A Reservoir Computing approach to Image Classification using Coupled Echo State and Back-Propagation Neural Networks

Alexander Woodward, Takashi Ikegami
University of Tokyo, 3-8-1 Komaba, Tokyo, 153-8902, Japan
Email: alex@sacral.c.u-tokyo.ac.jp

***Abstract*—This paper describes a Reservoir Computing approach for computer vision image and video processing. A reservoir of complexity is accessed in order to project data into a higher dimensional space. In doing so, higher order data correlations are accounted for and the dynamics of the reservoir can be read out and used for classification purposes. The reservoir takes the form of an Echo State Network (ESN), a recurrent neural network with random weights and connections, possessing strong spatio-temporal properties. This research uses face recognition as an example application - face images are coupled with the ESN and its dynamics are sampled to create a set of response vectors. These are then fed into a feed-forward neural network for training using back-propagation. Results show that the proposed system is capable of classifying face images and that using an ESN increases the robustness of a neural network based classifier. The proposed system is designed to naturally extend to processing video sequences where the full power of the ESN's spatio-temporal dynamics can be exploited.**

I. Introduction

The aim of this research is to investigate a Reservoir Computing approach for computer vision image and video processing. As an example application this work focuses on its application to face recognition. The idea is to access a reservoir of complexity in order to project human face image data into a higher dimensional space. In doing so, higher order spatial correlations are accounted for and the dynamics of the reservoir can be read out and used for classification purposes. This is achieved through feeding input images to the reservoir, sampling its dynamics and then feeding this into a second feed-forward neural network that is trained on the data using back-propagation.

The Reservoir Computing paradigm acts as an umbrella term for the similarly posed Echo State Network (ESN), Liquid State Machine (LSM) and Backpropagation Decorrelation and Temporal Recurrent Networks [1]. This work uses an ESN as a reservoir of complexity; this is a discrete time recurrent neural network (RNN) with the specific properties that it has random connections and random weights up to a scale factor. Additionally, the existence of directed cycles creates feedback loops within the network, generating temporal dynamics and an internal memory.

There exist a wide number of established solutions for image and video processing and classification, especially for faces. However, the Reservoir Computing framework has become an attractive method for tapping into a reservoir of complexity that, if so designed, can provide a variety of nonlinear input/output mappings. For this experiment we used static images with the premise that spatial correlations found by the reservoir can be exploited and that each static image provides a dynamic trace through the reservoirs state space. Because of the temporal nature of the ESN, the system design fits naturally with an extension to using video data over just static images. This could find application in such areas as facial expression recognition or any other visual scene where the temporal evolution contains salient information. Importantly, this work is seen as a pilot study for future research into the analysis of video sequences which will fully exploit the properties of ESNs.

The key benefits of an ESN, and broadly speaking, the Reservoir Computing approach, are as follows:

1) The internal memory and complexity of the system allows for the solution of nonlinear mapping problems, suitable for handling spatio-temporal data.
2) All connections and weights within the ESN are fixed; this makes for fast parallel implementation of large reservoirs.
3) The ESN approach is a simpler and more straightforward way to train RNNs as opposed to techniques such as back-propagation through time - this was traditionally one of the difficulties faced when working with RNNs [2]. Here, the only parameters of the ESN are the weights of the output projections that are usually adjusted to generate specific signals from the network. This work does not actually make use of the output projections, but instead samples the dynamics of the ESN as it progresses through time.

Despite these benefits, the exact design of optimal reservoirs is an active area of research and their black-box nature make them difficult to analyse. To reply to this, this work focuses on the general principle of providing a reservoir of complexity, where future work will investigate the strengths and weaknesses of alternative reservoir architectures, e.g. using Long Short Term Memory (LSTM) or Continuous Time Recurrent Neural Network (CTRNN) architectures.

II. Related Work

Echo State Networks were first introduced by Jaeger in [3] and the LSM approach was introduced by Maass et al. [4].

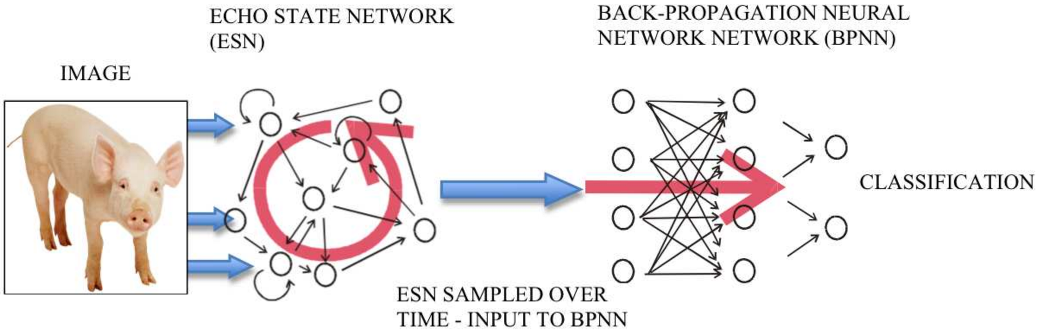

ECHO STATE NETWORK (ESN)
BACK-PROPAGATION NEURAL NETWORK NETWORK (BPNN)
IMAGE
CLASSIFICATION
ESN SAMPLED OVER TIME - INPUT TO BPNN

Fig. 1. System architecture, an input image is coupled with the ESN, injecting energy into the system. The dynamics of the ESN are left to run and sampled at given intervals. This generates a response vector that can be used for training the system as a classifier. Unseen images can be given to the system once it is trained to produce a classification vector. (Image adapted from [2].)

Together they appear under the umbrella term of Reservoir Computing - both approaches provide a reservoir of complexity that can be harnessed to solve a variety of non-linear problems. Whereas the ESN formulation uses recurrent neural networks, the LSM approach is more general, specifying just an input function, into a (liquid) function, then finally to a read out function. As long as the liquid state provides enough rich complexity and dynamics, any system can be used.

The Reservoir Computing approach has been prominently used for speech recognition due to fast performance, handling of temporal data and an easy training methodology. For example, Scherer et al. [5] used ESNs to recognise emotion in speech in real-time. However, there is little work on Reservoir Computing for face recognition - as examples, Grzyb et al. [6] used the LSM approach with spiking neurons to perform face expression recognition on images. Madane et al. [7][1] applied Linear Discriminant Analysis (LDA) for face images as a first step, then ran their features through an ESN in order to perform classification. C. Y. Tsai et al. [8] combined a Kalman filter with an ESN to perform face tracking for a mobile robot - although dealing with faces, this is a different task from the face classification problem tackled in this work.

The ESN used in this experiment was designed to reflect the nature of the input - here images are being used so a 2D signal is presented to the network for one moment in time, then removed, acting as an external field with which the ESN couples with. This is opposed to a 1D signal for speech recognition where the signal is gradually fed into the network at each time update. In this work the ESN is used directly on the input image under the assumption that the complexity of the network can draw out salient underlying features in the data. Additionally, the network size reflects the size of the input images (46×56 pixels and the same number of neurons), projected into a smaller size (a parameter of the system, e.g. 1/2, 1/4 the input image size) for sampling. Initial inspiration for this work was found in Ghani et al. [9] , where they similarly coupled an ESN with a feed-forward network but applied the system to speech recognition.

III. System Architecture

The system consists of two main components and operates as follows; firstly the ESN: here an input image is set up as an external field with which the ESN couples with. The input image is first normalised and then shifted by -0.5 to become signed, then it is scaled by an empirically chosen factor for input into the ESN. The input image is coupled with the network for only a single time step and then the ESNs dynamics are left to run. Then, at specific intervals the ESN state is spatially averaged into a smaller sized representation and sampled; these samples are concatenated to construct a vector representing the input image. For a classification task a number of input images from different classes are run through the ESN generating a set of response vectors. These are then fed into the second component of the system a feedforward three-layer neural network that is trained using the standard back-propagation algorithm i.e. a back-propagation neural network (BPNN). Once trained the system can then be used for classification purposes by running an unseen image through the ESN, then running its output through the feed-forward network to generate a classification vector.

IV. Echo State Network

As the name implies, an Echo State Network is a type of recurrent neural network that has random connections and random weights up to a scale factor. Therefore, initial construction of the network is a simple stochastic process where connectivity percentage between neurons and feedback connectivity percentage for a neuron can be specified. The ESN network is a discrete time recurrent network:

$$x_i(n+1) = f(\sum_j w_{ij} x_j(n) + u_i(n)) = f(q) \qquad (1)$$

[1] The article's title is ”BImplementation of High Speed Face Recognition Based on Karhunen Loeve Transform and Fishers Discriminant, Radial Basis Function of Echo State Neural Network”; we assume that ”BImplementation” is a typographic error.

where $x_i(n+1)$ is the state of neuron i at time-step $n+1$, w_{ij} is the synaptic weight between neurons i and j and $x_j(n)$ is the state of neuron j at time-step n. Here $u_i(n)$ is a scaled and signed pixel value from the external field representation, $\mathbf{u}(n)$, of the input image I, and f is a non-monotonic activation function, specified as follows:

$$\begin{aligned} f(x(n), q) = & r \cdot x(n) + (1-r)b(q) = \\ & r \cdot x(n) + (1-r)tanh(c_1 q) \cdot \\ & tanh(c_2 |q| - h) \end{aligned} \tag{2}$$

where q is defined as in Equ. 1. The general form of the ESN's activation function is based on [10] and the non-monotonic response $b(q)$ for improved retrieval dynamics when the RNN acts as a Hopfield associative memory is described in [11] and depicted in Fig. 2. The non-linearity of the activation function allows for rich dynamics and under certain conditions analysis of its return map has revealed chaotic behaviour [11].

All input images are of the same width and height, $W \times H$ - for this experiment $W = 46$ and $H = 56$. Given $N = W \times H$ neurons, the internal weight matrix is $\mathbf{W} = (w_{ij})$, of size $N \times N$. Due to the fact that a large number of neurons need not be connected, a sparse matrix representation of W is appropriate.

The structure of an ESN is fixed and if large enough, the network is capable of generating a wide variety of dynamics. Most importantly, when correctly designed the network posses the Echo State property, or fading memory property. Here, as the input data is presented to the network, a memory trace is formed within the ESN due to its feedback connections creating an internal memory. It is important to note that not all networks with this configuration have the Echo State property; Jaeger [2] investigated the conditions under which this property exists and found that it was succinctly tied to the spectral radius, α of the weight matrix W. The spectral radius is specified by the magnitude of the largest eigenvalue, $|\lambda_{max}|$, of the networks weight matrix, W, and it was found that the Echo State property is ensured if $\alpha < 1$, achievable by renormalising the spectral radius with respect to the initial magnitude of W. This scaling is tied with the intrinsic timescale of the specific problem being analysed. A small α is best suited for smaller timescales as this gives a 'fast' ESN and vice-versa.

A number of interesting properties of the implemented RNN used for constructing the ESN were studied; firstly, the RNN shows that it is capable of behaving as an associative memory. This nature was first described by Hopfield in [12]. The original work described a binary valued model that was later generalised into a real valued form. The Hopfield one shot memory imprinting scheme is the most interesting element of the original work: in a fully connected RNN, the network weights are determined by pixel correlations between every site across the image. A number of image memories can be stored by progressively adding new image pixel correlations to the appropriate weights. Under certain conditions - if the summation into f is $0 < q < 1$ and the network is left to run by randomly injecting energy into the system - chaotic activity across the neurons with macro-scale emergent patterns can be observed. The associative memory property of the RNN can also be observed by coupling the system with a partial representation of one of the original imprinted images. The network state then evolves to depict the full image memory as stored within the network. Analysis of the network found that even with an on average $\approx 20\%$ connectivity between neurons, the network was able to retrieve the appropriate memories when given a partial input image.

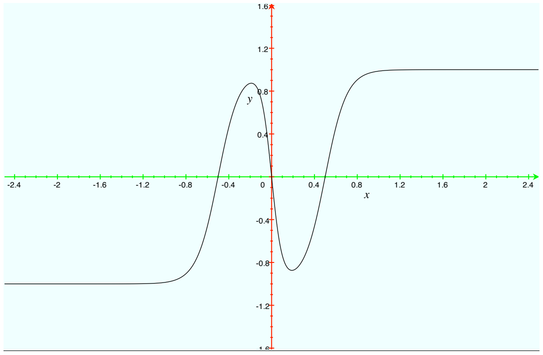


Fig. 2. Non-monotonic neuronal activation function, used for the ESN.

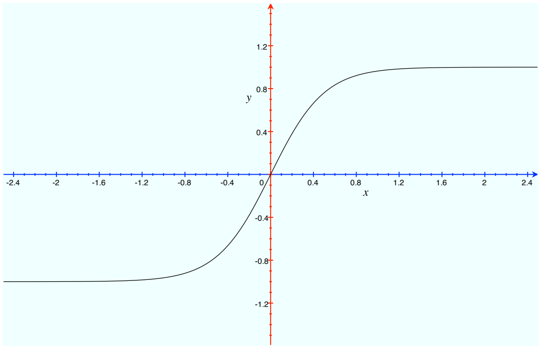


Fig. 3. Form of the sigmoidal activation function, used for the BPNN.

V. Back-Propagation Neural Network

A traditional feed-forward neural network (multi-layer perceptron), with a linear input layer, non-linear hidden layer and linear output layer was used for classification. The standard back-propagation algorithm was used to train the system and optimise synaptic weights - we therefore refer to this network as a back-propagation neural network (BPNN). As is well known, the universal approximation theorem states that every real-valued continuous function can be approximated arbitrarily close by this type of network, having a single hidden layer with sigmoidal activation function.

The value of a neuron x_i at step n is specified as follows:

$$x_i^l(n) = f(\sum_j w_{ij}^l x_j^{l-1}(n)) \tag{3}$$

where the superscript l denotes which layer the neuron is on. Again, w_{ij}^{l} is a synaptic weight, now between a neuron of the current layer l and the previous layer $l-1$. The form of f is sigmoidal for the hidden layer:

$$f(q) = tanh(q) \tag{4}$$

depicted in Fig. 3, otherwise it is linear. The BPNN can also be fed image data directly for classification and it is a point of analysis to compare the properties of the proposed system to the BPNN on its own, acting as a baseline model.

A. System parameters

One difficulty in dealing with neural networks is that a large number of parameters are involved that can greatly influence performance. Table I lists the parameters of the system along with a description of their function.

Parameters	Description
ESN Input scale:	Scales the normalised and signed image data for input into the ESN. Set to 0.001.
ESN weight magnitude:	Initial maximum magnitude for the ESN's random weights. Set to 0.002.
ESN neural connectivity:	Average percentage of connectivity for one neuron to all other neurons. Set to 0.2.
ESN feedback connectivity:	Average percentage of connectivity for a neuron to itself. Set to 0.9.
ESN activation function c_1:	Set to 30.
ESN activation function c_2:	Set to 15.
ESN activation function: r	Activation function response, set to 0.1
ESN sampling interval:	The interval between updates in which to sample the ESN, a longer interval means the ESN will be run for longer. Set to 10.
ESN projection size:	The size of the projection in which neuron values are spatially averaged to, i.e. smaller sizes emphasise the temporality of the ESN. Set to 1/4.
ESN spectral radius factor:	Scaling of the weights to ensure the Echo State property. Set to 0.8.
BPNN learning rate:	Learning rate for back-propagation. Set to $\approx$ 0.5.
BPNN learning momentum:	Momentum rate for back-propagation, can help the optimization escape from local minima. Set to 0.08
BPNN Input scale:	Scaling of the ESN sample vector for input into the BPNN. Set to 0.05
BPNN hidden layer size:	Number of neurons in the hidden layer, Empirically set to 10.

TABLE I
SUMMARY OF SYSTEM PARAMETERS.

For the ESN, the percentage of connectivity was set to $\approx 20\%$ - see Fig. 4 for an example. In this work we do not consider input connectivity percentage and instead set up the input image data as an external field. Secondly, scaling of the data is vital and can be affected by both the input amplitude and magnitude of the synaptic weights. For continued dynamic phenomena to occur the summations into neurons should not cause saturation for example chaotic dynamics have been observed within RNNs of the type used in this work where input into the neuron activation function f (of Equ. 1) is $-1 < q < 1$, i.e. keeping values under the saturation point [11]. The ESN's activation function has tunable parameters specifying DC offset, range and width - these can all contribute to changing the dynamics of the system.

Additionally the BPNN learning rate and momentum parameters are important in converging to a strong error minima - a correct training of the system should have strong generalisation properties, meaning that it should correctly classify unseen images. Currently the training terminates at either a specified number of epochs or when the classification error, ϵ, on the training set is below 0.001. A low error does not guarantee correct classification of images as a number of local minima exist within the error landscape. The momentum parameter can be set to help the optimisation escape from local minima. Section VII describes how the ESN helps smooth the error landscape.

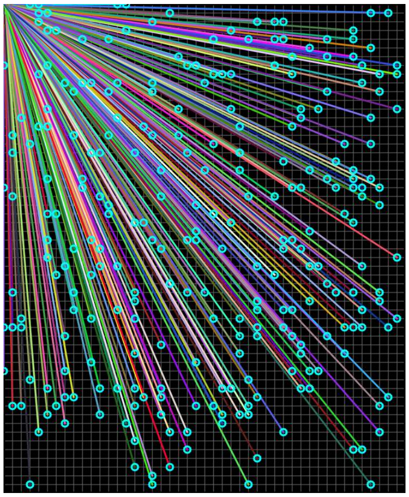
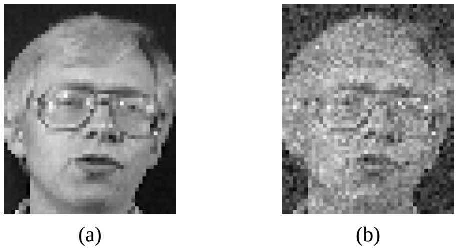


Fig. 4. Neuron connectivity for the neuron at (0,0), a connectivity value of 10% was used an example to more clearly display synaptic connections - the neurons have been laid out on the plane in this diagram, the actual spatiality comes from how the image is coupled with the ESN - here there is a bijection between input pixel and neuron.

VI. Training the System

The training dataset used was the *The ORL Database of Faces* face dataset [13] that consists of a number of images of different people. Each person has ten images of their face, each taken with a slightly different pose. Within the system each person represents a class and these are codified as binary strings. As an example, if there were five classes, the five classes would be codified as $\{10000, 01000, 00100, 00010, 00001\}$, using five output neurons. Presenting an unseen face image to the system will generate a real-valued output classification vector and the index of the maximal value in this vector will specify the classification. For the experiment a subset of five classes was chosen, nine images being given as training and the tenth image given as an unseen image for classification. Looking ahead, now that the system methodology has been validated this training set will be increased in future experiments.

The generalisation properties of the system can be evaluation by firstly measuring the percentage of correct classifications of unseen images. Secondly, different parameter settings can generate more definite output vector classifications, where the maximal value is maximal to a greater extent, relative to the other values in the vector. A search through parameter space in light of this criterion (and possibly the convergence time of training the system) can determine optimal parameter settings. These properties are explored in Section VII.


(a)

(b)

Fig. 5. Example dynamics - a) shows an example face image that is presented to the ESN and b) shows the ESN state after a sequence of time steps showing the randomly constructed dynamics that the network possesses. The perturbation was set to a low amplitude to preserve the appearance of the input image, but generally the network state would appear as a noisy pattern, a trace of the original image remaining hidden to the observer, but unique within the data.

VII. Results and Analysis

During operation, an input image is set up as an external field and coupled with the ESN for a single time-step and the dynamics of the network are then left to evolve. At certain intervals the state of the network is sampled and these vectors are concatenated to form a final response vector. For experimentation a fixed length response vector was chosen and different sub-samplings of the ESN were investigated. In this way we can investigate the effect of increasing the temporal resolution while reducing the spatial resolution. The hypothesis is that important information about the image is contained in the temporal evolution of the ESN and not just the spatial representation. If this is the case then it further validates the use of the ESN for future studies into video data processing. Subsequent analysis found that projecting the ESN into a spatial resolution of $W/4 \times H/4$ worked well in generating a response vector that converged well during BPNN training. Adjusting the time interval showed that the system maintained convergence to a correct classifier, pointing to the existence of a strong temporal memory trace left by an input image coupled with the ESN.

As mentioned earlier, the ESN should maintain its dynamics. This was checked by visualising the ESN as a 2D image where neural values were mapped into pixel gray values, e.g. as in Fig. 5. A certain set of parameters for the ESN could then be evaluated easily e.g. if the system settled to a fixed point then a static image of the neural state resulted. In practice an appropriate scaling of the input while all other parameters are fixed was a logical way to ensure sustained dynamics.

The BPNN network seemed sensitive to the number of hidden neurons - this was the case with or without the ESN. For testing a subset of five data classes, a configuration of 10 hidden neurons was appropriate for good convergence. The momentum parameter seems critical for BPNN to generalise to unseen inputs and also avoid local minima during the optimisation process.

A. Class separation

Table II shows an example classification result for the system (ESN+BPNN) compared to a baseline classifier (BPNN) where images are fed directly to it. For both results the system parameters were set to the same values (see the caption in Table II). The columns give the possible classifications and the rows are unseen test images of certain classes that were given to the trained system. For both situations we find maximal values in the correct locations, but it is clear that the ESN+BPNN results are more strongly separated. As a first step this analysis hints at the ESN providing a more unique signature to the input data for better class separation.

ESN+BPNN	Class 1	Class 2	Class 3	Class 4	Class 5
Test Image Class 1	1.376	0.430	-0.139	-0.246	0.616
Test Image Class 2	0.117	1.483	-0.405	-0.024	0.200
Test Image Class 3	0.044	0.303	0.740	0.196	-0.015
Test Image Class 4	-0.063	0.076	0.154	0.776	0.182
Test Image Class 5	0.109	-0.485	-0.206	-0.054	0.946

BPNN	Class 1	Class 2	Class 3	Class 4	Class 5
Test Image Class 1	0.540	0.471	0.032	0.027	-0.045
Test Image Class 2	-0.100	0.981	-0.168	0.079	0.345
Test Image Class 3	-0.046	-0.072	0.647	0.479	0.144
Test Image Class 4	0.064	0.082	0.247	0.693	0.045
Test Image Class 5	0.297	-0.004	0.129	0.251	0.436

TABLE II

Compares output of classification between the proposed design (ESN+BPNN) and the baseline classifier (BPNN) for unseen images using the parameter settings, learning rate $= 1.4$, learning momentum $= 0.08$, ESN $c_1 = 30$, $c_2 = 15$, $r = 0.1$. ESN projection size $= 1/4$ of input image size, ESN sample interval $= 1$. The rows and columns have maximum values in the correct locations showing that classification was successful for all images. Notably, for the same parameter settings of the BPNN, the ESN provides better class separation.

B. The ESN provides robustness and smooths the training error landscape, helping convergence to a correct minima

Figure 6 shows how robustly the proposed system can converge to a correct error minima, a property very useful for classification tasks. As before, the proposed ESN+BPNN system was compared to the baseline BPNN acting on its own. Robustness was tested by varying the learning rate parameter and observing the evolution of the training regime until the error value $\epsilon < 0.001$. The class separation measure, defined as the angular difference, in degrees, between the system's output vector and the desired class vector (e.g. $\{10000\}$ for class 1 with five possible classes) was used as an indicator. If this measure remains consistent over a large number of parameters then the system is considered robust. The important implication of this is that a robust system can converge to a

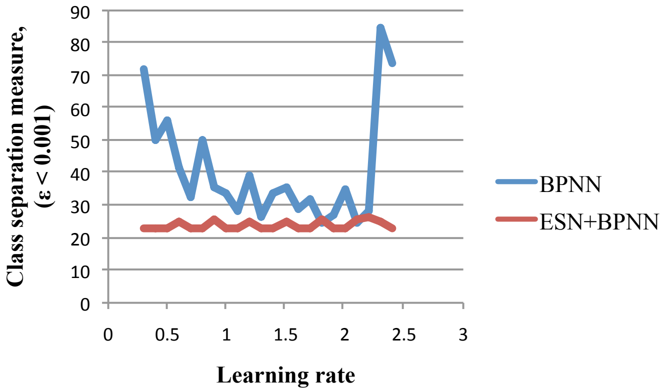


Fig. 6. Convergence variation for adjusting training parameters: the class separation measure, defined as the angular difference, in degrees, between the system's output vector and the desired class vector. For a wide range of learning rate values the ESN+BPNN approach consistently provides convergence to a useful classifier, BPNN on its own does not show this property and there is a large variance in its results. This indicates the robustness provided by the ESN.

good error minima under a wider range of parameters. This would make a neural network based system more reliable and much easier to set up, due to the large number of potential parameter settings. Clearly, from Fig. 6 the ESN+BPNN provides more robust convergence over a range of learning rate parameters (empirically chosen to be around the optimal learning rate for the BPNN on its own). This means that the ESN effectively smooths the training error landscape, reducing the number of poor local minima and making the optimisation regime more robust.

Still, due to the large number of parameters further experimentation and analysis shall be made and what was presented here can be considered only a preliminary analysis. A systematic search through parameter space will hopefully give further insights and determine the most useful parameter configurations.

VIII. Conclusion and Future Work

This research has proposed a novel Reservoir Computing design for face recognition. The system uses an Echo State Network coupled with a Back-Propagation Neural Network for robust data classification. It forms part of an investigation into the applicability of reservoirs and ESNs applied to computer vision problems. Current results show that the proposed system is capable of classifying human face images and that using an Echo State Network increases the robustness of a neural network based image classifier.

A. Implications of the approach: strong spatio-temporal properties suitable for video data processing

The system was designed with a long term goal of processing video data instead of just static images - something that will be explored in future research now that a working system has been implemented. By virtue of its design, the system can easily accommodate video sequences; instead of coupling the ESN with a single static image for one time-step, the ESN would be coupled with a video stream, a new video frame being shown at each time-step (or after a certain number of time-steps of the internal dynamics of the ESN). Thus the general idea is to exploit the ESN's internal memory and rich spatio-temporal dynamics in order to create new and powerful computer vision algorithms. Example applications where ESNs could be useful are in analysing video data for face expression recognition, detection and classification of human movements in general e.g. gait analysis, or the temporal fusion of image features into higher order scene representations.

References

[1] Eric Antonelo, the site administrator, "Reservoir Computing: Shaping Dynamics into Information," 2011, http://reservoir-computing.org, visited on October 12, 2011.

[2] H. Jaeger, "Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the "echo state network" approach," Fraunhofer Institute AIS, St. Augustin-Germany, Tech. Rep., 2002.

[3] ——, "The "echo state" approach to analysing and training recurrent neural networks," GMD - German National Research Institute for Computer Science, GMD Report 148, 2001. [Online]. Available: http://www.faculty.jacobs-university.de/hjaeger/pubs/EchoStatesTechRep.pdf

[4] W. Maass, T. Natschlaeger, and H. Markram, "Real-time computing without stable states: A new framework for neural computation based on perturbations," *Neural Computation*, vol. 14, no. 11, p. 2531–2560, 2002.

[5] S. Scherer, M. Oubbati, F. Schwenker, and G. Palm, "Real-time emotion recognition from speech using echo state networks," *Artificial Neural Networks in Pattern Recognition*, vol. 5064, pp. 205–216, 2008. [Online]. Available: http://www.springerlink.com/index/L238486078W4J352.pdf

[6] B. J. Grzyb, E. Chinellato, G. M. Wojcik, and W. A. Kaminski, "Facial expression recognition based on liquid state machines built of alternative neuron models," *IEEE International Joint Conference on Neural Networks*, no. 1, pp. 1011–1017, 2009. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5179025

[7] S. R. Madane, W. Banu, P. S., and S. C. R. Madane, "BImplementation of High Speed Face Recognition Based on Karhunen Loeve Transform and Fishers Discriminant, Radial Basis Function of Echo State Neural Network," *International Journal of Soft Computing*, vol. 3, no. 3, pp. 248–253, 2008.

[8] C. Y. Tsai, X. Dutoit, K. T. Song, H. V. Brussel, and M. Nuttin, "Robust face tracking control of a mobile robot using self-tuning Kalman filter and echo state network," *Asian Journal of Control*, vol. 12, no. 4, pp. 488–509, 2010.

[9] A. Ghani, T. M. McGinnity, L. Maguire, L. McDaid, and A. Belatreche, *Neuro-Inspired Speech Recognition Based on Reservoir Computing, Advances in Speech Recognition, Noam Shabtai (Ed.)*. InTech, 2010.

[10] H. Nozawa, "A neural network model as a globally coupled map and applications based on chaos," vol. 2, no. 3, pp. 377–386, 1992. [Online]. Available: http://dx.doi.org/10.1063/1.165880

[11] H. Nishimori and I. Opriş, "Retrieval process of an associative memory with a general input-output function," *Neural Networks*, vol. 6, pp. 1061–1067, November 1993. [Online]. Available: http://dx.doi.org/10.1016/S0893-6080(09)80017-8

[12] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the National Academy of Sciences*, vol. 79, no. 8, pp. 2554–2558, Apr. 1982. [Online]. Available: http://www.pnas.org/content/79/8/2554.abstract

[13] AT&T Laboratories Cambridge, "Our Database of Faces (formerly 'The ORL Database of Faces')," 2011, http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html, visited on October 9, 2011.