Proceeding of the IEEE
International Conference on Robotics and Biomimetics
Dali, China, December 2019

DBNet: A New Generalized Structure Efficient for Classification

Yonghao Dang[1], Fuxing Yang[1], Baiquan Su[1], Jianqin Yin[1,*] and Jun Liu[2,*]
[1]*Automation School, Beijing University of Posts and Telecommunication, Beijing, China*
[2]*Department of Mechanical Engineering, City University of Hong Kong, Hong Kong, SAR, China*

***Abstract*— In this paper, we propose a new deep learning structure named deep-broad network (DBNet) that is efficient for classification task. By modifying the decision-making mechanism of the deep structure, the proposed method can improve the testing efficiency while maintaining the testing accuracy. Specifically, the modified convolutional neural networks (CNNs) are first pre-trained and used to extract high-quality features. And then the dimension of extracted features is reduced by linear mapping. Finally, the broad learning system (BLS) uses processed features to make decisions. Compared with the previous deep structure, the efficiency of the proposed model is improved. Compared with the BLS, the DBNet has better performance. The proposed model is evaluated by using the CIFAR-10, CIFAR-100 and MNIST datasets. And experimental results show that the DBNet is effective and efficient. Code and models are available at *https://github.com/YHDang/DBNet*.**

***Index Terms*— Deep learning, broad learning system, classification.**

I. Introduction

Deep learning structures have been widely applied in many fields and achieved good results [1], [2], [3], [4]. Particularly, convolutional neural networks (CNNs) are successfully employed to image processing [5], action recognition [6] and recommender system [7]. New deep learning structures such as AlexNet [4], VGGNet [8], ResNet [9] and DenseNet [10], have been proposed and obtained improved performance in classification compared with the classic neural networks[11].

The deep-learning structure can be regarded as the representation-learning method which transforms the raw input data into a representation at a higher, more abstract level by stacking non-linear convolutional modules [12]. Previous deep CNNs are effective for extracting hierarchical, invariant and high-level local features [4], [8], [9], [10]. However, generalized CNNs apply the dropout and softmax mechanisms by integrating a high number of thinned networks to alleviate the overfitting problem in decision making [13]. Moreover, the softmax method contains a series of exponential computations, resulting in an increased computational time.

To reduce the computational cost, we had developed a DWnet structure by reducing the hierarchical levels to improve the efficiency for classification of sparse data (e.g., 3D skeleton) [14]. Although the pruned hierarchical CNN showed improved results on the classification of human actions by analyzing the simplified skeleton data, it cannot be directly used for processing the image data because of the high dimension of the features extracted from the original image. This motivates us to design a novel structure to deal with dense features. Specifically, the dimensionality reduction is applied to the features extracted by the pre-trained deep model. And then the broad learning system provides additional functions to enrich the simplified feature [15]. Therefore, the combination of the pruned deep CNN and BLS layer offers an effective framework for image classification while maintaining the high efficiency.

In this paper we propose a new framework named deep-broad network (DBNet). To achieve this framework for processing images, the pruned deep structures are trained and used to extract features. The high dimension of extracted features is then reduced by using a linear mapping method. Finally, the BLS is applied to classify samples, as shown in Fig. 1. The testing results on the standard image datasets (i.e., CIFAR and MNIST datasets) indicate that the DBNet is able to obtain an optimal classification. Moreover, compared with the original deep structure, the DBNet significantly saves computational time. The main contributions of this paper are highlighted below:

1) We propose a novel framework named deep-broad network (DBNet) to improve the modeling capability and efficiency of the model. The linear mapping is used to reduce the dimension of features extracted by deep structures. And then transformed features are input to the broad learning system for classification.
2) Compared with previous deep CNN structures, the DBNet significantly reduces the computational cost while improving or maintaining the approximation capability of the model.

In the remaining part of this paper, we review the previous researches about deep neural network that are related to this topic. The details of the proposed DBNet and the implementation of the model are explained in Section III. Experimental results are shown and discussed in Section IV. Lastly, a summary of this study and conclusions are provided

*Jianqin Yin and Jun Liu are corresponding authors (e-mail: jqyin@bupt.edu.cn; jun.liu@cityu.edu.hk).

978-1-7281-6321-5/19//$31.00 ©2019 IEEE

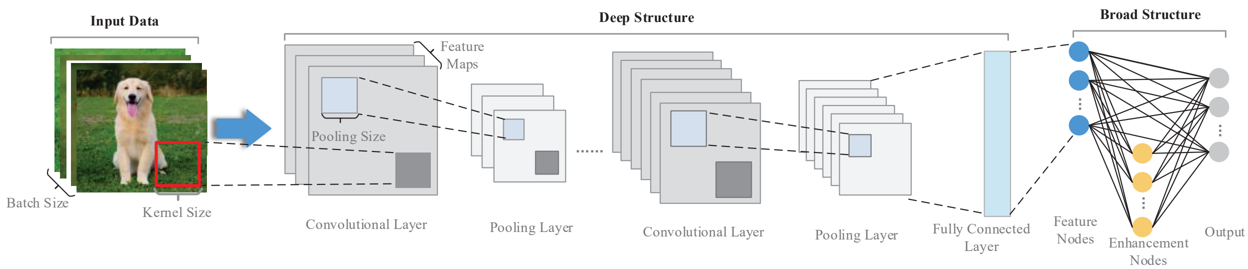


Fig. 1: The structure of the deep-broad network. First, the deep structure is used to extract features of input image. Secondly, the dimension of features is reduced by generating feature nodes (blue nodes). And then feature nodes are transformed into a high feature space (yellow nodes). Finally, the BLS integrates feature and enhancement nodes to make decisions.

in Section V.

II. Related Work

A. Deep Convolutional Neural Network

In the past few years, deep CNNs have attracted extensive attention of researchers, as the increased number of convolutional layers provides significantly improved classification results [16]. In 2012, a milestone work, AlexNet, won the ImageNet competition with five convolutional layers and three fully connected layers [8]. The AlexNet achieved the best result at that time as a result of the deeper structure and advanced performance of the hardware [17]. The appearance of VGGNet in 2014 demonstrated that deeper structure of a CNN model could further improve the learning ability [8]. Simonyan et al. replaced large convolutional kernels with a series of 3×3 convolutional kernels, which increases the number of layers in the deep CNN model [8].

However, gradient vanishing and exploding are unavoidable problems during deepening the model. To address these problems, He et al. introduced a connection-skipping mechanism to analyze the residual function [18] and proposed the ResNet [9]. Inspired by the ResNet, Huang et al. designed the DenseNet [10]. The DenseNet strengthens the feature propagation and reuses the old features, therefore reducing the number of parameters.

B. Broad Learning System

Chen et al. proposed the Broad Learning System (BLS) to offer an efficient and effective classification framework by expanding the network in a broad way [19]. The BLS is a flat fully connected network that is based on the Random Vector Functional-Link Neural Network (RVFLNN) [20], [21], [22]. In other studies, researchers have investigated the universal approximation property [23] and proposed multiple structural variations of the BLS [24]. Recently, BLS has attracted an increasing interest of researchers due to its flexible structure. For instance, Liu et al. combined the BLS and the k-Means algorithm to classify images on CIFAR-10 dataset [25]. Similarly, Liu and Chen applied the BLS to the common neural networks such as radial basis function neural network (RBF) and hierarchical extreme learning machine (H-ELM) [26]. To effectively process the uncertain and outlier data, Jin et al. introduced the distribution of errors into the training process and developed a robust broad learning system (RBLS) [19]. Additionally, the same group also proposed an alternative approach by building a graph regularized broad learning system (GBLS) based on manifold learning to determine the more discriminative output weights, which is proven effective for classifying outlier data [24].

III. Methodology

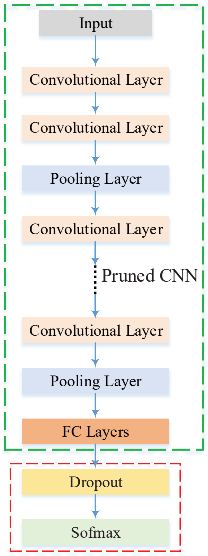
In general, the DBNet includes three major processes (see Fig. 1). Firstly, CNNs are used to extract primary features. Secondly, a mapping function is used to reduce the dimension of features. Finally, the DBNet integrates an extra BLS model to further classify the features.

A. Pruned Convolutional Neural Network

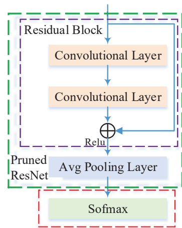
In this new DBNet, the fundamental structure is based on the popular CNNs such as AlexNet [4], VGGNet [8], ResNet [9] and DenseNet [10], to extract features from the raw data. The softmax guider is removed from the DBNet after training the model. And the dropout layer is pruned if it exists. As shown in Fig. 2(a), the dropout and softmax layers marked by the red frame are removed, leaving fully connected layers (FC) or the average pooling layer (i.e., marked in green frame). The FC layers and average pooling layer are kept because they are helpful to integrate local features extracted by previous convolutional layers. The softmax contains complex exponential computations, and the dropout incorporates ensemble into the model. Therefore, pruning the softmax and the dropout can improve the computational efficiency.

B. Reduction of the Dimension of Features

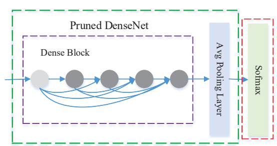
After removing the softmax and dropout layers, the pruned CNN is able to save the computational time. However, extracted features still have a high dimension. The high dimension can potentially pose a limitation in the number of non-linear enhancement nodes used in BLS, resulting in



(a) Convolutional neural network.



(b) ResNet block.



(c) DenseNet block.

Fig. 2: Structures of deep convolutional neural networks and pruned convolutional neural networks. Structures in the green box are the pruned structures used in this paper. And layers in the red box are removed.

larger computational errors. Therefore, the dimension of extracted features is reduced to further reduce the computational burden while meeting a minimal requirement of the number of enhancement nodes to build an effective BLS. To reduce the dimension of features, we adopt a random linear mapping operation (denoted as $f_{H\to L}$) to transfer extracted features from a high dimensional feature space into a low dimensional feature space. The mapping function is described as,

$$F_i = f_{H\to L}(Fea;W) = Fea{\cdot}W_{ei} + \beta_{ei}, i = 1,2,...,n \quad (1)$$

where Fea is the feature originally extracted by the pruned CNN, W_{ei} and β_{ei} are weights and bias generated randomly, n is the number of feature nodes. Empirically, the n is set to 10. The original features are represented as,

$$Fea = \theta\left(P\left(\mathrm{Relu}\left(X \otimes W_{con} + \beta_{con}\right)\right)\right) \quad (2)$$

where $\otimes$ denotes the convolutional function, W_{ei} and β_{ei} are weights and biases of the CNN; $\mathrm{Relu}(\cdot)$ is the activation function, and $P(\cdot)$ and $\theta(\cdot)$ are the pooling function and the non-linear activation function, respectively. Generally, the dimension of features extracted by the deep structure is more than 256. After the random linear mapping, the dimension of features is reduced to 100 (i.e. the dimension of each feature node is 10).

C. Generation of Non-Linear Nodes

While the random mapping reduces the dimension of features, it also damages the intrinsic classification capability because of the loss of inter-nodal information. To mitigate this damage and improve the modeling capability, we add an additional layer to transform the feature nodes into non-linear structure. The transformation function is represented as,

$$En_j = \xi\left(Fn{\cdot}W_{hj} + \beta_{hj}\right), j = 1,2,...,m \quad (3)$$

where $Fn = [F_1, F_2, \ldots, F_n]$ is the set of all the linearized feature nodes, W_{hj} and β_{hj} are weights and biases that are initialized randomly, En_j is the jth enhancement node and m is the number of enhancement nodes. In the end, all the feature nodes and nonlinearly processed nodes are connected and fed to the output layer for decision making (see Fig. 1). The decision-making mechanism of the DBNet is modified to improve the efficiency of the model. Experimental results (explained in the next section) also confirm that the pruned deep structure equipped with the BLS can keep or improve the modeling capability.

IV. Experiments

A. Experimental Settings

The proposed structure was evaluated and compared with the mainstream deep structures [4], [8], [9], [10] using the same dataset. The testing platform is equipped with Ubuntu 16.04 operating system, Intel-i7 3.5 GHz CPU, two Nvidia GeForce GTX 1080Ti graphic cards and 32 GB memory. And the performance of algorithms is assessed using the benchmark datasets including MNIST data [27], CIFAR-10 and CIFAR-100 [28].

In this paper, we mainly focus on the top-1 accuracy. The evaluation criteria is defined as follows.

$$Acc = \frac{\sum_{c=1}^{C}\sum_{j=1}^{m} sample\_pred_j^c}{\sum_{i=1}^{C} sample\_num_i} \times 100\% \quad (4)$$

where C is the total classes of samples, c is the ground truth of one sample, $sample\_num_i$ represents the number of samples with the ground truth c, $sample\_pred_j^c$ represents the recognizing result of the jth sample with the ground truth c. If the jth sample is recognized with the correct label c, the $sample\_pred_j^c$ is set to 1 and otherwise set to 0.

TABLE I: Testing accuracy on the MNIST dataset.

Method	Acc (Top-1)
BLS	**98.96%**
BP	98.57%
DBNet(*BP_BLS*)	98.59%
CNN	**99.83%**
DBNet(*CNN_BLS*)	99.50%
ResNet	**99.85%**
DBNet(*ResNet_BLS*)	99.47%

B. Comparison of Testing Accuracy with the State of the Art

1) Evaluation Results on MNIST Dataset: The feasibility of our method is verified on the classical MNIST dataset [27]. The testing accuracy of each method is shown in TABLE I. It can be seen from the TABLE I that the accuracy of *BP_BLS* is higher than that of BP, and results of *CNN_BLS* and *ResNet_BLS* are close to the CNN and ResNet respectively, which indicates that the DBNet can maintain the comparable approximation ability as the deep structure.

TABLE II: Testing Accuracy on the CIFAR-10 Dataset.

Method	Acc (Top-1)
BLS	42.34%
CNN	81.03%
DBNet(*CNN_BLS*)	**81.04%**
AlexNet	91.45%
DBNet(*AlexNet_BLS*)	**91.45%**
VGG	84.04%
DBNet(*VGG_BLS*)	**84.62%**
ResNet-56	93.85%
DBNet(*ResNet_BLS*)	**93.93%**
DenseNet	95.36%
DBNet(*DenseNet_BLS*)	**95.36%**

2) Evaluation Results on CIFAR-10 Dataset: To some degree, the testing accuracy is proportional to the number of enhancement nodes. Meanwhile, the computational efficiency decreases with the number of enhancement nodes increases. We seek the trade-off between the testing accuracy and computational time, and then record results that keep the balance between the accuracy and efficiency in TABLE II. Except for the *AlexNet_BLS* and *DenseNet_BLS*, other DBNets perform better than corresponding deep structures. And the *AlexNet_BLS* and *DenseNet_BLS* obtain the same accuracy as the original deep structures respectively. The features extracted by deep structures have been close to the optimal solution. It is easier for DBNet to solve the optimal solution.

3) Evaluation Results on CIFAR-100 Dataset: We further validated the robustness of DBNet on the CIFAR-100 dataset that has more categories than the CIFAR-10. TABLE III shows the best result of each method. Compared with original deep structures, results of the DBNet are improved. Features extracted by deep structures are discriminative and representative. And the BLS is able to effectively integrate features. Therefore, the DBNet performs better than the corresponding deep structure.

TABLE III: Testing Accuracy on the CIFAR-100 Dataset.

Method	Acc (Top-1)
AlexNet	64.49%
DBNet(*AlexNet_BLS*)	**65.54%**
VGG	69.66%
DBNet(*VGG_BLS*)	**69.97%**
ResNet-56	72.84%
DBNet(*ResNet_BLS*)	**73.01%**
DenseNet	75.86%
DBNet(*DenseNet_BLS*)	**75.95%**

TABLE IV: Average testing time on the MNIST dataset.

Method	Testing Time (ms)
BLS	**0.081**
BP	0.222
DBNet(*BP_BLS*)	0.141
CNN	0.360
DBNet(*CNN_BLS*)	**0.268**
ResNet	0.585
DBNet(*ResNet_BLS*)	**0.256**

C. Results of Computing Efficiency

1) Testing Results on MNIST Dataset: Results in TABLE IV correspond to the accuracy in TABLE I. It can be seen from TABLE IV DBNets are more efficient than other deep structures. Compared with the softmax, the BLS reduces the computational complexity during making decisions. Moreover, the *BP_BLS* spends less time to achieve a better result than the BP method, which indicates that the DBNet can improve the efficiency of the model while maintaining the approximation ability. The number of enhancement nodes of BLS and *BP_BLS* is 8200. The *BP_BLS* runs slower than the BLS. The reason is that compared with the standard BLS, the BP in DBNet spends more time to extract features.

2) Testing Results on CIFAR-10 Dataset: Results in TABLE V correspond to the accuracy in TABLE II. Results in TABLE V indicate that the DBNet spends less computational time than the corresponding deep structure. For

TABLE V: Average Testing Time on the CIFAR-10 Dataset.

Method	Testing Time (ms)
BLS	0.447
CNN	0.523
DBNet(*CNN_BLS*)	**0.229**
AlexNet	0.863
DBNet(*AlexNet_BLS*)	**0.787**
VGG	0.775
DBNet(*VGG_BLS*)	**0.549**
ResNet-56	1.023
DBNet(*ResNet_BLS*)	**0.751**
DenseNet	2.872
DBNet(*DenseNet_BLS*)	**1.745**

each testing sample, the *AlexNet_BLS* runs 0.076ms faster than AlexNet, and the *DenseNet_BLS* runs 1.127ms faster than the DenseNet. The efficiency of the model is improved by modifying the decision-making mechanism of the deep structure.

TABLE VI: Average Testing Time on the CIFAR-100 Dataset.

Method	Testing Time (ms)
AlexNet	1.395
DBNet(*AlexNet_BLS*)	**0.483**
VGG	0.786
DBNet(*VGG_BLS*)	**0.497**
ResNet-56	2.146
DBNet(*ResNet_BLS*)	**1.387**
DenseNet	2.857
DBNet(*DenseNet_BLS*)	**1.745**

3) Testing Results on CIFAR-100 Dataset: Results in TABLE VI correspond to the accuracy in TABLE III. Experimental results summarized in TABLE VI also demonstrate that our method performs efficiently on the large dataset. Furthermore, compared with the CIFAR-10, the efficiency of the DBNet is improved more obviously on CIFAR-100 dataset, which suggests that the DBNet is potential to process the large dataset.

D. Impact of Enhancement Nodes

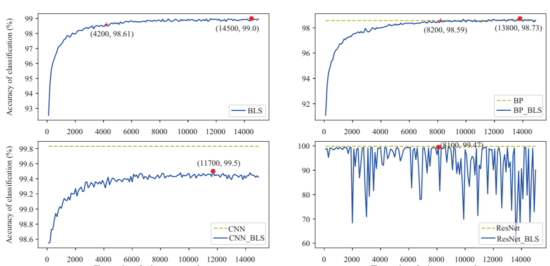


Fig. 3: Relationship between enhancement nodes and accuracy on the MNIST. The yellow dashed is the result of the original deep structure. Red points represent the best result of each method. And red '+' represents the point when the result of the DBNet exceeds that of the original deep structure.

In the DBNet, the number of enhancement nodes is the key parameter. In this section, we discuss the effect of the number of enhancement nodes on recognizing result. We evaluate our method on three datasets. The number of enhancement nodes is initialized to 100 and increased by 100 each time until it reaches 15000.

1) Results on MNIST Dataset: It can be seen from Fig. 3 that except for the *ResNet_BLS*, the testing accuracy of other methods increases as the number of enhancement nodes increases. And when the number of enhanced nodes is 8200, *BP_BLS* has performed better than BP. The result of *ResNet_BLS* oscillates constantly. We conjecture that there are skip connections in the ResNet, which makes the *ResNet_BLS* is sensitive to the number of enhancement nodes.

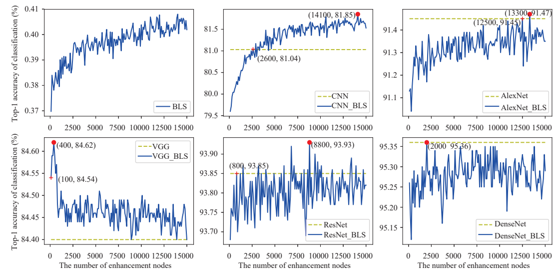


Fig. 4: Relationship between enhancement nodes and accuracy on the CIFAR-10. The yellow dashed is the result of the original deep structure. Red points represent the best result of each method. And red '+' represents the point when the result of the DBNet exceeds that of the original deep structure.

2) Results on CIFAR-10 Dataset: As shown in Fig. 4, results of the DBNet are oscillating since there is some randomness in the BLS. As the number of enhancement nodes increases, some of the results are improved over those recorded in TABLE II. But the recognizing accuracy levels off or falls off after a peak. If the number of enhancement nodes exceeds the necessary number, the final feature matrix connected to the output will be ill-conditioned leading to larger computational errors [29]. Therefore, an appropriate number of enhancement nodes can improve the modeling capability of the model.

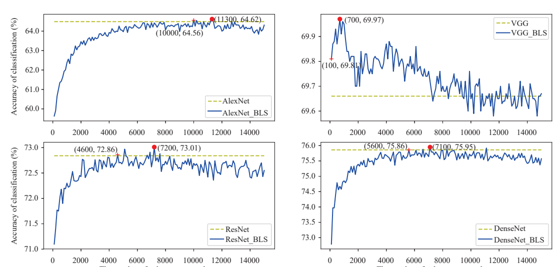


Fig. 5: Relationship between enhancement nodes and accuracy on the CIFAR-100. The yellow dashed is the result of the original deep structure. Red points represent the best result of each method. And red '+' represents the point when the result of the DBNet exceeds that of the original deep structure.

3) Resutls on CIFAR-100 Dataset: Fig. 5 shows that except for the *VGG_BLS*, the testing accuracy of DBNets increases as the number of enhancement nodes increases. The accuracy of the *VGG_BLS* has been declining. The possible reason for the decline is that the dimension of features extracted by the VGG is high. And too many enhancement nodes can cause the improvement of the complexity limiting the approximation capability of the model.

Results demonstrate that the DBNet performs better than the corresponding deep structure. And it is noteworthy that compared with the MNIST dataset, our method performs better on CIFAR datasets. Images in the MNIST dataset are grey level maps. After pre-processing, the data in the MNIST is transformed to 0 or 1. Compared with CIFAR datasets, processed images in the MNIST dataset have simpler features. We think that our method is suitable for the large dataset and able to process the dense feature effectively.

V. Conclusion

In this paper, we propose a novel efficient framework named deep-broad network (DBNet) for the classification task. The proposed method can not only extract representative features but also make decisions rapidly. Because of the improved structure by replacing the dropout and sofmax layers with the BLS, the DBNet is more efficient than the corresponding deep structure. Furthermore, the performance of the DBNet is better than that of the BLS. Experimental results prove that the DBNet is computationally efficient while producing comparable and improved results compared to its corresponding CNNs version.

Acknowledgment

This work was supported partly by the National Natural Science Foundation of China Grant No. 61673192, No. U1613212, No. 61573219, the Fund for Outstanding Youth of Shandong Provincial High School ZR2016JL023, Shandong Provincial Key Research and Development Plan 2017CXGC1504, Dominant Discipline and Talent Team of Shandong Province Higher Education Institutions and the Basic Scientific Research Project of Beijing University of Posts and Telecommunications 2018RC31.

References

[1] L. Wang, S. Guo, W. Huang, Y. Xiong, and Y. Qiao, "Knowledge guided disambiguation for large-scale scene classification with multi-resolution cnns," *IEEE Trans. Image Process.*, vol. 26, no. 4, pp. 2055–2068, 2017.

[2] Y. Xiong, K. Zhu, D. Lin, and X. Tang, "Recognize complex events from static images by fusing deep channels," in *CVPR*, 2015, pp. 1600–1609.

[3] J. Chang, G. Meng, L. Wang, S. Xiang, and C. Pan, "Deep self-evolution clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, p. in press, 2018, doi: 10.1109/TPAMI.2018.2889949.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012, pp. 1097–1105.

[5] P. H. O. Pinheiro, R. Collobert, and P. Dollár, "Learning to segment object candidates," in *NIPS*, 2015, pp. 1990–1998.

[6] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. V. Gool, "Temporal segment networks for action recognition in videos," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 11, pp. 2740–2755, 2019.

[7] A. v. d. Oord, S. Dieleman, and B. Schrauwen, "Deep content-based music recommendation," in *NIPS*, 2013, pp. 2643–2651.

[8] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *In CVPR*, 2016, pp. 770–778.

[10] G. Huang, Z. Liu, and K. Q. Weinberger, "Densely connected convolutional networks," in *CVPR*, 2017, pp. 2261–2269.

[11] B. Lodhi and J. Kang, "Multipath-densenet: A supervised ensemble architecture of densely connected convolutional networks," *Inf. Sci.*, vol. 482, pp. 63–72, 2019.

[12] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.

[13] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[14] Y. Dang, F. Yang, and J. Yin, "DWnet: Deep-Wide Network for 3D Action Recognition," *arXiv e-prints*, p. arXiv:1908.11036, Aug. 2019.

[15] C. L. P. Chen and Z. Liu, "Broad learning system: An effective and efficient incremental learning system without the need for deep architecture," *IEEE Trans. Neural Netw. Learning Syst.*, vol. 29, no. 1, pp. 10–24, 2018.

[16] K. Zhang, M. Sun, T. X. Han, X. Yuan, L. Guo, and T. Liu, "Residual networks of residual networks: Multilevel residual networks," *IEEE Trans. Circuits Syst. Video Techn.*, vol. 28, no. 6, pp. 1303–1314, 2018.

[17] Y. Xu, H. Wang, X. Liu, and W. Sun's, "An improved multi-branch residual network based on random multiplier and adaptive cosine learning rate method," *J. Visual Communication and Image Representation*, vol. 59, pp. 363–370, 2019.

[18] K. Zhang, Y. Guo, X. Wang, J. Yuan, and Q. Ding, "Multiple feature reweight densenet for image classification," *IEEE Access*, vol. 7, pp. 9872–9880, 2019.

[19] J. Jin and C. L. P. Chen, "Regularized robust broad learning system for uncertain data modeling," *Neurocomputing*, vol. 322, pp. 58–69, 2018.

[20] Y. . Pao and Y. Takefuji, "Functional-link net computing: theory, system architecture, and functionalities," *IEEE Computer*, vol. 25, no. 5, pp. 76–79, 1992.

[21] Y.-H. Pao, G.-H. Park, and D. J. Sobajic, "Learning and generalization characteristics of the random vector functional-link net," *Neurocomputing*, vol. 6, no. 2, pp. 163 – 180, 1994.

[22] B. Igelnik and Yoh-Han Pao, "Stochastic choice of basis functions in adaptive function approximation and the functional-link net," *IEEE Trans. Neural Netw.*, vol. 6, no. 6, pp. 1320–1329, 1995.

[23] C. L. P. Chen, Z. Liu, and F. Shuang, "Universal approximation capability of broad learning system and its structural variations," *IEEE Trans. Neural Netw. Learning Syst.*, vol. 30, no. 4, pp. 1191–1204, 2019.

[24] J. Jin, Z. Liu, and C. L. P. Chen, "Discriminative graph regularized broad learning system for image recognition," *Inf. Sci.*, vol. 61, no. 11, pp. 179–192, 2018.

[25] Z. Liu, J. Zhou, and C. L. P. Chen, "Broad learning system: Feature extraction based on k-means clustering algorithm," in *ICCSS*, 2017, pp. 683–687.

[26] Z. Liu and C. L. P. Chen, "Broad learning system: Structural extensions on single-layer and multi-layer neural networks," in *SPAC*, 2017, pp. 136–141.

[27] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *PROC IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[28] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Computer Science Department, University of Toronto, Tech. Rep, Tech. Rep., 2009.

[29] C. L. P. Chen, "A rapid supervised learning neural network for function interpolation and approximation," *IEEE Trans. Neural Netw.*, vol. 7, no. 5, pp. 1220–1230, 1996.