Abstract—Cerebral microbleeds (CMBs) are small perivascular hemosiderin deposits leaked through cerebral small vessels in normal or near normal tissue. The positions distribution of CMBs can indicate some underlying aetiologies. CMBs can be visualized by susceptibility-weighted imaging (SWI) which is high sensitivity to hemosiderin. In this paper, we proposed a hybrid method to detect the CMBs automatically. This method first applied discrete wavelet transform (DWT) to extract the features of the brain images, and then employed principal component analysis (PCA) to perform reduction of features. At last, the obtained features were inputted to back propagation shallow neural network (BPNN) with a single-hidden layer for training and prediction. K-fold cross validation was applied to avoid overfitting and evaluate the generalization ability of BPNN for selecting the best model. Based on this method, a good result was obtained with a sensitivity of 88.47± 0.96%, a specificity of 88.38± 1.00%, and an accuracy of 88.43± 0.97%, which is better than two state-of-the-art approaches.

Keywords—discrete wavelet transform, back propagation, shallow neural network, cerebral microbleeds

I. INTRODUCTION

With the improvement of medical technology, people's life span is also getting longer. This is not only due to the improvement of the technology for treating diseases but also due to the improvement of disease prevention technology. More and more researches show that there are relationships between cerebral microbleeds (CMBs) and intra-cerebral hemorrhage (ICH), which means CMBs can indicate some underlying aetiologies caused by ICH [1]. CMBs as a kind of subclinical sign can be detected in a few years before the onset of the disease, which provide an important reference for disease prevention [2].

CMBs are small perivascular hemosiderin deposits leaked through cerebral small vessels in normal or near normal tissue [3]. CMBs are visualized as small and rounded radiological entities by magnetic resonance imaging (MRI) due to the superparamagnetic of hemosiderin which lead to fast decay of signal. Developments in software and hardware of MRI have span is also getting longer. This is not only due to the

Above methods have achieved great improvements for detecting CMBs. To further explore the possibility of improving the accuracy of detecting CMBs, our team tried many methods and found an effective one which performed better than those state-of-the-art approaches based on traditional machine learning. Firstly, we used sliding neighborhood processing (SNP) technique to generate the input images and their target value. Then discrete wavelet transformation (DWT) was employed to extract the features of the input images. Afterwards, principal component analysis (PCA) was applied to reduce the features
and remove the possible correlated relationships between those features for avoiding the bias of prediction results. Finally, the features and their corresponding target values were sent to back propagation neural network (BPNN) for training and predicting.

II. MATERIALS

Ten patients diagnosed with cerebral microbleed and ten healthy controls (HCs) were enrolled. Syngo MR B17 software was employed to reconstruct the 3D SWI images. The size of the 3D SWI images are $364 \times 448 \times 48$. Three neuroradiologists with over twenty years of experience were employed to label the CMB voxels manually. In this study, both the voxels labelled “possible” and “definite” are considered as CMBs.

We used SNP technique to generate the input images and their target value. The size of sliding window is $7 \times 7$. The sliding window swept over twenty subjects and generated 68,847 CMB samples and 151,837,713 Non-CMB samples, more details about the procedure of generating the samples can be found in (Hou et al., 2017). In view of the imbalance of the samples, undersampling was used to remove the Non-CMBs randomly. Finally, 137,676 samples including 68,847 CMB samples and 68,829 Non-CMB samples were obtained.

III. METHODS

A. Discrete Wavelet Transformation

Discrete wavelet transformation (DWT) as an important means in signal processing has a critical advantage comparing to Fourier transforms: it can obtain both frequency and location information. It is available to obtain temporal resolution in multiple scale for DWT with this advantage[17-20]. The multiple scale information can be represented by a set of wavelet coefficients, and the wavelet coefficients are derived from decomposing the signal. Furthermore, wavelet coefficients can further be selected as features for classification.

Suppose there is a signal $x$, the scaling and translation parameters are $m$ and $n$ respectively, and their values are greater than 1 and 0 respectively. $i$ and $j$ are integers. The expression of DWT is as follows:

$$W[i,j] = \frac{1}{m^{1/2}} \sum_{t=0}^{\infty} x[t] \varphi\left(\frac{t-mjn}{m^i}\right)$$

where $\varphi$ denotes the wavelet function, $x[t]$ denotes the discretized signal function.

It is worth noting that the values of $m$ and $n$ are selected to make the mother wavelets form an orthonormal basis. It is practical to choose 2 and 1 as the values of $m$ and $n$ respectively for satisfying this condition. Afterwards, DWT can be implemented using filter blank methods when the coefficients can be considered as a filter [21]:

$$W(i,j) = \sum_{t=0}^{t-1} x(t) (2^{-i/2}) \varphi(2^{-j} t - j)$$

The meaningful of DWT is to decompose the original signal into multiple level coefficients. The method of decomposing signal developed by [22] can convert the signal to a series of coefficients including approximation and detail components in multiple level. The procedure of decomposing is shown in Fig. 1, $h(n)$ and $g(n)$ denote the low pass filter and the high pass filter respectively; $A(L)$ and $D(L)$ denote the approximation and detail coefficients of level $L$ respectively, and the $L$ is given according the specific situation. Approximation and detail coefficients are obtained after that signal is passed through $h(n)$ and $g(n)$ and downsampled which is used to maintain the length of the signal. Hence the approximation and detail coefficients represent the low and high frequency information of the signal respectively [23-26].

![Wavelet decomposition tree](image)

**Fig. 1.** Wavelet decomposition tree

B. Principal Component Analysis

Our goal is to classify the CMBs using those coefficients as the features of classification. Nevertheless, it is possible that some coefficients are correlated which can lead to bias of the prediction results and some coefficients are redundant which can make noise [27, 28].

PCA invented by [29] is a statistical technique for data analysis and processing. It converts the variables that possibly correlated to principal components that linearly uncorrelated using an orthogonal transformation. The principal components transformed by PCA sorted by variance. The first principal component has the largest possible variance and the second one has the second largest variance, and so on. Generally, the larger variance that the component has, the more important information that it has. Hence the PCA can perform reduction of variables by retaining a part of principal components that have larger possible variance and ignoring the rest part. The retained principal components contain most of the information of the original data set, and the relationships between the principal components are uncorrelated.

Consider a series of k-dimensional data vectors $\{x_n\}$ [30] which have been normalized, $n \in \{1, 2, \ldots, N\}$. The sample covariance matrix is calculated as $C = \sum_n (x_n - \bar{x})(x_n - \bar{x})^T/N$, where $\bar{x}$ is the average value of samples. The eigenvalues and eigenvectors can be calculated according to the covariance matrix $C$. The eigenvalues are in one-to-one correspondence with the eigenvectors. The eigenvectors corresponding to the q largest eigenvalues are retained and defined as $V = (v_1, v_2, \ldots, v_q)$. Finally, the q principal components are given as $p_n = (x_n - \bar{x})V^T$. The principal components $p_j$ are uncorrelated and lesser than $k$ in number generally.

C. Back Propagation Neural Network

Back propagation neural network (BPNN) as one kind of feed-forward neural-network is widely applied to pattern recognition, function approximation, classification, and so on [31-33].

BPNN was used as the classifier in this study. The features (namely principal components) derived from PCA were sent to
the neural network for training. The structure of BPNN contains input layer, hidden layer and output layer (Fig. 2). In this study, the number of the neurons in input layer is determined by the features, while that in hidden layer would be determined by grid searching method.

![Fig. 2. A two-layer BPNN structure](image)

Suppose the input is \( i \) and the target is \( j \). The values of neurons in the hidden layer can be expressed as below:

\[
g = f_h(w_h^T i + b_h) \quad (3)
\]

where \( f_h, w_h, \) and \( b_h \) are activation function, weights, and biases of hidden layer neurons, respectively. Then the values of neurons in output can be expressed as below:

\[
k = f_o(w_o^T g + b_o) \quad (4)
\]

where \( f_o, w_o, \) and \( b_o \) are activation function, weights, and biases of output neurons, respectively. It is worth noting that all of the weights and biases derived from learning via back propagation. Scaled conjugate gradient (SCG) method [34-36] was employed for training the BPNN. The activation functions need to be given before training. Sigmoid function was selected as the activation functions of hidden layer neurons and output neurons in this study.

D. K-fold Cross Validation

K-fold cross validation was applied to evaluate the generalization ability of BPNN for selecting the best model. The samples were randomly divided into \( K \) folds and each fold substantially contains the same number of samples. Then \( K-1 \) folds were used as the training samples and the rest one-fold was used as the test samples at every run. \( K \) runs were implemented [37, 38]. Finally, \( K \) test errors would be obtained, and their average value is the overall error of the model. In this study, the samples were divided into ten folds (\( K \) was set to 10), and the cross-validation procedure is shown in Fig. 3.

![Fig. 3. K-fold cross validation workflow](image)

E. Evaluation

Three statistical indicators were applied to evaluate the performance of our method. They are sensitivity, specificity and accuracy, and they defined as below:

\[
\text{Sensitivity} = \frac{TP}{(TP + FN)} \quad (5)
\]

\[
\text{Specificity} = \frac{TN}{(TN + FP)} \quad (6)
\]

\[
\text{Accuracy} = \frac{(TP + TN)}{(TP + TN + FN + FP)} \quad (7)
\]

TP represents a CMB voxel classified correctly, TN represents a Non-CMB voxel classified correctly, FP represents a Non-CMB voxel incorrectly classified as CMB, FN represents a CMB voxel incorrectly classified as Non-CMB.

IV. RESULTS

The hyperparameters of our method was tuned by either trial-and-error or grid search method. We finally set those hyperparameters in Table I.

<table>
<thead>
<tr>
<th>TABLE I. HYPERPARAMETER SETTING OF OUR METHOD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wavelet Name</td>
</tr>
<tr>
<td>Wavelet decomposition</td>
</tr>
<tr>
<td>PCA threshold</td>
</tr>
<tr>
<td># hidden neurons</td>
</tr>
<tr>
<td>Activation function</td>
</tr>
<tr>
<td>Neural network training method</td>
</tr>
</tbody>
</table>

The results over 10x10-fold cross validation was shown below in Table II and Fig. 4. Here we obtained a sensitivity of 88.47± 0.96%, a specificity of 88.38± 1.00%, and an accuracy of 88.43± 0.97%.

<table>
<thead>
<tr>
<th>TABLE II. STATISTICAL RESULT OF OUR METHOD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>9</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>Average</td>
</tr>
</tbody>
</table>

![Fig. 4. Statistical of our method](image)
Fig. 5. Performance against number of hidden neurons

We used grid search to check the optimal number of hidden neurons of this back-propagation neural network. Suppose n represents the number of hidden neurons, and we vary n from 10 to 15. The results are shown in Table III and Fig. 5, where the best results were highlighted in bold.

<table>
<thead>
<tr>
<th>Value of n</th>
<th>Sensitivity (±)</th>
<th>Specificity (±)</th>
<th>Accuracy (±)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>86.94±1.22</td>
<td>86.97±1.06</td>
<td>86.95±1.14</td>
</tr>
<tr>
<td>11</td>
<td>87.96±0.45</td>
<td>87.80±0.57</td>
<td>87.88±0.50</td>
</tr>
<tr>
<td>12</td>
<td><strong>88.47±0.96</strong></td>
<td><strong>88.38±1.00</strong></td>
<td><strong>88.43±0.97</strong></td>
</tr>
<tr>
<td>13</td>
<td>87.75±0.86</td>
<td>87.46±1.11</td>
<td>87.61±0.96</td>
</tr>
<tr>
<td>14</td>
<td>86.61±0.90</td>
<td>86.51±1.11</td>
<td>86.56±1.00</td>
</tr>
<tr>
<td>15</td>
<td>86.17±1.07</td>
<td>86.22±1.26</td>
<td>86.20±1.15</td>
</tr>
</tbody>
</table>

Table III. Performance against number of hidden neurons

Finally, we compared our method with two state-of-the-art approaches: (i) random forest (RF) [9]; and (ii) wavelet entropy and naive Bayesian classifier [16]. The results are shown in Table IV.

<table>
<thead>
<tr>
<th>Method</th>
<th>Sensitivity (%)</th>
<th>Specificity (%)</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RF [9]</td>
<td>85.7%</td>
<td>99.5%</td>
<td>Not available</td>
</tr>
<tr>
<td>WE-RBF [16]</td>
<td>76.90</td>
<td>76.91</td>
<td>76.90</td>
</tr>
<tr>
<td>Proposed</td>
<td><strong>88.47±0.96</strong></td>
<td><strong>88.38±1.00</strong></td>
<td><strong>88.43±0.97</strong></td>
</tr>
</tbody>
</table>

Table IV. Comparison to state-of-the-art approaches

V. CONCLUSIONS

In this paper, a hybrid method was developed to detect cerebral microbleeds. Compared to two state-of-the-art approaches, the result of this method achieved a better performance with a sensitivity of 88.47±0.96%, a specificity of 88.38±1.00%, and an accuracy of 88.43±0.97%. Future work should focus on collecting more brain images.

REFERENCES


