IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON GIGAPIXEL PANORAMIC VIDEO WITH VIRTUAL REALITY

Received May 26, 2020, accepted June 10, 2020, date of publication June 17, 2020, date of current version June 30, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3003059

Image Target Recognition Model of Multi-Channel Structure Convolutional Neural Network Training Automatic Encoder

SEN ZHANG[1], QIUYUN CHENG[1], DENGXI CHEN[1], AND HAIJUN ZHANG[2]
[1]School of Intelligent Engineering, Zhengzhou University of Aeronautics, Zhengzhou 450046, China
[2]School of Aeronautical Engineering, Zhengzhou University of Aeronautics, Zhengzhou 450046, China

Corresponding author: Qiuyun Cheng (chengqiuyun@zua.edu.cn)

This work was supported by the National Natural Science Foundation of China, through Balancing Dynamics and Optimization of Mixed-Model Assembly Line Network for Complex Products Based on Digital Twin, under Grant 51705472.

ABSTRACT The self-encoder is a typical unsupervised deep learning algorithm. In the field of unsupervised learning, it is very popular with researchers. Therefore, in view of the shortage of labeled training samples, the convolution kernel of a typical convolutional neural network is set by experience, and the network structure is fixed and it is difficult to re-learn later. This paper combines the convolutional neural network and the automatic encoder, and proposes a multi-based the method of integrated network structure to extract the features of the image for recognition. First, the SAE pre-trained CNN model convolution kernel is used to pre-train based on the classic CNN structure. Secondly, input and process image data of different scales to extract image space and spectral features respectively. Then, construct multiple channels, and use different scale filters and sampling intervals for different channels. Finally, after one layer of down sampling, the feature maps obtained from multiple channels are input into the fully connected layer, and after a hidden layer, the features finally used for classification are obtained. Experimental results show that the proposed method uses sparse automatic coding for pre-training time efficiency increased by 50%, and can further improve the recognition accuracy, the highest recognition rate reached 0.985.

INDEX TERMS Convolutional neural network, multi-channel, training, automatic encoder, recognition.

I. INTRODUCTION

Deep learning is a brand new branch of machine learning and a powerful core driver in the field of artificial intelligence [1], [2]. In recent years, it has swept almost all fields of research and industry and developed rapidly. In simple terms, deep learning can be understood as a neural network structure with multiple hidden layers. Compared with traditional neural networks, the biggest feature of deep learning is unsupervised feature learning, that is, it can automatically extract target features [3], [4], avoiding the various drawbacks caused by manual design feature selection. Deep learning algorithms include unsupervised learning and supervised learning algorithms, among which self-encoder is a typical unsupervised deep learning algorithm, in the field of unsupervised learning, it is very popular among researchers [5], [6]. Convolutional neural network is a supervised deep learning algorithm, which has achieved great success in speech recognition, natural image processing, and other fields [7], [8].

As neural networks become more complex, the number of network layers expands, gradient elimination occurs, and it is difficult to converge. The results obtained are related to initialization at a large level. The mainstream in image processing is the convolutional neural network. Convolutional neural networks are specifically proposed for image processing. Peng and Song [9] pointed out that in speech or object recognition, extracting multi-layer feature representations of objects can often achieve very good results. However, deep neural networks combined with gradient descent are unsatisfactory in tasks with few labels. In the field of image processing and computer vision, the most successful research and application is CNN. Wang *et al.* [10] studied the use of weakly labeled images and unlabeled images for multi-label image annotation. The author proposes to first train CNN with

The associate editor coordinating the review of this manuscript and approving it for publication was Zhihan Lv.

weakly labeled or unlabeled samples, and then fine-tunes the CNN with high-quality images collected. Experiments show that the use of weakly labeled or unlabeled images combined with deep learning works well. Wang *et al.* [11] pointed out that handwriting image recognition is very sensitive to structural noise. Due to the non-locality of structural noise and the indistinguishability of true and false regions, the author proposes to use a denoising auto-encoder to construct a deep neural network. Some good results have been achieved on the set. Herbel *et al.* [12] pointed out the shortcomings of traditional image aesthetic evaluation methods, and proposed to use deep learning methods to evaluate images. In this paper, a two-column CNN is proposed for feature extraction and classifier training, combined with image style, and semantic attributes. Experiments show that this method is better than the previous method. Son *et al.* [13] borrowed from the human visual cortex and intelligent perception to propose a semi-conducting two-line deep belief network (SBDBN). By comparing with the incomplete image recognition technology and the existing deep learning model, it is applied to two standard data sets and one artificial data set at the same time. The experimental results show that the model shows excellent recognition ability. Zhao and Du [14] proposed a deep learning method based on feature contours applied to hyperspectral image classification. The results show that the classification effect of the algorithm is relatively good. Pan *et al.* [15] studied image classification technology and proposed a deep learning framework that can automatically discover the underlying geometry without prior assumptions. First, the parameters are initialized using a layer-by-layer unsupervised preprocessing method based on Gaussian restricted Boltzmann machine, and then the depth of each class and specific class of the image is trained separately to construct the model. Zhong *et al.* [16] proposed an image recognition method for incomplete data (FEBDN) based on field-effect bilinear deep network to solve the problem of image recognition for incomplete data. Zhang *et al.* [17] proposed classification algorithm of a bilinear deep learning image (BDBN). BDBN aims to provide human-like judgments by drawing on the architecture of the human visual system and intelligently perceptive programs. Wu [18] investigated the semantic gap in the content-based image retrieval system. By examining a state-of-the-art deep learning method for CBIR tasks in different environments, the experiment found some encouraging results and summarized some important insights. Ye *et al.* [19] proposed a natural image recognition algorithm based on a large deep CNN, which achieved a high recognition rate on the ImageNet dataset. Garea *et al.* [20] proposed a CNN based on multi-core, and achieved a good recognition effect on the three-dimensional data set by using the GPU parallel operation method. Qayyum *et al.* [21] used sparse coding to extract the basis function of the training image as the initial filter of CNN. Wu *et al.* [22] applied independent component analysis (ICA) to the pre-training stage of CNN, and used ICA to train the filter set to improve the recognition rate. Daniel *et al.* [23] proved that the filter size has a great influence on the final recognition result, and gave a relatively optimal filter size under single-layer conditions. Saha *et al.* [24] proved that when the sampling interval is small, even after 2 convolutions and 2 maximum down sampling, the activation value of the network output can still reconstruct the seemingly the same pattern as the original input.

Auto-encoder is also a deep learning method, and there are many researches and applications on it, such as sparse auto-encoder, stacked auto-encoder, and so on [25]–[29]. Golkov *et al.* [30] pointed out that although deep learning technology has achieved remarkable results in many fields. However, there are still three problems in training speed, local minimum, and hyper-parameter selection. In order to overcome these three problems, the author proposes to use auto-encoder to assist in training deep networks. Silva *et al.* [31] proposed a spatially updated deep auto-encoder (SDAE) in order to extract and use the features of high-spectrum images. By adding regularization terms to the energy function to consider sample similarity, and by integrating context information to update features, good results were obtained. Eqlimi *et al.* [32] came up with a K-sparse classification system based on auto-encoder in order to improve these problems. Experiments have found that this classification system has been greatly improved in accuracy and complexity. Jia *et al.* [33] proposed the use of deep auto-encoder for compact shape feature expression and image retrieval, and achieved good results. Riaz *et al.* [34] compared and analyzed the performance of auto-encoder and principal component analysis in face recognition applications. Both methods are used for feature generation and selection, and it is found that auto-encoder is superior to principal component analysis.

Although the deformation algorithms of CNNs have achieved great success in image processing, when used for image recognition, it still has hidden problems such as local optimization, gradient elimination, and long training time. Using auto-encoder to initialize the parameters of each layer of CNN can not only eliminate the local optimal uncertainty caused by random initialization, obtain a good initial value, but also suppress the problem of gradient elimination. Furthermore, when the amount of data gradually increases and the network structure becomes gradually more complicated, this initialization can reduce the training time and achieve better image recognition results in a shorter time. Therefore, this article combines the auto-encoder with the convolutional neural network to solve the problems of the disappearance of the convolutional neural network gradient and the long training time. In order to use the convolutional neural network for image recognition, it takes less time to get better result.

Specifically, the technical contributions of our paper can be concluded as follows:

In this paper, the combination of the automatic encoder and the convolutional neural network can solve the problems such as the disappearance of the convolutional neural network gradient and the long training time. When the convolutional

neural network is used for image recognition, it takes less time to get better results.

The rest of our paper was organized as follows. Related work was introduced in Section II. Section III described the structure of the convolutional neural network algorithm proposed in this paper. Experimental results and analysis were discussed in detail in Section IV. Finally, Section V concluded the whole paper.

II. RELATED THEORETICAL KNOWLEDGE

A. BASIC CHARACTERISTICS OF CONVOLUTIONAL NEURAL NETWORKS

Convolutional neural networks (CNNs) are a multi-layer feedforward neural network for computer vision applications published in 1989 [35]. Currently, CNNs are becoming increasingly popular in deep learning methods. Because it can better learn the model of many computer vision tasks such as target detection, target recognition, semantic image segmentation. Compared with traditional neural networks, the main features of convolutional neural networks are local perception and weight sharing, also known as sparse connections, which can greatly reduce the number of parameters that the network model needs to train [36].

1) LOCAL PERCEPTION

Modern biological research believes that the brain's cognition of the world through vision is from local to global [37], which means that neurons of animal vision only have a part of neurons functioning when they perceive external objects. In computer vision, in a certain area of the image, the correlation between the pixels and the distance between the pixels are also related, the correlation between the pixels closer is stronger, and the correlation between the distances is weaker. From which we can see that the theory of local correlation is also applicable to the field of image processing. The neurons of the convolutional neural network are only connected to some neurons in the previous layer to sense local information. The neurons in the lower layer perceive local regions of the image, and the neurons in the upper layer synthesize the local features in the lower layer to obtain abstract global features.

In traditional neural networks, hidden units, also called neurons, consist of one-dimensional vectors. However, due to the characteristics of the image, the hidden units in CNNs are usually composed of two-dimensional planes, which we call feature maps [38]. According to the dimension of the layer, each convolutional layer has a certain number of feature maps. Use a linear filter to perform convolution operation on the input image or the previously obtained features, and then calculate through a nonlinear function to obtain the feature map output by this layer. In other words, each cell in the feature map receives input from the p × p region of the image or the previous feature map. These p × p regions are called the receptive field of this unit. The interval between the receptive fields of adjacent units is called the step size.

2) WEIGHT SHARING

All units in the feature map share the same weight, and the same convolution filter is used to apply to all receptive fields of the previous feature map. We call these shared weights filters or convolution kernels.

Since these basic features can appear in any area of the image, displacement invariance is very important for capturing these basic features. Moreover, through weight sharing, displacement invariance can be achieved. Weight sharing has another advantage, that is, it can greatly reduce the number of weight parameters that need to be trained.

In LeNet [39], the convolutional layer C3 has 16 feature maps, and each unit in each feature map is connected to a 5 × 5 neighborhood in each or part of the feature maps in the previous layer S2. Therefore, the number of weight parameters is shown in formula (1).

$$(\text{kernel size}+\text{bias}) \times |S2| \times |C3| = (5 \times 5 + 1) \times 6 \times 16, \tag{1}$$

B. BASIC STRUCTURE OF CONVOLUTIONAL NEURAL NETWORK

Convolutional neural network is a deep feed-forward artificial neural network and one of the representative algorithms of deep learning [40]. It can automatically learn multi-layer features directly from images and has very good representation capabilities. It is widely used in computer vision fields such as image classification, target detection, and semantic segmentation. The structure of a convolutional neural network usually includes modules such as an input layer, a convolution layer, an activation layer, a pooling layer, a fully connected layer, and an output layer. Its basic structure is shown in Figure 1.

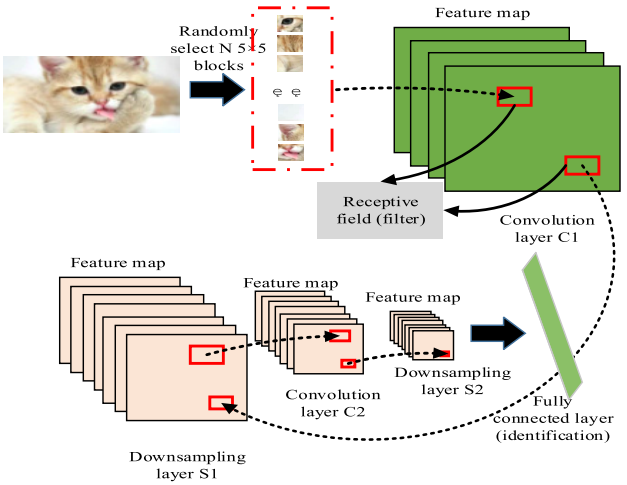

FIGURE 1. The basic structure of a convolutional neural network.

1) CONVOLUTIONAL LAYER

As shown in Figure 2, each convolutional layer of CNN consists of a 3D filter with a size of d × h × w, where h × w is the spatial dimension, which is equivalent to the set of

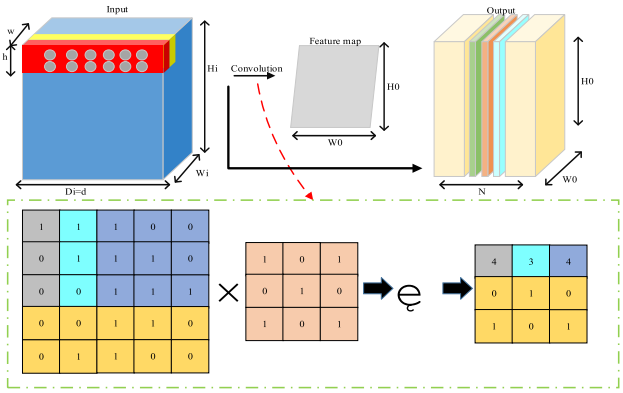

FIGURE 2. Operation of the convolutional layer.

neurons, and d is the number of nuclear feature channels. Each filter is convolved with the corresponding part of the input image, sliding across the entire image. Convolution refers to summing the neuron in each filter and the value corresponding to the input layer item by item. Therefore, if the input layer in CNNs is an image, the image can be convolved with each filter of each layer to obtain a two-dimensional output. The size of the output is affected by the step size and the filling parameters. The output is represented as a feature map or activation map. In each convolutional layer of CNNs, N filters are used, and each filter generates a feature map. Stack these feature maps together to get the output of the convolutional layer.

A single neuron in a filter can be mapped to the neurons connected in all previous layers. This is called the effective receptive field of neurons [41]. It is easy to see that the convolution results in a local connection between the neurons in the lower layer and the neurons in the smaller receptive field compared to the connections between the neurons in the upper layer and the neurons in the smaller receptive field closer. Lower layers learn to characterize small areas of input, while higher layers learn more specific semantics because they respond to larger subdivisions of the input image. Therefore, a feature hierarchy is generated from local to global.

The red and yellow regions in Figure 2 represent the two positions where the filter of size d × h × w is convolved by sliding the input space. The step size of the filter is defined as the interval at which the filter moves in each spatial dimension. The fill parameter p corresponds to the number of pixels added to the outer edge of the input. Therefore, the step size can be regarded as the input method of sub-sampling. A square filter in the form of h = w = f is usually used, and the output of this layer is calculated using equation (2), equation (3) and equation (4).

$$D_o = N \tag{2}$$

$$H_o = \frac{H_i - f + 2p}{s} + 1 \tag{3}$$

$$W_o = \frac{W_i - f + 2p}{s} + 1 \tag{4}$$

Figure 2 shows a 3 × 3 filter sliding over a binary image matrix of size 5 × 5. The filter slides from left to right to the end of the matrix, and its step size is one. Slide the filter in order to perform convolution to obtain the output feature map.

2) POOLING LAYER

Pooling is achieved by sliding a filter on the input image. The input image is divided into different sub-regions, and each sub-region is down-sampled by a nonlinear pooling function. The most commonly used of these functions are the maximum and average pooling functions. In addition, pooling removes unnecessary and redundant features, reduces network-computing costs, and improves network efficiency.

The pooling layer also has a stride parameter to control the output size. In Figure 3, the input features are down-sampled from 64 × 224 × 224 to 64 × 112 × 112 by the pooling kernel with a step size of two and 2 × 2. Implement pooling operations on each feature map to reduce the size of the feature map.

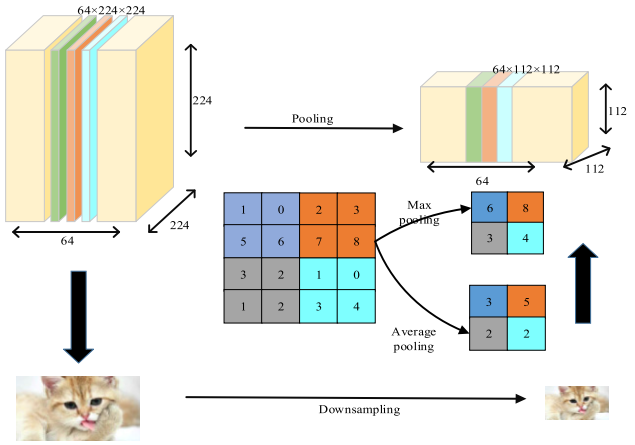

FIGURE 3. Pooling operation.

3) ACTIVE LAYER

In neural networks, three activation functions are commonly used for simulating cellular neuron activation, namely sigmoid function, tanh function, and relu function. The first two functions were widely used in the rise of neural networks, but due to their inherent defects, they were gradually replaced by the relu function.

The function sigmoid is an s-type function, see formula (5). The activation domain of the function sigmoid is between zero and one, and most of the time its output value is close to zero or one, as shown in Figure 4(a). It can be seen from Figure 4(a) that the sigmoid function is a non-decreasing function. When the independent variable is less than -4, the dependent variable is close to zero. When the independent variable is greater than four are, the dependent variable is close to one. The derivative of the function sigmoid is shown in formula (6), and the curve of the derivative function is shown in Figure 4(b). It can be seen that the function is less than or equal to 0.5 and greater than zero, which is a

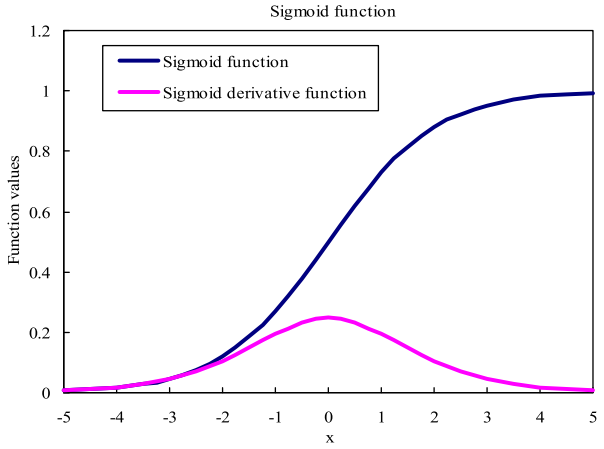

FIGURE 4. Function sigmoid curve and derivative curve.

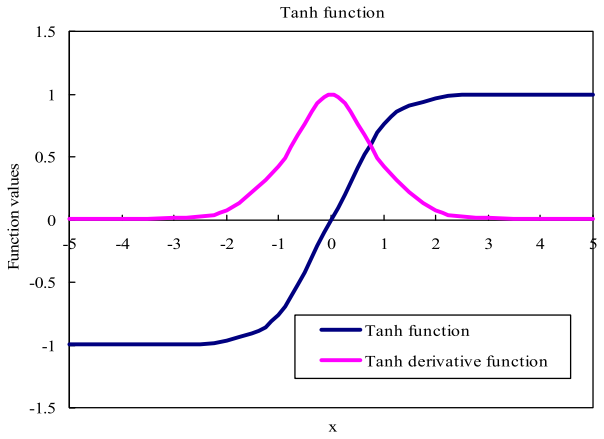

FIGURE 5. Function tanh curve and derivative curve.

symmetric function.

$$sigmoid(x) = \frac{1}{1+e^{-x}} \tag{5}$$

$$sigmoid'(x) = \frac{1}{1+e^{-x}} \times (1 - \frac{1}{1+e^{-x}}) \tag{6}$$

The expression of the function tanh is shown in formula (7), as shown in Figure 5(a). Compared with the sigmoid function, the activation range of the tanh function is wider, which is equivalent to the enlarged version of sigmoid. The derivative of the function tanh is shown in formula (8), and the derivative curve is shown in Figure 5(b). It can be found from the formula that its derivative is in the range of 0 to 1. However, the tanh function also has the problem of disappearing gradients, resulting in low training efficiency. Because the value of its derivative is between zero and one, when the value of the derivative is one, the function activation value is zero, which also causes the gradient to not be downloaded.

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{7}$$

$$\tanh'(x) = 1 - (\frac{e^{x} - e^{-x}}{e^{x} + e^{-x}})^2 \tag{8}$$

The function relu is a simple and efficient activation function. Compared with the first two functions, the relu function has a wider activation domain, has more sparse activations, and the relu function is unilaterally suppressed. When the relu function is not zero, its derivative is one. In theory, the gradient disappearance is much better than sigmoid and tanh. In actual training, relu also converges faster and is more computationally efficient than the other two functions. The disadvantage of the function relu is that some neurons may be necrotic during training, which may be consistent with the characteristics of nerve cells.

$$relu(x) = \max(0, \max) \tag{9}$$

4) FULLY CONNECTED LAYER

After extracting high-level features through the convolutional layer, pooling layer, and relu layer, the fully connected layer is usually placed at the end of the network. Neurons in this layer are completely dependent on all activations in the previous layer. The most important role of the fully connected layer is that the neurons in this layer determine which features correspond to which categories. In short, the fully connected layer can be seen as the layer that provides the classifier.

5) CLASSIFIER

The choice of classifier is determined by considering the current problem and the data used. The softmax function in equation (10) gives the probability that an input belongs to class c.

$$P_c = \frac{e^{(S_c)}}{\sum_{i=1}^{c} e^{(S_i)}} \tag{10}$$

In the formula, s is a specific type of network output. For a single input, the sum of all probabilities between classes is always equal to one. The loss function is defined as the negative logarithm of the probability of the softmax function.

C. AUTOMATIC ENCODER

The concept of auto-encoder was proposed earlier, and was originally applied to high-dimensional complex data processing, which promoted the development of neural networks [42]–[44]. The self-encoder is an unsupervised learning algorithm in the deep learning algorithm, to be more precise, a self-supervised learning algorithm whose label data is derived from the input samples. The principle of the self-encoder is not complicated. It can be understood as a system that attempts to restore the original input, so the self-encoder is often used to automatically learn features and data compression [45].

1) STACKED SELF-ENCODER

The general structure of the auto-encoder is a three-layer neural network model, including an input layer, a hidden layer, and an output layer. Stacked self-encoder is a variant of self-encoder. It is a neural network model formed by stacking

multiple common self-encoders. The stacking method is that the hidden layer output of the previous auto-encoder is used as the input layer of the latter auto-encoder, that is, layer-by-layer greedy training is implemented. This method solves the problem of gradient disappearance that is difficult to overcome by deep neural networks, and layer-by-layer training. The method can extract deeper features and help solve more problems that are complex. It is often used for unsupervised feature extraction and weight pre-training for other networks. The principle diagram is shown in Figure 6.

Stacked auto-encoder contains two important concepts.

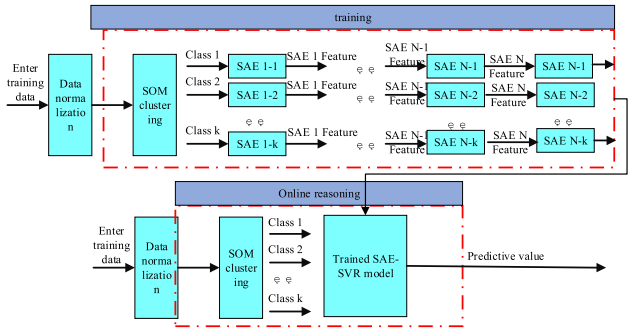

FIGURE 6. Schematic diagram of the stack self-encoder.

a: SPARSENESS RESTRICTION

The idea of the auto-encoder is to recover the original input even when the hidden layer is forced to be different from the input. There are two cases at this time, the number of hidden layer nodes is small, and the number of hidden layer nodes is large. When the number of hidden layer nodes is small, the original data can be compressed to extract effective features that can express the original data. When the number of hidden layer nodes is large, features can still be extracted with the auto-encoder. At this point, the sparsity limit is added.

The main idea of sparseness restriction is to put some constraints on the hidden layer to make it sparse, that is, not all nodes are active. When the output of a neuron node in the hidden layer is close to one, the neuron is considered to be activated. When the output of a neuron node is close to zero, the neuron is considered to be suppressed. The characteristic that suppresses neurons is called sparsity limitation. The specific implementation is shown in the following formula:

$$\hat{p}_j = \frac{1}{m}\sum_{i=1}^{m} \alpha_j^{(2)} x^{(i)} \tag{11}$$

Among them, the variable $\alpha_j^{(2)} x$ represents the activation degree of the hidden layer neuron j when the input is x. The variable $\hat{p}_j$ represents the average activation degree of j over the entire training set. The way to add sparseness restriction is to let the equation $\hat{p}_j = p$ hold. The variable p is the sparsity parameter. In order to make the average activity of neuron j close to p, this sparse penalty factor needs to be added to the optimization function. Commonly used sparse penalty factors are as follows:

$$\sum_{j=1}^{s_2} p \log \frac{p}{\hat{p}_j} + (1-p)\frac{1-p}{1-\hat{p}_j} \tag{12}$$

b: LAYER-BY-LAYER GREEDY TRAINING

Deep learning has been in a trough for a long time, and everyone has gradually forgotten it. The reason is mainly that scholars cannot overcome the phenomenon that the gradient of the deep neural network model disappears due to the deeper layers. Until 2006, Geoffrey Hinton proposed that the layer-by-layer greedy training method could be used to solve the problem of gradient dispersion and local minimum in deep neural networks, making deep learning return to the public's vision again [46]. The biggest difference between layer-by-layer greedy training and traditional training methods is that each training will not pass forward multiple layers and then propagate multiple layers forward from the last layer. Instead, it only trains a network with one hidden layer at a time. When the network training reaches the local optimum, the next network with the output of the hidden layer of the previous network as the input is trained, and the local optimum is reached again. Finally, the labeled data is used to fine-tune the weights trained layer by layer for classification or other tasks.

2) CONVOLUTIONAL AUTO-ENCODER

The traditional self-encoder fully connected method ignores the two-dimensional structure of the image, which not only forces the network to learn global features, but also easily causes parameter redundancy and affects training efficiency. Convolutional auto-encoder introduce convolution and pooling operations on the basis of traditional auto-encoder, realize local receptive fields and weight sharing, and are often used to process two-dimensional images.

The architecture of the convolutional auto-encoder is similar to that of the auto-encoder, and its mathematical calculation process is as follows [47]:

For input single-channel input x, the formula for extracting feature maps through k convolution kernels is as follows:

$$h^k = \sigma(x * W^k + b^k) \tag{13}$$

Among them, the variable σ represents the activation function; operation $*$ represents the convolution operation. The reconstruction process uses the following formula:

$$y = \sigma(\sum_{k \in H} \tilde{W} * h^k + c) \tag{14}$$

Among them, the variable σ represents the activation function, and H represents the number of feature map groups. The variable $\tilde{W}$ represents the transposition of the weight of the encoding part. The variable c represents the offset. The loss function generally uses the mean square error, and the formula is as follows:

$$E(\theta) = \frac{1}{2n}\sum_{i=1}^{n} (x_i - y_i)^2 \tag{15}$$

Like other neural networks, the back propagation algorithm is used to calculate the gradient of the error function relative to the parameters. This can be easily obtained using convolution operation, the formula is as follows:

$$\frac{\partial E(\theta)}{\partial W^k} = x * \delta h^k + \tilde{h}^k * \delta y \tag{16}$$

Among them, the variable δh^k and the variable δy represent the features extracted from the hidden layer and the reconstruction results, respectively.

III. MULTI-CHANNEL STRUCTURE AUTOMATIC CODING NETWORK

A. SPARSE AUTO-ENCODER

Sparse auto-encoder (SAE) is an unsupervised learning model, which re-encodes input data by making the output value equal to the input value as much as possible to learn the characteristics of the data.

Since the auto encoder is an unsupervised learning algorithm in the deep learning algorithm, its label data is derived from the input samples. Auto encoders are often used for automatic feature learning and data compression. Therefore, this paper uses a sparse automatic encoder to train the convolutional network, which can complete the re-encoding of the input data, so as to learn the characteristics of the data.

Suppose there are n images in the training sample, which are divided into k categories. Each training image is transformed into a column vector, and the corresponding label constitutes the sample set $\{(s_i, t_i), i = 1, 2, \ldots, n\}$, then sparse the output of the hidden layer of the auto-encoder can be obtained by combining the column vectors and corresponding weights and adding an offset term through a nonlinear function. This process is called forward propagation, as shown in equation (17).

$$a_i^{(l+1)} = f(z_i^{(l+1)}) = f(\sum_{j=1}^{m} W_{ij}^{(l)} * s_i + b_j^{(l+1)}) \tag{17}$$

Among them, the variable $a_i^{(l+1)}$ is the output value of i-th unit of the (l+1) layer. The variable $z_i^{(l+1)}$ is the input weighted sum of the unit i of (l+1) layer. The variable $W_{ij}^{(l)}$ is the weight between the j-th unit in the lth layer and i-th unit in the (l+1) layer. The variable $b_j^{(l+1)}$ is the offset of i-th unit of (l+1) layer. The variable m is the dimension of the lth layer. The variable f function is the activation function, and generally takes the sigmoid function or the hyperbolic tangent function.

Let $a(1) = S$ denote the activation value of the input layer, and express the f function in the form of a vector. Equation (17) can be simplified to the form of a vector. The meaning of the symbol remains unchanged:

$$z^{(l+1)} = W^{(l)} a^{(l)} + b^{(l+1)} \tag{18}$$

$$a^{(l+1)} = f(z^{(l+1)}) \tag{19}$$

In order to make the output value equal to the input value as much as possible, it is necessary to optimize the parameters of the weight and offset in equation (17). The method is to minimize the objective function like this:

$$J(W, b) = \frac{1}{n} \sum_{i=1}^{n} (\frac{1}{2} ||a^{(l)}(s_i) - s_i||^2) \tag{20}$$

The objective function is a variance cost function, which is optimized by the gradient descent method. However, when the input data with large data volume and high dimension is encountered in the calculation process, the objective function often converges slowly and the calculation complexity is too high. One solution is to add sparse constraints to the function, which constitutes a sparse auto-encoder. The objective function at this time is:

$$J'(W, b) = \frac{1}{n} \sum_{i=1}^{n} (\frac{1}{2} ||a^{(l)}(s_i) - s_i||^2) + \beta \sum_{j=1}^{s2} KL(\rho || \rho_j) \tag{21}$$

After adding the sparse limit, the average value of the hidden layer node output is close to zero, so that most of the hidden layer nodes are in an inactive state, which increases the sparseness of the model. After obtaining the objective function, the model parameters are updated according to the following formula:

$$W^{(l)} = W^{(l)} - \alpha \frac{\partial J'(W, b)}{\partial W^{(l)}} \tag{22}$$

$$b^{(l)} = b^{(l)} - \alpha \frac{\partial J'(W, b)}{\partial b^{(l)}} \tag{23}$$

where α is the learning rate. The back propagation algorithm is used to calculate the last two derivative terms of formula (22) and formula (23), and iterative updating is continued until the entire coding network is trained after the parameters converge, and the characteristic parameters W and b are obtained. Each parameter set is a filter, so that a pre-trained filter set is obtained.

B. SAE PRE-TRAINING CONVOLUTION KERNEL

This paper uses SAE pre-trained CNN model convolution kernel to pre-train based on classic CNN structure. The classic CNN network consists of six feature maps, corresponding to six convolution kernels of size 5×5. Therefore, in the sample image, 5×5 small blocks are randomly selected, and these small blocks are used as the input of SAE. The number of hidden layer neurons is set to six, the weight value W obtained by SAE training is 6×25, and W is converted into $6 \times 5 \times 5$, which is used as the initial value of the convolution kernel after C1 layer pre-training. After initializing the convolution kernel of the C1 layer, the remaining weights are still randomly initialized in the original way. The training network stops training after the specified number of iterations, and saves the feature output of the S2 layer of the input sample. In the same way, the S2 layer feature output saved by SAE training is used to obtain the initial value of the convolution kernel after the C3 layer pre-training. The classic structure is shown in Figure 7.

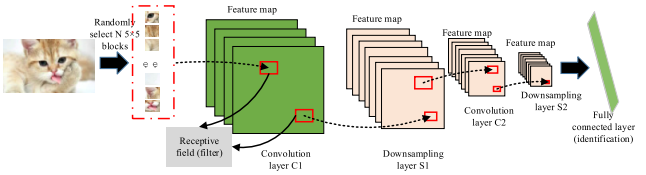

FIGURE 7. Model structure with SAE pre-training.

C. MULTI-CHANNEL STRUCTURE

In this paper, by constructing multiple channels on CNN, different channels use different scale filters and sampling intervals. Specifically, suppose the input image is X, first take three different scale image patches of X {1,2,3} to construct three channels, and select the corresponding filter size patch-Dim {1,2, 3} and down-sampling interval pool-Dim {1, 2, 3}. The selection criterion is to make the output feature dimensions of different channels as same as possible. The network structure is shown in Figure 8.

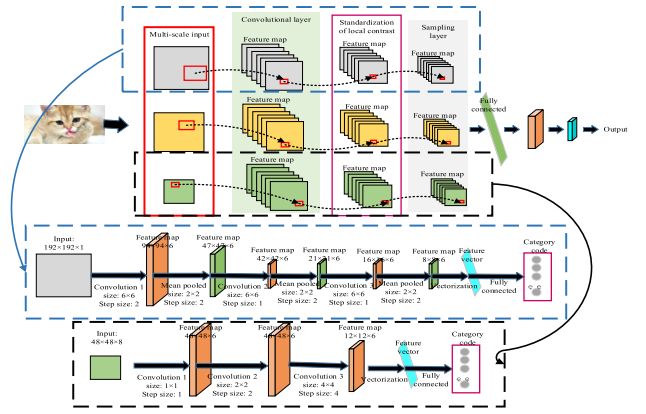

FIGURE 8. Convolutional neural network structure with multi-pass structure.

Although the traditional CNN is robust to translation and scaling, it has certain robustness. However, the literature [48] proved that small changes in the size of the input image would still result in different final recognition results of CNN. In order to enhance the robustness of the model, the network structure proposed in this paper adopts the method of multi-scale input, changing the direct input of the original image into the input image of different size image blocks. Three channels of three different sizes are used to form three channels. The three channels are performed separately during the convolution and down sampling operations, and the three channels are combined at the fully connected layer. While using multi-scale input, the input images of different channels are convolved with filters of different sizes. The features obtained after convolution of large-size filters have characteristics that are more global. The small size obtains features that more reflect local characteristics. The specific steps of convolution are the same as CNN. After the original input image is convolved with the filter, an activation function is used to obtain the first layer output feature map:

$$x_j^{(l+1)} = f\left(\sum_{i \in M_j} x_j^{(l)} * k_{ij}^{(l+1)} + b_j^{(l+1)}\right) \tag{24}$$

Among them, the superscript l indicates the number of layers. Operation "$*$" is a convolution operation. The variable $x_j^{(l+1)}$ represents the output of the j-th neuron after convolution. The variable $x_j^{(l)}$ represents i-th neuron in the lth layer, that is, the input data. The variable $k_{ij}^{(l+1)}$ represents the filter. The variable $b_j^{(l+1)}$ represents the offset. The variable M_j represents the set of selected input feature maps. Another difference between the network structure proposed in this paper and the traditional CNN is the addition of Local Contrast Normalization (LCN) operation, which has been shown to effectively improve the invariance of features and increase the sparsity of the model [49]. After applying the local contrast normalization to the convolutional layer in this paper, the specific normalization formula is:

$$x^{(l)\prime}{}_{u,v} = \frac{x_{u,v}^{(l)} - m_{N(u,v)}^{(l)}}{\sigma_{N(u,v)}^{(l)}} \tag{25}$$

Among them, the variable $x_{u,v}^{(l)}$ represents the output value of the corresponding position (u, v) of the layer 1 feature map. The variable $m_{N(u,v)}^{(l)}$ and variable $\sigma_{N(u,v)}^{(l)}$ represent the mean and variance of the local neighborhood $N\ (u, v)$, respectively. The features normalized by the local contrast are input to the down sampling layer. In order to ensure a high recognition rate while maintaining invariance, this paper adjusts the sampling interval of the down sampling according to the dimension of the output feature. The larger the sampling interval of the down sampling layer, the more fuzzy the output feature map and the stronger the feature invariance. Finally, the output dimensions of the three channels are the same.

For the large-scale convolutional neural network, the input is a full-color image with a size of 192 × 192 × 1. The designed large-scale convolutional neural network structure includes three convolutional layers, three pooling layers, and a fully connected layer. The first layer designs a convolutional layer with a size of 6 × 6 and a convolutional step size of two. In the second layer, a mean pooling layer with a template size of 2 × 2 and a step size of two is designed. The third layer designs a convolutional layer with a convolution kernel size of 6 × 6 and a step size of one. In the fourth layer, a mean pooling layer with a template size of 2 × 2 and a step size of two is designed. The fifth layer designs a convolutional layer with a convolution kernel size of 6 × 6 and a step size of one. In the sixth layer, a mean pooling layer with a template size of 2 × 2 and a step size of two is designed. The sixth layer features and the final output are fully connected. The fully connected layer contains 384 neurons.

For the small-scale convolutional neural network, in order to ensure that the input data of the large-scale network covers the same area, the input is a multi-spectral image with a size of 48 × 48 × 8. The designed small-scale convolutional neural

network structure consists of three convolutional layers and one fully connected layer. The first and second layers have a design step size of one, and a convolutional layer with a size of 1×1 for the extraction of spectral information features. The third layer uses a convolution kernel with a step size of four and a convolution kernel size of 4×4, which is mainly used for feature enhancement and dimension reduction. The third layer features and the final output are fully connected. The fully connected layer contains 864 neuron nodes.

After sampling under one layer, the network structure model constructed in this paper inputs all the feature graphs into the full connection layer. After passing through the hidden layer, the features used for classification are finally obtained. Enter this feature directly into softmax classifier for target classification identification.

In this paper, first, the SAE pre-trained CNN model convolution kernel is used to pre-train on the basis of the classic CNN structure, and the loss function at this time is recorded as loss1. Secondly, input and process image data of different scales to extract image space and spectral features respectively. Then, construct a convolution recognition network of multiple paths, and the loss function at this time is denoted as loss2. Therefore, the final loss function of this paper is the sum of loss1 and loss2. By optimizing the final loss function, the optimal parameters are obtained.

D. NETWORK STRUCTURE AND ALGORITHM FLOW

This paper proposes a method based on multi-channel integrated network structure to extract spatial features and spectral features of images and classify them. First, the SAE pre-trained CNN model convolution kernel is used to pre-train on the basis of the classic CNN structure. Secondly, input and process image data of different scales to extract image space and spectral features respectively. Then, construct multiple channels, and use different scale filters and sampling intervals for different channels. Finally, after one layer of down-sampling, the feature maps obtained from multiple channels are input into the fully connected layer, and after a hidden layer, the features finally used for classification are obtained.

The algorithm steps are as follows:

(1) Input

Image training set and test set with target, filter size patch-Dim{1, 2, 3}. Down-sampling interval pool-Dim{1, 2, 3}.

(2) Pre-training filter:

1) Crop the image in the training set into image patches with the same size as the filter.

2) Input the sparse auto encoder, and obtain the trained weight W through the training steps of formulas (17) to (23).

3) After obtaining W, transform the connection weight corresponding to the first hidden layer node into the required filter size to obtain the pre-trained filter set $k_{ij}^{(2)}$.

(3) Calculate the convolutional feature map $x^{(2)}$ by equation (24).

(4) The local contrast of $x^{(2)}$ is normalized by equation (25), and the characteristic map $x^{(3)}$ is output.

(5) Blur $x^{(3)}$ through the down sampling layer to obtain $x^{(4)}$.

(6) Combine all the output feature maps into a column vector as the input of the fully connected network, and use the softmax classifier to obtain the image recognition results.

(7) Calculate the difference between the recognition result and the label, and adjust and update the parameters $k_{ij}^{(2)}$ through the CNN-specific back-propagation algorithm until the loss function converges to a small value and the training is completed.

(8) Input the test set, and use the filter set obtained by training and the weight parameters of the fully connected network to perform target recognition on the test image.

IV. EXPERIMENTS AND RESULTS

A. IMAGE DATA SET AND EXPERIMENTAL ENVIRONMENT

The initialization scheme proposed in this paper is finally verified by the recognition effect of the image. This paper carried out experiments on the Minist handwritten digital database, MIT face database and Oxford-17-Flowers plant and flower dataset to verify the effect of the proposed method.

The Minist handwritten digit library contains 60000 handwritten digit-training pictures and 10000 test pictures, ten types of handwritten digits. The MIT face database contains 2429 face images and 4548 non-face images, which are of two types. The Oxford-17-Flowers data set contains 17 types of flower pictures, each flower contains 80 sample pictures, and the data set contains 1360 pictures. Minist handwritten numbers are simpler than MIT, and Oxford-17-Flowers data is more complicated. This article verifies the initialization scheme on simple data and complex data, in order to get better results. The experiment divides the training set of the three types of data into 80% and the test set into 20%.

B. THE IMPORTANCE OF SAE PRE-TRAINED MODELS

The effectiveness of pre-training on SAE in this paper is verified by comparing the training iterations of pre-training with SAE and pre-training without SAE on three kinds of data. The experimental results are shown in Figure 9. It can be seen from Figure 9 that the abscissa is the number of trainings and the ordinate is the error rate. The red line represents the training curve after the network is initialized by SAE, and the green line represents the training curve after the network is only randomly initialized. From the figure, the network has three data sets after the SAE initialization; it can speed up the convergence process and achieve better results.

C. COMPARISON TEST OF MULTI-CHANNEL STRUCTURE

In this paper, the output feature dimension of channel 1 is 2700, the output feature dimension of channel 2 is 4800, and the output feature dimension of channel 3 is 4800. Therefore,

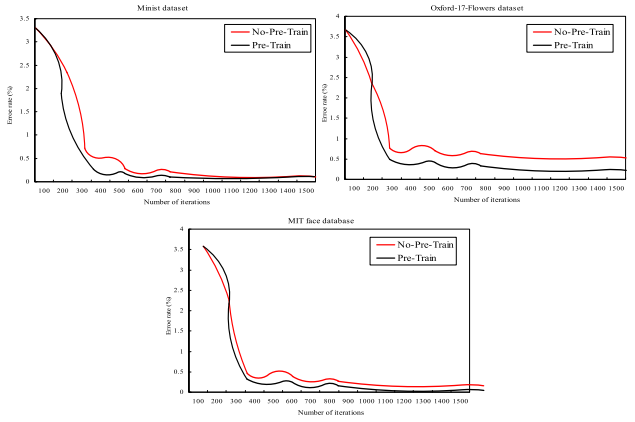

FIGURE 9. Effectiveness of pre-training with SAE.

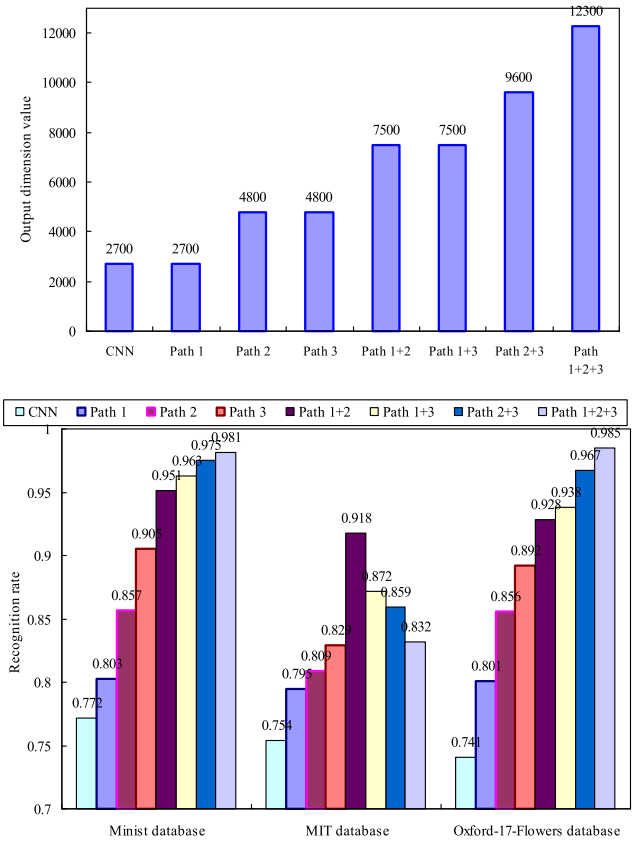

FIGURE 10. Comparison of recognition rates under different input conditions.

the output feature dimensions of the network structure proposed in this paper are 12300 in total. The training period of all algorithms is 50. The traditional CNN parameter setting is the same as the channel 1 parameter setting. It also uses 300 filters, and the initial value of the filter is obtained by random initialization. The output feature dimension is 2700. The experimental results are shown in Figure 10.

As can be seen from Figure 10, the network structure of this paper has a higher recognition rate than the original CNN on the three data sets. For the Minist dataset, it can be seen that the recognition rate of channel 2 is the highest among the three channels, and channel 3 is the lowest, because the image size of channel 3 is the smallest. The Minist class changes a lot and the target is not all in the center of the image, so the recognition rate has declined. The recognition rate is improved after two-to-one accumulation between the channels, and the recognition rate is the highest when the three channels work together. For the Oxford-17-Flowers image set, it can be seen that channel 2 has the highest recognition rate among the three paths, because the global feature difference between the different categories of the Oxford-17-Flowers image set is not obvious, but the local features can represent different categories. Due to the small input size of channel 3, the recognition rate drops slightly. In the same way, the superposition of different paths has improved the recognition rate. Finally, the recognition rate of the three-path feature fusion of the network structure in this paper reaches 0.985, which fully meets the needs for image target recognition.

It can also be seen from Figure 10 that the recognition rate of the network structure in this paper in the case of three channels CNN is higher than that of one channel or two channels CNN. It can be inferred that the features extracted by the three channels CNN have strong generalization ability and robustness. In addition, the three channels can take into account different scales, so that the model can extract features with different scales.

D. ALGORITHM ROBUSTNESS EXPERIMENT

In order to verify the robustness of the model in this paper, different types of images are selected in the data set to translate, scale, and rotate transforms. Then calculate the Euclidean distance between the first layer of fully connected features output by the algorithm and the output features after image transformation. According to the size of the distance, the robustness of the output feature to the change of the target can be measured. The smaller the Euclidean distance is, the less sensitive the feature is to target changes, and the better the robustness. For Minist, four types of targets were selected for experiments, and the comparison algorithm was CNN. For the MIT image set, randomly select 10 images for the experiment and take the average of the distance. The comparison algorithm is ICA and CNN. The test results are shown in Figure 11 and Figure 12.

It can be seen from Figure 11 that, regardless of the translation, scale or rotation transformation, the final feature vector change rate of the model in this paper is less than the CNN algorithm, which proves that its robustness is better than CNN.

It can also be seen from Figure 12 that the model in this paper shows good robustness to the translation, scale, and rotation of the remote sensing aircraft image set. In comparison, the features extracted by ICA are less robust. This resulted in a major change in characteristics. The reason why the model in this paper is more robust is firstly because the model in this paper adopts unsupervised pre-training mode, and the trained filter contains more image invariant features.

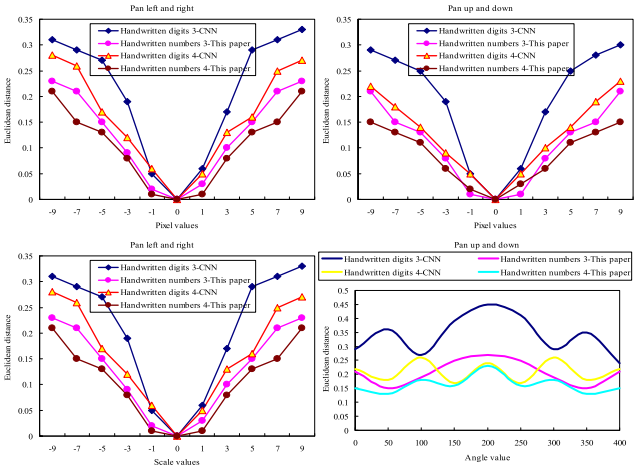

FIGURE 11. Robustness of the Minist dataset.

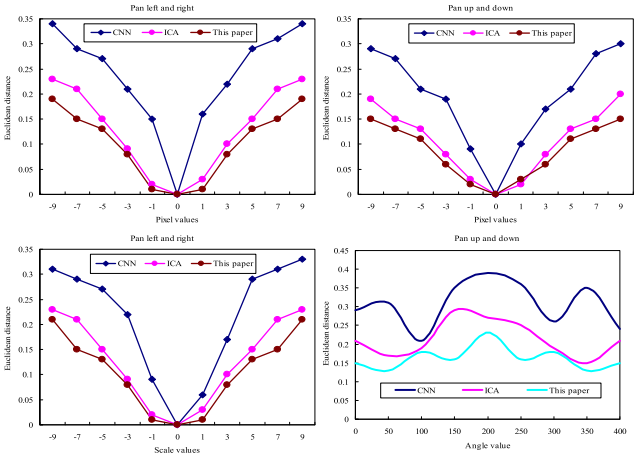

FIGURE 12. Robustness of the MIT dataset.

Secondly, because the model in this paper adopts multi-scale input, small image input is equivalent to another local feature to a certain extent, and these features have better invariance than full-scale input. Finally, the model in this paper adopts local contrast standardization, which significantly enhances the robustness of the target image with large brightness changes and noise.

In addition, the model in this paper adopts the multi-channel multi-scale block method, which will inevitably increase the network parameters, which will make the training time more time-consuming. However, in the test phase, the calculation of the input samples only includes some simple convolution and downsampling, and the complexity of the algorithm does not increase due to the increase of the channel. Therefore, the real-time performance in the test phase is not much different from the traditional CNN.

E. OPERATING EFFICIENCY

As shown in Figure 13, when the number of trainings reaches about 400, the loss function value of the model in this paper is already less than 0.05.

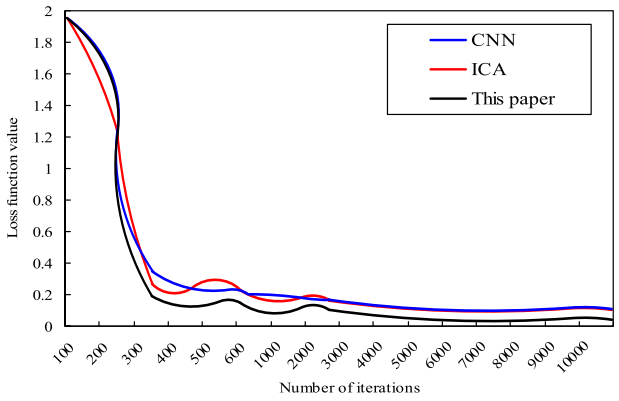

FIGURE 13. Graph of loss function results for different models.

The loss function value of the CNN part of the model in this paper is in a state of continuous decline during training. Before the training number reaches 340, the value of the loss function of the CNN model swings between 0.2 and 0.4, and there is no downward trend before the training number reaches 10000. The value of the loss function of the ICA model also declined during training. These results show that the model in this paper is much more efficient than the CNN model and ICA model.

As shown in Figure 14, the classification accuracy and time efficiency of different layers are initialized using SAE for pre-training. Time is used to indicate the initialization time plus the training network time. On the Minist data, after SAE initialized the second layer, it took 103 seconds and achieved an error rate of 1.88%. However, it takes 107 seconds to initialize the network using a random initialization method, and the error rate is 2.65%, which is higher than 1.88%. After SAE initializes five layers, better results can be achieved in the same time.

On Oxford-17-Flowers data, after SAE initialized layer 2 on the network, it took 155 seconds and the error rate was 31%. The random initialization takes 176 seconds and the error rate is 43%. That is, in the same time, using the initialization scheme of this article can achieve better results. On the Oxford-17-Flowers data, after the network 2 was initialized to five layers by SAE, it took 576 seconds to obtain a good result with an error rate of 11%. The network uses random initialization, which takes 9057 seconds and the error rate is 15%. Regardless of time or error rate, the use of random initialization is higher than the initialization scheme in this paper.

In general, the initialization scheme in this paper has better timeliness than random initialization. That is, the model in this paper can significantly improve the classification accuracy and time efficiency of the image classification process.

F. RECOGNITION OF MINIST DATASETS OF DIFFERENT MODELS

60000 samples from the Minist dataset are used for training and 10000 samples are used for testing. The stochastic

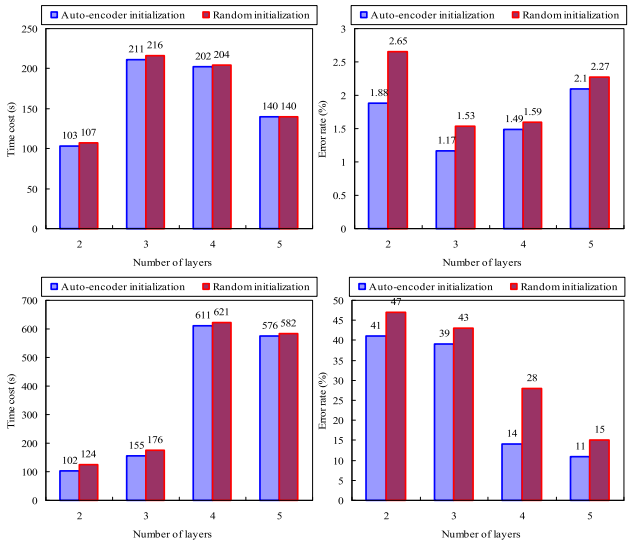

FIGURE 14. Using SAE for pre-training to initialize classification accuracy and time efficiency results for different layers.

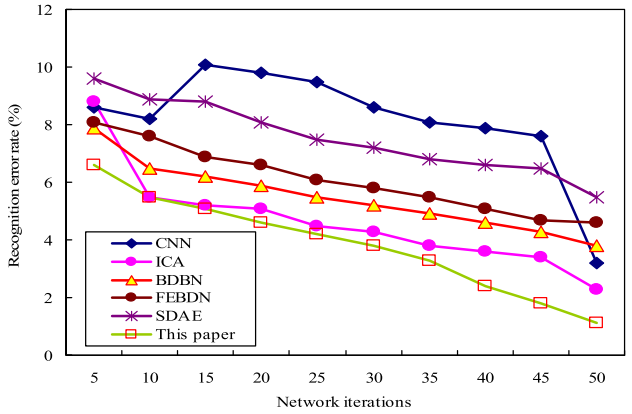

FIGURE 15. Recognition results of different models.

gradient descent method is used to train the network in batches, with 50 samples for each batch. In Figure 15, the results of the model and other models tested in the data set are given.

As can be seen from Figure 15, as the number of network iterations increases, the recognition error rate of different network models also decreases one after another, but the recognition error rate of different network models decreases to different degrees. Because there is no pre-training on the convolution kernels of CNN and ICA models, iterative training of the model is needed to extract features to improve the recognition accuracy of the test set, so it is greatly affected by the number of trainings. The BDBN model and FEBDN model memorize the new samples and new features, and improve the generalization performance. It may be that in the case of large samples, the amount of test data is large and diversified, and the new features that are further learned afterwards have little effect on the improvement of the recognition accuracy of the test set. The rate has only slightly increased. The recognition error rate of the test set of the SDAE network is significantly lower than that of the typical CNN model. The network model in this paper can achieve an ideal recognition effect in a relatively short period for large data set samples.

The experiments in this section compare and analyze the target detection performance of the model and five different network models in the test data set. The evaluation criteria used include the recall rate and false alarm rate. The calculation formula is as follows:

$$\text{Recall} = \frac{\textit{Number of correct targets detected}}{\textit{Actual number of goals}} \tag{26}$$

$$\textit{False alarm rate} = \frac{\textit{Number of detected false targets}}{\textit{Total number of detected targets}} \tag{27}$$

For the above indicators, different thresholds are set for the output of each model, where the threshold range is 0.05 $\sim$ 0.95, the interval is 0.05. Calculate and count the recall rate and false alarm rate corresponding to each threshold, and draw the curves as shown in Figure 16 and Figure 17.

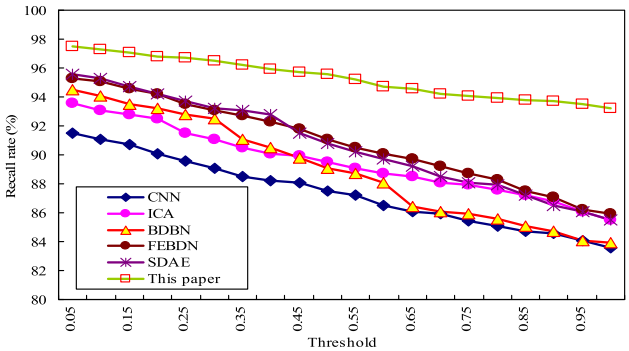

FIGURE 16. Recall rate curve.

As can be seen from Figure 16, different network models have differences in recall rates. The model proposed in this paper combines the feature information extracted by multiple CNNs with different structures, and its recall rate is improved by about 4% compared with the other five models. Therefore, the model proposed in this paper has obvious advantages in accuracy of target recognition compared to other network models. As can be seen from Figure 17, the model proposed in this paper combines the effective feature information of the samples extracted by CNNs of different structures, and merges redundant information to a certain extent, which increases the false alarm of the model. Nevertheless, the overall false alarm rate is still lower than that of other models.

In summary, the model proposed in this paper has a significant improvement in overall recognition performance compared to other network models. It can improve the recall rate, and reduce the false alarm rate to a certain extent, and can better distinguish the target and the background to verify the effectiveness of the proposed model in target recognition.

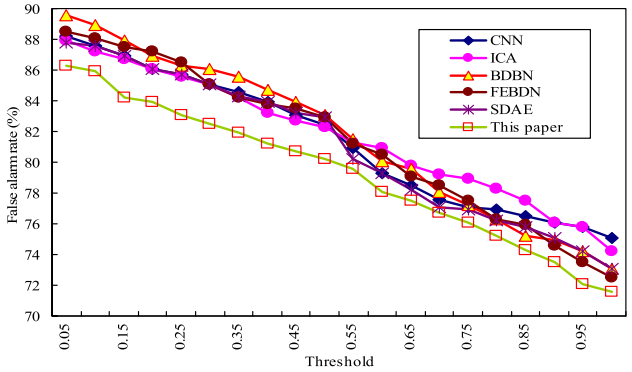

FIGURE 17. False alarm rate curve.

V. CONCLUSION

In this paper, a convolutional neural network and an automatic encoder are combined, and a method for extracting image features based on multi-pass integrated network structure for recognition is proposed. First, pre-train the convolution kernel with a sparse auto-encoder and increase the network branch to improve the original CNN model in the way of post-learning. Secondly, input and process image data of different scales to extract features separately. Then, construct a multi-channel structure, using different scale filters and sampling intervals for different channels. Finally, after sampling down from the first floor, the ground feature maps obtained by multiple channels are input into all the fully connected floors, and finally used to classify ground features. Experimental results show that the network proposed in this paper can effectively recognize images and has strong robustness to translation, scale, and rotation transformation.

REFERENCES

[1] Q. Zhang, L. T. Yang, Z. Chen, and P. Li, "A survey on deep learning for big data," *Inf. Fusion*, vol. 42, pp. 146–157, Jul. 2018.
[2] X. Sun, P. Wu, and S. C. H. Hoi, "Face detection using deep learning: An improved faster RCNN approach," *Neurocomputing*, vol. 299, pp. 42–50, Jul. 2018.
[3] S. Dorner, S. Cammerer, J. Hoydis, and S. T. Brink, "Deep learning based communication over the air," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 132–143, Feb. 2018.
[4] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," *Comput. Electron. Agricult.*, vol. 147, pp. 70–90, Apr. 2018.
[5] H. He, C.-K. Wen, S. Jin, and G. Y. Li, "Deep learning-based channel estimation for beamspace mmWave massive MIMO systems," *IEEE Wireless Commun. Lett.*, vol. 7, no. 5, pp. 852–855, Oct. 2018.
[6] R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, and R. X. Gao, "Deep learning and its applications to machine health monitoring," *Mech. Syst. Signal Process.*, vol. 115, pp. 213–237, Jan. 2019.
[7] Z. Zhang, J. Geiger, J. Pohjalainen, A. E. D. Mousa, W. Jin, and B. Schuller, "Deep learning for environmentally robust speech recognition," *ACM Trans. Intell. Syst. Technol.*, vol. 9, no. 5, pp. 1–28, 2018.
[8] Q. S. Zhang and S. C. Zhu, "Visual interpretability for deep learning: A survey," *Frontiers Inf. Technol. Electron. Eng.*, vol. v.19, no. 1, pp. 30–42, 2018.
[9] Q. Peng and Y. Song, "Object recognition and localization based on Mask R-CNN," *Qinghua Daxue Xuebao/J. Tsinghua Univ.*, vol. 59, no. 2, pp. 135–141, 2019.
[10] X. Wang, S. Feng, and C. Lang, "Semi-supervised dual low-rank feature mapping for multi-label image annotation," *Multimedia Tools Appl.*, vol. 78, no. 10, pp. 13149–13168, May 2019.
[11] Y.-B. Wang, Z.-H. You, X. Li, T.-H. Jiang, X. Chen, X. Zhou, and L. Wang, "Predicting protein-protein interactions from protein sequences by a stacked sparse autoencoder deep neural network," *Mol. BioSyst.*, vol. 13, no. 7, pp. 1336–1344, 2017.
[12] J. Herbela, T. Kacprzaka, A. Amaraa, A. Refregiera, and A. Lucchib, "Fast point spread function modeling with deep learning," *J. Cosmology Astroparticle Phys.*, vol. 2018, no. 7, p. 054, Jul. 2018.
[13] Y. Son, S.-B. Lee, H. Kim, E.-S. Song, H. Huh, M. Czosnyka, and D.-J. Kim, "Automated artifact elimination of physiological signals using a deep belief network: An application for continuously measured arterial blood pressure waveforms," *Inf. Sci.*, vol. 456, pp. 145–158, Aug. 2018.
[14] W. Zhao and S. Du, "Spectral–spatial feature extraction for hyperspectral image classification: A dimension reduction and deep learning approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 8, pp. 4544–4554, Aug. 2016.
[15] B. Pan, Z. Shi, and X. Xu, "R-VCANet: A new deep-learning-based hyperspectral image classification method," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 5, pp. 1975–1986, May 2017.
[16] S.-H. Zhong, Y. Liu, and K. A. Hua, "Field effect deep networks for image recognition with incomplete data," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 12, no. 4, pp. 1–22, Aug. 2016.
[17] X. Zhang, X. Hu, S. Wang, Y. Yang, Z. Li, and J. Zhou, "Learning geographical hierarchy features via a compositional model," *IEEE Trans. Multimedia*, vol. 18, no. 9, pp. 1855–1868, Sep. 2016.
[18] Q. Wu, "Image retrieval method based on deep learning semantic feature extraction and regularization softmax," *Multimedia Tools Appl.*, vol. 79, nos. 13–14, pp. 9419–9433, Apr. 2020.
[19] F. Ye, X. Li, and X. Zhang, "FusionCNN: A remote sensing image fusion algorithm based on deep convolutional neural networks," *Multimedia Tools Appl.*, vol. 78, no. 11, pp. 14683–14703, Jun. 2019.
[20] A. S. Garea, D. B. Heras, and F. Argüello, "Caffe CNN-based classification of hyperspectral images on GPU," *J. Supercomput.*, vol. 75, no. 3, pp. 1065–1077, Mar. 2019.
[21] A. Qayyum, A. S. Malik, N. M. Saad, M. Iqbal, M. F. Abdullah, W. Rasheed, T. A. R. Abdullah, and M. Y. B. Jafaar, "Scene classification for aerial images based on CNN using sparse coding technique," *Int. J. Remote Sens.*, vol. 38, nos. 8–10, pp. 2662–2685, May 2017.
[22] X. Wu, B. Zhou, Z. Lv, and C. Zhang, "To explore the potentials of independent component analysis in brain-computer interface of motor imagery," *IEEE J. Biomed. Health Informat.*, vol. 24, no. 3, pp. 775–787, Mar. 2020.
[23] D. Spasovski, G. Pesanski, and G. Madjarov, "The influence the training set size has on the performance of a digit speech recognition system in macedonian," *Expert Opinion Investigational Drugs*, vol. 24, no. 5, pp. 705–713, 2015.
[24] S. Saha, M. B. C. Khoo, P. S. Ng, and Z. L. Chong, "Variable sampling interval run sum median charts with known and estimated process parameters," *Comput. Ind. Eng.*, vol. 127, pp. 571–587, Jan. 2019.
[25] S. Li, H. Jiang, J. Bai, Y. Liu, and Y.-D. Yao, "Stacked sparse autoencoder and case-based postprocessing method for nucleus detection," *Neurocomputing*, vol. 359, pp. 494–508, Sep. 2019.
[26] C. S. N. Pathirage, J. Li, L. Li, H. Hao, W. Liu, and R. Wang, "Development and application of a deep learning-based sparse autoencoder framework for structural damage identification," *Struct. Health Monitor.*, vol. 18, no. 1, pp. 103–122, Jan. 2019.
[27] J. Zhang, F. Lin, P. Xiong, H. Du, H. Zhang, M. Liu, Z. Hou, and X. Liu, "Automated detection and localization of myocardial infarction with staked sparse autoencoder and TreeBagger," *IEEE Access*, vol. 7, pp. 70634–70642, 2019.
[28] Y. Tsuchiya, K. Taneishi, and Y. Yonezawa, "Autoencoder-based detection of dynamic allostery triggered by ligand binding based on molecular dynamics," *J. Chem. Inf. Model.*, vol. 59, no. 9, pp. 4043–4051, Sep. 2019.
[29] L. Chen, Q. Guan, Y. Xiong, J. Liang, Y. Wang, and Y. Xu, "A spatially constrained multi-autoencoder approach for multivariate geochemical anomaly recognition," *Comput. Geosci.*, vol. 125, pp. 43–54, Apr. 2019.
[30] V. Golkov, A. Dosovitskiy, J. I. Sperl, M. I. Menzel, M. Czisch, P. Samann, T. Brox, and D. Cremers, "Q-space deep learning: Twelve-fold shorter and model-free diffusion MRI scans," *IEEE Trans. Med. Imag.*, vol. 35, no. 5, pp. 1344–1351, May 2016.
[31] S. de Avila e Silva, S. Echeverrigaray, and G. J. L. Gerhardt, "BacPP: Bacterial promoter prediction—A tool for accurate sigma-factor specific assignment in enterobacteria," *J. Theor. Biol.*, vol. 287, pp. 92–99, Oct. 2011.

[32] E. Eqlimi, B. Makkiabadi, N. Samadzadehaghdam, H. Khajehpour, F. Mohagheghian, and S. Sanei, "A novel underdetermined source recovery algorithm based on k-Sparse component analysis," *Circuits, Syst., Signal Process.*, vol. 38, no. 3, pp. 1264–1286, Mar. 2019.
[33] W. Jia, K. Muhammad, S.-H. Wang, and Y.-D. Zhang, "Five-category classification of pathological brain images based on deep stacked sparse autoencoder," *Multimedia Tools Appl.*, vol. 78, no. 4, pp. 4045–4064, Feb. 2019.
[34] S. Riaz, Z. Ali, U. Park, J. Choi, I. Masi, and P. Natarajan, "Age-invariant face recognition using gender specific 3D aging modeling," *Multimedia Tools Appl.*, vol. 78, no. 17, pp. 25163–25183, Sep. 2019.
[35] R. Wang, H. Xie, J. Feng, F. L. Wang, and C. Xu, "Multi-criteria decision making based architecture selection for single-hidden layer feedforward neural networks," *Int. J. Mach. Learn. Cybern.*, vol. 10, no. 4, pp. 655–666, Apr. 2019.
[36] M. Shen, P. Yu, R. Wang, J. Yang, L. Xue, and M. Hu, "Multipath feedforward network for single image super-resolution," *Multimedia Tools Appl.*, vol. 78, no. 14, pp. 19621–19640, Jul. 2019.
[37] M. Ghahremani, J. Yoo, S. J. Chung, K. Yoo, J. C. Ye, and Y. Jeong, "Alteration in the local and global functional connectivity of resting state networks in Parkinson's disease," *J. Mov. Disord.*, vol. 11, no. 1, pp. 13–23, 2018.
[38] B. Kaur and J. Bhattacharya, "A convolutional feature map-based deep network targeted towards traffic detection and classification," *Expert Syst. Appl.*, vol. 124, pp. 119–129, Jun. 2019.
[39] A. Bouti, M. A. Mahraz, J. Riffi, and H. Tairi, "A robust system for road sign detection and classification using LeNet architecture based on convolutional neural network," *Soft Comput.*, vol. 24, no. 9, pp. 6721–6733, May 2020.
[40] B. He, Y. Guan, and R. Dai, "Classifying medical relations in clinical text via convolutional neural networks," *Artif. Intell. Med.*, vol. 93, pp. 43–49, Jan. 2019.
[41] F. Li, H. Liu, X. Xu, and F. Sun, "Haptic recognition using hierarchical extreme learning machine with local-receptive-field," *Int. J. Mach. Learn. Cybern.*, vol. 10, no. 3, pp. 541–547, Mar. 2019.
[42] Q. Zhou, B. Yong, Q. Lv, J. Shen, and X. Wang, "Deep autoencoder for mass spectrometry feature learning and cancer detection," *IEEE Access*, vol. 8, pp. 45156–45166, 2020.
[43] S. H. Hong, S. Ryu, J. Lim, and W. Y. Kim, "Molecular generative model based on an adversarially regularized autoencoder," *J. Chem. Inf. Model.*, vol. 60, no. 1, pp. 29–36, Jan. 2020.
[44] M. M. Elkholy, M. Mostafa, H. M. Ebied, and M. F. Tolba, "Hyperspectral unmixing using deep convolutional autoencoder," *Int. J. Remote Sens.*, vol. 41, no. 12, pp. 4799–4819, Jun. 2020.
[45] R. Zhou, Z. Xing, H. Wang, Z. Piao, Y. Huang, W. Guo, and R. Ma, "Prediction of contact fatigue life of AT40 ceramic coating based on neural network," *Anti-Corrosion Methods Mater.*, vol. 67, no. 1, pp. 83–100, Jan. 2020.
[46] A. Majumdar, R. Singh, and M. Vatsa, "Face verification via class sparsity based supervised encoding," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1273–1280, Jun. 2017.
[47] J. Fan and T. Chow, "Deep learning based matrix completion," *Neurocomputing*, vol. 266, pp. 540–549, Nov. 2017.
[48] Y. Hu, M. Lu, and X. Lu, "Driving behaviour recognition from still images by using multi-stream fusion CNN," *Mach. Vis. Appl.*, vol. 30, no. 5, pp. 851–865, Jul. 2019.
[49] C. Hou, S. C. Nicholas, and P. Verghese, "Contrast normalization accounts for binocular interactions in human striate and extra-striate visual cortex," *J. Neurosci.*, vol. 40, no. 13, pp. 2019–2043, Mar. 2020.

SEN ZHANG received the master's degree from the China University of Geosciences, in 2007. He is currently a Teacher with the School of Intelligent Engineering, Zhengzhou University of Aeronautics. His main research interests include image processing and information security.

QIUYUN CHENG received the master's degree from Shandong University, in 2007. She is currently a Teacher with the School of Intelligent Engineering, Zhengzhou University of Aeronautics. Her research interests include intelligent computing, deep learning, and image processing.

DENGXI CHEN received the master's degree from Yanshan University, in 2014. She is currently a Teacher with the Zhengzhou University of Aeronautics. Her main research interests include data mining, computer image, and information security.

HAIJUN ZHANG received the Ph.D. degree from the Wuhan University of Technology, in 2010. He is currently an Associate Professor and a Teacher with the School of Aeronautical Engineering, Zhengzhou University of Aeronautics. His research interests include intelligent computing and image processing.