frontiers in Neuroscience

ORIGINAL RESEARCH
published: 08 November 2018
doi: 10.3389/fnins.2018.00818

Multiple Sclerosis Identification by 14-Layer Convolutional Neural Network With Batch Normalization, Dropout, and Stochastic Pooling

Shui-Hua Wang[1,2†], *Chaosheng Tang*[1†], *Junding Sun*[1†], *Jingyuan Yang*[3†], *Chenxi Huang*[4*], *Preetha Phillips*[5*] and *Yu-Dong Zhang*[1,6*†]

[1] School of Computer Science and Technology, Henan Polytechnic University, Jiaozuo, China, [2] School of Architecture Building and Civil Engineering, Loughborough University, Loughborough, United Kingdom, [3] The Faculty of Computer Science and Engineering, Xi'an University of Technology, Xi'an, China, [4] Department of Computer Science and Technology, Tongji University, Shanghai, China, [5] West Virginia School of Osteopathic Medicine, Lewisburg, WV, United States, [6] Department of Informatics, University of Leicester, Leicester, United Kingdom

OPEN ACCESS

Edited by:
Nianyin Zeng,
Xiamen University, China

Reviewed by:
Xia-an Bi,
Hunan Normal University, China
Victor Chang,
Xi'an Jiaotong-Liverpool University, China

***Correspondence:**
Chenxi Huang
1710051@tongji.edu.cn
Preetha Phillips
pphillips@osteo.wvsom.edu
Yu-Dong Zhang
yudongzhang@ieee.org

[†]These authors have contributed equally to this work

Specialty section:
This article was submitted to Brain Imaging Methods, a section of the journal Frontiers in Neuroscience

Received: 12 September 2018
Accepted: 19 October 2018
Published: 08 November 2018

Citation:
Wang S-H, Tang C, Sun J, Yang J, Huang C, Phillips P and Zhang Y-D (2018) Multiple Sclerosis Identification by 14-Layer Convolutional Neural Network With Batch Normalization, Dropout, and Stochastic Pooling. Front. Neurosci. 12:818. doi: 10.3389/fnins.2018.00818

Aim: Multiple sclerosis is a severe brain and/or spinal cord disease. It may lead to a wide range of symptoms. Hence, the early diagnosis and treatment is quite important.

Method: This study proposed a 14-layer convolutional neural network, combined with three advanced techniques: batch normalization, dropout, and stochastic pooling. The output of the stochastic pooling was obtained via sampling from a multinomial distribution formed from the activations of each pooling region. In addition, we used data augmentation method to enhance the training set. In total 10 runs were implemented with the hold-out randomly set for each run.

Results: The results showed that our 14-layer CNN secured a sensitivity of 98.77 ± 0.35%, a specificity of 98.76 ± 0.58%, and an accuracy of 98.77 ± 0.39%.

Conclusion: Our results were compared with CNN using maximum pooling and average pooling. The comparison shows stochastic pooling gives better performance than other two pooling methods. Furthermore, we compared our proposed method with six state-of-the-art approaches, including five traditional artificial intelligence methods and one deep learning method. The comparison shows our method is superior to all other six state-of-the-art approaches.

Keywords: multiple sclerosis, deep learning, convolutional neural network, batch normalization, dropout, stochastic pooling

INTRODUCTION

Multiple sclerosis (abbreviated as MS) is a condition that affects the brain and/or spinal cord (Chavoshi Tarzjani et al., 2018). It will lead to a wide range of probable symptoms, likely with balance (Shiri et al., 2018), vision, movement, sensation (Demura et al., 2016), etc. It has two main types: (i) relapsing remitting MS and (ii) primary progressive MS. More than eight out of every ten diagnosed MS patients are of the "relapsing remitting" type (Guillamó et al., 2018).

MS diagnosis may be confused with other white matter diseases, such as neuromyelitis optica (NMO) (Lana-Peixoto et al., 2018), acute cerebral infarction (ACI) (Deguchi et al., 2018), acute disseminated encephalomyelitis (ADEM) (Desse et al., 2018), etc. Hence, accurate diagnosis of MS is important for patients and following treatments. In this study, a preliminary study that identifies MS from healthy controls with the help of magnetic resonance imaging (MRI) was investigated and implemented.

Recently, researchers tend to use computer vision and image processing (Zhang and Wu, 2008, 2009; Zhang et al., 2009a,b, 2010a,b) techniques to accomplish MS automatic-identification tasks. For instances, Murray et al. (2010) proposed to use multiscale amplitude modulation and frequency modulation (AM-FM) to identify MS. Nayak et al. (2016) presented a novel method, combining AdaBoost with random forest (ARF). Wang et al. (2016) combined biorthogonal wavelet transform (BWT) and logistic regression (LR). Wu and Lopez (2017) used four-level Haar wavelet transform (HWT). Zhang et al. (2017) proposed a novel MS identification system based on Minkowski-Bouligand Dimension (MBD).

Above methods secured promising results. Nevertheless, their methods need to extract features beforehand, and they need to validate their hand-extracted features effective (Chang, 2018a,b,c; Lee et al., 2018). Recently, convolutional neural network (CNN) attracts the research interest of scholars, since it can mechanically develop the features by its early layers. CNN has already been applied to many fields, such as biometric identification (Das et al., 2019), manipulation detection (Bayar and Stamm, 2018), etc. Zhang et al. (2018) is the first to apply CNN to identify MS, and their method achieved an overall accuracy of 98.23%.

This study is based on the CNN structure of Zhang et al. (2018). We proposed two other improvements: batch normalization and stochastic pooling. In addition, we used dynamic learning rate to accelerate the convergence. Learning rate is a parameter to control how quickly the proposed model converge to a local minimal. Low learning rate means a slow speed toward the downward slope. However, it can certain that we won't miss the local minimum but a long time to converge. Therefore, in our research, we set the learning rate a large value and reduce it by every given number of epochs instead of the fixed small learning rate until achieve convergence.

The rest of this paper is organized as follows: section Data Preprocessing described the data processing including data sources and data preprocessing. Section Methodology illustrates the method used in our research. Section Experiments, Results, and Discussions provided the experiment result and discussion.

DATA PREPROCESSING

Two Sources

The dataset in this study were obtained from Zhang et al. (2018). First, MS images were obtained from the eHealth laboratory (2018). All brain lesions were identified and delineated by experienced MS neurologists, and were confirmed by radiologists. Second, the healthy controls were used from 681 slices of 26 healthy controls provided in Zhang et al. (2018). **Table 1** shows the demographic characteristics of two datasets.

Figure 1A shows the original slice, and **Figure 1B** shows the delineated results with four plaques, Areas surrounded by red line denotes the plaque. **Figures 1C,D** presents two slices from healthy controls.

Contrast Normalization

The brain slices are from two different sources; hence, the scanner machines may have different hardware setting (scanning sequence) and software settings (reconstruction from k-space, the store format, etc.). It is necessary to match the two sources of images in terms of gray-level intensities. This is also called contrast normalization, with aim of achieving consistency in dynamic range of various sources of data.

Histogram stretching (HS) method (Li et al., 2018) was chosen due to ease of implementation. HS aims to enhance the contrast by stretching the range of intensity values of two sources of

TABLE 1 | Demographic characteristics of two datasets.

Dataset	Source	# Subjects	Number of Slice	Age	Gender (m/f)
Multiple sclerosis (2018)	eHealth	38	676	34.1 ± 10.5	17/21
Healthy control (Zhang et al., 2018)	private	26	681	33.5 ± 8.3	12/14

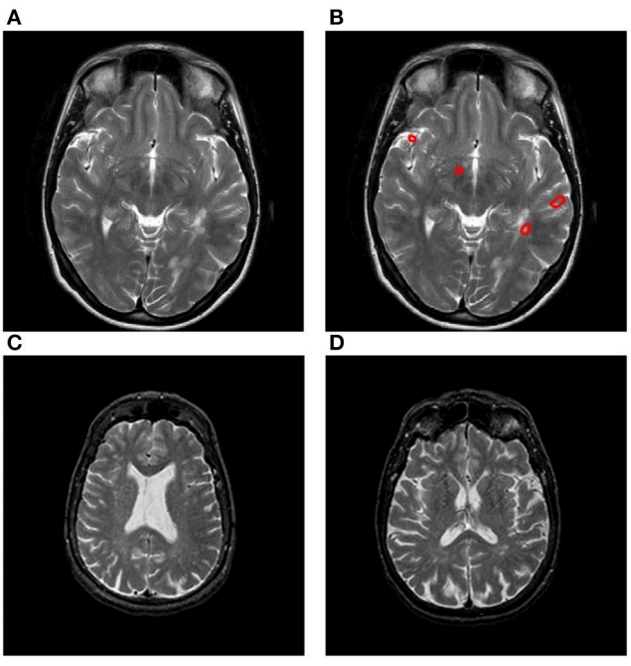

FIGURE 1 | Samples of our dataset. **(A)** Original MS image. **(B)** MS image with plaque delineated. **(C)** Healthy control image I. **(D)** Healthy control image II.

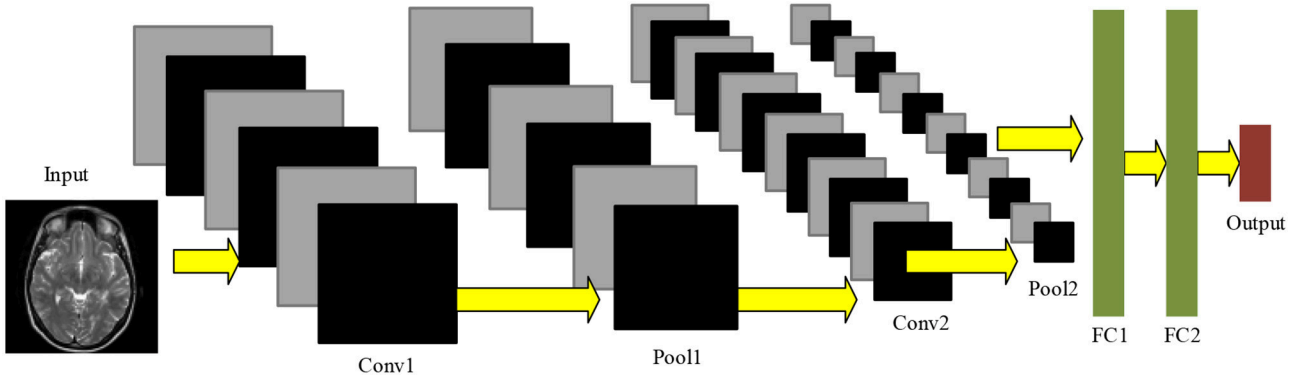

FIGURE 2 | Pipeline of convolutional neural network.

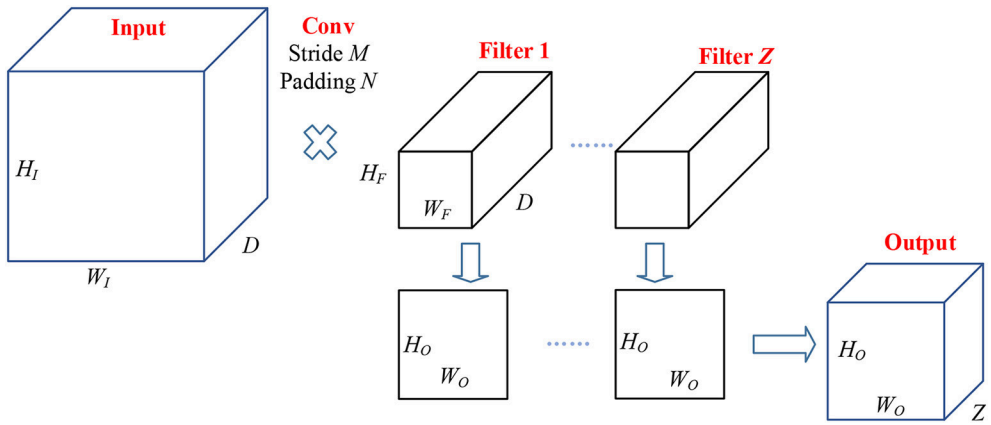

FIGURE 3 | Pipeline of conv layer.

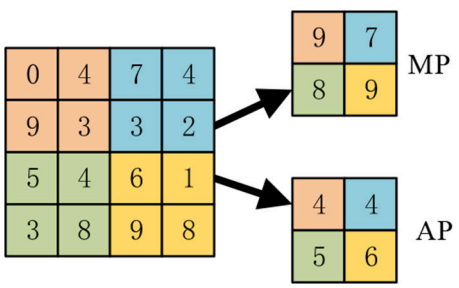

FIGURE 4 | A toy example of max pooling and average pooling.

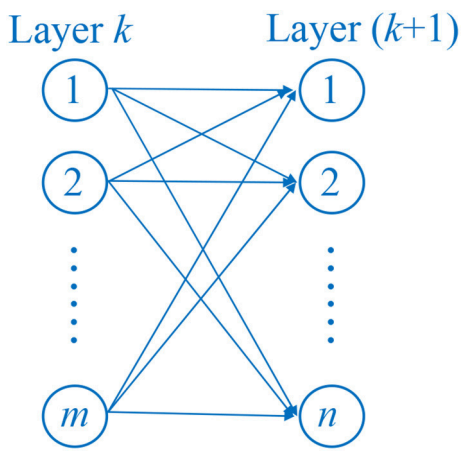

FIGURE 5 | Structure of FC layer.

images to the same range, providing the effect of inter-scan normalization.

The contrast normalization is implemented in following way. Let us assume μ is the original brain image, and φ is the contrast-normalized image, the process of HS can be described as

$$\varphi(x,y) = \frac{\mu(x,y) - \mu_{\min}}{\mu_{\max} - \mu_{\min}} \tag{1}$$

where (x, y) represents the coordinate of pixel, $\mu_{\min}$ and $\mu_{\max}$ represents the minimum and maximum intensity values of original brain image μ.

$$\mu_{\min} = \min_x \min_y \left(\mu(x,y)\right) \tag{2}$$

$$\mu_{\max} = \max_x \max_y \left(\mu(x,y)\right) \tag{3}$$

We do contrast normalization for both two data of different sources, and finally combine them together, forming a 676+681 = 1,357-image dataset.

METHODOLOGY

Convolutional neural network is usually composed of conv layers, pooling layer, and fully connected layers. **Figure 2** gives a toy example that consists of two conv layers, two pooling layers, and two fully connected layers. CNN can achieve comparable or even better performance than traditional AI approaches, while it does not need to manual design the features (Zeng et al., 2014, 2016a,b, 2017a,b).

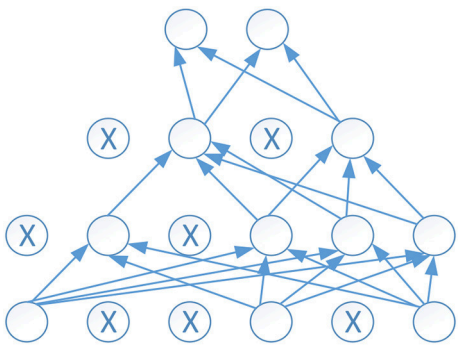

FIGURE 6 | An example of dropout neural network.

TABLE 2 | Variables used in batch normalization.

Parameter	Meaning
z	The output of a layer
z_{norm}	The normalization of z
$\sim$li	Input of the non-linearity layer
α	Mean value of the minibatch
δ^2	Variance of the minibatch
l	Layer index
i	i^{th} data in the mini batch
ε	A small constant
m	The number of samples of the minibatch

Conv Layer

The conv layers performed Two-dimensional convolution along the width and height directions (Yu et al., 2018). It is worth noting that the weights in CNN are learned from backpropagation, except for initialization that weights are given randomly. **Figure 3** shows the pipeline of data passing through a conv layer. Suppose there is an input with size of

$$\text{Input}: H_I \times W_I \times D \tag{4}$$

where H_I, W_I, and C represent the height, width, and channels of the input, respectively.

Suppose the size of filter is

$$\begin{aligned} &\text{Filter } 1: H_F \times W_F \times D \\ &\ldots \\ &\text{Filter } Z: H_F \times W_F \times D \end{aligned} \tag{5}$$

where H_F and W_F are height and width of each filter, and the channels of filter should be the same as that of the input. Z denotes the number of filters. Those filters move with stride of M and padding of N, then the channels of output activation map should be Z. The output size is:

$$\text{Output}: H_O \times W_O \times Z \tag{6}$$

where H_O and W_O are the height and width of the output. Their values are:

$$H_O = 1 + \left\lfloor \frac{2N + H_I - H_F}{M} \right\rfloor \tag{7}$$

$$W_O = 1 + \left\lfloor \frac{2N + W_I - W_F}{M} \right\rfloor \tag{8}$$

where $\lfloor\rfloor$ denotes the floor function. The outputs of conv layer are usually passed through a non-linear activation function, which normally chooses as rectified linear unit (ReLU) function.

TABLE 3 | Hold-out validation setting.

	Training	Test
MS	350	326
HC	350	331
Total	700	657

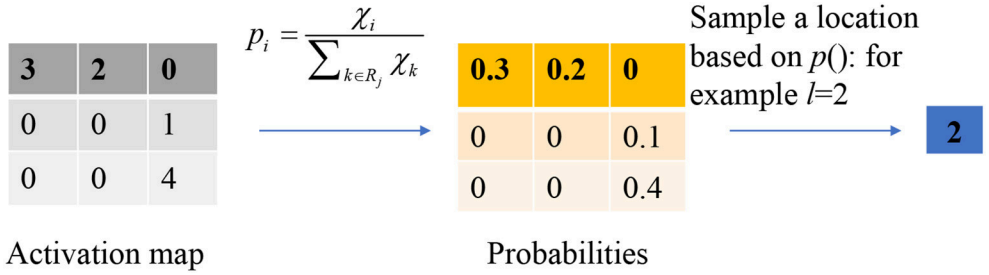

FIGURE 7 | A toy example of stochastic pooling.

Pooling Layer

The activation map contains too much features which can lead to overfitting and computational burden. Pooling layer is often used to implement dimension reduction. Furthermore, pooling can help to obtain invariance to translation. There are two commonly-used pooling methods: average pooling (AP), max pooling (MP).

The average pooling (Ibrahim et al., 2018) is to calculate the average value of the elements in each pooling region, while the max pooling is to select the max value of the pooling region. Suppose the region R contains pixels χ , the average pooling and max pooling are defined as:

$$\text{AP}:\{y_j = \chi_i / \sum_{i \in R_j} \chi_i \} \tag{9}$$

$$\text{MP}:\{y_j = \max_{i \in R_j} \chi_i\} \tag{10}$$

Figure 4 shows the difference, where the kernel size equals 2 and stride equals 2. The max pooling finally outputs the maximum values of all four quadrants, while the average pooling outputs the average values.

Softmax and Fully-Connected Layer

In fully connected (FC) layer, each neuron connects to all neurons of the previous layer, which makes this layer produce many parameters in this layer. The fully connected layer multiplied the input by a weight matrix and added to a bias vector. Suppose layer k contains m neurons, layer $(k+1)$ contains n neurons. The weight matrix will be of size of $m \times n$, and the bias vector will be size of $1 \times n$. **Figure 5** shows the structure of FC layer.

Meanwhile, fully connected layer is often followed by a softmax function used to convert the input to a probability distribution. Here the "softmax" in this study only denotes the softmax function. While some literature will add a fully-connected layer before the softmax function and call the both layers as "softmax function."

Dropout

Deep neural network provides strong learning ability even for very complex function which is hard to understand by human. However, one problem often happened during the training of the deep neural network is overfitting, which means the error based on the training set is very small, but the error is large when the test data is provided to the neural network. We name it as bad generation to new dataset.

Dropout was proposed to overcome the problem of overfitting. Dropout works as randomly set some neurons to zero in each forward pass. Each unit has a fixed probability p independent of the other units to be dropped out. The probability p is commonly set as 0.5. **Figure 6** shows an example of dropout neural network, where the empty circle denotes a normal neuron, and a circle with X inside denotes a dropout neuron. It is obvious using dropout can reduce the links and make the neural network easy to train.

Batch Normalization

As the change of each layer's input distribution caused by the updating of the parameter in the previous layer, which is called as internal covariate shift, can result the slow training. Thus, to solve this problem, we employ the batch normalization to normalizes the layer's inputs over a mini batch to make the input layer have a uniform distribution. All the variables are listed in **Table 2**, then the batch normalization can be implemented as follows:

$$\alpha^l = \frac{1}{m} \sum_i z^{li} \tag{11}$$

$$\sigma^{l2} = \frac{1}{m} \sum_i \left(z^{li} - \alpha^l\right)^2 \tag{12}$$

$$z^{li}_{norm} = \frac{z^{li} - \alpha^l}{\sqrt{\delta^{l2} + \varepsilon}} \tag{13}$$

$$\tilde{z}^{li} = \lambda^l z^{li}_{norm} + \beta^l \tag{14}$$

Here, ε is employed to improve numerical stability while the mini-batch variance is very small. Usually is set as default value e^{-5}. However, the offset β and scale factor γ are updated during training as learnable parameters.

Stochastic Pooling

The stochastic pooling is proposed to overcome the problems caused by the max pooling and average pooling. The average pooling has a drawback, that all elements in the pooling region are considered, thus it may down-weight strong activation due to many near-zero elements. The max pooling solves this problem,

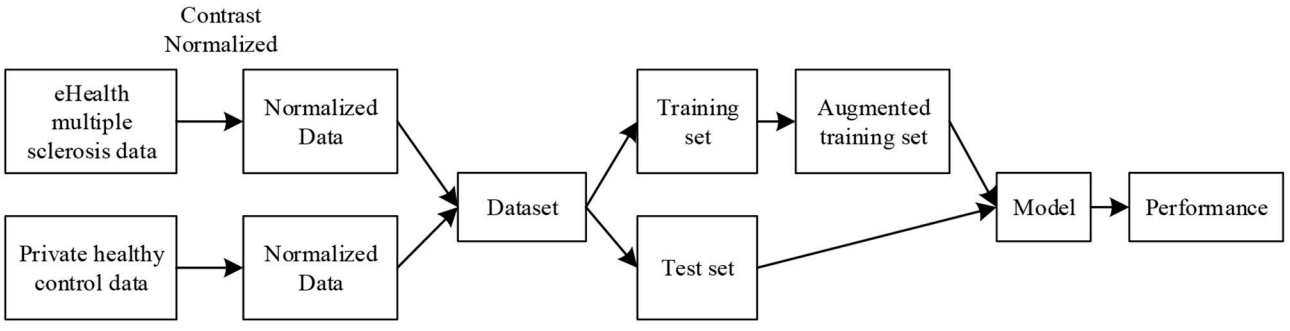

FIGURE 8 | Pipeline of data preprocessing.

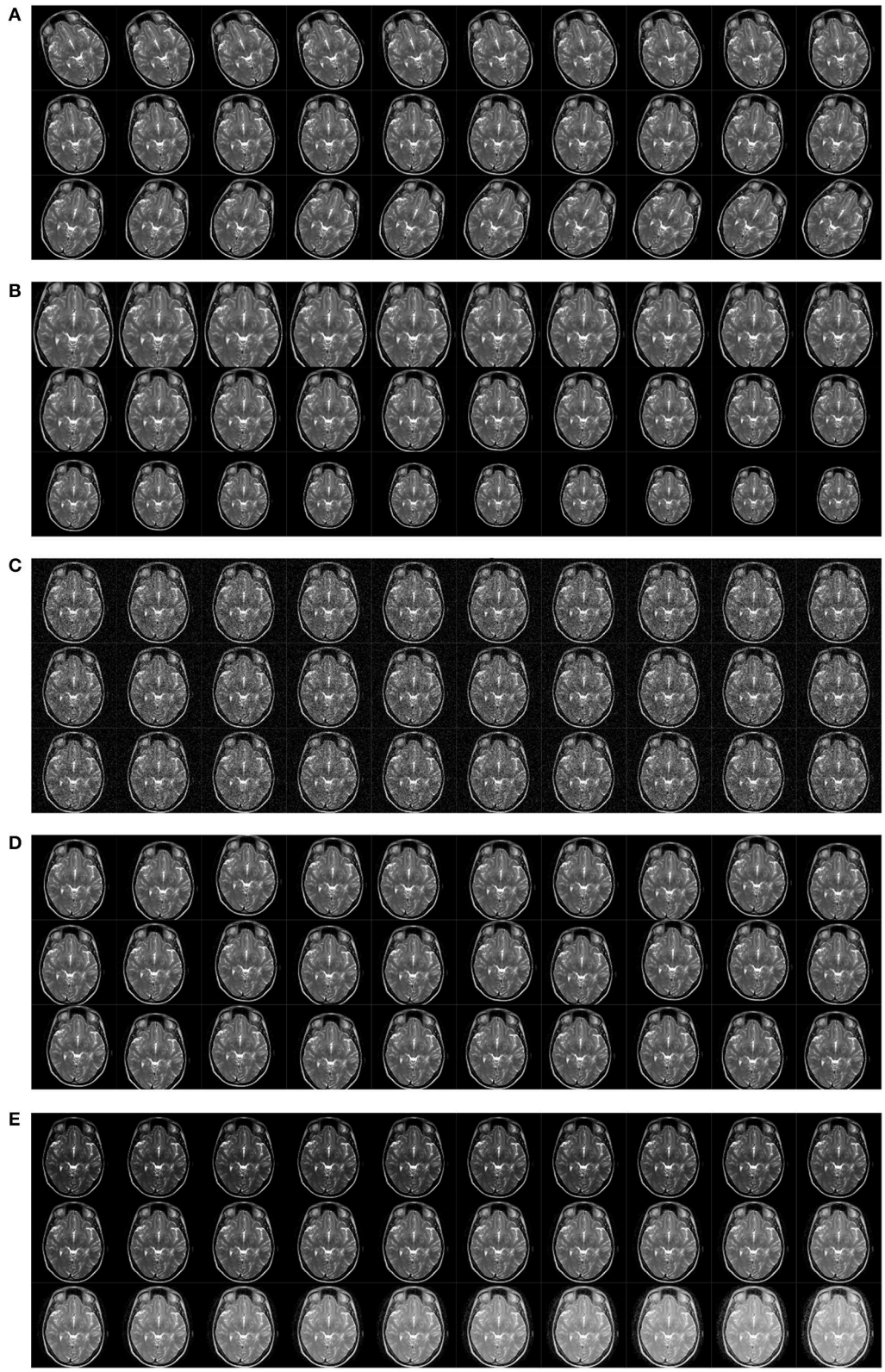

FIGURE 9 | Results of data augmentation. **(A)** Rotation. **(B)** Scaling. **(C)** Noise injection. **(D)** Random translation. **(E)** Gamma correction.

but it easily overfits the training set. Hence, max pooling does not generalize well to test set.

Instead of calculating the mean value or the max value of each pooling region, the output of the stochastic pooling is obtained via sampling from a multinomial distribution formed from the activations of each pooling region R_j. The procedure can be expressed as follows:

(1) Calculate the probability p of each element χ within the pooling region.

$$p_i = \frac{\chi_i}{\sum_{k \in R_j} \chi_k} \tag{15}$$

in which, k is the index of the elements within the pooling region.

TABLE 4 | Hyperparameters of Conv layers.

Layer	Filter size	# Channel	# Filters	Stride
Conv_1	3 × 3	1	8	2
Pool_1	3 × 3			2
Conv_2	3 × 3	8	8	2
Pool_2	3 × 3			2
Conv_3	3 × 3	8	16	1
Conv_4	3 × 3	16	16	1
Conv_5	3 × 3	16	16	1
Pool_3	3 × 3			2
Conv_6	3 × 3	16	32	1
Conv_7	3 × 3	32	32	1
Conv_8	3 × 3	32	32	1
Conv_9	3 × 3	32	64	1
Conv_10	3 × 3	64	64	1
Conv_11	3 × 3	64	64	1
Pool_4	3 × 3			2

TABLE 5 | Hyperparameters of Fully-connected layers.

Layer	Weights	Bias	Probability
FCL_1	20 × 1024	20 × 1	
DO_1			0.5
FCL_2	10 × 20	10 × 1	
DO_2			0.5
FCL_3	2 × 10	2 × 1	

(2) Pick a location l within the pooling region according to the probability p. It is calculated by scanning the pooling region from left to right and up to bottom.

$$A_j = \chi_l, l \sim P(p_1, ..., p_{|R_j|}) \tag{16}$$

Instead of considering the max values only, stochastic pooling may use non-maximal activations within the pooling region. **Figure 7** shows a toy example of using stochastic pooling. We first output the probabilities of the input matrix, then the roulette wheel falls within the pie of 0.2. Hence the location l is finally chosen as 2, and the output is the value at second position.

EXPERIMENTS, RESULTS, AND DISCUSSIONS

Division of the Dataset

Hold-out validation method (Monteiro et al., 2016) was used to divide the dataset. In the training set, there are 350 MS images and 350 HC images. In the test set, we have 326 MS images and 331 HC images. **Table 3** presents the setting hold-out validation method.

The dataset is divided into two parts without validation dataset for our research: training dataset and test dataset as shown in **Table 3**. The missing of validation set is mainly because of following reasons: First, according to the past research, validation

TABLE 6 | Statistical analysis of 10 runs.

Run	Sensitivity	Specificity	Precision	Accuracy
1	98.77	98.19	98.17	98.48
2	98.47	97.58	97.57	98.02
3	98.47	98.79	98.77	98.63
4	98.16	98.79	98.77	98.48
5	99.08	98.79	98.78	98.93
6	98.77	98.79	98.77	98.78
7	99.39	99.40	99.39	99.39
8	99.08	98.49	98.48	98.78
9	98.77	99.40	99.38	99.09
10	98.77	99.40	99.38	99.09
Average	98.77 ± 0.35	98.76 ± 0.58	98.75 ± 0.58	98.77 ± 0.39

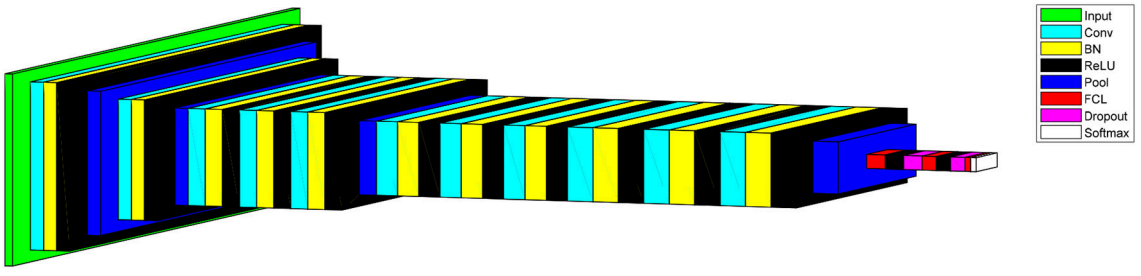

FIGURE 10 | Activation map of proposed CNN model.

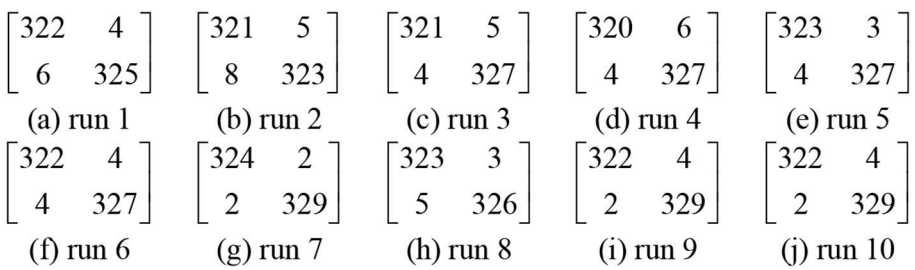

FIGURE 11 | Confusion matrixes of each run.

TABLE 7 | Ten random runs of MP and AP methods.

	Sensitivity (%)	Specificity (%)	Precision (%)	Accuracy (%)
MP				
R1	97.87	97.87	97.89	97.87
R2	98.63	98.63	98.66	98.63
R3	98.18	98.18	98.20	98.17
R4	96.04	96.04	96.11	96.04
R5	96.80	96.80	96.86	96.80
R6	98.78	98.78	98.81	98.78
R7	98.63	98.63	98.65	98.63
R8	97.86	97.86	97.88	97.87
R9	99.24	99.24	99.25	99.24
R10	98.63	98.63	98.64	98.63
Average	98.07 ± 0.93	98.07 ± 0.98	98.10 ± 0.96	98.07 ± 0.98
AP				
1	97.41	97.41	97.55	97.41
2	96.65	96.66	96.67	96.65
3	98.33	98.32	98.37	98.33
4	97.41	97.41	97.42	97.41
5	96.65	96.65	96.65	96.65
6	97.87	97.87	97.88	97.87
7	97.56	97.57	97.58	97.56
8	97.87	97.87	97.92	97.87
9	98.48	98.48	98.52	98.48
10	98.48	98.47	98.51	98.48
Average	97.67 ± 0.64	97.67 ± 0.67	97.71 ± 0.68	97.67 ± 0.67

TABLE 8 | Pooling method comparison and *p*-values of singed-rank test.

Pooling	Sensitivity (%)	Specificity (%)	Precision (%)	Accuracy (%)
MP	98.33 ± 0.75	98.33 ± 0.79	98.34 ± 0.79	98.33 ± 0.80
p-value (SP-MP)	0.0645	**0.0469**	0.0605	**0.0430**
AP	97.67 ± 0.64	97.67 ± 0.67	97.71 ± 0.68	97.67 ± 0.67
p-value (SP-AP)	**0.0020**	**0.0020**	**0.0020**	**0.0020**
SP (Ours)	98.77 ± 0.35	98.76 ± 0.58	98.75 ± 0.58	98.77 ± 0.39

Bold means the p-values are less than 0.05.

TABLE 9 | Comparison of the approach with and without data augmentation.

Approach	Sensitivity (%)	Specificity (%)	Precision (%)	Accuracy (%)
No augmentation	98.22 ± 0.71	98.19 ± 1.03	98.18 ± 1.01	98.20 ± 0.77
Data augmentation	98.77 ± 0.35	98.76 ± 0.58	98.75 ± 0.58	98.77 ± 0.39

set error rate may tend to overestimate the test error rate for the model fit on the entire data set (Bylander, 2002; Whiting et al., 2004). Second, as in order to avoid the overfitting, in addition of the training and test datasets, the validation dataset is necessary to tune the classification parameters. However, in this paper, we employed the drop out to overcome the problem of overfitting. The experiment result showed that there is no overfitting existing. Therefore, validation dataset is not used in our research.

Data Augmentation Results

The deep learning usually needs a large amount of samples. However, ass it is a well-known challenge to collect biomedical data so as to generate more data from the limited data. Meanwhile, data augmentation has been shown to overcome the overfitting and increase the accuracy of classification tasks (Wong et al., 2016; Velasco et al., 2018). Therefore, in this study, we employed five different data augmentation (DA) methods to enlarge the training set (Velasco et al., 2018). First, we used image rotation. The rotation angle θ was set from -30 to 30° in step of 2°. The second DA method was scaling. The scaling factors varied from 0.7 to 1.3 with step of 0.02. The third DA method was noise injection. The zero-mean Gaussian noise with variance of 0.01 was added to the original image to generate 30 new noise-contaminated images due to the random seed. The fourth DA method used was random translation by 30 times for each original image. The value of random translation t falls within the range of [0, 15] pixels, and obeys uniform distribution. The fifth DA method was gamma correction. The gamma-value r varied from 0.4 to 1.6 with step of 0.04.

The original training is presented in **Figures 1A**, **8** shows the pipeline of the data preprocessing, where the augmented training set is used to create a deep convolutional neural network model, and this trained model was tested over the test set, with final performance reported in **Table 6**. **Figure 9A** shows the results of image rotation. **Figure 9B** shows the image scaling results. **Figures 9C–E** shows the results of noise injection, random

TABLE 10 | Comparison to traditional AI approaches.

Approach	Sensitivity (%)	Specificity (%)	Precision(%)	Accuracy (%)
Multiscale AM-FM (Murray et al., 2010)	94.08	93.64	91.91	93.83
ARF (Nayak et al., 2016)	96.23 ± 1.18	96.32 ± 1.48	N/A	96.28 ± 1.25
BWT-LR (Wang et al., 2016)	97.12 ± 0.14	98.25 ± 0.16	N/A	97.76 ± 0.10
4-level HWT (Wu and Lopez, 2017)	N/A	N/A	N/A	87.65 ± 1.79
MBD (Zhang et al., 2017)	97.78 ± 1.29	97.82 ± 1.60	N/A	97.80 ± 1.40
CNN-DO-BN-SP (Ours)	98.77 ± 0.35	98.76 ± 0.58	98.75 ± 0.58	98.77 ± 0.39

TABLE 11 | Comparison to deep learning approaches.

Approach	Sensitivity (%)	Specificity (%)	Precision (%)	Accuracy (%)
CNN-PReLU-DO (Zhang et al., 2018)	98.22	98.24	N/A	98.23
CNN-DO-BN-SP (Ours)	98.77 ± 0.35	98.76 ± 0.58	98.75 ± 0.58	98.77 ± 0.39

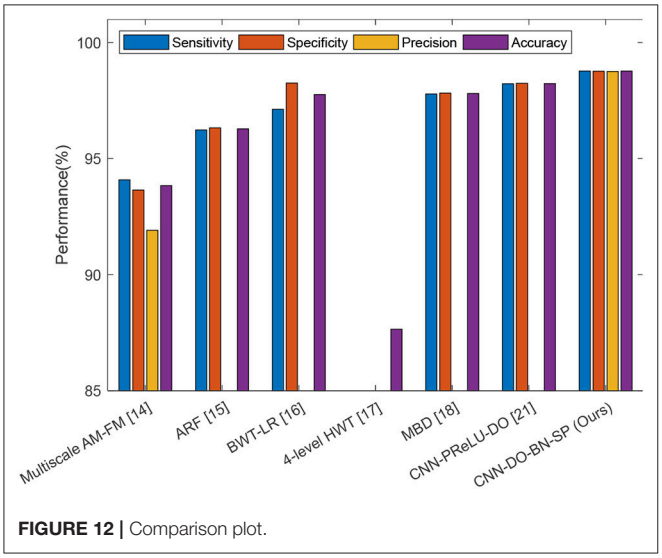

FIGURE 12 | Comparison plot.

translation, and Gamma correction, respectively. As is shown, one training image can generate 150 new images, and thus, the data-augmented training image set is now 151x size of original training set.

Structure of Proposed CNN

We built a 14-layer CNN model, with 11 conv layers and 3 fully-connected layers. Here we did not the number of other layers as convention. The hyperparameters were fine-tuned and their values were listed in **Tables 4**, **5**. The padding values of all layers are set as "same." **Figure 10** shows the activation map of each layer. It is obvious that the height of width of output of each layer shrinks as going to the late layers.

Statistical Results

We used our 14-layer CNN with "DO-BN-SP." We ran the test 10 times, each time the hold-out division was updated randomly. The results over 10 runs are shown in **Table 6**. The average of sensitivity, specificity, and accuracy are 98.77 ± 0.35, 98.76 ± 0.58, and 98.77 ± 0.39, respectively. The confusion matrix of all runs are listed in **Figure 11**.

Pooling Method Comparison

In this experiment, we compared the stochastic pooling (SP) with max pooling (MP) and average pooling (AP). All the other settings are fixed and unchanged. The results of 10 runs of MP and AP are shown in **Table 7**.

We performed Wilcoxon signed rank test (Keyhanmehr et al., 2018) between the results of SP and those of MP, and between the results of SP and those of AP. The results are listed in **Table 8**. It shows SP are significantly better than MP in terms of specificity and accuracy. Meanwhile, SP are significantly better than AP in all four measures.

In this section, Wilcoxon signed rank test was utilized instead of two-sample *t*-test (Jafari and Ansari-Pour, 2018) and chi-square test (Kurt et al., 2019) based on following reasons: two-sample *t*-test supposes the data comes from independent random samples of normal distributions, the same for chi-square goodness-of-fit test. However, our sensitivity/specificity/precision/accuracy data do not meet the condition of gaussian distribution.

Validation of the Data Augmentation

We compared the training process with and without data augmentation to explore the augmentation strategies. The data augmentation methods including: image rotation, scaling, noise injection, random translation and gamma correction as stated in section Data Augmentation Results. The respective performance is shown in **Table 9**. Training with data augmentation could provide better performance, particularly reducing the range of standard deviation.

Comparison to State-Of-The-Art Approaches

In this experiment, we compared our CNN-DO-BN-SP method with traditional AI methods: Multiscale AM-FM (Murray et al., 2010), ARF (Nayak et al., 2016), BWT-LR (Wang et al., 2016), 4-level HWT (Wu and Lopez, 2017), and MBD (Zhang et al., 2017). The results were presented in **Table 10**. Besides, we compared our method with a modern CNN method, viz., CNN-PReLU-DO (Zhang et al., 2018). The results were listed in **Table 11**. We can observe that our method achieved superior performance than all six state-of-the-art approaches, as shown in **Figure 12**.

The reason why our method is the best among all seven algorithms lies in four points. (i) We used data augmentation, to enhance the generality of our deep neural network. (ii) The batch normalization technique was used to resolve the internal

covariate shift problem. (iii) Dropout technique was used to avoid overfitting in the fully connected layers. (iv) Stochastic pooling was employed to resolve the down-weight issue caused by average pooling and overfitting problem caused by max pooling.

The bioinspired-algorithm may help the design or initialization of our model. In the future, we shall try particle swarm optimization (PSO) (Zeng et al., 2016c,d) and other methods. The hardware of our model can be optimized using specific optimization method (Zeng et al., 2018).

In this paper, we employed data augmentation, the main benefits mainly as follows: As it is a well-know challenge to collect biomedical data so as to generate more data from the limited data. Second, data augmentation has been shown to overcome the overfitting and increase the accuracy of classification tasks (Wong et al., 2016; Velasco et al., 2018).

CONCLUSION

In this study, we proposed a novel fourteen-layer convolutional neural network with three advanced techniques: dropout, batch normalization, and stochastic pooling. The main contributes are list as follows:

(1) In this paper, we first applied CNN with stochastic pooling for the Multiple sclerosis detection whose early diagnosis is important for patients' following treatment.
(2) In order to overcome the problems happened in the traditional CNN, such as the internal co shift invariant and overfitting, we utilized batch normalization and dropout.
(3) Considering the size of the dataset, data augmentation was employed in our research for the train set.
(4) The proposed method has the best performance compared to the other state of art methods in terms of sensitivity, specificity, precision and accuracy.

The results showed our method is superior to six state-of-the-art approaches: five traditional artificial intelligence methods and one deep learning method. The detail explanation is provided in section Comparison to State-of-the-art approaches. In the future, we shall try to test other pooling variants, such as pyramid pooling. The dense-connected convolutional networks will also be tested for our task. Meanwhile, we will also work on finding more ways to accelerate convergence (Liao et al., 2018).

AUTHOR CONTRIBUTIONS

S-HW conceived the study. CT and JS designed the model. CT and Y-DZ analyzed the data. S-HW, PP, and Y-DZ acquired the preprocessed the data. JY and JS wrote the draft. CH, PP, and Y-DZ interpreted the results. All authors gave critical revision and consent for this submission.

ACKNOWLEDGMENTS

This paper is supported by Natural Science Foundation of China (61602250), National key research and development plan (2017YFB1103202), Henan Key Research and Development Project (182102310629), Open Fund of Guangxi Key Laboratory of Manufacturing System & Advanced Manufacturing Technology (17-259-05-011K).

REFERENCES

(2018). *MRI Lesion Segmentation in Multiple Sclerosis Database, in eHealth laboratory*. University of Cyprus. Available online at: http://www.medinfo.cs.ucy.ac.cy/index.php/facilities/32-software/218-datasets

Bayar, B., and Stamm, M. C. (2018). Constrained convolutional neural networks: a new approach towards general purpose image manipulation detection. *IEEE Trans. Inform. Forensics Security* 13, 2691–2706. doi: 10.1109/TIFS.2018.2825953

Bylander, T. (2002). Estimating generalization error on two-class datasets using out-of-bag estimates. *Mach. Learn.* 48, 287–297. doi: 10.1023/A:1013964023376

Chang, V. (2018a). An overview, examples, and impacts offered by emerging services and analytics in cloud computing virtual reality. *Neural Comput. Appli.* 29, 1243–1256. doi: 10.1007/s00521-017-3000-1

Chang, V. (2018b). A proposed social network analysis platform for big data analytics. *Technol. Forecast. Soc. Change* 130, 57–68. doi: 10.1016/j.techfore.2017.11.002

Chang, V. (2018c). Data analytics and visualization for inspecting cancers and genes. *Multimed. Tools Appl.* 77, 17693–17707. doi: 10.1007/s11042-017-5186-8

Chavoshi Tarzjani, S. P., Shahzadeh Fazeli, S. A. H., Sanati, M. H., and Nabavi, S. M. (2018). Heat shock protein 70 and the risk of multiple sclerosis in the iranian population. *Cell J.* 20, 599–603. doi: 10.22074/cellj.2019.5620

Das, R., Piciucco, E., Maiorana, E., and Campisi, P. (2019). Convolutional neural network for finger-vein-based biometric identification. *IEEE Trans. Inform. Forensics Security* 14, 360–373. doi: 10.1109/TIFS.2018.2850320

Deguchi, I., Tanahashi, N., and Takao, M. (2018). Clinical Study of Intravenous, low-dose recombinant tissue plasminogen activator for acute cerebral infarction: comparison of treatment within 3 hours versus 3-4.5 hours. *J. Stroke Cerebrovasc. Dis.* 27, 1033–1040. doi: 10.1016/j.jstrokecerebrovasdis.2017.11.009

Demura, Y., Kinoshita, M., Fukuda, O., Nose, S., Nakano, H., Juzu, A., et al. (2016). Imbalance in multiple sclerosis and neuromyelitis optica: association with deep sensation disturbance. *Neurol. Sci.* 37, 1961–1968. doi: 10.1007/s10072-016-2697-4

Desse, N., Sellier, A., Bernard, C., and Dagain, A. (2018). Fatal acute disseminated encephalomyelitis (ADEM) after third ventricle colloid cyst resection with ultrasonic aspirator during neuroendoscopic procedure. *Acta Neurochir.* 160, 1789–1792. doi: 10.1007/s00701-018-3631-8

Guillamó, E., Cobo-Calvo, Á., Oviedo, G. R., Travier, N., Álamo, J., Niño-Mendez, O. A., et al. (2018). Feasibility and effects of structured physical exercise interventions in adults with relapsing-remitting multiple sclerosis: a pilot study. *J. Sports Sci. Med.* 17, 426–436.

Ibrahim, A., Peter, S., and Yuntong, S. (2018). Comparison of a vertically-averaged and a vertically-resolved model for hyporheic flow beneath a pool-riffle bedform. *J. Hydrol.* 557, 688–698. doi: 10.1016/j.jhydrol.2017.12.063

Jafari, M., and Ansari-Pour, N. (2018). Why, When and How to adjust your p values? *Cell J.* 20, 604–607. doi: 10.22074/cellj.2019.5992

Keyhanmehr, A. S., Movahhed, M., Sahranavard, S., Gachkar, L., Hamdieh, M., Afsharpaiman, S., et al. (2018). The effect of aromatherapy with rosa damascena essential oil on sleep quality in children. *Res. J. Pharmacognosy* 5, 41–46.

Kurt, M. N., Yilmaz, Y., and Wang, X. (2019). Real-time detection of hybrid and stealthy cyber-attacks in smart grid. *IEEE Trans. Inform. Forensics Security* 14, 498–513. doi: 10.1109/TIFS.2018.2854745

Lana-Peixoto, M. A., Pedrosa, D., Talim, N., Amaral, J. M., Horta, A., and Kleinpaul, R. (2018). Neuromyelitis optica spectrum disorder

associated with dengue virus infection. *J. Neuroimmunol.* 318, 53–55. doi: 10.1016/j.jneuroim.2018.02.003

Lee, J., Hong, B., Jung, S., and Chang, V. (2018). Clustering learning model of CCTV image pattern for producing road hazard meteorological information. *Fut. Generat. Comp. Syst.* 86, 1338–1350. doi: 10.1016/j.future.2018.03.022

Li, X., Hu, H., Zhao, L., Wang, H., Yu, Y., Wu, L., et al. (2018). Polarimetric image recovery method combining histogram stretching for underwater imaging. *Sci. Rep.* 8:12430. doi: 10.1038/s41598-018-30566-8

Liao, D., Sun, G., Yang, G., and Chang, V. (2018). Energy-efficient virtual content distribution network provisioning in cloud-based data centers. *Fut. Gener. Comp. Syst.* 83, 347–357. doi: 10.1016/j.future.2018.01.057

Monteiro, J. M., Rao, A., Shawe-Taylor, J., and Mourão-Miranda, J. (2016). A multiple hold-out framework for sparse partial least squares. *J. Neurosci. Methods* 271, 182–194. doi: 10.1016/j.jneumeth.2016.06.011

Murray, V., Rodriguez, P., and Pattichis, M. S. (2010). Multiscale AM-FM demodulation and image reconstruction methods with improved accuracy. *IEEE Trans. Image Process.* 19, 1138–1152. doi: 10.1109/TIP.2010.2040446

Nayak, D. R., Dash, R., and Majhi, B. (2016). Brain MR image classification using two-dimensional discrete wavelet transform and AdaBoost with random forests. *Neurocomputing* 177, 188–197. doi: 10.1016/j.neucom.2015.11.034

Shiri, V., Emami, and, M., and Shiri, E. (2018). Investigating the relationship between selective attention and cognitive flexibility with balance in patients with relapsing-remitting multiple sclerosis. *Arch. Rehabil.* 18, 296–305. doi: 10.21859/jrehab.18.4.4

Velasco, J. M., Garnica, O., Lanchares, J., and Botella, M. (2018). Combining data augmentation, EDAs and grammatical evolution for blood glucose forecasting. *Memetic Comp.* 10, 267–277. doi: 10.1007/s12293-018-0265-6

Wang, S., Zhan, T. M., Chen, Y., Zhang, Y., Yang, M., Lu, H. M., et al. (2016). Multiple sclerosis detection based on biorthogonal wavelet transform, RBF kernel principal component analysis, and logistic regression. *IEEE Access* 4, 7567–7576. doi: 10.1109/ACCESS.2016.2620996

Whiting, P., Rutjes, A. W., Reitsma, J. B., Glas, A. S., Bossuyt, P. M., and Kleijnen, J. (2004). Sources of variation and bias in studies of diagnostic accuracy: a systematic review. *Ann. Intern. Med.* 140, 189–202. doi: 10.7326/0003-4819-140-3-200402030-00010

Wong, S. C., Gatt, A., Stamatescu, V., and McDonnell, M. D. (2016). "Understanding data augmentation for classification: when to warp?" in *International Conference on Digital Image Computing: Techniques and Applications (DICTA)* (Gold Coast, QLD), 1–6.

Wu, X., Lopez, M. (2017). Multiple sclerosis slice identification by haar wavelet transform and logistic regression. *Adv. Eng. Res.* 114, 50–55. doi: 10.2991/ammee-17.2017.10

Yu, B., Yang, L., and Chen, F. (2018). Semantic segmentation for high spatial resolution remote sensing images based on convolution neural network and pyramid pooling module. *IEEE J. Selected Topics in Appl. Earth Observ. Remote Sens.* 11, 3252–3261. doi: 10.1109/JSTARS.2018.2860989

Zeng, N., Wang, Z., Zineddin, B., Li, Y., Du, M., Xiao, L., et al. (2014). Image-based quantitative analysis of gold immunochromatographic strip via cellular neural network approach. *IEEE Trans. Med. Imag.* 33, 1129–1136. doi: 10.1109/TMI.2014.2305394

Zeng, N. Y., Wang, Z., and Zhang, H. (2016b). Inferring nonlinear lateral flow immunoassay state-space models via an unscented Kalman filter. *Sci. China-Inform. Sci.* 59:112204. doi: 10.1007/s11432-016-0280-9

Zeng, N. Y., Wang, Z., Zhang, H., and Alsaadi, F. E. (2016c). A novel switching delayed PSO algorithm for estimating unknown parameters of lateral flow immunoassay. *Cognit. Comput.* 8, 143–152. doi: 10.1007/s12559-016-9396-6

Zeng, N. Y., Wang, Z., Zhang, H., Liu, W., and Alsaadi, F. E. (2016a). Deep belief networks for quantitative analysis of a gold immunochromatographic Strip. *Cognit. Comput.* 8, 684–692. doi: 10.1007/s12559-016-9404-x

Zeng, N. Y., Yi, Y., Lusheng, X., Hong, Z., Lishan, Y., Wenxing, H., et al. (2018). A new imaged-based quantitative reader for the gold immunochromatographic assay. *Optik* 152, 92–99. doi: 10.1016/j.ijleo.2017.09.109

Zeng, N. Y., Zhang, H., and Chen, Y. (2016d). Path planning for intelligent robot based on switching local evolutionary PSO algorithm. *Assem. Automat.* 36, 120–126. doi: 10.1108/AA-10-2015-079

Zeng, N. Y., Zhang, H., Li,., Y., and Liang, J., Dobaie, A. M. (2017a). Denoising and deblurring gold immunochromatographic strip images via gradient projection algorithms. *Neurocomputing* 247, 165–172. doi: 10.1016/j.neucom.2017.03.056

Zeng, N. Y., Zhang, H., Liu, W., Liang, J., and Alsaadi, F. E. (2017b). A switching delayed PSO optimized extreme learning machine for short-term load forecasting. *Neurocomputing* 240, 175–182. doi: 10.1016/j.neucom.2017.01.090

Zhang, Y., Jun, Y., Wei, G., and Wu, L. (2010a). Find multi-objective paths in stochastic networks via chaotic immune PSO. *Expert Syst. Appl.* 37, 1911–1919. doi: 10.1016/j.eswa.2009.07.025

Zhang, Y., and Wu, L. (2008). Pattern recognition via PCNN and tsallis entropy. *Sensors* 8, 7518–7529. doi: 10.3390/s8117518

Zhang, Y., and Wu, L. (2009). Segment-based coding of color images. *Sci. China Series F-Inform. Sci.* 52, 914–925. doi: 10.1007/s11432-009-0019-7

Zhang, Y., Wu, L., Neggaz, N., Wang, S., and Wei, G. (2009a). Remote-sensing image classification based on an improved probabilistic neural network. *Sensors* 9, 7516–7539. doi: 10.3390/s90907516

Zhang, Y., Wu, L., Wang, S., and Wei, G. (2010b). Color image enhancement based on HVS and PCNN. *Sci. China Informat. Sci.* 53, 1963–1976. doi: 10.1007/s11432-010-4075-9

Zhang, Y., Wu, L., and Wei, G. (2009b). A new classifier for polarimetric SAR images. *Prog. Electromag. Res.* 94, 83–104. doi: 10.2528/PIER09041905

Zhang, Y-D., Pan, C., Sun, J., and Tang, C. (2018). Multiple sclerosis identification by convolutional neural network with dropout and parametric ReLU. *J. Comput. Sci.* 28, 1–10. doi: 10.1016/j.jocs.2018.07.003

Zhang, Y. D., Zhang, Y., Phillips, P., Dong, Z., and Wang, S. (2017). Synthetic minority oversampling technique and fractal dimension for identifying multiple sclerosis. *Fractals* 25:1740010. doi: 10.1142/S0218348X17400102

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.