A Modified Echo State Network for Time Independent Image Classification

Steven D. Gardner*, Mohammad R. Haider*, Lee Moradi*, and Vladimir Vantsevich*
*School of Engineering, University of Alabama at Birmingham, Birmingham, AL, USA
{stevendg, mrhaider, moradi, vantsevi}@uab.edu

***Abstract*—Image classification is typically performed with highly trained feed-forward machine learning algorithms like deep neural networks and support vector machines. The image can be treated as a time-series input when applied to the network multiple times, opening the way for recurrent neural networks to perform tasks like image classification, semantic segmentation and auto-encoding. With this approach, ultra-fast training, network optimization, and short-term memory effects allows for dynamic, low-volume datasets to be quickly learned without heavy image pre-processing or feature extraction; the main limitation being that input images need labeled output images for training, as is also true of most standard approaches. In this work, the MNIST handwritten digit dataset is used as a benchmark to evaluate metrics of a modified Echo State Network for static image classification. The image array is passed through a noise filter multiple times as the Echo State Network converges to a classification. This highly dynamic approach easily adapts to sequential image (video) tasks like object tracking and is effective with small datasets. Classification rates reach 95.3% with sample size of 10000 handwritten digits and training time of approximately 5 minutes. Progression of this research enables discrete image and time-series classification under a single algorithm, with low computing power and memory requirements.**

***Index Terms*—reservoir computing, Echo State Network, image classification, MATLAB**

I. Introduction

Efficient time-series analysis plays a crucial role in feature extraction, classification, tracking, etc. Reservoir computing networks (RCNs) were created by H. Jaeger [1] and W. Maass [2] in 2001 and 2002, respectively, as a means of processing time-series signals with less training requirements, faster computing, and low memory utilization. Only the output of RCNs are trained, with their architectures randomly initialized and held constant, making them powerful approaches to machine learning. RCNs have since been extensively studied [3], [4] and used for applications such as robot control [5], traffic prediction [6], heartbeat monitoring [7], and recently have been considered for medical imaging semantic segmentation [8], [9] and vehicle perception [10]. However, their potential with time-independent signals is lacking. In [9] and [10] the Echo State Network (ESN) is used to perform pixel-by-pixel binary semantic segmentation on images. Their approach requires extensive feature extraction prior to classification, are limited to binary predictions, and cannot scale to process image sequence streams. The goal of this work is the evaluation of a new and adaptable ESN architecture that uses batch processing of full images for faster training/prediction rates and usage of parallel reservoirs for lower error rates, with the intent of application to time-sensitive domains such as autonomous vehicles and biosignals.

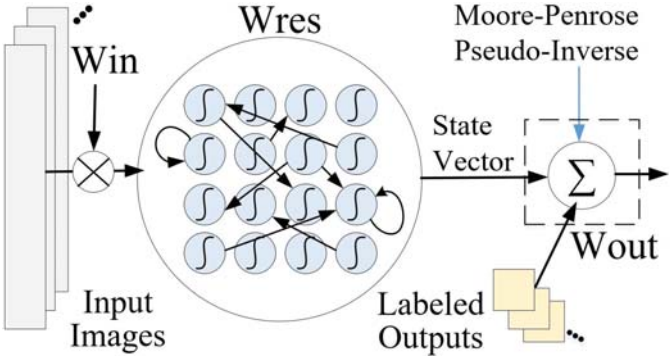

Fig. 1: The schematic of a conventional ESN architecture. A time-series signal is input to the reservoir, causing neuron states to converge upon a pattern read by the trained output weights.

The network can easily be adapted, such as neuron type, connection sparsity, batch or continuous outputs, number of tasks performed by the same dataset, etc. This presents an opportunity for its usage with applications needing sensor fusion, high-dimensional stochastic signal process, low-data volumes, and more. The modified ESN architecture is first explained in Section II, followed by an explanation of the testing conditions and parameters in Section III. The ESN performance is evaluated with common metrics in Section IV and then discussed in Section V. Lastly, a conclusion with future planned research is elaborated in Section VI.

II. Modified ESN Architecture

The basic Echo State Network architecture (shown in Fig. 1) demonstrates an input signal or an image represented as a vector get multiplied by a random input weight matrix and then passed through the reservoir. For ESNs, the reservoir is a recurrent neural network of typically leaky integrator neurons, which acts to transform the linear data into a high-dimensional state space. The neurons take on a value according to the network stimulus and the output is a set of values called the state vector. The desired output classification or annotation is then used with the state vector from the reservoir to generate an output weight vector via Ridge regression (Eq. 1) or Moore-Penrose Pseudo-inverse, which are the most commonly used training algorithms for ESNs. With the output weights

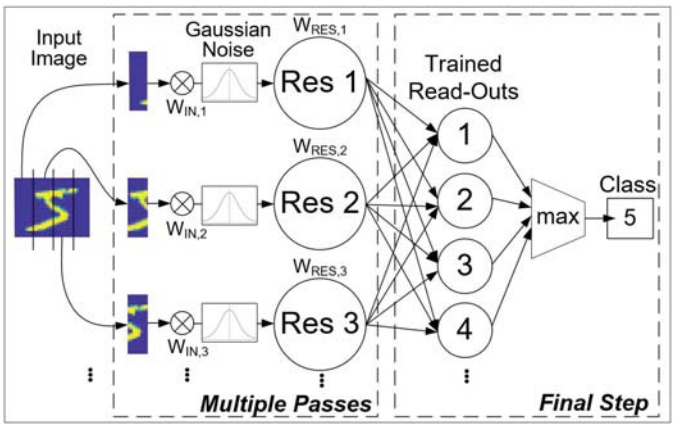

Fig. 2: A schematic of the proposed ESN architecture. The input image is split into equal parts and run through scaling, noise filter, and parallel reservoirs multiple times to allow for neuron convergence. The state vectors are joined and multiplied by the trained weights for each digit. The highest value of the readouts is considered the image's classification.

calculated, the ESN simply needs an input to generate a classification according to Eq. 2.

The concept of using multiple smaller parallel and/or series reservoirs for improved network performance is applied to the modified ESN by having multiple parallel reservoirs that split the input image into equal portions as visualized in Fig. 2, with the total neuron count being number of parallel reservoirs times neurons per reservoir. The neuron states of each parallel reservoir is concatenated into a single state vector, which can be defined as the input image's transformation into a hyperdimensionalized space. The parallel reservoir approach increases neuron-to-input ratio for high volume inputs like high-resolution images without exhibiting exponential training times associated with using a single reservoir. Instead, training times increase linearly with the parallel reservoir approach. The image size of the benchmark tests for this algorithm is small compared to typical images expected from cameras or other high-dimensional sensors.

$$W_{out} = Y_{target} * X^T (X * X^T + \beta * I)^{-1} \quad (1)$$

$$Y = W_{out} * X \quad (2)$$

where: W_{out} = output weights

Y_{target} = desired output classification

X = state vector

β = regularization term

Y = output classification

A. Data Augmentation

A static input image independent of time can be represented as a time-series image for compatibility with the ESN by running the image through a standard gaussian white noise filter multiple times to let the neurons in the reservoir converge. The added noise has been shown in many papers to improve classification results, and is explained well in [3]. By training the algorithm to a noisier signal than the actual one, the features of a noise-free image are more identifiable to the model. Thus, the final pass of the image through the reservoirs is without the added noise and the final updated state vectors of the neurons are multiplied by the trained output weights to generate a classification.

III. ESN Global Parameters and Conditions

The MNIST benchmark dataset of handwritten digits [11] is used to evaluate the performance of this modified ESN. This dataset contains 60,000 handwritten images that have been size-normalized and centered in a fixed-size image of 28x28 pixels. Many feed forward neural networks have been tested with this benchmark, and some papers have examined the ESN with it, such as in [12], [13]. Typically, since the ESN is built for time-series tasks, chaotic and random signals such as NARMA10 are used as the benchmark, although it is not considered in this work.

The modified ESN has many global parameters that define the system, with its performance depending strongly on what the values are initialized at before running the algorithm. As optimization is not within the scope of this work, a set of chosen parameters according to Table I have been used to generate the performance metrics of this work. The number of train/test samples in each epoch is split 80% train and 20% test. The added white Gaussian noise has signal-to-noise ratio of 10. These numbers are based on an understanding of the network dynamics and ability to perform quick evaluations from ultra-fast training times. The low spectral radius (i.e. the maximum absolute eigenvalue of the reservoir's weight matrix) is expected for signals exhibiting low non-linearity like discrete images as explored in [3], [9]. The pixels are already normalized to unity and since the neurons are excited between [-1,1], scaling is expected to not be very low. Otherwise, the neurons would not have the excitation energy to converge properly. In future works, automated optimization will replace manual variable selection for best performance and a more thorough study of the ESN.

TABLE I: List of parameters for the modified ESN.

Parameter	Value
Number of Epochs	4
Train/Test Samples per Epoch	10000
Number of Neurons per Reservoir	400
Number of Parallel Reservoirs	4
Input Scaling	0.01
Spectral Radius	0.0001
Learning Rate	0.01
Number of Reservoir Updates	100
Connectivity of the Reservoirs	10%

IV. Classification Performance Results

The images are all a 28x28 matrix for a total of 784 pixels ranging from [0,1], and split into 4 equal parts makes each reservoir process 196 pixels. Each reservoir holds 400

neurons, for a input-to-neuron ratio of approximately 1:2. It is well-established that sparsely connected reservoirs reduces memory and computing power without any significant change to the prediction rates, thus making 10% connection sparsity effective at increasing training speeds. The effect of the noise filter can be visualized in Fig. 3. As previously mentioned, the noisy image is first generated and then passed through the reservoir to allow convergence, followed by the original image in the final time step. Too high of an SNR value diminishes the performance, as the number becomes indistinguishable from the noise. Therefore, an SNR value of 10 was used as the noise is not high enough to cause poor performance, yet injects enough uncertainty to make the ESN more robust against each image's natural noise. Image tilting was not considered, but is expected to improve results.

Training and image processing time are particularly important when speed is the most critical factor. For 10,000 train/test digits and the values in Table I, the entire training time is 5.6 minutes. Comparatively, the train/test time would have taken 3 times longer if 1 reservoir with 1,600 neurons was used instead, with no appreciable effect on error rate. The contour plot in Fig. 4 shows the overall error rate and training time for different numbers of parallel reservoirs to visualize those metrics. In each case, the total number of neurons remains the same at 1,600 so that the work load is conserved between parameter changes. Thus, the error rates improve when higher values of neurons are used per reservoir and training time generally decreases when adding parallel reservoirs. An important note to make here is that the error rates do not significantly change between these settings in this test, so optimizing the network at 4 parallel reservoirs with 400 neurons or 2 reservoirs with 800 neurons makes little difference.

Given that the numerous global parameters of this ESN were manually tuned in this work, classification rates of the modified ESN for time independent inputs are lower than what optimization process will achieve. Regardless, competitive prediction rates can be visualized by the confusion matrix of Fig. 5. Along the x-axis, the digits 0,1,...,9 are shown as a

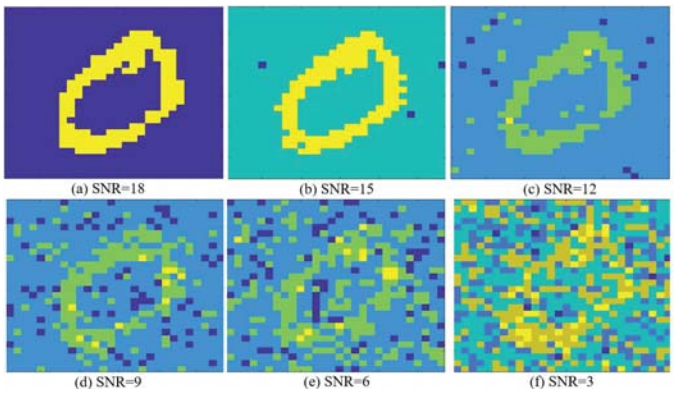

Fig. 3: Signal-to-noise ratio visualization. An SNR of 10 was used in the experiments.

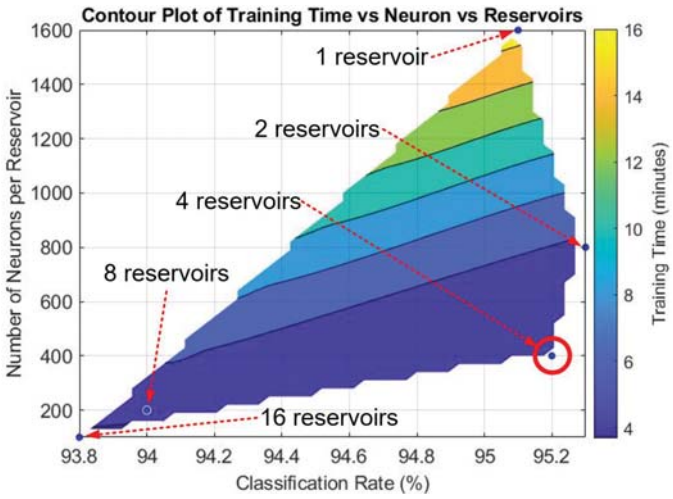

Fig. 4: The contour plot of classification accuracy with training time and number of neurons per reservoir. The plot shows that there is a significant rise in training time when fewer reservoirs are used to process the same workload. The data point circled in red is the test that used the values in Table I.

set of 10 classes. Correct classifications are shown along the diagonal in the green boxes and incorrect classes in the red boxes. The bottom right corner shows the overall classification rate of 95.3% and along the x-axis the performance for each digit is written as percentages (green text for correct classification, red text for the error rate). The digit 3 was correctly classified the least at 91.8% and 0 was recognized by the ESN more than any other digit at 99.1%.

V. Discussion and Future Work

The performance of the modified ESN is highly dependent on spectral radius, input scaling, and number of parallel reservoirs, but changing any of the variables affects the system enough to make its robustness an area in need of improvement. This obviates the need for hyperparameter optimization, which is a part of future planned research. The trade-off of error rates improving with higher numbers of neurons per reservoir and the training time generally decreasing with the addition of parallel reservoirs may dictate how the ESN is modeled, depending on the task. For instance, if the classification rate is more important than the training speed, the system can be optimized to that behavior whereas the opposite can be done to speed up the training time at a slight cost in prediction accuracy. Reaching optimized performance may include optimization for time independent variables first for fast training times, followed by the more time-consuming parameters.

With the modified ESN, classification rates reached 95.3% with manually chosen parameters, minimal pre-processing and short training times. Those metrics are an improvement to the standard ESN MNIST classification results of [12], who achieve 90.5% with 1,200 neurons, although they also achieve above 98% classification by altering their output layer in various ways. In [12], other recurrent neural networks are

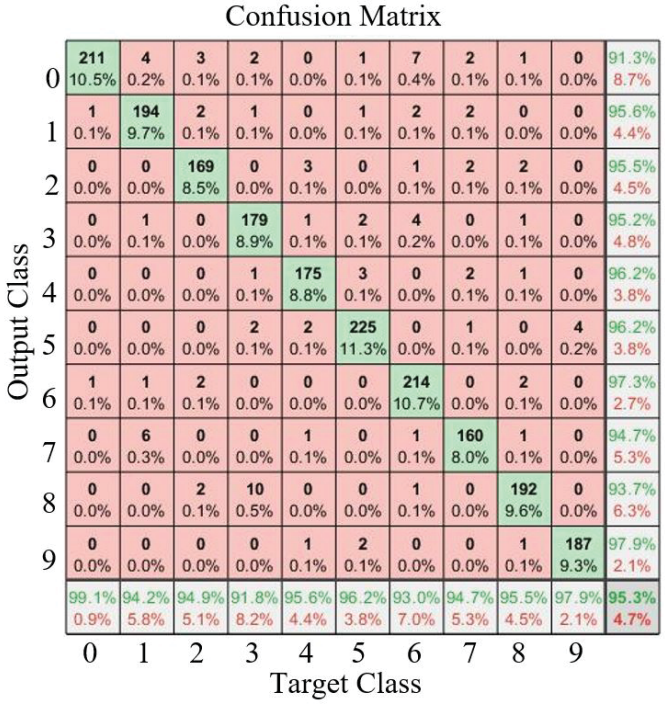

Fig. 5: Confusion matrix of the modified ESN-based classification model. The matrix shows the desired output digit along the x-axis and the ESN output classification along the y-axis. The diagonal (green) shows the correctly classified digits, with overall error rate in the bottom right corner. The error rates of each digit can be seen along the x-axis.

considered and show competitive classification rates of ESNs with MNIST benchmark starting at 93%, and some standard methods using convolutional neural networks work lower performance metrics. Highly-trained standard feed-forward neural networks have reached over 99%, but come with restrictions as to training time and computing power, mainly since all weights throughout the network must be updated during backpropagation.

There are many components of the ESN in this work that can be modified for better performance. Data augmentation could include translating and rotating images instead of just injecting noise. For more complex datasets, feature extraction may be performed prior to running the ESN for more enriched data. Series (i.e. deep) reservoirs can be implemented to images that are time independent in a more meaningful manner since memory capacity has less impact, and is affected by number of hidden reservoir layers. The read-out layer could be replaced with a more robust feed-forward network like the multilayer perceptron (MLP).

VI. Conclusion

This work shows that distributed processing via parallel reservoirs and injection of noise can prove competitive classification rates. The MNIST dataset is used to evaluate the performance of a modified Echo State Network that processes each handwritten digit as a batch that passes through a noise filter and parallel reservoir multiple times for convergence. The ability for this algorithm to scale to both time-series and time independent tasks makes it useful for problems typically approached with feed-forward machine learning networks. The powerful approach of ESNs is gathering momentum as more researchers consider it as a highly competitive method of machine learning using recurrent neural networks.

VII. Acknowledgment

This research was partially supported by Army Research Center Phase V – Cooperative Agreement No.1.A83 (Federal Award No. W56HZV-19-2-0001), and National Science Foundation (Award Nos. ECCS-1813949 and CNS-1645863). However, any opinions, findings, conclusions, or recommendations expressed herein are those of the authors and do not necessarily reflect the views of the funding agencies.

References

[1] H. Jaeger, "A tutorial on training recurrent neural networks, covering bppt, rtrl, ekf and the "echo state network" approach," *Fraunhofer Institute for Autonomous Intelligent Systems (AIS)*, vol. GMD Report 159, 2002.

[2] W. Maass, T. Natschlager, and H. Markram, "Real-time computing without stable states: A new framework for neural based on perturbations," *Neural Computation*, vol. 14, p. 2531–2560, 2002.

[3] M. Lukosevicius, *A Practical Guide to Applying Echo State Networks*. Springer, 2012, vol. 7700, p. 659–686.

[4] Q. Wu, E. Fokoue, and D. Kudithipudi, "On the statistical challenges of echo state networks and some potential remedies," *Rochester Institute of Technology*, 2018.

[5] E. Antonelo, B. Schrauwen, and D. Stroobandt, "Mobile robot control in the road sign problem using reservoir computing networks," *2008 IEEE International Conference on Robotics and Automation*, pp. 911–916, 2008.

[6] P. Yu, W. Jian-min, and P. Xi-yuan, "Traffic prediction with reservoir computing for mobile networks," *2009 Fifth International Conference on Natural Computation*, vol. 2, pp. 464–468, 2009.

[7] M. A. Escalona-Morán, M. C. Soriano, I. Fischer, and C. R. Mirasso, "Electrocardiogram classification using reservoir computing with logistic regression," *IEEE Journal of Biomedical and Health Informatics*, vol. 19, no. 3, pp. 892–898, 2015.

[8] B. Meftah, O. Lézoray, and A. Benyettou, "Novel approach using echo state networks for microscopic cellular image segmentation," *Cognitive Computation*, vol. 8, no. 2, pp. 237–245, 2015.

[9] A. Souahlia, A. Belatreche, A. Benyettou, Z. Ahmed-Foitih, E. Benkhelifa, and K. Curran, "Echo state network-based feature extraction for efficient color image segmentation," *Concurrency and Computation: Practice and Experience*, vol. 32, no. 21, 2020.

[10] S. Roychowdhury and L. S. Muppirisetty, "Fast proposals for image and video annotation using modified echo state networks," *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 1225–1230, 2018.

[11] Y. LeCun, C. Cortes, and C. J. Burges, "The mnist database of handwritten digits," *The Courant Institute of Mathematical Sciences, NYU Google Labs, New York Microsoft Research, Redmond*, 2012.

[12] L. Manneschi, M. O. A. Ellis, G. Gigante, A. C. Lin, P. Del Giudice, and E. Vasilaki, "Exploiting multiple timescales in hierarchical echo state networks," *Frontiers in Applied Mathematics and Statistics*, vol. 6, no. 76, 2021. [Online]. Available: https://www.frontiersin.org/article/10.3389/fams.2020.616658

[13] N. Schaetti, M. Salomon, and R. Couturier, "Echo state networks-based reservoir computing for mnist handwritten digits recognition," *International Conference on Computational Science and Engineering*, 2016.