Hindawi
Journal of Mathematics
Volume 2023, Article ID 4216012, 12 pages
https://doi.org/10.1155/2023/4216012

Research Article

Establishment and Test Effect of Artificial Intelligence Optimization Model Based on Convolutional Neural Network

Chunrong Zhou and Zhenghong Jiang

School of Big Data, Chongqing Vocational College of Transportation, Jiangjin 402247, Chongqing, China

Correspondence should be addressed to Zhenghong Jiang; jiangzhenghong@cqjy.edu.cn

Received 9 November 2022; Revised 4 January 2023; Accepted 28 March 2023; Published 19 May 2023

Academic Editor: Kuppusamy Sathishkumar

Convolutional neural networks (CNNs) are often used in tasks involving vision processing, and unclear images can hinder the performance of convolutional neural networks and increase its computational time. Furthermore, artificial intelligence (AI) and machine learning (ML) are related technologies, which are considered a branch of computer science, which are used to simulate and enhance human intelligence. In e-healthcare, AI and ML can be used to optimize the workflow, automatically process large amounts of medical data, and provide effective medical decision support. In this paper, the authors take several mainstream artificial intelligence models currently open on the market for reference. In this paper, the optimized model (AL-CNN) is tested for noise image recognition, and the AL-CNN model is established by using activation functions, matrix operations, and feature recognition methods, and the noisy images are processed after custom configuration. Not only does this model require no prior preparation when processing images, but it also improves the accuracy of dealing with noise in convolutional neural networks. In the AL-CNN in this paper, the architecture of the convolutional neural network includes a noise layer and a layer that can be automatically resized. After the comparison of the recognition experiments, the accuracy rate of AL-CNN is 20% higher than that of MatConvNet-moderate, and the accuracy rate is 40% higher than that of MatConvNet-chronic. In the second set of experiments, the accuracy exceeds MXNet and TensorFlow by 50% and 70%, respectively. In addition, the authors optimized the convolutional layer, pooling layer, and loss function of AL-CNN in different parameters, which improved the stability of noise processing, respectively. After customizing the two configuration optimizations, the authors found that the second optimized AL-CNN has higher recognition accuracy, and after the optimization test, the error rate can be continuously decreased as the number of recognition increases in a very short number of times.

1. Introduction

Convolutional neural network (CNN) is a kind of artificial neural network, unlike recurrent neural, Boltzmann machine, etc., because the visual system is stimulated by neural mechanisms, so the biological model of convolutional neural network can recognize two-dimensional shapes. Convolutional neural network systems can perform convolution operations. Because of its uniqueness, it has excellent performance in many aspects of life, such as image classification, retrieval, and computer-related visual tasks. With more and more research on convolutional neural networks today, this technology will be applied to road engineering, medical imaging, and artificial intelligence involving vision in the future. Also, it achieves better performance than traditional technology.

In the 21st century, artificial intelligence is constantly developing, but at present there is no intelligent model in the true sense that can possess the computing and recognition ability of the traditional neural network system of animals. The concept of deep learning is proposed in this context. Unlike traditional machine learning models, deep learning will complete learning tasks through feature learning and the participation of feature abstraction. The working principle behind the convolutional neural network is deep learning, and because of its excellent processing effect and accuracy, it has played an unprecedented role in many fields of production and life. It mainly includes printing and publishing, logistics and transportation, medical management, and other fields. In the case that there are so many precedents for convolutional neural networks to create excellent results in many fields, this article will demonstrate an intelligent model building and optimization process based on convolutional neural networks. This process has practical experience for practical operators or field beginners to construct their own program models. Moreover, in the field of e-healthcare, applications of AI and machine learning can be widely used in a variety of different scenarios. For example, they can be used to identify a patient's condition and provide personalized treatment advice. In addition, AI and ML can be used to track a patient's condition and to remind doctors to examine or adjust treatment options.

Most of the knowledge related to convolutional neural networks on the market is mostly obscure and complicated. The innovation of this paper is that this paper uses a convolutional neural network to conduct a short test case of artificial intelligence model optimization for image recognition and classification. It can provide ideas for beginners in this field to start practical operation.

2. Related Work

Understanding and predicting human visual attention mechanisms is an active area of research in neuroscience and computer vision. Among them, Kruthiventi et al. proposed DeepFix, and their model automatically learns features in a hierarchical manner and predicts saliency maps in an end-to-end manner [1]. However, their model structure is too monolithic. Qian et al. developed an architecture based on convolutional neural network to improve the recognition accuracy of noise recognition. In proposing their architecture, they investigated the optimal configuration of filters, pooling, and input feature map size. However, Qian and his team's in-depth analysis of the architecture is still insufficient [2]. In medical-related fields, Shen et al. utilized a novel multi-dimensional convolutional neural network model to classify pulmonary nodules for malignant suspicion. Their model extracts multi-scale nodule salient features in a single network, thus providing potential nodule-related applications of the proposed method [3]. Yu and Salzmann introduced a new class of convolutional neural networks utilizing second-order statistics. To this end, they designed a series of new layers that can be combined to form a covariance descriptor unit (CDU). It replaces the fully connected layers of standard convolutional neural networks, with the only drawback being a slight lack of accuracy [4]. Hou et al. proposed an efficient method for encoding with sequential spatiotemporal information and employed convolutional neural networks to learn discriminative features for action recognition. However, its disadvantage is that the scope of application is too small, and the learning steps are too complicated [5]. Inspired by the success of convolutional neural networks in the classification, a lot of effort has been devoted recently to applying convolutional neural networks to video-based action recognition problems. The challenge is that the videos contain different number of frames, which is not compatible with the standard input format of convolutional neural networks. Generally, existing methods address this problem by directly sampling a fixed number of frames or by introducing convolutional layers that perform convolutions in the spatiotemporal domain. Peng e al. proposed a novel network structure that allows any number of frames as network input. The key to their solution is to introduce a module consisting of an encoding layer and a temporal pyramid pooling layer [6]. Although this type of architecture is highly compatible, it is prone to inverse problems in imaging. In McCann et al.'s research, they reviewed the recent use of convolutional neural networks to solve inverse problems in imaging. Also, it makes it feasible to train deep convolutional neural networks on large image databases in their experiments. Motivated by these successes, they began to apply convolutional neural networks to solve inverse problems such as denoising, deconvolution, super-resolution, and image reconstruction [7].

3. Method and Steps for Establishing Noise Reduction Model according to CNN Architecture

Convolutional neural network is a multi-layer network, each layer of which is composed of multiple two-dimensional planes, and each two-dimensional plane is composed of multiple neurons [8]. The schematic diagram of the multi-layer structure of the convolutional neural network is shown in Figure 1. The structure of convolutional neural network can also be divided into three parts: input layer, hidden layer, and output layer. The input layer of the convolutional neural network structure directly inputs two-dimensional information, which is different from the traditional neural network input layer which needs to input a one-dimensional vector. The hidden layer consists of 3 kinds of networks—convolutional layer, pooling layer, and fully connected layer. In the convolutional layer, each neuron of this layer is connected to the corresponding local receptive field of the upper layer, and the features of the local receptive field are extracted through filters and nonlinear transformations. After each local feature is extracted, the spatial relationship between different local features is determined. In the pooling layer, the features extracted by the convolutional layer can be reduced in dimension, and the anti-distortion ability of the model can be increased at the same time [9]. Moreover, the fully connected layer is different from the other two layers, and the only difference between the fully connected layer and the convolutional layer is that the neurons in the convolutional layer are only connected to a local region in the input data, and the neurons in the convolutional column share the parameters. The function of the fully connected layer is to combine the things learned by each convolutional layer, complete the probability distribution after the fully connected layer, and get the final classification probability.

3.1. Activation Function. The activation function of the convolutional neural network refers to the input signal through the weight connection and the nonlinear activation function between the nodes, so as to calculate the output of

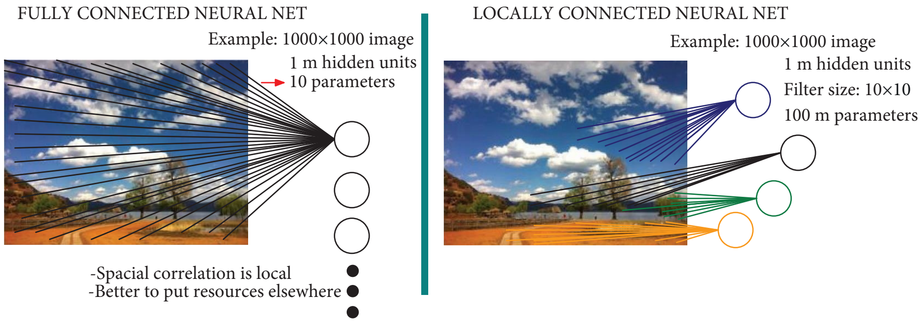

Figure 1: Schematic diagram of the multi-layer structure of convolutional neural network.

the nodes and the output of the network. Common activation functions include sigmoid, ReLU, and tan*h*, which are used to control the output of the network so that the network can output useful results.

In neural network, even though the linear function can be chosen theoretically, such as the identity function $f(x) = x$, we usually choose the nonlinear sigmoid function [10]. The nonlinear sigmoid function and function gradient are shown in Figure 2:

$$\begin{aligned}\sigma(x) &= \text{simg}(x)\\ &= \frac{1}{1+e^{-x}}.\end{aligned} \tag{1}$$

The function value of sigmoid is between 0 and 1 in the open interval. The closer the function value of the independent variable is to 0, the faster the function value changes; the larger the absolute value of the independent variable is, the slower the function value changes [11]. In addition to the nonlinear sigmoid function, the second most commonly used is the hyperbolic tangent function tah *n*:

$$\text{tahn}(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}. \tag{2}$$

In addition, the hard clipping function plus the slope function can also be used to activate the function:

$$\begin{aligned}\text{hardlimit}(x) &= \begin{cases} 1, & x \geq 0,\\ 0, & x \leq 0,\end{cases}\\ \text{ramp}(x) &= \begin{cases} 1, & x \geq 1,\\ x, & -1 \leq x \leq 1,\\ -1, & x \leq -1.\end{cases}\end{aligned} \tag{3}$$

At the same time, there are two good choices, namely, the positive linear unit ReLU and the leakage correction linear unit. Figure 3 shows the function and gradient of ReLU. The ReLU function is a commonly used nonlinear activation function, whose full name is rectified linear unit. The ReLU function, as the activation function of neurons, can be trained in the neural network to achieve accurate prediction and retrieval of learning resources [12].

$$\begin{aligned}\text{ReLU}(x) &= \max(0, x),\\ \text{LReLU}(x) &= \begin{cases} 1, & x \geq 0,\\ ax, & x \leq 0.\end{cases}\end{aligned} \tag{4}$$

In the classic structure, the activation function using a neural network is called a rectified function. In the convolutional neural network, it is often called the neural layer of ReLU; compared with the traditional activation function, ReLU can reduce the training time and improve the algorithm performance. Deep convolutional networks usually require a large amount of data for training, so that it is almost impossible for traditional activation functions to complete training with convolutional neural layers [13].

In this case, ReLU is almost the best choice. After the input image is processed by convolutional neural layers and ReLU, each pixel image in the image contains information of a small area around it, resulting in information redundancy [14]. If images containing redundant information continue to be used, not only the performance of the algorithm will be degraded, but also the translation invariance of the algorithm will be destroyed.

In order to alleviate the "dead zone" phenomenon, the researchers adjusted the part of $x < 0$ in the ReLU function to $\text{f}(x) = a * x$, where a is a small positive number of the order of 0.01 or 0.001 [15]. This new type of activation function is called Leaky ReLU:

$$\text{Leaky ReLU}(x) = \begin{cases} x(x \geq 0),\\ a \cdot x(x < 0).\end{cases} \tag{5}$$

For the setting of α in the randomized ReLU, its value obeys a uniform distribution in the training phase, and it is specified as the distribution expectation $1 + \text{u}/2$ corresponding to the uniform distribution in the testing phase:

$$\text{Randomized ReLU}(\text{x}) = \begin{cases} \text{x}(\text{x} \geq 0),\\ \alpha^{'} \cdot x(x \leq 0),\end{cases} \tag{6}$$

where

$$\alpha' \approx \text{U}(l, u), l \leq u, |, \text{u} \in [0, 1). \tag{7}$$

In addition, the exponential linear unit ELU has the advantages of the ReLU function, and the ELU also solves the "dead zone" problem of the ReLU function itself, but the

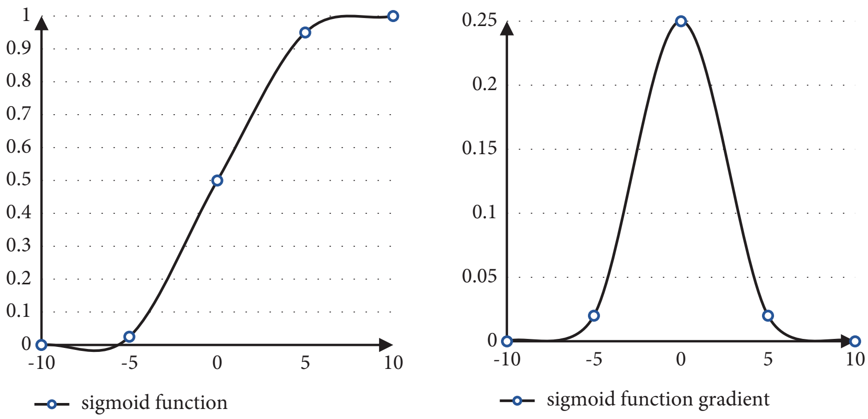

Figure 2: Sigmoid function and gradient.

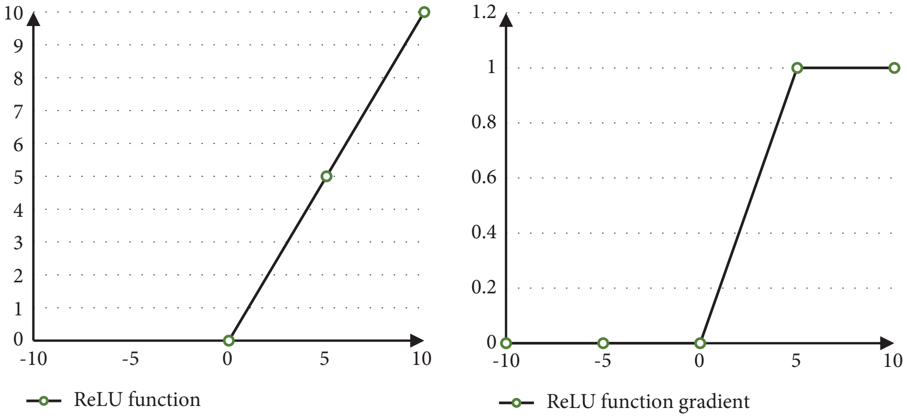

Figure 3: ReLU function and gradient.

exponential operation slightly increases the amount of calculation:

$$\mathrm{ELU}(\mathrm{x}) = \begin{cases} x, & \mathrm{x} \geq 0, \\ \lambda \cdot (\exp(x) - 1), & x < 0. \end{cases} \tag{8}$$

3.2. Matrix Operations. The matrix operation of the convolutional neural network is a special kind of matrix operation, which uses the convolution operation to process the input matrix and output the results. In convolutional neural networks, the matrix operation between input and output can be decomposed into multiple steps, including convolution, pooling, nonlinear activation function, gradient descent, and backpropagation. The combination of these steps allows the network to learn complex patterns and produce accurate results when processing the input.

Convolutional layers are the key layers of convolutional neural networks. Their main function is to extract features of various scales in the image through different convolution kernels and to ensure that the image size will not change. In functional analysis, convolution is implemented by two discrete or continuous functions [16, 17]. Convolution in image processing is usually discrete, and discrete convolution will be processed by convolution kernel. Convolution kernel is also called filter, which guarantees the core validity of convolution operation in convolutional neural network.

Suppose the length and width convolution kernels are $_{\mathrm{kl}}$ and $_{\mathrm{xp}}$, respectively, called $_{\mathrm{kl\times xp}}$ convolution. Common convolution methods include single- and multi-channel convolution, transposed convolution, and 1×1 convolution. Taking single-channel convolution as an example, the convolution process is to use a convolution kernel W (size $n \times m$) to scan the element x of the pixel matrix row of the image row by row and column by column. It multiplies each weight W in the convolution kernel w by each element x.

Adding the products in the matrix to get the dimensionality reduction matrix. If the matrix is $\mathrm{R} = (kl)_{x\times p}$, all elements of its transposed matrix $\mathrm{T} = (kl) = RT$ are defined as

$$\mathrm{T}_{\mathrm{kl}} = \mathrm{R}_{\mathrm{kl}}, \forall 1 \leq i \leq n, 1 \leq j \leq m. \tag{9}$$

If the matrix $\mathrm{R} = (rkl)_{x\times p}$, its 180° rotation is defined as

$$\mathrm{rot}\,180\,(\mathrm{R}) = \left(\mathrm{r}_{\mathrm{x-k+1,p-k+1}}\right)_{\mathrm{x\times}p}. \tag{10}$$

Given two matrices $\mathrm{R} = (rkl)_{x\times p}$ and $\mathrm{T} = (tkl)_{x\times p}$, all elements of their product $\mathrm{C} = (ckl)_{x\times p} = R \cdot T = RT$ are defined as

$$\mathrm{C}_{\mathrm{kl}} = \sum_{k=1}^{p} r_{kl} t_{kl}, \forall 1 \le i \le p, 1 \le j \le p. \qquad (11)$$

Given two matrices $\mathrm{R} = (rkl)_{x\times p}$ and $\mathrm{T} = (tkl)_{x\times p}$, their addition and subtraction are defined as

$$\mathrm{R} \pm \mathrm{T} = (\mathrm{r}_{kl} \pm t_{kl})_{x\times p}. \qquad (12)$$

The above formula is also known as element-wise product and is defined as

$$\mathrm{R} \circ \mathrm{T} = (\mathrm{r}_{kl} \bullet t_{kl})_{r\times t}. \qquad (13)$$

Given two matrices $\mathrm{R} = (rkl)_{x\times p}$ and $\mathrm{T} = (tkl)_{x\times p}$, their Kronecker product is defined as

$$\mathrm{R} \otimes \mathrm{T} = \begin{pmatrix} \mathrm{r}_{11} T \cdots r_{1p} T \\ \vdots \\ r_{x1} T \cdots r_{xp} T \\ \vdots \end{pmatrix}. \qquad (14)$$

If $\mathrm{x} = (x_1, x_2, \ldots, x_n)^T$ is a vector, then the elementwise vector function of the unary function $f(x)$ is defined as

$$f(x) = (f(x_1), f(x_2), \cdots, f(x_n))^T. \qquad (15)$$

If $\mathrm{x} = (xkl)_{x\times \mathrm{p}}$ is a matrix, then the elementwise matrix function of the unary function $f(x)$ is defined as

$$\mathrm{f}(X) = (f(x_{kl}))_{x\times p}. \qquad (16)$$

The two elementwise vector functions and elementwise matrix functions are collectively referred to as elementwise functions. If $\mathrm{x} = (x_1, x_2, \ldots, x_n)^T$, then the derivative of the elementwise vector function is

$$\mathrm{f}'(x) = \left(\mathrm{f}'(x_1), \mathrm{f}'(x_2), \cdots, \mathrm{f}'(x_n)\right)^T. \qquad (17)$$

Using this algorithm, in the practical operation of designing image processing, the resolution of the input image is reduced after each convolution operation. With the increase of layers of model network, the resolution of image will be smaller and smaller, which will cause some information loss. Also, since most noises are scanned only once, most key elements in the center of the image are lost [18]. In digital image processing, this situation can cause serious errors, so this operation method is proposed in convolution. Figure 4 shows a schematic diagram of the connected architecture of the matrix operations of this algorithm. That is to say, by performing additional operations on the edge of the image matrix, the computer core can process the image content more widely and increase the recognition range during the process of recognition and processing.

3.3. Feature Extraction Method. With feature extraction, we only need to train the topmost layer of the network, leaving the rest of the network the same. When the new dataset is relatively small and similar to the original dataset, it considers feature extraction methods. In this case, the high-level features learned from the original dataset should be transferred well to the new dataset. The biggest difference between deep learning and most machine learning algorithms is feature extraction. It is like a "black box" that relies on deep research on feature extraction to automatically process algorithms. It combines images of low-level features step by step to form high-level features through a multi-layer network structure, abstracting from local information to high-level semantic information and gradually forming a multi-level transformation [19].

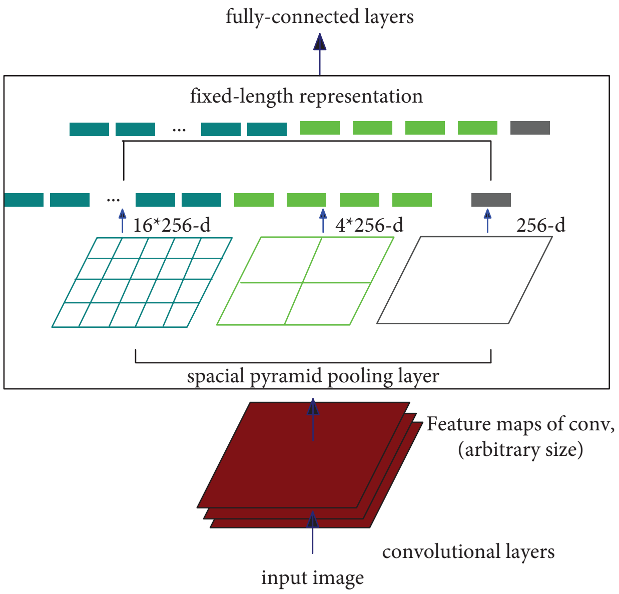

Figure 4: Schematic diagram of the connected architecture of matrix operations.

When the new dataset is large and similar to the original dataset, it considers using tuning methods. It should be safe to change the original weights because the network is less likely to overfit on new large datasets [20].

Let us consider a schematic of a pretrained convolutional neural network shown in Figure 5. Using this, we can study how the transferred knowledge can be used in different situations.

This is the method for establishing the model AL-CNN. During the establishment process, we use the basic architecture of the convolutional neural network as the basis and borrow the current open architecture to adjust the operation object of the model to the scene we need. At the same time, we prepare several open source models of the same type that need to be compared and use these different models to participate in the test and compare them in the next experiment, so that more efficient optimization can be carried out after problems are found.

4. Experimental Effect of AL-CNN Processing Different Noise Categories

Convolutional neural networks handle noise and edges by performing different operations in the image input through the convolution layer. Unlike other layers, the convolutional layer can use a filter to filter out some of the noise, thus

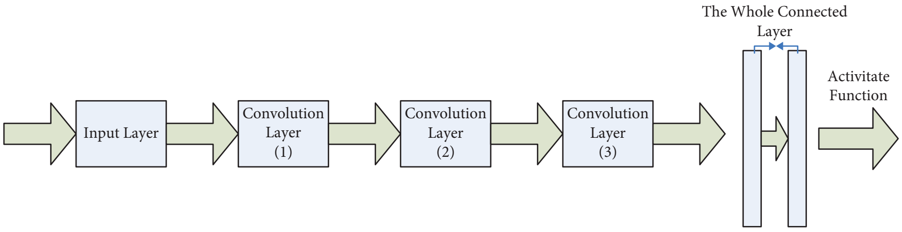

Figure 5: General convolution process trained in advance.

making the output clearer. For example, when the convolution layer is used for noise reduction, it is able to effectively eliminate the noise by using a large number of filters. Moreover, the convolution layer has a high spatial resolution (compared to the fully connected layer), thus enabling efficient smooth noise processing (i.e., edge detection).

In this section, we conduct several different classes of experiments to optimize the performance and evaluate the performance of artificial intelligence models of convolutional neural networks (AL-CNN) for vision processing. It is compared with the three models of medium and slow MatConvNet, MXNet, and TensorFlow at the same time, and the accuracy and error rate are used as evaluation criteria to determine the best optimization model, using the following formula to compare.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \tag{18}$$

4.1. Simulating Unclear Noise.

First, create different types of noise to make the image unsharp, then create an original sharp image, and add the noise to the image as lines, 2 pixels wide and 6 pixels long. Use this as a material to create a new unsharp image, and add a circular noise with a fixed radius of 4 pixels to the image. Then, to simulate the reduction of image sharpness in the packet, a solid rectangle with a width of 4 pixels and a length equal to the length of the input image is randomly added to the image. Finally, in order to create an experimental picture with incomplete image pixels, 22 rectangles are added to the image as noise. Figure 6 shows different types of noise at 5%.

4.2. Comparison of AL-CNN and MatConvNet Models.

To test the effectiveness of this model optimization, here we add the structure of the AL-CNN to the pretest stage and use it in the preprepared CNN structure. For example, the AL-CNN we want to test is configured with medium and slow architectures and is compared against the MatConvNet model. Table 1 demonstrates the parameters of the AL-CNN compared to the MatConvNet-chronic model. Here, AL-CNN contains 6 convolutional layers and 4 association layers. The size of AL-CNN is fixed, $336 * 336$ CMYK. In its convolutional layers, the model itself can automatically reduce noise. We apply the tools that can automatically reduce noise in the first algorithm to the AL-CNN model, using partial unified automation (PTA) to automatically reduce noise and optimize image sharpness. Different types of noise are changed in each experiment, and all functional layers that need to be used use the linear unit (ReLU).

In order to achieve the test effect, the authors used the OZDBTV-2017 image library containing 2200 types of images. In Table 2, the configuration of AL-CNN model parameters is compared with MatConvNet-chronic. The image library is divided into two groups: processing (800,000 high-definition images) and recognition (3,000 high-definition images) parameters included in the experiments. The top five processing deviations are used to evaluate the role of image processing, and the top five processing deviations are the main evaluation criteria for this comparison. Enter the size of the sample in OZDBTV so that the initial category is outside the category that the gallery can predict. The pixel size of the samples in the OZDBTV gallery is the product of the data to be processed. When using the algorithmic tools of the first sample, the image is much less noisy and clearer. So, we can say that the convolutional layer denoising tools and algorithms for the HD images in the gallery improved the processing of the model. Determining the algorithm at each step from the resulting graph can improve the overall tool effectiveness during processing. Automatically modifying the convolution in the proposed AL-CNN prevents the use of wrong image pixels when identifying noise.

Figure 7 shows the processing results using MatConvNet-moderate and AL-CNN when dealing with 5% noise. AL-CNN is better than MatConvNet-moderate in processing different types of samples, and the accuracy is about 20% higher in the same range. Similarly, it is still 5% noise recognition and classification. The comparison between MatConvNet-chronic and AL-CNN is shown in the second row of Figure 7. The results show that AL-CNN is also better than MatConvNet-chronic in processing noise samples. The accuracy exceeds MatConvNet-chronic by 40%.

4.3. Comparison of AL-CNN with MXNet and TensorFlow.

To further investigate the performance of AL-CNN, the paper conducted several experiments to compare it with MXNet while processing noisy images with TensorFlow. Table 3 shows the parameters of AL-CNN compared with MXNet, and Table 4 shows the comparison of parameters of AL-CNN with TensorFlow. Adjust the input image to $300*300$, at the same time adopt the method of automatic noise reduction, and use the automatic adjustment layer and automatic noise reduction layer in the highest layer. The

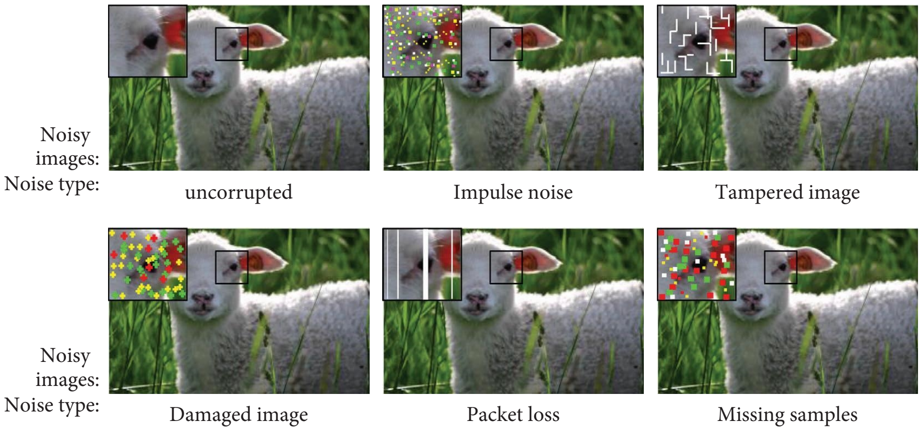

Figure 6: Different types of noise accounting for 5%.

Table 1: Parameters of AL-CNN compared with MatConvNet-moderate model.

Layer	MatConvNet-moderate	AL-CNN	Length and width	Amount	Pace	Filling
1st	Vortex	Noise observation	8 × 8	100	3	1
2nd	Type in	Automatic adjustment, type in	6 × 6	300	3	2
3rd	Vortex	Automatic vortex	6 × 6	100	3	1
4th	Tahn, PTA	Tahn, PTA	0	300	0	0
5th	Top-layer	Automatic top-layer	4 × 4	0	2	1
6th	Vortex	Vortex	6 × 6	300	2	2
7th	Tahn, PTA	Tahn, PTA	0	0	0	0
8th	Top-layer	Top-layer	4 × 4	0	2	1
9th	Vortex	Vortex	4 × 4	618	2	2
10th	Tahn	Tahn	0	0	3	1

Table 2: Parameters of AL-CNN compared with MatConvNet-chronic model.

Layer	MatConvNet-chronic	AL-CNN	Length and width	Amount	Pace	Filling
1st	Vortex	Noise observation	0	0	0	0
2nd	Type in	Automatic adjustment, type in	0	0	0	0
3rd	Vortex	Automatic vortex	8 × 8	100	1	1
4th	Tahn, PTA	Tahn, PTA	0	0	0	0
5th	Top-layer	Automatic top-layer	4 × 4	0	2	1
6th	Vortex	Vortex	6 × 6	1028	2	1
7th	Tahn, PTA	Tahn, PTA	0	0	0	0
8th	Top-layer	Top-layer	4 × 4	0	2	0
9th	Vortex	Vortex	4 × 4	1028	2	2
10th	Tahn	Tahn	0	0	0	0
11th	Vortex	Vortex	6 × 6	1028	2	2

gallery of OZDBTV-2018 used to test AL-CNN contains 2000 different types of high-definition images, and the top five deviations in the processing results will be used as the evaluation criteria for experimental testing.

Experiments show that AL-CNN, MXNet, and TensorFlow are used for noise processing image results as shown in Figure 8. Figure 8 shows the processing results under 5% noise level, and the AL-CNN configured with TensorFlow is better than other models. Compared with MXNet and TensorFlow, the AL-CNN optimized by these two configurations in this paper is 50% and 70% more accurate, respectively. It has a better effect on the reduction of noise recognition.

4.4. Comparison of Custom Configuration AL-CNN and CNN. In the following experiments, the authors customized the AL-CNN parameters and compared them with the original CNN. In this comparison, the authors used two different parameters to identify the layers damaged by noise. Table 5 shows the first custom parameters of AL-CNN, in which AL-CNN includes 4 base layers, 3 pooling layers, 2 sigmoid layers, and 2 correlation layers. In Table 6, the second parameter of AL-CNN compared with CNN is presented. In the second parameter, the tested AL-CNN contains 5 base layers, 3 pooling layers, 1 sigmoid layer, and 2 concatenated layers. In these two parameters, the added image is resized from 100∗100 to 52∗52, and using the automatic noise

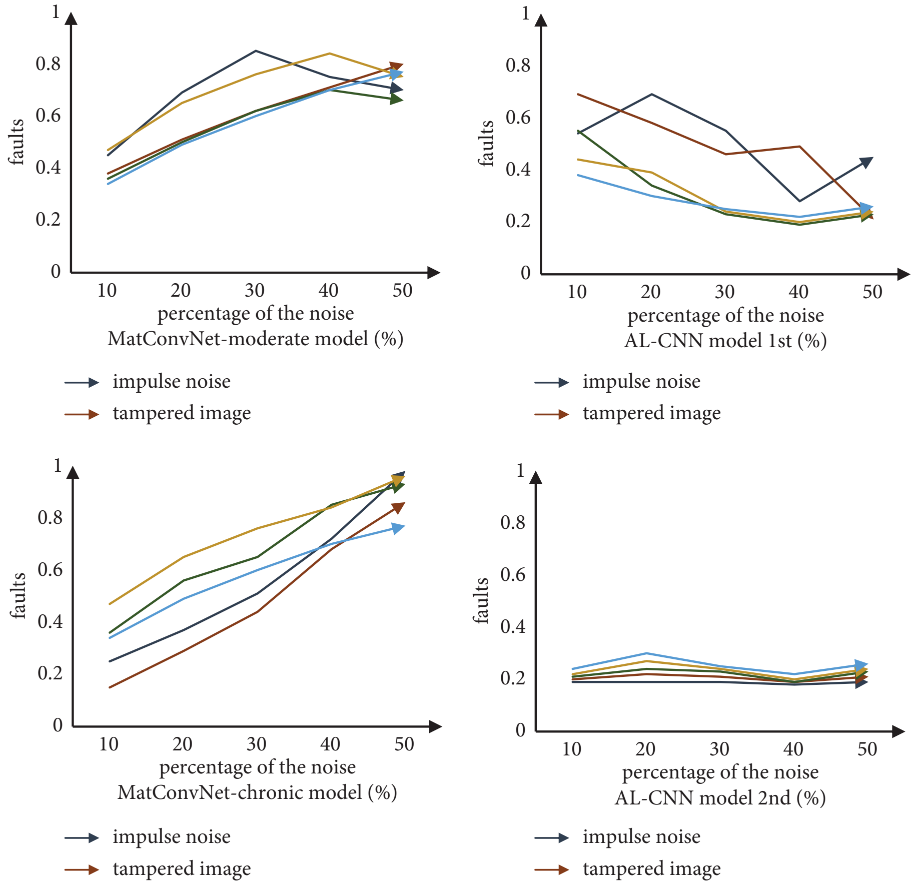

Figure 7: Comparison of AL-CNN, MatConvNet-moderate, and MatConvNet-chronic in identifying 5% noise.

Table 3: Comparison of AL-CNN parameters with MXNet.

#	MXNet	AL-CNN	Filling	Pace
1st	0	Noise process	0	0
2nd	Type in	Automatic adjustment, type in	4 × 6 × 4 × 6	0
3rd	Sigmoid	Sigmoid	4	5
4th	Tahn	Tahn	0	0
5th	Vortex	Automatic vortex	4 × 6 × 4 × 6	3
6th	PTA	PTA	0	0
7th	Sigmoid	Sigmoid	0	3
8th	Tahn	Tahn	0	0
9th	Sigmoid	Sigmoid	3	2
10th	Tahn	Tahn	0	0
11th	Sigmoid	Sigmoid	4	5
12th	Tahn	Tahn	0	3
13th	MXnet	MXnet	0	0
14th	Sigmoid	Sigmoid	4 × 6 × 4 × 6	3
15th	Tahn	Tahn	0	3
16th	Sigmoid	Sigmoid	0	3

Table 4: Comparison of AL-CNN parameters with TensorFlow.

#	TensorFlow	AL-CNN	Filling	Pace
1st	0	Noise process	0	0
2nd	Type in	Automatic adjustment, type in	4 × 6 × 4 × 6	0
3rd	Vortex	Automatic vortex	5	7
4th	Tahn	Tahn	0	0
5th	Layer	Automatic adjustment	4 × 6 × 4 × 6	3
6th	PTA	PTA	0	0
7th	Square	Square	0	7
8th	Tahn	Tahn	0	0
9th	Sigmoid	Sigmoid	5	3
10th	Tahn	Tahn	0	0
11th	Sigmoid	Sigmoid	0	0
12th	Tahn	Tahn	0	3
13th	Square	Square	4 × 6 × 4 × 6	6
14th	Maxmize	Maxmize	4 × 6 × 4 × 6	5
15th	Tahn	Tahn	0	3
16th	Maxmize	Maxmize	0	3

reduction method, the noise category varies in each layer. In testing this model, the QMODY database is used, which includes 90,000 images in 20 categories. In the experiments, 70,000 high-definition images and 40,000 noisy images are used in the test library. The evaluation criterion for model optimization is the accuracy of the recognition results.

According to the optimization experiments, it can be seen from the results in Figure 9 that AL-CNN is better than other models when using the second custom parameter, and the error rate is relatively lower. Also, it can reduce the error rate to less than 20% in only 5 times. As the number of times increases, the degree of decrease in the error rate will also be

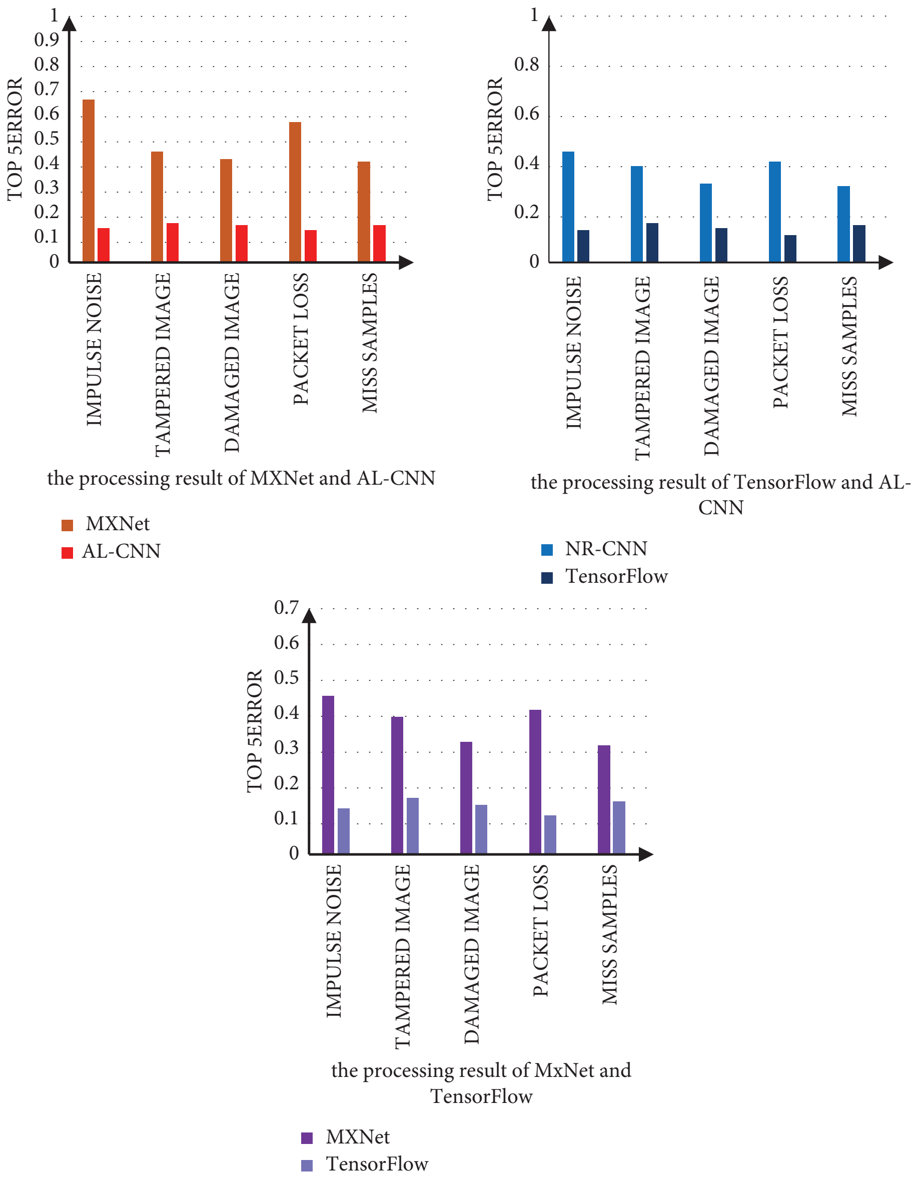

Figure 8: Comparison of recognition results of MXNet, AL-CNN, and TensorFlow on 5% density noise.

Table 5: Comparison of the first set of custom AL-CNN parameters with CNN.

Layer	CNN	AL-CNN	Length and width	Amount	Pace
1st	0	Noise process	0	0	0
2nd	Type in	Automatic adjustment, type in	0	0	0
3rd	Vortex	Automatic vortex	6*6	40	2
4th	Top-layer	Automatic top-layer	8*8	0	3
5th	Vortex	Vortex	9*9	50	4
6th	Top-layer	Top-layer	6*6	0	6
7th	Vortex	Vortex	8*8	600	4
8th	Tahn	Tahn	0	0	4
9th	Related	Related	5*5	40	4
10th	Activation	Activation	0	0	0

Table 6: Comparison of the second set of custom AL-CNN parameters with CNN.

Layer	CNN	AL-CNN	Length and width	Amount	Pace
1st	0	Noise process	0	0	0
2nd	Type in	Automatic adjustment, type in	0	0	0
3rd	Vortex	Automatic vortex	12 ∗ 12	30	3
4th	Top-layer	Automatic top-layer	8 ∗ 8	0	4
5th	Vortex	Vortex	4 ∗ 4	40	3
6th	Top-layer	Top-layer	3 ∗ 3	0	4
7th	Vortex	Vortex	8 ∗ 8	0	4
8th	Tahn	Tahn	0	0	3
9th	Related	Related	9 ∗ 9	20	5
10th	Activation	Activation	0	0	0

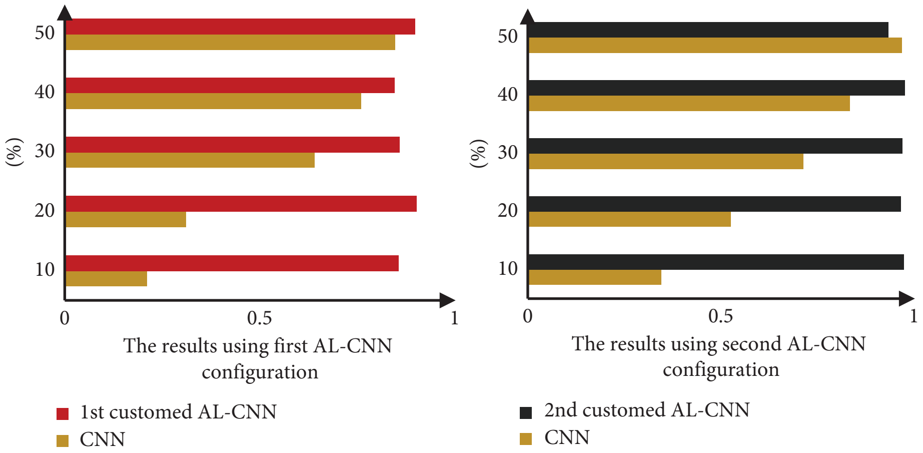

Figure 9: Comparison of noise processing results between AL-CNN and CNN with custom parameters.

greater, and finally a high-precision effect is achieved. In addition, compared with the original CNN, the optimized AL-CNN established this time has a better effect when processing images corrupted by different types of noise.

5. Discussion

Image processing is currently heavily used in fields such as computing, medicine, and engineering, and most of these scenarios require repairing missing pixels and unclear images. Data loss is a problem in the process of image processing by neural network, which can lead to the decrease of image sharpness [21]. In recent years, many scientists have begun to design automated systems that efficiently recognize and process images. Convolutional neural network image recognition is a very accurate image classification tool. It learns using convolutional classification, pooling layers, and layers of correlation connections. The network is a multilayer neural network consisting of neurons with testable weights and biases [22]. In our establishment of image processing model, it has the characteristics of high precision and high speed, which can efficiently help us process the work.

When dealing with unclear images, such images usually contain a lot of noise. The working efficiency of convolutional neural networks can be affected, and the working efficiency of convolutional neural networks can be hindered by various different types of noise during the recognition stage [23]. Moreover, when encountering low-resolution images, the noise and impurities contained in them are too complex, which will hinder the computing power of the convolutional neural network. Not to mention that resolving noise is a very CPU-intensive and time-consuming step [24, 25]. The AL-CNN established in this paper optimizes the structure of the CNN by adding a noise reduction layer and an automatic image adjustment tool to enhance its stability. In the test experiment, a large number of different types of noise have been designed for testing. The experiment shows that the established AL-CNN is efficient in processing noisy images.

After optimization tests, we can conclude that the advantages of AL-CNN are as follows. The model only needs one processing to complete the classification when dealing with a large number of images of different categories and different noise points at the same time, which not only has high speed but also has high stability. The efficiency of this model is better than other models in noisy image processing, and the contributions of this paper are as follows. Each part is described in the relevant subsections. (1) This paper establishes a convolutional neural network artificial intelligence model with excellent processing ability for noise, which improves the processing performance of noisy images. (2) The proposed AL-CNN for classifying noisy images does not require any off-architecture tool preparation. (3) In this paper, in order to improve the stability of the model in dealing with noise, an automatic adjustment noise reduction layer is added to AL-CNN. In this paper, the automatic adjustment noise reduction method is applied to the structure of the AL-CNN convolutional layer, and an automatic noise reduction algorithm is proposed to make the model stable to noise. (4) This paper proposes an automatically adjusted noise pooling operator to reduce the influence of noise on pooling layers. (5) In order to improve the recognition performance of AL-CNN, this paper proposes an automatic adjustment noise reduction method based on noise adjustment.

6. Conclusion

In this paper, the authors established an artificial intelligence model AL-CNN based on convolutional neural network and tested its computational efficiency when processing images with different noise points. When AL-CNN processes low-quality images, it does not need any external tools, does not occupy memory, and has fast computing speed. The authors designed a variety of different types of noise to simulate the processing effect of the experiment and compared the model with MatConvNet-moderate, MatConvNet-chronic, MXNet, and TensorFlow. The results show that the model is more efficient than other types in processing images. In order to make this model more stable, the authors made a custom optimization, adding a noise layer and an automatic adjustment tool to the convolutional neural network. At the same time, considering the operational efficiency of different components in AL-CNN, this paper also performs corresponding optimizations for different noise problems. In order to improve the processing performance of CNN, this paper also designs an automatic adjustment noise reduction algorithm based on convolutional neural network in response to different situations. Experimental results show that the optimized AL-CNN has better efficiency in noisy image processing compared with MatConvNet-moderate, MatConvNet-chronic, MXNet, and TensorFlow. Also, because the model does not require prior preparation or other tools, it is faster to process. Finally, AI and ML can be used to automatically identify medical images to improve diagnostic accuracy. They can also be used to analyze patient medical records and data, predict future disease trends, and provide more accurate treatment advice to physicians. Overall, AI and ML in e-healthcare contribute to improve efficiency, provide better care, and lower healthcare costs.

In the future, the goal of this paper is to use convolutional neural networks for detection in difficult-to-recognize impurity images. This paper will add other impurities, such as Gaussian blur and image liquefaction, to the new model, so that it can recognize a wider range and cover a wider range of types. On this basis, future models will have better performance after optimization.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest

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

References

[1] S. Kruthiventi, K. Ayush, and R. V. Babu, “DeepFix: a fully convolutional neural network for predicting human eye fixations,” *IEEE Transactions on Image Processing*, vol. 26, no. 9, pp. 4446–4456, 2017.

[2] Y. Qian, M. Bi, T. Tan, and K. Yu, “Very deep convolutional neural networks for noise robust speech recognition,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 12, pp. 2263–2276, 2016.

[3] W. Shen, M. Zhou, F. Yang et al., “Multi-crop Convolutional Neural Networks for lung nodule malignancy suspiciousness classification,” *Pattern Recognition*, vol. 61, no. 61, pp. 663–673, 2017.

[4] K. Yu and M. Salzmann, “Second-order convolutional neural networks,” *Clinical Immunology and Immunopathology*, vol. 66, no. 3, pp. 230–238, 2017.

[5] Y. Hou, Z. Li, P. Wang, and W. Li, “Skeleton optical spectra-based action recognition using convolutional neural networks,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 3, pp. 807–811, 2018.

[6] W. Peng, Y. Cao, C. Shen, L. Liu, and H. Tao, “Temporal pyramid layer based convolutional neural networks for action recognition,” *IEEE Transactions on Multimedia*, vol. 27, no. 12, pp. 2613–2622, 2017.

[7] M. T. Mccann, K. H. Jin, and M. Unser, “Convolutional neural networks for inverse problems in imaging: a review,” *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 85–95, 2017.

[8] U. R. Acharya, H. Fujita, O. S. Lih, M. Adam, J. H. Tan, and C. K. Chua, “Automated detection of coronary artery disease using different durations of ECG segments with convolutional neural network,” *Knowledge-Based Systems*, vol. 132, no. sep.15, pp. 62–71, 2017.

[9] L. Bondi, L. Baroffio, D. Güera, P. Bestagini, E. J. Delp, and S. Tubaro, “First steps toward camera model identification with convolutional neural networks,” *IEEE Signal Processing Letters*, vol. 24, no. 3, pp. 259–263, 2017.

[10] C. Li, Y. Hou, P. Wang, and W. Li, “Joint distance maps based action recognition with convolutional neural networks,” *IEEE Signal Processing Letters*, vol. 24, no. 5, pp. 624–628, 2017.

[11] J. Lee and J. Nam, “Multi-level and multi-scale feature aggregation using pre-trained convolutional neural networks for music auto-tagging,” *IEEE Signal Processing Letters*, vol. 24, no. 8, pp. 1208–1212, 2017.

[12] W. Yao and N. Li, “Construction of artificial intelligence-assisted English learning resource query system,” *Frontiers in Psychology*, vol. 13, Article ID 970497, 2022.

[13] I. Evo and A. Avramovi, “Convolutional neural network based automatic object detection on aerial images,” *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 5, pp. 740–744, 2017.

[14] Y. D. Hezaveh, L. P. Levasseur, and P. J. Marshall, “Fast automated analysis of strong gravitational lenses with

convolutional neural networks," *Nature*, vol. 548, no. 7669, pp. 555–557, 2017.

[15] W. Li, F. Jia, and Q. Hu, "Automatic segmentation of liver tumor in CT images with deep convolutional neural networks," *Journal of Computer and Communications*, vol. 03, no. 11, pp. 146–151, 2015.

[16] C. Zhang, P. Wang, C. Ke, and K. Joni-Kristian, "Identity-aware convolutional neural networks for facial expression recognition," *Journal of Systems Engineering and Electronics*, vol. 28, no. 4, pp. 784–792, 2017.

[17] M. Alotaibi and A. Mahmood, "Improved gait recognition based on specialized deep convolutional neural network," *Computer Vision and Image Understanding*, vol. 164, no. NOV, pp. 103–110, 2017.

[18] M. H. Yap, G. Pons, J. Marti et al., "Automated breast ultrasound lesions detection using convolutional neural networks," *IEEE Journal of Biomedical and Health Informatics*, vol. 22, no. 4, pp. 1218–1226, 2018.

[19] P. Ghamisi, B. Hofle, and X. X. Zhu, "Hyperspectral and LiDAR data fusion using extinction profiles and deep convolutional neural network," *Ieee Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no. 6, pp. 3011–3024, 2017.

[20] N. Liu, J. Han, T. Liu, and X. Li, "Learning to predict eye fixations via multiresolution convolutional neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 2, pp. 392–404, 2018.

[21] S. Poria, H. Peng, A. Hussain, N. Howard, and E. Cambria, "Ensemble application of convolutional neural networks and multiple kernel learning for multimodal sentiment analysis," *Neurocomputing*, vol. 261, no. 25, pp. 217–230, 2017.

[22] J. Zhu, S. Liao, Z. Lei, and S. Z. Li, "Multi-label convolutional neural network based pedestrian attribute classification," *Image and Vision Computing*, vol. 58, pp. 224–229, 2017.

[23] G. Cheng, Y. Wang, S. Xu, H. Wang, S. Xiang, and C. Pan, "Automatic road detection and centerline extraction via cascaded end-to-end convolutional neural network," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 6, pp. 3322–3337, 2017.

[24] K. B. Lee, S. Cheon, and C. O. Kim, "A convolutional neural network for fault classification and diagnosis in semiconductor manufacturing processes," *IEEE Transactions on Semiconductor Manufacturing*, vol. 30, no. 2, pp. 135–142, 2017.

[25] Y. Zhou, H. Wang, F. Xu, and Y. Q. Jin, "Polarimetric SAR image classification using deep convolutional neural networks," *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 12, pp. 1935–1939, 2016.