Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/MBE

MBE, 17(5): 5369–5394.
DOI: 10.3934/mbe.2020290
Received: 28 May 2020
Accepted: 03 August 2020
Published: 11 August 2020

Research article

A feature extraction and classification algorithm based on improved sparse auto-encoder for round steel surface defects

Xuguo Yan and Liang Gao *

State Key Lab of Digital Manufacturing Equipment and Technology, School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China

* **Correspondence:** Email: gaoliang@mail.hust.edu.cn; Tel: +86-27-87549419; Fax: +86-27-87549419.

Abstract: Traditional feature dimensionality reduction (FDR) algorithms can extract features by reducing feature dimensions. However, it may lose some useful information and affect the accuracy of classification. Normally, in traditional defect feature extraction, it first obtain the defect area of the defect image by image preprocessing and defect segmentation, select the original feature set of defects by prior knowledge, and extract the optimal features by traditional FDR algorithms to solve the problem of "curse of dimensionality". In this paper, a feature extraction and classification algorithm based on improved sparse auto-encoder (AE) is proposed. We adopt three traditional FDR algorithms at the same time, combine the defect features obtained in pairs, take the merged defect features as the input of sparse AE, then use the "bottleneck" of sparse AE to conduct the defects classification by Softmax classifier. The experimental results show that the proposed algorithm can extract the optimal features of round steel surface defects with less network training time than individual sparse AE, finally get higher classification accuracy than individual FDR algorithm in the actual production line.

Keywords: feature extraction; feature dimensionality reduction; auto-encoder; defect detection; round steel

1. Introduction

Round steel is a typical steel product, widely used in aerospace, transportation, civil and industrial construction, equipment manufacturing, marine engineering and other fields [1]. In the process of round steel production, due to the equipment failure, product material changes, product process fluctuations and other factors, the final products will inevitably have some surface quality problems, such as scratches, ears, cracks, scars, fold over, roll marks, etc [2]. Machine vision based defects detection has become one of the important technologies for modern steel enterprises to solve the product surface quality problems [3–7], which mainly includes image acquisition, image preprocessing, defect segmentation, defect feature extraction and classification [8]. Among all of them, an efficient and accurate defect feature extraction algorithm is core of the technology, so that enterprises can quickly and accurately detect the surface defects, adjust equipment parameters and optimize production processes.

In general, the defect features of steel surface mainly include color features [9–13], texture features [14–18] and shape features [19–23]. However, not all of the features are valuable for defects classification. Too many features will cause high dimensions of the classifier, which will increase network training time and reduce classification accuracy. On this basis, the feature dimensionality reduction (FDR) algorithms are adopted to find the optimal features from original image features, so as to reduce the complexity of classification and improve classification accuracy [24,25]. Principal component analysis (PCA), kernel principal component analysis (KPCA) and linear discriminant analysis (LDA) are the three commonly used traditional FDR algorithms. These FDR algorithms reduce the feature dimensions in particular theories, but all have certain limitations. The main one is, some useful information may be ignored, which will affect classification accuracy.

Deep learning can also be used for feature extraction [26–32]. It is s special type of machine learning and a kind of artificial intelligence [33]. With the development of deep learning, some scholars have begun to study the use of deep learning for steel surface defect detection. For example, Wang et al. [34] proposed a method of strip surface defect detection method based on deep learning. It took Resnet101 for feature extraction, then used the transfer learning method to train the network to ensure the stability and convergence of the network. Yi et al. [35] proposed a strip surface defect recognition system based on deep CNNs. It used original defect image as input and defect category as output for seven main classes of steel strip defects classification. In the practical application, due to the high dimensions of round steel surface defects, the network training of deep learning requires large amount of computation, which will increase the costs of enterprises.

In view of the above problems and causes, this paper proposed a feature extraction and classification algorithm based on improved sparse AE. It first selected the original feature set of round steel surface defects by the statistical features of the image, then adopts three traditional FDR algorithms at the same time to reduce the feature dimensions, combines these defect features obtained in pairs, takes the merged defect features as the input of sparse AE, and uses the "bottleneck" of sparse AE to conduct classification by Softmax. In order to verify the feasibility and effectiveness of proposed algorithms, we take four batches of round steel surface images collected on the actual production line as the test set, compare the defect classification accuracy of proposed algorithm with individual traditional FDR algorithm. As the consequence, the advantages of proposed algorithm can be proved.

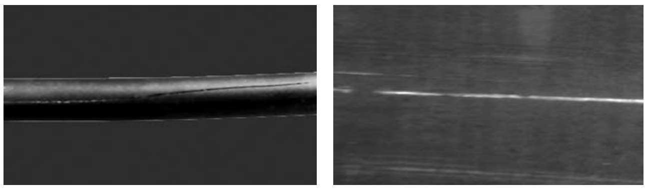
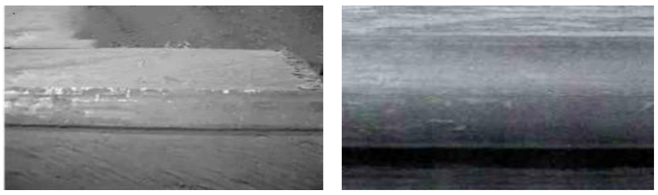
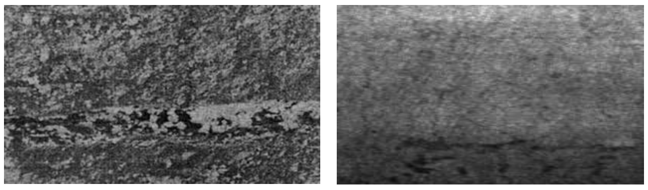
This paper is organized as follows: in Section 1, we give an overview of the research background and related works. In Section 2, we describe six typical round steel surface defects and their main image features, on this basis, the initial feature set is selected. In Section 3, we introduce the traditional FDR algorithms: PCA, KPCA and LDA and deep learning FDR algorithm: sparse AE, and their limitations in feature extraction. Our proposed algorithms are given in Section 4, which include: the feature extraction and classification algorithm based on PCA, KPCA and sparse AE

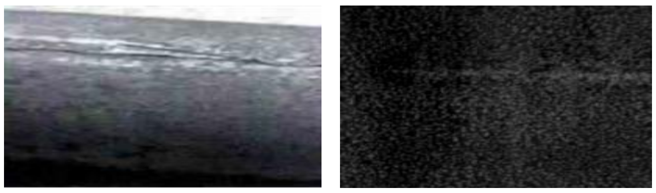
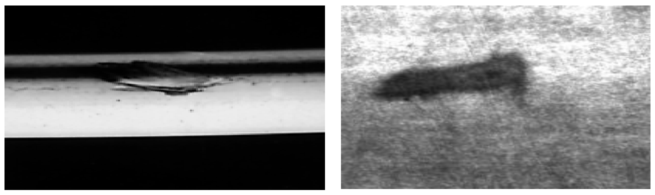
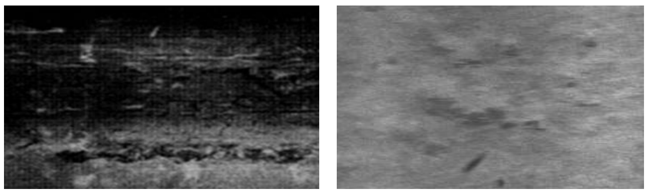
(PKAE), the feature extraction and classification algorithm based on PCA, LDA and sparse AE (PLAE) and the feature extraction and classification algorithm based on KPCA, LDA and sparse AE (KLAE). The experimental results are provided in Section 5, in which the algorithm with best classification accuracy and minimum network training time are selected as optimal algorithm, and the feasibility and effectiveness of the optimal algorithm are verified in an actual production line. Finally, concluding remarks and future work are presented in Section 6.

2. The initial feature set of round steel surface defects

2.1. Six typical round steel surface defects

The six typical round steel surface defects we selected in this research are: scratches, ears, cracks, scars, fold over and roll marks. The sample images of each defect and their main image features are shown as follows:

Scratches	
Image features	Generally, scratches are straight or arc-shaped. The defects are equal width linear stripes and bright in color. Compared with the background, the defects have obvious shape, texture and gray-scale differences. Therefore, they can be distinguished by shape, gray-scale and texture features.
Ears	
Image features	Generally, ears are stripe bulges and parallel to the axis. The defects may appear on one side or both sides of the products. Compared with the background, the defects have obvious shape and texture differences. Therefore, they can be distinguished by shape and texture features.
racks	
Image features	Generally, cracks are straight or Y-shaped. The direction of the defects basically consistent with rolling direction, and the defects are deep and black in color. Compared with the background, the defects have obvious shape, texture and gray-scale differences. Therefore, they can be distinguished by shape, gray-scale and texture features.

Fold over	
Image features	Fold over defects are similar to cracks, but the edges are zigzag. The defects distributed continuously or intermittently, with iron oxide inclusions in the middle and black in color. Compared with the background, the defects have obvious texture and gray-scale differences. Therefore, they can be distinguished by texture and gray-scale features.
Scars	
Image features	Generally, scars are tongue-shape or nail-shaped, sometimes in a close curve. Compared with the background, the defects have obvious shape and gray-scale differences. Therefore, they can be distinguished by shape and gray-scale features.
Roll marks	
Image features	Generally, roll marks are continuous or periodic depression and bulges, with different colors in the images. The occurrence of defects corresponds to the defects on roll handling equipment one by one. Compared with the background, the defects have obvious shape and gray-scale differences. Therefore, they can be distinguished by shape and gray-scale features.

2.2. Initial feature set of round steel surface defects

When using the traditional FDR algorithms to extract defect features, it is often based on prior knowledge to select a reasonable set of features as the initial feature set. On the basis of Section 2.1, the following features are selected as the initial feature set of round steel surface defects:

2.2.1. Shape features

• Perimeter

Perimeter of the defect is the length of boundary line between the defect area and the background. For binary image, if 1 is the target and 0 is the background, its perimeter is the number of pixels with a boundary value of 1. The calculation formula is as follows:

$$P = \sum_{(x,y)\in B} 1 \tag{1}$$

• Area

Area of the defect is the number of pixels occupied by the defect. For binary image, if 1 is the target and 0 is the background, its area is the number of pixels with a value of 1 in the defect part. The calculation formula is as follows:

$$A = \sum_{(x,y)\in R} 1 \tag{2}$$

• Centroid

Centroid of the defect is the center point of the defect. The calculation formula is as follows:

$$X = \frac{1}{A}\sum_{(x,y)\in R} x \tag{3}$$

$$Y = \frac{1}{A}\sum_{(x,y)\in R} y \tag{4}$$

• Compactness

Compactness represents the shape complexity of the defect. The smaller the value is, the simpler the defect shape is. The calculation formula is as follows:

$$C = \frac{P^2}{A} \tag{5}$$

• Linearity

Linearity is the maximum length of the defect measured from all directions. The smaller the value is, the more linear distribution the defect is. The calculation formula is as follows:

$$L = \frac{A}{P} \tag{6}$$

• Hu’s moment invariants

Hu’s moment invariants are the features with translation, rotation and size invariance. For image $f(x,y)$, the $p+q$ moment is defined as:

$$m_{pq} = \sum_x \sum_y x^p y^q f(x,y) \tag{7}$$

The central moment is defined as:

$$\mu_{pq} = \sum_x \sum_y (x-\bar{x})^p (y-\bar{y})^q f(x,y) \tag{8}$$

where $\bar{x}$ and $\bar{y}$ are the central coordinates of the defect: $\bar{x} = \frac{m_{10}}{m_{00}}$, $\bar{y} = \frac{m_{01}}{m_{00}}$.

The central moment can satisfy translation invariance in affine transformation. The normalized central moment formula of $f(x, y)$ is:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\ r}} \tag{9}$$

where $r = \frac{q+p}{2} + 1$.

Therefore, 7invariant matrices m1~m7 can be constructed by using the second and third order normalized central moments:

$$m_1 = \eta_{20} + \eta_{02} \tag{10}$$

$$m_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^{\ 2} \tag{11}$$

$$m_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \tag{12}$$

$$m_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \tag{13}$$

$$\begin{aligned} m_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})\left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] \\ &+ (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})\left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right] \end{aligned} \tag{14}$$

$$m_6 = (\eta_{20} - \eta_{02})\left[(\eta_{30} + \eta_{12})^2 - (\eta_{03} + \eta_{21})^2\right] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \tag{15}$$

$$\begin{aligned} m_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})\left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{03} + \eta_{21})^2\right] \\ &+ (3\eta_{12} - \eta_{30})(\eta_{03} + \eta_{21})\left[3(\eta_{30} + \eta_{12})^2 - (\eta_{03} + \eta_{21})^2\right] \end{aligned} \tag{16}$$

where η_{20} represents the extension range of the defect in the horizontal direction, η_{02} represents the extension range of the defect in the vertical direction, η_{11} represents the inclination of the defect, η_{30} represents the degree of shift of the defect gravity center in the horizontal direction, η_{03} represents the degree of shift of the defect gravity center in the vertical direction, η_{12} represents the degree of equilibrium of the vertical extension range of the defect, η_{21} represents the degree of equilibrium of the horizontal extension range of the defect.

2.2.2. Gray-scale features

The probability density function can be used to represent the gray-scale distribution of defect image. The first order probability distribution of image gray-scale is defined as:

$$p(b) = P\{F(x, y) = b\} \quad 0 \le b \le L - 1 \tag{17}$$

where b is the quantization level, L level in total. $F(x,y)$ is the gray-scale level of the pixel (x,y). Therefore, the first-order histogram is defined as:

$$P(b) \approx \frac{N(b)}{M} \tag{18}$$

where $N(b)$ is the total number of pixels with gray-scale value b, M is the total number of pixels. The gray-scale features obtained from the gray-scale histogram can describe the defect area of the defect image as a whole. On this basis, the gray-scale features selected in this research are:

• Gray-scale mean

Gray-scale mean represents the overall gray-scale level of defect area. The calculation formula is as follows:

$$\bar{b} = \sum_{b=0}^{L-1} bP(b) \tag{19}$$

• Gray-scale variance

Gray-scale variance represents the discrete degree of the gray-scale distribution of the defect area. The calculation formula is as follows:

$$\sigma_b^2 = \sum_{b=0}^{L-1} \left(b - \bar{b}\right)^2 P(b) \tag{20}$$

• Gray-scale skewness

Gray-scale skewness represents the degree of asymmetry of the gray-scale histogram between different defects. The calculation formula is as follows:

$$S = \frac{1}{\sigma^3} \sum_{b=0}^{L-1} \left(b - \bar{b}\right)^3 P(b) \tag{21}$$

• Gray-scale kurtosis

Gray-scale kurtosis represents the density of the gray-scale histogram around the gray-scale mean. The calculation formula is as follows:

$$K = \frac{1}{\sigma^4} \sum_{b=0}^{L-1} \left(b - \bar{b}\right)^4 P(b) - 3 \tag{22}$$

• Gray-scale energy

Gray-scale energy is used to measure the amount of information in the image. When the probability of gray-scale distribution of the defect image is equal, the minimum value is taken. The calculation formula is as follows:

$$P_G = \sum_{b=0}^{L-1} P(b)^2 \tag{23}$$

• Gray-scale entropy

Gray-scale entropy is also used to measure the amount of information in the image, which represents the amount of information contained in the aggregation feature of gray-scale distribution in the defect image. When the probability distribution of defect image is equal, it reaches the maximum value. The calculation formula is as follows:

$$E = -\sum_{b=0}^{L-1} P(b)\log_2\left[(P(b))\right] \tag{24)}$$

2.2.3. Texture features

Gray-scale co-occurrence matrix (GLCM) is an effective method for texture feature selection. It starts from the pixel (x, y) whose gray-scale level is i, and counts the probability $p(i, j, \delta, \theta)$ of its simultaneous occurrence with the pixel whose distance is δ, direction is θ ($\theta = 0^\circ, 45^\circ, 95^\circ, 135^\circ$) and gray-scale level is j $(x+\Delta x, y+\Delta y)$. The expression of GLCM is:

$$p(i, j, \delta, \theta) = \left\{(x, y), (x+\Delta x, y+\Delta y) \middle| f(x, y) = i, f(x+\Delta x, y+\Delta y) = j\right\} \tag{25}$$

where $x = 0,1,2,\ldots, N_x - 1$, $y = 0,1,2,\ldots, N_y - 1$, $i, j = 0,1,2,\ldots, L-1$. L is the gray-scale level of the image; x, y is the pixel coordinate in the image; N_X and N_y are the number of rows and columns of the image respectively.

According to GLCM, a large number of texture features can be defined. In this case, the following 4 most commonly used features are selected in this research:

• Texture energy

Texture energy represents the texture roughness of the defect image. The calculation formula is as follows:

$$Q_1 = \sum_{i=0}^{L-1}\sum_{j=0}^{L-1}[p(i, j)]^2 \tag{26}$$

where $p(i, j)$ expresses the joint probabilities between two pixels with gray level i and j in distance δ and direction θ.

• Texture inertia

Texture inertia represents the texture depth of the defect image. The larger the value is, the deeper the texture is. The calculation formula is as follows:

$$Q_2 = \sum_{i=0}^{L-1}\sum_{j=0}^{L-1}(i, j)^2 p(i, j) \tag{27}$$

• Texture entropy

Texture entropy represents the complexity of the texture in the defect image. The larger the value is, the greater the randomness of texture is. The calculation formula is as follows:

$$Q_3 = -\sum_{i=0}^{L-1}\sum_{j=0}^{L-1} p(i,j)\log_2 p(i,j) \tag{28}$$

• Texture correlation

Texture correlation represents the texture direction of the defect image. The larger the value is, the stronger the texture is in this direction than in other directions. The calculation formula is as follows:

$$Q_4 = \sum_{i=0}^{L\text{-}1}\sum_{j=0}^{L\text{-}1} \frac{i * j \cdot p(i,j) - \mu_x \mu_y}{\sigma_x \sigma_y} \tag{29}$$

where μ_x and σ_x represent the mean value and standard deviation of gray-scale rows; μ_y and σ_y represent the mean value and standard deviation of gray-scale columns.

GLCM only represents the comprehensive information of the defect image in certain range and angle of change, we also need to calculate its variance and inverse variance to represent the texture features of the whole image more comprehensively:

• Texture variance

Texture variance represents the texture's period of the defect image. The calculation formula is as follows:

$$Q_5 = \sum_{i=0}^{L\text{-}1}\sum_{j=0}^{L-1} (i-\mu)^2 p(i,j) \tag{30}$$

• Texture inverse variance

Texture inverse variance represents the texture changes in the defect image.The larger the value is, the slower the texture changes. The calculation formula is as follows:

$$Q_6 = \sum_{i=0}^{L\text{-}1}\sum_{j=0}^{L\text{-}1} \frac{p(i,j)}{1+(i,j)^2} \tag{31}$$

In summary, the initial feature set of round steel surface defects we selected in this research is shown in Table 1.

No.19~No.24 represent the texture features of the defect image in one direction. We need to calculate the texture features in 4 directions: 0°, 45°, 90°, 135°. Therefore, 42 features are selected as the initial feature set of round steel surface defects.

3. Feature dimensionality reduction (FDR) algorithms

3.1. Principal Component Analysis

PCA is a multivariate statistical method proposed by Pearson in 1901, and developed by Hotelling in 1993 [36]. It is the most widely used FDR algorithm. The main idea of PCA is to map n-dimensional features to k-dimensions, which is a set of orthogonal features also known as the principal components. The process of PCA is to find a set of mutually orthogonal coordinate axes in order from the original space. The first new axis selection of PCA is the direction of the largest variance in the original data, the second new axis selection is the plane orthogonal to the first axis to make the largest variance, and the third axis is the plane orthogonal to the first and second axes to make the largest variance. By analogy, we can get n such axes. From the new coordinate axes obtained in this way, it can be found that most of the variance is contained in the first k coordinate axes, and the variance of the latter is almost 0. Therefore, we can ignore the remaining coordinate axes and only keep the first k coordinate axes with most of the variance. In fact, this is equivalent to only retaining the dimension features that contain most of the variance, while ignoring the feature dimensions that contain almost 0 variance, so as to realize dimensionality reduction of features.

Table 1. Initial feature set of round steel surface defects.

No	Feature	Feature description	Feature type
1	P	Perimeter	Shape feature
2	A	Area	Shape feature
3	X/Y	Centroid	Shape feature
4	C	Compactness	Shape feature
5	L	Linearity	Shape feature
6	m_1	First order moment	Shape feature
7	m_2	Second order moment	Shape feature
8	m_3	Third order moment	Shape feature
9	m_4	Fourth order moment	Shape feature
10	m_5	Fifth order moment	Shape feature
11	m_6	Sixth order moment	Shape feature
12	m_7	Seventh order moment	Shape feature
13	$\bar{b}$	Gray-scale mean	Gray-scale feature
14	σ_b^2	Gray-scale variance	Gray-scale feature
15	S	Gray-scale skewness	Gray-scale feature
16	K	Gray-scale kurtosis	Gray-scale feature
17	P_G	Gray-scale energy	Gray-scale feature
18	E	Gray-scale entropy	Gray-scale feature
19	Q_1	Texture energy	Texture feature
20	Q_2	Texture inertia	Texture feature
21	Q_3	Texture entropy	Texture feature
22	Q_4	Texture correlation	Texture feature
23	Q_5	Texture variance	Texture feature
24	Q_6	Texture inverse variance	Texture feature

When PCA is used for feature extraction, it should meet the requirements that the variance of samples after dimensionality reduction is as large as possible, the mean square error is as small as possible. PCA looks for the principal axis direction which is used to effectively represent the common features of the same kind of samples [37], which is very effective for representing the common features of the same kind of data samples, but it is not suitable to distinguish different sample classes [38]. Therefore, PCA is generally not used for feature extraction, but for dimensionality reduction. It needs to be combined with other FDR algorithms to achieve the purpose of feature extraction.

3.2. Linear Discriminant Analysis (LDA)

LDA is a probabilistic model-based algorithm proposed by Blei et al in 2003 [39]. It is a supervised FDR algorithm. Compared with PCA, PCA and LDA have different constraints in the process of projection. The idea of PCA is "the projection method that makes the average points of different categories farthest away", so its main function is to remove the redundant dimensions of the original data. For data set with different distributions, the maximum variance target after PCA projection will result in data sample mixed or no longer linearly separable. However, the basic idea of LDA is to select an optimal projection direction, so that the data of the same category after projection is closely, and the data of different categories are as far away from each other as possible. The goal of LDA is to reduce the dimensionality of labeled data and project it into a low dimensional space, which meets three conditions at the same time: keep as much information as possible (that is, select the largest feature value as the direction represented by the corresponding feature vector), find the best projection direction that makes samples as easy to distinguish as possible, makes the same type of samples as close as possible and different types of samples as far as possible after projection.

As PCA has advantages in dimensionality reduction, LDA has better distinguishing ability, combine these two algorithms can effectively improving the ability of feature extraction [40].

3.3. Kernel Principal Component Analysis (KPCA)

Many machine learning algorithms assume that the input data is linearly separable. However, in real world, we may face non-linear problems in most cases [41]. In this case, PCA and LDA are not the optimal algorithms. KPCA can realize the non-linear dimensionality reduction of data, which is used to process the non-linear separable data set [42]. The basic idea of KPCA is: for the matrix X in the input space, we first map all the samples in X to a high-dimensional or even infinite dimensional feature space with a non-linear mapping to make it linearly separable, and then carry out PCA in this high-dimensional space.

Compared with PCA, KPCA can extract the information of the original data to the maximum extent, but it also has the same problem as PCA, that is, it is not suitable to distinguish different sample classes [43]. Therefore, KPCA and LDA can also be combined to better extract the features.

3.4. Auto-encoder (AE)

Deep learning is a feature learning method [44]. With the increase of network level, the more complex features are easier to learn, so it is suitable for processing the data with high dimensions. However, too many layers of network will lead to long training time and easy to over-fit. Auto-encoder (AE) is the basic model in deep learning and it can also be used for feature extraction [45]. The standard AE is a typical three-layer neural network unsupervised learning model, which includes an input layer, a hidden layer and an output layer, in which the input layer and the output layer have the same dimensions [46]. AE can learn the implicit features of input data, which is called "coding", at the same time, it can reconstruct the original input data with new features, which is called "decoding". Compared with traditional feature extraction algorithm, AE can better extract the features of original data. The expression of AE is as follows:

$$x \xrightarrow{f(x)} h \xrightarrow{g(x)} x' \tag{32}$$

In the training process, a loss function is designed to make the input and output of the encoder as similar as possible, that is:

$$x \approx x' \tag{33}$$

where $x' = g(f(x))$.

In general, the mean square error can be used to measure the similarity between the input data and the output data, and its expression is as follows:

$$L(x, g(f(x))) = E_{x \sim p_{data}} \|x - g(f(x))\|^2 \tag{34}$$

where $E_{x \sim p_{data}}$ indicates that sample x meets the expectation of sample distribution *p*.

In the learning process, the mean square error may become very small, which will lead to over-fitting. The expectation of AE is to have coding ability for the same type of data, that is to say, the network is required to have strong generalization ability. In this case, the L1 regularization or the Kullback-Leibler divergence (KLD) penalty item needs to be added to the loss function to make AE learn the sparse feature, that is sparse AE. The expression of two loss functions are as follows:

$$L_1(x, g(f(x))) = E_{x \sim p_{data}} \|x - g(f(x))\|^2 + \lambda \sum_i |h_i| \tag{35}$$

where λ is the regular parameter, which is used to balance loss function and regular term in penalty regression. h_i is the excitation value of the *i*th neuron.

$$L_2(x, g(f(x))) = E_{x \sim p_{data}} \|x - g(f(x))\|^2 + \beta \sum_{j=1}^{M} \left[\rho \cdot \log \frac{\rho}{\hat{\rho}_j} + (1-\rho) \cdot \log \frac{1-\rho}{1-\hat{\rho}_j} \right] \tag{36}$$

where β is the weight of control sparsity penalty factor. ρ is the sparsity parameter, generally, its value is set to be close to 0. $\hat{\rho}_j$ is the average excitation value of neuron *j*, which is calculated on the training set. If the value of $\hat{\rho}_j$ is close to 0, only a small number of samples can make this neuron work, which is called the sparse restriction. Since most of the hidden neurons are expected to be "inactive" to learn the specific structures, $\hat{\rho}_j$ is expected to be close to ρ.

When the activation function of hidden layer is Sigmoid, the output value of hidden layer is between (0,1), KLD penalty item can be used. When the activation function of hidden layer is ReLU, the output value of hidden layer is between $(0, +\infty)$, the L1 regulation function can be used.

In practical application, we can stack multiple sparse AE to obtain a better feature extraction result, which is called the stacked AE. However, the multi-layer network structure will increase the complexity of network training.

4. Feature extraction and classification based on improved sparse AE

We choose 6 kinds of typical round steel surface defect sample images collected in the actual production line as training set, and each defect contains 500 images. All the 3000 images are preprocessed and the defect parts are segmented. In Section 2.2, 42 features are selected as the initial feature set of round steel surface defects. On this basis, the horizontal direction of the data matrix is 3000 defect images and the vertical direction of the data matrix is 42 feature parameters. In our proposed algorithms, the feature matrix of 3000×42 is operated as follows (Figure 1):

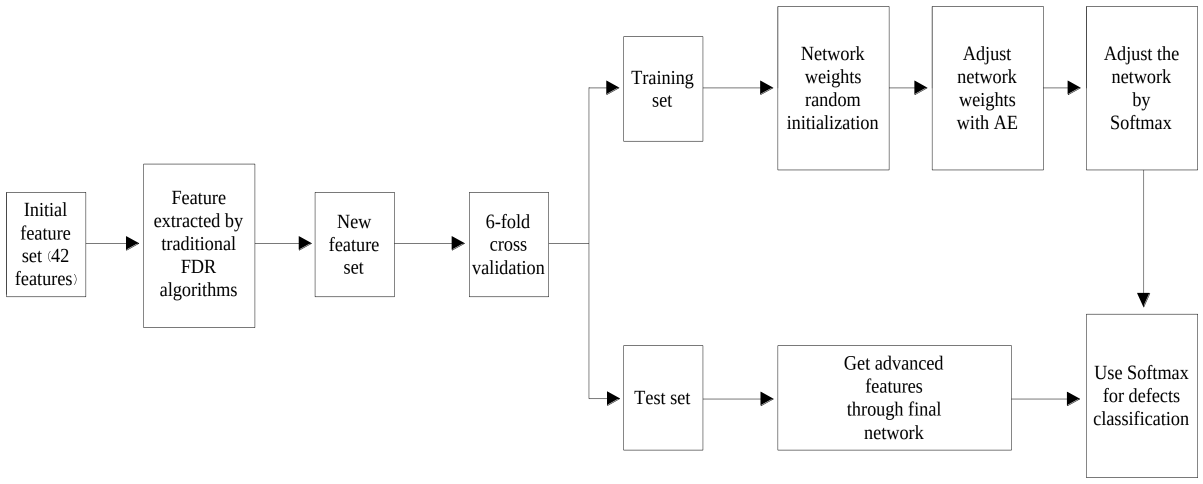

Initial feature set（42 features）
Feature extracted by traditional FDR algorithms
New feature set
6-fold cross validation
Training set
Network weights random initialization
Adjust network weights with AE
Adjust the network by Softmax
Test set
Get advanced features through final network
Use Softmax for defects classification

Figure 1. Feature extraction and classification based on improved sparse AE.

An architecture overview of the network of purposed algorithms is shown in Figure 2. The shaded part represents the neurons that are not activated in sparse AE. The sparse AE is pre-trained, then as the input of the Softmax classifier.

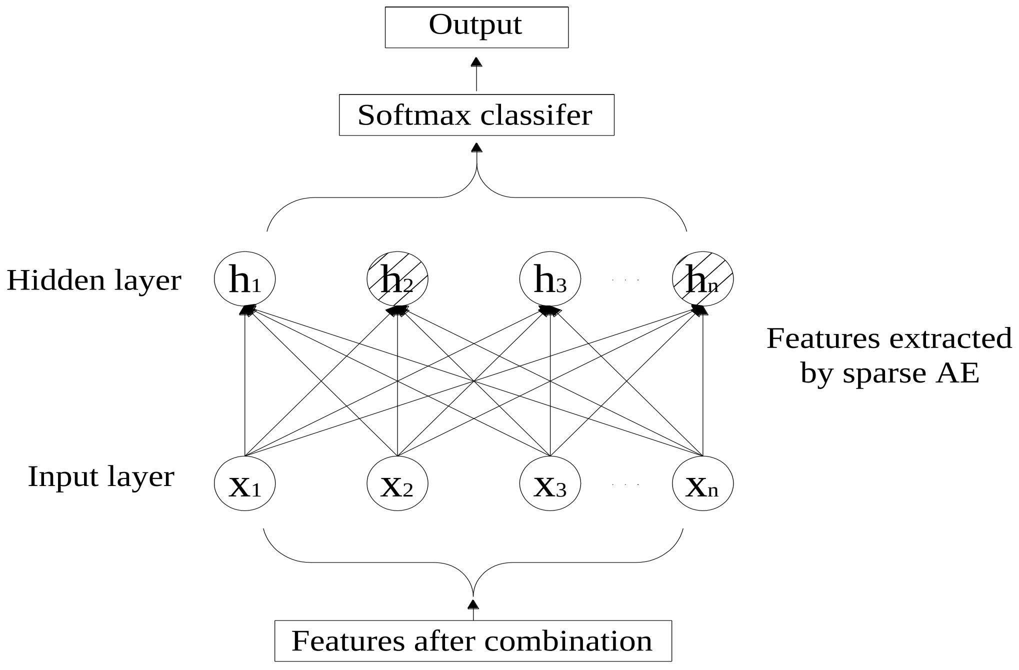


Figure 2. The architecture overview of the network.

The Softmax classifier is derived from statistical Logistic regression. The core idea of Logistic regression is to use the logical regression method in classification, which can judge the input data and then output a single discrete result to classify the extracted features. It first determines the probability of the extracted features and then classifies them. For the dataset, it gives the probability sum of all vectors to be 1. In Softmax, the specific mapping probability given by the function makes the sum of the probabilities of all categories 1. The Softmax functions are as follows:

$$f_j(z) = \frac{e^{z_j}}{\sum_k e^{z_k}} \tag{37}$$

Where z is the vector of inputs, the results are mapped from the exponential domain to the probability, and the sum of probabilities is guaranteed to be 1. The Softmax loss function mainly uses the cross-entropy loss function, and its formula is as follows:

$$L_i = -\log\left(\frac{e^{f_i}}{\sum_j e^{f_j}}\right) \tag{38}$$

For the Softmax loss function, it can be understood from the point of view of information theory, and the loss function can be regarded as the entropy of two probabilities, and its formula is as follows:

$$H(p,q) = -\sum_x p(x)\log q(x) \tag{39}$$

The purpose of the loss function is to measure the error between the true classification result and the predicted classification result, and then optimize and modify it based on this value. Where p represents the probability of true classification, and q represents the probability of predicting classification.

4.1. PKAE

PCA is a linear FDR algorithm and KPCA is a nonlinear FDR algorithm. If PCA or KPCA is used for feature extraction, the feature extracted is the linear or nonlinear representation of the original input data. However, the relationship between round steel surface defects is complex. There may be not only linear relationships, but also non-linear relationships. It is necessary to combine PCA and KPCA features to obtain more complex features, then as the input of sparse AE to extract optimal features. In order to verify the performance of the Softmax classifier, k-fold cross validation is adopted. The original training set is divided into six parts, one of which is taken as the test set each time, and the other five parts are taken as the training set. Iterating 20 times, and taking the mean value of classification accuracy as the final experimental results.

Suppose that the sample matrix is X ($X = (x_{ij})_{n\times m}$). Where n is the total number of samples, m is the number of features and x_{ij} represents the jth feature value of the ith sample, where $i = 1,2,\ldots n$, $j = 1,2,\ldots,m$. The specific steps of PKAE are as follows:

4.1.1. Feature extracted by PCA

• Step 1

Standardize the original data, all the samples are subtracted from the mean value of corresponding feature. The calculation formula of the mean value is as follows:

$$\overline{x_j} = \frac{1}{n}\sum_{i=1}^{n} x_{ij} \tag{40}$$

• Step 2

Calculate the covariance matrix P ($P = (r_{jk})_{m\times m}$), where m is the number of features. r_{jk} represents the correlation between the *j*th and *k*th feature, where $j = 1,2,\ldots,m$, $k = 1,2,\ldots,m$:

$$P = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1m} \\ r_{21} & r_{22} & \cdots & r_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ r_{m1} & r_{m2} & \cdots & r_{mm} \end{bmatrix} \tag{41}$$

• Step 3

Calculate the eigenvalue λ_i and the eigenvector e_i of the covariance matrix *P*:

$$\lambda_1 e_i = Pe_i \tag{42}$$

• Step 4

Record the resulting eigenvalues in the order of large to small: $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_k$, calculate the contribution rate of each principal component. The formula of contribution rate is as follows:

$$\frac{\lambda_g}{\sum_{g=1}^{k}\lambda_g} \tag{43}$$

where $g = 1,2,\cdots,k$. The higher the contribution rate is, the stronger the information of the original variables contained in the principal component is. Generally, the top eigenvalues with cumulative variance contribution is more than 85% are considered.

• Step 5

Transform the original sample matrix X into a new matrix Y_1 ($Y_1 = (Y_{ij})_{n\times m_1}$), where $i = 1,2,\ldots,n$, $j = 1,2,\ldots,m_1$:

$$Y_1 = X \times \lfloor e_1, e_2, \cdots e_{m_1} \rfloor \tag{44}$$

where $\lfloor e_1, e_2, \cdots e_{m_1} \rfloor$ represents a new feature space composed of m1 feature vectors. m1 are the principal components extracted by PCA.

4.1.2. Feature extracted by KPCA

• Step 1

Mapping each original sample data x_i into high dimensional space by nonlinear function $\Phi(x)$:

$$x \to \Phi(x_i) \tag{45}$$

where $i = 1,2,\ldots,n$.

• Step 2

Calculated the kernel matrix $[K]_{ij} = K_{ij} = (\Phi(x_i), \Phi(x_j))$. Use Gaussian function as the kernel function, then the expression function of K is as follows:

$$K(x_i, x_j) = \exp(-\frac{\|x_i - x_j\|^2}{\delta^2}) \tag{46}$$

where δ is the width, given by experience.

• Step 3

Calculate the central kernel matrix:

$$\tilde{K} = K - l_n K - K l_n - l_n K l_n \tag{47}$$

• Step 4

Calculate the eigenvalue and eigenvector of $\tilde{K}$

• Step 5

Record the resulting eigenvalues in order of large to small: $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_k$, calculate the contribution rate of each principal component. Generally, the top eigenvalues with cumulative variance contribution is more than 95% are considered.

• Step 6

Transform the original sample matrix X into a new matrix Y2 ($Y_2 = (Y_{ij})_{n \times m_2}$), where $i = 1,2,\ldots,n$, $j = 1,2,\ldots,m_2$:

$$Y_2 = X \times [e_1, e_2, \cdots e_{m_2}] \tag{48}$$

4.1.3. Feature extracted by PKAE

• Step 1

Add KPCA features belonging to the same sample to PCA features to get a new sample matrix

Y ($Y = [Y_1, Y_2]$):

$$Y = (Y_{ij})_{n \times (m_1 + m_2)} \tag{49}$$

• Step 2

The new sample matrix is divided in to training set and test set by 6-fold cross validation. Use sparse AE to learn the optimal features. Sigmoid is adopted as the activation function of hidden layer, in this case, the equation (36) is used as the loss function of sparse AE.

• Step 3

Add Softmax classifier at the top layer of the network, use the labeled data to train Softmax, and use back-propagation (BP) algorithm to adjust the weights in the network. The Softmax loss is used as the loss function.

• Step 4

Train the whole network and conduct classification by Softmax.

4.2. PLAE

PCA and LDA are both linear FDR algorithms, combine the features by these two FDR algorithms, then as the input of sparse AE, can learn the linear relationship and the possible hidden nonlinear relationship between round steel defects. In order to verify the performance of the Softmax classifier, k-fold cross validation is adopted. The original training set is divided into six parts, one of which is taken as the test set each time, and the other five parts are taken as the training set. Iterating 20 times, and taking the mean value of classification accuracy as the final experimental results.

Suppose that the sample matrix is X ($X = (x_{ij})_{n \times m}$). n is the total number of samples, m is the number of features, and x_{ij} represents the *j*th feature value of the *i*th sample, where $i = 1,2,\dots n$, $j = 1,2,\dots,m$. The specific steps of PLAE are as follows:

4.2.1. Feature extracted by PCA

This step is the same as Section 4.1, after feature extracted by PCA, the original sample matrix X is transformed into a new matrix Y_1.

4.2.2. Feature extracted by LDA

LDA uses the following Fisher criteria to obtain the projection vector:

$$J(\alpha) = \arg\max_{\alpha} \frac{\alpha^T S_b \alpha}{\alpha^T S_w \alpha} \tag{50}$$

where S_b represents the inter-class dispersion matrix, and S_w represents the intra-class dispersion matrix. The calculation formulas are as follows:

$$S_b = \frac{1}{n}\sum_{i=1}^{c} n_i (u_i - u)(u_i - u)^T \tag{51}$$

$$S_w = \frac{1}{n}\sum_{i=1}^{c}\sum_{j=1}^{n_i}(x_j^i - u_i)(x_j^i - u_i)^T \tag{52}$$

where $u_i = \frac{1}{n_i}\sum_{j=1}^{n_i} x_j^i$ represents the average value of the features of all samples belonging to class *i*, and $u = \frac{1}{n}\sum_{i=1}^{c}\sum_{j=1}^{n_i} x_j^i$ represents the average value of the features of all samples. *c* represents there are a total of *c* defect types.

The optimal projection vector α is the eigenvector corresponding to the maximum eigenvalue of the equation: $S_w\alpha = \lambda S_b\alpha$. Let $[\alpha_1, \alpha_2, \dots \alpha_k] \in R^{m\times k}$ be the set of the first k eigenvectors selected, where *k*=*c*-1. Then we can transform the original sample matrix X into a new matrix Y2 ($Y_2 = (y_{ij})_{n\times k}$), where $i = 1,2,\dots n$, $j = 1,2,\dots,k$:

$$Y_2 = X \times [\alpha_1, \alpha_2, \dots, \alpha_k] \tag{53}$$

4.2.3. Feature extracted by PLAE

• Step 1

Add LDA features belonging to the same sample to PCA features to get a new sample matrix: $Y = [Y_1, Y_2]$

• Step 2

The new sample matrix is divided in to training set and test set by 6-fold cross validation. Use sparse AE to learn the optimal features. Sigmoid is adopted as the activation function of hidden layer, in this case, the equation (36) is used as the loss function of sparse AE.

• Step 3

Add Softmax classifier at the top layer of the network, use the labeled data to train Softmax, and use back-propagation (BP) algorithm to adjust the weights in the network. The Softmax loss is used as the loss function.

• Step 4

Train the whole network and conduct classification by Softmax.

4.3. KLAE

KPCA is a nonlinear FDR algorithm. The feature extracted by LDA is the linear representation of the original data. Combining these two FDR algorithms can extract the linear and nonlinear relationship of the round steel surface defects, then as the input of sparse AE to extract optimal features. In order to verify the performance of the Softmax classifier, k-fold cross validation is adopted. The original training set is divided into six parts, one of which is taken as the test set each time, and the other five parts are taken as the training set. Iterating 20 times, and taking the mean value of classification accuracy as the final experimental results.

Suppose that the sample matrix is X ($X = (x_{ij})_{n\times m}$). n is the total number of samples, m is the number of features, and x_{ij} represents the *j*th feature value of the *i*th sample, where $i = 1,2,\dots n$, $j = 1,2,\dots,m$. The specific steps of KLAE are as follows:

4.3.1. Feature extracted by KPCA

This step is the same as Section 4.1, after feature extracted by KPCA, the original sample matrix X is transformed into a new matrix Y_1 .

4.3.2. Feature extracted by LDA

This step is the same as Section 4.2, after feature extracted by LDA, the original sample matrix X is transformed into a new matrix Y_2 .

4.3.3. Feature extracted by KLAE

• Step 1

Add LDA features belonging to the same sample to KPCA features to get a new sample matrix: $Y = [Y_1, Y_2]$

• Step 2

The new sample matrix is divided in to training set and test set by 6-fold cross validation. Use sparse AE to learn the optimal features. Sigmoid is adopted as the activation function of hidden layer, in this case, the equation (36) is used as the loss function of sparse AE.

• Step 3

Add Softmax classifier at the top layer of the network, use the labeled data to train Softmax, and use back-propagation (BP) algorithm to adjust the weights in the network. The Softmax loss is used as the loss function.

• Step 4

Train the whole network and conduct classification by Softmax.

5. Experimental results and discussions

The data samples used in the experiment are 3000 selected round steel surface defect images as introduced in Section 4, which include 6 kinds of defects and 500 images of each defect. The images are preprocessed and defect parts are segmented. In order to verify the advantage of our proposed algorithms, we compared proposed algorithms with individual sparse AE and individual traditional FDR algorithm in classification accuracy. At the same time, we also provided a comparison of network training time between the proposed algorithms and individual sparse AE. Through the comparisons, the algorithm with best classification accuracy and minimum network training time are selected as the optimal algorithm. Finally, the feasibility and effectiveness of the optimal algorithm are verified in the context of actual production, 4 batches of round steel surface images collected on an actual production line were used as the test set to compare the defects classification accuracy between optimal algorithm and individual traditional FDR algorithm.

5.1. Experimental environment and parameter settings

The experimental environment was win 7, 32-bit system, i7 processor, 3.4 GHz main frequency and 4GB memory. All the programmings were completed on Matlab 2014. In order to obtain more advanced and complex features, the Sigmoid was selected as the active function. At the same time, the number of network training will affect the classification accuracy. After several experiments, the number of network training was selected as 1500, and the learning rate was set as 0.5. In the actual production line, the detection of surface defects, the change of production process parameters, the adjustment of energy data and the diagnosis of equipment operation parameters all share an ERP system, in this case, each function unit is required not to occupy too much computing resources. Therefore, the lower the network complexity is and the less network training time is, the better.

5.2. Results and discussions

In order to retain as much information as possible, in this research, the cumulative principal component contribution rate of PCA and KPCA was selected as 99%. The kernel function of KPCA is Gaussian kernel function. As a result, 15 features were remained after PCA, 12 features were remained after KPCA. The features remained after LDA were the number of defect types minus 1 (that is 5).

The number of hidden layer nodes will affect the performance of final network. At present, the common method to determine the number of hidden layer nodes is trail and error method. The same sample set is used to train the network with different numbers of hidden layer nodes, and the value corresponding to the minimum network error is selected. If the number of nodes in the input layer is n and the number of nodes in the output layer is l, the determination of the initial number of nodes in the hidden layer m is generally calculated by the following empirical formulas:

$$m = \sqrt{n+l} + \alpha \tag{54}$$

where α is a constant between 1 and 10.

$$m = \log_2 n \tag{55}$$

$$m = \sqrt{nl} \tag{56}$$

The advantage of sparse AE is that when the number of nodes in the hidden layer is large, it can learn the advanced features of input data by adding sparsity constraints. Therefore, formula (56) was adopted to determine the initial number of nodes in the hidden layer in this paper, then increased the number of nodes in hidden layer for training and compared the error of network with different numbers of nodes in hidden layer. The results are as follows (from Table 2 to Table 4):

Table 2. The number of nodes in hidden layer comparison (PKAE).

Number of nodes in hidden layer	**27**	**29**	**31**
Error	0.00025471	0.00025449	0.00026228

Table 3. The number of nodes in hidden layer comparison (PLAE).

Number of nodes in hidden layer	**20**	**22**	**24**
Error	0.00012659	0.00012046	0.00011305

Table 4. The number of nodes in hidden layer comparison (KLAE).

Number of nodes in hidden layer	**17**	**19**	**21**
Error	0.00011545	0.00010816	0.00010538

It can be seen from Table 2 to Table 4 that, in most cases, with the increase of the number of nodes in the hidden layer, the output error of the network decreases slightly, but not obviously. At the same time, when the number of nodes in the hidden layer is equal to the input layer and output layer, the error value is far less than the error requirement of 0.001. Therefore, in order to reduce the network complexity and reduce the network training time, the number of nodes in the hidden layer was set to be the same as the input layer and the output layer. In this case, in PKAE, the node number of input layer, hidden layer and output layer were the same, which was the sum of dimensions of PCA features and KPCA features (that is 27). In PLAE, the node number of input layer, hidden layer and output layer were the same, which was the sum of dimensions of PCA features and LDA features (that is 20). In KLAE, the node number of input layer, hidden layer and output layer were the same, which was the sum of dimensions of KPCA features and LDA features (that is 17). The experimental results of defects classification accuracy are shown in Table 5. For the better comparison, Softmax classifier is used for defects classification by all algorithms.

It can be seen from Table 5:

(1) The classification accuracy of the proposed algorithms (PKAE, PLAE and KLAE) are significantly better than individual traditional FDR algorithm and sparse AE.

(2) The classification accuracy of PKAE is worse than PLAE and KLAE. This may be due to the similarity between different types of round steel surface defects, and LDA has the advantage in distinguishing different defect types.

(3) The classification accuracy of KLAE is better than PLAE. It is proved that there are both linear and nonlinear relationships between round steel surface defects.

(4) KLAE has the best classification accuracy, and the classification accuracy on scratches reaches 99.12%. However, the classification accuracy of fold over and cracks are relatively low, which may be due to the similarity between these two defect types in shape and grayscale features, and sometimes it is difficult to distinguish them in the image.

Table 5. Defects classification accuracy comparison.

	PCA	LDA	KPCA	Sparse AE	PKAE	PLAE	KLAE
Scratches	88.53%	69.97%	89.31%	92.01%	92.54%	94.36%	99.12%
Cracks	80.12%	60.01%	81.43%	84.36%	88.23%	90.49%	98.41%
Ears	85.33%	66.30%	85.19%	90.62%	90.77%	93.12%	98.87%
Fold over	81.04%	61.12%	81.90%	85.27%	89.01%	91.22%	97.96%
Scarring	82.67%	63.32%	82.77%	87.30%	91.95%	92.97%	98.64%
Roll marks	87.26%	66.43%	97.94%	90.43%	90.83%	93.75%	98.79%

We also compared the network training time of the proposed algorithms with individual sparse AE. The experimental results of network training time are shown in Table 6.

It can be seen from Table 6 that compared with individual sparse AE, the network training time of the proposed algorithms is greatly reduced. Among of them, KLAE requires less network training time, which is 45.99s. As KLAE has advantages in both classification accuracy and network training time, we finally chose KLAE as the final feature learning network.

Table 6. Feature learning time comparison (s).

Algorithm	Sparse AE	PKAE	PLAE	KLAE
Time	624.26	70.62	46.80	45.99

The experimental results of defects classification sensitivity and specificity were also provided to prove the effectiveness of KLAE. The results are shown in the table below (Table 7).

It can be seen from the Table 7 that KLAE has both high defects classification sensitivity and specificity, so the algorithm proposed in this paper is very effective.

Table 7. Defects classification accuracy, sensitivity and specificity of KLAE.

	Accuracy	Sensitivity	Specificity
Scratches	99.12%	98.59%	99.23%
Cracks	98.41%	97.30%	98.63%
Ears	98.87%	98.17%	99.01%
Fold over	97.96%	96.61%	98.22%
Scarring	98.64%	97.75%	98.82%
Roll marks	98.79%	98.01%	98.94%

In order to further verify the feasibility of KLAE in the practical industrial application, 4 batches of round steel surface images collected on the actual production line were used as the test set to compare the defects classification accuracy of KLAE and traditional FDR algorithm. The defects classification process of KLAE is shown in Figure 3:

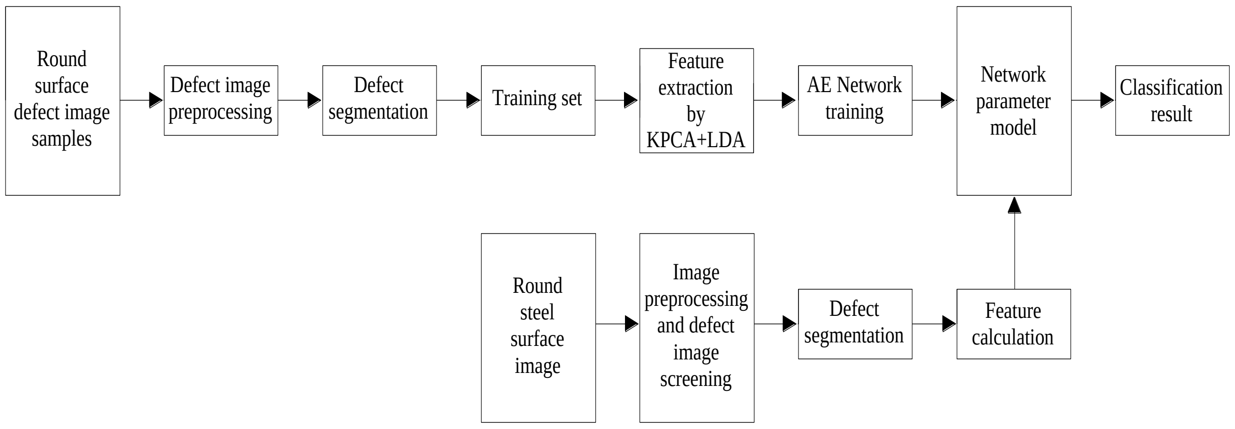

Round surface defect image samples
Defect image preprocessing
Defect segmentation
Training set
Feature extraction by KPCA+LDA
AE Network training
Network parameter model
Classification result
Round steel surface image
Image preprocessing and defect image screening
Defect segmentation
Feature calculation

Figure 3. The process of defects classification in actual production line.

After image preprocessing and defect image screening, the number of defect images in each batch is: 401, 428, 500, 379. Then the features of defect part were extracted by KLAE, and classified by Softmax. The experimental results are shown in Table 8.

As can be seen from Table 8 compared with traditional FDR algorithm, KLAE can significantly improves round steel surface defects classification accuracy in the actual production line. In actual production process, there may be multiple defects coverage in one round steel surface image, which may affect the accuracy of classification. In general, the defects classification accuracy in Table 5 can satisfy the actual production application requirements. In this case, the proposed algorithm is feasible and effective.

Table 8. Defects classification results in actual production line.

	PCA	**LDA**	**KPCA**	**KLAE**
Batch 1	84.15%	65.51%	89.19%	98.64%
Batch 2	86.19%	67.59%	84.33%	98.91%
Batch 3	85.53%	66.79%	85.10%	99.23%
Batch 4	87.74%	60.96%	81.29%	98.62%

6. Conclusion

Surface defects defection is particularly important for enterprises to meet different production standards and requirements. Normally, in defect feature extraction, we first obtain the defect area of the defect image by image preprocessing and defect segmentation, select the original feature set of defects by prior knowledge, evaluate the importance of selected features by using the traditional FDR algorithms and extract the optimal features according to the importance rank of features. In this paper, a feature extraction and classification algorithm based on improved sparse AE is proposed. We adopt three traditional FDR algorithms at the same time, combine the defect features obtained in pairs, take merged defect features as the input of sparse AE, then use the "bottleneck" of sparse AE to conduct defects classification. The experimental results show that our proposed algorithm KLAE can extract the optimal features of round steel surface defects with less network training time than individual AE, and finally get higher classification accuracy than individual traditional FDR algorithm in the actual production line.

Due to the limitation of time and conditions, this research still has some limitations. In the future research, improvements can be made in the following aspects:

(1) Parameters in the network, such as the number of nodes in the hidden layer, the number of network training times and the activation function are all given based on experience, which will affect the classification accuracy of the final network.

(2) Considering that stacked AE may increase the complexity of network training, this research only use the basic sparse AE with one hidden layer, which is proved can satisfy the actual production application requirements. However, stacked AE is also valuable for study to obtain better defect classification accuracy.

(3) With the increase of production, new types of defects will also appear. In this research, we have considered the 6 typical round steel surface defects. In the future research, more abundant defects information can be obtained by updating the defect sample database to better classify the defects in the actual production line.

Funding

This research was funded by the Natural Science Foundation of China (NSFC) under Grant no. 51435009, Science and technology innovation 2030 - a major project of new generation artificial intelligence: virtual reality integration and adaptive production scheduling of complex manufacturing under Grant no. 2018AAA0101704 and supported by "Program for HUST Academic Frontier Youth Team under Grant no. 2017QYTD04."

Conflict of interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

1. B. Wu, T. Xue, T. Zhang, S. H. Ye, A novel method for round steel measurement with a multi-line structure light vision sensor, *Meas. Sci. Technol.*, **21** (2010), 283–293.
2. C. I. Brannan, M. A. Bedell, J. L. Resnick, J. J. Eppig, M. A. Handel, D. E. Williams, et al., Developmental abnormalities in Steel17H mice result from a splicing defect in the steel factory sytoplasmic tail, *Gene Dev.*, **10** (1992), 1832–1842.
3. Z. Y. Wang, Research on steel plate surface defects detection method based on machine vision, *Comput. Modern.*, **7** (2013), 97–117.
4. C. Xie, T. Z. Xie, Key technology of detecting hot heavy rail steel surface faults based on machine vision, *J. Chongqing. Univ.*, **36** (2013), 16–21.
5. Q. W. Luo, Y. G. He, A cost-effective and automatic surface defect inspection system for hot-rolled flat steel, *Robot Com-int. Manuf.*, **38** (2016), 16–30.

6. F. Meriaudeau, G. Lavallee, E. Fauvet, Machine vision prototype for defect detection on metallic tubes, *SPIE- Int. Soc. Opt. Eng., Proc.*, **4664** (2002), 190–197.
7. B. Tang, J. Kong, X. Wang, L. Chen, Steel surface defect recognition based on support vector machine and image processing. *China Mechan. Eng*., **22** (2011), 1402–1405.
8. J. Iiarinen, K. Heikkinen, J. Rauhanmaa, A defect detection scheme for web surface inspection, *Int. J. Pattern Recogn*., **14** (2000), 735–755.
9. J. V. D. Weijer, T. Gevers, A. D. Bagdanov, Boosting color saliency in image feature detection, *IEEE T. Pattern Anal*., **28** (2005), 150–156.
10. Z. W. Qiu, T. W. Zhang, New image color feature extraction method, *J. Harbin Ins. Tech.*, **36** (2004), 1699–1701.
11. Y. G. Wang, J. Yang, Y. Zhou, Y. Z. Wang, Region partition and feature matching based color recognition of tongue image, *Pattern Recogn. Lett*., **28** (2007), 11–19.
12. S. S. Patil, A. V. Dusane, Use of color feature extraction technique based on color distribution and relevance feedback for content based image retrieval, *Int. J. Comput. Appl*., **52** (2012), 9–12.
13. W. T. Chen, W. H. Liu, M. S. Chen, Adaptive color feature extraction based on image color distributions, *IEEE T. Image Process*, **19** (2010), 2005–2016.
14. Z. H. Tang, Y. Y. Sun, W. H. Gui, J. P. Liu, Flotation froth image texture feature extraction based on wavelet transform, *Comp. Eng*., **37** (2011), 206–208.
15. C. Y. Pang, J. K. Liu, Improved LFP algorithm on Leukocyte image texture feature extraction and recognition, *Acta Photon. Sinica*., **42** (2013), 1375–1380.
16. Q. Liu, X. P. Liu, L. J. Zhang, L. M. Zhao, Image texture feature extraction & recognition of Chinese herbal medicine based on gray level co-occurrence matrix, *Adv. Mater. Res*., **605–607** (2012), 2240–2244.
17. V. Yu, D. Ruan, D. Nguyen, T. Kaprealian, K. Sheng, SU-F-R-17: Advancing Glioblastoma Multiforme (GBM) recurrence detection with MRI image texture feature extraction and machine learning, *Med. Phys*., **43** (2016), 3376–3377.
18. B. Yuan, B. Xia, D.Zhang, Polarization image texture feature extraction algorithm based on CS-LBP operator, *Procedia. Comp. Sci*., **131** (2018), 295–301.
19. D. X. Wei, X. Y. Chen, R. C. Xu, Image shape feature extraction method based on corner point detection, *Comp. Eng.,* **36** (2010), 220–222.
20. O. Mari, N. Seishi, Plant Shape Discrimination of several taxa without shape feature extraction using neural networks with image input, *Breed. Sci*., **50** (2000), 189–196.
21. B. Vijayalakshmi, A new shape feature extraction method for leaf image retrieval, *4th Int. Conf. Signal Image (SIPRO)*, **221** (2013), 235–245.
22. T. Meruliya, P. Dhameliya, J. Patel, D. Panchal, P. Kadam, S. Naik, Image processing for fruit shape and texture feature extraction—review, *Int. J. Comput. Appl*., **129** (2015), 30–33.
23. C. Li, Q. Cao, Extraction method of shape feature for vegetables based on depth image, *Trans. Chin. Soc. Agric. Mach*., **43** (2012), 242–245.
24. L. M. Bruce, C. H. Koger, L. Li, Dimensionality reduction of hyperspectral data using discrete wavelet transform feature extraction, *IEEE T. Geosci. Remote*, **40** (2002), 2331–2338.
25. P. Pudil, J. Novovičová, J. Kittler, Floating search methods in feature selection, *Pattern Recogn. Lett.*, **15** (1994), 1119–1125.
26. E. K. Wang, N. Zhe, Y. Li, Z. D. Liang, Y. Ye, A spare deep learning model for privacy attack on remote sensing images, *Math. Biosci. Eng*., **16** (2019), 1300–1312.

27. H. Y. Zhao, C. Che, B. Jin, X. P. Wei, A viral protein identifying framework based on temporal Convolutional network, *Math. Biosci. Eng.*, **16** (2019), 1709–1717.
28. Z. C. Zhang, Y. Zhang, T. Zhou, Y. L. Pang, Medical assertion classification in Chinese EMRs using attention enhanced neural network, *Math. Biosci. Eng.*, **16** (2019), 1966–1977.
29. E. K. Wang, L. Xi, R. Sun, F. Wang, L. Pan, A new deep learning model for assisted diagnosis on electrocardiogram, *Math. Biosci. Eng.*, **16** (2019), 2481–2491.
30. H. J. Deng, L. X. Peng, J. J. Zhang, C. M. Tang, H. L. Fang, H. H. Liu, An intelligent aerator algorithm inspired by deep learning, *Math. Biosci. Eng.*, **16** (2019), 2990–3002.
31. S. Y. Chen, Y. Zhang, Y. H. Zhang, J. J. Yu, Y. X. Zhu, Embedded system for road damage detection by deep Convolutional neural network, *Math. Biosci. Eng.*, **16** (2019), 7982–7994.
32. Y. X. Zhang, L. C. Jin, B. Wang, D. H. Hu, L. Q. Wang, P. Li, et al., DL-CNV: A deep learning method for identifying copy number variations based on next generation target sequencing, *Math. Biosci. Eng.*, **17** (2020), 202–215.
33. Y. Lecun, Y. Bengio , G. Hinton, Deep learning, *Nature*, **521** (2015), 436.
34. L. Z. Wang, S. Q. Guan Strip steel surface defect recognition based on deep learning, *J. Xi'an Polytech. Univ*., **31** (2017), 669–674.
35. L. Yi, G. Li, M. Jiang, An end-to-end steel strip surface defects recognition system based on convolutional neural networks, *Steel Res. Int*., **87** (2016), 1–5.
36. H. H. Hotelling, Analysis of Complex Statistical Variables into Principal Components. *Br. J. Educ. Psychol*., 24 (1932), 417–520.
37. M. T. Kluger, H. Owen, Advantages of PCA exaggerated? *Anaesth. Intens. Care*, **18** (1990), 588–589.
38. M. Rezghi, O. Askar, Noise-free principal component analysis: An efficient dimension reduction technique for high dimensional molecular data, *Expert. Syst. Appl*., **41** (2014), 7797–7804.
39. D. M. Blei, A. Y. Ng, M. I. Jordan, J. Lafferty, Latent dirichlet allocation, *J. Mach. Learn. Res.,* **3** (2003), 993–1022.
40. W. Chen, J. E. Meng, S. Q. Wu, PCA and LDA in DCT domain, *Pattern Recogn. Lett.,* **26** (2005), 2474–2482.
41. E. Shchepakina, Black swans and canards in self-ignition problem, *Nonlin. Anal-real.,* **4** (2003), 45–50.
42. J. N. Wu, J. Wang, L. Liu, Feature extraction via KPCA for classification of gait patterns, *Hum. Movem. Sci.,* **26** (2007), 393–411.
43. L. J. Cao, K.S. Chua, W. K. Chong, H. P. Lee, Q. M. Gu, A comparison of PCA, KPCA and ICA for dimensionality reduction in support vector machine, *Neurocomputing,* **55** (2003), 321–336.
44. L. Liu, K. L. Liu, Z. H. Cong, J. L. Zhao, Y. F. Ji, J. He, Long Length Document Classification by Local Convolutional Feature Aggregation, *Algorithm.,* 11 (2018), 109.
45. D. Mellado, C. Saavedra, S. Chabert, R. Torres, R. Salas, Self-improving generative artificial neural network for pseudorehearsal incremental class learning, *Algorithm,* **12** (2019), 206.
46. F. Aziz, A. S. W. Wong, S. Chalup, Semi-supervised manifold alignment using parallel deep autoencoders, *Algorithms,* **12** (2019), 186.