Journal of Computers Vol. 30 No. 6, 2019, pp. 168-180
doi:10.3966/199115992019123006013

Glass Defect Recognition Method Based on Improved Convolutional Neural Networks

Dan-Dan Zhang[1], Yong Jin[*], Bin-Yu Hu[1]

[1] Institute of Information and communication, North Univ. of China, Taiyuan 030051, China
jinyong601@163.com

Received 7 April 2018; Revised 23 August 2018; Accepted 5 October 2018

Abstract. The features of the same type of glass defects are quite different, thus it is difficult to accurately recognize the type of glass defects. This paper proposes an improved Convolutional Neural Networks model to solve the problem of glass defects recognition. Image processing methods are used to reduce the noise of the image, so that the edge of the defect can be clearly recognized. A sparse auto-encoder is used to learn the feature of the input samples, and trained weights are used as the convolution kernel of the network to increase the speed of recognition. K-means clustering is used instead of softmax classifier of Convolutional Neural Networks. The defect images are first clustered and then classified to improve the recognition accuracy. Experimental results show that the recognition accuracy of this method reaches up to 96%.

Keywords: convolution neural network, glass defect recognition, image processing, k-means clustering, parse auto-encoder

1 Introduction

With the development of the country, the demand for glass from industry and the public is increasing, and the demand for high-quality glass is increasing. The appearance quality of glass is an important factor affecting the performance of its products. The quality of the glass is measured by detecting whether the surface is defective. Due to the destruction of the process system in the glass production engineering or the errors in the operation process, the glass often has defects of different types and sizes. For example, inclusions, bubbles, tin, pitting, lines and scratches [1], not only affect the appearance quality of the product, but also affect the transparency and optical uniformity of the glass, reducing the mechanical strength and thermal stability of the product. Therefore, the recognition of glass defects in the production process is of great significance for improving product quality level and guiding process improvement.

Currently, glass defect recognition mainly uses two different methods: (1) Extract the geometric features of the glass defects for recognition. For example, wavelet transform [2], gray level co-occurrence matrix [3] and other techniques to extract the grayscale and texture features of glass defects to improve the defect representation ability, but glass defects have no fixed shape features and detail features, target size and grayscale difference Smaller. If it is described by the same type of features, it is only used for a specific glass image, which is contrary to the characteristics of the glass defect itself. (2) Use a classifier for recognition. For example, error back propagation neural network (BP neural network) [4], support vector machine (SVM) [5], etc. as classifiers. In [4], the three-layer feedforward neural network using BFGS algorithm is used as a classifier for identification, but the fixed learning rate leads to the disadvantage that BP algorithm has relatively long training time. In [5], the multi-strategy support vector machine in the statistical pattern recognition theory is used as a classifier to recognize the defect type, but it requires a large memory and a slow speed. With the rapid development of computer technology and electronic technology in recent years, the automatic recognition technology based on machine learning has also developed rapidly, providing a reliable and reliable solution for glass surface

* Corresponding Author

quality inspection.

Deep learning is a new field in machine learning research. In-depth machine learning methods are divided into supervised learning and unsupervised learning. The learning models established under different learning frameworks are different. For example, Convolutional Neural Networks (CNNs) is a deep machine learning model under supervised learning, and Deep Belief Nets (DBNs) is a machine learning model under unsupervised learning. CNNs have an advantage in image recognition compared to DBNs. CNNs have the features of autonomous learning. The images can be directly input into the features of network learning defect targets without complicated preprocessing. Even images that interfere with illumination, geometric distortion and other factors have a good recognition effect [6]. Through the method of structural reorganization and weight reduction, the feature extraction function is integrated with the convolutional neural network in the multi-layer perceptron, which is successfully applied in face recognition [7] and license plate recognition [8]. This avoids the need for complex image pre-processing and cumbersome manual extraction processes in traditional glass defect identification methods. Li et al. [9] proposed an efficient depth C-NNs model for image deblocking by considering the effect of local small patches to mitigate the conflict between bit reduction and quality preservation. Zang et al. [10] proposed an adaptive convolutional neural network (ACNN), which can determine the structure of CNN without performance comparison. Praveen et al. [11] proposed an unsupervised feature learning method based on the Stacked Sparse auto-encoder (SSAE) framework, which is used to automatically learn the features of accurate segmentation of stroke lesions from brain MR images. However, it uses SVM as a classifier, which requires disadvantages such as a relatively large memory and a slower speed.

Therefore, this paper proposes an improved convolutional neural network model (SK_CNN) using a combination of supervised and unsupervised. Firstly, the piecewise linear transformation method in the image processing method is used to reduce the original noise of the image and realize the image preprocessing work. Then the KL distance and L1 norm are introduced as the sparse term of the sparse auto-encoder (SAE), and the input sample features are learned by the sparse auto-encoder. The trained weights are used as the convolution kernel of the convolutional neural network to improve the recognition speed. Finally, the k-means clustering algorithm is used as the classifier of the network, which effectively improves the recognition efficiency. The model proposed in this paper is verified in AR database and MNIST database respectively, and the recognition effect is good.

2 Convolution Neural Network

The convolutional neural network is a feedforward neural network whose artificial neurons can respond to a surrounding area of a part of the coverage and have excellent performance for large image processing.

The structural part of the convolutional neural network is shown in Fig. 1, including the following, the input layer is the input layer, which is generally the training set picture. The C1 layer is a convolutional layer, that is, a feature extraction layer, and the input of each neuron is connected to a local receptor (detector) of the previous layer, and the local feature is extracted, and once the local feature is extracted, it is combined with other features. The positional relationship between them is also determined. The S2 layer is a down-sampling layer, that is, a feature mapping layer. Each computing layer of the convolutional neural network is composed of a plurality of feature maps, and each feature is mapped into a plane, and the weights of all the neurons on the plane are equal. After several processes, it enters the F5 layer, the fully connected layer, which connects all the previous feature maps, similar to the full connection of the neural network. The Output layer is a classifier that classifies the entire feature map. Therefore, a basic convolutional neural network framework is obtained.

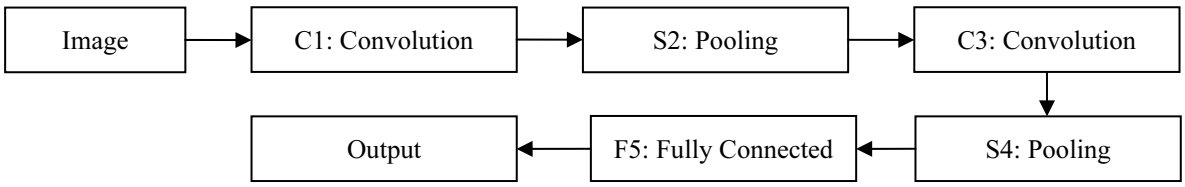


Fig. 1. Convolutional neural network framework

2.1 Forward Comunication Phase

According to [12], the feature graphs x_j^l of each C layer may be convoluted by multiple feature maps x_j^{l-1} obtained from multiple convolution cores and the previous layer. The formula for calculating the j feature map x_j^l of the l level C layer is shown in Eq 1.

$$x_j^l = f(\sum_{i \in M_j} x_i^{l-1} * b_j^l). \tag{1}$$

Where l is the number of layers; $f(\cdot)$ is an incentive function; M_j is the previous layer feature map collection; w_{ij}^l is the convolution kernel; b_j^l is the unique additive bias of the convolution map x_j^l.

The feature map x_j^l of each S layer is only obtained by locally averaging the corresponding feature maps x_j^{l-1} of the previous layer. The calculation formula of the j feature map x_j^{l-1} of the l level S layer is shown in Eq 2.

$$x_j^l = f(\beta_j^l down(x_i^{l-1}) + b_j^l). \tag{2}$$

Where $down(\cdot)$ is a sub-sampling function; l is the number of layers; $f(\cdot)$ is an incentive function; β_j^l is the unique multiplicative bias of the feature map x_j^1; b_j^l is the unique additive bias of the feature map x_j^1.

2.2 Reverse Correction Phase

According to [12], the weight update is carried out in the reverse correction phase, and the C layer weights from t to $t+1$ are updated, as in Eq 3.

$$w(t+1) = w(t) + \frac{\partial E}{\partial w(t)}. \tag{3}$$

The S layer weights are updated, as in Eq 4.

$$\beta(t+1) = \beta(t) + \eta \frac{\partial E}{\partial \beta(t)}. \tag{4}$$

Where η is a network learning rate; E is a square error cost function for the existence of a total of N training samples of the C class in the multi classification problem.

$$E = \frac{1}{2} \sum_{n=1}^{N} \sum_{k=1}^{C} (y_k^n - o_k^n)^2. \tag{5}$$

Eq 5 show, where y_k^n is the k dimension value of the ideal output of the n training sample. o_k^n is the k value of the actual output of the n training sample after the forward propagation stage, and C is the number of types to be classified.

3 Improved Convolution Neural Network Construction

The recognition rate of the CNN depends on the structure of the network. However, as for how to determine the network structure, there is no prior knowledge as the basis. Generally, as the number of neurons in each layer increases, the feature information extracted by the network is more perfect, which is beneficial to the improvement of recognition accuracy. However, this requires sufficient training samples to ensure adequate network training. In addition, blindly increasing the number of neurons leads

to an increase in the computational complexity of the convolution kernel, which may slightly increase the network recognition rate, but the network training time will be longer in [13].

In order to solve this problem, we use a combination of supervised and unsupervised methods, where we use the sparse auto-encoder to optimize the weights so that realize the improvement on training time. Moreover, the traditional CNN classifier softmax [14] is replaced by k-means [15] to improve the recognition accuracy. The flow chart shown in Fig. 2.

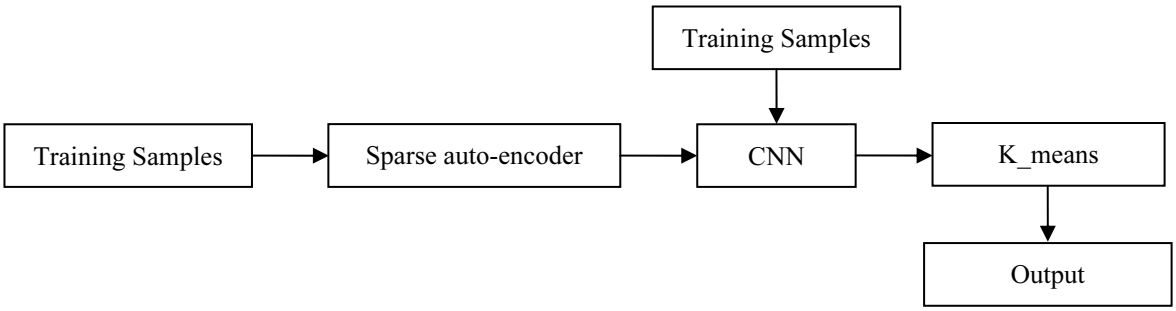


Fig. 2. Improved convolution neural network model

3.1 Sparse Auto-encoder

Sparse auto-encoding means that the hidden layer features have sparse response features. The pre-training structure is shown in Fig. 3. First, extracting the small block information of the input picture through the input layer to the hidden layer is to encode the input data, and then passing through the hidden layer to the output layer is a process of decoding the input data to reconstruct a small picture. In general, the introduction of sparsity in hidden layer neurons can bring many advantages: (1) the coding scheme has large storage capacity, associative memory capability, and simple calculation; (2) make the structure of natural signals clearer; (3) the coding scheme is in line with the energy minimum economic strategy that is universal in biological evolution, and also meets the conclusions of electrophysiological experiments.

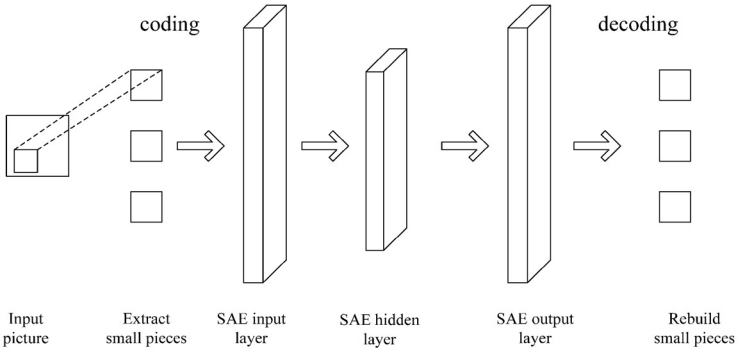


Fig. 3. Sparse auto-encoder pre-training framework

This paper will introduce two kinds of sparsity constraints, one is to consider the relationship between the dimension of the hidden layer feature and the input dimension, and use the KL distance to introduce the sparsity constraint [11]; the second is that the dimension of the hidden layer feature is greater than When entering the dimension, sparsity is introduced using the norm L1 regular term.

First, introduce sparsity constraints using KL distances: for datasets with hidden layer features:

$$\{x^n \in R^u\}_{n=1}^N \to \{X^n \in R^v\}_{n=1}^N. \tag{6}$$

Calculate the average of each node of the hidden layer output using the following formula:

$$\overline{X} = \frac{1}{N}\sum_{n=1}^{N} X^n \in R^v. \tag{7}$$

Let the average output value of each node of the hidden layer be 0 as much as possible, and most of the hidden layer nodes are in a silent state. In order to quantify the characteristics of the hidden layer, it is usually assumed that each node of the hidden layer responds with a certain probability, and between nodes Independent. Let each hidden layer node respond or occur with the expected probability of $\rho = 0.05$. The sparse regular term constructed using the KL distance is:

$$KL(\rho \| \overline{X}(j)) = \rho \log \frac{\rho}{\overline{X}(j)} + (1-\rho) \log \frac{1-\rho}{1-\overline{X}(j)}. \tag{8}$$

Where $\overline{X}(j)$ is the jth element of $\overline{X}$, which is the average of the response of the jth node of the hidden layer, where j=1, ..., v. The optimized objective function of the resulting sparse self-coding network is:

$$\min_{\theta} J(\theta) = \frac{1}{N} \sum_{n=1}^{N} \| \hat{x}^{(n)} - x^{(n)} \|_2^2 + \lambda \cdot R(\theta) + \beta \cdot \sum_{j=1}^{v} KL(\rho \| \overline{X}(j)). \tag{9}$$

The first term is the reconstruction error term, constructed with the L2 norm, using the L-BFGS method to minimize the error term, the second term is the regular term to prevent overfitting, and the third term is the introduced KL sparse term. To control the probability of neuron activation.

Second, using the norm L1 regularity to introduce sparsity, the norm constraint for the output structure of the hidden layer node is:

$$\frac{1}{N} \sum_{n=1}^{N} \| X^{(n)} \|_1 = \frac{1}{N} \sum_{n=1}^{N} | X^{(n)}(j) |. \tag{10}$$

The optimized objective function of the resulting sparse auto-encoding network is:

$$\min_{\theta} J(\theta) = \frac{1}{N} \sum_{n=1}^{N} \| x^{(n)} - x^{(n)} \|_2^2 + \lambda \cdot R(\theta) + \beta \cdot \frac{1}{N} \sum_{n=1}^{N} \| X^{(n)} \|_1 . \tag{11}$$

3.2 K-means Clustering

K-means algorithm [16-17] is a typical distance-based clustering algorithm, using distance as the evaluation index of similarity, which is, the closer the two objects are, the greater their similarity. The algorithm considers that the cluster is composed of objects near the distance, so a compact and independent cluster is taken as the ultimate goal. Suppose a given sample set D={x1, x2, …, xm}.

(1) Select k objects randomly from D data objects as the initial cluster centers;

(2) According to the mean (center object) of each clustering object, the distance between each object and these central objects is calculated, and the corresponding objects are divided according to the minimum distance;

(3) Recalculate the mean of each (changeable) cluster (center object);

(4) Cycle (2) to (3) until each cluster does not change.

The square error criterion and the maximum number of iterations are usually used as stop conditions. The maximum number of iterations is related to the size of the data set and the degree of aggregation. The square error is defined as shown in Eq 12.

$$E = \sum_{i=1}^{k} \sum_{p \in c_i} (p - c_i)^2. \tag{12}$$

Where p is the sample in the sample space and c_i is the centroid of cluster c_i (Cluster). But finding the optimal solution for it requires examining all possible cluster partitions of sample set D. This is an NP-hard problem. Therefore, the k-means algorithm adopts a greedy strategy and approximates the solution by iterative optimization (12).

4 Experiments

4.1 Database

Glass defects are the primary factors affecting the quality of glass. Glass defects are divided into solid phase inclusions and gas phase inclusions. Solid inclusion, refers to the addition of glass bubbles outside knob, stripes, stones, fog spots, scratches and other defects in glass. Common glass defects are mainly bubbles, light distortion, scratches, inclusions, neoplasms. Therefore, we choose inclusions, bubbles, scratches, stains, nodules, broken board as our research defect types. Due to the small amount of data collected, the distribution of positive and negative samples in the data set is often uneven, and over-fitting often occurs. To prevent over-fitting, regularization methods [18-19] or data set amplification methods can be used [20]. Because our data samples are relatively small, we use the data set amplification method to eliminate the over-fitting phenomenon and improve the recognition accuracy.

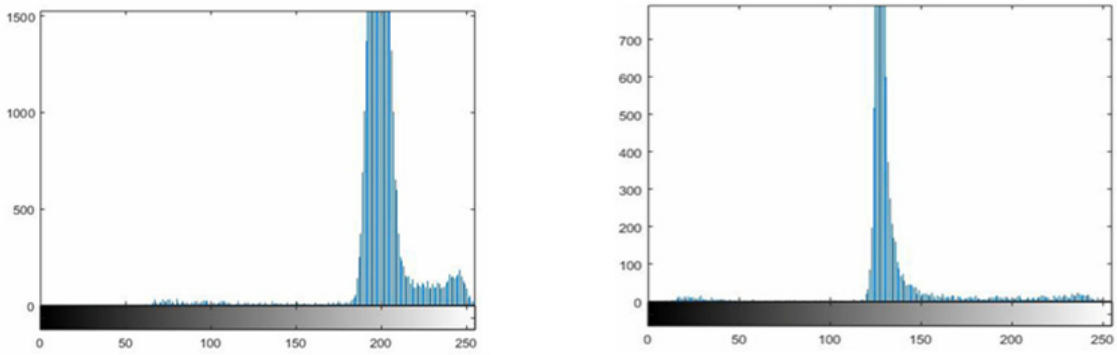
First pan the original image: Move 20 units to the left and right in the horizontal direction centering on the defects on the picture, and move 10 units up and down. Then, the angle θ is rotated with the translated image as a sample, and the rotation step is 100 from 00-900. The final part of the image transformation is expanded by the $\begin{bmatrix} -1 & 0 & f_w \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ horizontal mirroring formula and the $\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & f_H \\ 0 & 0 & 1 \end{bmatrix}$ vertical mirroring formula, and finally 1800 defective samples are obtained.

However, in this process, the defect maps due to inclusions, tumors, bubbles, stains and scratches are greatly affected by system noise, so that the edge portions of the defects cannot be clearly identified, thereby reducing system noise and increasing defects and background. For the difference in gray scale, a median filter and a gray scale transform method can be used. However, some small defects and edge defects will cause some damage when filtering out noise, so we use the piecewise linear transformation method. The piecewise linear transformation formula is as shown in Eq 13.

$$g(x,y)=\begin{cases} (c/a)(x,y) & 0<f(x,y)<a \\ \dfrac{d-c}{b-a}(x,y)+c & a\le f(x,y)\le b \\ \dfrac{255-d}{255-b}[f(x,y)-b+d] & b<f(x,y)\le 255 \end{cases} . \quad (13)$$

Where control points a and b are the grayscale ranges that need to be converted, and c and d determine the grayscale slope of the linear transformation.

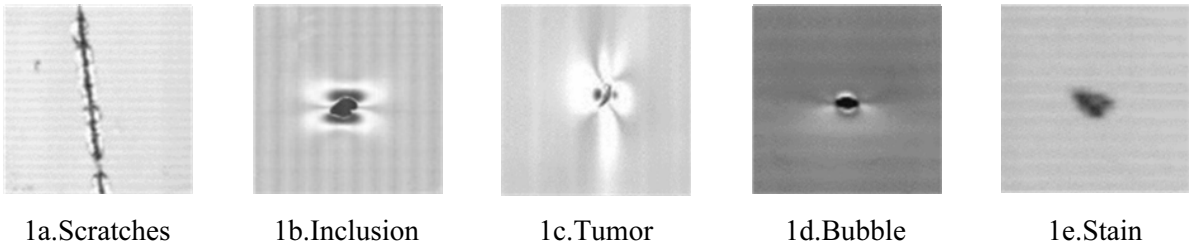
Taking the inclusions as an example, a piecewise linear transformation is performed on the gray value of its characteristic image. As shown in Fig. 4. The defect image before and after the piecewise linear transformation is shown in Fig. 5 to Fig. 6.



(a) Gray histogram before piecewise linear transformation

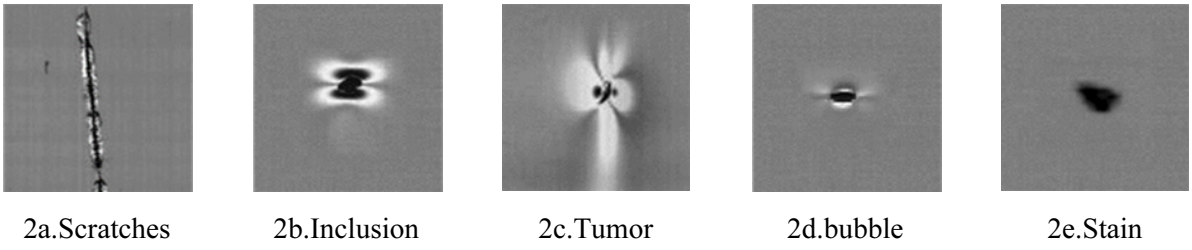
(b) Gray histogram after piecewise linear transformation

Fig. 4. Grayscale histogram of inclusions defects before and after piecewise linear transformation



1a.Scratches 1b.Inclusion 1c.Tumor 1d.Bubble 1e.Stain

Fig. 5. Defect image before piecewise linear transformation



2a.Scratches 2b.Inclusion 2c.Tumor 2d.bubble 2e.Stain

Fig. 6. Defect image after piecewise linear transformation

4.2 Experimental Procedure

The experimental data is 1800 pictures which are collected by Coupled Charge Device (CCD) camera from in-field production line. The pictures have six types of defects and each type of defect presents in 300 pictures. 200 pictures from each type are selected as training picture which means 1200 pictures are chosen as training pictures from 1800 pictures. The other 600 pictures are test pictures. Pictures are unified as 28 * 28 grayscale pictures. The research on small data sets first uses deep convolutional neural networks to train and extract features, of which relu acts as an activation function. The parameter settings are shown in Fig. 7:

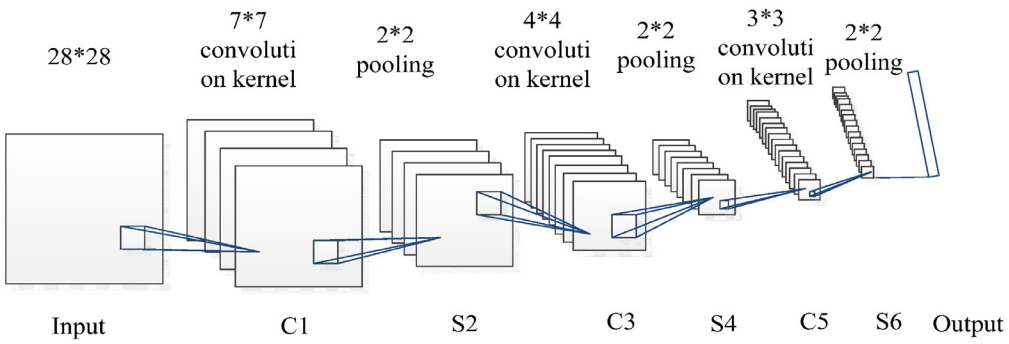


Fig. 7. Deep convolutional neural network

The training set is taken as an input picture, and after Fig. 7, the S2 layer feature map h1 and the S4 layer feature map h2 are respectively extracted. Then use them as input images for SAE. The following is the processing of SK_CNN in Fig. 8:

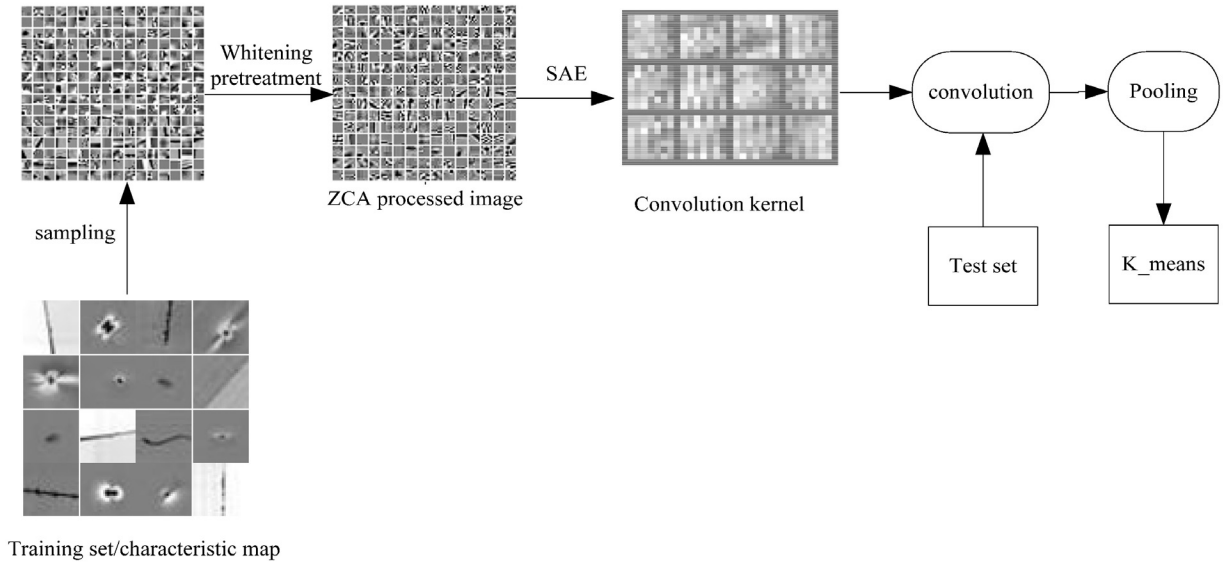


Fig. 8. SK_CNN model

(1) Local image blocks are randomly selected from the training samples for Zero-phase Component Analysis (ZCA) whitening.

(2) The processed image is pre-trained by sparse auto-encoding (this process introduces a sparse term with KL) to obtain the weight W1.

(3) The feature map h1 performs ZCA whitening processing.

(4) The processed image is pre-trained to obtain the weight W2 through sparse auto-encoding (this process introduces a sparse term with the L1 norm).

(5) The feature map h2 performs ZCA whitening processing.

(6) The processed image is pre-trained to obtain the weight W3 through sparse auto-encoding (this process introduces a sparse term with the L1 norm).

(7) W1, W2, and W3 get the optimal weight after undergoing supervised training by CNNs.

(8) After the test set is input into the CNNs, it is recognition by the k-means classifier.

4.2.1 Whitening

The purpose of whitening is to reduce the correlation between image pixels [21]. ZCA whitening is a standardized method of variance, which can be seen as a reconstruction of the whitened data of the Principal Component Analysis (PCA) [22].

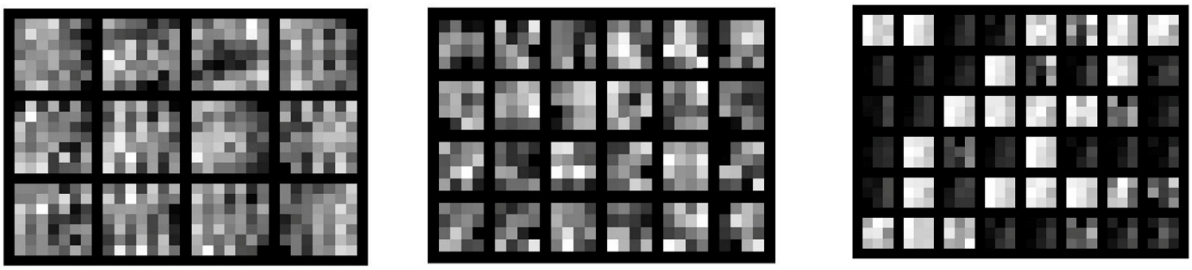
In order to obtain more local feature information of the image blocks, some image blocks are randomly selected from the training sample images for ZCA whitening. The simple steps are as follows: (1) each data subtracts the mean of the data set and gets the data after the average; (2) the covariance matrix of the data after the mean is calculated; (3) decompose the matrix by singular value to find all eigenvalues and corresponding eigenvectors; (4) PCA whitening of data sets; (5) PCA whitened data is left multiplied by the eigenvector matrix. Fig. 9 is part of the original image block diagram. Fig. 10 shows the ZCA whitening part of the image block, we can see it is more prominent than the original image block outline information.

Fig. 9. Part of the original image block diagram

Fig. 10. After ZCA whitening part of the image block

4.2.2 Sparse Auto-Encoder Experiment

The filter weight W of CNNs is trained unsupervised using SAE. Set the first layer weight of CNNs to 12*7*7, that is, 12 filters of 7*7 size. First cut 30,000 7*7patches in the whitened picture, and use these patches as the input of SAE. The weight W is 49*12. As shown in Fig. 11(a), the number of input layers is 49, the number of hidden layers is 12, and the dimension of the input layer is greater than the dimension of the hidden layer, and the KL sparse constraint is introduced to perform thinning; The second layer has a weight of 24*4*4, which is 24 filters of 4*4 size. The extracted feature map h1 is used as the input of the SAE, the number of hidden layers is 24, and the weight W obtained by the training is 24*16. As shown in Fig. 11(b), the input layer dimension is smaller than the hidden layer dimension, and the L1 norm is introduced. The number is thinned out; The third layer has a weight of 48*3*3, that is, 48 filters of 3*3 size. The extracted feature map h2 is used as the input of the SAE, and the number of hidden layers is set to 48, and the weight W is obtained through training. 48*9, as shown in Fig. 11(c), the input layer dimension is smaller than the hidden layer dimension, and the L1 norm is introduced for thinning.



(a) The weight W1 of the C1 layer is 12*49 (b) The weight W2 of the C3 layer is 24*16 (c) The weight W3 of the C5 layer is 48*9

Fig. 11.

4.2.3 K-means Experiment

Using the k-means operation method, a total of 14400 small patches of the last pooling layer are recognized and processed, and each group has 1000 image inputs. It can be seen from Fig. 12 that the number of centers set by humans in turn is 6, 10, 20, 30, 50, 100, 200, 300.The minimum recognition accuracy is 70.17% when 6 is taken, and the recognition accuracy increases with the increase of the number of clusters. The recognition accuracy is 97.44% when 100 cluster centers are reached, and then the recognition rate growth slows down. The recognition accuracy at 300 is already close to 100%. Since the number of clustering centers is too large, which affects the speed of operation, we select 100 as the best clustering center points. Fig. 13 shows the center point of the figure. The results of the clustering center are shown in Fig. 14.

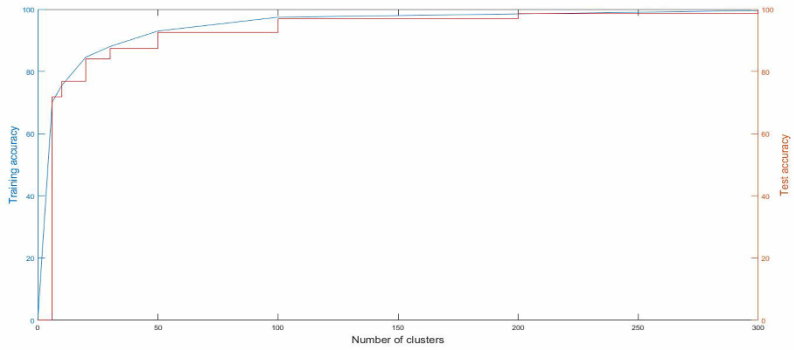


Fig. 12. The selection process of the optimal cluster center point

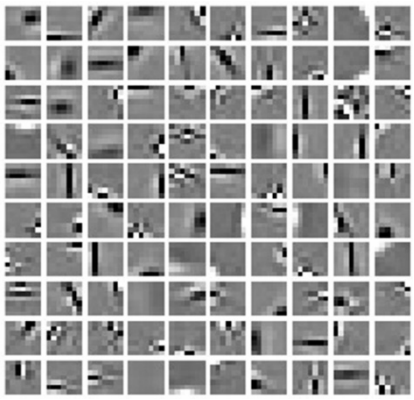


Fig. 13. Image formed by 100 cluster center points

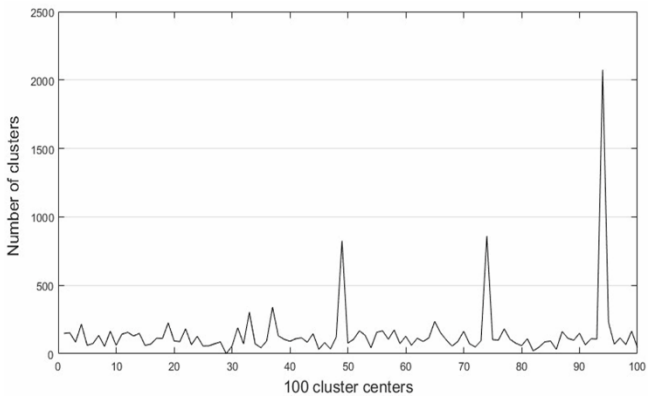


Fig. 14. The number of 14400 images corresponding to 100 cluster centers

4.3 Experimental Result

4.3.1 Glass Database

We proposed a glass defect recognition method on our own database and compared its performance with several recently proposed algorithms, including softmax, SVM, Random Forests and SVD. The parameters used by all algorithms in the experiment were all the same. The recognition results show that the recognition rate of each type of the algorithm in the recognition of defects is more than 90% except for the tumor defects. This may be due to the small amount of data of the tumor samples and the lack of covering all types of defects. But there is a significant improvement over other methods. The recognition results are shown in Table 1.

Table 1. Calculated results for inclusion, tumor, bubbles, scratch, stain and broken board

Classification algorithm	Inclusion/One	Tumor/One	Bubble/One	Scratch/ One	Stain/One	Broken board/ One
softmax	89	85	94	94	93	97
SVM	90	87	93	95	95	96
Random Forests	92	89	94	95	96	97
SVD	91	89	93	95	96	96
k-means	93	90	95	98	98	99

The experimental results of Table 2 are completed on the basis of Table 1 and the Kappa coefficient is added to illustrate. The experimental parameters are unchanged. It can be observed from the table that the algorithm in this paper increases the recognition time and accuracy of the SAE compared to the simple k-means, and reduces the amount of data processing during the running process. All accuracy and recognition time are averages taken after 10 training sessions.

Table 2. Classification algorithm comparison result

Classification algorithm	Softmax	SVM	Random Forests	SVD	k-means	k-means +SAE
Average accuracy /%	92.04	92.62	93.85	94.02	96.43	96.43
Kappa coefficient	0.8729	0.8872	0.9038	0.9256	0.9502	0.9502
Average time /s	432.1	397.8	341.0	322.7	289.3	255.9

4.3.2 AR Database

The AR database corresponds to a data set of more than 4,000 color images of 126 people (70 males, 56 females). The images have different facial expressions, and each person has 26 face images. We used 2,600 images of 50 men and 50 women as new data sets. We randomly selected 20 images of each person as training and the remaining 6 as tests. The algorithm parameters remain unchanged during the experiment. We project the face image $\in R^{192\times168}$ to a vector $\in R^{540}$ using Randomface [23]. The recognition results are summarized in Table 3. Our approach is superior to all competitive methods.

Table 3. AR database recognition result

Classification algorithm	Softmax	SVM	Random Forests	SVD	k-means +SAE
Average accuracy /%	88.04	88.62	89.85	90.43	92.92
Average time /s	556.6	503.1	463.0	441.4	357.7

4.3.3 Mnist Database

The Mnist database consists of 70,000 samples of 0-9 digits, with a training set of 60,000 samples and a test set of 10,000 samples, each with 7,000 images, each consisting of 28*28 pixels. Each pixel is represented by a gray value. We randomly selected 500 sheets of each number as training pictures and 100 sheets as test pictures. The experimental process parameters were unchanged, and all the accuracy and recognition time were average values taken after 10 training sessions. The results are shown in Table 4, the algorithm is optimal compared to other algorithms.

Table 4. Mnist database recognition result

Classification algorithm	Softmax	SVM	Random Forests	SVD	k-means +SAE
Average accuracy /%	93.7	92.6	94.3	94.8	96.2
Average time /s	643.7	593.6	558.2	487.9	438.5

5 Conclusion and Discussions

In traditional glass defect recognition methods, the selection and extraction of defect types have disadvantages like the effect of man-made subjective factors, low efficiency, large amounts of calculations and other deficiencies. However, there are many problems in the glass defect target, such as complex and changeable shape and too small target. It is difficult to express the defect target effectively by the method of human design feature extraction. Therefore, a defect recognition model based on sparse self-encoding and convolutional neural networks has been built. Combined with the advantages of multi-level and fine conversion of convolutional neural network parameters, this paper uses SAE pre-training convolution kernel to replace the random initialization of weights in traditional CNN. The parameters can be more suitable before optimization to improve the recognition time. And use k-means for recognition operations to improve recognition accuracy. Experimental results show that this method can accurately classify common glass defects. However, the performance in the face field is not very good. The next research focus is how to improve the recognition of various fields.

Acknowledgments

This work is supported by the Shanxi Scholarship Council of China, No.2016-084.

References

[1] Y. Jin, Z.B. Wang, L. Zhu, J. Yang, B. Wei, Study on Glass Defect Inspection Based on Projecting Grating Method, Journal of Testing and Evaluation 41(2)(2013) 332-339.

[2] H.G. Liu, Y.P. Chen, X.Q. Peng, J.M. Xie, A classification method of glass defect based on multiresolution and information fusion, International Journal of Advanced Manufacturing Technology 56(9-12)(2011) 1079-1090.

[3] K. Sachdeva, A. Girdhar, A technique for glass defect detection, International Journal of Innovative Research and Development (2)(2013) 25-31.

[4] S.Z. Wang, Z.B. Wang, Y. Jin, Y.X. Chen, Research of recognition technology for glass defect based on BP neural network, Modern Electronics Technique 14(2012) 45-48.

[5] S.H. Hanzaei, A. Afshar, F. Barazandeh, Automatic detection and classification of the ceramic tiles' surface defects, Pattern Recognition 66(2017) 174-189.

[6] S. Lawrence, C.L. Giles, A.C. Tsoi, A.D. Back, Face recognition: a convolutional neural-network approach, IEEE transactions on neural networks 8(1)(1977) 98-113.

[7] H.M. Moon, H.S. Chang, S.B. Pan, A face recognition system based on convolution neural network using multiple distance face, Soft Computing 21(17)(2017) 4995-5002.

[8] C.N. Anagnostopoulos, I.E. Anagnostopoulos, V. Loumos, E. Kayafas, A license plate-recognition algorithm for intelligent transportation system applications, IEEE Transaction on Intelligent Transportation Systems 7(3)(2006) 377-392.

[9] K. Li, B. Bare, B. Yan, An efficient deep convolutional neural networks model for compressed image deblocking, in: Proc. IEEE International Conference on Multimedia and Expo, 2017.

[10] Y.Y. Zhang, D. Zhao, J.D. Sun, G.F. Zou, W.T. Li, Adaptive convolutional neural network and its application in face recognition, Neural Processing Letters 43(2)(2016) 389-399.

[11] G.B. Praveen, A. Agrawal, P. Sundaram, S. Sanjay, Ischemic stroke lesion segmentation using stacked sparse autoencoder, Computers in Biology and Medicine 99(38-52) (2018).

[12] J. Bouvrie, Notes on convolutional neural networks. <http://cogprints.org/5869/index.html>, 2006.

[13] YW. Yu, GF. Yin, YD. Yin, Q. Liu, Radiographic image defect recognition method based on deep learning network, Chinese Journal of Scientific Instrument 35(9)(2014) 2012-2019.

[14] F. Zhang, J. Zhang, Softmax discriminant classifier, in: Proc. 3rd International Conference on Multimedia Information Networking and Security, 2011.

[15] D. Li, M.M. liu, S. Guo, Application of improved K_means algorithm in image segmentation, Computer Knowledge and Technology 12(8)(2016) 166-168.

[16] Y.G. Li, H. Wu, A clustering method based on K-means algorithm, Applied Mechanics and Materials 380-384(2013)1697-1700.

[17] J.J. Qing, X.F. Ling, R.F. Bie, X.Z. Gao, A new clustering algorithm based on artificial immune network and k-means method, Sixth International Conference on Natural Computation IEEE 6(2010) 2826-2830.

[18] R. Tibshirani, Regression shrinkage and selection via the lasso: a retrospective, Journal of the Royal Statistical Society 73(3)(2011) 273-282.

[19] Z. Jiang, Y. Wang, L. Davis, W. Andrews, V. Rozgic, Learning discriminative features via label consistent neural network, in: Proc. 2017 IEEE Winter Conference on Applications of Computer Vision, 2017.

[20] R. Andonie, Extreme data mining: Inference from small datasets, International Journal Of Computers, Communications and Control 5(3)(2010) 280-291.

[21] A. Coates, A.Y. Ng, Selecting receptive fields in deep networks, in: Proc. International Conference on Neural Information Processing Systems, 2011.

[22] Y. Dong, S.J. Qin, A novel dynamic PCA algorithm for dynamic data modeling and process monitoring, Journal of Process Control 67(2016)1-11.

[23] J. Wright, A.Y. Yang, A. Ganesh, S.S. Sastry, Y. Ma, Robust face recognition via sparse representation, in: Proc. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2009.