Research Article

Detection of Dendritic Spines Using Wavelet-Based Conditional Symmetric Analysis and Regularized Morphological Shared-Weight Neural Networks

Shuihua Wang, 1,2 Mengmeng Chen, 3,4,5 Yang Li, 1 Yudong Zhang, 2,6 Liangxiu Han, 7 Jane Wu, 3,4 and Sidan Du 1

1 Department of Electronic Engineering, Nanjing University, Nanjing 210024, China
2 School of Computer Science and Technology, Nanjing Normal University, Nanjing 210023, China
3 State Key Laboratory of Brain and Cognitive Science, Institute of Biophysics, Chinese Academy of Sciences, Beijing 100101, China
4 Department of Neurology, Lurie Cancer Center, Center for Genetic Medicine, Northwestern University School of Medicine, Chicago, IL 60611, USA
5 University of Chinese Academy of Sciences, Beijing 100101, China
6 Translational Imaging Division, Columbia University, New York, NY 10032, USA
7 School of Computing, Mathematics and Digital Technology, Manchester Metropolitan University, Manchester M1 5GD, UK

Correspondence should be addressed to Sidan Du; coff128@nju.edu.cn

Received 17 June 2015; Revised 2 September 2015; Accepted 27 September 2015

Academic Editor: Valeri Makarov

Copyright © 2015 Shuihua Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Identification and detection of dendritic spines in neuron images are of high interest in diagnosis and treatment of neurological and psychiatric disorders (e.g., Alzheimer's disease, Parkinson's diseases, and autism). In this paper, we have proposed a novel automatic approach using wavelet-based conditional symmetric analysis and regularized morphological shared-weight neural networks (RMSNN) for dendritic spine identification involving the following steps: backbone extraction, localization of dendritic spines, and classification. First, a new algorithm based on wavelet transform and conditional symmetric analysis has been developed to extract backbone and locate the dendrite boundary. Then, the RMSNN has been proposed to classify the spines into three predefined categories (mushroom, thin, and stubby). We have compared our proposed approach against the existing methods. The experimental result demonstrates that the proposed approach can accurately locate the dendrite and accurately classify the spines into three categories with the accuracy of 99.1% for “mushroom” spines, 97.6% for “stubby” spines, and 98.6% for “thin” spines.

1. Introduction

Dendritic spines are small "doorknob" shaped extensions from neuron's dendrites, which can number thousands to a single neuron. Spines are typically classified into three types based on the shape information: mushroom, stubby, and thin. “Mushroom” spine has a bulbous head with a thin neck; “stubby” spine only has a bulbous head; “thin” spine has a long thin neck with a small head. Research has shown that the changes in shape, length, and size of dendritic spines are closely linked with neurological and psychiatric disorders, such as attention-deficit hyperactivity disorder (ADHD), autism, intellectual disability, Alzheimer's disease, and Parkinson's disease [1–5]. Therefore, the morphology analysis and identification of structure of dendritic spines are critical for diagnosis and further treatment of these diseases [6, 7].

Traditional manual detection approach of dendritic spines detection is costly and time consuming and prone to error due to human subjectiveness. With the recent advances in biomedical imaging, computer-aided semiautomatic or automatic approaches to detect dendritic spines based on
image analysis have shown the efficacy. SynD method proposed by Schmitz et al. [8] is a semiautomatic image analysis routine to analyze dendrite and synapse characteristics in immune-fluorescence images. For the fluorescence imaging, the neurite and soma were captured in the separated imaging channels. In that case, soma and synapse were detected without intervention from neurite [9–11] based on the channel information. However, this method cannot be extended to the images, of which the information is captured in the same channel. Therefore, many other methods were proposed to solve this problem, for instance, ImageJ [12], NeuronStudio [13], Neuron [14], and NeuronIQ [15]. However, these methods have some limitations. For example, NeuronIQ was designed for the confocal multiphoton laser scanning. Neuron was used to trace the dendrite growing in the condition of manually marking the dendrite first. Koh et al. detected spines from stacks of image data obtained by laser scanning microscopy [16]. The algorithm first extracted the dendrite backbone defined as the medial axis and then geometric information was employed to detect the attached and detached spines according to the shape of each candidate spine region. Features including spine length, volume, density, and shape for static and time-lapse images of hippocampal pyramidal neurons were used as key points for the detection. The disadvantage of this method is that it might lose many spines during the detection because of the thresholding method used in this case. To overcome this problem, Xu et al. proposed a new detection algorithm for the attached spines from the dendrites by two grassfire steps [17]: a global threshold was chosen to segment the image and then the medial axis transform (MAT) was applied to find the centerlines of the dendrites. Then some large spines (noncenterlines) were removed from the centerlines. After the backbone was extracted, two grassfire procedures were applied to separate the spine and dendrite. The results of the proposed method were similar to the results of the manual method. Cheng et al. proposed a method using an adaptive threshold based on the local contrast to determine the foreground, containing the spine and dendrite, and detect attached and detached spines [18]. Fan et al. used the curvilinear structure detector to find the medial axis of the dendrite backbone and spines attached to the backbone [19]. To locate the boundary of dendrite, an adaptive local binary fitting (aLBF) energy level set model was proposed for localization. Zhang et al. extracted the boundaries and the centerlines of the dendrite by estimating the second-order directional derivatives for both the dendritic backbones and spines [20]. Then a classifier based on Linear Discriminate Analysis (LDA) was built to classify the attached spines into true and false types. The accuracy of the algorithm was calculated according to the backbone length, spine number, spine length, and spine density. Janoos et al. used the medial geodesic to extract the centerlines of the dendritic backbone [21]. He et al. proposed a method based on NDE to classify the dendrite and spines [22]. The principle of their method was that spine and dendrite had different shrink rates. Shi et al. proposed a wavelet-based supervised method for classifying 3D dendritic spines from neuron images [23].

Existing work is encouraging. However, the problems remain on how to improve accuracy (e.g., accurate extraction of backbone, accurate detection of attached and detached spines). Different from existing approaches, in this paper, we have proposed new algorithms for efficient detection of dendritic spines using wavelet-based conditional symmetric analysis and regularized morphological shared-weight neural network. Our contributions include the following:

1. A new extraction model for dendrite backbone and its boundary localization using wavelet-based conditional symmetric analysis and pixel intensity difference, which can allow accurate extraction of backbone, the first important step for dendritic spines.

2. A new way for spine detection based on regularized morphological shared-weight neural networks (RMSNN) to efficiently detect spines and classify them into right categories, that is, mushroom, thin, and stubby.

The rest of this paper is organized as follows. Section 2 describes the proposed methods including wavelet-based conditional symmetry analysis and pixel intensity difference for the dendrite detection and localization and regularized shared-weight neural networks for the spine detection. In Section 3, we have conducted experimental evaluation and demonstrated the effectiveness of the proposed algorithm. Section 4 discusses the results. Section 5 concludes the proposed approach and highlights the future work.

2. Methods

Figure 1 shows the steps of our proposed approach to dendritic spines. In the image acquisition phase, we demonstrated the process for the neuron culture, label, and imaging. In the second step, we preprocessed the images by reducing the noise and smoothing the background [24, 25]. Then, we extracted the dendrite backbone based on the conditional symmetric analysis and located the dendrite boundary based on the difference of the pixel intensity. Afterwards, the spines were detected, classified, and characterized by RMSNN.

2.1. Image Acquisition. The neurons used for imaging in this paper were cortical neurons, primary cultured from Embryonic 18th- (E18-) day rat and next cultured until the 22nd day in vitro. Then, the neurons were transfected by Lipofectamine 2000 and imaged at the 24th day by Leica SP5 confocal laser scanning microscopy (CLSM) by 63x. The size of the image is 1024 × 1024, and the resolution is 0.24 um/pixel at the confocal layer. The images used for the morphology analysis were obtained by the maximum intensity projection (MIP) of the original 3D image stack. As the images were captured as Z-stack series, we projected the 3D image stack onto the xy, yz, and zx planes, respectively. Since the slices along the optical direction (z) provided very limited information and the computation time based on the 3D image stacks is highly increased, it was desired to consider only the 2D projection onto the xy plane. The 2D image used for analysis was a maximum intensity projection of
Embryonic (E18) rat
Primary cultured cortical neurons
Transfected (22nd day) by Lipofectamine 2000
Imaging (24th day) by Leica SP5 (CLSM) by 63x

Figure 1: Flowchart of the proposed detection method of the dendritic spines.

Figure 2: Samples of the subimages used in the image library.

2.2. Image Preprocessing. Considering the limitation of imaging technique, we have employed the 2D median filter to deal with the noise introduced by the imaging mechanism of the photomultiplier tubes (PMT) and then used the partial differential equation (PDE) proposed by Wang et al. [26] to enhance the image. Figure 3 shows an example of the original image and the preprocessed result.

2.3. Backbone Extraction Using the Wavelet Transformation Based Conditional Symmetric Analysis. Considering the attached spines, it is necessary to firstly locate the dendrites in order to segment the spines from the dendrite. The backbone extraction and boundary localization are critical for dendritic spine classification and analysis, which include the following steps.

Step 1. Remove the noise and small isolated point-set.

Step 2. Locate the backbone of the dendrite.

Step 3. Locate the boundary of the dendrite.

The backbone is defined as the thinning of the dendrite. Due to the variance of width of dendrite, attached and detached spines, it is a challenging task to locate the boundary.
of the dendrite directly from the preprocessed images. Therefore, we have developed a new extraction model utilizing wavelet transform based conditional symmetric analysis. The essence of this model is to conduct a local conditional symmetry analysis of the contour of the region of interest (ROI) and then compute the center points to produce the backbone of the dendrite.

Due to the complexity of the dendrites and dendrite spines’ distribution, we have employed morphological operation to remove the small isolated point-set for the dendrite in the binary image obtained by local Otsu [27–29] via (1), which could decrease the disconnection rate of the dendrite detection:

\[
P = \begin{cases} 
1, & \text{more than } n \text{ positive pixels in its 3-by-3 window}, \\
0, & \text{otherwise},
\end{cases} \tag{1}
\]

in which \(n\) is the threshold of the number of positive pixels. The value of \(n\) could be determined by trial and error method and means that the pixel belongs to the major line if there are more than \(n\) positive pixels in its \(3 \times 3\) neighborhood window. Otherwise, the value of the pixel is forced to be 0, treated as the small isolated point-set. The determination of the centerline of the dendrite is based on the conditional symmetric analysis.

The symmetric analysis was accomplished via the wavelet transform. We have applied the wavelet transform to detect a pair of contour curves:

\[
\varphi_x(x, y) = \frac{\partial}{\partial x} \theta(x, y) = \phi'(\sqrt{x^2 + y^2}) \frac{x}{\sqrt{x^2 + y^2}},
\]

\[
\varphi_y(x, y) = \frac{\partial}{\partial y} \theta(x, y) = \phi'(\sqrt{x^2 + y^2}) \frac{y}{\sqrt{x^2 + y^2}}, \tag{2}
\]

in which \(x\) and \(y\) stand for the coordinate of the contour curve. \(\varphi_x(x, y)\) means the partial derivative of \(x\) and \(\varphi_y(x, y)\) stands for the partial derivative of \(y\), respectively. \(\theta(x, y)\) is a low pass filter.

For \(\varphi_x(x, y)\) and \(\varphi_y(x, y)\), the scale wavelet transform (WT) could be written as the following equations:

\[
W_{x,s}f(x, y) = (f * \varphi_{x,s})(x, y) = s \frac{\partial}{\partial x} (f * \theta_s)(x, y),
\]

\[
W_{y,s}f(x, y) = (f * \varphi_{y,s})(x, y) = s \frac{\partial}{\partial y} (f * \theta_s)(x, y). \tag{3}
\]

Here, \(\theta_s = (1/s^2)\theta(x/s, y/s)\). We can get the modulus of the gradient vector as

\[
\nabla W_s f(x, y) = \left( \begin{array}{c} W_{x,s}f(x, y) \\ W_{y,s}f(x, y) \end{array} \right), \tag{4}
\]

\[
|\nabla W_s f(x, y)| = \sqrt{\left| W_{x,s}f(x, y) \right|^2 + \left| W_{y,s}f(x, y) \right|^2}, \tag{5}
\]

\[
A_s f(x, y) = \arctan \left( \frac{W_{y,s}f(x, y)}{W_{x,s}f(x, y)} \right), \tag{6}
\]

where \(\nabla\) is the gradient vector and the gradient direction is given as (6). The contour points \((x, y)\) are the local maxima of \(|\nabla W_s f(x, y)|\) in the direction of \(A_s f(x, y)\) at scale \(s\). However, the local maxima modulus is not the exact edge point.

We selected (7) as the basis function. We set \(\varphi^\ast(-x) = -\varphi^\ast(x)\) and had \(\varphi(x) = \varphi^\ast(x) + \varphi^\ast(-x)\) as the wavelet function, which had the following properties: gray invariant, slope invariant, width invariant, and symmetric [29, 30]. The advantage is to make the extraction of a pair of contours with accurate protrusions. Consider
The distance between two symmetric points is equal to the scale of the wavelet transform. If the distance between two symmetric points is larger than or equal to the width of regular region, the center point of the symmetric pair can potentially be located outside of the dendrite. The regular region is defined as the dendrite is smooth, where the function has a stable variation along the axis. Thus, we defined the stable symmetry as follows.

If the scale of wavelet transform is larger than or equal to the width of regular region, the modulus maxima points generate two new parallel contours inside the periphery of the dendrite. All the symmetric pairs of the wavelet transforms that do not have a counterpart are defined as the unstable symmetry. In this case, we have considered the width as the constraint condition. In the direction of the perpendicular to the gradient direction, we selected the width nearest to the regular region.

The center of every symmetric pair located on the centerline of the original regular region of the stroke point. Finally, the backbone of the regular region was defined by the curve of all connected symmetric points.

2.4. Boundary Location Based on the Pixel Intensity Difference.

The morphological operation of removing noise blurred the boundary. Therefore, after localization of backbone, the boundary of the dendrite was detected via varies of the pixel intensity of the preprocessed image from Section 2.2. We can observe that the pixel intensity of the line pixel changes abruptly at the boundary locations. The boundary location was performed in two steps. In the first step, we have searched the image along the two directions perpendicular to the local line direction until the pixel intensity of the line pixel changed sharply. We set a threshold for each pixel. The local line direction is determined as

\[
A_{i}f(x, y) = \arctan \left( \frac{W_{y,i}f(x, y)}{W_{x,i}f(x, y)} \right).
\]

The formulation of each pixel is given by \((\alpha, I(p))\), in which \(I(p)\) is the pixel intensity of point \(p\) in the original image and \(\alpha\) is a predefined pixel intensity value, that is,

\[
\begin{align*}
1(p) &\geq \alpha, \quad p \text{ belongs to the line pixel} \\
1(p) &< \alpha, \quad p \text{ does not belong to the line pixel.}
\end{align*}
\]
including laser radar (LADAR), forward-looking infrared (FLIR), synthetic aperture radar, and visual spectrum image. The existing research demonstrates that the MSNN is robust for detection with rotation, image intensity translation, and occlusion variables [36]. In this paper, we have proposed to apply the regularized morphological shared-weight neural network to spine classification.

Dilation is defined as
\[ A \oplus B = \{ x | (\hat{B})_x \cap A \neq \emptyset \}, \quad (10) \]
in which \( A \) and \( B \) are sets in \( \mathbb{Z}^2 \) and \( \hat{B} \) is the reflection of \( B \). \( \emptyset \) is the empty set. Equation (10) is termed the dilation of \( A \) by \( SE \). Dilation is the reflection of \( B \) about its origin, then translated by \( x \), with the set of all \( x \), which allow \( \hat{B} \) to intersect \( A \) with at least one element.

Erosion is defined as (11) or (12) by the duality of the erosion-dilation relationship:
\[ A \ominus B = \{ x | (B)_x \subseteq A \}, \quad (11) \]
\[ A \ominus B = (A^c \oplus \hat{B})^c, \quad (12) \]
in which \( A^c \) is defined as the complement of \( A \).

Hit-Miss transform is defined as an operation that detects a given pattern in a binary image based on a pair of disjoint structure elements, one for Hit and the other one for Miss. The result of the Hit-Miss transform is a set of positions, structure elements, one for Hit and the other one for Miss.

A Hit-Miss transform is defined as an operation that detects how a shape \( f \) fits under \( f \) using erosion and how a shape \( m \) fits above \( f \) via dilation. The high value of Hit-Miss transform means good fit. The gray scale Hit-Miss transform is independent of shifting in gray scale.

2.5.1. The Feature Extraction Phase. There are four elements associated with each layer of feature extraction phase: feature maps, input, and two structure elements. In the first layer, the subimage is used as input, and the last layer’s output is the input of the classification stage. In each feature extraction layer, a pair of Hit-Miss SEs is shared within all the feature maps. These SEs are translated as input weights for the feature map nodes in the feature extraction layer. Table 1 shows the input parameters and output parameters related to the feature extraction phase.

According to the above parameters, we can define the Hit-Miss transform as follows:
\[ \text{net}^h_y = \min_{x \in D_{ij}} \{ a(x) - f^h_y(x) \}, \]
\[ \text{net}^m_y = \max_{x \in D_{ij}} \{ a(x) - f^m_y \}, \quad (19) \]
\[ a_y = \text{net}^h_y - \text{net}^m_y. \]
Here, \( \text{net}^h_y \) stands for the input for Hit operation in node \( y \) and \( h \) means the Hit operation. \( \text{net}^m_y \) means the net input for the Miss operation in node \( y \). \( m \) and \( \hat{m} \) here mean the Miss operation and reflection of \( m \), respectively. \( a_y \) is the result of Hit-Miss transform performed at node \( y \). The learning rule...
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a(x) )</td>
<td>The input to a node ( y ) from node ( x )</td>
</tr>
<tr>
<td>( t_{y}(x) )</td>
<td>Connections associating the node ( y ) with node ( x )</td>
</tr>
<tr>
<td>Input</td>
<td></td>
</tr>
<tr>
<td>( t^{h}_{y}(x) )</td>
<td>Hit SE associating node ( y ) with node ( x )</td>
</tr>
<tr>
<td>( t^{m}_{y}(x) )</td>
<td>Miss SE associating node ( y ) with node ( x )</td>
</tr>
<tr>
<td>( w^{h}_{y}(x) )</td>
<td>Weight for Miss SE node ( y ) with node ( x )</td>
</tr>
<tr>
<td>( w^{m}_{y}(x) )</td>
<td>Weight for Hit SE node ( y ) with node ( x )</td>
</tr>
</tbody>
</table>

Output \( a_{y} \): The output of node \( y \) for the Hit and Miss SE is derived based on the gradient decent as

\[
\Delta t_{y}^{h} = \eta \delta_{y} \frac{\partial \text{net}_{y}^{h}}{\partial y}(x),
\]

\[
\Delta t_{y}^{m} = -\eta \delta_{y} \frac{\partial \text{net}_{y}^{m}}{\partial y}(x),
\]

where \( \eta \) is the learning rate of the network and \( \delta_{y} \) is expressed as

\[
\delta_{y} = \delta(y) = \sum_{k} \delta_{k} w_{k}(y).
\]

Equation (21) is for the top level or final extraction layer. \( \delta_{y} \) for the lower layers of multiple-layer feature extraction is expressed as

\[
\delta_{y} = \delta(y) = \sum_{k} \delta_{k} \left( \frac{\partial \text{net}_{y}^{h}}{\partial a}(y) - \frac{\partial \text{net}_{y}^{m}}{\partial a}(y) \right),
\]

in which \( k \) is the node in the layer next to the node \( y \).

For the output layer nodes, \( w_{kj} \) stands for the connection strength to node \( k \) from node \( j \):

\[
\delta_{j} = (t_{j} - O_{j}) f'(\text{net}_{j})
\]

and for the hidden layer nodes,

\[
\delta_{j} = f'(\text{net}_{j}) \sum_{k} \delta_{k} w_{kj}.
\]

2.5.2. The Classification Phase. The classification phase takes the output directly from the last feature extraction layer as its input. The parameters used for the classification phase are predefined in the feature extraction phase. There are three output nodes for the classification stage of our algorithm, indicating which type of spines the subimage contains.

2.5.3. Acceleration of the MSNN Based on the Regularization. In order to accelerate the learning rate and decrease the learning epochs, we employed the regularization factor. Regularization is used to reduce near-zero connection weight value to zero, therefore reducing the complexity of the network. It is defined as

\[
R(w) = E_{c}(w) + \lambda E_{c}(w),
\]

\[
E_{c}(w) = \sum_{w \text{ in network}} \frac{(w_{i}w_{o})^{2}}{1 + (w_{i}w_{o})^{2}},
\]

where \( E_{c}(w) \) is the performance measure of the learning algorithm, the total network error, and \( E_{c}(w) \) is the complexity penalty of the network model. \( \lambda \) is the regularization factor. \( w_{o} \) is a predefined parameter. Meanwhile, research shows that a network with proper SEs produces better result [36]. Therefore, it is essential to choose the suitable SEs. In this paper, according to the average size of spine and the comparison result in Table 3, we defined the SE as a disk with the radius of 4 pixels.

For the training procedure, the RMSNN takes the subimage as the input and makes one output value for each image. For the testing procedure, our proposed algorithm scans the whole ROI and generates an image named the detection plane, which is based on the outputs from the target class nodes.

3. Experimental Evaluation

3.1. Experiment Design. We have trained neural networks with the back-propagation algorithm. The subimages were submitted to the input nodes of the neural network. The error of the output was propagated through all the connections. The process repeated until the network converged to a stable state with required MSE. When the MSE approximated to a preset value or the maximum epoch was achieved, the algorithm converged and the training would stop. During the training, the RMSNN took each subimage as the input and produced one output value for each of the three categories. Figure 2(a) shows the samples of subimages containing mushroom type.
spine. Figure 2(b) shows the samples of the subimages containing the stubby type, and Figure 2(c) shows the samples of thin type subimage.

In the training step, the subimage samples were input to the network sequentially. The median-squared error was employed to measure the training effectiveness. For each subimage, the RMSNN produced one output value, which indicated the type of spine in the subimage. Then, we scanned the entire microscopy image and finally generated a detection plane according to the output nodes of RMSNN.

In order to test the classification accuracy, we randomly selected 900 samples for each type of spine, respectively. Following common convention and ease of stratified cross validation, 10 \times 10-fold stratified cross validation (CV) was used for the dataset to perform an unbiased statistical analysis. The RMSNN was constructed in the form as two feature extraction layers, one hidden layer with ten hidden neurons and one output layer with three neurons. The input subimage size was 20 by 20 pixels, and the size of the structure elements was with the radius of 4 pixels. The initial weight was in the range of $[-1.0, 1.0]$. The learning rate was set to 0.0015. The maximum training epoch was predefined as 15000. The expected output values for mushroom, stubby, and thin type spines were $[1\ 0\ 0]$, $[0\ 1\ 0]$, and $[0\ 0\ 1]$.

3.2. Experiment Results

3.2.1. Backbone Extraction. The extraction result is shown in Figure 5. Figure 5(a) shows the original image. Figure 5(b) shows the extracted backbone, of which the width covers merely one pixel.

3.2.2. Boundary Location. Figure 6(a) shows the mark of the located backbone of the dendrite based on the original image, and Figure 6(b) shows the marked boundary of the dendrite after the backbone is extracted. Figure 6(c) shows the marked dendrite that determines the starting point of the spine.

3.2.3. Spine Analysis. Figure 7 shows a ROI of our sample image, and Figure 7(b) shows the detection result of the spines. The backbone is marked by the purple color and the boundary is marked by the red color. The spines are marked by their periphery of blue color.

Figure 8(a) shows the original image with the marked region of interest. Figure 8(b) shows the classification result based on the features extracted in the first phase. The corresponding SE gets respect features around each pixel, but it is blind for readers to understand which features are obtained. The detected spines contain 8 mushroom types, 8 stubby types, and 4 thin types. The average of the classification accuracy of RMSNN is shown in Table 2 based on the 2700 samples is shown in Table 3. We can find that the disk with a radius of 4 pixels reaches the best performance. Therefore, we finally defined the SEs as a disk with the radius of 4 pixels.

3.2.4. Algorithm Comparison. To further validate the efficacy of our proposed approach, we have compared the proposed algorithm with Cheng et al.’s method [18] and the manual method. In Cheng et al.’s paper, the authors employed the adaptive threshold to segment the image and Chen and Molloi’s algorithm [37] to extract the backbone and local spine morphology for the detection of the attached spines. The comparison results based on ROI in Figure 8 and 15 images collected in our database are shown in Table 4. It is found from Figure 9 that Cheng et al.’s method missed some small protrusions whose number of pixels is more than 5. The number of detected spines via our algorithm is 19, 13 by Cheng et al.’s method, and 20 via the manual method as shown in Table 4. Cheng et al.’s method is robust at dealing with the spines detached from the dendrite but weak at spines attached with the dendrite. However, the detached spines from the dendrite are caused by the deconvolution to denoise the image. Our proposed algorithm overcomes the problem of detecting attached spines.

4. Discussion

In this paper, we have proposed new algorithms using conditional symmetric analysis and regularized morphological shared-weight neural network to detect and analyze the dendrite and dendritic spines.

Figure 5 shows that backbone extraction result based on the conditional symmetry analysis. Compared to the second-order directional derivatives method in [14], our proposed algorithms reduced the computation time of linking the breaking point of the backbone.

Figure 6 shows the result of the marked backbone and the boundary of the dendrite, which is used to determine the starting point of the spines.

Table 2 shows the classification result of the different types of spines. The row in Table 2 stands for the actual class and the column in Table 2 stands for the predicted class. The “mushroom” type has an obvious head and thin neck. The “stubby” type lacks obvious neck, and the “thin” type lacks obvious head. In Table 2, the detection accuracy of

<table>
<thead>
<tr>
<th>Spine types</th>
<th>Mushroom</th>
<th>Stubby</th>
<th>Thin</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mushroom</td>
<td>99.1%</td>
<td>1.3%</td>
<td>1.1%</td>
</tr>
<tr>
<td>Stubby</td>
<td>0.7%</td>
<td>97.6%</td>
<td>0.3%</td>
</tr>
<tr>
<td>Thin</td>
<td>0.2%</td>
<td>1.1%</td>
<td>98.6%</td>
</tr>
</tbody>
</table>

Table 2: Average of the classification accuracy on a 10-by-10 CV.
### Table 3: Classification accuracy by different SEs (unit is in pixel, bold denotes the best, \( r \) is radius, and \( w \) is width).

<table>
<thead>
<tr>
<th></th>
<th>Disk ((r = 5))</th>
<th>Disk ((r = 4))</th>
<th>Disk ((r = 3))</th>
<th>Square ((w = 3))</th>
<th>Square ((w = 4))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mushroom</td>
<td>98.7%</td>
<td><strong>99.1%</strong></td>
<td>95.4%</td>
<td>85.3%</td>
<td>89.2%</td>
</tr>
<tr>
<td>Stubby</td>
<td>96.2%</td>
<td><strong>97.6%</strong></td>
<td>94.1%</td>
<td>87.2%</td>
<td>91.2%</td>
</tr>
<tr>
<td>Thin</td>
<td>94.3%</td>
<td><strong>98.6%</strong></td>
<td>96.2%</td>
<td>79.1%</td>
<td>75.3%</td>
</tr>
</tbody>
</table>

The mushroom type is higher than the other two types, and part of the stubby type is misclassified into mushroom and thin types as its head and neck ratio is at the level of average. A percent of 1.1 of thin spines are misclassified into mushroom type and 0.3% into stubby type, which is caused by the similar size of the head and neck. Table 4 shows the result of detected spines of Figure 8, respectively, by manual, ALS [18], and our proposed method SRMSNN. The results demonstrate...
that our algorithm has better performance than the other two methods for the images obtained by the confocal laser scanning microscopy.

5. Conclusion

In this paper, we proposed a new automatic approach to accurately identify dendritic spines with different shapes. The novelty of this approach includes (1) a new model using wavelet-based conditional symmetry analysis for dendrite backbone extraction and localization, which is the first step towards identification of dendritic spines; (2) a new algorithm based on regularized morphological shared-weight neural networks for classification of spines into the right classes (i.e., mushroom, stubby, and thin), entitled “RMSNN.” This research was based on our collected microscopy images. We
have applied our approach to image base library containing around 2700 subimage samples, 900 for each type of spines, and have compared the proposed method with the existing methods. The experimental results demonstrate that our algorithm outperforms existing methods with a significant improvement in accuracy in terms of classifying spines into the different spine categories. The classification accuracy is 99.1% for mushroom spines, 97.6% for stubby spines, and 98.6% for thin spines.

The future work will be focusing on further validation of the robustness of the algorithms through collecting more samples and testing on different datasets. A user-friendly interface will be also built for usability improvement and enhancement. Meanwhile, we will be focusing on reducing the computation time while improving the classification accuracy based on the 3D image stacks. Other feature extraction tools (such as wavelet packet analysis [38], wavelet entropy [39], and 3D-DWT [40]) and other advanced classification tools [41,42] will be tested. Besides, swarm intelligence method will be used to find optimal parameters [43].

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work was financially supported by the National Natural Science Foundation of China (no. 61271231).

References


