

# Leaf Recognition for Plant Classification Based on Wavelet Entropy and Back Propagation Neural Network

Meng-Meng Yang<sup>1</sup>, Preetha Phillips<sup>2,3</sup>, Shuihua Wang<sup>1,4</sup>, and Yudong Zhang<sup>1,5</sup>(✉)

<sup>1</sup> School of Computer Science and Technology, Nanjing Normal University, Nanjing 210023, Jiangsu, China

yudongzhang@ieee.org, zhangyudong@njnu.edu.cn

<sup>2</sup> School of Natural Sciences and Mathematics, Shepherd University, Shepherdstown, WV 25443, USA

<sup>3</sup> West Virginia School of Osteopathic Medicine, 400 N Lee St., Lewisburg, WV 24901, USA

<sup>4</sup> Department of Electrical Engineering, The City College of New York, CUNY, New York, NY 10031, USA

<sup>5</sup> Jiangsu Key Laboratory of Advanced Manufacturing Technology, Huaiyin 223003, Jiangsu, China

**Abstract.** In this paper, we proposed a method for plant classification, which aims to recognize the type of leaves from a set of image instances captured from same viewpoints. Firstly, for feature extraction, this paper adopted the 2-level wavelet transform and obtained in total 7 features. Secondly, the leaves were automatically recognized and classified by Back-Propagation neural network (BPNN). Meanwhile, we employed K-fold cross-validation to test the correctness of the algorithm. The accuracy of our method achieves 90.0%. Further, by comparing with other methods, our method arrives at the highest accuracy.

**Keywords:** Feature extraction · Classification · Back-Propagation · K-fold cross-validation · Pattern recognition

## 1 Introduction

With acceleration of the extinction rate of plant species, it is essential urgent to protect plants [1, 2]. A preliminary task is the classification of plant type, which is challenging, complex and time-consuming. As we all know, plants includes the flowering plants, conifers and other gymnosperms and so on. Most of them do not blooming and have fruit, but almost all of them contain leaves [3]. Therefore, in this paper we focus on feature extraction and classification of leaves.

Recent studies are analyzed below: Heymans, Onema and Kuti [4] proposed a neural network to distinguish different leaf-forms of the opuntia species. Wu, Bao, Xu, Wang, Chang and Xiang [5] employed probabilistic neural network (PNN) for leaf recognition system. Wang, Huang, Du, Xu and Heutte [6] classified plant leaf images with complicated background. Jeatrakul and Wong [7] introduced Back Propagation Neural Network (BPNN), Radial Basis Function Neural Network (RBFNN), Probabilistic Neural Network (PNN) and compared the performances of them. Dyrmann, Karstoft and Midtiby [8] used convolutional neural network (CNN) to classify plant species.

Zhang, Lei, Zhang and Hu [9] employed semi-supervised orthogonal discriminant projection for plant leaf classification.

Although these methods have achieved good results, ANN has the higher accuracy and less time-consuming in classification than other approaches. Therefore, in this paper, we employ the BPNN algorithm on the leaves of the classification of automatic identification.

Our contribution in this paper includes: (i) We proposed a five-step preprocessing method, which can remove unrelated information of the leaf image. (ii) We developed a leaf recognition system.

## 2 Methodology

### 2.1 Pretreatment

We put a sheet of glass over the leaves, so as to unbend the curved leaves. We pictured all the leaves indoor (put the leaves on the white paper) using a digital camera with (Canon EOS 70D) by two cameraman with experience over five years. The pose of the camera is fixed on the tripod during the imaging. Two light-emitting diode (LED) lights are hanged 6 inches over the leaves. Those images out of focus are removed.

### 2.2 Image Preprocessing

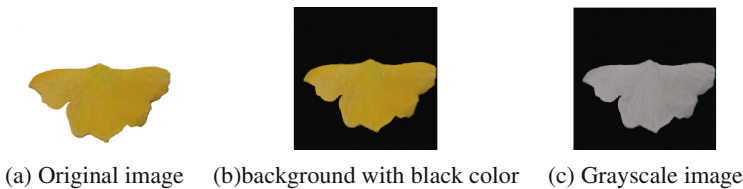
During the imaging, the background information is captured, which will inference the detection of the leaves. Therefore, the necessary preprocessing is to remove the irrelevant background and color channels.

First, we suppose texture is related to leave category, and color information is of little help. Hence, we removed the background of the image. Second, we convert the RGB image into a grayscale image. Table 1 shows the steps of image preprocessing.

**Table 1.** Steps of image preprocessing

Step 1: Segment the leaves manually
Step 2: Crop the left and right margins, to form a rectangular image
Step 3: Resize to the size of [200, 200].
Step 4: Fill the background with black color (value = 0)
Step 5: Convert RGB to gray-level image

Figure 1(a) shows the original image. Figure 1(b) shows the image of background with black color. Figure 1(c) shows the grayscale image



**Fig. 1.** Illustration of leaves

### 2.3 Feature Extraction

Fourier transform (FT) [10] is a method of signal analysis, which decomposes the continuous signal into harmonic waves with different frequency [11]. It can more easily deal with the original signal than traditional methods. However, it has limitations for non-stationary processes, it can only get a signal which generally contains the frequency, but the moment of each frequency is unknown. A simple and feasible handling method is to add a window. The entire time domain process is decomposed into a number of small process with equal length, and each process is approximately smooth, then employing the FT can help us to get when each frequency appears. We call it as Short-time Fourier Transform (STFT) [12, 13].

The drawback of STFT is that we do not know how to determine the size of windowing function. If the window size is small, the time resolution will be good and the frequency resolution will be poor. On the contrary, if the window size is large, the time resolution will be poor and the frequency resolution will be good.

Wavelet transform (WT) [14–16] has been hailed as a microscope of signal processing and analysis. It can analyze non-stationary signal and extract the local characteristics of signal. In addition, wavelet transform has the adaptability for signal in signal processing and analysis [17], therefore, it is a new method of information processing which is superior to the FT and STFT.

In this paper, the feature extraction [18, 19] of the original leaf images is carried out by 2-level wavelet transform, which it decompose the leaves images with the low-frequency and high-frequency coefficients. However, a massive of features not only increase the cost of computing but also consume much storage memory does little to classification [20]. We need to take some measures to select important feature. Entropy [21, 22] is used to measure the amount of information of whole system in the information theory and also could represent the texture of the image.

$$\begin{aligned} \begin{bmatrix} X \\ p(x) \end{bmatrix} &= \begin{bmatrix} x_1 & x_2 & x_3 & \dots & x_n & x_{n+1} \\ p_1 & p_2 & p_3 & \dots & p_n & p_{n+1} \end{bmatrix}, \\ 0 \leq p_i \leq 1, \sum_{i=1}^{n+1} p_i &= 1 \end{aligned} \tag{1}$$

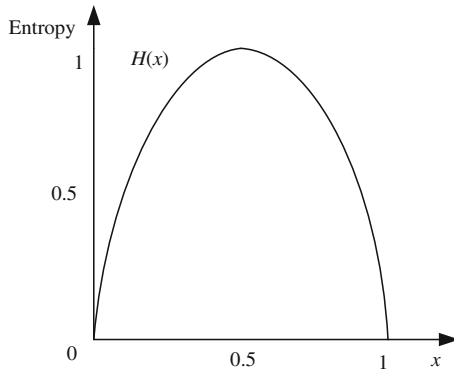
where  $X$  is a discrete and random variable,  $p(x)$  is the probability mass function, the amount of information contained in a message signal  $x_i$  can be expressed as

$$I(x_i) = -\log p_i \tag{2}$$

where  $I(x_i)$  is a random variable, it can't be used as information measure for the entire source [23], Shannon, the originator of modern information theory, defines the average information content of  $X$  as information entropy [24, 25]:

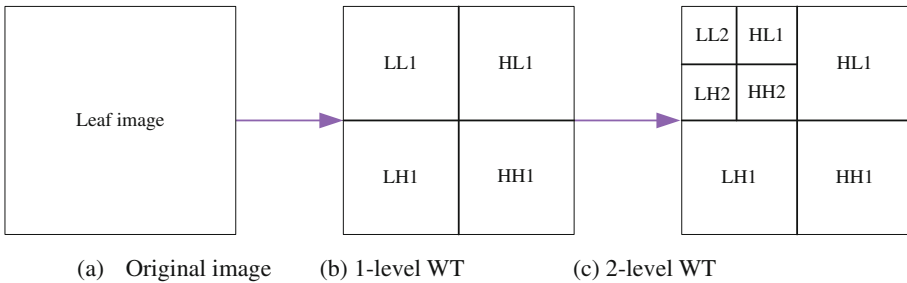
$$H(X) = E[I(x_i)] = -\sum p_i \log p_i \tag{3}$$

where  $E$  is the expected value operator,  $H$  represents entropy. Figure 2 shows the entropy of source.



**Fig. 2.** The entropy of source

There are seven comprehensive indexes (as shown in Fig. 3) are obtained after adopting 2-level WT, with four in size of  $50 \times 50$  and three  $100 \times 100$ . The entropy of these matrices is calculated as the input to the following BP neural network (BP).



**Fig. 3.** 2-level WT

### 2.4 Back-Propagation Neural Network

Back Propagation algorithm is mainly used for regression and classification. It is one of the most widely used training neural networks model. Figure 4 represents the diagram of BP, the number of nodes in the input layer and output layer is determined but the number of nodes in the hidden layer is uncertain. There is an empirical formula can help to determine the number of hidden layer nodes, as follow

$$t = \sqrt{r + s} + c \tag{4}$$

where  $t, r, s$  represents the number of nodes in the hidden layer, input layer, and output layer, respectively.  $c$  is an adjustment constant between zero and ten.

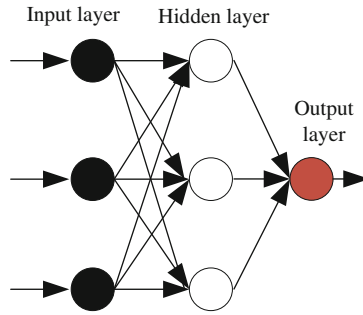


Fig. 4. The diagram of BP

In our experiment, the features of leaves were reduced to seven after feature selection, i.e.  $r = 7$ . Meanwhile, there is one output unit, which stands for the predictable result, so  $s = 1$ . Because  $c$  is constant, according to formula (4), we set  $t = 15$ .

BPNN algorithm is part of a supervised learning method. The following is the main ideas of the BP algorithm learning rule:

Known vectors: input learning samples  $\{P^1, P^2, \dots, P^q\}$ , the corresponding output samples  $\{T^1, T^2, \dots, T^q\}$ .

Learning objectives: The weights are modified with the error between the target vector  $\{T^1, T^2, \dots, T^q\}$  and the actual output of the network  $\{A^1, A^2, \dots, A^q\}$  in order to  $A^i$  ( $i = 1, 2, \dots, q$ ) is as close as possible to the expected  $T^i$ , i.e., error sum of squares of the network output layer is minimized.

BPNN algorithm has two parts: the forward transfer of working signal and error of the reverse transfer. In the forward propagation process, the state of each layer of neurons only affects the state of the next layer of neurons.

$$x_j = \sum_i w_{ij} P^i \tag{5}$$

where  $x_j$  represents the input of hidden layer.  $w_{ij}$  is the weight between input layer and the hidden layer.

$$x'_j = f(x_j) = 1/(1 + e^{(-x_j)}) \tag{6}$$

Equation (6) indicates the function of hidden layer.

$$A^i = \sum_j w_{jk} x'_j \tag{7}$$

$A^i$  is the actual output of output layer.

$$e = \frac{1}{2} \sum_i (A^i - T^i)^2 \tag{8}$$

where  $T^i$  is the desired output.

If the desired output  $T^i$  is not obtained at the output layer, we will use the Eq. (8) and the error value will be calculated, then the error of the reverse transfer is carried out.

### 2.5 Implementation of the Proposed Method

The aim of this study was to distinguish the type of leaves with high classification accuracy. The steps of proposed system are sample collection, pretreatment, preprocessing, feature extraction, classifier classification (Fig. 5). Table 2 shows the pseudocode of the proposed system.

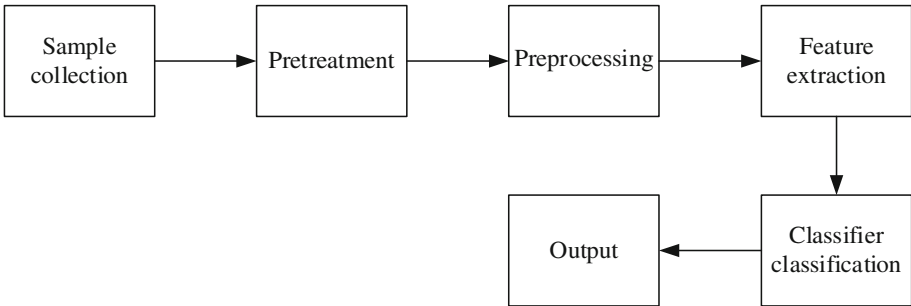


Fig. 5. Pipeline of our proposed method

Table 2. Pseudocode of the proposed system

Proposed Approach
<p><b>Input:</b> grayscale image.</p> <p><b>Parameter:</b> <math>NIM</math>, total image number. <math>NK</math>, number of k-fold cross validation.</p> <p><b>Phase1.</b> 2-level WT.</p> <p>For <math>i=1:NIM</math></p> <p style="padding-left: 20px;">Read in the image file.</p> <p style="padding-left: 20px;">A matrix <math>I[200 \times 200 \times NIM]</math> is applied to store all grayscale images</p> <p style="padding-left: 20px;">Employ the WT to extract 7 coefficients.</p> <p style="padding-left: 20px;">A matrix <math>out[1 \times NIM]</math> is employed to store the classification of leaves</p> <p>End</p> <p><b>Phase2.</b> Extract entropy from the grayscale image</p> <p>For <math>j=1:NIM</math></p> <p style="padding-left: 20px;">Extract entropy from matrix <math>I[200 \times 200 \times NIM]</math>.</p> <p style="padding-left: 20px;">Put the entropy into a matrix <math>M[7 \times NIM]</math>.</p> <p>End</p> <p><b>Phase3.</b> BP Classification using <math>5 \times 5</math> cross validation.</p> <p>Divide the input data <math>M[7 \times NIM]</math> and target data <math>out[1 \times NIM]</math> into five different groups randomly.</p> <p>For <math>i=1:NK</math></p> <p style="padding-left: 20px;">Regard the <math>i</math>th group as test, and other 4 groups to train the BPNN.</p> <p style="padding-left: 20px;">Output the classification results.</p> <p>End</p> <p>Calculate the average accuracy.</p>

## 2.6 Five-Fold Cross Validation

The training criterion of the standard BP neural network is to require that the error sum of squares (or the fitting error) of the expected and output values of all samples is less than a given allowable error  $\varepsilon$ . In general, the smaller the value of  $\varepsilon$  is, the better the fitting accuracy. Nevertheless, for the actual application, the prediction error decreases with the decrease of the fitting error at the first. However, when the fitting error on training set decreases to a certain value, the prediction error on test set increases, which indicates that the generalization ability decreases. This is the “over-fitting” phenomenon which the BPNN modeling encountered. In this paper, cross-validation is used to prevent over-fitting.

The basic of the cross validation method is that the original dataset of the neural network were divided into two parts: the training set and the validation set. First of all, the classifier was trained by training set, and then test the training model using the validation set, through the above results to evaluate the performance of the classifier.

The steps of K-fold cross validation:

- Step 1. All training set  $S$  is divided into  $k$  disjoint subsets, suppose the number of training examples in  $S$  is  $m$ , then the training examples of each subset is  $m/k$ , the corresponding subset is called  $\{s_1, s_2, \dots, s_k\}$ .
- Step 2. A subset is selected as test set from subset  $\{s_1, s_2, \dots, s_k\}$  each time, and the other  $(k - 1)$  as the training set.
- Step 3. Training model or hypothesis function is obtained through training.
- Step 4. Put the model into the test set and gain the classification rate.
- Step 5. Calculate the average classification rate of the  $k$ -times and regard average as the true classification rate of the model or hypothesis function.

## 3 Experiment and Discussions

The method of BP is implemented in MATLAB R2016a (The Mathworks, Natick, MA, USA). This experiments were accomplished on a computer with 3.30 GHz core and 4 GB RAM, running under the Window 8 and based on 64-bit processors.

### 3.1 Database

We used leaves as the object of study, which are all size of  $200 \times 200$  and in jpg format. The input dataset contains 90 images, which 30 of ginkgo biloba, 30 of Phoenix tree leaf and 30 of Osmanthus leaves. As mentioned above, we use the 5-fold cross validation to prevent the case of over-fitting, Thence, we ran one trials with each 72 (25 ginkgo biloba and 26 Osmanthus and 21 Phoenix tree) are used for training and the left 18 (5 ginkgo biloba and 4 Osmanthus and 9 Phoenix tree) are used for test. Figure 6 show one trail of 5-fold cross-validation.

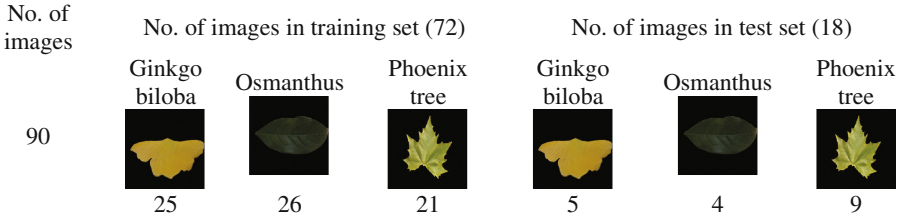


Fig. 6. One trail of five-fold cross-validation

### 3.2 Algorithm Comparison

The wavelet-entropy features were fed into different classifiers. We compared our method (BPNN) with Medium KNN [26], Coarse Gaussian SVM [27], Complex Tree [28], Cosine KNN [29]. The results of comparing were shown in Table 3.

Table 3. Comparison with different methods

Method	Accuracy
Medium KNN [26]	84.4%
Coarse Gaussian SVM [27]	66.7%
Complex Tree [28]	81.1%
Cosine KNN [29]	82.2%
BPNN (Our)	90.0%

(KNN = k-nearest neighbors, SVM = support vector machine, BPNN = back-propagation neural network)

Results in Table 3 are the central contribution of our experiment, we can see that the accuracy of our method achieves 90.0%, and it is better than other methods. Why BPNN performs the best among the fiver algorithms? The reason lies in the universal approximation theorem proven in reference [30].

## 4 Conclusion and Future Research

This paper introduced a novel automatic classify method for the leaves images. With the combination of BP neural network and wavelet-entropy, the accuracy of our method achieves 90.0%.

It costs 1.2064 s to finish feature extraction for 90 leaves images and the average time for each image is 0.0134 s.

Nevertheless, there are still several problems remaining unsolved: (1) we try to improve the accuracy of the algorithm; (2) we can employ other methods of feature extraction in order to decrease the time of extraction; (3) we may extend this method to other type of images, such as car image, tree image, etc.



**Acknowledgment.** This paper is financially supported by Natural Science Foundation of China (61602250), Natural Science Foundation of Jiangsu Province (BK20150983).

## References

1. Carro, F., Soriguer, R.C.: Long-term patterns in Iberian hare population dynamics in a protected area (Donana National Park) in the southwestern Iberian Peninsula: effects of weather conditions and plant cover. *Integr. Zool.* **12**, 49–60 (2017)
2. Lim, S.H., et al.: Plant-based foods containing cell wall polysaccharides rich in specific active monosaccharides protect against myocardial injury in rat myocardial infarction models. *Sci. Rep.* **6**, 15 (2016). Article ID: 38728
3. Du, J.X., et al.: Computer-aided plant species identification (CAPSI) based on leaf shape matching technique. *Trans. Inst. Meas. Control* **28**, 275–284 (2006)
4. Heymans, B.C., et al.: A neural network for *Opuntia* leaf-form recognition. In: *IEEE International Joint Conference on Neural Networks*, pp. 2116–2121. IEEE (1991)
5. Wu, S.G., et al.: A leaf recognition algorithm for plant classification using Probabilistic Neural Network. In: *International Symposium on Signal Processing and Information Technology*, p. 120. IEEE (2007)
6. Wang, X.F., et al.: Classification of plant leaf images with complicated background. *Appl. Math. Comput.* **205**, 916–926 (2008)
7. Jeatrakul, P., Wong, K.W.: Comparing the performance of different neural networks for binary classification problems. In: *Eighth International Symposium on Natural Language Processing, Proceedings*, pp. 111–115. IEEE (2009)
8. Dyrmann, M., et al.: Plant species classification using deep convolutional neural network. *Biosyst. Eng.* **151**, 72–80 (2016)
9. Zhang, S.W., et al.: Semi-supervised orthogonal discriminant projection for plant leaf classification. *Pattern Anal. Appl.* **19**, 953–961 (2016)
10. Meier, D.C., et al.: Fourier transform infrared absorption spectroscopy for quantitative analysis of gas mixtures at low temperatures for homeland security applications. *J. Testing Eval.* **45**, 922–932 (2017)
11. Tiwari, S., et al.: Cloud point extraction and diffuse reflectance-Fourier transform infrared spectroscopic determination of chromium(VI): A probe to adulteration in food stuffs. *Food Chem.* **221**, 47–53 (2017)
12. Garrido, M.: The feedforward short-time fourier transform. *IEEE Trans. Circ. Syst. II-Express Briefs* **63**, 868–872 (2016)
13. Saneva, K.H.V., Atanasova, S.: Directional short-time Fourier transform of distributions. *J. Inequal. Appl.* **10**, Article ID: 124 (2016)
14. Huo, Y., Wu, L.: Feature extraction of brain MRI by stationary wavelet transform and its applications. *J. Biol. Syst.* **18**, 115–132 (2010)
15. Ji, G.L., Wang, S.H.: An improved reconstruction method for CS-MRI based on exponential wavelet transform and iterative shrinkage/thresholding algorithm. *J. Electromag. Waves Appl.* **28**, 2327–2338 (2014)
16. Yang, M.: Dual-tree complex wavelet transform and twin support vector machine for pathological brain detection. *Appl. Sci.* **6**, Article ID: 169 (2016)
17. Liu, A.: Magnetic resonance brain image classification via stationary wavelet transform and generalized eigenvalue proximal support vector machine. *J. Med. Imaging Health Inform.* **5**, 1395–1403 (2015)

18. Bezawada, S., et al.: Automatic facial feature extraction for predicting designers' comfort with engineering equipment during prototype creation. *J. Mech. Des.* **139**, 10 (2017). Article ID: 021102
19. Gerdes, M., et al.: Decision trees and the effects of feature extraction parameters for robust sensor network design. *Ekspluat. Niezawodn.* **19**, 31–42 (2017)
20. Zhang, Y.: Binary PSO with mutation operator for feature selection using decision tree applied to spam detection. *Knowl. Based Syst.* **64**, 22–31 (2014)
21. Yang, J.: Preclinical diagnosis of magnetic resonance (MR) brain images via discrete wavelet packet transform with Tsallis entropy and generalized eigenvalue proximal support vector machine (GEPSVM). *Entropy* **17**, 1795–1813 (2015)
22. Phillips, P., et al.: Pathological brain detection in magnetic resonance imaging scanning by wavelet entropy and hybridization of biogeography-based optimization and particle swarm optimization. *Prog. Electromag. Res.* **152**, 41–58 (2015)
23. Sun, P.: Pathological brain detection based on wavelet entropy and Hu moment invariants. *Bio-Med. Mater. Eng.* **26**, 1283–1290 (2015)
24. Wei, L.: Fruit classification by wavelet-entropy and feedforward neural network trained by fitness-scaled chaotic ABC and biogeography-based optimization. *Entropy* **17**, 5711–5728 (2015)
25. Yang, J.: Identification of green, Oolong and black teas in China via wavelet packet entropy and fuzzy support vector machine. *Entropy* **17**, 6663–6682 (2015)
26. Zhou, X.-X.: Comparison of machine learning methods for stationary wavelet entropy-based multiple sclerosis detection: decision tree, k-nearest neighbors, and support vector machine. *Simulation* **92**, 861–871 (2016)
27. Sharma, B., et al.: Traffic accident prediction model using support vector machines with Gaussian kernel. In: *Fifth International Conference on Soft Computing for Problem Solving*, pp. 1–10. Springer, Berlin (2016)
28. Maleszka, M., Nguyen, N.T.: Using subtree agreement for complex tree integration tasks. In: Selamat, A., Nguyen, N.T., Haron, H. (eds.) *ACIIDS 2013. LNCS*, vol. 7803, pp. 148–157. Springer, Heidelberg (2013). doi:[10.1007/978-3-642-36543-0\\_16](https://doi.org/10.1007/978-3-642-36543-0_16)
29. Anastasiu, D.C., Karypis, G.: Fast parallel cosine k-nearest neighbor graph construction. In: *6th Workshop on Irregular Applications: Architecture and Algorithms (IA3)*, pp. 50–53. IEEE (2016)
30. Nguyen, H.D., et al.: A universal approximation theorem for mixture-of-experts models. *Neural Comput.* **28**, 2585–2593 (2016)