

Received April 13, 2020, accepted April 30, 2020, date of publication May 11, 2020, date of current version May 22, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2993618

Convolutional Neural Network Based on Complex Networks for Brain Tumor Image Classification With a Modified Activation Function

ZHIGUAN HUANG¹, XIAOHAO DU², LIANGMING CHEN², YUHE LI¹, MEI LIU², YAO CHOU³, AND LONG JIN^{1,2}, (Member, IEEE)

¹Guangdong Provincial Engineering Technology Research Center for Sports Assistive Device, Guangzhou Sport University, Guangzhou 510000, China

²School of Information Science and Engineering, Lanzhou University, Lanzhou 730000, China

³Department of Electrical and Computer Engineering, Brigham Young University, Provo, UT 84602, USA

Corresponding authors: Yuhe Li (13922738963@139.com) and Long Jin (jinlongsysu@foxmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61703189, in part by the Guangzhou Sport University Innovation and Strengthen Project under Grant 5200080589, in part by the Ministry of Education Industry-Academic Cooperation Collaborative Education Program of China under Grant 201901007048, in part by the Baidu Online Network Technology (Beijing) Company, Ltd., in part by the Team Project of Natural Science Foundation of Qinghai Province, China, under Grant 2020-ZJ-903, in part by the Key Laboratory of IoT of Qinghai under Grant 2020-ZJ-Y16, and in part by the Fundamental Research Funds for the Central Universities under Grant lzujbky-2019-89.

ABSTRACT The diagnosis of brain tumor types generally depends on the clinical experience of doctors, and computer-assisted diagnosis improves the accuracy of diagnosing tumor types. Therefore, a convolutional neural network based on complex networks (CNNBCN) with a modified activation function for the magnetic resonance imaging classification of brain tumors is presented. The network structure is not manually designed and optimized, but is generated by randomly generated graph algorithms. These randomly generated graphs are mapped into a computable neural network by a network generator. The accuracy of the modified CNNBCN model for brain tumor classification reaches 95.49%, which is higher than several models presented by other works. In addition, the test loss of brain tumor classification of the modified CNNBCN model is lower than those of the ResNet, DenseNet and MobileNet models in the experiments. The modified CNNBCN model not only achieves satisfactory results in brain tumor image classification, but also enriches the methodology of neural network design.

INDEX TERMS Convolutional neural network, complex networks, randomly generated graph, network generator, brain tumors.

I. INTRODUCTION

In recent decades, an increasing number of hospitals have adopted artificial intelligence methods to assist medical diagnosis as computer technology thrives [1]–[7], which promotes the reform and development of intelligent medical care at the same time [8]–[13]. Human higher neural activities such as memory, intelligence and consciousness are controlled by the central nervous system of the brain which is the most complicated structure [14]. Once the tumor metastasizes to any part of the brain, it will damage the different functions of the human body whether it is benign or malignant [15]. In addition, brain tissue is more complex than any other part of the body, making treatment and diagnosis difficult. Traditionally, in addition to analyzing the symptoms of a patient,

The associate editor coordinating the review of this manuscript and approving it for publication was Mohammad Shorif Uddin¹.

doctors usually need physiological test results and a series of images generated by magnetic resonance imaging (MRI), which is a technology that sends electromagnetic waves to an object and returns images of its internal structure [16]) to diagnose the classification of brain tumors.

Medical image analysis is a revolution of practicability and innovative concepts due to the rapid development of hardware, and the use of complex mathematical tools, which can obtain clearly visible medical images [17]. Based on these medical images, effective image analysis can help doctors diagnose and treat patients. The application of machine learning in medical image analysis, such as support vector machines (SVMs) and random forests, has greatly promoted the development of computer-aided medicine [18], [19]. Since the rapid development of deep learning, medical image analysis has made great progress, many new technologies have emerged for medical image analysis, such as

convolutional neural networks, 3-dimensional convolutional neural networks, and neural computing [20]–[23].

Image classification for traditional machine learning algorithms, for example, SVM, is hard to implement for large-scale training samples. In addition, they are quite likely to fail to solve multi-classification problems. The performance of a convolutional neural network is determined by the depth, width, and residual connections of the network. At present, models used for image classification are generally composed of convolutional neural networks and are almost regularly connected. Models with excellent performance rely on more complex connection modes, and the connection methods of these modes are effective. In addition, some models achieve higher accuracy as the depth of the network increases. However, it increases production costs for industrial realization and mobile terminal deployment.

A convolutional neural network based on complex networks, a new type of model for image classification, is termed the CNNBCN model [24]. Different from traditional regular connections, the connection of the CNNBCN model is not manually designed, but randomly generated by some random graphs. Although CNNBCN is based on random graphs, the nodes in these graphs are highly correlated, not simply pure random. In addition, the network structure of these random graphs is close to the distribution of neurons in the human brain. The CNNBCN model uses a neural network generator to generate random neural networks, and the performance on classification tasks is even better than classic convolutional neural networks. The main task of the network generator of the CNNBCN model is to generate a random graph and convert it into an applicable network. The distribution of connections among nodes in the generated graph is then applied to a network. Then, the CNNBCN model uses a heuristic approach to convert the generated random graph into a directed acyclic graph (DAG). Finally, the generated DAG is mapped into the CNNBCN model in which each neuron has a series of operations such as deconvolution, batch-normalization, and activation function processing. Therefore, the network that can be trained directly is attained. In the experiment, a brain tumor dataset [25] consisting of glioma, meningioma, and pituitary tumors, is used for training. After testing, the accuracy of the trained model can reach 95.49% for brain tumor classification.

The related work of this paper is as follows:

- The structural design of convolutional neural networks has changed from simply increasing the depth and width to exploring the complex connection modes such as ResNet [26] and DenseNet [27]. Currently, in addition to artificially designed more sophisticated connections, the automatic generation of neural networks is also an important research direction. Our work is the application of randomly generated convolutional neural networks to the classification of brain tumors.
- With the enhancement of medical equipment and the promotion of deep learning, computer-aided

detection (CAD) has made rapid progress in lesion detection, image segmentation, image registration, and image fusion [17], [19], [20]. Our work applies a new method based on a convolutional neural network for medical image analysis.

- Neural network architecture search (NAS) is an algorithm for finding the optimal neural network architecture, which can create some network structures that have not been explored before [24]. NASNet only allows skip connections with one downsampling, but other types of connections are also worth trying. The foundation of the model on which our work relies is presented in [24], and the skip connections in the model structure are of multiple groups and random. It is worth exploring in more fields, not just for brain tumor classification.
- Over the past decade, with the rise of research on complex networks, random graphs have become an important model of complex networks, called random networks. Applying random graphs to neural networks is also a valuable research direction. The results of evaluating the mean of mutual information for random graph models are provided in [28]. The approach presented in [29] uses a pre-trained CNN based on the Caffe deep learning framework for the analysis and classification of the random graphs and networks. The presented CNNBCN model is based on the connection patterns of random graph and networks to be applied to the backbone network for feature extraction.

The rest of this paper is divided into four parts. In Section II, the theoretical analysis of the CNNBCN network generator is presented. Then, the structure of the CNNBCN model is given in Section III. For verification, brain tumor classification experiments among the CNNBCN model and other classic image classification models are conducted in Section IV. At length, in Section V, the conclusion of the paper is summarized. In addition, the main contributions of this paper are as follows:

- In this paper, a new type of convolutional neural network based on complex networks is applied for medical imaging classification of brain tumors. It enriches practical and effective methods for medical image analysis.
- Based on the original CNNBCN model, a combination of activation functions is proposed to replace the original activation functions, which improves the performance of the CNNBCN model and the classification accuracy by about 0.5% – 0.75%.
- The experimental results confirm that the CNNBCN model is practical and effective. In addition, the combination of activation functions improves the performance of the original model. A neural network generated by a random complex network performs as well as or even better than an artificially designed convolutional neural network.
- The CNNBCN model provides a new idea and method for the construction of neural networks. Randomly generated networks reduce the manual

intervention, provide greater exploration space for neural network structure design, and provide the potential for better performance.

II. METHODOLOGY

In this section, the theory of randomly generated graphs and the concept of a network generator, which is the foundation of the CNNBCN model, are presented.

A. RANDOM GENERATED GRAPH

Constructing a randomly generated graph is the first step of the network generator. Three algorithms for randomly generating graphs namely, the *Erdos-Renyi* (ER) algorithm, *Watts-Strogatz* (WS) algorithm and *Barabasi-Albert* (BA) algorithm, are provided in [30]–[32] and are used to construct the randomly generated graph.

First, the simplest representation of the ER random network model is provided in [30], [33]. The total number of generated graph nodes is denoted as V_{ER} where the subscript ER denotes the corresponding algorithm. At each step, two nodes are arbitrarily selected and connected with probability p . The p is determined as

$$p = 2E/V_{ER}(V_{ER} - 1), \tag{1}$$

where E is the number of edges, $E \leq V_{ER}(V_{ER} - 1)/2$. The process stops when the number of edges reaches E . The crucial statistical properties of the generated graph are obtained in the following discussions. The average degree of the network can be written as [33]

$$\langle \kappa \rangle = p(V_{ER} - 1) \simeq pV_{ER}, \tag{2}$$

where $\langle \cdot \rangle$, κ and \simeq represent the corresponding average value, degree of a node, and asymptotic equality, respectively. The degree distribution is obtained:

$$P(\kappa)_{ER} = \Lambda_{V_{ER}-1}^\kappa p^\kappa (1-p)^{V_{ER}-1-\kappa} \underset{V_{ER} \rightarrow \infty}{\simeq} e^{-\langle \kappa \rangle} \frac{\langle \kappa \rangle^\kappa}{\kappa!}. \tag{3}$$

The average cluster coefficient of random networks can be written as

$$\Lambda_{ER} = p \simeq \frac{\kappa}{V_{ER}} \propto V_{ER}^{-1}, \tag{4}$$

where $\alpha \propto \beta$ denotes that α is proportional to β . The average distance of the nodes in random networks is obtained as follow:

$$\langle \zeta \rangle_{ER} \simeq \frac{\ln V_{ER}}{\ln \langle \kappa \rangle} \propto \ln V_{ER}. \tag{5}$$

Second, the WS small-world network model, which is between the rule network and the random network, is provided in [31], [33]. Starting from a regular net model, each edge is randomly reconnected with a probability p while remaining as a simple graph. Consequently, each node has μ neighbors, and the generated model is displayed as a regular net model when $p = 0$. However, when $0 < p < 1$, the expectation value of the random reconnection edge is $p\mu V_{WS}(V_{WS} \rightarrow \infty)$, where the symbol V_{WS} indicates the

number of generated graph nodes along with the subscript WS denoting the related algorithm. Additionally, the generated model is displayed between the rule and the randomness. All edges of the model are randomly reconnected and transformed into an ER random mesh model when $p = 1$. λ is used to denote the constant degree value of every node in the rule network corresponding to the small-world model. When $p = 0$, the average cluster coefficient $\hat{\Lambda}_{WS}$ is defined as:

$$\hat{\Lambda}_{WS} = \frac{3(\lambda - 2)}{4(\lambda + 1)}. \tag{6}$$

When $0 < p < 1$, the probability that neighbors of any node remain unchanged is $(1 - p)$, and the probability of their adjacency is also $(1 - p)$. Therefore, the expectation value of the average cluster coefficient of the small-world network is obtained:

$$\Lambda_{WS} = \frac{3(\lambda - 2)}{4(\lambda + 1)}(1 - p)^3. \tag{7}$$

Using the mean field method to obtain the analytical expression [30]

$$\langle \zeta \rangle_{WS}(V_{WS}, p) \approx \frac{V_{WS}^{1/\vartheta}}{\lambda} \Gamma(p\lambda V_{WS}), \tag{8}$$

where ϑ is the dimension of added edges and

$$\Gamma(\theta) = \begin{cases} \text{constant}, & \theta \ll 1, \\ \frac{4}{\sqrt{\theta^2 + 4\theta}} \operatorname{artanh} \frac{\theta}{\sqrt{\theta^2 + 4\theta}}, & \theta \approx 1, \\ \frac{\ln \theta}{\theta}, & \theta \gg 1. \end{cases}$$

When $p = 0$, the degree distribution of the small-world network is the same as the rule network. The degrees of all nodes are λ . When $p > 0$, since each edge retains one unchanged endpoint, each node has at least $\lambda/2$ edges after reconnection. The degree distribution is provided in [30]:

$$P(\kappa)_{WS} = \sum_{i=0}^{\varphi(\kappa, \lambda)} \Lambda_{\lambda/2}^i (1-p)^i p^{\lambda/2-i} \frac{(p\lambda/2)^{\kappa-\lambda/2-i}}{(\kappa - \lambda/2 - i)!} e^{-p\lambda/2}, \tag{9}$$

where

$$\varphi(\kappa, \lambda) = \min(\kappa - \lambda/2, \lambda/2).$$

It is concluded that the degree distribution follows with a Poisson distribution.

Third, the BA scale-free network closer to the actual complex networks is presented in [32], [33]. Growth and preference are two essential attributes of the BA scale-free network model. The former indicates that a complex network is an open system, that new basic units are continuously added, and that the total number of nodes is increasing. The latter means that the probability of a node connecting a new edge should be monotonously dependent on its existing degree. The model presented on the basis of these two principles is expressed as follows:

- When $\epsilon = 0$, the network has ω_0 nodes, and $\omega(\omega \leq \omega_0)$ old nodes are connected with a newly added node where symbol ϵ denotes time step in the network.
- A new node is connected to the old node i with a probability that is proportional to the degree of the node. Therefore, the connection probability is written as:

$$\prod(\tau_i) = \frac{\tau_i}{\sum_{j=1}^{V_{BA}-1} \tau_j}, \quad (10)$$

where symbol τ_i and V_{BA} with subscript $_{BA}$ indicate the concerned algorithm indicates the node i 's degree and the number of network nodes, respectively.

- Finally, the BA scale-free network model reaches a stable evolutionary state.

The BA scale-free model degree distribution is provided by the solution of the master equation in [34], [35]. The steady state distribution can be written as

$$P(\kappa)_{BA} = \lim_{\epsilon \rightarrow \infty} P(\kappa, \epsilon) = \lim_{\epsilon \rightarrow \infty} \frac{2}{W} (s/\epsilon)^{\frac{3}{2}} \Big|_{s=W^2\epsilon\kappa^{-2}} = 2W^2\kappa^{-3}, \quad (11)$$

where s means the index of a special node. Consequently, $P(\kappa)_{BA}$ follows an exponential function [32], [33]. The average distance of the BA scale-free model changes with V_{BA} is provided in [30]:

$$\langle \zeta \rangle_{BA} \approx \ln(V_{BA}) / \ln \ln(V_{BA}). \quad (12)$$

The rule that the average cluster coefficient changes with the increase of the network size ζ was proposed in [36]:

$$\Lambda_{BA} = \frac{\omega^2(\omega + 1)^2}{4(\omega - 1)} \left[\ln \frac{\omega + 1}{\omega} - \frac{1}{\omega + 1} \right] \frac{[\ln(\zeta)]^2}{\zeta}. \quad (13)$$

Therefore, the average clustering coefficient of the rule network does not change with increasing the network size.

B. NETWORK GENERATORS

The main process steps of the network generators are described in this section. The first step is to obtain a random undirected graph. Three models of randomly generated graphs were described in detail in Section II-A. To make the network generator structured and complete, a method that converts the generated graph to a trainable network is presented. The complexity of a graph is determined by the number of nodes and edges in a graph. The randomly generated graph applied in the network is defined as follows.

- ER algorithm: the initial state of the random graph contains V_{ER} nodes and 0 edges. Each pair of nodes of the random graph, is connected with a probability of P . The algorithm can be written as $ER(P)$. When $P > \ln(V_{ER})/V_{ER}$, the graph has a large probability of becoming a connected component.
- WS algorithm: in the initial state of the random graph, a total of V_{WS} nodes are distributed in circles. Every node of the random graph is connected to $Z/2$ neighboring nodes in total (Z is an even number). Nodes in the

random graph are traversed in clockwise order. For each node, examine the i th node connected to it. This node is reconnected with probability P . ‘‘Reconnect’’ means randomly selecting a node except itself to connect a node unconnected with the current node, and the connected edge is not a multiple edge. When $1 < i < Z/2$, a new random map is obtained after repeating the operation $Z/2$ times. This algorithm can be written as $WS(Z, P)$.

- BA algorithm: the initial state of the random graph contains Q ($1 \leq Q < V_{BA}$) nodes and 0 edges. For each node added, Q edges are added to the random graph. The newly added node is connected to the old nodes with a particular probability that is related to the degree of nodes. The algorithm terminates when the number of nodes increases to V_{BA} . This algorithm can be written as $BA(Q)$.

The second step converts the undirected graph into a directed acyclic graph (DAG). The graphs generated by the ER, BA, and WS algorithms are all random undirected graphs. The generated random undirected graph is converted to a DAG by a simple method: every node in the graph is assigned an index, and the direction of each edge is set to point from the node of the smaller index to the node of the larger index. Evidently, the directed graph generated by this algorithm has no cycle. The node indexing strategy of the ER algorithm is that the indices are assigned randomly. The node indexing strategy of the WS algorithm is that the order of the index is assigned in clockwise order. The node indexing strategy of the BA algorithm is that indices of Q nodes in the initial random graph are assigned from 1 to Q , and the remaining indices are assigned according to the order in which they are added to the graph.

The third step is to map the generated DAG to a trainable network. The network is composed of a data stream line for transmitting data and an arithmetic module for processing the data stream. The former forms a mapping with edges, and the latter forms a mapping with nodes. This is consistent with the intuitive understanding. The mapping process is divided into edge operations and node operations. Since our goal is mapping a directed graph to a computable neural network, edge operations are defined as passing data streams from one node to another. The node operations consist of the following three steps:

- Aggregation: the input data are summed based on their weights.
- Transformation: the results after aggregation are processed by transformation which contains a series of operations such as activation functions, 3×3 convolutions, and batch normalization (BN) [37].
- Distribution: the results after transformation are transmitted to the next node.

Node operations are shown in Fig. 1. The data are transmitted by the previous node through four input edges. Then, the learnable and positive weight parameters W_1, W_2, W_3 , and W_4 , which are processed by the sigmoid function to make them positive, are used to calculate the sum of the weights.

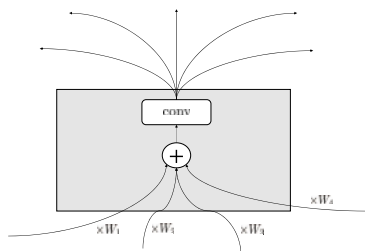


FIGURE 1. Schematic diagram of node operations.

In addition, the aggregated data are transformed by a series of operations such as activation functions, convolutions, and BN. Finally, data are output to the next node through five channels. The node operation achieves satisfactory results. Due to aggregation, the number of channels, both input and output, is limited.

A generating graph with edge operations and node operations is obtained. Since the number of input and output nodes is not determined, the graph cannot be called an effective neural network. The CNNBCN model is used for image classification, so we transform the generated graph into a network with explicit input and output. A short and effective method is used to achieve this goal. The approach is similar to streaming network technology. An input node and an output node are added to the graph. The original input nodes are all connected by the new input node and receive the input data in a module. Similarly, the original output nodes are all connected to the new output node. These two new input and output nodes are not convolved, and they are not counted in the total number of nodes V . An effective neural network is attained because of the single input nodes and output nodes. An effective neural network is often composed of multiple parts, and each part has a disparate structure with different tasks. Hence, it is necessary to obtain the feature map by downsampling to achieve network partitioning. The graph network generated above is considered as a module. A complete and effective neural network consists of multiple modules. Each module is connected to other modules through its input nodes or output nodes. Thus far, a CNNBCN model including several modules that can be directly input into training is obtained.

C. MODIFIED MODEL

According to the previous section, a CNNBCN model consisting of several modules is obtained. To optimize the generated model, a new combination of activation functions is applied. After many rigorous tests, the combination of activation functions composed of Gaussian error linear units (GeLUs) [38] and Rectified linear units (ReLUs) is applied to several modules of the model. The GeLU activation function is applied to the first and second modules, and the ReLU activation function is used in the remaining modules and the classifier. The GeLU nonlinearity is the expected transformation of a stochastic regularizer that randomly applies the identity or zero map to a neuron’s input. The GeLU nonlinearity weights are input by their magnitude, rather than gates input by their sign as in ReLU [38]. For $GeLU(x)$, which is assumed

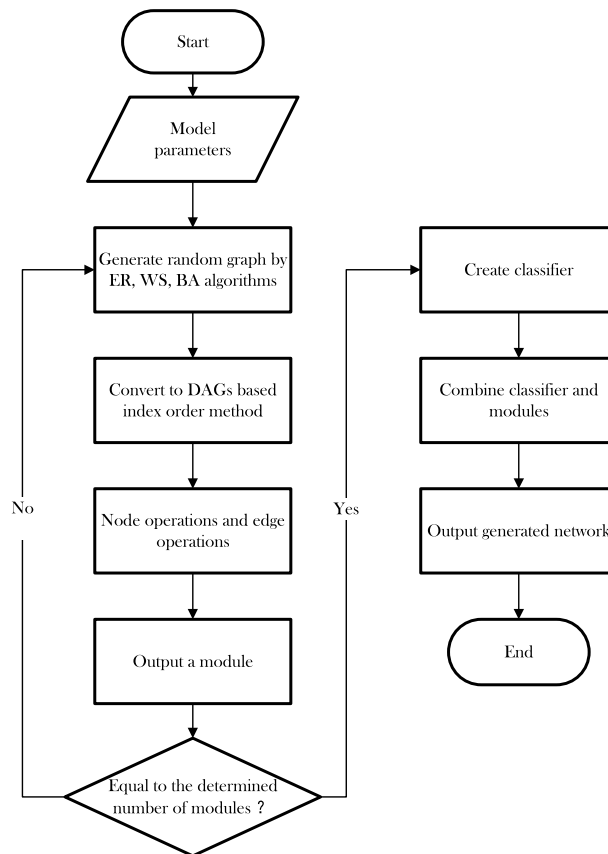


FIGURE 2. Flowchart for generating the CNNBCN model.

to be a standard normal distribution, a mathematical formula for approximate calculation is provided in [38]:

$$GeLU(x) = 0.5x(1 + \tanh[\sqrt{2/\pi}(x + 0.044715x^3)])$$

Finally, a modified CNNBCN model that can be directly put into training is generated.

III. STRUCTURE OF THE CNNBCN MODEL

In this section, the structure of the CNNBCN model is presented. A computable network is constructed by a network generator in Section II-B. The generated random graph is transformed into the DAG and then mapped into a neural network. The random graph used to construct the neural network is generated by ER, WS, and BA algorithms. It is easy to understand the complete generation process of the model in Fig. 2. A complete CNNBCN model is composed of several modules. The CNNBCN model is divided into simple mode and regular mode due to the complexity of the neural networks. The number of channels in data feature extraction is one of their differences. The former has 78 channels and the latter has 109 channels. The structure of the CNNBCN model is shown in Table 1. In the table, V refers to the number of nodes and C denotes the number of channels. The pixel size of the input is 224×224 after resizing.

The structure of the CNNBCN model is provided in Fig. 3. The neural network is mapped from a random graph, and all

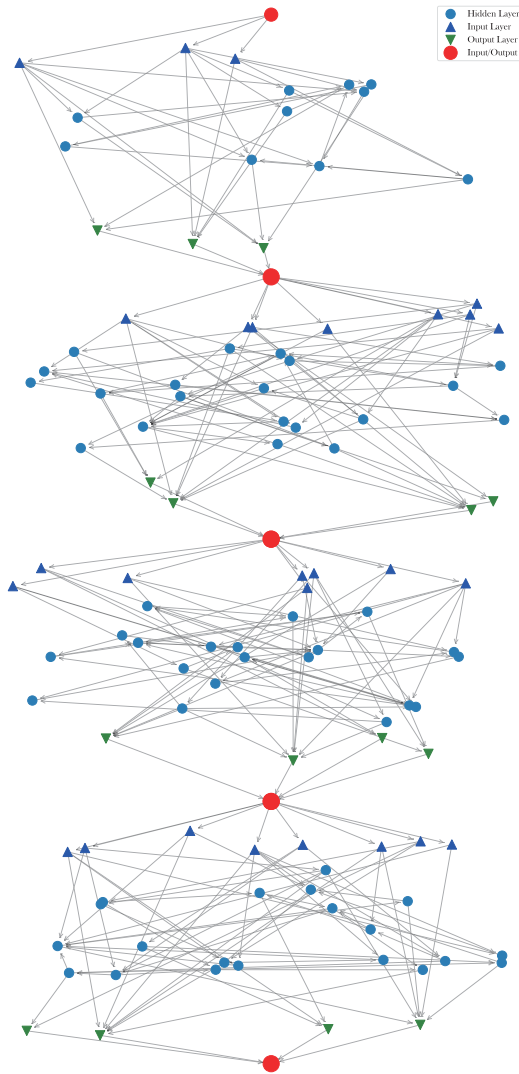


FIGURE 3. Structure of the CNNBCN model constructed by WS($Z = 4, P = 0.75$) algorithm.

nodes (except custom nodes) are randomly generated and distributed. To make the structure of the generated network more intuitive, the nodes of the input, output, and hidden layers are manually limited to be randomly generated within a certain range. The custom input and output nodes are represented by a large circular mark, and the input nodes of the random graph are represented by regular triangle mark. An inverted triangle means that the output nodes and random nodes in the network are indicated by small circles. The random graph of this model is constructed by the WS algorithm. Therefore, the first layer is input to the random graph network through module₁, and then passes through the four random graph networks in sequence. Finally, the final result is obtained after passing through the classifier. Therefore, the CNNBCN model is composed of several modules and a classifier is obtained.

IV. COMPARATIVE EXPERIMENTS

A brain tumor dataset provided from Nanfang Hospital, Guangzhou, China, and General Hospital, Tianjing Medical

TABLE 1. Structure of the CNNBCN model.

Modules	Output size	Simple mode	Regular mode
module ₁	(112, 112)	3×3 conv, C/2	3×3 conv, C/2
module ₂	(56, 56)	3×3 conv, C	graph block 3×3 conv, V/2, C
module ₃	(28, 28)	graph block 3×3 conv, V, C	graph block 3×3 conv, V, 2C
module ₄	(14, 14)	graph block 3×3 conv, V, 2C	graph block 3×3 conv, V, 4C
module ₅	(7, 7)	graph block 3×3 conv, V, 4C	graph block 3×3 conv, V, 8C
classifier	(1, 1)	1×1 conv 1280-d, 3-d fc	1×1 conv 1280-d, 3-d fc

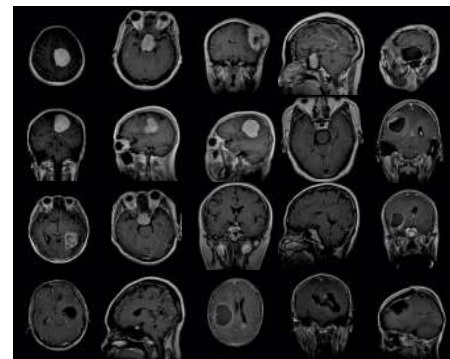


FIGURE 4. Sample dataset for three brain tumors.

University, China, from 2005 to 2010 is used for comparative experiments of the models. This dataset is collected from 233 patients and consists of 708 meningioma images, 1426 glioma images, and 930 pituitary tumor images [39], [40]. The images have an inplane resolution of 512 × 512 with a pixel size of 0.49 × 0.49 mm². In addition, the dataset is publicly available and can be obtained in [25]. Samples of some brain tumors are provided in Fig. 4.

The experiment consists of three parts. The first part is a comparison of the three generation algorithms of the original model and the modified model. To increase the credibility of the model and the persuasiveness of the experiment, the second part of the experiment is a comparison of other researchers’ models on brain tumor classification using the same dataset. The last part is the performance comparison of other effective image classification models. The number of nodes in the randomly generated graph is set as 32. In the comparison experiment, it is reasonable to choose a deeper neural network model, because a complete CNNBCN network consists of five modules and each module has 32 nodes. In addition, the parameters were set to ER($P = 0.2$), WS($Z = 4, P = 0.75$), and BA($Q = 5$) in the three graph generation algorithms. Due to the small number of classification labels, the model is set to a simple mode. The loss function of this experiment is selected as the cross-entropy loss function [41], [42]. The experimental results are shown in Table 2, from which the parameters are intended for evaluating

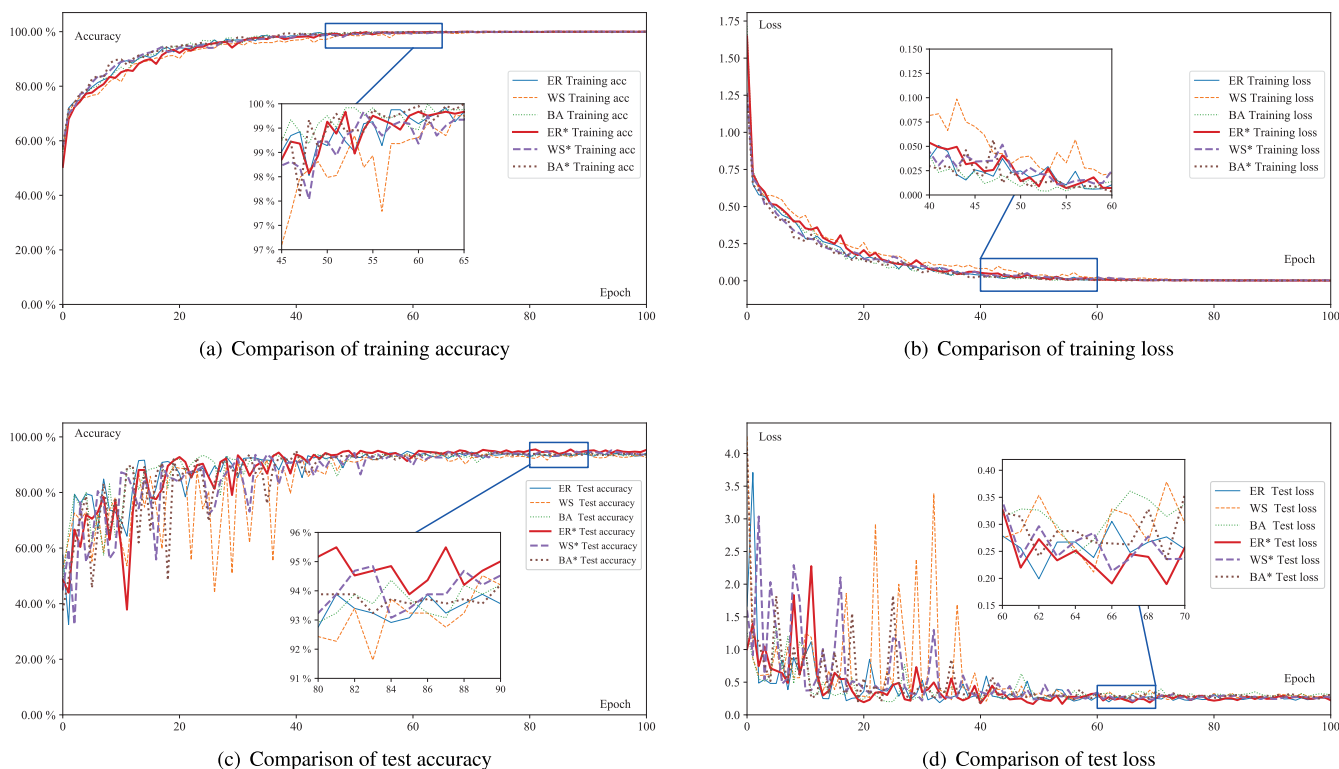


FIGURE 5. Training and test results of the original CNNBCN model and the modified CNNBCN model generated by three algorithms: ER, WS, and BA. (a) Comparison results of training accuracy of the modified CNNBCN model and the original CNNBCN model. (b) Comparison results of training loss of the modified CNNBCN model and the original CNNBCN model. (c) Comparison results of test accuracy of the modified CNNBCN model and the original CNNBCN model. (d) Comparison results of test loss of the modified CNNBCN model and the original CNNBCN model (* refers the modified CNNBCN model).

the model performance of the original CNNBCN model, the modified CNNBCN model and other models.

We separately tested the CNNBCN model and the modified CNNBCN model generated by the ER, WS, and BA algorithms for accuracy, loss of training and test for identifying the type of brain tumor. The modified CNNBCN model has significantly higher test accuracy than the original model. The highest test accuracy of the original model is 94.53%, the test accuracy of the modified model increases by 0.5% – 0.75% on average, and the highest accuracy reaches 95.49%. The modified CNNBCN model achieves satisfactory experimental results with the combination of GeLU and ReLU activation functions. Specifically, subfigures (a)-(d) in Fig. 5 provide more training and test information about the original model and the modified model, respectively. The modified CNNBCN model uses less training time to reach the convergence state, as can be seen in subfigures (a)-(b). Besides, the comparison results between the modified CNNBCN model and the original CNNBCN model in terms of test accuracy and test loss are given in subfigures (c)-(d) from which it is clear to see that the modified model is better than the original model.

Some models presented by other researchers for brain tumor image classification are provided in Table 2. Since these models use the same dataset as our models, they are in

appropriate contrast. The modified CNNBCN model achieves better results compared to other works. In addition, the test accuracy of the modified CNNBCN model and other performance parameters are better than those of the other presented models. The experimental results compared with other classic convolutional neural network models are shown at the bottom of Table 2. From this table, the results of training and test are visually presented. The modified CNNBCN model generated by the randomly generated graph algorithm achieves satisfactory results. Their training accuracy rates are all 100%, and the test accuracy rates are all above 95%. In comparison with other image classification models, the modified CNNBCN model also achieves a higher ranking. The test loss of the CNNBCN model is minimal in comparison with other image classification models. In particular, the training time of the modified CNNBCN model is lower than that of other convolutional neural network models of the same depth, such as ResNet-151 and DenseNet-161. In addition, the test accuracy of the modified CNNBCN model is higher than that of EfficientNet-b0 which is equipped with the strongest image classification performance in Table 2. The results show that the modified CNNBCN model generated by the randomly generated graph algorithm achieves satisfactory results in the classification of brain tumor images, which provides a feasible prospect for the construction of neural networks.

TABLE 2. Comparisons of training accuracy, test accuracy, training loss and test loss among the CNNBCN model and the modified CNNBCN models generated by three random graph generation algorithms, several models proposed by other work, and other image classification models.

Network	Training Accuracy	Test Accuracy	Minimum	Minimum	Training time (min)
	Top-1 Acc.	Top-1 Acc.	Training Loss	Test Loss	
CNNBCN-ER	100.00%	94.85%	1.43×10^{-3}	1.86×10^{-1}	46
CNNBCN-WS	100.00%	94.53%	6.61×10^{-4}	2.02×10^{-1}	44
CNNBCN-BA	100.00%	94.53%	1.43×10^{-3}	1.72×10^{-1}	49
CNNBCN-ER*	100.00%	95.49%	1.07×10^{-3}	1.69×10^{-1}	47
CNNBCN-WS*	100.00%	95.17%	1.46×10^{-3}	2.13×10^{-1}	42
CNNBCN-BA*	100.00%	95.01%	1.13×10^{-3}	1.71×10^{-1}	50
Model-1(Cheng et al.) [39]	-	91.28%	-	-	-
Model-2(Cheng et al.) [40]	-	94.68%	-	-	-
Model-3(Anaraki et al.) [44]	99.54%	94.20%	2.53×10^{-2}	1.82×10^{-1}	-
Model-4(Das et al.) [45]	99.85%	94.39%	5.34×10^{-3}	2.38×10^{-1}	-
Model-5(Liu et al.) [46]	-	95.00%	-	-	-
Model-6(Ghosal et al.) [47]	-	93.83%	-	-	-
ResNet-101 [26]	99.96%	91.63%	5.35×10^{-3}	2.89×10^{-1}	44
ResNet-152 [26]	99.92%	91.30%	5.00×10^{-3}	3.49×10^{-1}	56
DenseNet-121 [27]	100.00%	95.01%	6.31×10^{-4}	2.21×10^{-1}	69
DenseNet-161 [27]	100.00%	94.85%	8.53×10^{-4}	2.61×10^{-1}	49
ResNext-101 [48]	100.00%	94.53%	9.88×10^{-4}	2.08×10^{-1}	61
Wide-ResNet-101 [49]	99.51%	91.14%	1.72×10^{-2}	2.90×10^{-1}	95
MobileNet-v1 [50]	100.00%	93.88%	2.96×10^{-3}	2.23×10^{-1}	21
MobileNet-v2 [51]	100.00%	94.04%	1.01×10^{-3}	2.09×10^{-1}	25
ShuffleNet-v2 [52]	100.00%	93.88%	1.07×10^{-3}	2.18×10^{-1}	26
EfficientNet-B0 [53]	100.00%	95.17%	4.48×10^{-4}	2.13×10^{-1}	32

* refers the modified CNNBCN model.

V. CONCLUSION

The convolutional neural network based on complex networks with a modified activation function for image classification of brain tumors, abbreviated as CNNBCN, generated by ER, WS, and BA algorithms has been presented in this paper. Experimental results have shown that the classification accuracy of the original CNNBCN model and modified model are better than some manually designed neural networks. In addition, its performance is comparable to one of the best current image classification models. The CNNBCN model has not only achieved satisfactory results in the field of brain tumor image classification, but also provided a reference for the design of network structures. The construction of neural networks by using connectomes of animals and even humans will be considered in our future work.

REFERENCES

- [1] L. Jin, Z. Huang, L. Chen, M. Liu, Y. Li, Y. Chou, and C. Yi, "Modified single-output chebyshev-polynomial feedforward neural network aided with subset method for classification of breast cancer," *Neurocomputing*, vol. 350, pp. 128–135, Jul. 2019.
- [2] H. Li, Z. Huang, J. Fu, Y. Li, N. Zeng, J. Zhang, C. Ye, and L. Jin, "Modified Weights-and-Structure-Determination neural network for pattern classification of flatfoot," *IEEE Access*, vol. 7, pp. 63146–63154, 2019.
- [3] L. Jin, Z. Huang, Y. Li, Z. Sun, H. Li, and J. Zhang, "On modified multi-output chebyshev-polynomial feed-forward neural network for pattern classification of wine regions," *IEEE Access*, vol. 7, pp. 1973–1980, 2019.
- [4] L. Chen, Z. Huang, Y. Li, N. Zeng, M. Liu, A. Peng, and L. Jin, "Weight and structure determination neural network aided with double pseudo-inversion for diagnosis of flat foot," *IEEE Access*, vol. 7, pp. 33001–33008, 2019.
- [5] T. D. Pham, K. Wardell, A. Eklund, and G. Salerud, "Classification of short time series in early Parkinson's disease with deep learning of fuzzy recurrence plots," *IEEE/CAA J. Automatica. Sinica*, vol. 6, no. 6, pp. 1306–1317, Nov. 2019.
- [6] H. Lu, L. Jin, J. Zhang, Z. Sun, S. Li, and Z. Zhang, "New joint-drift-free scheme aided with projected ZNN for motion generation of redundant robot manipulators perturbed by disturbances," *IEEE Trans. Syst., Man, Cybern. Syst.*, early access, Dec. 19, 2019, doi: [10.1109/TSMC.2019.2956961](https://doi.org/10.1109/TSMC.2019.2956961).
- [7] Z. Xie, L. Jin, X. Du, X. Xiao, H. Li, and S. Li, "On generalized RMP scheme for redundant robot manipulators aided with dynamic neural networks and nonconvex bound constraints," *IEEE Trans. Ind. Informat.*, vol. 15, no. 9, pp. 5172–5181, Sep. 2019.
- [8] Z. Xie, L. Jin, X. Luo, S. Li, and X. Xiao, "A data-driven cyclic-motion generation scheme for kinematic control of redundant manipulators," *IEEE Trans. Control Syst. Technol.*, early access, Jan. 15, 2020, doi: [10.1109/TCST.2019.2963017](https://doi.org/10.1109/TCST.2019.2963017).
- [9] Y. Qi, L. Jin, H. Li, Y. Li, and M. Liu, "Discrete computational neural dynamics models for solving time-dependent Sylvester equations with applications to robotics and MIMO systems," *IEEE Trans. Ind. Informat.*, early access, Jan. 14, 2020, doi: [10.1109/TII.2020.2966544](https://doi.org/10.1109/TII.2020.2966544).
- [10] Y. Qi, L. Jin, Y. Wang, L. Xiao, and J. Zhang, "Complex-valued discrete-time neural dynamics for perturbed time-dependent complex quadratic programming with applications," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Nov. 8, 2019, doi: [10.1109/TNNLS.2019.2944992](https://doi.org/10.1109/TNNLS.2019.2944992).
- [11] J. Zhou, J. Wang, and J. Wang, "A simulation engine for stochastic timed Petri nets and application to emergency healthcare systems," *IEEE/CAA J. Automatica Sinica*, vol. 6, no. 4, pp. 969–980, Jul. 2019.
- [12] J. Zhang, L. Jin, and L. Cheng, "RNN for perturbed manipulability optimization of manipulators based on a distributed scheme: A game-theoretic perspective," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jan. 28, 2020, doi: [10.1109/TNNLS.2020.2963998](https://doi.org/10.1109/TNNLS.2020.2963998).

- [13] L. Wei, L. Jin, C. Yang, K. Chen, and W. Li, "New noise-tolerant neural algorithms for future dynamic nonlinear optimization with estimation on hessian matrix inversion," *IEEE Trans. Syst., Man, Cybern.*, early access, May 27, 2019, doi: 10.1109/TSMC.2019.2916892.
- [14] S. Bauer, C. May, D. Dionysiou, G. Stamatakos, P. Buchler, and M. Reyes, "Multiscale modeling for image analysis of brain tumor studies," *IEEE Trans. Biomed. Eng.*, vol. 59, no. 1, pp. 25–29, Jan. 2012.
- [15] M. Vaidya, R. D. Flint, P. T. Wang, A. Barry, Y. Li, M. Ghassemi, G. Tomic, J. Yao, C. Carmona, E. M. Mugler, S. Gallick, S. Driver, N. Brkic, D. Ripley, C. Liu, D. Kamper, A. H. Do, and M. W. Slutzky, "Hemicraniectomy in traumatic brain injury: A noninvasive platform to investigate high gamma activity for brain machine interfaces," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 27, no. 7, pp. 1467–1472, Jul. 2019.
- [16] M. Guerquin-Kern, L. Lejeune, K. P. Pruessmann, and M. Unser, "Realistic analytical phantoms for parallel magnetic resonance imaging," *IEEE Trans. Med. Imag.*, vol. 31, no. 3, pp. 626–636, Mar. 2012.
- [17] G. Mohan and M. M. Subashini, "MRI based medical image analysis: Survey on brain tumor grade classification," *Biomed. Signal Process. Control*, vol. 39, pp. 139–161, Jan. 2018.
- [18] J. Ker, S. P. Singh, Y. Bai, J. Rao, T. Lim, and L. Wang, "Image thresholding improves 3-Dimensional convolutional neural network diagnosis of different acute brain hemorrhages on computed tomography scans," *Sensors*, vol. 19, no. 9, p. 2167, 2019.
- [19] J. Ker, Y. Bai, H. Y. Lee, J. Rao, and L. Wang, "Automated brain histology classification using machine learning," *J. Clin. Neurosci.*, vol. 66, pp. 239–245, Aug. 2019.
- [20] J. Ker, L. Wang, J. Rao, and T. Lim, "Deep learning applications in medical image analysis," *IEEE Access*, vol. 6, pp. 9375–9389, 2018.
- [21] X. Bai, S. I. Niwas, W. Lin, B.-F. Ju, C. K. Kwoh, L. Wang, C. C. Sng, M. C. Aquino, and P. T. K. Chew, "Learning ECOC code matrix for multiclass classification with application to glaucoma diagnosis," *J. Med. Syst.*, vol. 40, no. 4, pp. 1–10, Apr. 2016.
- [22] B. He, J. Cui, B. Xiao, and X. Wang, "Image analysis by two types of franklin-Fourier moments," *IEEE/CAA J. Automatica Sinica*, vol. 6, no. 4, pp. 1036–1051, Jul. 2019.
- [23] S. Mitra and B. Uma Shankar, "Medical image analysis for cancer management in natural computing framework," *Inf. Sci.*, vol. 306, pp. 111–131, Jun. 2015.
- [24] S. Xie, A. Kirillov, R. Girshick, and K. He, "Exploring randomly wired neural networks for image recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1284–1293.
- [25] Cheng. (Jun. 2017). *Brain Tumor Dataset Figshare. Dataset*. [Online]. Available: <https://doi.org/10.6084/m9.figshare.1512427.v5>
- [26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
- [27] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4700–4708.
- [28] K. Jiseob and B. Zhang, "Random graph methods for incremental structure learning of neural network models," *J. Korean. Data. Inform. Sci. Soc.*, pp. 903–905, Jun. 2014.
- [29] S. Pal, Y. Dong, B. Thapa, N. V. Chawla, A. Swami, and R. Ramanathan, "Deep learning for network analysis: Problems, approaches and challenges," in *Proc. MILCOM IEEE Mil. Commun. Conf.*, Nov. 2016, pp. 588–593.
- [30] R. Albert and A.-L. Barabási, "Statistical mechanics of complex networks," *J. Rev. Mod. Phys.*, vol. 74, pp. 47–97, Jan. 2002.
- [31] D. J. Watts and S. H. Strogatz, "Collective dynamics of small-world networks," *Nature*, vol. 26, pp. 440–442, Jun. 1998.
- [32] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, Oct. 1999.
- [33] H. D. Ren, Z. H. Liu, and B. H. Wang, *Complex Systems and Complex Networks* Beijing, China: Higher Education Press, 2009, pp. 147–161.
- [34] A. Chakraborty and B. S. Manoj, "The reason behind the scale-free world," *IEEE Sensors J.*, vol. 14, no. 11, pp. 4014–4015, Nov. 2014.
- [35] C. T. Chou, "Extended master equation models for molecular communication networks," *IEEE Trans. Nanobiosci.*, vol. 12, no. 2, pp. 79–92, Jun. 2013.
- [36] M. Lei, L. Liu, and D. Wei, "An improved method for measuring the complexity in complex networks based on structure entropy," *IEEE Access*, vol. 7, pp. 159190–159198, 2019.
- [37] S. Wu, G. Li, L. Deng, L. Liu, D. Wu, Y. Xie, and L. Shi, "L1-Norm batch normalization for efficient training of deep neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 7, pp. 2043–2051, Jul. 2019.
- [38] D. Hendrycks and K. Gimpel, "Gaussian error linear units (GELUs)," 2016, *arXiv:1606.08415*. [Online]. Available: <http://arxiv.org/abs/1606.08415>
- [39] J. Cheng, W. Huang, S. Cao, R. Yang, W. Yang, Z. Yun, Z. Wang, and Q. Feng, "Enhanced performance of brain tumor classification via tumor region augmentation and partition," *PLoS ONE*, vol. 10, no. 10, 2015, Art. no. e0140381.
- [40] J. Cheng, W. Yang, M. Huang, W. Huang, J. Jiang, Y. Zhou, R. Yang, J. Zhao, Y. Feng, Q. Feng, and W. Chen, "Retrieval of brain tumors by adaptive spatial pooling and Fisher vector representation," *PLoS ONE*, vol. 11, no. 6, 2016, Art. no. e0157112.
- [41] Z. Zhang and M. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Montral, QC, Canada, 2018, pp. 8778–8788.
- [42] D. P. Kroese, R. Y. Rubinstein, and T. Taimre, "Application of the cross-entropy method to clustering and vector quantization," *J. Global Optim.*, vol. 37, no. 1, pp. 137–157, Dec. 2006.
- [43] P. Afshar, A. Mohammadi, and K. N. Plataniotis, "Brain tumor type classification via capsule networks," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 3129–3133.
- [44] A. K. Anaraki, M. Ayati, and F. Kazemi, "Magnetic resonance imaging-based brain tumor grades classification and grading via convolutional neural networks and genetic algorithms," *Biocybern. Biomed. Eng.*, vol. 39, no. 1, pp. 63–74, Jan. 2019.
- [45] S. Das, O. F. M. R. Aranya, and N. N. Labiba, "Brain tumor classification using convolutional neural network," in *Proc. Int. Conf. Adv. Sci., Eng. Robot. Technol. (ICASERT)*, Dhaka, Bangladesh, 2019, pp. 1–5.
- [46] D. Liu, Y. Liu, and L. Dong, "G-ResNet: Improved ResNet for brain tumor classification," in *Proc. Int. Conf. Neural Inf. Process. (ICONIP)*, Sydney, NSW, Australia, 2019, pp. 535–545.
- [47] P. Ghosal, L. Nandanwar, S. Kanchan, A. Bhadra, J. Chakraborty, and D. Nandi, "Brain tumor classification using ResNet-101 based squeeze and excitation deep neural network," in *Proc. 2nd Int. Conf. Adv. Comput. Commun. Paradigms (ICACCP)*, Feb. 2019, pp. 1–6.
- [48] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1492–1500.
- [49] J. Shi, Z. Li, S. Ying, C. Wang, Q. Liu, Q. Zhang, and P. Yan, "MR image super-resolution via wide residual networks with fixed skip connection," *IEEE J. Biomed. Health Informat.*, vol. 23, no. 3, pp. 1129–1140, May 2019.
- [50] Y. Wang, J. Yan, Q. Sun, J. Li, and Z. Yang, "A MobileNets convolutional network for GIS partial discharge pattern recognition in the ubiquitous power Internet of Things context: Optimization, comparison, and application," *IEEE Access*, vol. 7, pp. 150226–150236, 2019.
- [51] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.
- [52] N. Ma, X. Zhang, H. Zheng, and J. Sun, "ShuffleNet V2: Practical guidelines for efficient CNN architecture design," in *Proc. ECCV*, Munich, Germany, Sep. 2018, pp. 116–131.
- [53] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. ICML*, Long Beach, CA, USA, May 2019, pp. 6105–6114.



ZHIGUAN HUANG was born in Changning, Hunan, China, in 1980. He received the Ph.D. degree in human kinesiology from South China Normal University, Guangzhou, China, in 2009. He is currently an Associate Professor with the Guangdong Provincial Engineering Technology Research Center for Sports Assistive Device, Guangzhou Sport University. He has published around 30 articles in various journals and conferences. His current research interests include neural networks, sport biomechanics, and sport engineering.



XIAOHAO DU received the B.E. degree from Harbin Engineering University, Harbin, China, in 2019. He is currently pursuing the M.E. degree in computer technology with the School of Information Science and Engineering, Lanzhou University, Lanzhou, China. He is also with Guangzhou Sport University, Guangzhou, China, for cooperative research. His research interests include neural networks, computer vision, and reinforcement learning.



MEI LIU received the B.E. degree in communication engineering from Yantai University, Yantai, China, in 2011, and the M.E. degree in pattern recognition and intelligent system from Sun Yat-sen University, Guangzhou, China, in 2014. In 2017, she was a Lecturer with the College of Physics, Mechanical and Electrical Engineering, Jishou University, Jishou, China. She is currently a Teacher with the School of Information Science and Engineering, Lanzhou University, Lanzhou, China. Her main research interests include neural networks, computation, and optimization.



LIANGMING CHEN received the B.E. degree from Wuhan University, Wuhan, China, in 2017. He is currently pursuing the M.E. degree in computer technology with the School of Information Science and Engineering, Lanzhou University, Lanzhou, China. He is also with Guangzhou Sport University, Guangzhou, China, for cooperative research. His research interests include neural networks, pattern recognition, and broad learning.



YAO CHOU is currently pursuing the Ph.D. degree in electrical and computer engineering with Brigham Young University, USA. Her principal research interest is in machine learning with applications to computer vision and acoustical signal processing.



YUHE LI was born in Dabu, Guangdong, China, in 1967. He received the master's degree in human kinesiology from Guangzhou Sport University, Guangzhou, China, in 1997. He is currently a Professor with the Guangdong Provincial Engineering Technology Research Center for Sports Assistive Device, Guangzhou Sport University. He has published around 40 articles in various journals and conferences. His current research interests include exercise and health promotion, human kinesiology, and sports engineering.



LONG JIN (Member, IEEE) received the B.E. degree in automation and the Ph.D. degree in information and communication engineering from Sun Yat-sen University, Guangzhou, China, in 2011 and 2016, respectively. In 2017, he was a Postdoctoral Fellow with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong. He is currently a Full Professor with the School of Information Science and Engineering, Lanzhou University, Lanzhou, China. He is also with Guangzhou Sport University, Guangzhou, for cooperative research. His main research interests include neural networks, robotics, and intelligent information processing.

...