

Multiple sclerosis detection via 6-layer stochastic pooling convolutional neural network and multiple-way data augmentation

Jian WANG^{1*}, Dimas LIMA^{2*}

¹ School of Computing and Mathematical Sciences, University of Leicester, Leicester, LE1 7RH, UK

² Department of Electrical Engineering, Federal University of Santa Catarina, 88040-900, Florianópolis, Brazil

*Correspondence: jw830@le.ac.uk, dimaslima@ieee.org

<https://doi.org/10.37175/stemedicine.v2i8.101>

ABSTRACT

Background: Multiple sclerosis is one of most widespread autoimmune neuroinflammatory diseases which mainly damages body function such as movement, sensation, and vision. Despite of conventional clinical presentation, brain magnetic resonance imaging of white matter lesions is often applied to diagnose multiple sclerosis at the early stage.

Methods: In this article, we proposed a 6-layer stochastic pooling convolutional neural network (CNN) with multiple-way data augmentation for multiple sclerosis detection in brain magnetic resonance imaging. Our approach does not demand hand-crafted features unlike those traditional machine learning methods. Via application of stochastic pooling and multiple-way data augmentation, our 6-layer CNN achieved equivalent performance against those deep learning methods which consist of so many layers and parameters that ordinarily bring difficulty to training. Further, we also conducted ablation experiments to examine the contribution of stochastic pooling and multiple-way data augmentation to the original CNN model.

Results: The results showed that this 6-layer CNN obtained a sensitivity of $95.98 \pm 0.46\%$, a specificity of $95.67 \pm 0.92\%$, and an accuracy of $95.82 \pm 0.58\%$. According to comparison experiments, our results are better than state-of-the-art approaches.

Conclusion: Our scheme of stochastic pooling and multiple-way data augmentation enhanced the original 6-layer CNN model compared to those using maximum pooling or average pooling and inadequate data augmentation.

Keywords: Multiple sclerosis · Convolutional neural network · Stochastic pooling · Data augmentation · Magnetic resonance imaging

Introduction

Multiple sclerosis (MS) is an autoimmune disease characterized by demyelinating inflammatory white matter lesions of the central nervous system. MS does harm for patient's health by impeding nerve-signal transmitted

between brain and other parts of body. It often involves periventricular white matter, spinal cord, brainstem, cerebellum, and optic nerve. Multiple sclerosis may cause loss of muscle coordination, impaired vision, and loss of body function to people. And it is still not clear about its etiology and pathogenesis, which needs further study by medical researchers. Multiple sclerosis can be divided into four categories: (i) relapsing-remitting MS, (ii) secondary-progressive MS, (iii) primary-progressive MS, and (iv) progressive-relapsing MS. Relapsing-remitting MS (R-R) is most commonly observed in clinic, accounting for

Received: Aug 27, 2021; Accepted: Sep 16, 2021.

© The Author(s). 2021 This is an **Open Access** article distributed under the terms of the Creative Commons License (<http://creativecommons.org/licenses/by/4.0/>) which permits unrestricted use, distribution, and reproduction in any medium or format, provided the original work is properly cited.

around 85% of the total. R-R patients usually endure several times of relapse while in remission period the condition is stable. Secondary-progressive MS (S-P) is derived from R-R gradually. About 80% of R-R patients lead to S-P within twenty-five years and the condition would not be relieved like what it occurs in R-R period. Unlike the R-R and S-P MS, primary-progressive MS (P-P) skips the beginning stage. Patient's condition is aggravated since firstly affected by multiple sclerosis. This category of MS accounts for almost 10% of the total. As for progress-relapsing (P-R) MS, it is rarely seen in clinic. As we can see, R-R and S-P take major proportion of multiple sclerosis. If patients receive effective and suitable treatment in the early stage of MS, it will decrease the chance of turning from R-R to S-P which means patients could suffer less relapse and pain. Therefore, detection for multiple sclerosis as soon as possible is momentous to doctors fighting against multiple sclerosis.

Though researchers realized the significance of diagnosing multiple sclerosis in the early stage, it is not effortless to identify MS from healthy people accurately. In terms of clinical manifestations, multiple sclerosis is similar with other white matter diseases including disseminated encephalomyelitis (ADEM), acute cerebral infarction (ACI) and neuromyelitis optica (NMO). Under this circumstance, researchers had to look for other techniques to improve the success rate of MS diagnosis. Magnetic resonance imaging (MRI) is often utilized for the diagnosis of MS due to its characteristics of less ionizing radiation damage to human body, clear soft tissue imaging quality, and the ability to obtain original three-dimensional cross-sectional images without reconstruction. In the meanwhile, scientists also realized that computer-aided diagnosis was playing an increasingly important role in the field of medical image analysis. The methods applying computer vision and digital image processing to brain MRI have surpassed humans in diagnosing diseases such as Alzheimer's (1), epilepsy (2), Creutzfeldt-Jakob disease (3), and cerebral glioma (4). Therefore, the application of computer vision and digital image processing in craniocerebral MRI to improve the diagnosis rate of multiple sclerosis has become the focus of researchers. For example, Wang, et al. (5) proposed a method for multiple sclerosis detection based on biorthogonal wavelet transform, RBF kernel principal component analysis, and logistic regression. Nayak, et al. (6) presented an approach using discrete wavelet transform and AdaBoost with random forests. Recently, Zhang, et al. (7) applied dropout and parametric ReLU in building convolutional neural network (CNN) for MS identification. Eitel, et al. (8) proposed their CNN-based method for MS detection with layer-wise relevance propagation. Alijamaat, et al. (9) put forward wavelet CNN for MS detection in brain MRI images. Han, et al. (10) used adaptive genetic algorithm (AGA) for MS recognition. Han, et al. (11) employed particle swarm optimization (PSO) for MS recognition. Tang (12) used a five-layer CNN (5I-CNN) for MS detection.

These previous works could be divided into two

categories. The first category of methods (5, 6) is based on traditional hand-crafted features. They need to coin specific features manually and it is usually boring and time-consuming. The second category of methods (7-9) is based on deep learning. It is common that they adopted deep neural networks which may contain over fifty layers or even two hundred layers to conduct the classification. These huge neural networks, nevertheless, are hard to train and cost too much computational resources (mainly GPUs) which are expensive for some researchers to afford.

In this study, we proposed an approach based on 6-layer convolutional neural network to identify brain MRI images for diagnosis of multiple sclerosis. Compared with the traditional methods based on manual feature extraction, our approach applies CNN. So it has stronger capability of feature extraction and object classification and also avoid the complicated process of manual feature selection. Compared with the methods based on deep neural network, our model structure has only six layers, instead of dozens or even hundreds of layers. Large networks tend to be time-consuming, laborious, and difficult to reach convergence, and also easy to overfit, while our 6-layer neural network does not have these disadvantages. The second strength of the proposed 6-layer CNN is that it adopts stochastic pooling, which brings better generalization performance compared to those deep neural networks using max pooling. And crucially, we conducted up to sixteen methods of data augmentation. To the best of our knowledge, there is no other previous work in a model of diagnosing MS using so many ways of data augmentation. Our approach has the most diverse and comprehensive methods of data augmentation at present. In general, the proposed approach has the characteristics of simple network architecture, fast training speed and easy convergence. At the same time, due to the application of stochastic pooling and multiple-way data augmentation, this approach achieved competitive results in the detection of multiple sclerosis on brain MRI images.

In the next chapter of the paper, we will first introduce the experiment data and the preprocessing operation for the data set. Data and preprocessing are also vital for building a successful neural network of vision task. In the third chapter, we will present the CNN architecture, stochastic pooling, and multiple-way data augmentation step by step. Then we will show the design of our experiments, including validation and evaluation. In the fourth chapter, we will give discussion of experimental results. It is worth noting that our experimental results contain ablation experimental results to demonstrate how much stochastic pooling and multiple-way data augmentation we applied could boost the performance of a simple CNN in MS diagnosis. Finally, in the fifth chapter, we will provide the summary of this study and put forward some possible improvement directions in the future.

Dataset Sources

We acquired the same dataset as (7). This dataset totally

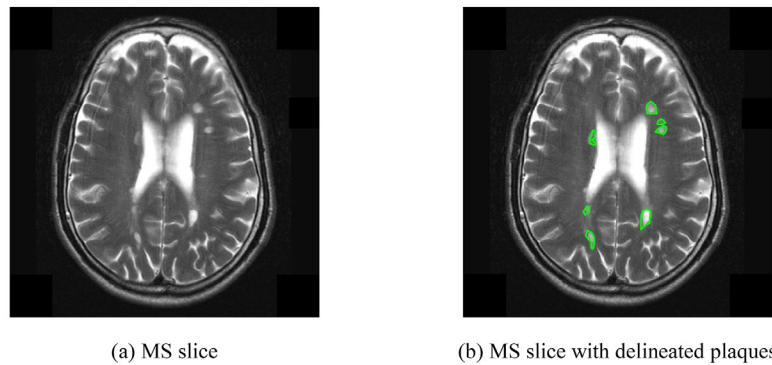


Figure 1. Samples of our dataset.

consists of 1357 MRI images in which 676 slices (13) are multiple sclerosis images and 681 slices (7) are health controls. We randomly selected two samples from the dataset as shown in **Figure 1**. **Figure 1** (a) presented one original MS slice and **Figure 1** (b) described the delineated plaques on **Figure 1** (a). We also provided **Table 1** to illustrate the demographic characteristics of the dataset.

Data preprocessing

As we mentioned, our dataset was combined from two sources of images. This would lead to difference of image characteristics between two sources of images on account of factors such as scanning equipment and reconstruction process. In order to restrain the difference, we need to apply contrast normalization technique to balance the two sources of images into a same range of gray-level intensity. In this study, we adopted histogram stretching (14) as our method of contrast normalization because of its effectiveness and simplicity.

The histogram stretching method could be presented as following:

$$\varphi(x,y) = (g(x,y) - g_1)/(g_2 - g_1) \quad (1)$$

$$g_2 = \max(g(x,y)) \quad (2)$$

$$g_1 = \min(g(x,y)) \quad (3)$$

in which parameters are defined in **Table 2**.

Via this operation, we can observe that the distribution of gray-level intensity in two sources of images are stretched to the same field. As a result, we as far as possible formed the two sources of images into one entire dataset and avoided the negative influence on subsequent process.

Convolutional Neural Network

Pooling Layer

Pooling layer, also named as subsampling or downsampling, is often used behind of convolutional layer in a classic architecture of CNN (15). Its main purposes include reducing feature dimension of convolutional layer output (16), suppressing noise, reducing quantity of parameters and computation cost, and dampening overfitting (17).

Unlike most of other neural network using CNN (18) as backbone, our 6-layer CNN applied stochastic pooling rather than max pooling or average pooling. Suppose there existed a pooling window upon the region of feature map which covered k elements. Each element of feature map was recorded as v_i , and i was the index of element (19). After the pooling window slide upon this region, the output of pooling operation was written as u . Then the max pooling operation could be described as:

$$u = \max(v_i) \quad (4)$$

which means max pooling always selects the biggest element within a region of feature map (20). The average pooling operation could be described as:

$$u = (\sum v_i)/k \quad (5)$$

which means average pooling adopts the mean value of k elements. In stochastic pooling, we first calculated the probability map of the chosen region.

$$p_i = v_i/\sum v_i \quad (6)$$

Then stochastic pooling would choose the value of one element as sampling value according to the probability distribution (21). The p_i was bigger, v_i was more likely chosen as the sampling value, but not definitely. The mechanism could be described as:

$$u = v_i, i \sim P(p_1, \dots, p_i, \dots, p_k) \quad (7)$$

Figure 2 illustrated the comparison between max pooling,

Table 1. Demographic characteristics of the dataset.

Dataset	Source	Slices	Subjects	Male/Female	Age
Health controls [7]	private	681	26	12/14	33.5 ± 8.3
Multiple sclerosis [13]	eHealth	676	38	17/21	34.1 ± 10.5

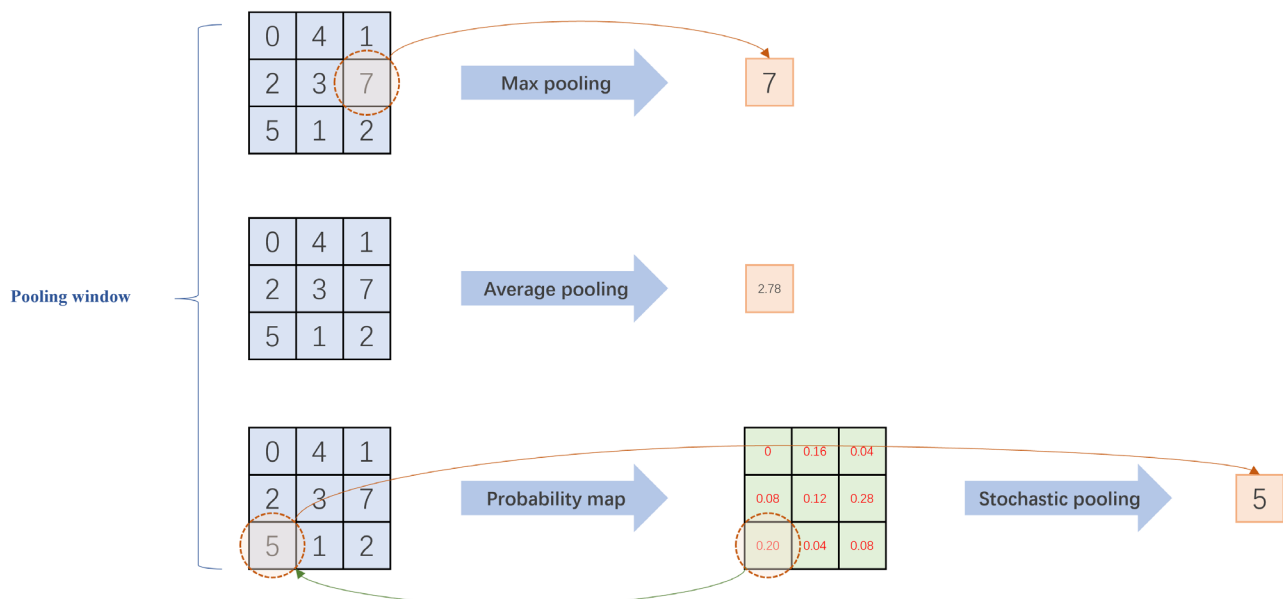


Figure 2. Comparison of max pooling, average pooling, and stochastic pooling.

average pooling, and stochastic pooling.

Structure of our six-layer CNN

Our proposed CNN structure consisted of three convolutional layers and three fully-connected layers (22). Generally speaking, convolutional layers are meant to extract features while fully-connected layers are used for classification. Each convolutional layer was followed by activation function and a stochastic pooling layer (23, 24). Activation function was applied for nonlinear transformation after convolution calculation. During the procedure of activation function and pooling layer, there exists no learnable weights (25). Hence, we usually do not count them in neural network structure. As it is shown in **Table 3**, three convolutional layers and three stochastic pooling layers formed the 3-layer convolutional network in the structure.

In our 6-layer CNN structure, it contained three fully-connected layers and three dropout layers. Ahead of each fully-connected layer, a dropout layer was inserted to make CNN more robust to training. The retention probabilities of three dropout layers are set as 0.5, 0.5, and 0.5, respectively, by trial-and-error method. **Table 4**

presented the structure of fully-connected layers in our proposed model. At last, we offered **Figure 3** to portray the whole structure of the proposed 6-layer stochastic pooling CNNs.

Multiple-way data augmentation

It is known that the learning process of neural network cannot leave the support of massive data samples. On most occasions, the more training data are fed to neural network, the better model we attain (26, 27). However, in reality, data samples is often insufficient. Lack of samples will not only impair the model to obtain the best performance, but also lead to difficult training and frequent overfitting. Data augmentation (DA) technology (28, 29) is aimed at expanding the original small dataset into a larger one by means of digital signal processing, so as to alleviate the problem of insufficient samples. In previous work, data augmentation has been applied but with only five ways (rotation, scaling, Gaussian noise, random translation, and Gamma correction) (30). It is our contribution that we exploited up to 18-way data augmentation methods. As far as we are aware, this study applied most ways of data augmentation among existing

Table 2. Parameters in histogram stretching.

Parameter	Meaning
x	Horizontal coordinate in an image
y	Vertical coordinate in an image
$g(x,y)$	The original gray-level intensity in location (x,y)
g_2	The highest gray-level intensity in an image
g_1	The lowest gray-level intensity in an image
$\varphi(x,y)$	The stretched gray-level intensity in location (x,y)

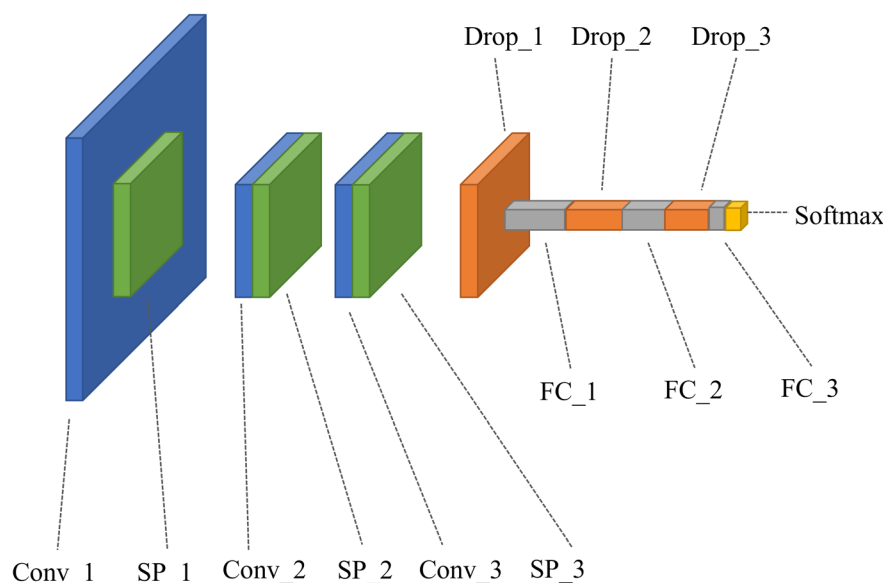


Figure 3. Structure of 6-layer stochastic pooling CNN.

CNN-based MS detection approaches (31). Our data augmentation methods contained three categories that were geometric-based methods, noise-based methods, and photometric-based methods. At first, there were nine ways of data augmentation (32). Then via reflecting these augmented samples horizontally, we got double sets of data augmentation methods which contained twenty ways in total. **Table 5** provided a list of data augmentation methods we used.

Geometric-based methods

In geometry, geometric-based methods are also named

as affine transformation, which indicates transforming one existing vector space to another one (33). Affine transformation is combined with a linear transformation plus a shift (34). Assume an original vector space was recorded as \vec{v} , and the linear transformation could be described as a matrix written as A , meanwhile the shift was written as \vec{b} . Then the new transformed vector space \vec{u} was calculated as below.

$$\vec{u} = A\vec{v} + \vec{b} \quad (8)$$

Back to digital image processing, affine transformation means transform points in the image from their previous

Table 3. Convolutional layers.

Layer	Filter size	Channel	Filters	Stride
Conv_1	7×7	1	1	2
SP_1	3×3			2
Conv_2	3×3	16	16	2
SP_2	3×3			1
Conv_3	3×3	32	32	2
SP_3	3×3			1

Table 4. Fully-connected layers.

Layer	Weights	Bias	Probability
Drop_1			0.5
FC_1	200×16384	200×1	
Drop_2			0.5
FC_2	100×200	100×1	
Drop_3			0.5
FC_3	2×100	2×1	



Figure 4. Data augmentation results.

coordinates to new ones. Assume the coordinates of the raw image were recorded as $\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$, and the transformed

coordinates were written as $\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$. Then the transformation process could be described as:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} A & \vec{b} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (9)$$

in which $\begin{bmatrix} A & \vec{b} \\ 0 & 0 & 1 \end{bmatrix}$ was called affine transformation matrix and was also frequently written as $\begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix}$.

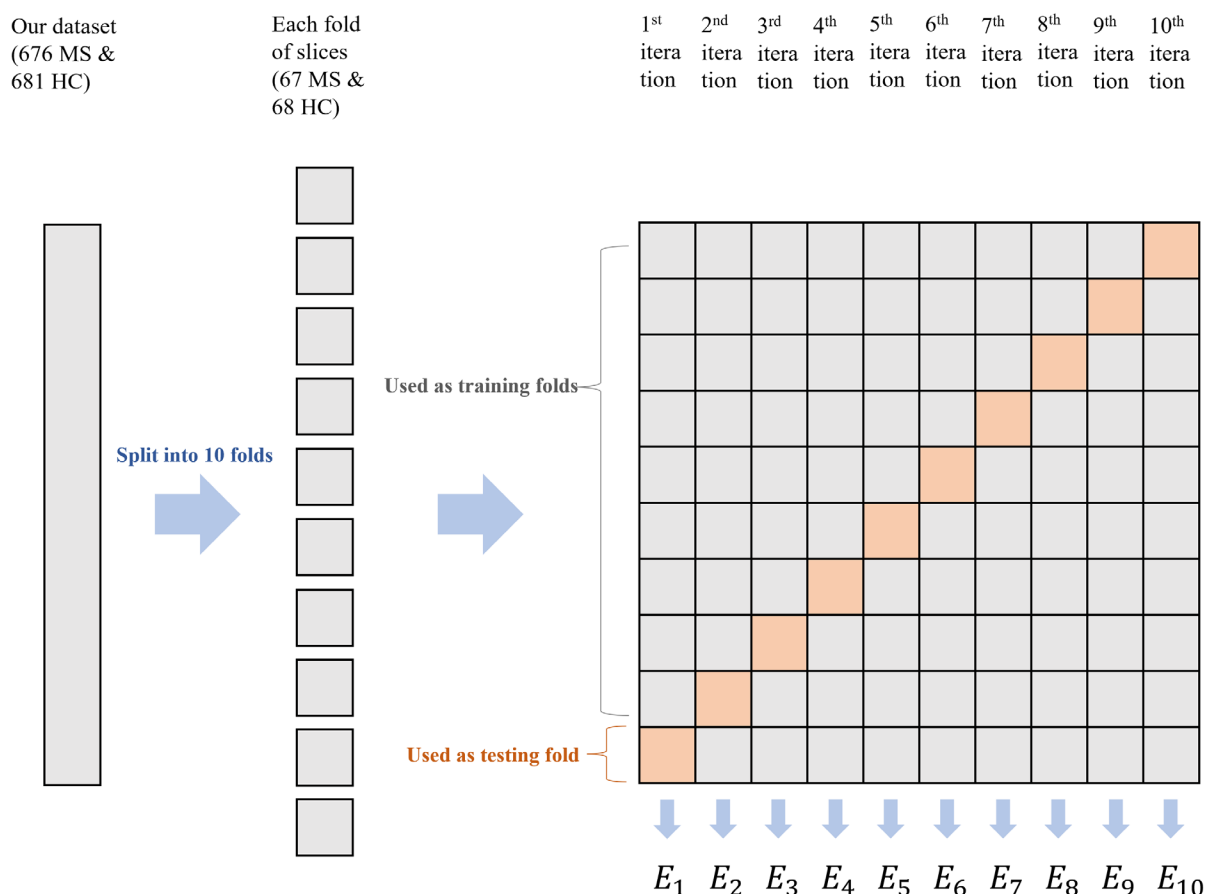


Figure 5. 10-fold cross validation.

Every affine transformation can be represented through particular affine transformation matrix. Here we introduce six ways of affine transformation that employed in this study.

Horizontal flipping. It is a geometric transformation operation performing on the raw image to generate a mirror image which is symmetrical about y-axis (35). Compared to vertical flipping, horizontal flipping is more often adopted and has been tested effectiveness on popular datasets such as ImageNet and CIFAR-10 (36). The affine transformation matrix of horizontal flipping could be written as $\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$. Thus, we get the transformed coordinates as follows.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (10)$$

Horizontal shear. It is defined as changing location of each point in the image horizontally, along the x-axis. And the amount of displacement along the x-axis is determined by each point's coordinate of y-axis (37). The affine transformation matrix of horizontal shear could be written as $\begin{bmatrix} 1 & \tan \alpha & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$. So we attain the transformed

coordinates as below.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & \tan \alpha & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (11)$$

Figure 4 (a) showed examples of horizontal shear.

Vertical shear. In contrast to horizontal shear, vertical shear is defined as changing location of each point in the image vertically, along the y-axis. And the amount of displacement along the y-axis is determined by each point's coordinate of x-axis. The affine transformation matrix of vertical shear could be written as $\begin{bmatrix} 1 & 0 & 0 \\ \tan \alpha & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$. Therefore, we obtain the transformed coordinates as follows.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ \tan \alpha & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (12)$$

Figure 4 (b) showed examples of vertical shear.

Rotation. Like its name, it rotates an image around a fixed point. In most cases, rotation is around the central point of the image in data augmentation. The affine transformation matrix of rotation could be written as

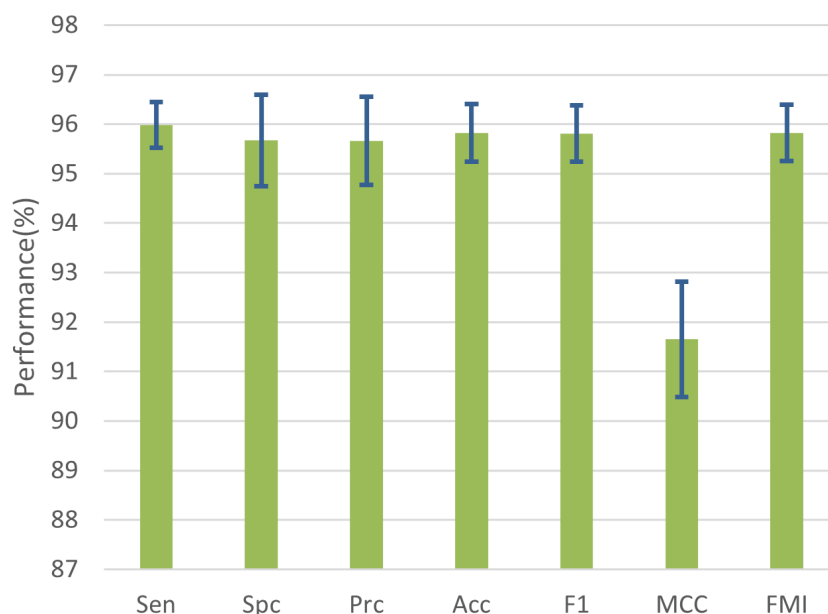


Figure 6. Error bar of stochastic pooling results.

$\begin{bmatrix} 1 & 0 & c \\ 0 & 1 & f \\ 0 & 0 & 1 \end{bmatrix}$. Then the transformation operation could be described as below.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & c \\ 0 & 1 & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (15)$$

Figure 4 (e) showed examples of random translation.

Noise-based methods

In data augmentation, noise-based methods are defined as injecting noise to image samples. Via adding noise into images, training dataset would enhance its sampling variance so as to overcome the lack of data.

Gaussian noise. As one the most commonly-used noises, Gaussian noise is often added to raw images in data augmentation. Mark z as the gray level, z obeys the probability density function as follow:

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(z-\mu)^2/2\sigma^2} \quad (16)$$

where μ represents the mean gray value, and σ means standard deviation of z . Figure 4 (f) showed examples of Gaussian noise.

Salt-and-pepper noise. It is a widely-used noise in data augmentation as well. In salt-and-pepper noise augmentation, z obeys the probability density function which could be depicted as:

$$p(z) = \begin{cases} P_a & z = a \\ P_b & z = b \\ 0 & \text{otherwise} \end{cases} \quad (17)$$

where a and b are threshold values for salt noise and

pepper noise. Figure 4 (g) showed examples of salt-and-pepper noise.

Speckle noise. As a granular interference, speckle noise naturally occurs in radar or ultrasound images. Suppose F was the observed image, f was the image without noise, N_m referred to multiplicative noise, and N_a referred to additive noise. Then speckle noise could be defined as follows.

$$F = f + N_m f + N_a \quad (18)$$

Figure 4(h) showed examples of speckle noise.

Photometric-based method

Gamma correction. In the beginning, Gamma correction was made for luminance adjustment in imaging or display system (38). Because human's perception of luminance is not linear with light power, but with a relation of power function. And the exponent of this power function was recorded as γ . The gamma correction is usually written as follows.

$$v_{out} = A v_{in}^\gamma \quad (19)$$

In this equation, v_{out} represents output gray value and v_{in} represents input gray value. When $\gamma < 1$, we often regard this gamma correction as gamma compression. While $\gamma > 1$, we regard it as gamma expansion (39). In this study, we applied gamma correction on raw images of datasets as one of data augmentation methods to enlarge our training samples. Figure 4 (i) showed examples of Gamma correction.

In the end we provided Figure 4 to illustrate effects of multiple-way data augmentation.

10-fold Cross Validation

In this study, we utilized 10-fold cross validation as our method of dataset division. We divided the whole dataset into ten folds (40). Each fold contains 67 multiple

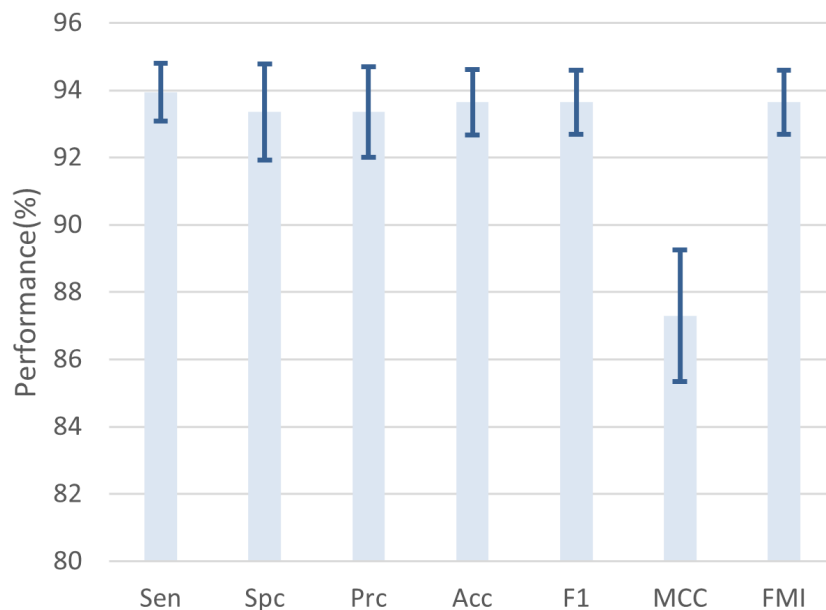


Figure 7. Error bar of average pooling results.

sclerosis slices and 68 health controls. In every iteration, we adopted nine folds of data as training set while the other one fold as testing set and repeated this procedure ten times. **Figure 5** described this process of splitting the dataset into ten folds and repeating training-testing for ten iterations.

In the end, we obtained ten expectations of model performance and calculated the final expectation by averaging the expectations of each iteration (41). This calculation could be written as below:

$$E = \frac{1}{10} \sum_{i=1}^{10} E_i \quad (20)$$

Measure

In this study, we applied confusion matrix (shown in **Table 6**) to measure the performance.

In confusion matrix, we counted true positive (TP), true negative (TN), false positive (FP), and false negative (FN) and used these values to calculate sensitivity (SEN), specificity (SPC), precision (PRC), accuracy (ACC), F_1 score, Matthews correlation coefficient (MCC), and Fowlkes-Mallows index (FMI). The calculation processes were described as below:

$$SEN = \frac{TP}{TP + FN} \quad (21)$$

$$SPC = \frac{TN}{TN + FP} \quad (22)$$

$$PRC = \frac{TP}{TP + FP} \quad (23)$$

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \quad (24)$$

$$F_1 = \frac{2PRC * SEN}{PRC + SEN} \quad (25)$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (26)$$

$$FMI = \frac{TP}{\sqrt{(TP + FP)(TP + FN)}} \quad (27)$$

In addition to original measures of sensitivity, specificity, precision, accuracy, F_1 score, MCC, and FMI, we calculated the standard deviation and the average based on these seven measures for further performance comparison experiments as well.

Results and Discussions

Statistical Analysis

Table 7 showed the results of 10 runs. Our approach based on 6-layer stochastic pooling CNNs and multiple-way data augmentation secured a sensitivity of $95.98 \pm 0.46\%$, a specificity of $95.67 \pm 0.92\%$, a precision of $95.66 \pm 0.89\%$, an accuracy of $95.82 \pm 0.58\%$, a F_1 score of $95.81 \pm 0.57\%$, a MCC of $91.65 \pm 1.16\%$, and a FMI of $95.82 \pm 0.57\%$. Also, we gave **Figure 6** to present the error bar of 10-run results.

Pooling Methods Comparison

In order to inspect into the contribution stochastic pooling made on the model's performance, we conducted comparison experiments which replaced stochastic pooling with average pooling and max pooling respectively in proposed CNNs. As we can see in **Table 8**, our 6-layer CNN achieved a sensitivity of $93.93 \pm 0.86\%$, a specificity of $93.35 \pm 1.42\%$, a precision of $93.35 \pm 1.35\%$, an accuracy of $93.64 \pm 0.98\%$, a F_1 score of 93.64

Table 5. 18-way data augmentation.

DA category	Method	Representation
Geometric-based methods		
	Horizontal flipping	$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
	Horizontal shear	$\begin{bmatrix} 1 & \tan \alpha & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
	Vertical shear	$\begin{bmatrix} 1 & 0 & 0 \\ \tan \alpha & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
	Rotation	$\begin{bmatrix} \cos \beta & \sin \beta & c \\ -\sin \beta & \cos \beta & f \\ 0 & 0 & 1 \end{bmatrix}$
	Scaling	$\begin{bmatrix} r & 0 & c \\ 0 & r & f \\ 0 & 0 & 1 \end{bmatrix}$
	Random translation	$\begin{bmatrix} 1 & 0 & c \\ 0 & 1 & f \\ 0 & 0 & 1 \end{bmatrix}$
Noise-based methods		
	Gaussian noise	$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(z-u)^2/2\sigma^2}$
	Salt-and-pepper noise	$p(z) = \begin{cases} P_a & z = a \\ P_b & z = b \\ 0 & \text{otherwise} \end{cases}$
	Speckle noise	$F = f + N_m f + N_a$
Photometric-based method		
	Gamma correction	$v_{out} = Av_{in}^\gamma$

$\pm 0.96\%$, a MCC of $87.29 \pm 1.96\%$, and a FMI of $93.64 \pm 0.96\%$. And the error bar of results using average pooling was shown in **Figure 7**.

As it was presented in **Table 9**, with max pooling, the proposed CNN obtained a sensitivity of $94.06 \pm 1.54\%$, a specificity of $94.56 \pm 1.44\%$, a precision of $94.54 \pm 1.41\%$, an accuracy of $94.31 \pm 1.27\%$, a F1 score of $94.30 \pm 1.29\%$, a MCC of $88.64 \pm 2.54\%$, and a FMI of $94.30 \pm 1.28\%$. **Figure 8** depicted the error bar of results using max pooling.

Via these comparison experiments, we could observe that using stochastic pooling earned best performance in almost every measure including sensitivity, specificity,

precision, accuracy, F1 score, MCC, and FMI. To make it convenient for presenting the advantage of stochastic pooling, we drew **Figure 9**. It revealed that our model achieved improvement of nearly 2% in each measure when applying stochastic pooling compared with results using average pooling or max pooling.

Comparison with State-of-the-art Algorithms

We compared our 6-layer stochastic pooling CNN with state-of-the-art algorithms for multiple sclerosis detection such as AGA (10), PSO (11), and 5l-CNN (12). These three state-of-the-art algorithms were tested with the same dataset as ours. The comparison results was given in **Table 10**.

Table 6. Confusion matrix of binary classification.

		Predicted	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

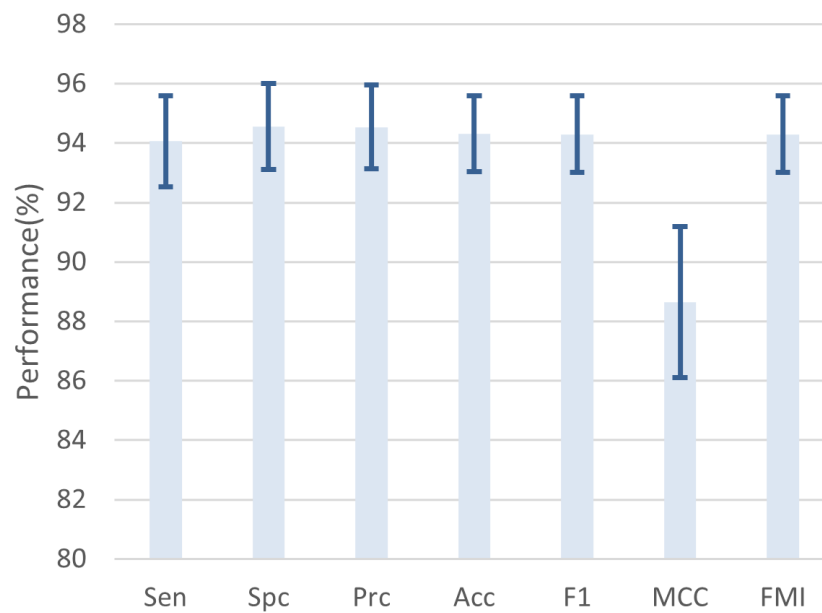


Figure 8. Error bar of max pooling results.

Table 7. Results of ten runs.

Run	Sen	Spc	Prc	Acc	F1	MCC	FMI
1	95.86	95.45	95.43	95.65	95.65	91.31	95.65
2	96.60	96.77	96.74	96.68	96.67	93.37	96.67
3	95.71	96.33	96.28	96.02	95.99	92.04	95.99
4	95.41	95.59	95.56	95.50	95.48	91.01	95.48
5	96.89	95.89	95.90	96.39	96.39	92.78	96.40
6	96.15	96.92	96.87	96.54	96.51	93.08	96.51
7	95.56	96.04	96.99	95.80	95.77	91.60	95.77
8	96.01	94.42	94.47	95.21	95.23	90.43	95.23
9	95.86	95.15	95.15	95.50	95.50	91.01	95.51
10	95.71	94.13	94.18	94.92	94.94	89.84	94.94
MSD	95.98 ± 0.46	95.67 ± 0.92	95.66 ± 0.89	95.82 ± 0.58	95.81 ± 0.57	91.65 ± 1.16	95.82 ± 0.57

Table 8. Experimental results using average pooling.

Run	Sen	Spc	Prc	Acc	F1	MCC	FMI
1	93.05	92.51	92.50	92.78	92.77	85.56	92.77
2	95.27	94.27	94.29	94.77	94.78	89.54	94.78
3	93.05	92.95	92.91	93.00	92.98	86.00	92.98
4	92.90	94.13	94.01	93.52	93.45	87.04	93.45
5	94.08	91.63	91.77	92.85	92.91	85.73	92.92
6	94.38	91.92	92.06	93.15	93.21	86.32	93.21
7	93.34	91.63	91.72	92.48	92.52	84.98	92.53
8	95.12	95.45	95.40	95.28	95.26	90.57	95.26
9	94.38	95.15	95.08	94.77	94.73	89.54	94.73
10	93.79	93.83	93.79	93.81	93.79	87.62	93.79
MSD	93.93 ± 0.86	93.35 ± 1.42	93.35 ± 1.35	93.64 ± 0.98	93.64 ± 0.96	87.29 ± 1.96	93.64 ± 0.96

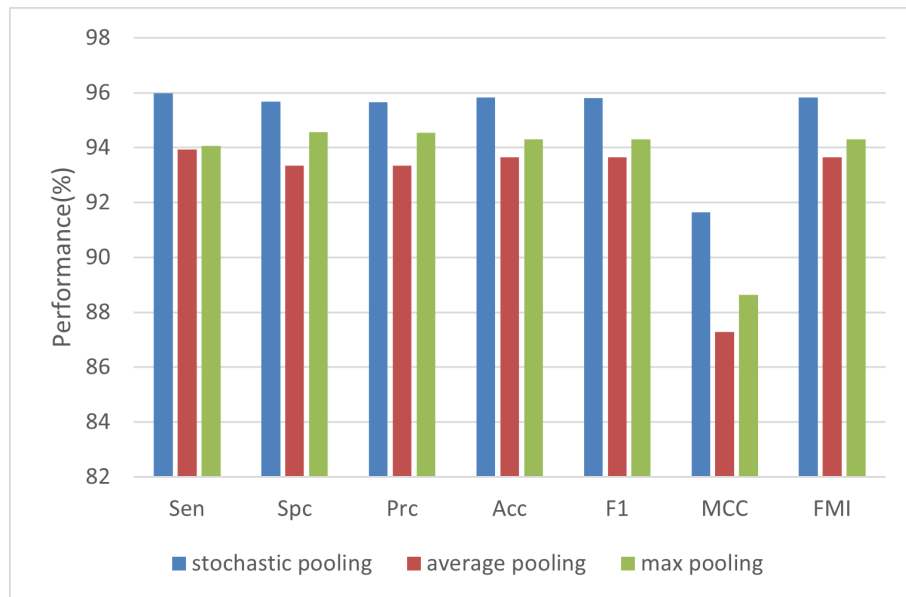


Figure 9. Pooling methods comparison.

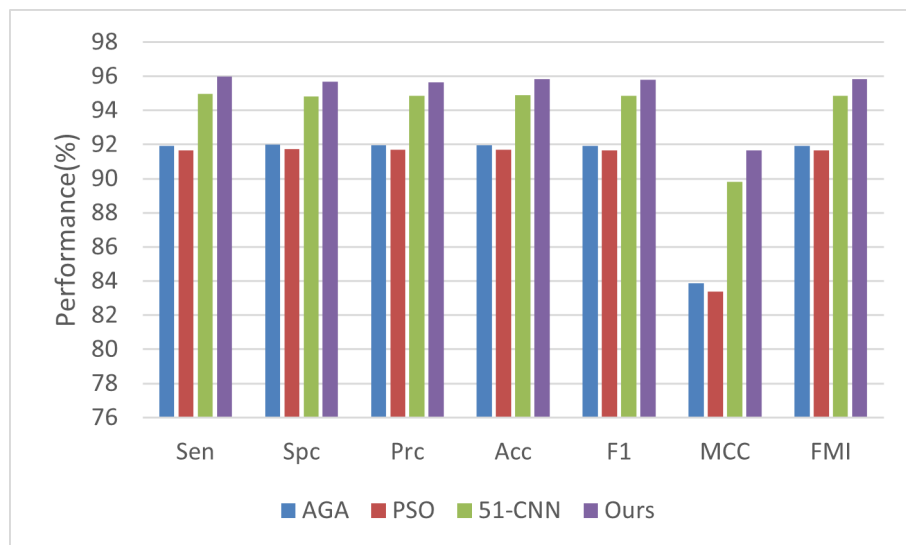


Figure 10. Comparison plot.

Table 9. Experimental results using max pooling.

Run	Sen	Spc	Prc	Acc	F1	MCC	FMI
1	95.00	94.69	94.70	94.84	94.85	89.69	94.85
2	95.00	95.00	95.00	95.00	95.00	90.00	95.00
3	93.12	95.31	95.21	94.22	94.15	88.46	94.16
4	94.69	95.62	95.58	95.16	95.13	90.32	95.13
5	95.31	96.88	96.83	96.09	96.06	92.20	96.07
6	95.31	95.94	95.91	95.62	95.61	91.25	95.61
7	95.31	92.50	92.71	93.91	93.99	87.85	94.00
8	93.12	93.12	93.12	93.12	93.12	86.25	93.12
9	93.12	93.12	93.12	93.12	93.12	86.25	93.12
10	90.62	93.44	93.25	92.03	91.92	84.10	91.93
MSD	94.06 ± 1.54	94.56 ± 1.44	94.54 ± 1.41	94.31 ± 1.27	94.30 ± 1.29	88.64 ± 2.54	94.30 ± 1.28

Table 10. Comparison with SOTA.

Method	Sen	Spc	Prc	Acc	F1	MCC	FMI
AGA [10]	91.91 ± 1.24	91.98 ± 1.36	91.97 ± 1.32	91.95 ± 1.19	91.92 ± 1.20	83.89 ± 2.41	91.92 ± 1.19
PSO [11]	91.67 ± 1.41	91.73 ± 0.77	91.70 ± 0.78	91.70 ± 0.97	91.67 ± 1.00	83.40 ± 1.98	91.67 ± 0.99
5l-CNN [12]	94.96 ± 1.43	94.82 ± 1.67	94.85 ± 1.62	94.89 ± 1.37	94.87 ± 1.37	89.83 ± 2.74	94.87 ± 1.37
Ours	95.98 ± 0.46	95.67 ± 0.92	95.66 ± 0.89	95.82 ± 0.58	95.81 ± 0.57	91.65 ± 1.16	95.82 ± 0.57

We also offered **Figure 10** to show our method's strength against state-of-the-art algorithms. We could observe that our proposed method acquired the best sensitivity, specificity, accuracy, F1 score, MCC, and FMI, surpassing the second-best algorithm by nearly 1% in every measure. Besides, our method gained the least standard deviation compared state-of-the-art algorithms. These results showed the effectiveness of our proposed approach.

There are some shortcomings of our proposed approach: (i) The dataset we used in this study is not abundant. We will seek for bigger datasets or collecting more sample images. (ii) We will try some new deep learning technologies in multiple sclerosis detection such as attention mechanism.

Conclusions

In this study, we proposed a novel framework for multiple sclerosis detection using 6-layer stochastic pooling CNN combined with multiple-way data augmentation. We added stochastic pooling in our framework and tested its superiority to other pooling methods via comparison experiments. We also proposed 18-way data augmentation methods including geometric-based methods, noise-based methods, and photometric-based methods. Our approach beat several state-of-the-art algorithms, attaining a sensitivity of $95.98 \pm 0.46\%$, a specificity of $95.67 \pm 0.92\%$, a precision of $95.66 \pm 0.89\%$, an accuracy of $95.82 \pm 0.58\%$, and a F1 score of $95.81 \pm 0.57\%$. The experimental results showed that our approach achieved highest performance in multiple sclerosis detection compared to several state-of-the-art algorithms.

Conflict of interest

The authors declare that they have no conflicts of interest to disclose.

References

1. Dua M, Makhija D, Manasa PYL, Mishra P. A CNN–RNN–LSTM based amalgamation for Alzheimer's Disease detection. *J Med Biol Eng.* 2020;40(5):688-706.
2. Si X, Zhang X, Zhou Y, Sun Y, Jin W, Yin S, et al. Automated detection of juvenile myoclonic epilepsy using CNN based transfer learning in diffusion MRI. *IEEE*; 2020. p.1679-82.
3. Lee SM, Hyeon JW, Kim S-J, Kim H, Noh R, Kim S, et al. Sensitivity and specificity evaluation of multiple neurodegenerative proteins for Creutzfeldt-Jakob disease diagnosis using a deep-learning approach. *Prion.* 2019;13(1):141-50.
4. Yang Y, Yan LF, Zhang X, Han Y, Nan HY, Hu YC, et al. Glioma grading on conventional MR images: a deep learning study with transfer learning. *Front Neurosci.* 2018;12:804.
5. Wang S., Zhan T, Chen Y, Zhang Y, Yang M, Lu H, et al. Multiple sclerosis detection based on biorthogonal wavelet transform, RBF kernel principal component analysis, and logistic regression. *IEEE Access.* 2016. 4: p.7567-7576.
6. Nayak DR, Dash R, Majhi B. Brain MR image classification using two-dimensional discrete wavelet transform and AdaBoost with random forests. *Neurocomputing (Amsterdam).* 2016;177:188-97.
7. Zhang Y-D, Pan C, Sun J, Tang C. Multiple sclerosis identification by convolutional neural network with dropout and parametric ReLU. *J Comput Sci.* 2018;28:1-10.
8. Eitel F, Soehler E, Bellmann-Strobl J, Brandt AU, Ruprecht K, Giess RM, et al. Uncovering convolutional neural network decisions for diagnosing multiple sclerosis on conventional MRI using layer-wise relevance propagation. *Neuroimage Clin.* 2019;24:102003.
9. Alijamaat A, NikravanShalmani A, Bayat P. Multiple sclerosis identification in brain MRI images using wavelet convolutional neural networks. *Int J Imaging Syst Technol.* 2021;31(2):778-85.
10. Han J, Hou S-M. Multiple sclerosis detection via wavelet entropy and feedforward neural network trained by adaptive genetic algorithm. *Lect Notes Comput Sci.* 2019;11507:87-97.
11. Han J, Hou S-M. A multiple sclerosis recognition via hu moment invariant and artificial neural network trained by particle swarm optimization. *Lect Notes Inst Comput Sci Soc-Inform Telecommun Eng.* 2020;327:254-64.
12. Tang, C. Multiple sclerosis detection via 5-layer convolutional neural network. *J Med Imaging Health Inform.* 2021. 11: p. 1-8.
13. MRI lesion segmentation in multiple sclerosis database.
14. Alparslan E, Fuatince M. Image enhancement by local histogram stretching. *IEEE Trans Syst Man Cybern B Cybern.* 1981;11:376-85.
15. Pasban S, Mohamadzadeh S, Zeraatkar-Moghaddam J, Shafiei AK. Infant brain segmentation based on a combination of VGG-16 and U-Net deep neural networks. *IET Image Process.* 2020;14(17):4756-65.
16. Arulmozhi P, Abirami S. DSHPoolF: deep supervised hashing based on selective pool feature map for image retrieval. *Vis Comput.* 2021;37(8):2391-405.
17. Krestinskaya O, James AP. Analogue neuro-memristive convolutional dropout nets. *Proc Math Phys Eng Sci.* 2020;476(2242).
18. Zhang YD, Zhang Z, Zhang X, Wang SH. MIDCAN: A multiple input deep convolutional attention network for Covid-19 diagnosis based on chest CT and chest X-ray. *Pattern Recognit Lett.* 2021;150:8-16.
19. Clement JC, Indira N, Vijayakumar P, Nandakumar R. Deep learning based modulation classification for 5G and beyond wireless systems. *Peer Peer Netw Appl.* 2021;14(1):319-32.
20. Wang SH, Satapathy SC, Anderson D, Chen SX, Zhang YD. Deep fractional max pooling neural network for COVID-19 recognition. *Front Public Health.* 2021;9:726144.
21. Mahmoudi MA, Chetouani A, Boufera F, Tabia H. Learnable

- pooling weights for facial expression recognition. *Pattern Recognit Lett.* 2020;138:644-50.
22. Wu K. SOSPCNN: structurally optimized stochastic pooling convolutional neural network for tetralogy of fallot recognition. *Wirel Commun Mob Comput.* 2021;2021:5792975.
 23. Sahani M, Swain BK, Dash PK. FPGa based favourite skin colour restoration using improved histogram equalization with variable enhancement degree and ensemble extreme learning machine. *IET Image Process.* 2021;15:1247-59.
 24. Lyu C, Hu G, Wang D. Attention to fine-grained information: hierarchical multi-scale network for retinal vessel segmentation. *Vis Comput.* 2020.
 25. Satapathy SC, Wu D. Improving ductal carcinoma in situ classification by convolutional neural network with exponential linear unit and rank-based weighted pooling. *Complex Intell Syst.* 2021;7:1295-310.
 26. Gugushvili S, Mariucci E, der Meulen F. Decomposing discrete distributions: a nonparametric Bayesian approach. *Scand J Statist.* 2020; 47:464-492.
 27. Mukuno J, Matsui. Natural gradient descent of complex-valued neural networks invariant under rotations. *IEICE T Fund Electr.* 2019. E102A(12): p.1988-1996.
 28. Zhang Y-D, Dong Z, Wang S-H, Yu X, Yao X, Zhou Q, et al. Advances in multimodal data fusion in neuroimaging: Overview, challenges, and novel orientation. *Inf Fusion.* 2020;64:149-87.
 29. Wang SH, Zhou Q, Yang M, Zhang YD. ADVIAN: Alzheimer's disease VGG-inspired attention network based on convolutional block attention module and multiple way data augmentation. *Front Aging Neurosci.* 2021;13:687456.
 30. Ryan P, Dilley B, Davies D, Glass T, Abadi F. Short-term movement patterns, population estimates and breeding biology of an island endemic bird, the Tristan Thrush. *Revista Brasileira De Ornitologia.* 2019. 27(4): p.245-252.
 31. Chakraborty S, Khare K. Consistent estimation of the spectrum of trace class Data Augmentation algorithms. *Bernoulli.* 2019;25(4B):3832-63.
 32. Zhang Y, Zhang X, Zhu W. ANC: attention network for COVID-19 explainable diagnosis based on convolutional block attention module. *Comput Model Eng Sci.* 2021;127(3):1037-58.
 33. Shahid AI, Almotairi S. Automated Arabic sign language recognition system based on deep transfer learning. *Int J Comput Sci Net Secur.* 2019;19(10):144-52.
 34. Wang SH, Zhang Y, Cheng X, Zhang X, Zhang YD. PSSPNN: PatchShuffle stochastic pooling neural network for an explainable diagnosis of COVID-19 with multiple-way data augmentation. *Comput Math Methods Med.* 2021;2021:6633755.
 35. Aldabbas H, Asad M, Ryalat MH, Malik KR, Qureshi MZA. Data augmentation to stabilize image caption generation models in deep learning. *Int J Adv Comput Sci Appl.* 2019;10(10):571-9.
 36. Nguyen KD, Nguyen K, Le DD, Duong DA, Nguyen TV. YADA: you always dream again for better object detection. *Multimed Tools Appl.* 2019;78(19):28189-208.
 37. Zhang Y-D, Satapathy SC, Guttery DS, Górriz JM, Wang S-H. Improved breast cancer classification through combining graph convolutional network and convolutional neural network. *Inf Process Manag.* 2021;58(2):102439.
 38. Pho K, Mohammed Amin MK, Yoshitaka A. Segmentation-driven hierarchical retinaNet for detecting protozoa in micrograph. *Int J Semant Comput.* 2019;13(03):393-413.
 39. Wang SH, Nayak DR, Guttery DS, Zhang X, Zhang YD. COVID-19 classification by CCSHNet with deep fusion using transfer learning and discriminant correlation analysis. *Inf Fusion.* 2021;68:131-148.
 40. Arvanitakis K, Avlonitis M, Papadimitriou E. Introducing stochastic recurrence interval to classification algorithms for identifying asperity patterns. *Phys A: Stat Mech Appl.* 2018;512:566-77.
 41. Baquirin RBM, Fernandez PL. Artificial neural network (ANN) in a small dataset to determine neutrality in the pronunciation of English as a foreign language in Filipino Call Center Agents. *Inteligencia Artificial-Iberoamerica Journal of Artificial Intelligence.* 2018;21(62):134-44.