

Novel Approach Using Echo State Networks for Microscopic Cellular Image Segmentation

Boudjelal Meftah, Olivier Lézoray, A. Benyettou

▶ **To cite this version:**

Boudjelal Meftah, Olivier Lézoray, A. Benyettou. Novel Approach Using Echo State Networks for Microscopic Cellular Image Segmentation. Cognitive Computation, 2016, 8 (2), pp.237-245. 10.1007/s12559-015-9354-8 . hal-01252847

HAL Id: hal-01252847
https://hal.science/hal-01252847

Submitted on 12 Jan 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Novel approach using Echo State Networks for Microscopic Cellular Image Segmentation

Boudjelal Meftah · Olivier Lézoray · Abdelkader Benyettou

Received: date / Accepted: date

Abstract This paper concentrates on the use of Echo State Networks (ESN), an effective form of reservoir computing, for improving microscopic cellular images segmentation. ESN is a sparsely connected recurrent neural network with most of its weights fixed a priori to randomly chosen values. The only trainable weights are those on links connected to the outputs. The process of segmentation is done with two approaches: the basic form with one reservoir and our approach that corresponds to using multiple reservoirs. Obtained results confirm the benefits of the second approach that outperforms all the state-of-the-art methods considered in this paper for the microscopic image segmentation problem.

Keywords Cell microscopic images · Classification · Echo State Networks · Reservoir Computing · Segmentation.

1 Introduction

A central problem in many studies, and often regarded as the cornerstone of image analysis, is image segmentation. Microscopic cellular image segmentation is a very important and challenging task for the medical image processing community. In particular, image analysis in the field of cancer screening is a significant tool for

B. Meftah
LRSBG, Department of Mathematics and Computer Science
University of Mustapha Stambouli
Mascara, Algeria
E-mail: meftahb@yahoo.fr

O. Lézoray
Normandie Université, UNICAEN, ENSICAEN
GREYC UMR 6072
Caen, France

A. Benyettou
SIMPA, Department of Computer Science
University of Mohamed Boudiaf (USTO)
Oran, Algeria

cytopathology [1],[2]. Two principal reasons can be highlighted. First, the quantitative analysis of shape and structure of nuclei coming from microscopic color images brings to the pathologist valuable information for diagnosis assistance. Second, the quantity of information that the pathologist must deal with is large, in particular when the number of cancer screening increases. Consequently, a segmentation scheme for microscopic cellular imaging must be efficient for further reliable analysis. Many cellular segmentation methods have been developed so far (see [1] for a review). They include active contours [3], [4], neural networks [5], spiking neuron networks [6] and watershed [7]. However, efficient segmentation of such images is still an open issue.

In pathology, cells are usually stained with Papanicolaou international staining and this makes it possible to classify the pixels among three classes according to their color: background (white), cytoplasm (green) or nuclei (blue). However, this classification cannot be perfect. Indeed, a fraction of nuclei pixels can have the same color than cytoplasm pixels because of the variability of the nuclei, either according to the type of the cells or to the chromatin distribution. In addition, for some cytology, the mucus present in the background can have the same color than some cells (either cytoplasm and nuclei). Another problem for the design of cellular segmentation schemes is on how to evaluate the segmentation quality. Indeed, almost all segmentation schemes have some parameters. Human observation highlights that the values chosen for these parameters are very significant for the quality of the segmentation. However, for an automatic selection of the optimal parameter values, the quality of segmentation must also be automatically evaluated [8].

On the other hand, reservoir computing (RC) refers to a recent paradigm in computational intelligence and deals with separate training of recurrent neural networks (RNNs) and its readouts. Jaeger introduced Echo State Networks (ESNs), one possible realization of RC, that have an important distinction regarding traditional RNNs: only a linear output layer needs to be adjusted [9]. This simplifies greatly the task of learning. The ESN is usually composed of hundreds or thousands of internal neurons. This promising model reflects some features of biological brain. ESN is a sparsely connected recurrent neural network with most of its weights fixed a priori to randomly chosen values [10]. The only trainable weights are those on links connected to the outputs. ESN has been successfully applied in data clustering [11], image classification [12], spectral image clustering [13], image segmentation [14], cell recognition [15], extraction of informations [16] and object classification [17].

So, the main objective of this research is to classify different types of pixels in cellular images using Echo State Networks. The process is done with two approaches: the basic form with one reservoir (Figure 3) and our approach (Figure 4) that corresponds to using multiple reservoirs. This is the one-versus-rest approach used in many learning machine approaches relying on a decomposition scheme [18]. Similar structure with Self Organizing Maps can be found in [19]. As we experimentally demonstrate, our approach outperforms all the state-of-the-art methods presented in this paper. With artificial neural networks, choosing a right set of network parameters is important for a successful design, and so it is for an ESN. Therefore, the penultimate section of this paper discusses choices on ESN parameters and provides guidance towards their setting.

The structure of the paper is as follows. Section 2 introduces the main aspects of Echo State Networks. Section 3 describes the basic ESN approach and the proposed architecture for cellular image segmentation. Section 4 is devoted to cellular segmentation results, evaluation of the performance of our proposal and comparison with other methods. Section 5 discusses the performance of Echo State Networks. Finally, the main conclusions of this work are presented in Section 6.

2 Echo State Networks

Neurons are recurrently connected in the brain and similar computational models integrating temporal information are needed for solving cognitive tasks. However it is difficult to modify the synapses in such recurrent systems to improve their computational performance. One strategy is to view the recurrent system as a generic computational model and to concentrate learning efforts only on the training of linear readouts. Such a strategy has been developed with Echo State Networks [9]. We detail them in the sequel.

2.1 Network architecture

An Echo State Network can be described as a graph with three sets of nodes, namely input units, internal network units and output units [9]. Figure 1 depicts the overall scheme of an Echo State Networks. We distinguish four sets of weights: w^{in} the weights from the input layer to the reservoir layer, w the reservoir weights, w^{back} the recurrent connections from the output to the reservoir layer, and w^{out} the connections from the input and reservoir layer to the output layer.

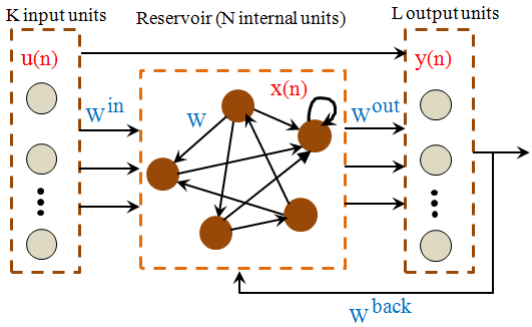


Fig. 1 Representation of an Echo State Network.

The reservoir is the big layer, there can be connections directly from the input to the output layers and there can also be feedback connections from the output to the reservoir and to itself. Inside the reservoir, there are feedback connections since the neurons are randomly connected to each other. These connections are typically sparse. The size N of the reservoir and the sparsity of the connections within the reservoir are meta-parameters that are usually not critical. The computational power of ESN comes with the short-term memory capability of the reservoir.

We can write the state $x(t)$ at time t of a reservoir of N neurons, given an ESN with K inputs and L outputs, as:

$$x(t) = f(w^{in}u(t+1) + wx(t-1) + w^{back}y(t-1)) \tag{1}$$

where $f(.)$ is the reservoir nonlinear activation function, w is the $N \times N$ reservoir weight matrix, w^{in} is the $N \times K$ input weight matrix, w^{back} is the $N \times L$ output feedback weight matrix and $u(t)$ is the input data. The L-dimensional network output is given by:

$$y(t) = g(w^{out}(u(t), x(t), y(t-1))) \tag{2}$$

where $g(.)$ is the output activation function (typically the identity or a sigmoid) and w^{out} is a $L \times (K + N + L)$ output weight matrix.

2.2 ESN Training

To overcome the training difficulties of recurrent neural networks, the concept of reservoir computing (RC) has been proposed. The core idea of RC consists in using a large recurrent neural networks as a "pool" of excitable complex neural dynamics, from which readout neurons can learn to extract the current state of the network. This reduces the complexity of training to simple linear regression while preserving the recurrent property of the network.

Otherwise, the concept consists in teaching only the weights of the outgoing connections from the reservoir to the output (see Figure 1). The internal topology and weights remain unchanged during teaching. All weights except w^{out} are fixed prior to training with random values chosen to ensure that there will be Echo States Properties (ESP). ESP means that the reservoir states asymptotically depend on the input and not the initial state of the reservoir [20]. To account for ESP, the eigenvalues of W should lie inside the unit circle by scaling the reservoir connection weights w as $w \leftarrow \alpha w/\rho_{max}$, where $|\rho_{max}|$ is the spectral radius, which is the largest among the absolute values of the eigenvalues of w and $0 < \alpha < 1$ is a scaling parameter. The spectral radius of the reservoir matrix should be smaller than 1 to ensure that the network possesses the 'Echo State Property' (see [9]; [21] for more details).

When training an ESN, the input vectors are fed into the ESN and its outputs are read only after several iterations, that is when reservoir output becomes stable for a given input vector. The desired output $y_{teach}(t)$ and actual ESN output $y(t)$ are compared, and the errors are used to update only the readout output weights while all other weights in the ESN remain fixed. Different methods can then be used to train the linear readout function, and thus to determine the elements of the $L \times (K+N+L)$ output weight matrix w^{out}. A complete overview and discussion of the different available techniques reported in the literature for training the readout function of the reservoir can be found in [9], [22] and [23].

As demonstrated in [24] one tries to compute the output weights w^{out} such that the mean squared training error MSE_{train} is minimized. The error $e_{train}(t)$ at the current time step is the difference between a target signal $y_{teach}(t)$ and the network output $y(t)$:

$$e_{train}(t) = g^{-1}(y_{teach}(t)) - g^{-1}(y(t)) \tag{3}$$

$$e_{train}(t) = g^{-1}(y_{teach}(t)) - g^{-1}(w^{out}(x(t))) \tag{4}$$

where, the effect of nonlinearity output is undone by g^{-1}. Now the output weights w^{out} are determined such that e_{train} is minimized in a mean square error (MSE) sense.

$$MSE_{train} = \frac{1}{(n_{max} - n_{min})} \sum_{n=n_{min}}^{n_{max}} e^2_{train}(t) \tag{5}$$

where, n_{max} is the number of training examples and n_{min} is an initial washout time. Due to the echo state property, the dynamic reservoir needs some time until the effects of initial transients are washed out and these initial samples should not be considered when calculating the output weights.

In a training algorithm, based on the pseudo-inverse, the following steps calculate optimal output weights w^{out}:

- Dismiss data from initial transients where $n < n_{min}$ and collect the remaining input and network states $(u_{teach}(t), x(t))$ for each time step row-wise into a $(n_{max} - n_{min}) \times (N + K)$ matrix M [N : reservoir size, K : input units];
- Collect target signals $g^-1(y_{teach}(t))$ for each output neuron and time step into a $(n_{max} - n_{min}) \times (L)$ matrix S [L : output units];
- Compute pseudo-inverse M^+ and put

$$w^{out} = (M^+ S)^T \tag{6}$$

  where $(.)^+$ denotes the pseudo-inverse and $(.)^T$ the transposition of a matrix.
- The ESN is now trained and can be used. Once trained, the system can then be used for classification purposes by running an unseen data through the ESN network to generate a classification output.

3 Proposed ESN architecture for Microscopic pixel classification

In this section we present the ESN architecture in its basic form and the one we have developed for the classification of pixels in microscopic images. We start by a presentation of the problem.

3.1 Microscopic pixel classification

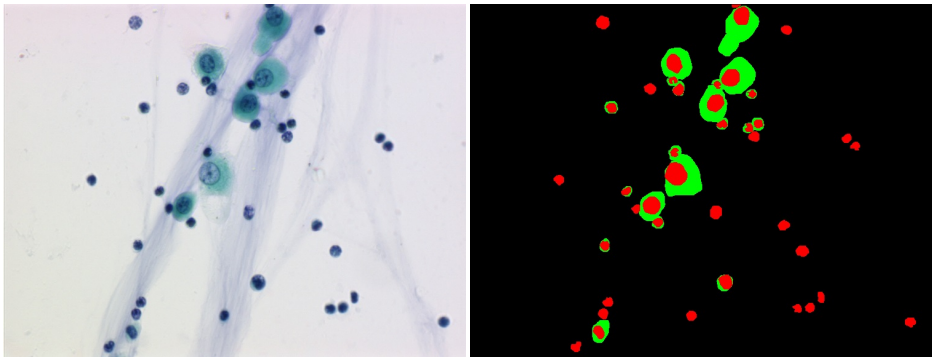
For the considered class of microscopic images, a microscopy expert has to choose representative images that well describe the whole segmentation problem (i.e., a ground truth). This ground truth database can be used for the learning step and also as a reference segmentation to evaluate the relevance of an automatic segmentation. In the sequel, we will consider a publicly available database of microscopic images from bronchial cytology [1][25]. The pixels of these images have to

[1] https://lezoray.users.greyc.fr/researchDatabasesBronchialImages.php

be classified into one of the three following classes: background, cell cytoplasm and cell nuclei. Pixels have a label specifying their classes (2: nucleus, 1: cytoplasm, 0: background). Figure 2(a)-(b) show a microscopic color image and its ground truth. Images dataset has been split to produce training and test sets. All images have same width and height (752×574 pixels).

The problem we want to solve with such a dataset is pixel classification. Using an ESN, we want to classify each pixel of an image into a class among background, cytoplasm and nuclei. Once this classification has been performed, each cell can be isolated and features computed from it so that the cell can then be classified into benign or malignant. This last task is beyond the scope of the paper and we will consider only the task of pixel classification. As detailed in [26], even if image processing automation is one of the oldest fields in cytopathology, the accurate classification of nuclei pixels still poses large difficulties for computational approaches.

In addition, the dataset [25] is the only available public dataset with associated labeling information. Previous works on this dataset [6], [25], [27] and [28] have shown that such a real-world classification problem is very difficult.



(a) (b)

Fig. 2 (a) Microscopic cellular image; (b) Expert segmentation result on the microscopic image: background (black), cytoplasm (green), nuclei (red).

3.2 Basic ESN approach

The basic architecture consists of three layers of neurons, an input layer, a recurrently connected reservoir layer and an output layer (Figure 3). Each neuron in a layer is randomly connected to every neuron of the next layer. The first layer is composed of three inputs units that correspond to normalized values of color pixels of each image in the sequence training. Images are normalized to the $[0; 1]$ range:

$$x_i = \frac{x_i - min(x_i)}{max(x_i) - min(x_i)} \tag{7}$$

Where x_i correspond to RGB values of color pixels and $max(x_i)$, $min(x_i)$ are the maximum and minimum values of x_i, respectively.

In the second layer, we distinguish the reservoir that recognizes three outputs. After completing the supervised learning of the input pixel in correspondence with the ground truth image, it will be assigned to one winning class (background, nuclei or cytoplasm). The third layer corresponds to three outputs units. The maximum response is used to determine the pixel class.

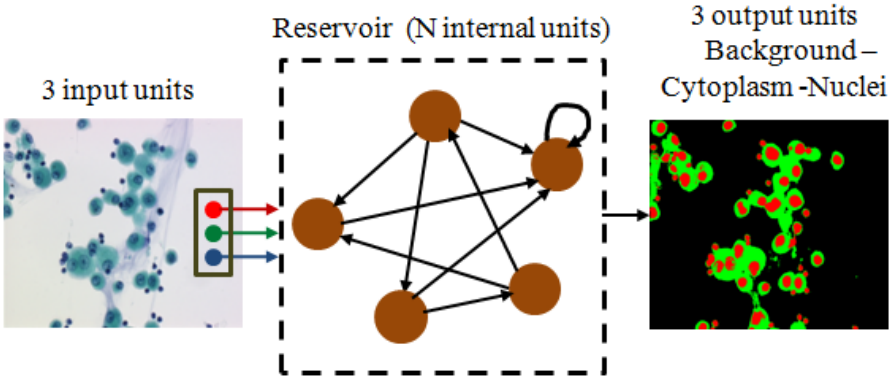


Fig. 3 Network topology with one reservoir for cellular image segmentation.

3.3 Proposed ESN approach

Our proposed architecture consists of four layers of neurons (Figure 4). The first layer corresponds to three inputs units that correspond to normalized values of color pixels of each image in the sequence training. Images are normalized to the $[0; 1]$ using Eq.7.

In the second layer, we distinguish two reservoirs that have the same properties. Each reservoir is independent of the other in its computation. The first allows separating background versus both cytoplasm and nuclei. The second allows separating nuclei versus both background and cytoplasm. After completing the supervised learning of the input pixel in correspondence with the ground truth image for each reservoir, it will be assigned to one winning class (background versus rest and nuclei versus rest) in the third layer. The final output is obtained by superposition of the two previous outputs outcome from the third layer based on the maximum value of the class number.

In both architectures, one should note that there are no direct connections between input and output layers. Similarly, there are no feedback connections from the output layer to previous layers. Only the reservoir layer contains recurrent connections. The input layer is fully connected to the reservoir. In this case, equations 1 and 2 will become respectively:

$$x(t) = f(w^{in}u(t+1) + wx(t-1)) \tag{8}$$

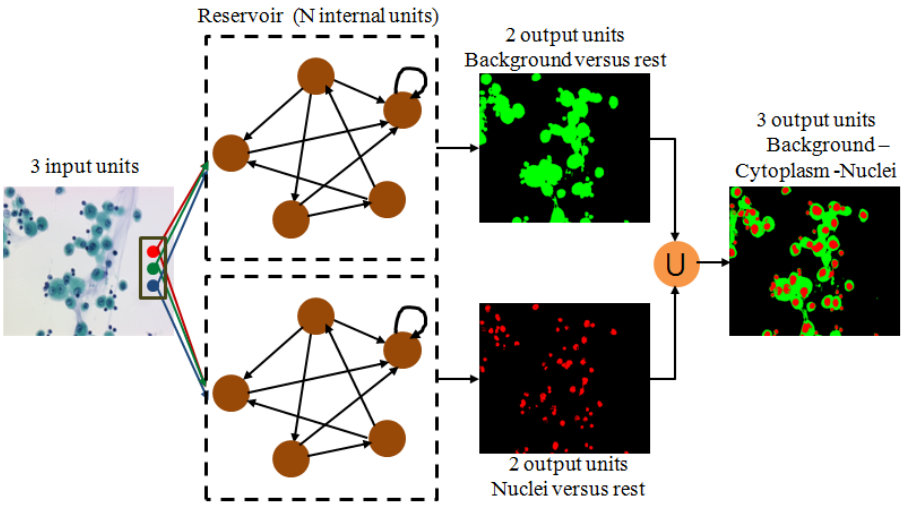


Fig. 4 Network topology with two reservoirs for cellular image segmentation.

$$y(t) = g(w^{out}(x(t))) \tag{9}$$

Our approach towards pixel classification using ESN is summed up in Algorithm 1.

Algorithm 1 Implementation of pixel classification using ESN

Step 1: Build an untrained network that has the Echo State Property and whose internal units exhibit different dynamics

- Randomly generate an internal weight matrix w_0 with a particular degree of connectivity
- Normalize w_0 to a matrix w_1 by putting $w_1 = (1/(|\rho(w_0)|))$ where $|\rho(w_0)|$ is the spectral radius of w_0
- Scale w_1 to $w = \alpha w_1$ where $\alpha < 1$, such that finally w has a spectral radius of α
- Randomly create the input and output weights matrices w^{in} and w^{out}

Step 2: Train the ESN

- Feed the training images into the network and save the reservoir states for each input

Step 3: Compute output weights w^{out}

- After having showed the complete training set to the network, find the output weights using a linear solver (pseudo-inverse approach), given the state of the reservoir and the desired target for each input image

Step 4: Use the ESN

- With the trained output layer the network can be used in test mode

In both the training and the exploitation of the ESN, it is a common procedure to not use the first 100 values of a sequence to wash out the effect of the initial network state [24].

4 Experimental Results

4.1 Classification Results

Our pixel classification belongs to supervised classification techniques. To perform its learning, we generate a training set from three images containing objects with a wide variability. These images have been manually segmented by an expert in cytopathology (Figure 2.b). A testing set was also created from five other representative images.

However, in all of the experiment results, the reservoir size is kept constant at 100 units whose weight matrix has sparse connectivity of 10% and which is scaled to a spectral radius of 0.8 (see next section for the choice of parameters performance). Hyperbolic tangent activation functions are used for the reservoir units: $f(x) = tanh(x)$. The output units act only as summation units: $g(x) = x$. The input, reservoir and readout weights are randomly initialized at the beginning of the training [29]. During the process of training, the weights are updated using the pseudo inverse of different states. The network is stopped once all images are presented for training.

Images in Figure 5 show segmentation results with our segmentation scheme in comparison with the expert segmentation and ESN basic approach. It is worth to note that the mucus present in all images is correctly identified as background.

4.2 Evaluation Methods

The resulting evaluation step is very important. The evaluation method uses a reference manual segmentation provided by an expert and provides a recognition quality index. A common accuracy measures the average percentage of pixels being correctly classified for both foreground and background classes. Thus, several classification rates are used to assess our approach performance and are expressed as follows [30]:

$$R_0 = \frac{Number\ of\ pixels\ well\ classified}{Number\ of\ pixels\ of\ the\ image} \tag{10}$$

$$R_1 = \frac{Number\ of\ nuclei\ pixels\ well\ classified}{Number\ of\ nuclei\ pixels\ of\ the\ image} \tag{11}$$

$$R_2 = \frac{Number\ of\ background\ pixels\ well\ classified}{Number\ of\ background\ pixels\ of\ the\ image} \tag{12}$$

$$R_3 = \frac{R_1 + R_2}{2} \tag{13}$$

Tables 1 and 2 present the recognition accuracy of our approach for each cellular image of the training and test sets. The most important (and difficult) task is

the efficient classification of nuclei pixels since they allow to differentiate between benign and malign cells (Figure 5-fifth row). The tables show the different accuracy rates for the four different measures expressed above. From Table 1, the obtained results by the ESN based on our approach indicate a high accuracy of nuclei recognition (R_1) that lies between 88.27% and 92.45%, with an average accuracy of 89.76%. On the other side from Table 2, the obtained results by the ESN based on our approach indicate a high accuracy of nuclei recognition (R_1) that lies between 75.05% and 91.87%, with an average accuracy of 86.06%.

Table 1 Classification rates of the training set.

Training set	R_0	R_1	R_2	R_3
0024-rgb	95.29%	88.27%	98.91%	93.59%
0042-rgb	95.37%	88.57%	97.35%	92.96%
0055-rgb	97.62%	92.45%	98.24%	95.35%
Average	**96.09%**	**89.76%**	**98.17%**	**93.97%**

Table 2 Classification rates of the test set.

Test set	R_0	R_1	R_2	R_3
0033-rgb	95.34%	91.87%	97.87%	94.87%
0036-rgb	98.22%	91.83%	99.48%	95.65%
0056-rgb	96.96%	75.05%	99.18%	87.11%
0061-rgb	96.85%	79.87%	98.83%	89.35%
0070-rgb	96.29%	91.70%	97.48%	94.59%
Average	**96.73%**	**86.06%**	**98.57%**	**92.32%**

Table 3 presents a comparison of the classification accuracies obtained by Meftah [6], Meurie [25] and Dumont [27] and for different classifiers as well as with ESN method based on the test set. Our approach clearly outperforms all these state-of-the-art methods and exceeds by far the performance of the ESN basic approach.

Table 3 Segmentation rates and comparison with other approaches with best rates bold faced

Classifier	R_1
ESN with one reservoir (basic approach)	67.55%
KNN [25]	70.00%
Fisher 1 [25]	72.30%
MLP [25]	73.00%
SVM [25]	74.20%
K-means [25]	74.40%
Bayes [25]	74.60%
Extra trees [27]	76.90%
Supervised SNN [6]	80.37%
ESN with multiple reservoirs (our approach)	**86.06** %

This very good performance of our approach is illustrated in Figure 5. As it can be seen, background and cytoplasm classification is almost perfect even for some very difficult cases where the pixel color is very close to that of nuclei (because of the mucus in the background). For the nuclei classification, the results are good for all cells except those having a very pale color close to the background or the cytoplasm. This inaccuracy can however be easily corrected by the use of a local region refinement to obtain the final segmentation, as it is done in [30].

5 Echo state performance and discussion

One difficulty in dealing with ESN Networks is that a large number of parameters are involved that can greatly influence performance. Jaeger [31] identified some parameters to be appropriately selected for developing an effective ESN: reservoir size, spectral radius of reservoir weight matrix and the scaling of input. For more details Alexandre et al. [32], Yuanbiao et al. [33], Venayagamoorthy et al. [34], Koryakin et al. [35] and Rodan et al. [36] have also developed some literatures towards the choice of ESN parameters.

Indeed, in order to achieve an appropriate performance for the ESN, there is a need to develop a specific experimental design for each individual image from microscopic cells database cited in section 3.1, and testing different values for each parameter in a wide range. However, in the following are given some guidance on individual parameter settings.

The metric used to check the network performance is Root Mean Square Error (RMSE). RMSE, sometimes called Root Mean Square Deviation (RMSD), is commonly used to quantify the difference between a target signal $y_{teach}(t)$ and the network output $y(t)$ i.e. the current class output (background or cytoplasm or nuclei) (Figure 4.last layer)

$$RMSE = \sqrt{\sum_{i=1}^{n}(y(t) - y_{teach}(t))^2} \tag{14}$$

5.1 Reservoir size

One obviously crucial parameter of the model is the number of units in the reservoir. The topology of the reservoir has an important role in the performance of the segmentation process. In general, it is necessary to increase the size of the reservoir to find a good result. Figure 6 shows the variation of the system performance with different reservoir sizes varying from 5 to 500 neurons. For each image in the dataset and each reservoir size, experiments are repeated and the average test RMSE is calculated.

Generally, the more neurons there are in the reservoir, the better are the results but longer is the simulation time. Nevertheless, the number of neurons has to be previously designed to really correspond to the desired application. Since training and running an ESN is computationally cheap compared to other RNN approaches, reservoir sizes of order 10^4 are not uncommon [37].

5.2 Spectral radius

Another test we performed was to find out the optimal value for the spectral radius α. The diligent choice of the spectral radius α of the reservoir weight matrix is of crucial importance for the eventual success of ESN training. This is because α is intimately connected to the intrinsic timescale of the dynamics of the reservoir state. The spectral radius determines how fast the influence of an input dies out in a reservoir with time, and how stable the reservoir activations are. It is small for the fast teacher dynamics and large for the slow teacher dynamics [9].

Experiments are carried out to verify whether the spectral radius can be extended to a wider range by using the ESN. Reservoir size of 100 neurons is used. For each image in the dataset and each spectral radius, experiments are repeated and the average test RMSE is calculated.

Figure 7 shows the variation of the system performance with different settings of spectral radius. Note that, the range of the spectral radius is set to [0.05; 0.9], in steps of 0.05. We can conclude that the system performed well at $\alpha = 0.8$ and this value is also confirmed by [34].

5.3 Input scaling

The input scaling will determine how far the hidden states are pushed away from the linear part of the activation function by the input, in other words: it will determine the overall nonlinearity of the reservoir.

Experiments are carried out to verify whether the input scaling can be used by the ESN. A reservoir size of 100 neurons is used. For each image in the data set and each input scaling, experiments are repeated and the average test RMSE is calculated.

Figure 8 shows the variation of the system performance with different settings of input scaling. Note that, the range of the input scaling is set to $[0; 10]$. It is concluded that the system performed well for any positive value far away from zero of input scaling. This parameter is not of critical impact for the ESN training once it is larger than 4.

5.4 Complexity evaluation

The complexity is an important element to consider as well, since its estimation is often used to understand the behavior of classifiers. This gives an idea of time and memory requirements. Table 4 presents the complexity of the learning of the classifiers under consideration for performance comparison in the Table 3 Section 4.2. We assume that n denotes the number of training examples, d is the number of dimensions of each example and k is the number of classes. Based on Table 4, it can be seen that the complexity of the proposed approach with a ESN is very close to the least complex approaches such as KNN or Bayes, when the ESN reservoir size is reasonable. This is the case of our proposal since we have chosen in all our experimentations a reservoir of 100 units. This behavior is normal since an ESN roughly performs linear regression and its complexity is due to the cost of inverting the matrix (Step 3 of Algorithm 1).

Table 4 Complexity evaluation

Classifier	Complexity
KNN	$O(knd)$
Fisher 1	$O(9/2n^3)$
MLP	$O(nM^2d^2 + nM^2k^2)$ where M is the number of units
SVM	$O(dn^3)$
K-means	$O(n^{dk+1}log(n))$
Bayes	$O(kdn)$
Extra trees	$O(Mdnlog(n))$ where M is the number of trees
SNN	$O(nM^2d^2 + nM^2k^2)$ where M is the number of units
ESN	$O(M^3)$ where M is the reservoir size

6 Conclusion

In this paper, the Echo State Neural Network (ESN) model is applied to classify different types of pixels in cellular images. The process is done with two approaches: the basic form with one reservoir and our approach that corresponds to use multiple reservoirs. Based on the results presented in the previous section and the remarks listed above, it is concluded that the ESN can be used as an efficient tool for improving the output performance of cellular segmentation. Echo State Network yields highly satisfying results when compared with others methods.

Echo State Neural network is widely recognized and studied, although it has only recently emerged. An essential and difficult to solve for this kind of model question is whether a particular architecture for the reservoir could improve performance. Usually the reservoir is a random structure, optionally subjected to an adaptation mechanism. This study has also shown the importance of the choice of the ESN parameters.

References

1. E. Meijering, Signal Processing Magazine, IEEE **29**(5), 140 (2012). DOI 10.1109/MSP.2012.2204190
2. W.E. Khalbuss, L. Pantanowitz, A.V. Parwani, Pathology Research International **2011**(264683) (2011)
3. P. Wu, J. Yi, G. Zhao, Z. Huang, B. Qiu, D. Gao, IEEE Trans. Biomed. Engineering **62**(1), 284 (2015)
4. Z. Zeng, H. Strange, C. Han, R. Zwiggelaar, in *Image Analysis and Recognition - 10th International Conference, ICIAR 2013, Póvoa do Varzim, Portugal, June 26-28, 2013. Proceedings* (2013), pp. 605–612
5. C.Y.C. M.-C. Su, P.C. Wang, The Scientific World Journal **2014**(796371) (2014)
6. B. Meftah, O. Lézoray, S. Chaturvedi, A. Khurshid, A. Benyettou, in *Artificial Intelligence, Evolutionary Computing and Metaheuristics*, *Studies in Computational Intelligence*, vol. 427, ed. by X.S. Yang (Springer Berlin Heidelberg, 2013), pp. 525–544
7. S. Chourasiya, G. Rani, International Journal of Computer Science and Information Technologies **5**(4), 4834 (2014)
8. S. Takemoto, S. Yoshizawa, Y. Tsujimura, H. Yokota, in *Computing and Networking (CANDAR), 2013 First International Symposium on* (2013), pp. 294–299
9. M. Lukoševičius, H. Jaeger, Computer Science Review **3**(3), 1574 (2009)
10. T. Strauß, W. Wustlich, R. Labahn, Neural Computation **24**(12), 3246 (2012)

11. P. Koprinkova-Hristova, K. Alexiev, in *Artificial Neural Networks and Machine Learning – ICANN 2013*, *Lecture Notes in Computer Science*, vol. 8131 (Springer Berlin Heidelberg, 2013), pp. 343–350
12. A. Woodward, T. Ikegami, in *26th Int. Conf. on Image and Vision Computing* (2011), pp. 543–548
13. P. Koprinkova-Hristova, D. Angelova, D. Borisova, G. Jelev, in *Innovations in Intelligent Systems and Applications (INISTA), 2013 IEEE International Symposium on* (2013), pp. 1–5
14. D. Suganthi, S. Purushothaman, International Journal of Image Processing **2**(1), 1 (2008)
15. P. Kainz, M. Mayrhofer-Reinhartshuber, H. Burgsteiner, M. Asslaber, H. Ahammer, in *48th Annual Conference of the German Society for Biomedical Engineering* (2014)
16. Z. Malik, A. Hussain, J. Wu, Cognitive Computation **6**(3), 595 (2014)
17. A. Mitul, M. Rabin, M. Rakeeb, A.A.M. Khan, G. Rana, A. Mollah, M. Rahman, in *Informatics, Electronics Vision (ICIEV), 2013 International Conference on* (2013), pp. 1–6
18. C. Bishop, *Pattern recognition and machine learning*. Information Science and Statistics (Springer, 2007)
19. Q. Cai, H. He, H. Man, in *The 2011 International Joint Conference on Neural Networks, IJCNN 2011, San Jose, California, USA, July 31 - August 5, 2011* (2011), pp. 2313–2320
20. A. Goudarzi, P. Banda, M. Lakin, C. Teuscher, D. Stefanovic, CoRR **abs/1401.2224** (2014)
21. I. Yildiz, H. Jaeger, S. Kiebel, Neural Networks **35**, 1 (2012)
22. R. Lovlid, Adv. in Artif. Intell. **2013**(2514027), 9:9 (2013)
23. A. Goudarzi, D. Stefanovic, Procedia Computer Science **41**, 176 (2014)
24. H. Jaeger, A tutorial on training recurrent neural networks, covering bppt, rtrl, ekf and the echo state network approach. Tech. Rep. 159, German National Research Center for Information Technology (2002)
25. C. Meurie, O. Lezoray, C. Charrier, A. Elmoataz, International Journal of Robotics and Automation **20**(2), 63 (2005)
26. T. Fuchs, J. Buhmann, Computerized Medical Imaging and Graphics **35**(7–8), 515 (2011)
27. M. Dumont, R. Marée, L. Wehenkel, P. Geurts, in *VISAPP 2009 - Proceedings of the Fourth International Conference on Computer Vision Theory and Applications, Lisboa, Portugal, February 5-8, 2009 - Volume 2* (2009), pp. 196–203
28. Y. Song, W. Cai, D. Feng, in *Digital Image Computing Techniques and Applications (DICTA), 2012 International Conference on* (2012), pp. 1–6
29. G.B. Huang, Cognitive Computation **6**(3), 376 (2014)
30. O. Lezoray, H. Cardot, IEEE Transactions on Image Processing **11**(7), 783 (2002)
31. H. Jaeger, in *Neural Networks, 2005. IJCNN '05. Proceedings. 2005 IEEE International Joint Conference on*, vol. 3 (2005), vol. 3, pp. 1460–1462 vol. 3
32. L. Alexandre, M. Embrechts, in *Artificial Neural Networks – ICANN 2009*, *Lecture Notes in Computer Science*, vol. 5768 (Springer Berlin Heidelberg, 2009), pp. 1015–1024
33. W. Yuanbiao, J. Ni, X. Zhiping, in *Internet Computing for Science and Engineering (ICICSE), 2009 Fourth International Conference on* (2009), pp. 102–108
34. G.K. Venayagamoorthy, B. Shishir, Neural Networks **22**(7), 861 (2009)
35. D. Koryakin, M. Butz, in *Artificial Neural Networks and Machine Learning – ICANN 2012*, *Lecture Notes in Computer Science*, vol. 7552 (Springer Berlin Heidelberg, 2012), pp. 499–506
36. A. Rodan, P. Tino, Neural Networks, IEEE Transactions on **22**(1), 131 (2011)
37. F. Triefenbach, A. Jalalvand, B. Schrauwen, J.P. Martens, in *Advances in Neural Information Processing Systems*, vol. 23 (2010), vol. 23, p. 9

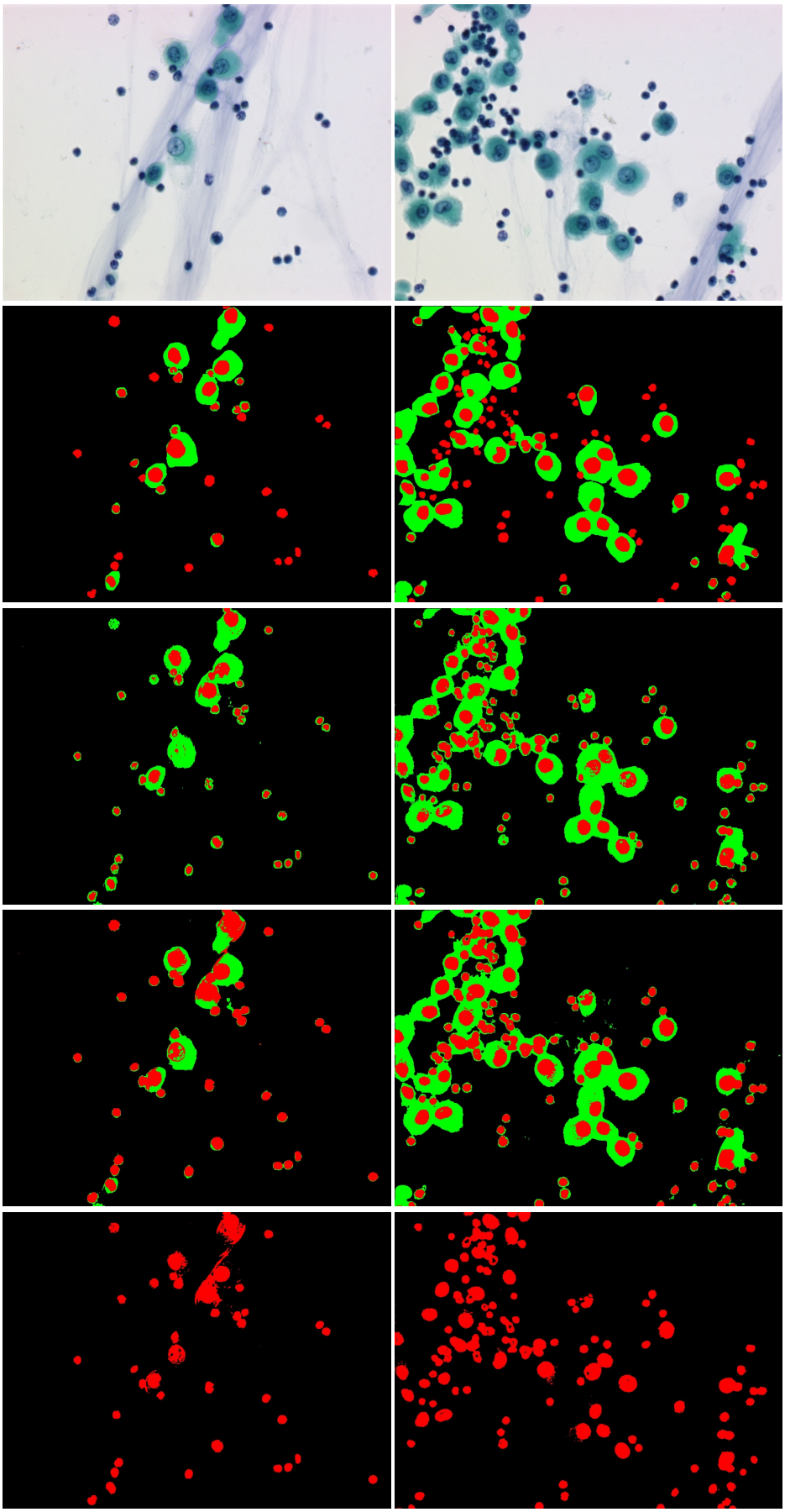


Fig. 5 Cell microscopic images (First row), expert segmentation (Second row), segmentation produced by ESN basic approach (Third row), segmentation produced by our approach (Fourth row) and nuclei extracted by our approach (Fifth row).

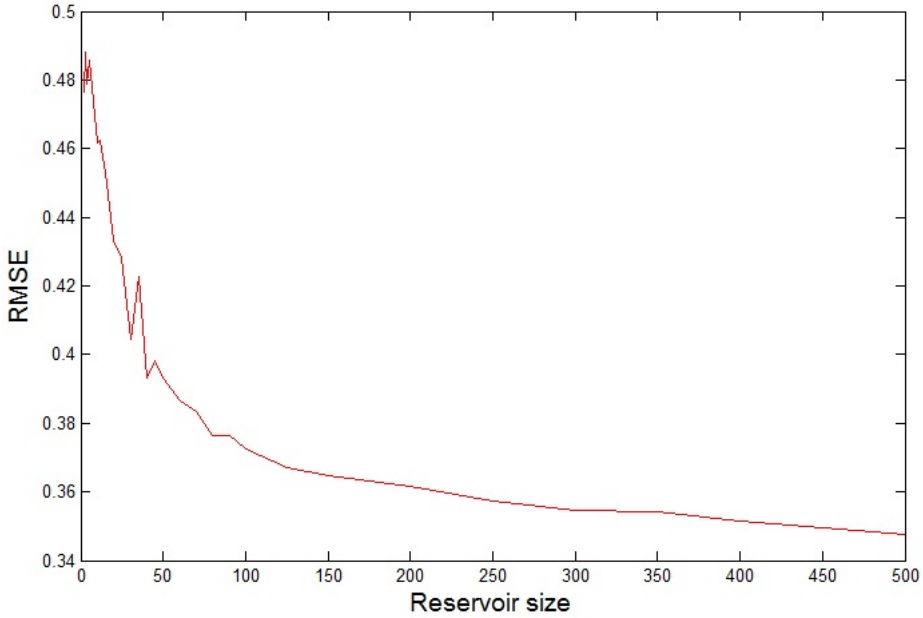


Fig. 6 Echo State Networks performance with variations of reservoir size.

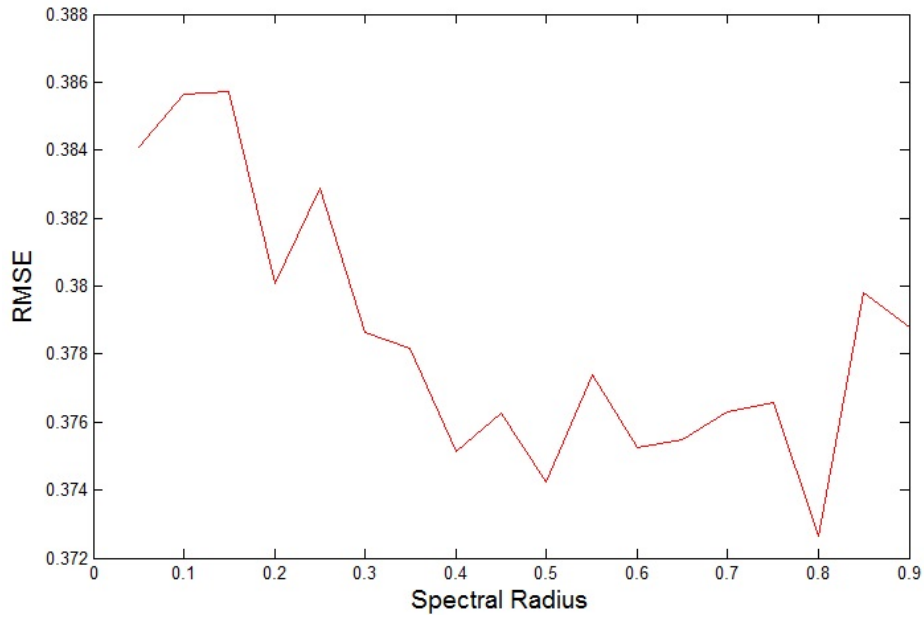


Fig. 7 Echo State Networks performance with variations of spectral radius.

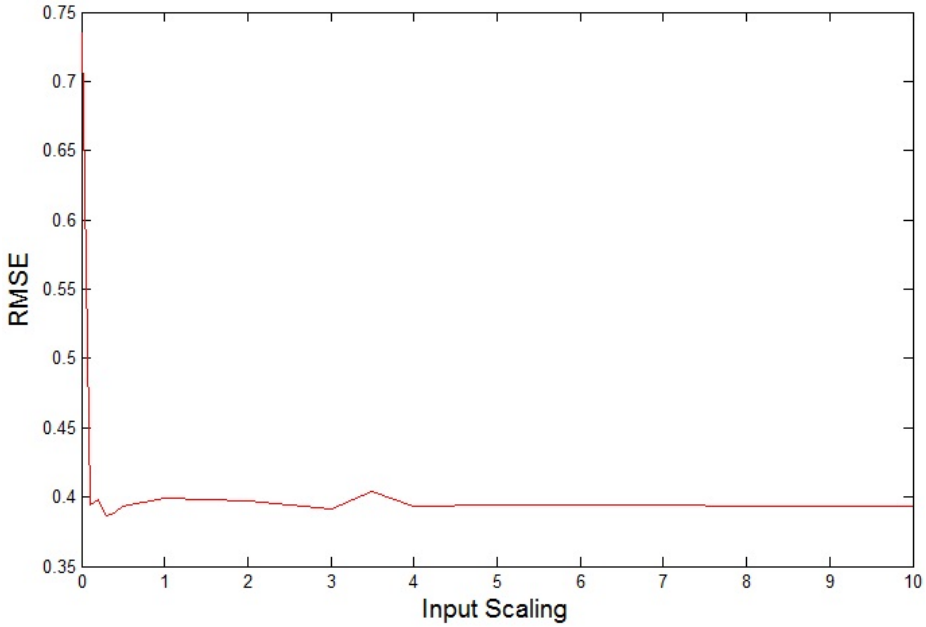


Fig. 8 Echo State Networks performance with variations of input scaling.