Multiple Convolutional Neural Network for Feature Extraction

Guo-Wei Yang[1,2] and Hui-Fang Jing[1(✉)]

[1] College of Information Engineering,
Nanchang Hangkong University, Nanchang 330063, China
{ygw_ustb, jacellyl3l4}@163.com
[2] School of Technology, Nanjing Audit University, Nanjing 211815, China

Abstract. Recent theoretical studies indicate that Deep Neural Network has been applied to many image processing tasks. However, learning in deep architectures is still difficult. One of the neural network, Convolutional Neural Network (CNN) has gained great success in image recognition and it builds features by automatic learning. More importantly, CNN can operate directly on the gray image, so it can be directly used for processing classification of the image. In order to utilize CNN to recognize plant leaf, a hierarchical model based on CNN is proposed in this paper. We firstly do some pre-processing, such as illumination changes, rotation and leaf distortion. After that, we applied the method of CNN to extract the features of leaves pictures. One focus on our network is about the depth of CNN, which affects the ability of capability of convolution. Thus, we try our best to choose the best depth of CNN with several experiments. Moreover, in order to destroy the symmetry of networks, the strategies used in this paper is to add a mathematical formula for feature map connection between convolutional layer and sampling layer. The experimental results show that the proposed method is quite effective and feasible. And we also applied other classification methods to the ICL dataset. By contrast, our classification is much better than other methods.

Keywords: Classification · CNN · Image recognition · Plant leaf classification · Feature extraction

1 Introduction

Image Classification is an active research topic in computer vision and pattern recognition. In addition, it has been used in many applications, such as face recognition, biomedical image analysis. Traditional method depends largely on the choice of the characteristics, and the researchers tend to choose the characteristics, which is blind in some extent. In order to solve the above problems, this paper proposes a Deep Learning (DL) method that can learn features automatically on two-dimensional image.

The concept stems from the *Deep Learning* of artificial networks, including multi-layer, multi-hidden Layer Perceptron (MLP) is a kind of *Deep Learning* structure. Deep structure of local (non-linear processing involves multiple cell layers) non-convex objective minimum cost function is prevalent mainly training difficult.

D.-S. Huang et al. (Eds.): ICIC 2015, Part II, LNCS 9226, pp. 104–114, 2015.
DOI: 10.1007/978-3-319-22186-1_10

Before 2006, verities of machine learning method has been applied to all kinds of classification tasks. Such as Support Vector Machine (SVM), Boosting and Decision Tree have gained great success in the classification task [1, 2]. Many scholars focus on SVM and have got state-of-the-art results [3]. However, deep neural network re-boomed, when Hinton and Salakhutdinov proposed an efficient way to train deep architecture through layer-wised pre-training [4] for solving optimization problems associated with the deep structure of hope, and then proposed a multi-layer automatic encoder deep structure. *Deep learning* have great advantage in feature extracting, because they have better designs for modeling and training and get much more complex features than shallow architectures [5, 6]. In 1980, CNN was first proposed by Japanese scholars Fukushima, then Yann LeCun, et al. got further improve and perfect [7], which uses spatial relationship relative reduction in the number of training parameters to improve the performance of BP. In 2003, Sven Behnke conducted generalization, the same year, Patrice Simard, et al. made further simplified [8]. After Dan Ciresan, et al. further optimizing in 2011 and 2012, they created the best results at the time of recognition in multiple image databases (MNIST, NORB, HWDB1.0, CIFAR10) with CNN [9, 10]. Moreover, depth study also appeared in many deformed structures such as DAE, DCN and so on.

In this paper, we propose a Convolutional Neural Network (CNN) to solve image recognition problem, and this is a difficult problem because of lighting conditions, and the images are usually noisy or broken. We applied to extract plant leaf features and then classify these features. Then we analyzed the results of this process and the results prove the performance surpass pure other classification method.

The rest of the article is organized as follows. First, we describe the classical architecture about CNN and other approaches on Sect. 2. We then detail our CNN architecture, followed by experiments that evaluate our method. Finally, we conclude the whole paper with several discussions of future works.

2 Related Works

2.1 Convolutional Neural Network Architecture

Convolutional Neural Network is specifically designed to deal with a class of two-dimensional data of the multi-layer neural network. CNN is one of the important algorithm for *Deep Learning*, because it can be reached prior to the dissemination of improving network efficiency BP by reducing the number of network-related spatial data mining on the training parameters and requires very little data pre-processing. And CNN shows excellent results in many applications. In [7], it contrasts multiple recognition algorithm, the conclusion is superior to all over CNN algorithms. CNN is improved by BP neural network and similar to BP. Moreover, they both have adopted a forward propagation to calculate the output value, back propagation adjust the weights and bias. The biggest difference between CNN and standard is that the neurons of CNN between adjacent layers is not fully connected, but part of the connection, which is the perception of a nerve cell area from the upper part of the nerve cells, rather than BP, as connected with all the neurons.

CNN has three important ideological framework: local perception of the area, weights sharing, down-sampling space or time domain, which makes the network to achieve a relative displacement, scaling and distortion characteristics have remained unchanged. To reflect these three basic concepts, CNN will integrate feature extraction capabilities into a Multilayer Perceptron (MLPs) through restructuring or reducing weights. From [11], it is easy to understand that the right values and the required number of training data in close contract. With weights sharing, it is more similar to biological network, which can reduce the complexity of the network model [12].

CNN in close contact and spatial information makes it particularly suitable for image processing and understanding, and can automatically extracted the wealth of relevant characteristics from the image. In image and voice recognition tasks, multi-layer back-propagation neural network training network can learn complex, multi-dimensional, non-linear classification of surfaces from a large number of training examples [13].

2.2 Basic Framework for CNN

CNN is a multi-layer neural network, which is composed of multiple convolution layer and sub-sampling layer. Moreover, each layer is composed of multiple independent neurons. It basic framework is shown in Fig. 1. Basic principle and derivation can also be learned [14] simply. CNN consists of three different types of layers, convolution layer, sub-sampling (optional), and a fully connected layer. Convolution layer is responsible for feature extraction, which uses two key concepts: weights sharing and field accepting. The sub-sampling layer performs local averaging and sub-sampling, reducing the sensitivity of the offset and distortion. What full connectivity layer do is to perform a normal classification just like a traditional MLP networks.

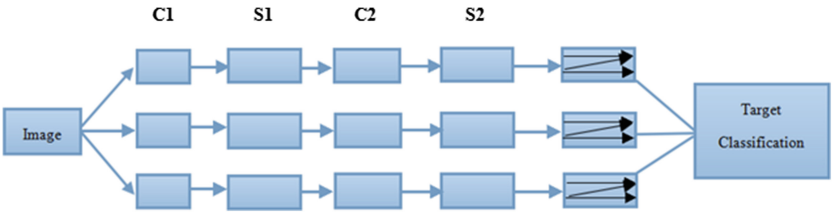

Fig. 1. Basic framework for CNN

Figure 1 shows that CNN consists of two layers, composed of two sub-sampling alternately. C layer is identified as Convolutional layer, also referred to as feature extraction layer. The input of each neuron is connected with a local receptive field of previous layer and it extracts the local feature. C layer has many different two-dimensional feature maps, a characteristic diagram extracting a characteristic, which means that can extract various features. When extracting features, the weight of the same feature map are shared. Even if the same convolution kernel, different feature maps use different convolution kernel. C layer preserved different local features, so that the extracted features have a rotation and translation invariance.

S layer identifies the sub-sampling level, also called characteristic mapping layer, which is responsible for the characteristic C layer obtained sub-sampling, so that the extracted features having scaling invariance. S layer only do a simple scaling layer mapping, relatively few neurons weights, which is also simply calculation. In general, convolution and sub-sampling layer appear assembly. But [8] indicates that convolution and sampling layer can be produced through a convolution layer embodiment convolution step size which is equal to embodiment 1 of the sub-sampling layer removing. Convolution is equivalent to one layer of the two functions previously without affecting the performance of CNN, so sub-sampling layer is optional layer.

In the CNN end layer is generally connected to one or several full connection, the final output of the number of nodes is classified target number. Moreover, the purpose of training is to make the output of CNN as close as possible to the original label. The layer is always in use in CNN, its role is similar to the standard MLP. After all the connections in the hidden layer and output layer information conversion and calculation process is complete forward propagation of a learning process, the final result output to the outside by the output layer.

In the learning process, the update of the CNN weights is based on back-propagation algorithm (B-P) [15], which update the weights ϖ of neurons by the time t to t + 1.

$$\varpi(t+1) = \varpi(t) + \eta\delta(t)\chi(t) \tag{1}$$

Here, η is the learning rate, $\chi(t)$ is the input to neurons. And $\delta(t)$ is the error term of neurons, the error term for the output neurons and hidden neurons have different expression. It should be noted, though generally increasing with the number of iterations the learning rate decreasing, but usually, the learning rate is the same for all weights η.

Results of sampling layer generates down-sampling enter after. If there have N input maps, they also have N output maps, although output map is smaller than input map.

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

Here, down() represents down-sampling function. Difficulty here is that the calculation of the error signal in FIG. Assume down-sampling layer and the next layer is a layer of convolutional layer. If you fall behind all sampling layer fully connected network, then the error signal diagram can be obtained directly by the back-propagation algorithm.

2.3 Compared with Other Approaches

The easiest way of Deep Learning is to take advantage of the characteristics of artificial neural networks. Artificial Neural Network (ANN) have a system hierarchy itself, if given a neural network, we assume that the output and the input are the same. Then we train adjust its parameters, obtaining in each layer weights. Bourlard and Kamp [16] first proposed auto-association in 1988, to achieve this reproduction, the automatic

encoder is necessary to capture the most input of the input data, so as PCA, to find the main component that may represent the original information. AE can be simply learned by error back propagation algorithm [17]. Auto-encoder (AE) is a neural network as much as possible to reproduce the input signal. AE neural is an unsupervised learning algorithm that applies back propagation, setting the output equal to the input. It directly encodes the input layer and then decodes the hidden layer. The aim of an AE is to learn a compressed or sparse representation (encoding) for a set of data.

Auto-encoder has some variants, here briefly the next two. If we add the L1 Regularity Auto-encoder limit on the basis (L1 mainly constrained nodes in each layer as 0, most have only a few non-zero, this is the Sparse source of the name), we can get Sparse Auto-encoder. In fact, every expression was to limit as much as possible sparse code each time.

Automatic noise reduction encoder DA is based on automatic encoder, the training data is added noise, so automatic encoder must learn to eliminate this noise and get real noise pollution had not been entered [18, 19]. Therefore, it forced the encoder to learn more robust expression of the input signal, which is than encoder.

There are some other methods of Deep Neural Network, here is not outlined. By other means, we summarize some of the advantages of CNN. CNN is mainly used to identify displacement, scaling and other forms of distortion invariant two-dimensional graphics. Because of CNN's feature detection layer lean through training data, when using CNN, avoiding explicit feature extraction, and implicitly learn from training data. Also due to the same neuron weights, the same surface feature mapping can be a parallel study, which is the convolution of the network with respect to a big advantage neuronal networks connected to each other. Convolutional Neural Network with its special structure has a local weights sharing in speech recognition and image processing unique advantages, its layout closer to the actual biological neural networks, the weights sharing reduces the complexity of the network, in particular multi-dimensional input vector images can be directly input network feature that avoids the process of feature extraction and classification data reconstruction complexity.

Convolutional neural network, it avoids the explicit characteristics of sampling, implicitly learning from the training data. This makes it significantly different from other convolution neural network classifier based on neural networks, through restructuring and reducing weights will feature extraction capabilities integrated into a multi-layer perceptron. It can directly handle grayscale images, can be directly used for processing image-based classification.

CNN more general neural network has the following advantages in image processing: First, inputting image and the network topology can be a good fit. Second, feature extraction and pattern classification is at the same time, and at the same time producing in training. Third, weight sharing can reduce training parameters of the network, so that the neural network structure becomes more simple and adaptable.

The close relationship between this layer and the airspace contact information of CNN is making it suitable for image processing and understanding. Moreover, its salient features automatic extraction of images side also showed a relatively better performance.

3 The Architecture Used in This Paper

First, we assume that the input gray-scale image is an image of the blades 64 × 64. Plant leaves frame identification of CNN based on this design is shown in Fig. 2.

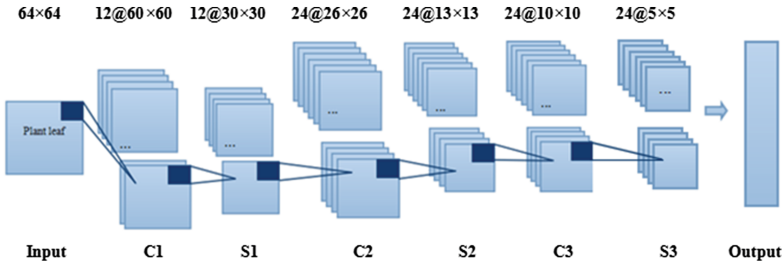

Fig. 2. Plant leaf identification framework

Input. We first do pre-processing for our leaves image as the input, it includes gradation processing, filtering process and gray-scale. When the image size is not 64 × 64, we use bilinear difference computation to scale image. Thus, it can meet the requirements of a standard input.

C1 Layer. C1 is a feature extraction layer, and its input is either from the input layer or sampling layer. In our method, C1 gets 12 two-dimensional feature maps for 60 size. In fact, it is obtained by a 5 × 5 convolution kernel. Each map of a convolution layer has the same size as the convolution kernel. When the convolution kernel is too small, it cannot be effectively extracted local feature, and when the convolution kernel is too large, the complexity extraction features may far exceed the ability of the convolution kernel represents. Therefore, it is important to set the appropriate convolution kernel to improve the performance of CNN, and it is also the difficulty to tune parameters for CNN. We use a 5 × 5 convolution kernel in the image. When rolling again, we finally get a (64–5 + 1) × (64–5 + 1) feature map. Convolution kernel move one step every time. Each features of convolution layer is a layer different from the convolution kernel on the front of all map elements and corresponding accumulated a bias, and then it obtained by seeking Sigmoid. In practice, it plus an offset term when convolution. For image χ, it uses convolution kernel ϖ convolution, offset term b. The output y convolution is:

$$y = Sigmoid(\varpi\chi + b) \tag{3}$$

S1 Layer. S1 is a sub-sampling layer, it gets 12 feature maps for 30 × 30 size. Sub-sampling layer is a layer of a map of the sampling handling, what sampling model here is a small area on one of the adjacent map of the statistical aggregation. Moreover, it is get maximum value of the small region. In this paper, it sums by all non-overlapping sub-block χ of 2 × 2 in C1. Then it multiplies by weights ϖ and pluses an offset b. The calculation of sub-sampling is:

$$y = Sigmoid(\varpi \cdot \underset{x_i \in x}{sum}(\chi_i) + b) \tag{4}$$

Because the feature map size of C1 is 60 × 60, the final result for sub-sampling is 30 × 30 feature map. In CNN, the general scaling factor is 2. The reason for this is to control the zoom rate of decline, since scaling is scaling exponent. Reduced too fast also means more rough image feature extraction, and image detail will lose more features.

C2 Layer. C2 is also a feature extraction layer. It has a similar place with C1, but also some differences. Because of C1 and S1 processing, virtually every neuron receptive field of S1 has been the equivalent of the original image 10 × 10. When making convolution, several or all feature maps by S1 are combined into an input, and the do the convolution. The reason why not always all of the features as input S1 is that incomplete connection mechanism allows the connection number keeping within a reasonable range. At the same time, the most important is destroying symmetry networks. Strategies used in this paper is that the feature map of S1 is connected to all that can be divisible by 1 feature maps. That is to say all of the feature maps of C2. The feature map 2 of S1 is connected to all even-numbered feature maps. The following is connected through this method (Fig. 3). Here, 'o' indicates a connection.

	0	1	2	3	4	5	6	7
0	o							
1	o	o						
2	o		o					
3	o	o		o				
4	o				o			
5	o	o	o			o		
6	o						o	
7	o	o		o				o
8	o		o					
9	o	o			o			
10	o							
11	o	o	o	o		o		

Fig. 3. The connection between C2 layer and S1 layer

The remaining layers of convolution and sub-sampling do the same thing as previous layers. Only as the depth increases, the extracted features is more abstract and more expressive capability. That is the depth meaning of Deep Neural Network. By C1, S1, C2, S2, the extracted feature have the ability to express. But the correct accuracy rates is low. It shows that the depth affects the ability of CNN.

4 Experiments

4.1 Data Set and Experiment Setup

The first step in identifying the image processing matter blade, blade pretreated by contour information calculation, obtaining the identification of the blade eigenvectors.

Here, we used the leaves images from ICL dataset. For more information to this database, please infer to [20].

First, we pick and choose those typical leaf shape and full leaf (Fig. 4), then do pre-processing for leaves image. Our image processing including gradation processing, filtering process. First, in order to remove the interference color, we go to the image color processing, the color difference in the background by drinking blade, converting to the grayscale image. Second, the filtering processing, the image of the blade after filtering operation to remove the blade internal hole and the surrounding noise such as small debris, making it more suitable for contour extraction.

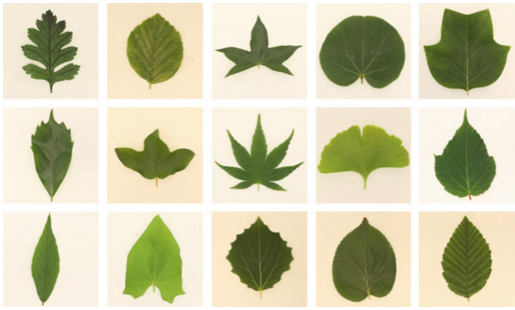

Fig. 4. The typical leaves of ICL

The original data set contains some leaf petiole, which may affect the robustness of classification algorithm. We transform those pictures into 64 × 64 gray pixels. Because the acquisition of the blade may result in the acquisition of the operator petioles of varying lengths, the information it provides cannot be used as basis for classification. So before making classification, we first remove the petiole.

Because convolutional neural network in Fig. 2 only accept gray-scale images for 64 × 64 size, the ICL database is high-definition color images. After above pre-processing, we then change the picture size with bilinear interpolation geometric. Finally, we put the processing data to our Convolutional Neural Network.

The choice of CNN layers is also a problem. We know the number of parameters is the convolution kernel and offset + sampling layer scale factor and offset + full connectivity layer weights and bias, and the number of parameters entire link layer is more than two orders of magnitude than the number of arguments before the four-story.

Another innovation in this paper is to destroy the symmetry of networks, our strategies here used is adding a mathematical formula for the feature map connection between convolutional layer and the sampling layer. We can see the connection between C2 and S1 in Fig. 3.

In our experiment, our method calls CNN-1, and the most important parameter is the size of mini-batch. In this paper, the mini batch setting as 1 seem to take the fastest time of convergence. Random samples of training sequence, then, each gradient produced is random. There is also an argument learning rate. In the experiment, increasing the situation does not move mini-batch training will appear appropriate to increase the

learning rate that can make training cost to decline. The parameters used in our experiment are in Table 1.

During this experiment, our experimental environment is shown in Table 2.

Table 1. The parameters of the architecture

Method	Activation function	Batch size	Learning rate	Alpha	Kernel size	Scale
CNN-1	Sigmoid	100	0.01	1	5	2 × 2

Table 2. The experimental environment

Hardware environment	Software environment
CPU: Intel Core i3-2370	Operating System: Windows 64-bit
RAM: 8G	Development Tool: Matlab 2012b

4.2 Results on the ICL Database

In order to compare the performance of CNN-1 algorithm proposed in this paper, 10 classes of Table 3 report the resulting classification on the best set for new model. We also took another 15 kinds of random leaf data from ICL with the same CNN-1. As can be seen in the table, our training works remarkably well, and in most cases yield significantly better classification performance than SVM.

Table 3. The comparison of the accuracy

Method	10 Classes	15 Classes
SVM	80 %	77 %
CNN-1	95 %	92 %

In these experiments, we can observe three different topics, which is showed by feature extraction obviously. First, the input layer of the data is the better the more normalized. For image processing, such as scaling and graying, is on the back of a great influence classification results. Second, right convolutional neural network values shared not only significantly reducing the number of required training of weights, greatly reducing the amount of training data required, but also reduced the complexity of neural convolutional neural network, reducing the over-fitting problem. Network layer choice is also important, great depth on CNN performance, inadequate depth will weaken CNN feature extraction capability. In order to break the symmetry, from the first layer to the second convolution layer is not full link layer. They can automatically learn the combination coefficient or human requirements. Finally, the choice of the classifier is not discussed as an important point of this article.

In conclusion, Convolutional Neural Network gives a better performance than the state-of-the-art classification method in our experimental data.

5 Conclusions and Future Works

For the experiment above, we have introduced a very simple training principle for CNN, and we find that CNN have better performance on feature extraction. Traditional artificial leaf extracting feature recognition method based on experience from the blade, and then on the classification features, blindness and low defect classification accuracy. This principle can be used to train and stack features to initialize a deep neural network. A series of image classification experiments were performed to evaluate this new training principle.

Moreover, on a conceptual prospective, CNN is just a Deep Neural Network method to classify images. Other classification method can also achieve classification task. The experimental results with SVM suggest that SVM may also encapsulate a form of robustness in the representations they learn, possibly because of their stochastic nature, which introduces noise in the representation during training.

Although CNN give us a better resolution on feature extraction, there are some problems about CNN architecture we should notice. So for the future work, we will try to identify the blade-based parallelization framework of CNN in order to accelerate the training speed. We also want to play some other more interesting things, such as dropout, max-out, max pooling. At the same time, we want to collect more leaves data to train CNN, so that it can identify more plants. Finally, we hope to get a simple method, widely used, the speedup satisfactory results.

Acknowledgments. The authors would like to sincerely thank the Institute of Machine Learning and Systems Biology of Tongji University and Professor Guo-Wei Yang (No.61272077).

References

1. Cai, C.Z., et al.: SVM-Prot: web-based support vector machine software for functional classification of a protein from its primary sequence. Nucleic Acids Res. **31**(13), 3692–3697 (2003)
2. Christian, S., Laptev, I., Caputo, B.: Recognizing human actions: a local SVM approach. In: Proceedings of the 17th International Conference on Pattern Recognition, 2004, ICPR 2004. vol. 3, IEEE (2004)
3. Mavroforakis, M.E., Theodoridis, S.: A geometric approach to support vector machine (SVM) classification. IEEE Trans. Neural Netw. **17**(3), 671–682 (2006)
4. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science **313**(5786), 504–507 (2006)
5. Bengio, Y.: Learning deep architectures for AI. Found. Trends Mach. Learn. **2**(1), 1–127 (2009)
6. Bengio, Y., Courville, A., Vincent, P.: Representation learning: a review and new perspectives, 1–1 (2013)
7. Lecun, Y., Bottou, L., Bengio, Y.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
8. Simard, P.Y., Steinkraus, D., Platt, J.C.: Best practice for convolutional neural networks applied to visual document analysis. In: ICDAR, pp. 958–962. IEEE, Los Alamitos (2003)

9. Cires, D.C., Meier, U., Masci, J.: Flexible, high performance convolutional neural networks for image classification. In: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, vol. 2, pp. 1237–1242 (2011)
10. Cires, D.C., Meier, U., Schmidhuber, et al.: Multi-column deep neural networks for image classification. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 3642–3649, New York (2012)
11. Baum, E., Haussler, D.: What size net gives valid generalization (J). Neural Comput. **1**, 151–160 (1989)
12. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet Classification with Deep Convolutional Neural Networks. In: NIPS (2012)
13. Duda, R., Hard, P., Stork, D.: Pattern Recognition, 2nd edn. Wiley-Interscience, New York (2000)
14. Behnke, S.: Hierarchical Neural Networks for Image Interpretation [M], 1(4), 541–551 (1989)
15. Lippmann, R.: An introduction to computing with neural nets [J]. IEEE ASSP Magazine **4**, 22 (1987)
16. Bourlard, H., Kamp, Y.: Auto-association by multilayer perceptrons and singular value decomposition. Biol. Cybern. **59**(4–5), 291–294 (1988)
17. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. Nature **323**(6088), 533–536 (1986)
18. Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.-A.: Extracting and composing robust features with denoising AEs. In: ICML (2008)
19. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.-A.: Stackeddenoising AEs: learning useful representations in a deep network with a local denoising criterion. J. Mach. Learn. Res. **11**, 3371–3408 (2010)
20. http://mlsbl.tongji.edu.cn/chinese/file-database.asp
21. Li, B., Huang, D.S.: Locally linear discriminant embedding: an efficient method for face recognition. Pattern Recogn. **41**(12), 3813–3821 (2008)
22. Huang, D.S., Du, J.-X.: A constructive hybrid structure optimization methodology for radial basis probabilistic neural networks. IEEE Trans. Neural Netw. **19**(12), 2099–2115 (2008)
23. Huang, D.S., Horace, H.S.Ip, Chi, Z.-R.: A neural root finder of polynomials based on root moments. Neural Comput. **16**(8), 1721–1762 (2004)
24. Huang, D.S., Jiang, W.: A general CPL-AdS methodology for fixing dynamic parameters in dual environments. IEEE Trans. Syst. Man Cybern. Part B **42**(5), 1489–1500 (2012)
25. Huang, D.S.: Systematic Theory of Neural Networks for Pattern Recognition (in Chinese). Publishing House of Electronic Industry of China, Beijing (1996)
26. Huang, D.S.: Radial basis probabilistic neural networks: model and application. Int. J. Pattern Recogn. Artif. Intell. **13**(7), 1083–1101 (1999)
27. Wang, X.-F., Huang, D.S.: A novel density-based clustering framework by using level set method. IEEE Trans. Knowl. Data Eng. **21**(11), 1515–1531 (2009)