Heliyon 6 (2020) e05625

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Research article

Detecting pathological brain via ResNet and randomized neural networks

Siyuan Lu [a], Shui-Hua Wang [a,b,**], Yu-Dong Zhang [a,b,*]

[a] School of Informatics, University of Leicester, Leicester, LE1 7RH, UK
[b] Department of Information Systems, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, 21589, Saudi Arabia

ARTICLE INFO

Keywords:
Computer science
Magnetic resonance image
Computer aided diagnosis
Convolutional neural network
Randomized neural networks
Extreme learning machine
Random vector functional link
Schmidt neural network
Chaotic bat algorithm

ABSTRACT

Brain disease is one of the leading causes of death nowadays. Medical imaging is the most effective method for brain disease diagnosis, which provides a clear view of the interior brain. However, manual interpretation requires too much time and effort because medical images contain a large volume of information. Computer aided diagnosis is playing a more and more significant role in the clinic, which can help doctors and physicians to analyze medical images automatically. In this study, a novel pathological brain detection system was proposed for brain magnetic resonance images based on ResNet and randomized neural networks. Firstly, a ResNet was employed as the feature extractor, which was a famous convolutional neural network structure. Then, we used three randomized neural networks, i.e., the Schmidt neural network, the random vector functional-link net, and the extreme learning machine. The weights and biases in the three networks were trained by the chaotic bat algorithm. The three proposed methods achieved similar results based on five runs, and they yielded comparable performance in comparison with state-of-the-art approaches.

1. Introduction

Brain diseases are among the most severe health killers, and some of them are still incurable with modern technology. For example, Alzheimer's disease (AD) belongs to dementia that develops slowly among the elderly. AD affects the patient's thinking, behavior and memory gradually and leads to death eventually. However, early diagnosis of brain diseases can help control the illness conditions, prolong the patients' life and even cure them. So far, the most used brain diagnosis method is through medical imaging. Compared with other imaging modalities like computed tomography (CT) and X-ray, magnetic resonance image (MRI) is non-invasion and non-radioactive with a clearer view of soft tissues. So, it is often used in brain imaging. Traditional manual interpretation and analysis of the massive information in brain MRI require too much time and effort, and weak in reproducibility. Fortunately, with the development of artificial intelligence, computer aided diagnosis (CAD) is becoming an important tool in clinical diagnosis. CAD systems are used for automatic medical image analysis, such as segmentation [1, 2], classification [3, 4] and data fusion [5, 6, 7, 8].

Pathological brain detection (PBD) is a branch of CAD that focuses on brain diagnosis based on MRIs. Conventionally, a PBD system contains four procedures: feature extraction, feature reduction, classifier training and testing. As MRIs are large in volume, it will cause the curse of dimensionality if every intensity value of a pixel is fed into the classifier for training. Hence feature extraction is an important step to generate some representations from the images. In feature reduction or selection, the number of features is further reduced by some strategy like principal component analysis (PCA). Afterward, a classifier should be trained using the features as the input and the labels as the expected output. Finally, the trained classifier is tested on a testing set for performance evaluation. With the aim to improve the accuracy and reduce the running time of PBD systems, researchers have devoted time and energy to this field.

Chaplot, et al. [9] employed wavelet transform as the feature extractor and chose the self-organization map (SOM) and support vector machine (SVM) as classification algorithms. Experiment results showed that SVM outperformed SOM in terms of accuracy. Nevertheless, their feature vector contained thousands of features which increased the computational complexity. El-Dahshan, et al. [10] utilized discrete wavelet transform (DWT) to generate features and leveraged PCA for feature reduction. For classifier, feedforward back propagation neural network and k nearest neighbors were selected. Both classifiers achieved over 95% accuracy in the experiment. Zhang, et al. [11] suggested using feedforward neural network and optimize its parameters by chaotic artificial bee colony algorithm. Zhang and Wu [12] employed kernel

* Corresponding author.
** Corresponding author.
E-mail addresses: shuihuawang@ieee.org, shuihuawang@ieee.org (S.-H. Wang), yudongzhang@ieee.org (Y.-D. Zhang).

https://doi.org/10.1016/j.heliyon.2020.e05625
Received 16 December 2019; Received in revised form 1 March 2020; Accepted 25 November 2020

SVM for classification and compared the performance of SVMs using different kernel functions. K-fold cross validation (CV) was utilized to evaluate out-of-sample performance. Kalbkhani, et al. [13] proposed to model the coefficients of DWT by generalized autoregressive conditional heteroscedasticity and leverage PCA and linear discriminant analysis for feature reduction. For classification, *k* nearest neighbors and SVM were used. Padma and Sukanesh [14] put forward a PBD system based on brain CT. They extracted wavelet coefficients and texture features, and selected only six features to form feature vectors by student's *t*-test. The obtained feature vectors were fed into an SVM for training and testing. Saritha, et al. [15] employed wavelet entropy-based spider web plots as the feature extractor, which was a geometric structure. A probabilistic neural network was trained to classify brain MRIs as pathological or normal. Zhang, et al. [16] proposed to utilize DWT and PCA to extract features from brain MRI and employ kernel SVM for classification. The parameters in kernel SVM was optimized by particle swarm optimization (PSO) which belongs to a swarm intelligent algorithm. Yang, et al. [17] employed a novel method called wavelet energy for feature extraction and optimized the weights in SVM by biogeography-based optimization. Ahmmed, et al. [18] firstly used K-means and modified fuzzy C-means for the segmentation of brain MRI. Then, an SVM was trained to detect pathological brains from normal ones, and an artificial neural network was trained to classify the samples into five categories: benign and four malignant stages. Armato, et al. [19] proposed a deep learning method to predict the survival time based on brain MRI. They used a pre-trained deep convolutional neural network (CNN) for feature extraction, and introduced transfer learning for fine-tuning of the fully connected layers. Gilanie, et al. [20] combined Gabor filter and SVM for detecting pathological brain in MRI. Firstly, local histogram equalization was performed on brain MRI for preprocessing. Then, the Gabor texture descriptor and co-occurrence matrix were employed to generate features. Finally, the combined feature vector was used to train an SVM for prediction. Gurusamy and Subramaniam [21] utilized various filters for brain MRI de-noising and leveraged Otsu and clustering for tumor segmentation. Afterward, DWT and PCA were performed for feature extraction and reduction, respectively. The classifier was a kernel SVM. Islam and Zhang [22] constructed an eleven layer CNN for AD detection in brain MRI. Their system was capable of classifying the samples into four classes: nondemented, mild AD, very mild AD and moderate AD. Korolev, et al. [23] proposed a deep learning based method for AD detection. Inspired by VGG, they constructed VoxCNN, and ResNet was also used for comparison. However, their dataset was a small one, which is not suitable for training deep CNN. Kumar, et al. [24] applied the genetic algorithm for feature extraction after DWT and PCA. They selected SVM to label the samples as pathological and normal. Lahmiri [25] proposed a hybrid method for glioma detection. He firstly segmented brain MRIs by PSO methods: classical PSO, Darwinian particle swarm optimization (DPSO), and fractional-order DPSO (FODPSO). Then, image features were generated using the directional spectral distribution signature and generalized Hurst exponents. Finally, an SVM was trained to label the input samples as healthy or glioma. Results showed that the proposed approach outperformed a bunch of state-of-the-art. Lebedeva, et al. [26] extracted structure information from brain MRI to form the feature vector and employed the random forest algorithm to distinguish mild cognitive impairment from dementia. Nayak, et al. [27] proposed to leverage contrast limited adaptive histogram equalization for brain MRI enhancement and employ a novel orthogonal discrete ripplet-II transform for feature extraction. Extreme learning machine (ELM), a randomized neural network (RNN), served as a classifier. The parameters in ELM were optimized with an improved Jaya algorithm. The proposed method performed better than back propagation neural network (BPNN) and conventional ELM. Usman and Rajpoot [28] calculated wavelet texture, intensity information and local information to form the feature vector. Random forest algorithm was used to classify the samples into five categories. Ateeq, et al. [29] proposed a cerebral microbleed detection system for brain MRI. They first removed the skull in images and obtain potential candidates using thresholding. Then, a bunch of shapes based features and transform based features were extracted. Finally, quadratic discriminant analysis and SVM were trained for CMB identification. Hasan [30] employed PSO to find the cores of pathology in brain MRI and extracted the pathological areas using an active contour method. Islam and Zhang [31] suggested predicting different stages of AD based on CNN. They trained three identical CNNs with dense blocks and generated the output by counting votes. Li, et al. [32] proposed to predict pathological brain connectome patterns in functional brain MRI. They trained a stacked sparse autoencoder to learn from healthy connectivity patterns and transferred it to a transfer learning model for classification. Rieke, et al. [33] not only used CNN for AD classification but also tried to interpret the results from CNN by visualization. Zurita, et al. [34] proposed an SVM based CAD system for multiple sclerosis detection. Four different features were extracted including fractional anisotropy maps, structural connectivity, functional connectivity, and a combination of the former two connectivity. Sajjad, et al. [35] utilized a deep learning method to segment brain MRIs. A CNN structure was trained by the segmented images and fine-tuned on the augmented dataset [36]. constructed a ResNet 34 based CNN architecture and trained it in a three steps manner with transfer learning and augmented samples. The developed system achieved 100% accuracy for classifying abnormal brains and normal controls in their experiment.

From the abovementioned, we found that these methods achieved good results, but there are still some problems remained. Firstly, many systems used DWT for feature extraction, but the DWT features are

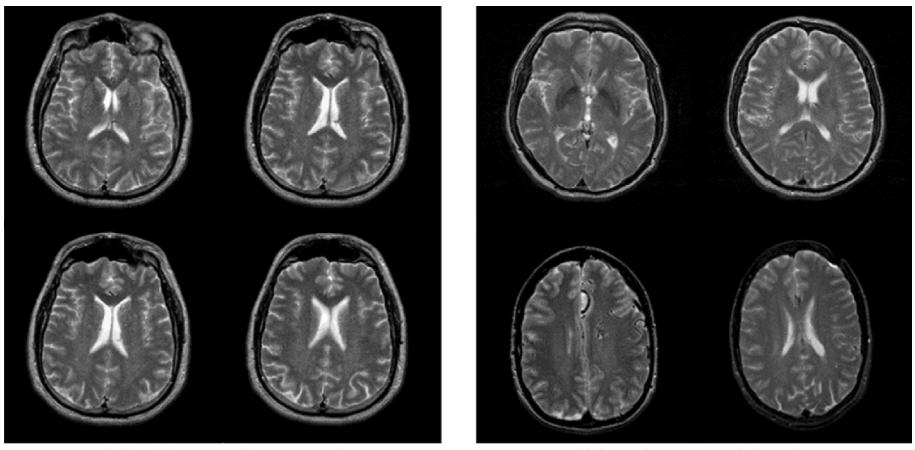

Figure 1. MRI samples.

domain dependent, which may not work on new samples. Secondly, MRI datasets are usually small, which often consist of hundreds of samples, so they are not suitable for training deep learning models which will cause overfitting problem. Last but not least, the performance of classifiers can be improved in terms of both training time and accuracy. To overcome these problems, we proposed our PBD methods based on ResNet-50 and randomized neural networks. ResNet [37] is a deep CNN model, it used residual function in training and achieved good classification results on various open datasets. In this study, we utilized a pre-trained ResNet-50 model for extracting brain MRI features instead of training ResNet on the MRI dataset. Then, the obtained feature was fed into three randomized neural networks for training and testing. We optimized the randomized neural networks parameters by chaotic bat algorithm (CBA) in order to boost its classification performance. CBA is an improved form of bat algorithm which was inspired by the flying, avoiding obstacles and prey of bats. The Chaotic map is introduced to better exploring the solution space in order to reach the global best solution.

The organization of the rest of this work is as follows. Section 2 presents the MRI datasets. Section 3 provides detailed methods, including ResNet, three randomized neural networks and CBA. Section 4 is about the experiment settings. Section 5 gives the experiment results and discussion. Finally, the conclusion is in Section 6.

2. Material

We obtained brain T2 weighted MRIs from the Whole Brain Atlas-Harvard Medical School (HMS) dataset, which is an open dataset. The slices are selected in axial orientation by experts of ten years' experience. The slices are resized in 256 × 256 pixels. The diseases of abnormal samples include AD, glioma, sarcoma, cerebral calcinosis, Huntington, etc. Originally, we obtained 177 pathological samples and only 28 normal samples, which is imbalanced. Firstly, 14 pathological and 14 normal samples were randomly selected to form the test set. Then, the rest 163 pathological and 14 normal samples form the training set. To balance the training set, resample was employed to generate 168 normal samples, i.e., the 14 normal samplings were copied for 11 times. So, we obtained a training set with 163 pathological and 168 normal samples at last. Figure 1 presented some samples in our dataset.

3. Methods

PBD systems usually start with feature extraction and use the feature to train a classifier for automated diagnosis. However, manual feature extraction is tedious, and the features are domain dependent, which means the features work on some datasets but fails on the others. The rapid development of deep learning and applications sheds light on automatic representation and feature learning. Therefore, we employed ResNet for feature extraction, which is a deep CNN structure. CNN implemented automated feature learning from low level to high level through convolution and pooling. The feature vectors were sent to randomized neural networks for classification training. Randomized neural networks converge much faster than conventional BPNN because they are based on gradient descent methods. CBA was used to optimize the parameters in randomized neural networks to improve its classification performance further.

3.1. ResNet

ResNet was proposed by He, et al. [37]. With residual learning mechanisms, deep CNN can be trained more effectively. Traditionally, a CNN contains three types of layers: convolution layer, pooling layer and fully connected layer.

Convolution operation generates feature maps from the input by a set of filters with trainable weights. Figure 2 presented a toy example of convolution, in which a 4 × 4 image was convoluted by a 2 × 2 filter with stride 2, and the feature map is 2 × 2. Given an image I in size of (M, N) and a filter F in size of (p, q), the convolution formula can be expressed as

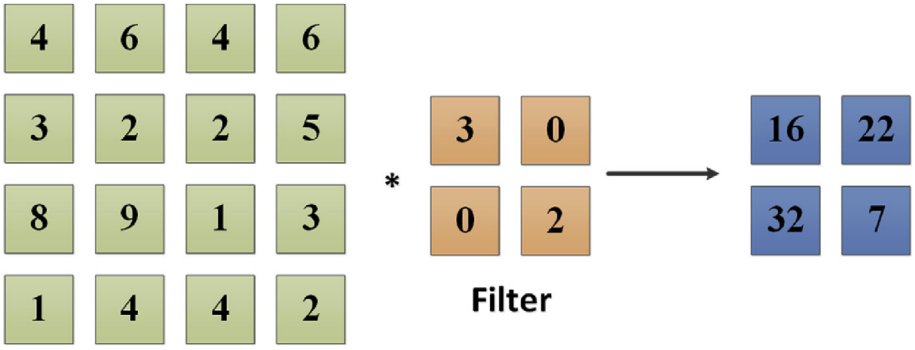

Figure 2. Convolution operation.

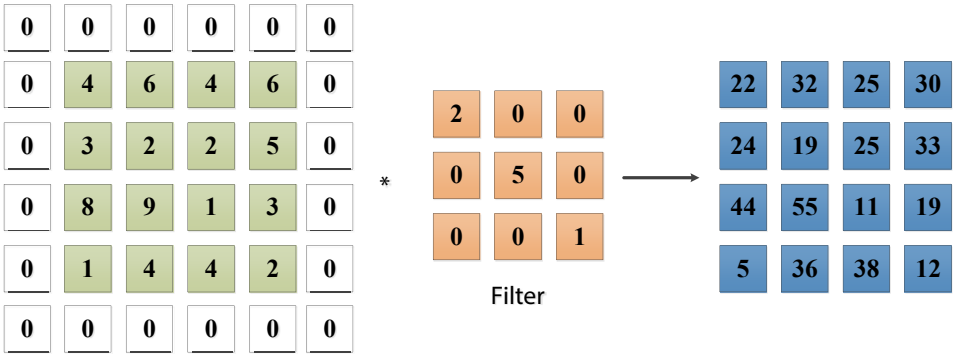

Figure 3. Convolution with zero padding.

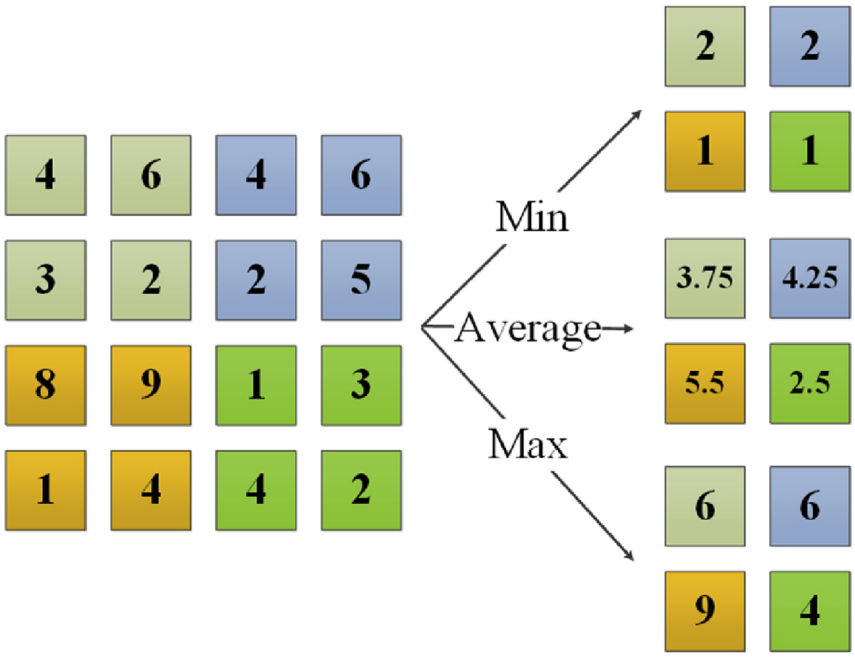

Figure 4. Pooling methods.

$$conv = (I * F)(x, y) = \sum_{M}\sum_{N} I(x-p, y-q)F(p, q) \tag{1}$$

The filters sweep from the top left to bottom right with some stride, and the feature maps can be computed. But the map size will shrink inevitably and the edge information can be lost during convolution. As in Figure 2, the input was 4 × 4, but the feature map was 2 × 2. To handle this problem, zero padding can be employed. Zero padding adds zero valued pixels around the input image before convolution, shown in Figure 3. A 4 × 4 image with 1 × 1 padding was convoluted by a 3 × 3 filter with 1 stride, and a 4 × 4 feature map was obtained. The detailed relationship of input size and feature map size is given below:

$$\begin{cases} \text{height}_{\text{map}} = \dfrac{(\text{height}_{\text{input}} - \text{height}_{\text{filter}} + 2 \times \text{padding})}{\text{stride}} + 1 \\ \text{width}_{\text{map}} = \dfrac{(\text{width}_{\text{input}} - \text{width}_{\text{filter}} + 2 \times \text{padding})}{\text{stride}} + 1 \end{cases} \tag{2}$$

Pooling is a simple but effective operation, and pooling layers are usually placed after convolution layers. Pooling generates feature maps with a local perceptive field. Pooling layers can extract main features from the input and reduce the dimension, which helps accelerate the training and improve the generalization ability. An example is given in Figure 4, which illustrates three pooling strategies: min, average and max with the local perceptive field of 2 × 2.

The fully connected layers are usually arranged at the end of a CNN structure, which was used for classification and recognition. Every node in fully connected layers is linked to all the nodes in the adjacent fully connected layers with trainable weights, shown in Figure 5.

The activation function is an important part of neural networks, which offers nonlinearity. The network may become a linear system composed of matrix multiplications without activation functions. In classical BPNN, the sigmoid function was the most used, which can be expressed as:

$$S(x) = \frac{1}{1 - e^{-x}} \tag{3}$$

It is widely used because BPNN is trained by gradient descent algorithms, and the gradient of sigmoid function is computationally effective, which can be calculated by:

$$S'(x) = S(x)(1 - S(x)) \tag{4}$$

However, in deep learning architecture, the sigmoid function doesn't work anymore because it causes the gradient vanishing problem in which the learning speed was very slow. So, in deep CNN, the rectified linear unit (ReLU) is often a good option. The formula of ReLU is:

$$\text{ReLU}(x) = \max(x, 0) \tag{5}$$

For positive input, the output of ReLU is always 1, and for negative input, the output is always 0. ReLU is simple to compute and it can accelerate the forward progress in training. For the final classification layer, Softmax function is often used:

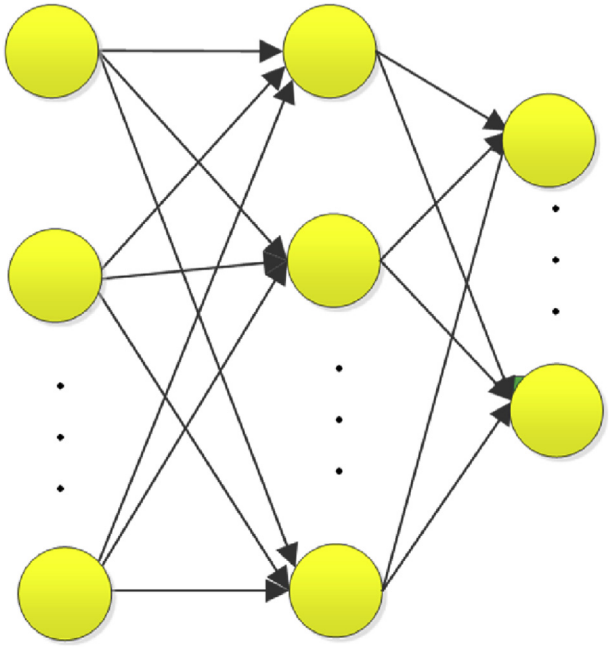

Figure 5. Fully connected layers.

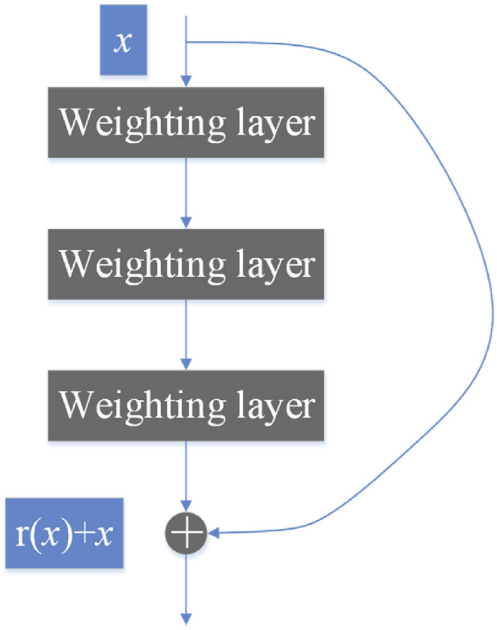

Figure 6. Residual block.

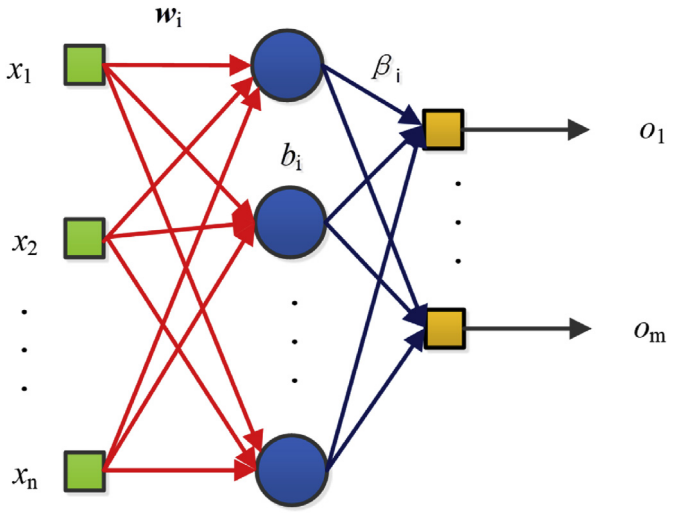

Figure 7. ELM structure.

$$\mathrm{softmax}(\mathbf{x})_i = \frac{\exp(x_i)}{\sum_{j=1}^{n}\exp(x_j)} \tag{6}$$

Softmax function maps the input vector into probabilities, so the overflow can be avoided.

ResNet was proposed by introducing the residual learning method. The idea is that network layers are capable of approximating any function asymptotically. For instance, $f(x)$ denotes the learned mapping of several layers, then it is equivalent to train these layers to approximate the residual function: $r(x) = f(x)$-x. So the target function becomes:

$$f(x) = r(x) + x \tag{7}$$

The analysis suggested that nonlinear layers may be difficult to reach identity mapping. So, if the identity mapping is optimal, the weights of theses stacked layers will be simply driven to zero with residual learning block, shown in Figure 6.

ResNet has been widely evaluated on a variety of open datasets including CIFAR 10, ImageNet classification, and COCO. The residual learning mechanism is a generic, easy to optimize and effective training method for deep CNN architectures. The ResNet-50 in our experiment was pre-trained by stochastic gradient descent (GSD) on a subset of ImageNet, which is used in the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC).

3.2. Randomized neural networks

ResNet is an effective tool for solving computer vision tasks, but our pathological brain dataset is too small to train such a giant structure because overfitting is inevitable. Hence, ResNet was only used for feature extraction in this study, and for classification, we use three types of randomized neural networks, i.e., the ELM, SNN and RVFL, as summarized below.

3.2.1. Extreme learning machine (ELM)

ELM was a training algorithm for single hidden layer feedforward network, proposed by Guang-Bin, et al. [38] in 2006, shown in Figure 7. BPNN is the most famous classical feedforward neural network, but it often stops at local extrema because gradient descent methods are greedy algorithms and they cannot jump out of local best solutions. The performance of BPNN is also sensitive to the learning rate. Compared with BPNN, which is trained by iteration, ELM learns faster because it doesn't depend on gradient descent. At the same time, the generalization ability of ELM is also good because the parameters are more likely to reach the global best solution [39, 40]. Generally, the training of ELM can be described as follows.

Given a training set M:

$$M = \{(\mathbf{x}_i, \mathbf{t}_i) | \mathbf{x}_i \in R^n, \mathbf{t}_i \in R^m, i = 1, ..., N\} \tag{8}$$

where $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{in})^T \in R^n$ is the sample input and $\boldsymbol{t}_i = (t_{i1}, t_{i2}, \ldots, t_{im})^T \in R^m$ denotes sample target, and a network with $\widehat{N}$ hidden nodes, the output of the ELM can be expressed as:

$$\sum_{i=1}^{\widehat{N}} \beta_i g_i(\boldsymbol{w}_i \boldsymbol{x}_j + b_i) = \boldsymbol{o}_j, j = 1, \ldots, N \tag{9}$$

where $\boldsymbol{w}_i = (w_{i1}, w_{i2}, \ldots, w_{in})^T$ denotes the input weight and b_i represents the bias of hidden layer, $g(x)$ is the activation function in the hidden layer, and $\beta_i = (\beta_{i1}, \beta_{i2}, \ldots, \beta_{im})^T$ is the output weight. The training purpose is to achieve:

$$\sum_{i=1}^{\widehat{N}} \beta_i g_i(\boldsymbol{w}_i \boldsymbol{x}_j + b_i) = \boldsymbol{t}_j, j = 1, \ldots, N \tag{10}$$

The above Eq. (10) can be abbreviated as

$$\mathbf{H}\beta = \boldsymbol{T} \tag{11}$$

where

$$\mathbf{H}(\boldsymbol{w}_1, \ldots, \boldsymbol{w}_{\widehat{N}}, b_1, \ldots, b_{\widehat{N}}, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_N) = \begin{bmatrix} g(\boldsymbol{w}_1 \boldsymbol{x}_1 + b_1) & \cdots & g(\boldsymbol{w}_{\widehat{N}} \boldsymbol{x}_1 + b_{\widehat{N}}) \\ \vdots & \ddots & \vdots \\ g(\boldsymbol{w}_1 \boldsymbol{x}_N + b_1) & \cdots & g(\boldsymbol{w}_{\widehat{N}} \boldsymbol{x}_N + b_{\widehat{N}}) \end{bmatrix}_{N \times \widehat{N}} \tag{12}$$

Table 1. ELM algorithm.

Input: training set M in Eq. (8)
Step 1: initialize the input weight w_i and bias b_i randomly. Step 2: calculate the output matrix H using Eq. (12). Step 3: calculate the output weight β by pseudo inverse in Eq. (14).
Output: the trained ELM structure

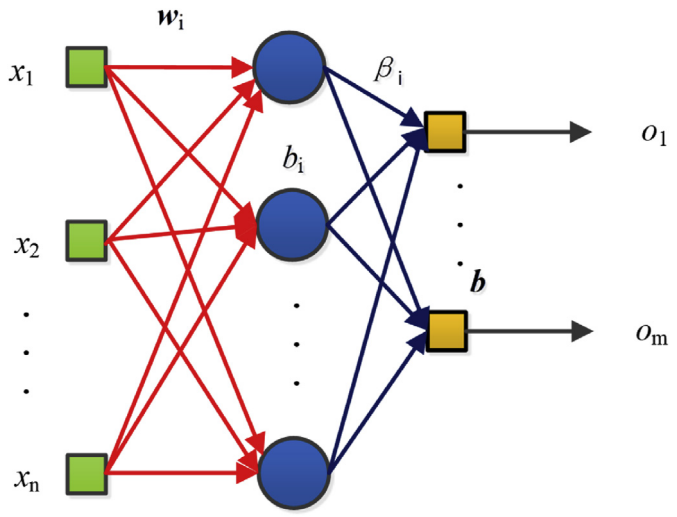

Figure 8. Structure of SNN.

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_{\widehat{N}}^T \end{bmatrix}_{\widehat{N}\times m}, \boldsymbol{T} = \begin{bmatrix} \boldsymbol{t}_1^T \\ \vdots \\ \boldsymbol{t}_N^T \end{bmatrix}_{N\times m} \tag{13}$$

By universal approximation theorem, any network with the single hidden layer can approximate the training samples asymptotically. Nevertheless, how to find optimal $\boldsymbol{w}_i$, b_i and β_i remains the challenge.

ELM consists of the following three steps shown in Table 1.

Pseudo inverse:

$$\beta = \mathbf{H}^{\dagger}\mathbf{T} \tag{14}$$

where $H^{\dagger}$ denotes the Moore-Penrose of **H**.

All the parameters in ELM were determined in only three steps without iteration. Its generalization is also good because its parameters are easy to optimize compared with SVM [41].

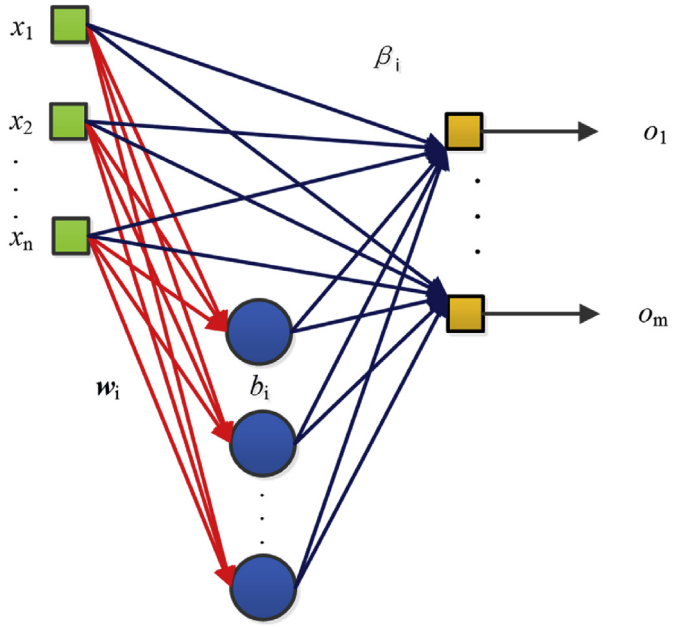

Figure 9. Architecture of RVFL.

Table 2. Ideal rules of bat algorithm.

All the bats can distinguish object and background somehow, and they navigate by echolocation.
The bats can adjust their pulse frequencies and emission rate according to their distances to the object.
The loudness of their pulse decreases from a large constant to a predefined minimum.

3.2.2. Schmidt neural network (SNN)

Schmidt neural network (SNN), was proposed by Schmidt, et al. [42] in 1992. SNN and ELM are non-iterative training algorithms for SLFN [43]. The only difference between SNN and ELM is that the bias of the output neurons in SNN may be non-zero, whereas the bias of the output neurons in ELM is always zero. The structure of SNN is shown in Figure 8. The output of SNN can be obtained by

$$\sum_{i=1}^{\widehat{N}} \left[\beta_i g_i\left(\boldsymbol{w}_i \boldsymbol{x}_j + b_i\right)\right] + \boldsymbol{b} = \boldsymbol{o}_j, j = 1, \ldots, N \tag{15}$$

3.2.3. Random vector functional link (RVFL)

RVFL was developed by Pao, et al. [44] in 1994. The only difference between RVFL and ELM is that RVFL has direct links from the input layer to the output layer, as shown in Figure 9, whereas ELM does not have such direct links.

Through a very comprehensive evaluation by using 121 UCI classification datasets, Zhang and Suganthan [45] showed that the direct link in RVFL improves performance, while the bias term in the output neuron (in SNN) had no statistically significant effect. Similar conclusions were drawn by Ren, et al. [46] for time series prediction.

3.3. CBA

The original randomized neural networks used random input weights and bias, which has a bad effect on classification performance. Researchers noticed this issue and proposed variants to obtain better performance [47, 48, 49]. In this paper, we proposed to employ the chaotic bat algorithm to optimize the weights and biases in the three randomized neural networks. The pure bat algorithm (BA) was invented by Yang [50], inspired by the echolocation of bats. As a swarm optimization algorithm, BA is better than PSO and genetic algorithm. A set of bat particles with each containing a potential solution was utilized to search the solution space by ultrasound with different frequencies and loudness. CBA was built upon a set of ideal rules for simplicity in Table 2:

The Chaotic mechanism was utilized in CBA to achieve better optimization capability and improve the global search ability [51]. It is used to update the position of bats in our algorithm. There are several chaotic strategies.

Cubic chaotic:

$$x_{k+1} = 3x_k\left(1 - x_k^2\right) \tag{16}$$

where k denotes the iteration times.

Gaussian chaotic:

$$x_{k+1} = \begin{cases} 0, & x_k = 0 \\ \dfrac{1}{x_k} - \left\lfloor \dfrac{1}{x_k} \right\rfloor, & x_k \in (0,1) \end{cases} \tag{17}$$

where the $\lfloor x \rfloor$ denotes the largest integer no more than x.

Logistic chaotic:

$$x_{k+1} = rx_k(1 - x_k) \tag{18}$$

where r denotes a positive integer, and the graph is presented in Figure 10.

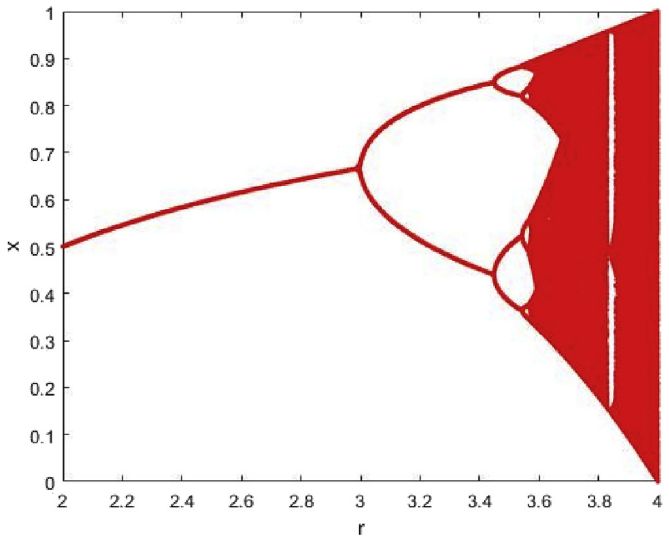

Figure 10. Logistic chaotic map.

Given a fitness function min $g(x)$, and the solution vector x:

$$x = (x_1, x_2, \ldots, x_d)^{\mathrm{T}} \tag{19}$$

The training steps of CBA are presented in Table 3.

There are three important types of operations in CBA, which are listed below.

Updating the parameters in bats using the chaotic mechanism, including pulse frequency, velocity and position:

$$f_i = f_{\min} + (f_{\max} - f_{min}) \times \beta \tag{20}$$

$$v_i^t = v_i^{t-1} + \left(x_i^t - x^*\right) \times f_i \tag{21}$$

$$x_i^t = x_i^{t-1} + v_i^t + w \times chaotic\left(x_i^{t-1}\right) \tag{22}$$

where f_i represents the searching frequency of the i^{th} bat, β varies from 0 to 1 randomly, v_i^t and x_i^t denote the velocity and position of the i^{th} bat in the t^{th} iteration, respectively, *chaotic*(x) means a chaotic strategy, which can be like in Eq. (16) (17) (18), and w is the weight factor for chaotic (see Figure 11)

Generating a new solution from the current best one:

$$x_{new} = x_{old} + \varepsilon A^t \tag{23}$$

Table 3. Pseudocode of CBA steps.

Input: fitness function min $g(x)$

Step 1: random initialization of the parameters in bats, including: position, velocity, searching frequency range [$f_{\min}$, $f_{\max}$], the max pulse rate R_0, max pulse loudness A_0, frequency enhancement factor γ, the loudness attenuation factor α, and max iteration *i_max*.
While (iteration < *i_max*)
{
 	
Step 2: compute the fitness of every bat using their x_i and get the current best solution by sort.
Step 3: bats parameter updating using chaotic method (in Eqs. (20)–(22)).
Step 4: assign a random number to *rand*
Step 5: if *rand* >γ_i, generate a new solution around the current best solution (in Eq. (23)).
Step 6: if *rand* < A_i and $f(x_i)$<$f(x^*)$, update the loudness and rate (in Eqs. (24), (25)).
}

Output: the optimal solution

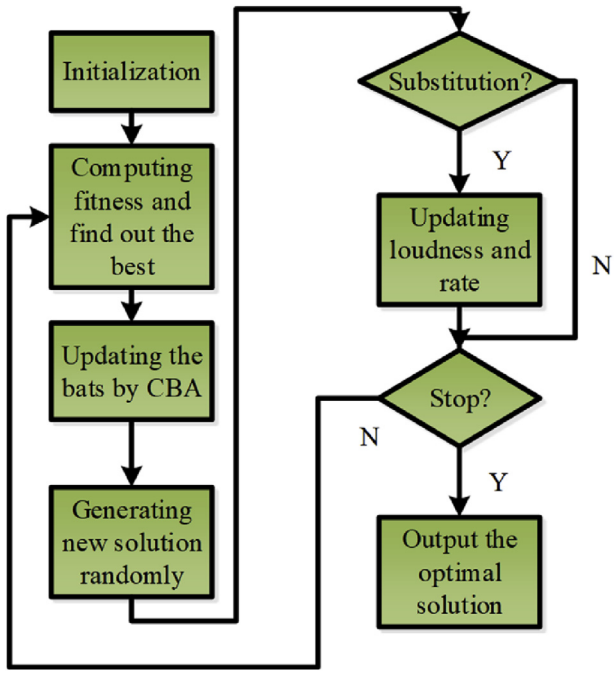

Figure 11. Diagram of CBA.

where x_{new} and x_{old} denotes new solution and the current best solution by bats, respectively, ε varies from -1 to 1, and A^t represents the average bats loudness in the t^{th} iteration.

Updating the loudness and pulse rate:

$$A_i^{t+1} = \alpha \times A_i^t \tag{24}$$

$$r_i^{t+1} = R_0 \times (1 - e^{-\gamma t}) \tag{25}$$

where A_i^t and r_i^t represent the loudness and pulse emission rate of the i^{th} bat in the t^{th} iteration, respectively.

3.4. ResNet-RNNs-CBA

In this paper, we proposed three PBD methods named ResNet-SNN-CBA, ResNet-RVFL-CBA and ResNet-ELM-CBA. A pre-trained ResNet-50 was firstly employed for feature extraction from brain MRIs. The output of the last forth layer in ResNet was extracted by training images and test

Table 4. Our PBD algorithms.

Input: training set and testing set
Step 1: A ResNet-50 pre-trained on ILSVRC was loaded. Step 2: The ResNet-50 was divided into two parts: the last three layers and the rest of ResNet-50. The last three layers were removed, and the rest of ResNet served as feature extractor. Step 3: Use the feature extractor to generate features from training images and test images. Step 4: The training features were combined with training labels and sent into SNN, RVFL and ELM for training. Step 5: The loss function value of SNN, RVFL and ELM was calculated by Eq. (28). Step 6: The weights and biases of SNN, RVFL and ELM were optimized by CBA. The solution vector in CBA contains the weights and biases of SNN, RVFL and ELM by Eq. (26) (27). Step 7: Use the updated weights and biases by CBA to reconstruct the SNN, RVFL and ELM. Step 8: Return to Step 5 until the stopping criteria is met which can be the max iteration time or the value of loss function. Step 9: Use the trained SNN, RVFL and ELM and test feature to generate the predicted labels. Step 10: Generate the performance of our method by the predicted labels and the actual test labels.
Output: The ResNet-SNN-CBA, ResNet-RVFL-CBA and ResNet-ELM-CBA structures and their classification statistics

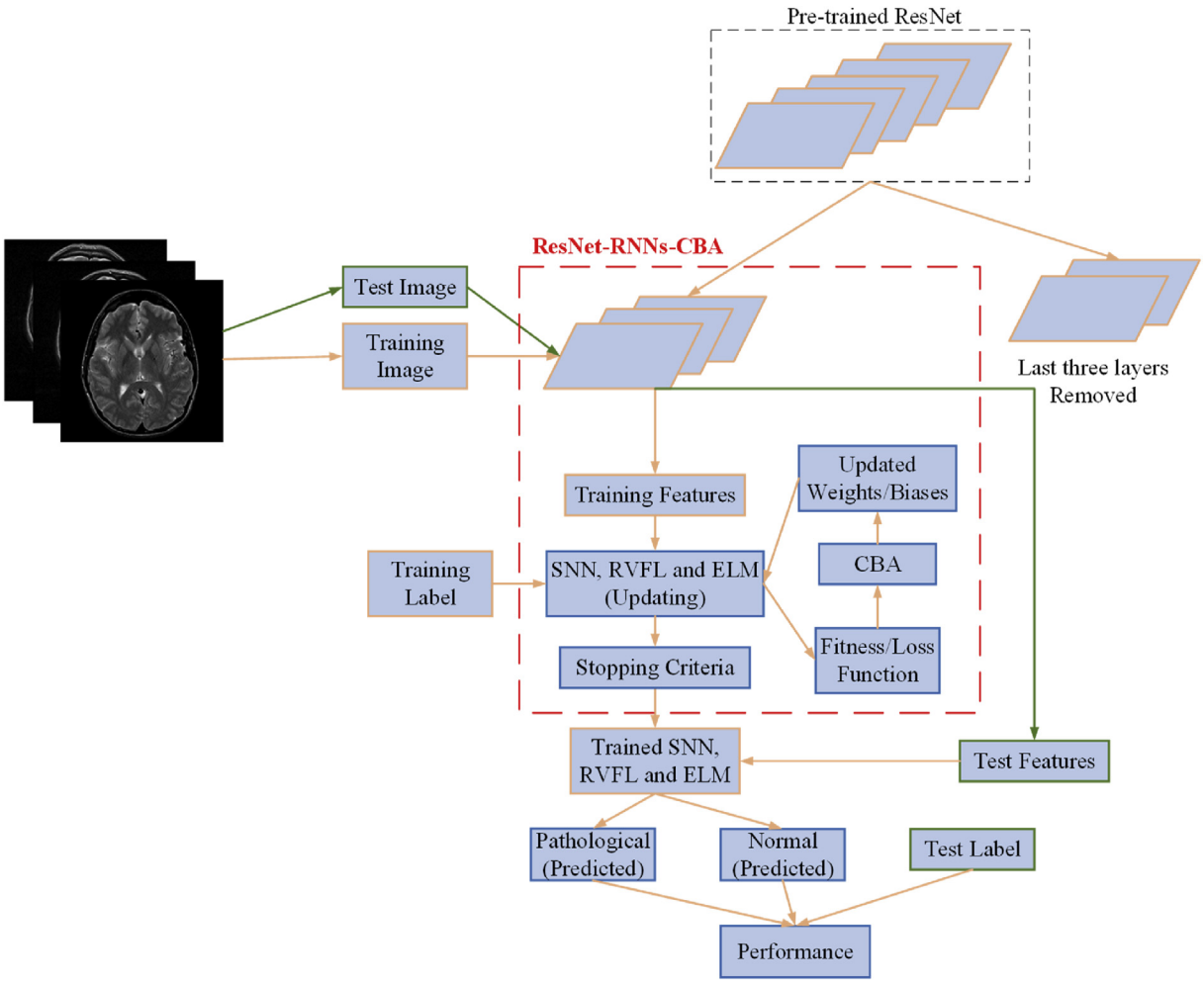

Figure 12. Diagram of the proposed methods.

images, and vectorized as the features. Then, the obtained feature vector was fed into SNN, RVFL, and ELM for training and classification. CBA was leveraged to optimize the parameters in the three randomized neural networks. The solution vector s in CBA contains the weights and biases:

$$s = \text{concatenate}(w', b) \tag{26}$$

$$w' = \text{vectorization}\ (w) \tag{27}$$

where vectorization (w) denotes to reshape the weight matrix w to vector. The fitness function of CBA is chosen as the loss function of the randomized neural networks, which can be expressed as:

$$\text{mse} = \sum_{N}^{i=1} (o_i - t_i)^2 \tag{28}$$

where o_i and t_i denote the network output and sample label, respectively.

Finally, the developed system was evaluated on the test set. The steps of ResNet- RNNs-CBA were summarized in Table 4 and the diagram was presented in Figure 12.

4. Experiment settings

The proposed ResNet-ELM-CBA was implemented on MATLAB 2018a, based on a laptop with CPU i5 8250U and 16 GB RAM. The detailed dataset setting is presented in Table 5.

The pre-defined parameters are listed in Table 6. The number of hidden nodes in SNN, RVFL and ELM is set as 500, because the feature vector dimension is 1000. The parameters for CBA are set according to

Table 5. Training and testing setting.

Total samples			
359			
Pathological		Normal	
177		182	
Training		Testing	
331		28	
Pathological	Normal	Pathological	Normal
163	168	14	14

Table 6. Hyperparameter value.

Hyperparameter	Value
# of hidden nodes in RNNs	500
# of population of bats	20
w for chaotic	0.3
i_max	20
A_0	1.6
R_0	1e-3
α	0.9
γ	0.99
$[f_{min}, f_{max}]$	[0,2]

Table 7. Performance of different classifiers (5 runs).

Methods	Sensitivity	Specificity	Accuracy
ResNet-SNN-CBA	94.29% ± 3.19%	92.86% ± 0.00%	93.57% ± 1.60%
ResNet-RVFL-CBA	95.71% ± 3.91%	90.00% ± 3.92%	92.86% ± 2.52%
ResNet-ELM-CBA	95.71% ± 3.91%	94.29% ± 5.98%	95.00% ± 3.19%

Table 8. Performance comparison.

Methods	Sensitivity	Specificity	Accuracy
RBFNN [52]	95.89%	92.78%	95.44%
CNN [35]	88.41%	96.12%	94.58%
DCNN [31]	~	~	93.18%
ResNet-ELM-CBA(ours)	95.71% ± 3.91%	94.29% ± 5.98%	95.00% ± 3.19%
ResNet-SNN-CBA(ours)	94.29% ± 3.19%	92.86% ± 0.00%	93.57% ± 1.60%
ResNet-RVFL-CBA(ours)	95.71% ± 3.91%	90.00% ± 3.92%	92.86% ± 2.52%

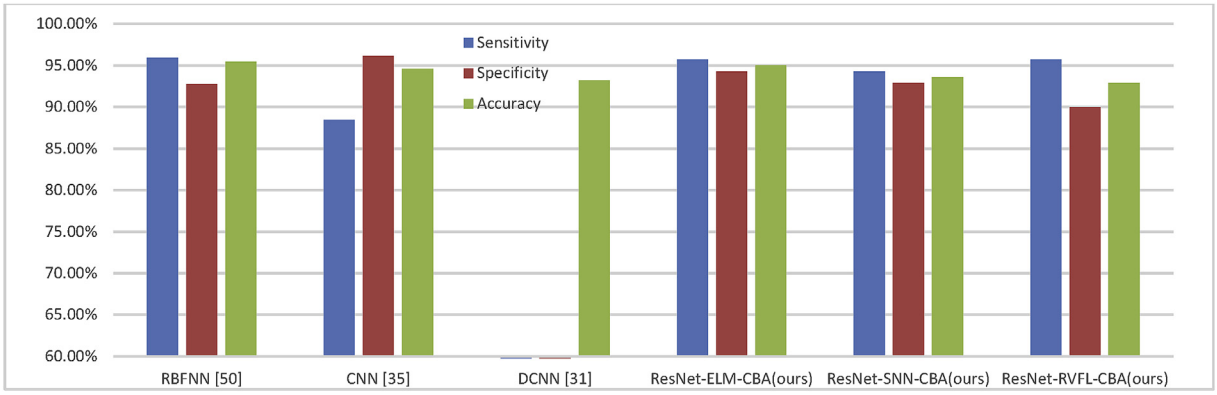

Figure 13. Performance comparison.

previous experience. The max iteration times *i_max* is set as 20 to prevent overfitting.

5. Results and discussion

5.1. Performance of proposed methods

We evaluated the performance of our three ResNet-RNNs-CBA methods based on 5×hold-out validation, i.e., we run the system for 5 times and calculated the average performance. The result is presented below in Table 7. The three methods all achieved over 90% sensitivity, but the specificity varied. The deviation of ResNet-SNN-CBA was the smallest, which means its performance was more stable than the other two algorithms. Generally, the performance of the three proposed methods was at the same level. We shall re-test our approaches when a bigger dataset is available. We consider sensitivity is more important than specificity because if a patient with brain disease is misclassified as normal, he/she may miss the valuable opportunity to be cured.

5.2. Comparison with state-of-the-art

Our three methods were compared with state-of-the-arts including: RBFNN [52], CNN [35], and DCNN [31]. The result is listed in Table 8 and Figure 13. Our performance was obtained by 5 run hold-out validation. Our ResNet-ELM-CBA was not the best in three measurements, but its accuracy, sensitivity and specificity are all around 95.00%. However, the specificity of RBFNN was 92.78% and the sensitivity of CNN was 88.41%, which were worse than ResNet-ELM-CBA. The ResNet we used is not trained on the brain images but employed for feature extraction. We only trained an ELM by CBA optimization, which belongs to a classical neural network structure: SLFN, but its performance was better than the deep neural network CNN [35] in terms of sensitivity. Hence, our method offers a new solution for pathological brain detection.

There are also some methods achieved better classification results than ours in terms of accuracy, like WE-ELM-BA [53], AlexNet-TL [54], and IJaya-ELM [27], which are listed in Table 9. However, there are some problems in these methods. AlexNet-TL achieved perfect 100.00%

Table 9. Methods with good classification performance.

Methods	Sensitivity	Specificity	Accuracy	Defects
WE-ELM-BA [53]	99.04%	93.89%	98.33%	Imbalanced dataset, handcrafted features
AlexNet-TL [54]	100.00%	100.00%	100.00%	Training AlexNet is time-consuming, AlexNet requires more memory than ResNet
IJaya-ELM [27]	~	~	99.69%	Imbalanced dataset, handcrafted features

accuracy, but training such a deep CNN on a small brain image dataset can bring about problems because there are massive parameters in deep CNNs. Additionally, AlexNet requires more memory than ResNet. WE-ELM-BA and IJaya-ELM also outperformed our method, but their dataset was not balanced. Moreover, the features in the two methods were handcrafted, extracted by wavelet transform and ripplet transform, so these features are less transferable.

6. Conclusion

In this study, a novel pathological brain detection method was proposed based on ResNet, randomized neural networks and chaotic theory. The development system ResNet-ELM-CBA achieved the best average accuracy of 95.00% in distinguishing pathological brain from normal control, which is comparable to several state-of-the-art methods. It can help doctors and radiologists in daily diagnosis.

However, the interpretation of the ResNet and RNNs was difficult, so the reasons for the diagnosis results of the system are still unknown. Additionally, it only differentiates the pathological brain from healthy ones, it remains a problem what exact diseases are in the pathological brains. Hence, multi-classification is needed.

In the future, we shall collect more brain MRIs to enlarge our datasets. We will also try to use transfer learning to fine tune deep networks on brain MRI datasets. Visualization of the neural network is another research direction which can help human to understand how the network works in a direct way. We shall apply our method to detect other diseases like Alzheimer's disease, hearing loss, etc.

Declarations

Author contribution statement

S. Lu: Conceived and designed the experiments; Performed the experiments; Analyzed and interpreted the data; Wrote the paper.

S. Wang: Analyzed and interpreted the data; Contributed reagents, materials, analysis tools or data.

Y. Zhang: Analyzed and interpreted the data; Contributed reagents, materials, analysis tools or data; Wrote the paper.

Funding statement

This paper is supported by Hope Foundation for Cancer Research, UK (RM60G0680); International Exchanges Cost Share 2018, UK (RP202G0230); Medical Research Council Confidence in Concept Scheme, UK; Henan Key Research and Development Project, CN (182102310629).

Declaration of interests statement

The authors declare no conflict of interest.

Additional information

No additional information is available for this paper.

References

[1] P. Wang, X. Hu, Y. Li, Q. Liu, X. Zhu, Automatic cell nuclei segmentation and classification of breast cancer histopathology images, Signal Process. 122 (2016) 1–13.

[2] S. Kaur, J.S. Sahambi, Curvelet initialized level set cell segmentation for touching cells in low contrast images, Comput. Med. Imag. Graph. 49 (Apr 2016) 46–57.

[3] Z. Wang, G. Yu, Y. Kang, Y. Zhao, Q. Qu, Breast tumor detection in digital mammography based on extreme learning machine, Neurocomputing 128 (2014) 175–184.

[4] W. Xie, Y. Li, Y. Ma, Breast mass classification in digital mammography based on extreme learning machine, Neurocomputing 173 (2016) 930–941.

[5] C. Huang, et al., A new framework for the integrative analytics of intravascular ultrasound and optical coherence tomography images, IEEE Access 6 (2018) 36408–36419.

[6] C. Huang, et al., A new pulse coupled neural network (PCNN) for brain medical image fusion empowered by shuffled frog leaping algorithm, Front. Neurosci. 13 (2019) 210.

[7] C.X. Huang, et al., Automatic side branch detection in optical coherence tomography images using adjacent frame correlation information, J. Med. Imag. Health Inf. 8 (7) (Sep 2018) 1513–1518 (in English).

[8] C.X. Huang, et al., A new dynamic path planning approach for unmanned aerial vehicles, Complexity (2018) 17 (in English), Art. no. 8420294.

[9] S. Chaplot, L.M. Patnaik, N.R. Jagannathan, Classification of magnetic resonance brain images using wavelets as input to support vector machine and neural network, Biomed. Signal Process Contr. 1 (1) (2006) 86–92.

[10] E.-S.A. El-Dahshan, T. Hosny, A.-B.M. Salem, Hybrid intelligent techniques for MRI brain images classification, Digit. Signal Process. 20 (2) (2010) 433–441.

[11] Y. Zhang, L. Wu, S. Wang, Magnetic resonance brain image classification by an improved artificial bee colony algorithm, Progr. Electromag. Res. 116 (2011) 65–79.

[12] Y. Zhang, L. Wu, An mr brain images classifier via principal component analysis and kernel support vector machine, Progr. Electromag. Res. 130 (2012) 269–388.

[13] H. Kalbkhani, M.G. Shayesteh, B. Zali-Vargahan, Robust algorithm for brain magnetic resonance image (MRI) classification based on GARCH variances series, Biomed. Signal Process Contr. 8 (6) (2013) 909–919.

[14] A. Padma, R. Sukanesh, Segmentation and classification of brain CT images using combined wavelet statistical texture features, Arabian J. Sci. Eng. 39 (2) (2013) 767–776.

[15] M. Saritha, K. Paul Joseph, A.T. Mathew, Classification of MRI brain images using combined wavelet entropy based spider web plots and probabilistic neural network, Pattern Recogn. Lett. 34 (16) (2013) 2151–2156.

[16] Y. Zhang, S. Wang, G. Ji, Z. Dong, An MR brain images classifier system via particle swarm optimization and kernel support vector machine, Sci. World J. 2013 (2013) 130134.

[17] G. Yang, et al., Automated classification of brain images using wavelet-energy and biogeography-based optimization, Multimedia Tools Appl. (2015).

[18] R. Ahmmed, A.S. Swakshar, M.F. Hossain, M.A. Rafiq, Classification of tumors and it stages in brain MRI using support vector machine and artificial neural network, in: Presented at the 2017 International Conference on Electrical, Computer and Communication Engineering (ECCE), 2017.

[19] S.G. Armato, et al., Fine-tuning convolutional deep features for MRI based brain tumor classification, in: Presented at the Medical Imaging 2017: Computer-Aided Diagnosis, 2017.

[20] G. Gilanie, U.I. Bajwa, M.M. Waraich, Z. Habib, H. Ullah, M. Nasir, Classification of normal and abnormal brain MRI slices using Gabor texture and support vector machines, Signal Image Video Proc. 12 (3) (2017) 479–487.

[21] R. Gurusamy, V. Subramaniam, A machine learning approach for MRI brain tumor classification, Comput. Mater. Continua (CMC) 53 (2) (2017).

[22] J. Islam, Y. Zhang, A novel deep learning based multi-class classification method for Alzheimer's disease detection using brain MRI data, in: Presented at the International Conference on Brain Informatics, 2017.

[23] S. Korolev, A. Safiulliny, M. Belyaev, Y. Dodonova, Residual and plain convolutional neural networks for 3D brain MRI classification, in: Presented at the 2017 IEEE 14th International Symposium on Biomedical Imaging, ISBI 2017, 2017.

[24] S. Kumar, C. Dabas, S. Godara, Classification of brain MRI tumor images: a hybrid approach, Proc. Comp. Sci. 122 (2017) 510–517.

[25] S. Lahmiri, Glioma detection based on multi-fractal features of segmented brain MRI by particle swarm optimization techniques, Biomed. Signal Process Contr. 31 (2017) 148–155.

[26] A.K. Lebedeva, et al., MRI-based classification models in prediction of mild cognitive impairment and dementia in late-life depression, Front. Aging Neurosci. 9 (13) (2017).

[27] D.R. Nayak, R. Dash, B. Majhi, Development of pathological brain detection system using Jaya optimized improved extreme learning machine and orthogonal ripplet-II transform, Multimed. Tool. Appl. 77 (17) (2017) 22705–22733.

[28] K. Usman, K. Rajpoot, Brain tumor classification from multi-modality MRI using wavelets and machine learning, Pattern Anal. Appl. 20 (3) (2017) 871–881.

[29] T. Ateeq, et al., Ensemble-classifiers-assisted detection of cerebral microbleeds in brain MRI, Comput. Electr. Eng. 69 (2018) 768–781.

[30] A.M. Hasan, A hybrid approach of using particle swarm optimization and volumetric active contour without edge for segmenting brain tumors in MRI scan, Ind. J. Elect. Eng. Inf. (IJEEI) 6 (3) (2018).

[31] J. Islam, Y. Zhang, Brain MRI analysis for Alzheimer's disease diagnosis using an ensemble system of deep convolutional neural networks, Brain Inf. 5 (2) (May 31 2018) 14.

[32] H. Li, N.A. Parikh, L. He, A novel transfer learning approach to enhance deep neural network classification of brain functional connectomes, Front. Neurosci. 12 (2018) 491.

[33] J. Rieke, F. Eitel, M. Weygandt, J.-D. Haynes, K. Ritter, Visualizing convolutional networks for MRI-based diagnosis of Alzheimer's disease presented at the, in: D. Stoyanov, et al. (Eds.), Understanding and Interpreting Machine Learning in Medical Image Computing Applications, 2018.

[34] M. Zurita, et al., Characterization of relapsing-remitting multiple sclerosis patients using support vector machine classifications of functional and diffusion MRI data, Neuroimage Clin. 20 (2018) 724–730.
[35] M. Sajjad, S. Khan, K. Muhammad, W. Wu, A. Ullah, S.W. Baik, Multi-grade brain tumor classification using deep CNN with extensive data augmentation, J. Comp. Sci. 30 (2019) 174–182.
[36] M. Talo, U.B. Baloglu, Ö. Yıldırım, U. Rajendra Acharya, Application of deep transfer learning for automated brain abnormality classification using MR images, Cognit. Syst. Res. 54 (2019) 176–188.
[37] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Presented at the the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
[38] H. Guang-Bin, Z. Qin-Yu, S. Chee-Kheong, Extreme learning machine: theory and applications, Neurocomputing 70 (1-3) (2006) 489–501.
[39] G.-B. Huang, X. Ding, H. Zhou, Optimization method based extreme learning machine for classification, Neurocomputing 74 (1-3) (2010) 155–163.
[40] Y. Wang, F. Cao, Y. Yuan, A study on effectiveness of extreme learning machine, Neurocomputing 74 (16) (2011) 2483–2490.
[41] G. Huang, G.B. Huang, S. Song, K. You, Trends in extreme learning machines: a review, Neural Network. 61 (Jan 2015) 32–48.
[42] W.F. Schmidt, M.A. Kraaijveld, R.P.W. Duin, Feedforward neural networks with random weights, in: Presented at the proceedings, 11th IAPR international conference on pattern recognition. Vol.II. Conference B: Pattern Recognition Methodology and Systems, 1992.
[43] P.N. Suganthan, Letter: on non-iterative learning algorithms with closed-form solution, Appl. Soft Comput. 70 (2018) 1078–1082.
[44] Y.H. Pao, G.H. Park, D.J. Sobajic, Learning and generalization characteristics of random vector functional-link net, Neurocomputing 6 (1994) 163–180.
[45] L. Zhang, P.N. Suganthan, A comprehensive evaluation of random vector functional link networks, Inf. Sci. 367–368 (2015) 1094–1105.
[46] Y. Ren, P.N. Suganthan, N. Srikanth, G. Amaratunga, Random vector functional link network for short-term electricity load demand forecasting, Inf. Sci. 367–368 (2016) 1078–1093.
[47] Q.-Y. Zhu, A.K. Qin, P.N. Suganthan, G.-B. Huang, Evolutionary extreme learning machine, Pattern Recogn. 38 (10) (2005) 1759–1763.
[48] G.-B. Huang, L. Chen, Convex incremental extreme learning machine, Neurocomputing 70 (16-18) (2007) 3056–3062.
[49] Y. Zhang, J. Wu, Z. Cai, P. Zhang, L. Chen, Memetic extreme learning machine, Pattern Recogn. 58 (2016) 135–148.
[50] X.-S. Yang, A new metaheuristic bat-inspired algorithm, in: Nature Inspired Cooperative Strategies for Optimization, 284, NICSO, 2010, pp. 65–74.
[51] A.M. Arasomwan, A.O. Adewumi, An investigation into the performance of particle swarm optimization with various chaotic maps, Math. Probl Eng. 2014 (2014) 1–17.
[52] Z. Lu, S. Lu, G. Liu, Y. Zhang, J. Yang, P. Phillips, A pathological brain detection system based on radial basis function neural network, J. Med. Imag. Health Inf. 6 (5) (2016) 1218–1222.
[53] S. Lu, et al., A pathological brain detection system based on extreme learning machine optimized by bat algorithm, CNS Neurol. Disord. - Drug Targets 16 (2017).
[54] S. Lu, Z. Lu, Y.-D. Zhang, Pathological brain detection based on AlexNet and transfer learning, J. Comp. Sci. 30 (2019) 41–47.