Design of Reservoir Computing Systems for Noise-Robust Speech and Handwriting Recognition

Azarakhsh Jalalvand, Kris Demuynck, Wesley De Neve, Rik Van de Walle, Jean-Pierre Martens
Multimedia Lab, ELIS, Ghent University
Ghent 9000, Belgium
{Azarakhsh.Jalalvand, Kris.Demuynck, Wesley.DeNeve, Rik.VandeWalle, JeanPierre.Martens}@UGent.be

***Abstract*—In this work, we address the noise robustness of the pattern recognition systems by investigating the application of Reservoir Computing Networks (RCNs) on speech and image recognition tasks. Our work introduces different architectures of RCN-based systems along with a coherent task-independent strategy to optimize the reservoir parameters. We show that such systems are more robust that the state-of-the-arts in the presence of noise and RCNs can be used for both robust recognition tasks as well as denoising approaches. Moreover, the successful application of RCNs on different tasks using the proposed strategy supports our claim that it is task-independent.**

***Keywords*-Reservoir computing networks; speech processing; image processing; artificial neural networks; noise robustness;**

I. Introduction

The claims made by researchers in machine learning is that neuro-dynamical systems are powerful as they can analyze long-term relationships in a natural way. Therefore, they can make a distinction between the dynamics of the signal and those of distortions that have corrupted the signal before the signal reaches the recognition system. In this thesis, we focused on a specific category of neuro-dynamical systems, namely reservoir computing networks [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11] [1].

In order to verify these claims for the case of speech recognition, we devised two research paths. One path was to demonstrate the ability of neuro-dynamical systems to extract long-term dynamical properties of the speech and as such raise the recognition performance. In this way, we should also show that neuro-dynamical systems better suppress the effects of ambient noise and channel distortion than conventional generative model-based systems that only look at local information. As noise robustness is commonly achieved in the acoustic model, and since the performance of a small vocabulary continuous speech recognition (SVCSR) system is almost entirely determined by the quality of the acoustic model it encompasses, we decided to study the robustness in the context of continuous spoken digit recognition and to introduce a strategy to optimize the RCN hyperparameters for the speech processing tasks.

In the second path, we demonstrated that the aforementioned properties also holds in general. We proved that the strategy we introduced to optimize the reservoir parameters also works for image recognition and in particular, handwritten isolated digit recognition.

Apart from the pattern recognition tasks, we also studied the ability of an RCN-based denoising auto encoders (DAEs) to denoise the speech features as well as images.

The rest of this paper is organized as follows. First, we briefly recall the general principles underlying RCNs (Section II). Section III proposes RCN architectures for performing speech processing along with some experimental results. In Section IV, we describe the application of reservoir computing in image processing including a brief overview of the results reported in the dissertation. The paper ends with a number of conclusions, as well as a number of ideas for future research.

II. Reservoir Computing Network (RCN)

In its simplest form, an RCN is a neural network with two particular computational layers: (1) a hidden layer (pool) of recurrently interconnected non-linear neurons, called *reservoir*, driven by inputs and by delayed feed-backs of its outputs and (2) an output layer of linear neurons, called *readouts*, driven by the hidden neuron outputs (Fig. 1). A fundamental point is that the input weights and the recurrent connection weights are initialized from random distributions, and that only the output weights are optimized (trained) for solving the targeted problem.

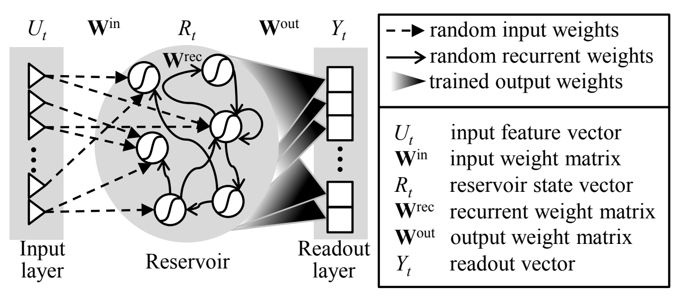


Fig. 1. A basic RCN consists of a reservoir and a readout layer. The reservoir is composed of interconnected **non-linear** neurons with **fixed** random weights. The readout layer consists of **linear** neurons with **trained** weights.

If U_t, R_t and Y_t represent the reservoir inputs, the reservoir outputs and the readouts at time t, the RCN equations can be written as follows:

$$R_t = (1-\lambda)R_{t-1} + \lambda\, f_{res}(\mathbf{W}^{in}U_t + \mathbf{W}^{rec}R_{t-1}) \qquad (1)$$

$$Y_t = \mathbf{W}^{out}R_t \qquad (2)$$

[1]The work is related to the PhD thesis which was defended successfully on 27/02/2015.

with $0 < \lambda \leq 1$, with f_{res} being the non-linear activation function of the reservoir neurons (*hyperbolic tangent* in this work) and with $\mathbf{W}^{in}$, $\mathbf{W}^{rec}$ and $\mathbf{W}^{out}$ being the input, recurrent and output weight matrices, respectively. The constant λ is called the leak rate because Equation (1) represents a leaky integration of the neuron (LIN) activation.

Each individual input is normalized so that it has a zero mean and unit variance over the training examples. The initialized input and recurrent weights to the reservoir nodes are assigned from a normal distribution and they are characterized by four parameters [4]: (1) α_U, the largest singular value of the input weight matrix $\mathbf{W}^{in}$, (2) ρ, the maximal absolute eigenvalue of the recurrent weight matrix $\mathbf{W}^{rec}$, (3) K^{in}, the number of inputs driving each reservoir neuron and (4) K^{rec}, the number of delayed reservoir outputs driving each reservoir neuron. The first two parameters control the absolute and the relative importance of the inputs and the delayed reservoir outputs in the reservoir neuron activation. The latter two control the sparsity of the input and the recurrent weight matrices.

The output weights are such that they minimize the mean squared error (MSE) between the *computed* readouts Y_t and the *desired* readouts D_t over the training examples [4].

Optimization strategy: In [4] we conceived a straightforward strategy to design an RCN. The theory presented there leads to the following conclusions: (1) The input and recurrent weight matrices ($\mathbf{W}^{in}$ and $\mathbf{W}^{rec}$) can be very sparse (5 to 10 elements per row regardless of the reservoir size and the input feature vector size); (2) The spectral radius, ρ, must be tuned to the bandwidth of the observed reservoir inputs (interpreted as time series); (3) The constant λ must be tuned to the minimum number of time steps the reservoir output can be expected to remain constant; (4) The constant α_U must be tuned so that the expected variances of the activation components due to the inputs and the recurrent connections are balanced.

III. Application of RCNs on Speech Processing

Standard Hidden Markov Models (HMMs) incorporate Gaussian Mixture Models (GMMs) to compute state-level acoustic likelihoods. Such systems have reached a high level of performance, but they remain very sensitive to mismatches between the training and the test circumstances. Many research efforts have been directed towards the development of novel front-end and/or back-end techniques for making the systems more resistant to these mismatches.

In this section an RCN-HMM hybrid for continuous digit recognition is investigated. It will be demonstrated that an RCN-HMM hybrid comprising reservoirs can yield good performance for continuous digit recognition in clean and noisy circumstances (tested on Aurora-2 dataset).

A. A hybrid RCN-HMM for speech recognition

An RCN-HMM hybrid works with an HMM that represents the task and a neural network that is supposed to convert the inputs U_t at time t into HMM state likelihoods. The search for the best path through the HMM is found using a Viterbi search. In the case of an RCN-HMM hybrid, the readouts $y_{t,i}$ (with i indexing the readouts) are assumed to resemble the posterior probabilities $P(q_t = i|U_t)$. This means that $z_{t,i} = y_{t,i}/P(q_t = i)$ is a scaled likelihood and consequently, that the best state sequence follows from

$$\hat{\mathbf{q}} = \arg\max_q P(\mathbf{q}, \mathbf{U}) \approx \arg\max_q \prod_{t=1}^{T} z_{t,q_t}\, P(q_t|q_{t-1}),$$

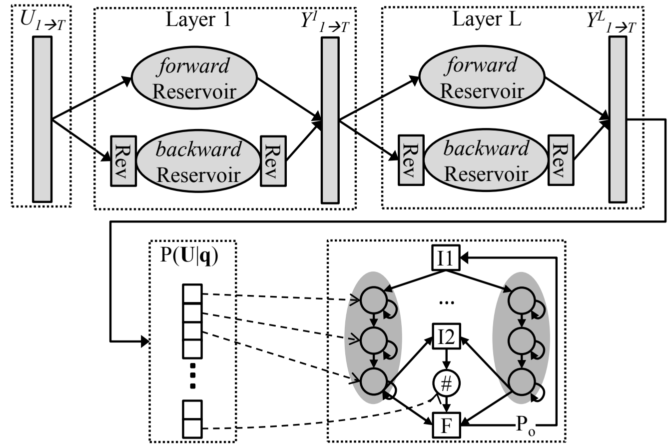


Fig. 2. Architecture of an RCN-HMM hybrid comprising a multi-stage bi-directional reservoir networks for CDR. The HMM has two initial states (I1 and I2), one final state (F) and it comprises 11 multi-state digit models and a single state silence model (#)

Fig. 2 shows the architecture for the case of continuous digit recognition (CDR) and a multi-stage RCN in which each network output is supplied to the next stage [5]. The argument for cascading layers is that the new layer can correct some of the mistakes made by the preceding layers because it offers additional temporal modeling capacity and a new inner space in which to perform the classification. The transition probability P_0 which is added to the digit loop controls the balance between deletions and insertions.

Since $y_{t,i}$ is not confined to [0,1] it is first mapped to that interval before computing $z_{t,i}$. The mapping is achieved by a simple clip-and-scale approach, as described in [3], [5]. The different stages of the RCN are trained independently, one after the other.

As in [5], [10], we use bi-directional RCNs in the different layers. Such an RCN encompasses two (identical) reservoirs: one reservoir processes the frames from left-to-right while the other processes them from right-to-left (see Fig. 2). The readouts at time t are then computed as concatenation of the two reservoir states reached after having processed input vector U_t.

As the acoustic features, we worked with the standard Mel Frequency Cepstral Coefficient (MFCC) setup, delivering 13 static (c_1,.., c_{12} and log$E$), 13 velocity and 13 acceleration features, as well as Mel filterbank features (MelFB). Also, we utilized some noise robust features such as mean and variance normalized features (MVN) and advanced feature extraction systems (AFE). Table I lists the performance of RCN supplied with different acoustic features along with some state-of-the-art systems.

B. An RCN-based denoising auto-encoder

The neural network-based systems that are used to reconstruct the clean feature vectors from the noisy feature vectors are called denoising auto-encoder (DAE). We argue that a complex non-linear dynamical system with memory such as an RCN should be able to do a good job in feature denoising.

Since the MFCCs emerge from a multi-stage process (See Fig. 3), the first question is at which position in this processing scheme, the insertion of a DAE would be the most effective. Traditionally, speech enhancement is often acting upon the Discrete-time Fourier Transform (DFT) (at position P1 in Fig. 3). On the other hand, denoising in the log Mel-frequency domain (at position P2) or the MFCC domain (at position P3) are more appealing due to the much lower dimensionality. A high dimensionality normally results in a higher computational cost. In the case of an RCN-DAE however, this is not true since each reservoir neuron is only simulated by a few inputs. Therefore, we can easily consider an RCN-DAE at all positions.

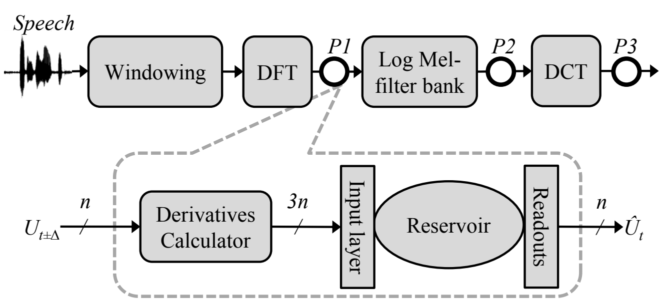


Fig. 3. Possible points to apply the denoising system (top) and their structure in more details (down)

C. Experimental results

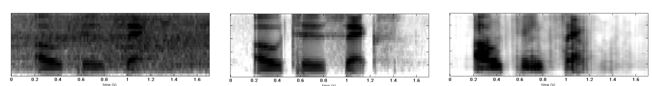
In order to evaluate the robustness of the RCN-based speech recognizer and denoiser, we opted for the renowned Aurora dataset. This database contains clean and noisy utterances, sampled at 8 kHz and filtered with a G712/MIRS characteristic. There are 8440 clean training samples. We have tested our systems on the clean test data (4004 utterances, 13159 digits) as well as on the noisy test sets A-C. The latter sets are created by artificially adding noise to the clean test data at Signal-to-Noise Ratios (SNR) between 20 and -5dB. The vocabulary consists of the digits 0 to 9 and 'oh' (a substitute for 'zero'). The performances are reported as the word error rate (WER%) and in the dissertation we compared the proposed systems with many state-of-the-art system such as Gaussian mixture model-based methods (GMM) and other neural network-based methods.

During the training and validation step, we considered different architectures such as single and multi-layer RCNs, uni- and bi-directional reservoirs, training the neural network with clean samples only as well training with clean and noisy samples (multi-style training), and simple and leaky integrated neurons.

TABLE I
COMPARING AVERAGE WER%S (OVER DIFFERENT NOISE TYPES AND LEVELS) FOR TEST SETS A-C OF AURORA-2 USING A 3-LAYER BI-DIRECTIONAL HYBRID RCN-HMM FOR BOTH CLEAN AND MULTI-STYLE TRAINING.

System	Clean train	Multi-style
GMM (MFCC)	19.7	8.5
GMM (SPLICE)	17.6	12.7
GMM (MFCC-MCE)	15.7	6.4
GMM (AFE)	13.2	8.4
GMM (JUD)	10.3	-
GMM (VTS)	9.4	-
RCN (MelFB)	11.0	**5.4**
RCN (MFCC)	10.7	6.2
RCN (AFE)	**8.9**	5.7

Moreover, in order to illustrate the effect of the DAE, on Fig. 4 we have depicted the MelFB spectrograms for a noisy speech sample (SNR = 5dB) before and after denoising, together with the clean speech spectrogram. It is especially noteworthy that the DAE does an excellent job in the silence parts. This is partly due to the large number of non-speech (silent) frames in the training data.



(a) Noisy features (b) Clean copy (c) Denoised features

Fig. 4. Denoising MelFB features of a sample with street noise of SNR 5dB from Test C.

We inserted an RCN-based DAE at different points in the MFCC extractor, namely after applying DFT, after the Mel-filter bank, and after the DCT. However, none of these approaches resulted in a significant improvement of the noise robust digit recognition with respect to the case of other noise robust features such as AFEs.

IV. APPLICATION OF RCNS ON IMAGE PROCESSING

With the aim of improving robustness and the simplicity of training, we have shown that reservoir computing networks (RCNs) are able to offer an elegant alternative model in the field of speech recognition and enhancing speech features. The aforementioned observations motivated us to experimentally investigate whether the devised techniques could also be successfully applied in a domain different from speech, namely the visual domain, and in particular handwriting recognition.

In this section, we show that RCNs have great potential for achieving good performance in image processing from noise corrupted images. We try to prove our case by considering handwritten digit recognition (HDR) on MNIST as a standard benchmark in the field.

It is now generally acknowledged that conventional and modern neural networks such as Deep Neural Networks

(DNNs) and Convolutional Neural Network (CNN) perform well for image recognition, but that they are still hard to train; it takes a lot of time and the hyperparameters of the training process must be set properly.

Moreover, in spite of the impressive results that have thus far been achieved in clean conditions, many of these approaches dramatically fail to recognize digits from noisy samples. Consequently, new research has been directed towards improving the robustness of HDR against the presence of noise.

A. RCN-based architectures for image processing

In many neural network-based HDR systems, the input is a pixel array of the whole image. However, in order to exploit the dynamical system properties of an RCN, we need to create a sequential input stream. This can be achieved by scanning the image column-wise (horizontal scanning) or row-wise (horizontal scanning) or combinations of these two approaches.

The readouts of the RCN that will be encompassed in the recognizer correspond to the ten digits and to the white space which is present in each digit image.

1) Basic architecture: A trivial procedure leading to the desired input stream is horizontal scanning: the image is scanned column-wise from left to right and the subsequent columns (called frames) form the input vector sequence (see Figure 5).

The digit scores are obtained by accumulating the digit readouts across time (the Σ component) and a winner-take-all algorithm returns the winner digit with the highest readout activity.

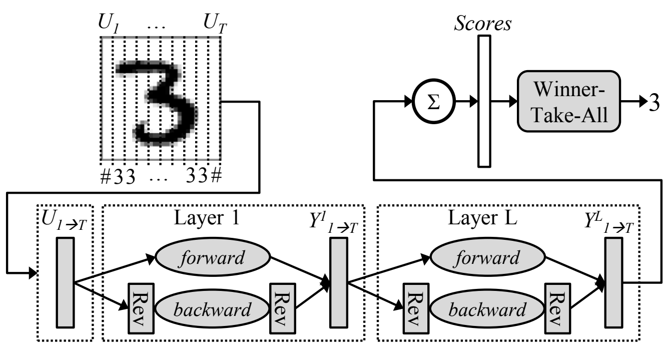


Fig. 5. Architecture of a deep RCN-based digit recognizer leveraging bi-directional processing in each layer.

2) More complex architectures: Given that it is also suitable for continuous HWR, horizontal image scanning seems to be an obvious choice. However, for isolated digit recognition, one can also consider vertical scanning, as well as a combined scanning approach. The ones we propose here are depicted in Figure 6.

B. Experimental setup

In this section, we present the experimental framework that was set-up in order to assess the potential of the proposed system configurations.

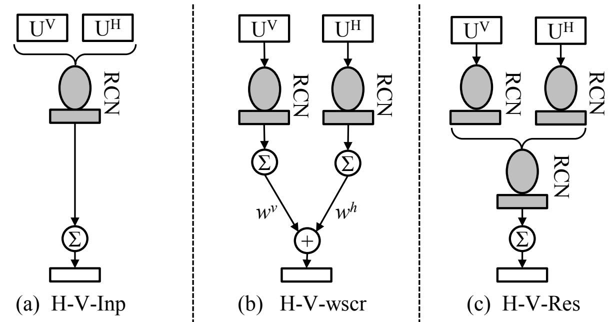


Fig. 6. Different ways of combining horizontal (H) and vertical scanning (V) in a system: (a) supply the RCN with one row and one column of the image, (b) compute a weighted sum of the digit scores (accumulations over time) emerging from an H-RCN and a V-RCN and (c) supply the H-RCN and V-RCN outputs to another RCN and accumulate the scores of the readouts of this RCN.

MNIST corpus: The MNIST corpus consists of clean handwritten isolated digit samples, grouped into two datasets: a training set consisting of 60K samples and a test set consisting of 10K samples. Each sample is represented by a 28×28 gray-scale encoded pixel array. We report the digit error rate (DER%) on the validation or test set as the recognition performance measure.

In order to conduct experiments on noise robustness, we construct a multi-condition dataset by dividing the dataset into six equally large parts. One part is left unaltered and serves as a clean dataset. The images of the other five parts are corrupted with noise, one noise type per part. The considered noise types are Gaussian, Salt & Pepper, Speckle, Block and Border (see Fig. 7).

Fig. 7. From left to right, a clean MNIST sample and its corresponding noisy versions: salt & pepper, border, Gaussian, block, and speckle, respectively.

C. Experimental results on clean dataset

In this phase, we assess the performance of our systems as a function of the reservoir size (the number of neurons in the reservoir), the depth of the RCN (the number of layers) and the direction of scanning in the front-end. Based on the findings, we designed a system of type H-V-res that consists of two 2-layer systems comprising a 16K reservoir in each layer, followed by a single layer RCN encompassing a 16K reservoir. This system has 880K trainable parameters and it achieves a DER of 0.81% on the MNIST test set (see Table II), showing that it is competitive with formerly reported systems working with the same inputs and being trained on the same training samples.

D. Recognition for raw noisy images

In this case, we distinguish two experimental settings: one in which the system is trained on clean images only (clean

TABLE II
REFERENCE RESULTS ON MNIST USING THE ORIGINAL TRAINING SET.

Model	Year	DER%
DNN committee	2011	1.70
DBN	2010	1.03
CNN	1998	0.95
MLP + maxout + dropout	2013	0.94
DCN	2011	0.83
Deep RCN	[This work]	**0.81**
DBM + dropout	2012	0.79
CNN + maxout + dropout	2013	0.45

training) and one in which the system is trained on a mix of clean samples and samples corrupted by the five noise types that are also present in the test set (multi-condition training).

A brief list of results are listed in Table III. For comparison with the state-of-the-art, the table also includes the results for deep belief networks (DBN) systems we could find in the literature. In the clean training case, the presence of noise induces a dramatic increase of the DER in all systems. None of the systems stands out on all conditions. The DBN system wins in three of the six conditions, the RCN in the other three, be it that on average the DBN system yields the lowest DER. It is fair to say that RCNs degenerate at more or less the same pace as DBNs when the mismatch between the training and the test conditions increases. We interpret this as a positive result because deep neural networks are acknowledged for their good noise robustness and because the research on RCNs is still in its initial phase.

In the multi-condition training case, the effect of the noise is much more moderate. The H-V-res system now yields an average error rate of only 3.54% and it outperforms the DBN systems in all conditions for which a comparison is possible. Combining two scanning directions seems to help significantly as long as there is no big mismatch between the training and the test conditions (this means clean test for clean training and all tests for multi-condition training). However, more research is needed to establish why the advantage of the combination disappears in mismatched conditions.

E. Recognizing connected digits

As described in Section IV-A, the capability of processing temporal information makes it possible to recognize the digit by scanning the image. Consequently, one can train an RCN by scanning the isolated digits horizontally and operate this system on the connected samples without any extra pre-processing (e.g., digit segmentation). This is a noticeable discrepancy between RCNs and many other conventional neural networks. Figure 8 depicts the output of a multi-conditionally trained RCN with horizontal scanning (the H system in Table III) which has been supplied with a concatenation of multiple noisy digits.

F. Removing the noise in the front-end

Denoising the input images in the front-end is another approach to reduce the mismatch between training and testing

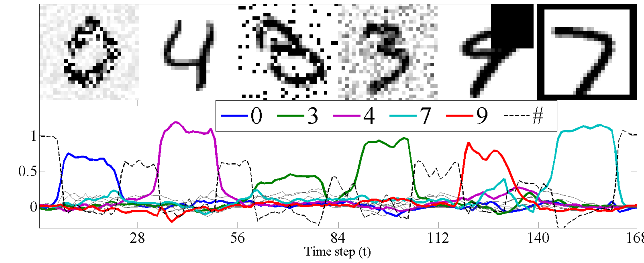


Fig. 8. The readouts of a multi-conditionally trained RCN with horizontal scanning supplied with a concatenation of multiple noisy digits.

TABLE IV
THE INFLUENCE OF ADDING AN RCN-BASED DAE IN FRONT OF THE CLASSIFIER ON THE PERFORMANCE OF THE RCN-BASED RECOGNIZER (AS DER%) ON THE NOISY VERSION OF THE MNIST DATASET.

Classifier	DAE	Clean	Average
DBN-2010	RBM-based	1.24	-
DBN-2013	AMC-SSDA	1.50	2.27
H-V-res	-	**0.81**	37.06
H-V-res	RCN	1.03	2.08
H-V-res (RT)	RCN	1.22	**2.06**

conditions. Like in speech processing phase, we propose an RCN-based denoising Auto-Encoder (DAE) to denoise the input features.

For fixing the hyper-parameters of the DAE reservoirs, we follow the same strategy as before, but this time under the assumption that the dynamics of the targeted outputs are identical to the dynamics of the inputs. Moreover, we established that bi-directional processing is also helpful for this task but that it suffices to stack three (instead of five) successive frames in the DAE input. Since the output of the DAE is a denoised version of the input feature vector, the number of trainable parameters of such an RCN-based DAE of the size N is $28 \times (N + 1)$, with 28 being the number of pixels per column/row.

Without reporting the results in detail, we mention that neither changing the scanning direction nor combining two scanning directions in an H-V-res like system leads to any significant improvement. Because the aim of denoising is to find and remove the noise patterns and the noise types encountered in this work are direction-independent.

Based on the above findings, we also considered a 2-layer RCN with 32K reservoirs in each layer as the reference (1.8M trainable parameters).

Figure 9 shows the performance of the RCN-DAE on denoising some examples.

G. Recognition for denoised images

In order to evaluate the influence of the RCN-based DAE on the recognition, we test the cascade of the RCN-based DAE and the H-V-res system we formerly trained on clean images. The results obtained with this cascade are listed in Table IV. It is clear that the denoiser introduces a dramatic gain in noise

TABLE III

DER (IN %) PER NOISE TYPE FOR THE CASES OF CLEAN AND MULTI-CONDITION TRAINING. THE LAST ROW SHOWS THE DER OF A MULTI-COLUMN RCN-BASED RECOGNIZER COMPRISING TWELVE SUB-SYSTEMS EACH TRAINED ON ONE NOISE CONDITION AND ONE DIRECTION.

	System	Clean	Gaussian	S & P	Speckle	Block	Border	Average
Clean	DBN-2010	1.03	-	-	-	33.78	66.14	-
	DBN-2013	1.09	**29.17**	**18.63**	**8.11**	25.72	90.05	**28.80**
	H-V-res	**0.81**	32.10	38.91	49.32	**21.85**	79.34	37.06
Multi	DBN-2010	1.68	-	-	-	8.72	1.95	-
	H-V-res	**1.50**	**3.08**	**3.75**	**4.32**	**6.82**	**1.75**	**3.54**

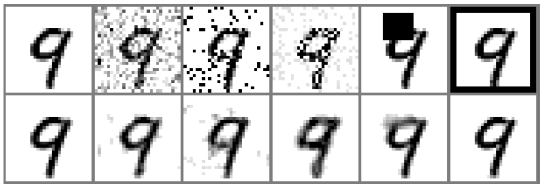


Fig. 9. One clean and five noise corrupted samples of digit 9 (top) and the corresponding outputs of the RCN-based DAE.

robustness of the H-V-res system at the cost of only a minor loss of accuracy in the case of clean images.

In theory, the just tested configuration is sub-optimal because it implies a mismatch between training and testing. Therefore, we also trained an H-V-res system on denoised training images (called H-V-res (RT)). However, to our surprise, the figures in Table IV show no significant improvement over the sub-optimal system. Apparently, there is no need to retrain the recognizer every time the DAE is improved (e.g., by taking a new noise type into account).

The results obtained for the H-V-res system embedding a DAE show that image denoising in combination with clean training is more effective than multi-condition training, even though the latter is over optimistic because it is tested on noise types that were present during training. This is a remarkable result since a limited study involving border noise and block noise came to the opposite conclusion for a system encompassing sparse DBNs. In that study, a clean trained DBN, a multi-conditionally trained DBN, and a clean trained DBN supplied with the denoised images led to the DERs of 22.7%, 4.6% and 6.4%, respectively.

V. CONCLUSION

The main focus of this dissertation was of course on the noise-robustness of Reservoir Computing. In this respect, I conducted a large number of experiments on continuously spoken digit recognition and a small number of experiments on isolated handwritten digit recognition.

This work showed that RCNs focus more on the salient relation between the input features and the acoustic units and less on the fine details, hence they lose some performance in perfectly matched conditions with sufficient training data (clean speech training and clean speech testing), but gain robustness in the other conditions.

The conceived reservoir design strategy which was tuned to the task of clean spoken digit recognition led to quasi-optimal reservoirs for noisy spoken digit recognition, phone recognition (the same input features but another task) and handwritten digit recognition (different input features but a related task).

We hope to apply some of the ideas which lead to uncertainty decoding and convolutional neural networks in an RC-backend. Also, we intend to extend our research to the noise robust large vocabulary speech recognition (e.g. Aurora-4). Moreover, it would also be interesting to explore other image processing applications, such as recognizing street view house numbers and medical image analysis.

ACKNOWLEDGMENT

The research presented here has received funding from Ghent University, iMinds, the European Community's Seventh Framework Program (FP7) and the Flemish Science Foundation (FWO).

REFERENCES

[1] A. Jalalvand, "Design of reservoir computing systems for the recognition of noise-corrupted speech and handwriting," Ph.D. dissertation, Ghent University, 2015.

[2] A. Jalalvand, F. Triefenbach, D. Verstraeten, and J.-P. Martens, "Connected digit recognition by means of reservoir computing," in *Proc. Interspeech*, 2011, pp. 1725–1728.

[3] A. Jalalvand, F. Triefenbach, and J.-P. Martens, "Continuous digit recognition in noise: Reservoirs can do an excellent job!" in *Proc. Interspeech*, 2012, p. ID:644.

[4] A. Jalalvand, F. Triefenbach, K. Demuynck, and J.-P. Martens, "Robust continuous digit recognition using reservoir computing," *Computer Speech and Language*, vol. 30, no. 1, pp. 135 – 158, 2015.

[5] A. Jalalvand, K. Demuynck, and J.-P. Martens, "Noise robust continuous digit recognition with reservoir-based acoustic models," in *Proc. ISPACS*, 2013, p. ID:99.

[6] ——, "Feature enhancement with a reservoir-based denoising auto encoder," in *International Symposium on Signal Processing and Information Technology, Proceedings*. Proc. ISSPIT, 2013, p. 6.

[7] A. Jalalvand, W. De Neve, R. Van de Walle, and J.-P. Martens, "Noise robust handwritten digit recognition with reservoir computing networks," submitted, 2014.

[8] A. Jalalvand, G. Van Wallendael, and R. Van de Walle, "Real-time reservoir computing network-based systems for detection tasks on visual contents," in *CICSyN*, 2015.

[9] F. Triefenbach, A. Jalalvand, B. Schrauwen, and J.-P. Martens, "Phoneme recognition with large hierarchical reservoirs," in *Proc. NIPS*, 2010, pp. 2307–2315.

[10] F. Triefenbach, A. Jalalvand, K. Demuynck, and J.-P. Martens, "Acoustic modeling with hierarchical reservoirs," *IEEE Trans. Audio, Speech and Language Processing*, vol. 21, no. 11, pp. 2439–2450, 2013.

[11] ——, "Context-dependent modeling and speaker normalization applied to reservoir-based phone recognition," in *Proc. Interspeech*, 2013, pp. 3342–3346.