Neural Process Lett (2016) 43:389–399
DOI 10.1007/s11063-015-9420-y

Adaptive Convolutional Neural Network and Its Application in Face Recognition

Yuanyuan Zhang[1] · Dong Zhao[2] · Jiande Sun[2,3] · Guofeng Zou[4] · Wentao Li[2]

Published online: 31 March 2015

Abstract Convolutional neural network (CNN) has more and more applications in image recognition. However, the structure of CNN is often determined after a performance comparison among the CNNs with different structures, which impedes the further development of CNN. In this paper, an adaptive convolutional neural network (ACNN) is proposed, which can determine the structure of CNN without performance comparison. The final structure of ACNN is determined by automatic expansion according to performance requirement. First, the network is initialized by a one-branch structure. The system average error and recognition rate of the training samples are set to control the expansion of the structure of CNN. That is to say, the network is extended by global expansion until the system average error meets the requirement and when the system average error is satisfied, the local network is expanded until the recognition rate meets the requirement. Finally, the structure of CNN is determined automatically. Besides, the incremental learning for new samples can be achieved by adding new branches while keeping the original network unchanged. The experiment results of face recognition on ORL face database show that there is a better tradeoff between the consumption of training time and the recognition rate in ACNN.

Keywords Convolutional neural network · Network construction · Adaptive convolutional neural network · Global expansion · Local expansion · Incremental learning

✉ Jiande Sun
jd_sun@sdu.edu.cn

[1] Information Research Institute, Shandong Academy of Sciences, Jinan 250014, China

[2] School of Information Science and Engineering, Shandong University, Jinan 250100, China

[3] The Hisense State Key Laboratory of Digital-Media Technology, Qingdao 266061, China

[4] College of Electrical and Electronic Engineering, Shandong University of Technology, Zibo 255049, China

1 Introduction

Convolutional neural network (CNN) can extract feature from 2D images directly, so that it has been applied to image classification gradually. LeCun et al. [1] first designed CNN and trained it based on the error gradient in 1989. They used CNN to classify the handwritten digits and achieved the best result at that time. Lawrence et al. [2] presented a hybrid neural network solution for face recognition, which combined SOM and CNN. Christophe et al. [3] realized face detection using CNN with three layers, including a convolutional layer, a sampling layer and a MLP layer. In 2012, Hinton et al. [4] trained a large and deep CNN and achieved unprecedented success in the ImageNet contest. They classified the 1.2 million high-resolution natural images into 1000 different classes by the deep CNN, which enhanced the researchers' confidence about CNN.

Apparently, CNN has an advantage in image recognition. However, there is no theory about the network construction of CNN, such as the number of layers, the number of feature maps each layer, and so on. Researchers construct several CNN candidates based on their experience and determine the final one based on the performance comparison, which brings a huge cost and impedes the further development of CNN. To solve this problem, some studies are on hardware acceleration, which is made to speed up the performance comparison, for instance, GPU-based programming and scalable hardware architecture [5–7]. Other studies are on optimization of the structure of CNN. Cheung et al. [8] applied a hybrid evolutionary search procedure to define the initialization and architectural parameters of convolutional networks. They made use of stochastic diagonal Levenberg–Marquardt to accelerate the convergence of training, lowering the time cost of fitness evaluation. Wu et al. [9] constructed cascaded heterogeneous CNNs. Each CNN recognizes a proportion of input samples with high-confidence and feeds the rejected samples into the next CNN. They achieved an error rate 0.23 % using only five CNNs in MNIST dataset experiments. Sermanet et al. [10] applied convolutional networks (ConvNets) to the task of traffic sign classification and the ConvNets were biologically-inspired multi-stage architectures that automatically learn hierarchies of invariant features. The traditional ConvNet architecture was modified by feeding first stage features in addition to second stage features to the classifier. Wu et al. [11] presented a hybrid DNN (HDNN), by dividing the maps of the last convolutional layer and the maxpooling layer of DNN into multiple blocks of variable receptive field sizes or max-pooling field sizes, to enable the HDNN to extract variable-scale features. Although these variants of CNN show a better performance, they still depend on the basic structure of traditional CNN which relies on prior knowledge and experience. Gu et al. [12] tried to get rid of prior knowledge and constructed an incremental convolutional neural network (ICNN). ICNN simplifies the structure of CNN and makes the structure adjustable; however, the final structure is still determined through the performance comparison of a variety of structures, instead of in an automatic way.

In this paper, we propose an adaptive convolutional neural network (ACNN), whose structure can be determined by automatic expansion according to performance requirements, i.e. the system average error and recognition rate of training rate. First, the network is initialized with a simple architecture and each layer of this network has a single feature map. Second, the initial network is judged whether it is convergent or not in ACNN, which is different from ICNN. If it is convergent, there is no global expansion and the initial network will be trained to satisfy the predefined system average error. If not, the network will be extended by global expansion until system average error meet the requirement. The scale of global expansion is determined automatically in ACNN instead of the artificial controlling in ICNN. Third, different from that in ICNN, ACNN takes the recognition rate of training samples to control

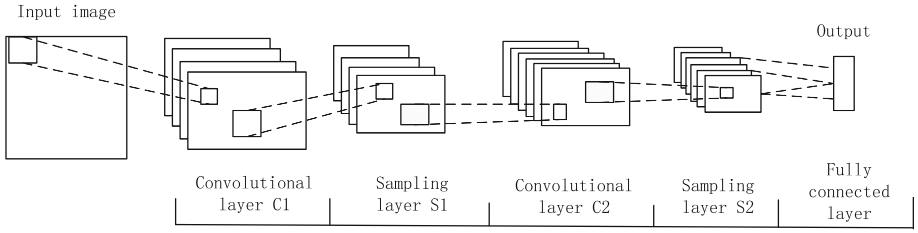

Fig. 1 Architecture of traditional CNN

the local expansion. If the recognition rate of training sets on the global network is unexpected, the network will be extended further through local expansion until the recognition rate of training sets reaches the target. Finally, the structure of ACNN is determined in a fully automated process. Based on the network adaptive expansibility, the incremental learning for new samples can be achieved by adding new branches while keeping the original network unchanged. We apply the proposed ACNN to face recognition and the experiment results on ORL face database show that there is a better tradeoff between the consumption of training time and the recognition rate in ACNN.

2 Convolutional Neural Network

CNN is a multi-layer neural network and is designed especially for 2D image classification. Each layer of CNN is composed of multiple 2D planes and each 2D plane consists of many independent neurons. Figure 1 shows the architecture of the traditional CNN, which involves three operations: convolution, sampling and outputting.

As shown in Fig. 1, the input image is involved with four trainable filters to produce four feature maps at convolutional layer C1. Each 2 × 2 block in the feature maps are added, weighted, combined with a bias and passed through a sigmoid function to produce four feature maps at sampling layer S1. These are filtered again to produce the convolutional layer C2. The hierarchy then produces S2 in a manner analogous to S1. Finally, the fully connected layer combines all pixels of S1 into a 1D output vector.

The feature map of convolutional layer is defined as:

$$x_j^l = f\left(\sum_{i \in M} x_i^{l-1} * k_{ij}^l + b_j^l\right), \tag{1}$$

where x_j^l is the jth feature map at the lth layer, $f(\circ)$ is the sigmoid function, M is the number of feature maps at layer $l-1$, x_i^{l-1} is the ith feature map at the $l-1$th layer, "$*$" means the operation of convolution, k_{ij}^l is the kernel between the ith feature map at the $l-1$th layer and the jth feature map at the lth layer, and b_j^l is the bias for the jth feature map at the lth layer.

The feature map of sampling layer is defined as:

$$x_j^l = f(\delta_j^l S(x_j^{l-1}) + b_j^l), \tag{2}$$

where $S(\circ)$ is the down-sampling function, which added and averaged each group of $2 * 2$ block in the feature map, and δ_j^l is the multiplier deviation for the jth feature map at the lth layer.

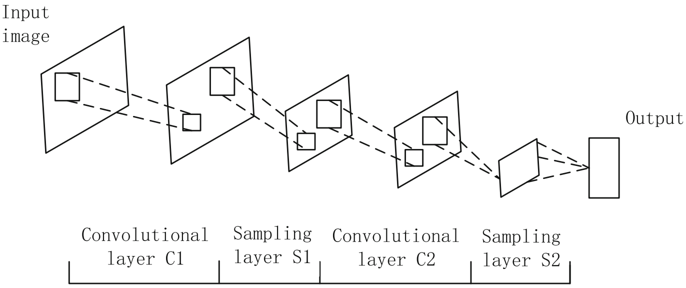

Fig. 2 The initial network

The output layer, i.e., the fully connected layer, is defined as:

$$
\begin{aligned}
Q_i &= \sum_{j=1}^{N}\sum_{k=1}^{M} x_k^j \omega_{ijk} + b_i \\
y_i &= f(Q_i) \\
Q &= [Q_1, \ldots, Q_i, \ldots, Q_m]' \\
y &= [y_1, \ldots, y_i, \ldots, y_m]',
\end{aligned}
\tag{3}
$$

where y_i is the value of the ith output unit, N is the number of feature maps of the layer before the output layer, M is the number of neurons of each feature map, ω is the weight, and m is the number of classes. For quoting easier, Q denotes what the sigmoid function affects. When input an image, both Q and Y are the matrices with size of $m \times 1$.

3 Adaptive Convolutional Neural Network

Generally, network performance will be improved when the number of neurons is increased [13], however, the experiment indicates that the benefit by increasing the network size is out of proportion with the time consumption if the network is expanded blindly. This is mainly because that the architecture of traditional CNN is constructed by experience and there is no theory about the parameters setting, such as the number of layers, the size of convolutional kernel, and so on.

To solve the problems mentioned above partially, we propose an ACNN, which can construct the network automatically with a simple initial network structure and several predefined parameters.

The algorithm of ACNN is as following:

Step 1 Network Initialization.

The initial network is set with only one branch as shown in Fig. 2. This branch includes two convolutional layers and two sampling layer. Each of the four layers has one feature map.

The weights of the network are updated via back-propagation (BP).

Different from ICNN in [9], the initial network will be judged whether it is convergent or not under the specified number of training times, which is helpful to control the time consumption. The condition of convergent trend is given by the expression

$$err_{initial} - err_{present} \geq T, \tag{4}$$

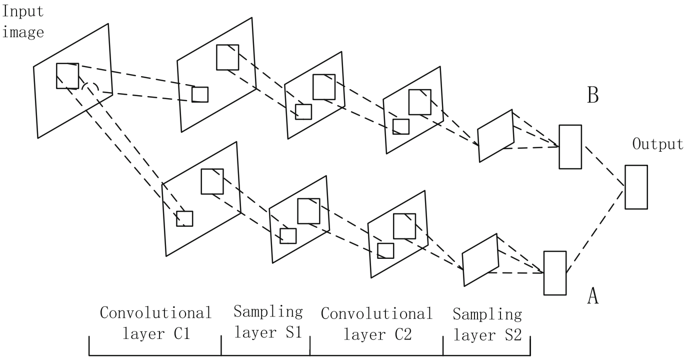

Fig. 3 Global expansion network with two branches

where $err_{initial}$ is system average error of the former training, $err_{present}$ is average error of the current training, and T is the threshold value of convergence rate, which is set to 0.1.

The system average error err is

$$err = \frac{1}{2}\frac{\sum_{j=1}^{a}\sum_{i=1}^{m}(y - y_{label})^2}{a}, \tag{5}$$

where a is the total number of training samples and m is the number of output units. If (4) is satisfied, the present network can be considered convergent and it is trained until the system average error meets the requirement. Otherwise the present network isn't convergent and the global expansion is needed.

Step 2 Global Network Expansion.

The global network expansion of a new branch B on the foundation of initial branch A is shown in Fig. 3. Before expansion, Q_A of training and testing samples are preserved separately at the output layer of branch A, and Q_A is overlaid to the output layer of branch B. The output of branch B is defined as:

$$y = f(Q_A + Q_B). \tag{6}$$

The weights of branch A are fixed and only the weights of branch B are updated by BP. If the expanded branch can't also be convergent under the specified number of training times, another global branch will be expanded. When the convergent branch is obtained, it will be trained till the average error of the system reaches the target. The learning of global network of ACNN is finished. As we can see that the way of global expansion of ACNN is better than that of ICNN on reduce human intervention.

Step 3 Local Network Expansion.

In [9], when to expand global network or local network is decided artificially and the precondition for local expansion is same with global expansion.

When the system average error is less than the threshold value after global expansion, but the recognition rate of training samples still can't reach the target, a local network expansion is needed. To improve the precision of CNN, a fusion of global branches is proposed to produce local network as shown in Fig. 4.

Before expansion, the feature maps of every branch in global network are preserved as inputs of local network, and the proposed fusion is

$$\mathrm{C1}_{local} = f(\mathrm{S1}_A * k_A + \mathrm{S1}_B * k_B), \tag{7}$$

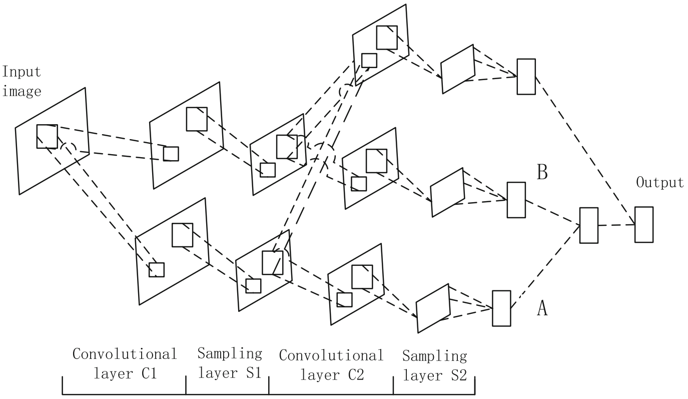

Fig. 4 Global and local expansion network

where $C1_{local}$ is the feature map at first convolutional layer of local network, $S1_A$ and $S1_B$ are the feature maps at the layer S1 of branch A and B. k_A and k_B are the convolutional kernels.

At the same time, Q_{global} at output layer of global network is preserved and is overlaid to the output layer of the local branch. The output of local network is defined as:

$$\begin{aligned} Q_{global} &= Q_A + Q_B \\ y &= f(Q_{global} + Q_{local}). \end{aligned} \tag{8}$$

The weights of global network are fixed and only the weights of local branch are updated by BP. The learning of whole network is finished until the recognition rate of the training samples reaches the target, which ensures the precision of ACNN.

Specially, there has no fusion when the global network includes only one branch, so the feature map at layer S1 of global network is convolved with different convolution kennel as shown in Fig. 5.

Obviously, ACNN constructs network automatically instead of depending on experience. Meanwhile, ACNN trains network referring to the predefined system error and recognition rate rather than increasing the number of training times blindly like CNN, which can reduce training time and ensure accuracy.

4 Incremental Learning Based on ACNN

In practical applications, incremental learning is important. Due to the complexity, CNN will spend a huge cost when we retrain the classifier for the new samples. And even if CNN learns new knowledge through retraining, it will lose the memory of previous training.

Based on the expansibility of ACNN, incremental learning can be achieved only by global expansion based on the existing network. Figure 6 shows the architecture of incremental learning. Branch 1 and branch 2 are original global branches, branch 3 is the original local network and branch 4 is the new branch for increasing samples.

The weights of original network are fixed and only the weights of new branch are updated by BP. The learning of whole network is finished until the system average error reaches the target, and the training process of incremental learning is same as that of global expansion.

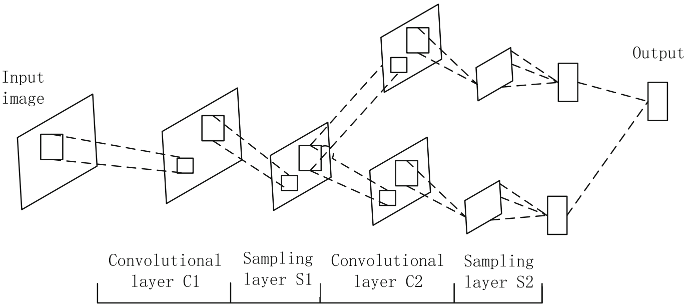

Fig. 5 Global and local expansion network where the global expansion has only one branch

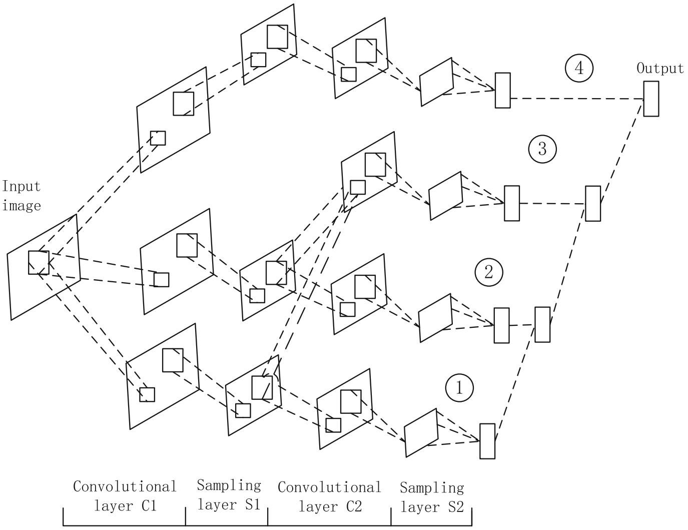

Fig. 6 The architecture of incremental learning

The output of whole network is defined as:

$$y = f(Q_1 + Q_2 + Q_3 + Q_4). \tag{9}$$

This method can preserve the previous memory by keeping the original network unchanged; on the other hand, it can learn knowledge from new samples by expanding a new branch.

5 Experiment and Analysis

5.1 Configuration of Face Database

In order to verify the validity of the proposed method, based on the ORL face database, some experiments and analysis have been done. There are a total of 40 persons with 10 different face images for each person. The images are grayscale with resolution of 92 × 112. All images are preprocessed, including size normalization to 64 × 64 pixels, rotating and illumination normalization. The original images and images after preprocessing are showed in Fig. 7.

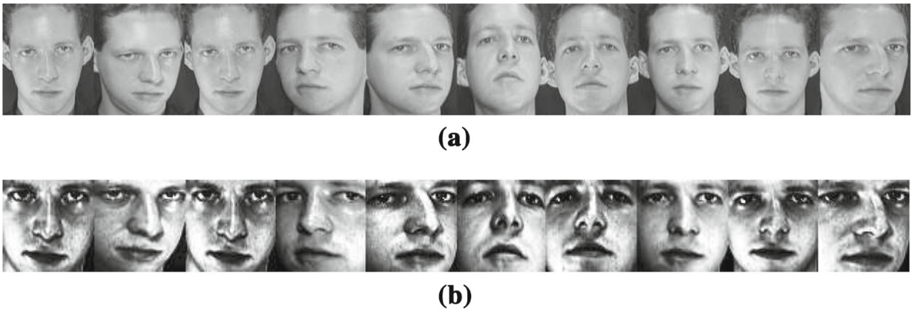

Fig. 7 **a** The original images, **b** the images after preprocessing

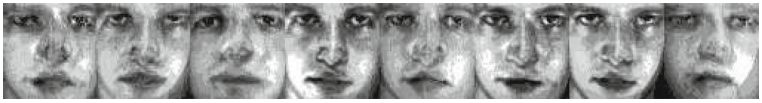

Fig. 8 Examples of the generated training images from the original training images

Table 1 Experiments of CNN with different structure

Structure	Training time (s)	Training recognition rate	Test recognition rate
2-2-4-4	350	0.9983	0.9417
4-4-4-4	400	0.9983	0.9417
4-4-6-6	561	0.9967	0.9333
6-6-6-6	803	0.9967	0.9417
6-6-8-8	1695	0.9967	0.9250
8-8-8-8	1206	0.9983	0.9333
8-8-10-10	2363	0.9983	0.9417

The first seven images of each person are selected as the training samples, and the remaining images are used as the test samples. In order to increase the number of training samples, we generate another eight training images for each person by taking a weighted average of three images chosen from the original training images of each person randomly. Figure 8 shows the generated images. In total, the training samples consist of 600 images and the testing samples consist of 120 images.

5.2 The Experiment and Analysis About CNN

First of all, the network is determined as 4-layer network, which is used mostly. The activation function is sigmoid function and the learning rate is 0.3. The kernel size at convolution layer 1 and convolution layer 2 are 5×5 and 3×3, respectively. These parameters are same throughout the experiment.

To determine the structure of CNN, we train some CNNs with different structure and choose one with best performance of them, which are listed in Table 1. The structure with 2-2-4-4 means the number of feature maps of C1, S1, C2 and S2 in Fig. 1 are 2, 2, 4 and 4 respectively. Each feature map of convolution layer is connected with all the feature maps of the previous sampling layer by convolving with different kernels.

Table 2 Experiment results of ACNN

Algorithm	Structure	Training time (s)	Test recognition rate
Global expansion ACNN	1-1-1-1	275	0.9167
Global + local expansion ACNN	1-1-2-2	343	0.9333

Table 1 shows the performance of CNN with different structure. According to the data in Table 1, the final structure is determined as 2-2-4-4 for its least training time and highest training recognition rate. The test experiment is done based on the test samples, and its recognition rate is 94.17 %. In Table 1, the sum of the training time of different structures is taken as the whole training time of the traditional CNN structure determination method.

5.3 The Experiment and Analysis About ACNN

The initial network structure of ACNN is 1-1-1-1 as shown in Fig. 2 and local network structure is 1-1 as shown in Fig. 4.

Table 2 gives the experiment results of ACNN. From Table 2, we have the following observations:

- The recognition rate has increased by 1.66 %, from 91.67 to 93.33 %, through expanding the local branch based on global expansion of ACNN, which verifies the feasibility of the ACNN.
- It takes only 343 s to train the network and finally get a determinate network structure 1-1-2-2, without any performance comparison of different structures as in Table 1. ACNN reduces the manual intervention of CNN greatly and constructs the network automatically.
- The network with structure 4-4-6-6 and 8-8-8-8 in Table 1 have the same recognition rate 93.33 % with our method, but their training time are more than 343 s and their structure are far more complex than ACNN. When the data set is very large and the training resource is limited, ACNN is expected to show greater advantages.

5.4 The Cross Validation Experiments

In this section, the cross validation experiments have been done to show the feasibility of ACNN. In cross validation experiments, we randomly select seven images of each person and also generate another eight images to be the training samples, the rest three images are testing samples. We have done the experiment 1000 times with different date sets. The experiment results are shown in Fig. 9 and Table 3.

According to the data of Table 3, we can see that the mean of recognition rate has increased by 2.36 %, from 88.03 to 90.39 %, through expanding the local branch based on global expansion, which verifies the feasibility of ACNN.

5.5 The Experiment and Analysis About Incremental Learning

We use the data set same with that in Sects. 5.2 and 5.3 to do incremental learning experiment.We select 10 images from the 15 training images per person to train the network and the remaining five images are used as the samples of incremental learning. The test samples are still three images per person.

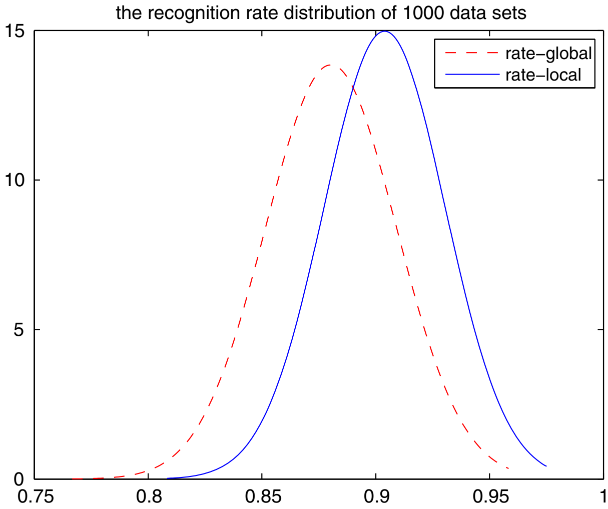

Fig. 9 The performance comparison of the network with local expansion and without local expansion

Table 3 The results about cross validation experiments

Algorithm	Mean recognition rate	Standard deviation
Global expansion ACNN	0.8803	0.0288
Global + local expansion ACNN	0.9039	0.0266

Table 4 The experiment results about incremental learning

Algorithm	Structure	Training time (s)	Test recognition rate
Global expansion ACNN	1-1-1-1	296	0.9083
Global + local expansion ACNN	1-1-2-2	391	0.9333
Global + local expansion + incremental learning ACNN	2-2-3-3	391+10	0.9500

Table 4 shows the strong feasibility of incremental learning in ACNN. It takes only 10 s to learn the new samples, which is much less that of global + local expansion ACNN. Obviously, ACNN has an advantage of incremental learning because of the very small time overhead.

6 Conclusion

In this paper, an ACNN is proposed. By taking system average error and recognition rate into account to control control the network expansion, ACNN constructs the network automatically without any performance comparison, which takes much less training time and makes the training easier than the traditional CNN. ACNN makes the network structure controllable and adjustable by expanding the network adaptively. Based on adaptive expansion of ACNN, incremental learning is realized with a little retraining cost. The experiments results based on

ORL face recognition verify the feasibility and advantage of ACNN. The proposed ACNN provides the probability of finding the approximate optimal structure of CNN in a simple way and it is extremely essential in practical application with large scale data.

Acknowledgments Project supported by the Nature Science Foundation of Shandong Province (No. ZR2014FM012, and No. ZR2014YL010), the Shandong Science and Technology Development Program (No. 2012GSF12004).

References

1. LeCun Y, Boser B, Denker JS et al (1989) Backpropagation applied to handwritten zip code recognition. Neural Comput 1(4):541–551
2. Nebauer C (1998) Evaluation of convolutional neural networks for visual recognition. IEEE Trans Neural Netw 9(4):685–696
3. Garcia C, Delakis M (2002) A neural architecture for fast and robust face detection 2002. In: Proceedings 16th international conference on pattern recognition, IEEE, vol 2, pp 44–47
4. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, pp 1097–1105
5. Farabet C, Martini B, Akselrod P, et al. (2010) Hardware accelerated convolutional neural networks for synthetic vision systems. In: Proceedings of 2010 IEEE international symposium on circuits and systems (ISCAS) IEEE, pp 257–260
6. Peemen M, Setio AAA, Mesman B, et al. (2013) Memory-centric accelerator design for convolutional neural networks. In: IEEE 31st International conference on computer design (ICCD), IEEE, 2013, pp 13–19
7. Strigl D, Kofler K, Podlipnig S (2010) Performance and scalability of GPU-based convolutional neural networks. In: 18th Euromicro international conference on parallel, distributed and network-based processing (PDP), IEEE, 2010, pp 317–324
8. Cheung B, Sable C (2011) Hybrid evolution of convolutional networks. In: 10th International conference on machine learning and applications and workshops (ICMLA), IEEE, 2011, vol 1, pp 293–297
9. Wu C, Fan W, He Y, et al. (2012) Cascaded heterogeneous convolutional neural networks for handwritten digit recognition. In: 21st International conference on pattern recognition (ICPR), IEEE, 2012, pp 657–660
10. Sermanet P, LeCun Y (2011) Traffic sign recognition with multi-scale convolutional networks. In: The 2011 international joint conference on neural networks (IJCNN), IEEE, pp 2809–2813
11. Chen X, Xiang S, Liu CL et al (2014) Vehicle detection in satellite images by hybrid deep convolutional neural networks. IEEE Geosci Remote Sens Lett 11(10):1797–1801
12. Gu JL, Peng HJ (2009) Incremental convolution neural network and its application in face detection. J Syst Simul 21(8):2441–2445
13. Saito J H, de Carvalho T V, Hirakuri M, et al. (2005) Using CMU PIE human face database to a convolutional neural network-neocognitron. ESANN. pp 491–496