A Deep Neural Network Approach for Missing-Data Mask Estimation on Dual-Microphone Smartphones: Application to Noise-Robust Speech Recognition

Iván López-Espejo*, José A. González†, Angel M. Gomez*, and Antonio M. Peinado*

*Dept. of Signal Theory, Telematics and Communications, University of Granada, Spain
†Dept. of Computer Science, University of Sheffield, UK
{iloes,amgg,amp}@ugr.es,j.gonzalez@sheffield.ac.uk

Abstract. The inclusion of two or more microphones in smartphones is becoming quite common. These were originally intended to perform noise reduction and few benefit is still being taken from this feature for noise-robust automatic speech recognition (ASR). In this paper we propose a novel system to estimate missing-data masks for robust ASR on dual-microphone smartphones. This novel system is based on deep neural networks (DNNs), which have proven to be a powerful tool in the field of ASR in different ways. To assess the performance of the proposed technique, spectral reconstruction experiments are carried out on a dual-channel database derived from Aurora-2. Our results demonstrate that the DNN is better able to exploit the dual-channel information and yields an improvement on word accuracy of more than 6% over state-of-the-art single-channel mask estimation techniques.

Keywords: Dual-microphone, Robust speech recognition, Mask estimation, Smartphone, Deep neural network, Missing data imputation

1 Introduction

Robustness in automatic speech recognition (ASR) is still a key issue for enabling ASR to operate in real world conditions. In fact, with the increasing availability of ASR software running on mobile devices, this issue is now more important than ever before. Thus, distortions such as acoustic noise, reverberation, channel distortion, and so on, which are expected to occur during the normal use of ASR in mobile platforms, can harm ASR performance to a level that makes it simply useless.

One way to improve the robustness against noise consists of equipping the device with several microphones. Hardware price decreasing, along with its miniaturization, has allowed latest smartphones to come with a dual-microphone to

* This work has been supported by the MICINN TEC2013-46690-P project.

perform noise reduction. Recent works propose taking advantage of this new feature in robust speech recognition [1] and speech enhancement [2]. In [2], power level difference (PLD) between the two microphones is used to estimate a spectral gain mask. In a conversational position (i.e. phone loudspeaker placed at the ear) speech power at the primary microphone of the device (related to the first channel) tends to be greater than at the secondary one (related to the second channel), while the noise power received at both microphones is almost the same. Thus, a power level ratio measure (i.e. the ratio between the noisy speech power at the first and second channels) can be used to determine if speech or noise is dominant in each time-frequency (T-F) bin.

Another recent development in the field of ASR has been the application of deep neural networks (DNNs) to improve ASR performance in different ways. Besides their use for acoustic modeling [3, 4], other authors have successfully applied DNNs to the problem of mask estimation in noise-robust ASR [5, 6]. This problem involves identifying the T-F bins in a noisy spectrogram that are dominated by speech or noise. Once the mask is estimated, it can be used either to perform imputation (i.e. spectral reconstruction) on the missing spectral features (i.e. the T-F bins dominated by the noise) [7, 8] or to perform speech recognition with incomplete data, the so-called marginalization approach [9]. In both cases, the performance of these approaches depends heavily on the quality of the estimated binary masks.

In this paper we focus on exploiting the dual-microphone equipped on modern smartphones in order to improve the robustness against acoustic noise of ASR systems. In particular, we propose a novel technique for missing-data mask estimation that takes advantage of the availability of two microphones and the learning capabilities of DNNs. The proposed technique consists of a DNN that is trained on the noisy speech log-Mel features extracted from the signals captured by both microphones. Thus, the DNN is able to estimate a binary mask for a signal acquired by the first channel. Then, the truncated-Gaussian based imputation (TGI) algorithm proposed in [8] is used to compensate the features that are dominated by noise. To assess the performance of the proposed technique, a new database called AURORA2-2C (AURORA2 - 2 Channels - Conversational Position) has been developed [1]. This noisy speech database is based on the well-known Aurora-2 database [10] and tries to emulate speech signals that were recorded with a dual-microphone smartphone in a conversational position. Unlike other related work such as [5, 6] where a wide set of features (amplitude modulation spectrogram, relative spectral transform and perceptual linear prediction, pitch-based features, etc.) is extracted to feed the DNN, in our proposed system the DNN directly provides an estimation of the binary mask by just using dual-channel log-Mel spectral features. Through this approach we can obtain very good results with less computation.

This paper is organized as follows. In Section 2 the spectral imputation technique used here is briefly described. The proposed method for missing-data mask estimation on smartphones with dual-microphone is explained in Section 3. The experiments and results are presented in Section 4. Finally, conclusions and future work are summarized in Section 5.

2 Missing-Feature Compensation

Binary masks allow us to distinguish between spectro-temporal regions dominated by speech or noise, classifying each T-F bin of a noisy speech spectrogram as reliable (speech dominates) or unreliable (noise dominates). Binary masks are very useful in robust speech recognition, since they are used for missing-feature approaches such as marginalization and data imputation. In the case of marginalization, output probabilities for decoding are calculated by only taking into account the reliable features [9]. On the other hand, data imputation (also known as spectral reconstruction) employs reliable spectro-temporal regions in order to estimate values for the unreliable parts of the noisy spectrogram [7, 8].

One of these reconstruction algorithms is the TGI technique [8], which is chosen in this work. This technique is based on the well-known *log-max* model [11] which states that $y \approx \max(x, n)$, where y, x and n represent the noisy speech, clean speech and noise features, respectively, expressed in the log-Mel domain. Thus, y is an upper bound for the masked clean speech energy, i.e. $x \in (-\infty, y]$. This fact is exploited by the TGI method to achieve accurate estimates.

The algorithm operates on a frame-by-frame basis. At time frame t, the mask segregates the noisy observation into reliable and unreliable components, i.e. $\mathbf{y}^{(1)}(t) = \left\{\mathbf{y}_r^{(1)}(t), \mathbf{y}_u^{(1)}(t)\right\}$[1]. Clean speech estimates for reliable elements are the observations themselves, $\hat{\mathbf{x}}_r^{(1)}(t) = \mathbf{y}_r^{(1)}(t)$, while unreliable elements are estimated using minimum mean square error (MMSE) estimation. Taking into account that clean speech is modeled by means of a Gaussian mixture model (GMM) with M components, it can be shown that, for those features labeled as unreliable, the clean speech estimate is

$$\hat{\mathbf{x}}_u^{(1)}(t) = \sum_{\mathrm{k}=1}^{M} P\left(\mathrm{k} \left| \mathbf{y}_r^{(1)}(t), \mathbf{y}_u^{(1)}(t) \right.\right) \hat{\mathbf{x}}_u^{(1,\mathrm{k})}(t), \tag{1}$$

where $\hat{\mathbf{x}}_u^{(1,\mathrm{k})}(t)$ corresponds to the mean of a right-truncated Gaussian distribution defined in the interval $\left(-\infty, \mathbf{y}_u^{(1)}(t)\right]$ given the k-th Gaussian of the clean speech model. The posterior $P\left(\mathrm{k} \left| \mathbf{y}_r^{(1)}(t), \mathbf{y}_u^{(1)}(t) \right.\right)$ can be understood as the weight of the partial estimate $\hat{\mathbf{x}}_u^{(1,\mathrm{k})}(t)$. Note that correlations between the different elements in the feature vector can be exploited in a precise way, since $\mathbf{y}_r^{(1)}(t)$ conditions the value of $\hat{\mathbf{x}}_u^{(1)}(t)$ according to the posterior probabilities.

3 DNN-Based Proposed System

A DNN, which is an MLP (multilayer perceptron) with many hidden layers, is used here to estimate binary masks from dual-channel noisy speech. Particularly, a feedforward neural network with two hidden layers is employed as in [5, 6]. In order to apply such an approach, the speech features used as input data and

[1] Superscript $^{(1)}$ indicates that TGI is applied on the signal from the first (main) channel.

the desired output (target) must be defined. These are described in detail in Subsection 3.1. In addition, to overcome some difficulties in training a neural network with many hidden layers from scratch, the parameters (e.g. weights) of the DNN are initialized according to an unsupervised generative pre-training by considering each pair of layers as restricted Boltzmann machines (RBMs) [12]. These are introduced in Subsection 3.2. The input layer and the first hidden layer behave as a Gaussian-Bernoulli RBM (GRBM), while the rest of pairs of layers (first and second hidden layers, and second and output layers) behave as Bernoulli-Bernoulli RBMs. Features described in Subsection 3.1 are used to train the GRBM, while the inferred states of its hidden units are used to train the second RBM, and so on. The resulting deep belief net (i.e. the multilayer generative model that consists of the stack of RBMs) is used to initialize the feedforward neural network. Then, a fine-tuning second stage is performed in which the DNN is trained in a supervised manner by using the backpropagation algorithm. The cross-entropy criterion was chosen for backpropagation learning.

3.1 Features and Target

Binary mask estimation is performed in the log-Mel domain, where most of the spectral reconstruction algorithms operate. The proposed DNN works on a frame-by-frame basis, i.e. the DNN returns a binary mask for each frame in the utterance. Let the dual-channel noisy speech log-Mel features at time frame t be

$$\mathbf{y}(t) = \begin{pmatrix} \mathbf{y}^{(1)}(t) \\ \mathbf{y}^{(2)}(t) \end{pmatrix}, \tag{2}$$

where $\mathbf{y}^{(i)}(t)$, $i = 1, 2$, is the noisy speech log-Mel feature vector obtained from the signal acquired by the i-th microphone of the device (channel i). Then, the input for the DNN at time t is the stacked vector

$$\mathcal{Y} = \begin{pmatrix} \mathbf{y}(t-L) \\ \vdots \\ \mathbf{y}(t+L) \end{pmatrix}, \tag{3}$$

where $L \geq 0$ determines the size of the temporal window around frame t, that is $2L+1$. Thus, the dimensionality of the input vector is $d = 2 \cdot \mathcal{M} \cdot (2L+1)$, where $\mathcal{M}$ is the number of channels of the Mel filterbank.

On the other hand, the target is an oracle binary mask vector corresponding to feature vector $\mathbf{y}^{(1)}(t)$. Thus, the size of each output vector is $\mathcal{M} \times 1$. Note that oracle masks were obtained by direct comparison between the clean and noisy utterances using a threshold of 7 dB signal-to-noise ratio (SNR).

3.2 Restricted Boltzmann Machines

A diagram of a restricted Boltzmann machine (that can be seen as a two-layer neural network) is shown in Figure 1. RBMs are mainly used to initialize the set of parameters of a DNN to avoid falling into local minima during backpropagation learning. This could happen because of the complex error surface derived

from the large number of hidden layers [12]. An RBM consists of a visible layer with stochastic units (that represent input data) which are only connected to the stochastic units in the hidden layer. Hidden units are usually modeled by Bernoulli distributions. On the other hand, visible units can be modeled with either Bernoulli or Gaussian distributions. In the first case the resulting model is referred as Bernoulli-Bernoulli RBM (BRBM), while the second as Gaussian-Bernoulli RBM (GRBM). GRBMs are very useful to model real-valued input data (e.g. input features), so that they are often used as the first level of a multilayer generative model built with stacked RBMs [3].

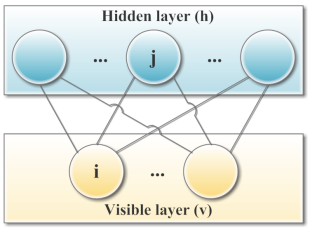

Fig. 1: Example of a restricted Boltzmann machine.

Let $\mathbf{v}$, $\mathbf{h}$ and θ be the visible units, the hidden units and the set of parameters (e.g. weights) of an RBM, respectively. The probability of a visible vector given the set of parameters is obtained by summing over all hidden vectors as

$$P(\mathbf{v}|\theta) = \frac{1}{Z}\sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h}|\theta)}, \tag{4}$$

where $Z = \sum_{\mathbf{v}}\sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h}|\theta)}$ is known as the partition function and $E(\mathbf{v},\mathbf{h}|\theta)$ is an energy function that defines the joint configuration of the visible and hidden units. For a BRBM, the energy function is

$$E_B(\mathbf{v},\mathbf{h}|\theta) = -\sum_{i=1}^{V}\sum_{j=1}^{H} w_{ij}v_i h_j - \sum_{i=1}^{V} a_i v_i - \sum_{j=1}^{H} b_j h_j, \tag{5}$$

and in the case of a GRBM,

$$E_G(\mathbf{v},\mathbf{h}|\theta) = -\sum_{i=1}^{V}\sum_{j=1}^{H} w_{ij}v_i h_j + \frac{1}{2}\sum_{i=1}^{V}(v_i - a_i)^2 - \sum_{j=1}^{H} b_j h_j, \tag{6}$$

where w_{ij} represents the symmetric weight between the visible, v_i, and hidden, h_j, units, and a_i and b_j their respective bias terms. The total number of visible and hidden units are V and H, respectively.

The set of parameters θ is estimated by maximizing $\log P(\mathbf{v}|\theta)$ from training data. For instance, this approach yields the following simple updating equation for the set of weights:

$$\Delta w_{ij} = \epsilon\cdot(\mathbb{E}_{data}[v_i h_j] - \mathbb{E}_{model}[v_i h_j]), \tag{7}$$

where ϵ is the learning rate and $\mathbb{E}[\cdot]$ indicates expectation under the corresponding distribution. To overcome the difficulties in getting samples of $\mathbb{E}_{model}[v_i h_j]$,

Hinton proposed in [13] a fast algorithm called contrastive divergence (CD). Briefly, this algorithm performs alternating Gibbs sampling from visible units initialized to a training data vector [3]. In order to perform the CD algorithm, the following conditional probabilities are employed in the case of a BRBM: $P_B(h_j = 1|\mathbf{v}, \theta) = \sigma(\sum_i w_{ij} v_i + b_j)$ and $P_B(v_i = 1|\mathbf{h}, \theta) = \sigma(\sum_j w_{ij} h_j + a_i)$, where $\sigma(x) = 1/(1+\exp(-x))$ is the sigmoid function. For the case of a GRBM, conditional probabilities can be calculated as $P_G(h_j = 1|\mathbf{v}, \theta) = \sigma(\sum_i w_{ij} v_i + b_j)$ and $P_G(v_i = 1|\mathbf{h}, \theta) = \mathcal{N}(\sum_j w_{ij} h_j + a_i, 1)$, where v_i is real-valued in this case and $\mathcal{N}$ denotes the normal distribution.

4 Experimental Results

4.1 Experimental Framework

In this work, the European Telecommunication Standards Institute front-end (ETSI FE, ES 201 108) [14] is used to extract acoustic features from the speech signal. Twelve Mel-frequency cepstral coefficients (MFCCs) along with the 0th order coefficient and their respective velocity and acceleration form the 39-dimensional feature vector used by the recognizer. Cepstral mean normalization (CMN) is applied to improve the robustness of the system against channel mismatches. For spectral reconstruction, 23-component log-Mel feature vectors are employed (i.e. $\mathcal{M} = 23$). After reconstruction, the discrete cosine transform (DCT) is applied to obtain the final cepstral parameters.

All the techniques are evaluated on the AURORA2-2C database reported in [1]. AURORA2-2C is generated from Aurora-2 [10] data and emulates speech acquisition using a dual-microphone mobile device in a conversational position. In AURORA2-2C two test sets (A and B) are defined, each one with utterances contaminated with different kind of noises at the same SNRs (referred to the first channel) as in Aurora-2.

The acoustic models used by the recognizer are trained on clean speech. Left to right continuous density hidden Markov models (HMMs) with 16 states and 3 Gaussians per state are used to model each digit. Silences and short pauses are modeled by HMMs with 3 and 1 states, respectively, and 6 Gaussians per state [10].

The binary masks estimated by the proposed DNN-based technique are compared with those calculated by thresholding an estimation of the *a priori* SNR of the first channel and used by the TGI algorithm (T-SNR) [8]. The *a priori* SNR for each T-F bin, $\xi(k,t)$, is approximated by using the following maximum likelihood (ML) estimator [15]:

$$\hat{\xi}(k,t) = \max\left(\frac{|Y_1(k,t)|^2}{|\hat{N}_1(k,t)|^2} - 1, 0\right), \tag{8}$$

where $|Y_1(k,t)|^2$ is the filterbank output power spectrum of the noisy speech in the first channel at frequency bin k and time frame t, being $|\hat{N}_1(k,t)|^2$ the corresponding noise power spectrum estimate. As in [8], noise estimates are obtained by linear interpolation between the averages of the first and last 20

frames in the log-Mel domain. Finally, each T-F bin of the mask, $\hat{m}(k,t)$, is calculated as

$$\hat{m}(k,t) = \begin{cases} 1 & \text{if } 10\log_{10}\hat{\xi}(k,t) \geq \gamma; \\ 0 & \text{otherwise,} \end{cases} \quad (9)$$

where $\gamma = 0$ dB is the SNR threshold. This value was experimentally chosen by means of a validation dataset.

The DNN was trained using 19200 sample pairs of input-output vectors. Training input data consisted of a mixture of samples contaminated with the noises of test set A (bus, babble, car and pedestrian street) at several SNRs (-5 dB, 0 dB, 5 dB, 10 dB, 15 dB and 20 dB). Noises of test set B are reserved to evaluate the generalization ability of the DNN when exposed to unseen noises during the training phase (cafe, street, bus station and train station). 100 epochs per each RBM were used during the unsupervised pre-training phase while 1000 epochs were used for the backpropagation algorithm. A learning rate of 0.1 was established and the training dataset was divided into mini-batches (small subsets of training data) of 10 samples by following the recommendations in [16]. Preliminary experiments revealed that increasing L (the number of look-forward and look-backward frames) from zero to a few units provides a better performance. Finally, $L = 2$ was chosen (i.e. temporal window size of 5 frames). Thus, the input layer has 230 units or nodes, both hidden layers have 460 nodes and the output layer has 23 nodes. The DNN implementation was carried out by employing the MatLab toolbox referenced in [17].

TGI is performed using a 256-component GMM with diagonal covariance matrices. GMM training is performed by the expectation-maximization (EM) algorithm on the same dataset used for acoustic model training. The ETSI advanced front-end (AFE) [18], TGI with oracle masks and SPLICE (Stereo-based Piecewise LInear Compensation for Environments) [19] were also evaluated as a reference. SPLICE was included in order to compare with another stereo data technique (as it is the case of our proposal). For this last method, a 256-component GMM with diagonal covariance matrices was trained for each acoustic ambient (noise type & SNR). For a fair comparison, these acoustic ambients were the same as those used for the DNN training phase. Notice that a multi-environment soft-compensation scheme was followed, where every clean speech vector is estimated as a weighted sum of partial clean speech estimates obtained for every acoustic ambient seen during GMM training [20]. Finally, the baseline corresponds to the results obtained when the noisy speech features are employed. For all these cases, only the signals from the first channel were used.

4.2 Results

Figure 2 shows the results on word accuracy for both test sets as well as the averaged results across them. In Table 1, the word accuracy results averaged from -5 dB to 20 dB are shown for the different test sets and techniques evaluated. Similarly, percentages of wrong estimated mask bins with respect to oracle masks are included for our proposal and T-SNR. The DNN-based proposed system outperforms, for all the SNR values, AFE, SPLICE and T-SNR. In addition,

and according to the results for the test set B, we can observe that the DNN exhibits some generalization ability. An example of the TGI reconstruction of a dual-channel noisy utterance by using our DNN-based system can be seen in Figure 3.

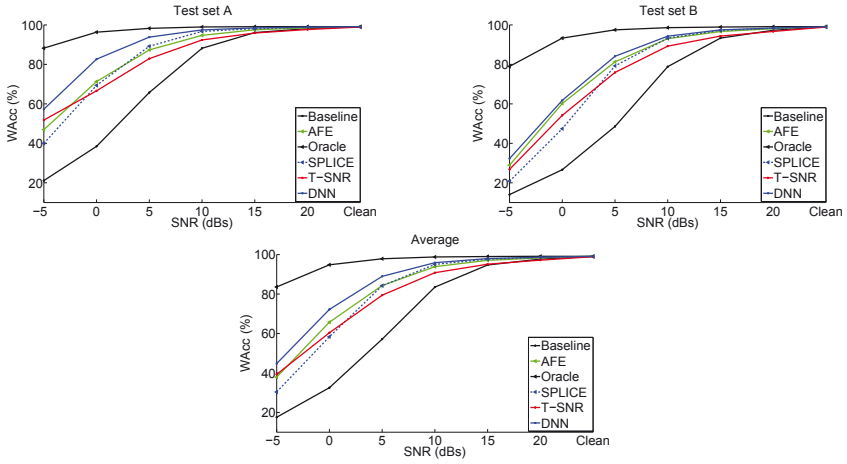

Fig. 2: WAcc results for the different techniques evaluated and for both test sets.

As mentioned in Subsection 4.1, the DNN could exploit temporal correlations by increasing the frame context through the number of look-forward and look-backward frames, L. The relative improvements in terms of word accuracy (average) over $L = 0$ were 1.43%, 3.17% and 3.47% for L values of 1, 2 and 3, respectively. As could be experimentally checked, the performance tends to saturate for $L = 2$ and greater values. Because of this fact, one can guess that the DNN is mainly exploiting the PLD between the first and second channels. Since most of the information required to provide a PLD-based estimate at frame t is close to that frame, the proposed DNN approach does not benefit of further increasing the length of the analysis window.

5 Conclusions and Future Work

In this paper we have proposed a new DNN-based system to estimate binary masks for robust speech recognition in the context of a smartphone with a dual-microphone used in a conversational position. The DNN has been able to take advantage of the dual-channel information, providing significant improvements on performance and again confirming the potential of this tool for speech recognition. Furthermore, one of the benefits of the DNN approach, with respect to other mask estimation techniques, is that no assumptions are made as well as it is able to learn complex non-linear dependencies between the input features and the target, thus overcoming the analytical modeling capabilities and allowing better performances.

As future work, an exhaustive search regarding the architecture and training configuration of the DNN should be carried out in order to further improve

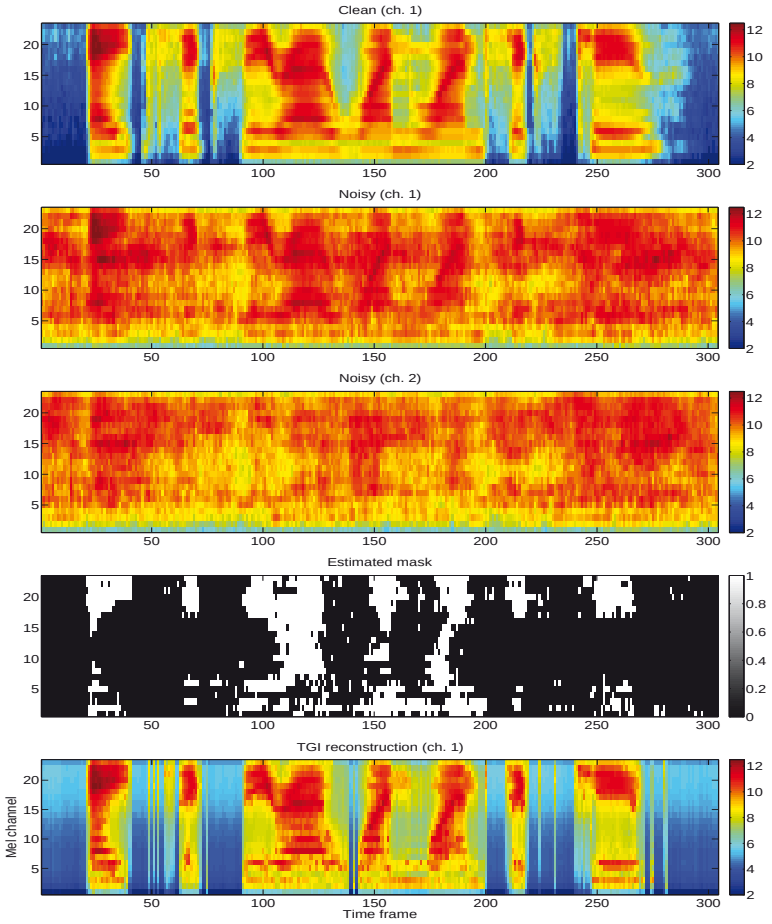

Fig. 3: Example of the TGI reconstruction of an utterance. All the spectrograms are in the log-Mel domain. From top to bottom: clean utterance (1st ch.), corrupted by bus noise at 0 dB (1st & 2nd chs.), mask estimated by the proposed DNN-based system and the resulting reconstruction (over the 1st ch.).

	WAcc (%)			Wrong mask bins (%)		
	Test A	Test B	Average	Test A	Test B	Average
Baseline	67.96	59.78	63.87	-	-	-
AFE	82.71	76.37	79.54	-	-	-
Oracle	96.67	94.41	95.54	0	0	0
SPLICE	82.03	72.72	77.38	-	-	-
T-SNR	81.21	72.87	77.04	17.97	19.89	18.93
DNN	88.10	78.07	83.08	10.07	16.19	13.13

Table 1: WAcc results and wrong estimated mask bin percentages for the different techniques evaluated. Results are averaged for SNRs from -5 dB to 20 dB.

the system performance. Also, the use of additional or different kind of features (e.g. pitch-based features) could be an interesting research topic. Finally, our objective is to extend this method in order to deal with a hands-free scenario. This scenario is more challenging, as the PLD assumptions are not completely valid since both speech and noise are in far field conditions.

References

1. López-Espejo I., et al.: "Feature Enhancement for Robust Speech Recognition on Smartphones with Dual-Microphone". *In: EUSIPCO*. Lisbon (2014)
2. Zhang, J., et al.: "A Fast Two-Microphone Noise Reduction Algorithm Based on Power Level Ratio for Mobile Phone". *In: ISCSLP*, pp. 206–209. Hong-Kong (2012)
3. Hinton, G. et al.: "Deep Neural Networks for Acoustic Modeling in Speech Recognition". *IEEE Signal Processing Magazine*, vol. 29, no. 6 (2012)
4. Seltzer, M.L., Yu, D., Wang, Y.: "An Investigation of Deep Neural Networks for Noise Robust Speech Recognition". *In: ICASSP*, pp. 7398–7402. Vancouver (2013)
5. Wang, Y., Wang, DL.: "Towards Scaling Up Classification-Based Speech Separation". *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 21, no. 7 (2013)
6. Narayanan, A., Wang, DL.: "Ideal Ratio Mask Estimation Using Deep Neural Networks for Robust Speech Recognition". *In: ICASSP*. Vancouver (2013)
7. Raj, B., Seltzer, M.L., Stern, R.M.: "Reconstruction of Missing Features for Robust Speech Recognition". *Speech Comm.*, vol. 48, no. 4, pp. 275–296 (2004)
8. González, J.A., Peinado, A.M., Ma, N., Gomez, A.M., Barker, J.: "MMSE-Based Missing-Feature Reconstruction with Temporal Modeling for Robust Speech Recognition". *IEEE Trans. on Audio, Speech and Language Proc.*, vol. 21, no. 3 (2013)
9. Cooke, M., et al.: "Robust Automatic Speech Recognition with Missing Data and Unreliable Acoustic Data". *Speech Communication* 34, pp. 267–285 (2001)
10. Pearce D., Hirsch, H.G.: "The Aurora Experimental Framework for the Performance Evaluation of Speech Recognition Systems Under Noisy Conditions". *In: ICSLP*. Beijing (2000)
11. Roweis, S.T.: "Factorial Models and Refiltering for Speech Separation and Denoising". *In: EUROSPEECH*, pp. 1009–1012. Geneva (2003)
12. Hinton, G., Salakhutdinov, R.: "Reducing the Dimensionality of Data with Neural Networks". *Science*, vol. 313, no. 5786 (2006)
13. Hinton, G.: "Training Products of Experts by Minimizing Contrastive Divergence". *Neural Computation*, vol. 14, pp. 1771–1800 (2002)
14. ETSI ES 201 108 - *Distributed speech recognition; Front-end feature extraction algorithm; Compression algorithms*
15. Ephraim, Y., Malah, D.: "Speech Enhancement Using a Minimum Mean-Square Error Short-Time Spectral Amplitude Estimator". *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. ASSP-32, no. 6, pp. 1109–1121 (1984)
16. Hinton, G.: "A Practical Guide to Training Restricted Boltzmann Machines". *UTML TR 2010-003* (2010)
17. Tanaka, M.: *Deep Neural Network Toolbox for MatLab* (2013)
18. ETSI ES 202 050 - *Distributed speech recognition; Advanced front-end feature extraction algorithm; Compression algorithms*
19. Deng, L. et al.: "Large-Vocabulary Speech Recognition Under Adverse Acoustic Environments". *In: ICSLP*, pp. 806–809. Beijing (2000)
20. González, J.A., et al.: "Efficient MMSE Estimation and Uncertainty Processing for Multienvironment Robust Speech Recognition". *IEEE Trans. on Audio, Speech, and Language Proc.*, vol. 19, no. 5, pp. 1206–1220 (2011)