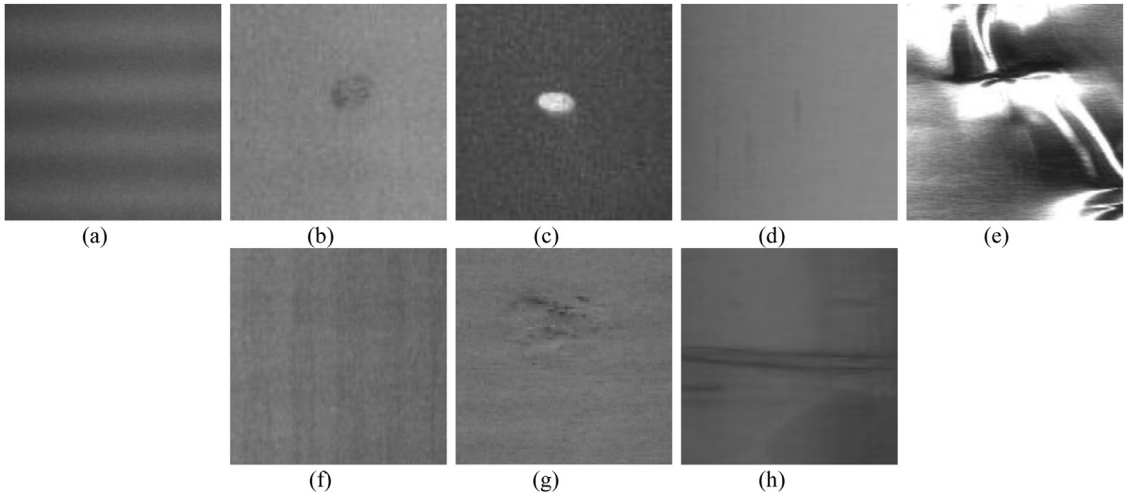

Fig. 11. Eight typical defects of hot rolled strips: (a) ripple(Rp); (b) stain(St); (c) corrosion(Cr); (d) longitudinal scratch(Ls); (e) wrinkle(Wr); (f) scale(Sc); (g) pit(Pt); and (h) transverse crack(Tc).

Table 7
The experiment result of hot rolled steel strips.

Algorithms	Training	Test
ST	86.2%	85.6%
G-ELM	87.2%	84.3%
RNAMlet	86.4%	85.6%
VGG16	95.3%	93%
CAE-SGAN	98.6%	98.2%

Table 8
The specific number of each type of defects in hot rolled plates.

Defects	Rp	St	Cr	Ls	Wr	Sc	Pt	Tc	Total
Training	400	400	400	400	400	400	400	400	3200
Testing	200	200	200	200	200	200	200	200	1600
Total	600	600	600	600	600	600	600	600	600

Table 9
The experiment result of cold rolled strips.

Algorithms	Training	Test
ST	93.2%	93%
G-ELM	91%	91.2%
RNAMlet	95%	94.5%
VGG16	94.6%	94.2%
CAE-SGAN	98%	96.7%

including ripple(Rp), stain(St), corrosion(Cr), longitudinal scratch(Ls), wrinkle(Wr), scale(Sc), pit(Pt), and transverse crack(Tc). All samples except stains are true defects.

In the experiment, we used 5600 images of hot rolled steel strips used, among which 4000 sample images are selected randomly as training sets, the remaining 1800 sample images are used as testing sets. The specific number of each type of defects is shown in Table 8.

Table 9 shows the experiment result of different algorithms tested in the defect classification for the cold rolled strips. Since the number of defects in cold rolled strips is much less than that in hot rolled steels, the classification rate of VGG16 is at the same level as the traditional methods. For CAE-SGAN, the classification rate is 96.5% in test set which is around 2% higher than RANMlet.

5. Conclusions

(1) In order to improve the classification accuracy of steel surface defects, a new semi-supervised learning method named as CAE-SGAN is proposed in this paper. The new method first trains a stacked CAE through massive unlabeled data to extract features of the defects. Passthrough layer is introduced into CAE to extract fine-grained features. After CAE is trained, the encoder network of CAE is reserved as the feature extractor and fed into a soft-max classification layer to form a new classification network. Then, SGAN is introduced for semi-supervised learning with the new classification network to classify the steel surface defects.
(2) We have also improved the training process. When training the classifier (discriminator), the decoder network of CAE is not truncated. CAE still reconstruct the images from real product lines. That is, the encoder network takes gradients from both the image classification and image reconstruction. The reconstruction process works as a regularization item which further improves the generalization ability of CAE-SGAN.
(3) Compared with the tradition methods, CAE-SGAN can make full use of sample images of steel surface (labeled and unlabeled images), which improves the accuracy of defect classification with limited training samples.

Acknowledgments

This work is sponsored by the National Natural Science Foundation of China (No. 51674031) and National Key R&D Program of China (no. 2018YFB0704304).

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.optlaseng.2019.01.011.

References

[1] Ravikumar S, Ramachandran KI, Sugumaran V. Machine learning approach for automated visual inspection of machine components. Expert Syst Appl 2011;38(4):3260–3266.
[2] Tian S, Xu K. An algorithm for surface defect identification of steel plates based on genetic algorithm and extreme learning machine. Metals 2017;7(8):311.
[3] Xu K, Xu Y, Zhou P, Wang L. Application of RNAMlet to surface defect identification of steels. Opt Lasers Eng 2018;105:110–17.
[4] Suvdaa B, Ahn J, Ko J. Steel surface defects detection and classification using SIFT and voting strategy. Int J Softw Eng Appl 2012;6(2):161–5.
[5] Xu K, Liu S, Ai Y. Application of shearlet transform to classification of surface defects for metals. Image Vis Comput 2015;35:23–30.
[6] Paulraj MP, Shukry AM, Yaacob S, Adom AH, Krishnan RP. Structural steel plate damage detection using DFT spectral energy and artificial neural network. In: Signal Processing and Its Applications (CSPA), 2010 6th International Colloquium on. IEEE; 2010, May. p. 1–6.
[7] Yun JP, Choi S, Kim JW, Kim SW. Automatic detection of cracks in raw steel block using Gabor filter optimized by univariate dynamic encoding algorithm for searches (uDEAS). NDT & E Int 2009;42(5):389–97.
[8] Landstrom A, Thurley MJ. Morphology-based crack detection for steel slabs. IEEE J Sel Top Signal Process 2012;6(7):866–75.
[9] LeCun Y, et al. Gradient-based learning applied to document recognition. In: Proceedings of the IEEE 86.11; 1998. p. 2278–324.
[10] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems; 2012. p. 1097–105.
[11] Simonyan K, Zisserman A, 2014. Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556.
[12] Lin M, Chen Q, Yan S, 2013. Network in network. arXiv:1312.4400.
[13] Szegedy C, et al. Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2015.
[14] Szegedy C, et al. Inception-v4, Inception-ResNet and the impact of residual connections on learning AAAI; 2017.
[15] Huang G, Liu Z, Van Der Maaten L, Weinberger KQ, et al. Densely connected convolutional networks. In: Girshick R, et al., editors. CVPR. Rich feature hierarchies for accurate object detection and semantic segmentation, 1; 2017. p. 3. Proceedings of the IEEE conference on computer vision and pattern recognition. 2014, July.
[16] He K, et al. Spatial pyramid pooling in deep convolutional networks for visual recognition European conference on computer vision Cham. Springer; 2014.
[17] Girshick, R. "Fast r-cnn." arXiv:1504.08083 (2015).
[18] Ren S, et al. Faster r-cnn: towards real-time object detection with region proposal networks. Adv Neural Inf Process Syst 2015.
[19] Redmon J, et al. You only look once: unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016.
[20] Redmon, J., and A. Farhadi. "YOLO9000: better, faster, stronger." arXiv preprint (2017).
[21] Liu W, et al. Ssd: single shot multibox detector European conference on computer vision Cham. Springer; 2016.
[22] Ng H-W, et al. Deep learning for emotion recognition on small datasets using transfer learning. In: Proceedings of the 2015 ACM on international conference on multimodal interaction. ACM; 2015.
[23] Huynh BQ, Li H, Giger ML. Digital mammographic tumor classification using transfer learning from deep convolutional neural networks. J Med Imaging 3.3 2016:034501.
[24] Hoo-Chang S, et al. Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. IEEE Trans Med Imaging 2016;35(5):1285.
[25] Li Q, et al. Medical image classification with convolutional neural network Control Automation Robotics & Vision (ICARCV), 2014 13th International Conference on. IEEE; 2014.
[26] Masci J, et al. Stacked convolutional auto-encoders for hierarchical feature extraction International conference on artificial neural networks Berlin, Heidelberg. Springer; 2011.
[27] Odena, A. "Semi-supervised learning with generative adversarial networks." arXiv:1606.01583 (2016).
[28] Farabet C, Couprie C, Najman L, LeCun Y. Learning hierarchical features for scene labeling. IEEE PAMI 2013;35(8):1915–29.
[29] Goodfellow I, et al. Generative adversarial nets. Adv Neural Inf Process Sys 2014.
[30] Badrinarayanan, V., A. Kendall, and R. Cipolla. "Segnet: a deep convolutional encoder-decoder architecture for image segmentation." arXiv:1511.00561 (2015).