2021 NDIA GROUND VEHICLE SYSTEMS ENGINEERING AND TECHNOLOGY SYMPOSIUM
Autonomy, Artificial Intelligence & Robotics (AAIR) TECHNICAL SESSION
AUGUST 10-12, 2021 - NOVI, MICHIGAN

RAPID HIGH-DIMENSIONAL SEMANTIC SEGMENTATION WITH ECHO STATE NETWORKS

S. Gardner[1], M.R. Haider[1], J. Smereka[2], P. Jayakumar[2], K. Kulkarni[2], D. Gorsich[2], L. Moradi[1], and V. Vantsevich[1]
[1]Electrical and Computer Engineering, University of Alabama Birmingham, Birmingham, AL
[2]Ground Vehicle Systems Center (GVSC), Warren, MI

ABSTRACT

Recurrent Neural Networks have largely been explored for low-dimensional time-series tasks due to their fading memory properties, which is not needed for feed-forward methods like the Convolutional Neural Network. However, benefits of using a recurrent-based neural network (i.e. reservoir computing) for time-independent inputs includes faster training times, lower training requirements, and reduced computational burdens, along with competitive performances to standard machine learning methods. This is especially important for high-dimensional signals like complex images. In this report, a modified Echo State Network (ESN) is introduced and evaluated for its ability to perform semantic segmentation. The parallel ESN containing 16 parallel reservoirs has an image processing time of 2 seconds with an 88% classification rate of 3 classes, with no prior feature extraction or normalization, and a training time of under 2 minutes.

Citation: S. Gardner, M. R. Haider, J. Smereka, P. Jayakumar, K. Kulkarni, D. Gorsich, L. Moradi, and V. Vantsevich, "Rapid High-dimensional Semantic Segmentation With Echo State Networks", In Proceedings of the Ground Vehicle Systems Engineering and Technology Symposium (GVSETS), NDIA, Novi, MI, Aug. 10-12, 2021.

1. INTRODUCTION

The Echo State Network (ESN) has been a popular approach to time-series signal supervised learning since 2008 [1-3], but not many have explored it for image sequence processing [4-8]. A challenge is that camera frames are high-dimensional signals that have variable dynamics according to the sampling frequency of the camera, how quickly regions of interest change between frames, and overall image complexity. ESNs have recently been used for medical image semantic segmentation (SS) [6], agriculture aerial-view SS [7], and ESN binary SS for self-driving vehicles [8], with the last processing each image in 1 second, with extensive preprocessing not included in that metric. Each publication has processing times that are limited due to the standard reservoir architecture approach and computationally-expensive feature extraction preprocessing. This report explores a more scalable option that reduces the need for extensive image preprocessing or long training times by taking advantage of the ESN architecture.

Applying one reservoir to process a 1-D signal is the most common approach when using ESNs, but is not viable for high-dimensional signals like

images due to high neuron requirement, which exponentially increases processing times (O^2). Thus, the common approach is to generate a small set of feature values per pixel that are fed through the ESN to generate the desired pixel. The output image is generated 1 pixel at a time, and substantial preprocessing is required before the ESN operates. Instead, numerous parallel reservoirs may be used to process large sections of the image, with a large output matrix being trained to the concatenation of all the parallel reservoir state vectors. Such a concept has been considered in literature [9], but not heavily for complex image processing. This approach avoids the problem of exponential training/processing time of a single central reservoir and distributes the computational load among many smaller reservoirs, allowing for substantially greater neuron-to-pixel ratios.

This report explores the metrics of the modified ESN using an online benchmarked dataset that uses complex off-road camera images. Section 2 reviews the mathematics of the ESN and explores the parallel ESN architecture. Section 3 covers the benchmark dataset and how it is used in the experiments. Section 4 shows the results of the tests. Section 5 contains a discussion, followed by a conclusion in Section 6.

2. The Parallel ESN

The basic Echo State Network architecture (shown in Figure (1)) demonstrates an input signal or image represented as a vector get multiplied by a random input weight matrix and then passed through the reservoir. For the ESN, 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, as defined by Equation (1). In that equation, X is the input data, W^{in}, W^{res}, and W^{OUT} are the input line weight vectors, reservoir weight vectors, and output line weight vectors, respectively, s is the state vector of the reservoir, and α is the learning rate. The desired output

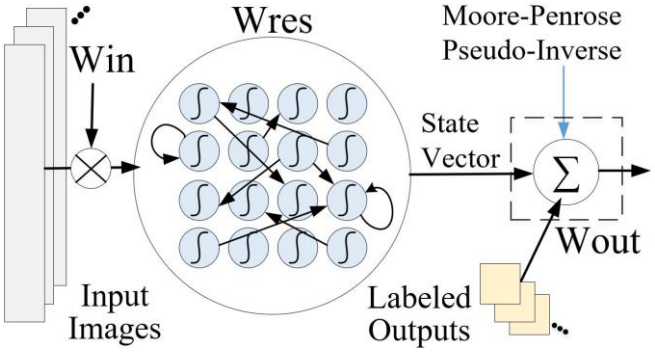

Figure 1. Basic ESN architecture.

classification or annotation is then used with the state vector from the reservoir to generate an output weight vector via Ridge regression in Equation (2) or Moore-Penrose Pseudo-inverse, which are the most commonly used training algorithms for ESNs. There, β is a regularization term to prevent overfitting, I is the identity matrix, and Y^{TARGET} is the output. With the output weights calculated, the ESN simply needs an input to generate a classification according to Equation (3).

$$s = (1-\alpha)s + \alpha\tanh(W^{res}s + W^{in}X) \quad (1)$$

$$W^{OUT} = Y^{TARGET}X^T(XX^T + \beta I)^{-1} \quad (2)$$

$$Y^{TARGET} = W^{OUT}X \quad (3)$$

The use of multiple smaller parallel and/or series reservoirs has been shown to have more improved network performance than a single large reservoir [9-10]. This concept is applied to the modified ESN by having multiple parallel reservoirs that split the input image into equal portions as visualized in Figure (2), with the total neuron count being the number of parallel reservoirs multiplied by the 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-

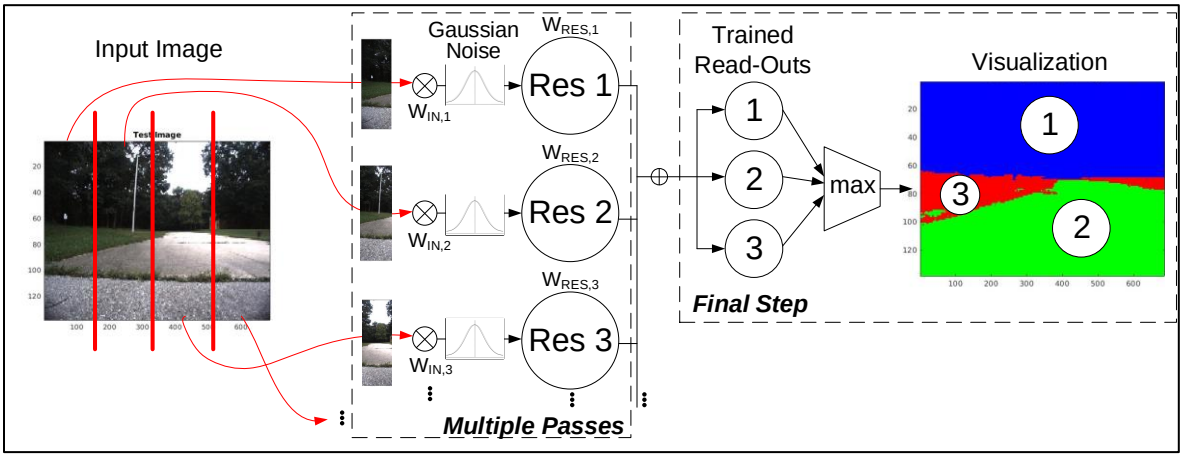

Figure 2. Modified ESN showing the parallel distribution of the image. Multiple passed are done through the reservoirs, and the final time step is then multiplied by the trained output weights to generate the segmented output.

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 bounds of these trends when given significantly large image sizes is yet to be explored. The image size of the benchmark tests for this algorithm is small compared to typical images expected from high-resolution cameras and other high-dimensional sensor datasets like point-clouds from Lidar and stereoscopic cameras. Thus, training times may become unwieldy for inputs of significant sizes.

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 and reach a classification. The added noise has been shown in many papers to improve classification results and is explained well in [12]. 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.

3. Testing Dataset and ESN Parameters

The tests of this report use a benchmark dataset that is then formatted to work with the ESN. The parameters of the ESN are essential for competitive performances. The setup of these components for the tests in this report is described in the remainder of this section.

3.1. Benchmark Dataset

The Robot Unstructured Ground Driving (RUGD) dataset was used for these tests [11]. The dataset contains color image sequences of complex off-road terrains such as trails, parks, and fields during ideal sunny conditions. They were taken using a camera mounted on a small mobile robot platform. The dataset has annotated images with 24 different classes. This report only uses three classes. Thus, the annotated images were reclassified according to three classes: (1) desired pathway, (2) drivable but off-road terrain, (3) non-drivable terrains. This report is among the first to use the ESN with more than two output classifications.

3.2. ESN Testing Parameters

The modified ESN has many global parameters that define the system, with its performance

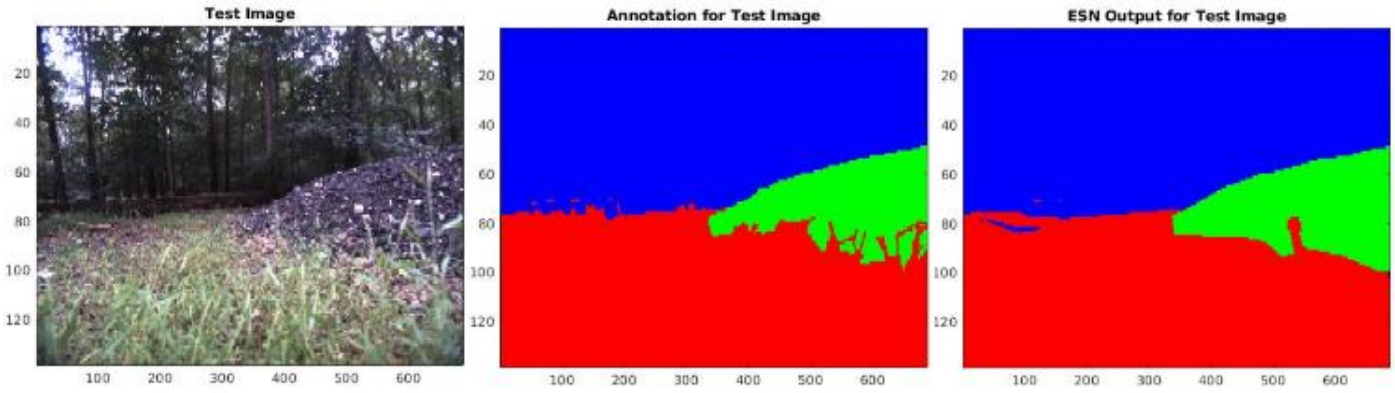

Figure 3. Visualization of an input test image, the annotation, and the output of the ESN. Note the similarities between the annotation and the ESN output.

depending strongly on what the values are initialized at before running the algorithm. As full network optimization is not within the scope of this report, a set of chosen parameters according to Table (1) have been used to generate the performance metrics of this report.

Table 1. ESN Parameters

ESN Parameters	Value
Train/Test Samples	60
Neurons	100
Parallel Reservoirs	16
Input Scaling	0.0002
Spectral Radius	0.0001
Learning Rate	0.04
Time Steps	60
Res. Connectivity	10%

The number of train/test samples in each epoch is split 80% train and 20% test from a randomly selected 60 images. The added white Gaussian noise has signal-to-noise ratio of 10 and the reservoirs will have 60 time steps to converge upon a classification. These numbers are based on an understanding of the network dynamics and ability to perform quick evaluations from ultra-fast training times. A low spectral radius, input scaling factor, and learning rates are expected for signals exhibiting highly non-linearly separable data like discrete images, as explored in [12-13].

4. Modified ESN Metrics and Analysis

With the global parameters defined, input signals pre-processed, and training complete, the ESN performances can be explored. As seen from the visualizations of Figure (3), the unregulated RUGD data is mapped to a corresponding three-class semantic segmented image that highly correlates with the actual annotated image. Pixel error is calculated by subtracting the output and training image and then dividing by the total number of pixels. The performance error for the parameters defined in Table (1) is 88%, with a training time of only 2 minutes and individual image processing time of 2 seconds. Note that the ESN output has less arbitrary detail that was perceived with the manually labeled training data. However, the ESN output recognized a non-drivable region of pixels where a log is located in the background, which the manually labeled annotation does not pick up on. Inaccuracies of the manually labeled images are mostly in the fine details, such as from grass, close-up objects, highly pixelated or obscure objects, etc. Therefore, when the ESN detects objects that were not manually labeled, it implies that the real error

rate of the ESN is lower by a small, insignificant amount, as it slightly out-classified the labels for certain pixels.

4.1. Architecture Optimization

While network optimization is outside the scope of this report, a grid search was performed on the number of parallel reservoirs and number of neurons per reservoir to understand how improved the classification is from higher pixel-to-neuron ratios. Figure (4) shows the results of sweeping.

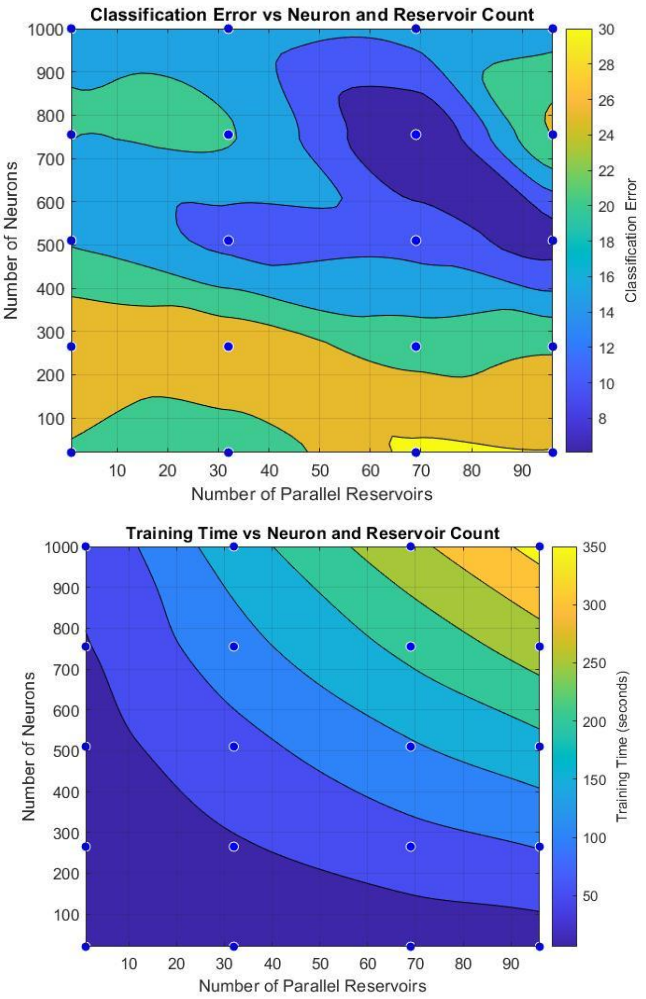

Figure 4. Grid Search Results: (top) the classification error is observed for the optimization sweep. (Bottom) The training error is visualized for the same sweeping conditions.

the number of parallel reservoirs from 1 to 96 and the number of neurons per reservoir from 20 to 1000, making the total neuron count range from 20 to 96,000. With a total of 94,944 pixels, the pixel-to-neuron ratios range from approximately 1:1 to 4747:1. Performances generally improved for higher numbers of total neurons.

The top plot of Figure (4) shows error rates reaching a minimum of 6% for 69 parallel reservoirs and 760 neurons per reservoir, validating the hypothesis that architecture optimization improves error rates for the given image size. The bottom plot of Figure (4) is the same sweep conditions but focuses on total training time. For higher neuron counts, the training time increases. The extent of these trends are promising but require further investigation to understand how higher resolution images and multi-sensor inputs affect the ESN architecture, error rates, and training time. Dissimilar datasets from the one used in this work will significantly alter the number of parallel reservoirs and number of neurons per reservoir that result in minimum error rates. Thus, when new data is introduced, the modified ESN can be quickly re-trained with a grid search to find the new optimal architecture. The duration/stability of this optimal point is proportional to the consistency of the environment. For wildly changing scenes such as single-cell thunderstorms or dust storms, the modified ESN performance is expected to dramatically decrease.

5. Discussion of Results and Future Work

The processing time per image is 2 seconds but needs to be under 60 milliseconds for usage on moving vehicles, since the response time is critical for reacting to tire-soil mobility dynamics.

The optimization of the architecture is a unique concept since generally the network is established and unchanged during all training. Since training times are so short, the grid search can be performed significantly faster than standard convolutional neural network. In most applications where Convolutional Neural Networks (CNN) are used,

they are trained prior to field operation, making training time an insignificant variable. However, in highly dynamic settings where the input data has not been pre-trained, the CNN is unable to adapt to the new data in a reasonable time. The resnet18 deep neural network was trained to this dataset and performed at 96% but had training times that exceeded 2 hours. Comparatively, the ESN of this report only needed 2 minutes and achieved a maximum of 94% classification. There are clear trade-offs to using a CNN over the ESN, and neither particularly outperforms the other, but for this application the ESN is a significantly more effective algorithm that can be scaled to contain a more robust ontology and adapt to new settings rapidly. Furthermore, the ESN can be adapted to different modalities that either perform faster training at the expense of error rates or vice versa. The biggest flaw of the ESN in this work is relatively high-performance instability when the finely tuned global parameters are changed, making optimization crucial to maintaining competitive error rates.

In future works, this ESN approach will be explored for automatic feature extraction via graph neural network concepts, faster image processing speeds, automated hyper-parameter optimization, and usage as a recurrent auto-encoder.

6. Conclusion

This work evaluates hyper-parameter effects on processing time and pixel error of the RUGD dataset using the parallel ESN architecture. Test results show decreasing average pixel error when increasing the number of parallel reservoirs and reservoir size. More investigation into the effects of reservoir size and the number of parallel reservoirs is included. Training takes a few minutes instead of hours, with as few as 60 training/testing samples, making it a promising approach to terrain mapping with unmanned autonomous vehicles.

7. Acknowledgements

This work is supported by Autonomous Vehicle Mobility Institute (AVMI) and University of Alabama at Birmingham (UAB), with special thanks to Ground Vehicle Support Center (GVSC) for overseeing this research.

REFERENCES

[1] E. Antonelo, B. Schrauwen, and D. Stroobandt, "Mobile robot control in the road sign problem using Reservoir Computing networks," in 2008 IEEE International Conference on Robotics and Automation, 2008, pp. 911-916.

[2] P. Yu, W. Jian-min, and P. Xi-yuan, "Traffic Prediction with Reservoir Computing for Mobile Networks," in 2009 Fifth International Conference on Natural Computation, 2009, vol. 2, pp. 464-468.

[3] F. Triefenbach, A. Jalalvand, and B. Schrauwen, "Phoneme Recognition with Large Hierarchical Reservoirs," in Advances in neural information processing systems (NIPS 2010), vol. 23Cambridge: MIT Press, 2010, pp. 2307-2315.

[4] 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.

[5] A. Souahlia, A. Belatreche, A. Benyettou, and K. Curran, "An experimental evaluation of echo state network for colour image segmentation," 2016.

[6] 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.

[7] P. Koprinkova-Hristova, D. Angelova, D. Borisova, and G. Jelev, "Clustering of Spectral Images using Echo State Networks," 2013.

[8] S. Roychowdhury and L. S. Muppirisetty, "Fast Proposals for Image and Video Annotation Using Modified Echo State Networks," presented at the 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), 2018.

[9] X. Liu, M. Chen, C. Yin, and W. Saad, "Analysis of Memory Capacity for Deep Echo State Networks," presented at the 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), 2018.

[10] 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," (in English), Frontiers in Applied Mathematics and Statistics, Original Research vol. 6, no. 76, 2021-February-17 2021.

[11] M. Wigness, S. Eum, J. G. Rogers, D. Han, and H. Kwon, A RUGD Dataset for Autonomous Navigation and Visual Perception in Unstructured Outdoor Environments (International Conference on Intelligent Robots and Systems (IROS)). 2019.

[12] M. Lukosevicius, "A Practical Guide to Applying Echo State Networks," in Neural Networks: Tricks of the Trade, Reloaded, vol. 7700, G. Montavon, G. B. Orr, and K. R. Muller, Eds.: Springer, 2012, pp. 659–686.

[13] 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.