Analysis of CNN-based Speech Recognition System using Raw Speech as Input

Dimitri Palaz[1,2], *Mathew Magimai.-Doss*[1], *Ronan Collobert*[3,1]

[1]Idiap Research Institute, Martigny, Switzerland
[2]Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland
[3]Facebook AI Research, Menlo Park, CA, USA
{dimitri.palaz, mathew}@idiap.ch, ronan@collobert.com

Abstract

Automatic speech recognition systems typically model the relationship between the acoustic speech signal and the phones in two separate steps: feature extraction and classifier training. In our recent works, we have shown that, in the framework of convolutional neural networks (CNN), the relationship between the raw speech signal and the phones can be directly modeled and ASR systems competitive to standard approach can be built. In this paper, we first analyze and show that, between the first two convolutional layers, the CNN learns (in parts) and models the phone-specific spectral envelope information of 2-4 ms speech. Given that we show that the CNN-based approach yields ASR trends similar to standard short-term spectral based ASR system under mismatched (noisy) conditions, with the CNN-based approach being more robust.

Index Terms: automatic speech recognition, convolutional neural networks, raw signal, robust speech recognition.

10.21437/Interspeech.2015-3

1. Introduction

State-of-the-art automatic speech recognition (ASR) systems typically model the relationship between the acoustic speech signal and the phones in two separate steps, which are optimized in an independent manner [1]. In a first step, the speech signal is transformed into features, usually composed of a dimensionality reduction phase and an information selection phase, based on the task-specific knowledge of the phenomena. These two phases have been carefully hand-crafted, leading to state-of-the-art features such as Mel frequency cepstral coefficients (MFCCs) or perceptual linear prediction cepstral features (PLPs). In a second step, the likelihood of subword units such as, phonemes is estimated using generative models or discriminative models.

In recent years, in the hybrid HMM/ANN framework [1], there has been growing interests in using "intermediate" representations instead of conventional features, such as cepstral-based features, as input for neural networks-based systems. ANNs with deep learning architectures, more precisely, deep neural networks (DNNs) [2, 3], which can yield better system than a single hidden layer MLP have been proposed to address various aspects of acoustic modeling. More specifically, use of context-dependent phonemes [4, 5]; use of spectral features as opposed to cepstral features [6, 7]; CNN-based system with Mel filter bank energies as input [8, 9, 10]; combination of different features [11], to name a few. Features learning from the raw speech signal using neural networks-based systems has also been investigated in [12]. In all these approaches, the features extraction step and the acoustic modeling step are trained independently. More recently, neural network-based systems where the features and the model are trained jointly have been proposed. CNN-based system taking power spectrum as input has been proposed in [13]. Using temporal raw speech directly as input has been proposed in the context of DNNs [14] and in the context of end-to-end sequence-discriminative training of CNNs [15].

In our recent studies [16, 17], it was shown that it is possible to estimate phoneme class conditional probabilities by using raw speech signal as input to convolutional neural networks [18] (CNNs). On phoneme recognition task and on continuous speech recognition task, we showed that the system is able to learn features from the raw speech signal, and yields performance similar or better than conventional ANN-based system that takes cepstral features as input. We also showed that the first convolutional layer of the network can be seen as a set of matching filters, processing the speech signal at a sub-segmental level, 2-4 ms speech. We showed that these filters respond to different frequency bandwidths [16], and that they show some level of invariance across databases [17].

In this paper, we first analyze the CNN to understand the speech information that is modeled between the first two convolution layers. To that end, we present a method to compute the mean frequency responses of the filters in the first convolution layer that match to the specific inputs representing vowels. Our studies on TIMIT task indicate that the mean frequency response tends to model the envelope of the sub-segmental (2-4 ms) speech signal. We then present a study to evaluate the susceptibility of the CNN-based system to mismatched conditions. This is an open problem in systems trained in a data-driven manner. We investigate this aspect on two tasks, namely, TIMIT phoneme recognition task and Aurora2 connected word recognition task. Our studies show that the performance of the CNN-based system degrades with the decrease in signal-to-noise ratio (SNR) like in a standard spectral feature based system. However, when compared to the spectral feature based system, the CNN-based system using raw speech signal as input yields better performance.

The remainder of the paper is organized as follows. Section 2 presents the architecture of the network. Section 3 presents the experimental setup. Section 4 presents the network analysis and Section 5 presents the noise study. Finally, Section 6 summarizes and concludes the paper.

2. Convolutional Neural Networks

We present briefly the architecture of the CNN-based system. More details can be found in [17].

All research was conducted at the Idiap Research Institute, before Ronan Collobert joined Facebook AI Research.

Copyright © 2015 ISCA

PCT Exhibit 1019 - Page 1 of 5

2.1. Architecture

The convolutional neural network is given a sequence of raw input signal, split into frames, and outputs a score for each classes, for each frame. The network architecture is composed of several filter stages, followed by a classification stage. A filter stage involves a convolutional layer, followed by a temporal max-pooling layer and a non-linearity ($\tanh()$). Our optimal architecture included three filter stages. Processed signals coming out of these stages are fed to a classification stage, which in our case is a multi-layer perceptron, with one hidden layer. It outputs the conditional probabilities $p(i|x)$ for each class i, for each frame x using a SoftMax layer [19]. The network is trained under the cross-entropy criterion, maximized using the stochastic gradient ascent algorithm [20].

2.2. Convolutional layer

While "classical" linear layers in standard MLPs accept a fixed-size input vector, a convolution layer is assumed to be fed with a sequence of T vectors/frames: $X = \{x^1 \quad x^2 \quad \ldots \quad x^T\}$. A convolutional layer applies the same linear transformation over each successive (or interspaced by dW frames) windows of kW frames. For example, the transformation at frame t is formally written as:

$$M \begin{pmatrix} x^{t-(kW-1)/2} \\ \vdots \\ x^{t+(kW-1)/2} \end{pmatrix}, \tag{1}$$

where M is a $d_{out} \times d_{in}$ matrix of parameters. In other words, d_{out} filters (rows of the matrix M) are applied to the input sequence.

2.3. Max-pooling layer

These kind of layers perform local temporal $\max$ operations over an input sequence. More formally, the transformation at frame t is written as:

$$\max_{t-(kW-1)/2 \leq s \leq t+(kW-1)/2} x_s^d \qquad \forall d \tag{2}$$

with x being the input, kW the kernel width and d the dimension.

3. Experimental Setup

3.1. Databases

The TIMIT acoustic-phonetic corpus consists of 3,696 training utterances (sampled at 16kHz) from 462 speakers, excluding the SA sentences. The cross-validation set consists of 400 utterances from 50 speakers. The core test set is used to report the results. It contains 192 utterances from 24 speakers, excluding the validation set. The 61 hand labeled phonetic symbols are mapped to 39 phonemes with an additional garbage class, as presented in [21]. For the noise studies, the utterances from the TIMIT corpus are corrupted by noises from the NoiseX-92 corpus [22]. The core testset is corrupted with the speech, F-16 and factory noises, at SNR level between 0dB and 30 dB. We also present a multi-conditional training study, where the trainset is randomly split in 20 subsets, each one containing 184 utterances. The 20 subsets represent 4 noise types (car, operation, lynx, minigun) different from the testset, at 5 different SNRs (20dB, 15dB, 10dB, 5dB and clean). The corrupted utterances are obtained using the FaNT tool [23].

The Aurora2 corpus [24] is a connected digit corpus which contains 8,440 sentences of clean and multi-condition training data and 70,070 sentences of clean and noisy test data. We report the results on test A and test B, composed of 10 different noises at 7 different noise levels (clean, 20dB, 15dB, 10dB, 5dB, 0dB, -5dB), totaling 70 different test scenarios, each containing 1,001 sentences. The alignment is obtained using the HTK-based HMM/GMM system provided along with the database. It consists of whole word HMM models with 16 states per word to model the digits. The states are connected in a simple left-to-right fashion. The number of state is 179. The language model provided by the corpus is used.

3.2. Tasks

For the connected word recognition task on the Aurora2 corpus, the CNN-based system is used to compute the posterior probabilities of word states. The decoder is an HMM, modeling words. The scaled likelihoods are estimated by dividing the posterior probability by the prior probability of each class, estimated by counting on the training set. The hyper parameters such as, language scaling factor and the word insertion penalty are determined on the validation set.

For the phoneme recognition task on the TIMIT corpus, the CNN-based system is used to estimate phoneme class conditional probabilities. The decoder is a standard HMM decoder, with constrained duration of 3 states, and considering all phoneme equally probable. We do not use a phonetic language model. 39 classes are used.

3.3. Features Input

Raw features are simply composed of a window of the temporal speech signal (hence, $d_{in} = 1$ for the first convolutional layer). The window is normalized such that it has zero mean and unit variance.

We also performed baseline experiments with MFCC as input features. They are computed (with HTK [25]) using a 25 ms Hamming window on the speech signal, with a shift of 10 ms. The signal is represented using 12th-order coefficients (without the zeroth coefficient) and the logarithmic frame energy, along with their first and second derivatives, computed on a 9 frames context.

3.4. Baseline systems

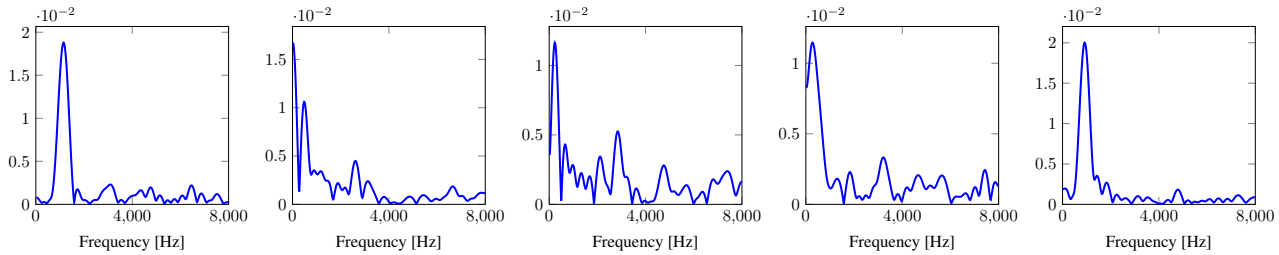
We compare our approach with the standard HMM/ANN system using cepstral features. We train an ANN with one single hidden layer, referred to as *ANN*. The input to the ANNs are MFCC features with several frames of preceding and following context. We do not pre-train the network.

3.5. Networks hyper-parameters

The hyper-parameters of the network are: the input window size w_{in}, corresponding to the context taken along with each example, the kernel width of the first convolution, expressed in samples, the kernel width kW_n, the shift dW_n and the number of filters d_n of the other n^{th} convolution layers, the pooling width kW_{mp} of maxpooling layers and the hidden layer width. They are tuned by early-stopping on the validation set.

The architecture is composed of 3 convolutional and max-pooling layers and 1 hidden layer. The best performance on TIMIT was found with: 50 samples kernel width for the first convolution, 310 ms of context, 5 frames kernel width, 80, 60 and 60 filters, 500 hidden units and 3 pooling width. The ANN

PCT Exhibit 1019 - Page 2 of 5



(a) $\lambda_1 = 0.28$ (b) $\lambda_2 = 0.23$ (c) $\lambda_3 = 0.22$ (d) $\lambda_4 = 0.17$ (e) $\lambda_5 = 0.08$

Figure 1: *Illustration of the five most firing filters, with their proportion factor, for the center frame of phoneme* $/iy/$.

baseline uses 500 nodes for the hidden layer. On Aurora2, 10-fold cross-validation was used for tuning the hyper-parameters. The best performance was found with: 50 samples kernel width for the first convolution, 310 ms of context, 7 frames kernel width, 80, 60 and 60 filters, 500 hidden units and 3 pooling width. The ANN baseline uses 500 nodes for the hidden layer. The experiments were implemented using the *torch7* toolbox [26].

4. Filters Analysis

In most of the recent convolutional neural networks-based systems proposed in the literature, the input features are either conventional cepstral-based features [5] or spectral-based representations, such as Mel filterbank coefficients [10]. These features are usually computed on a 25 ms window, with a shift of 10 ms. The key difference in our system is that the CNN takes the temporal raw speech directly as input. Thus, the first convolutional layer should act as a filterbank, learned in a data-driven manner. The kernel width of this first convolutional layer, representing the length of the temporal input window, was selected empirically, on the validation set. The best performance was found with a very short window, around 2-4 ms speech, with a shift of 0.6 ms, which is ten times shorter than in conventional cepstral-based features processing. The filters learned by the first convolution can be seen as matching filters. In our previous studies, we showed that they respond to various frequency bandwidths [16], and that they show some level of invariance across databases [17].

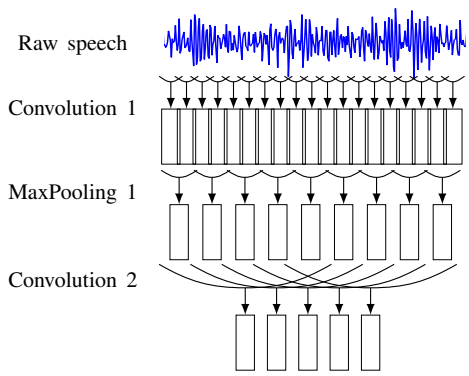


Figure 2: *Detailed view of the first two stages of the CNN. The outputs of the first convolution are combined in the second convolution.*

As illustrated in Figure 2, in the proposed architecture the outputs of the first convolution layer which consists of a bank of matched filters are linearly combined after maxpooling operation. We take a simplified view of this process where a speech signal corresponding to a sound is passed through a bank of linear time invariant filters and the outputs are then combined linearly. In such a case, we can visualize the total frequency response of the filter banks after linear combination as a sum of the frequency responses of the filters that are firing/matching to the spectral characteristics of the sound. Using this simplified view, we studied the average spectral characteristics of the vowel sound that is being modeled in the following manner:

- The center frame for a given vowel in a sequence is selected and forwarded to the network.
- We determine which filters are firing the most for each frame by taking the argmax.
- The first two operations are repeated over the whole validation data set keeping track of the number of times n_i the filter i is triggered.
- The frequency response F_i of each filter i is computed by taking the magnitude of the Fourier Transform and normalizing it.
- Finally, the mean frequency response for a given vowel f_{vowel} is computed by adding the frequency responses of the five most firing filters, weighted by a proportion factor λ_i, which is given by the number of time the filter i is triggered, normalized by the total number of appearances: $\lambda_i = \frac{n_i}{\sum_j n_j}$.

$$f_{vowel} = \sum_i \lambda_i F_i \tag{3}$$

The number of firing filters was set to five, because it represents most of the contribution to the filters output. An illustration of the five most firing filters for one phoneme $/iy/$ is provided in Figure 1.

The frequency responses of selected vowels, computed on the validation set of TIMIT are presented in Figure 3. The ripples present in the plots can be attributed to the fact that these filters are learned on sub-segmental speech signal, i.e. a duration of 2-4 ms. It could be observed that the average frequency response is like a smooth spectral envelope. Given that we could hypothesize that in the mismatched (noisy) conditions the CNN-based system should have a trend similar to standard cepstral features, which tends to model spectral envelope information (of about 25ms speech signal). We ascertain this aspect in the next section.

PCT Exhibit 1019 - Page 3 of 5

Table 1: *Results on the Aurora tests A and B, given in Word Recognition Rate (WRR), averaged over the four noises. Both systems have 250k parameters.*

	Test A							Test B						
SNR [dB]	clean	20	15	10	5	0	-5	clean	20	15	10	5	0	-5
	Clean Training													
ANN	96.9	86.1	74.1	51.4	25.5	13.9	10.1	97.4	87.3	77.8	59.8	33.5	15.0	8.9
CNN	97.3	88.3	76.1	53.0	24.7	11.2	8.0	97.2	90.4	83.2	64.9	38.7	19.1	10.1
	Multi-conditional Training													
ANN	92.1	91.6	89.0	83.4	70.0	38.2	14.5	92.1	85.1	80.9	73.8	59.7	34.1	14.5
CNN	97.6	97.4	96.6	93.9	84.8	55.1	19.5	97.6	94.8	93.4	89.0	77.4	48.0	18.7

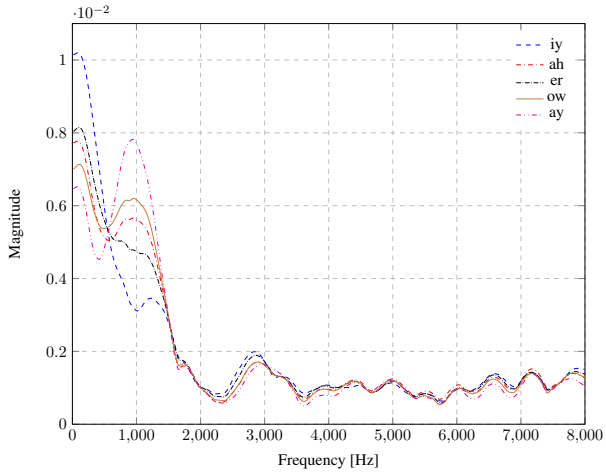


Figure 3: *Mean frequency responses on the TIMIT validation set for phonemes* $/iy/, /ah/, /er/, /ow/$ *and* $/ay/$.

5. Noise Studies

DNN and CNN based systems had been shown to yield state-of-the-art results in ASR. However, this data-driven approach could raise questions about the susceptibility of the system to mismatched conditions. In this section, we present noise robustness studies on two tasks: connected word recognition task on the Aurora2 corpus and phoneme recognition task on the TIMIT corpus. For a fair comparison, in these studies we do not perform any further normalization on the MFCC features, such as cepstral mean normalization. The reason being that such normalizations could be envisioned at signal level as filtering operations. Studying in detail these aspects is open for further research and as discussed later in Section 6, is part of future work.

5.1. Word recognition study

Table 1 presents the results on Aurora2 corpus. It can be observed that in both clean condition training and multi-condition training, the CNN-based system outperforms the ANN-based system. This can be seen prominently in the case of multi-condition training. The performance of the CNN-based system is similar to the first reported system on Aurora2 corpus [24]. The weak performance of the ANN-based system could be attributed to lack of feature normalization and low capacity.

Table 2: *Results for the clean and multi-condition training on the TIMIT core testset, given in PRR.*

SNR [dB]	ANN		CNN	
Training	clean	multi	clean	multi
30dB	52.5	54.3	65.5	66.8
25dB	46.7	50.8	59.7	64.8
20dB	40.3	46.6	50.5	60.8
15dB	32.7	41.1	39.1	53.5
10dB	26.1	34.2	27.8	42.8
5dB	21.2	26.4	18.3	30.8
0dB	17.4	20.2	9.9	21.4

5.2. Phoneme recognition study

Table 2 presents the results on TIMIT corpus for the baseline and the CNN-based system, expressed in term of Phoneme Recognition Rate (PRR). In the case of clean training, it can be observed that the CNN-based system is slightly more robust than the baseline. However, the performance of the CNN-based system degrades at very low SNR level compared to the baseline. This could be due to the small amount of variability available in the TIMIT corpus, leading to filters which do not generalize very well to mismatched conditions, as already shown in [17]. In the case of multi-conditional training, it could be observed that the CNN-based system is consistently more robust than the baseline. Overall, these results indicate that the CNN-based system follows a similar trend to baseline system using cepstral-based features as input.

6. Summary and Future Work

In summary, the filter analysis study indicates that the features learned between the first two convolution layers of the CNN tends to model the spectral envelope of sub-segmental speech signal. The noise robust ASR studies shows that these features are susceptible to noise but not to the same extent as MFCC features (without any normalization). To improve the robustness of the CNN-based system, we can exploit the parallel between time domain processing and frequency domain processing. For instance, we could improve the robustness by filtering the speech signal using the Wiener filter technique in the Aurora Advanced Front End [27] and then feeding it into the CNN. Our future work will investigate these aspects and will study in comparison with noise robust spectral-based feature extraction.

7. Acknowledgements

This work was supported by the HASLER foundation (www.haslerstiftung.ch) through the grant "Universal Spoken Term Detection with Deep Learning" (DeepSTD).

PCT Exhibit 1019 - Page 4 of 5

8. References

[1] H. Bourlard and N. Morgan, *Connectionist speech recognition: a hybrid approach*. Springer, 1994, vol. 247.

[2] G. E. Hinton, S. Osindero, and Y. W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[3] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, and T. N. Sainath, "Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, p. 8297, 2012.

[4] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks," in *Proc. of Interspeech*, 2011, pp. 437–440.

[5] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, p. 3042, 2012.

[6] A. Mohamed, G. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 14 –22, jan. 2012.

[7] H. Lee, P. Pham, Y. Largman, and A. Y. Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks," in *Advances in Neural Information Processing Systems 22*, 2009, pp. 1096–1104.

[8] O. Abdel-Hamid, A. Mohamed, H. Jiang, and G. Penn, "Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition," in *Proc. of ICASSP*, 2012, pp. 4277–4280.

[9] T. N. Sainath, A. Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep convolutional neural networks for lvcsr," in *Proc. of ICASSP*, 2013, pp. 8614–8618.

[10] P. Swietojanski, A. Ghoshal, and S. Renals, "Convolutional neural networks for distant speech recognition," *Signal Processing Letters, IEEE*, vol. 21, no. 9, pp. 1120–1124, September 2014.

[11] E. Bocchieri and D. Dimitriadis, "Investigating deep neural network based transforms of robust audio features for lvcsr," in *Proc. of ICASSP*, 2013, pp. 6709–6713.

[12] N. Jaitly and G. Hinton, "Learning a better representation of speech soundwaves using restricted boltzmann machines," in *Proc. of ICASSP*, 2011, pp. 5884–5887.

[13] T. Sainath, B. Kingsbury, A.-R. Mohamed, and B. Ramabhadran, "Learning filter banks within a deep neural network framework," in *Proc. of ASRU*, Dec. 2013, pp. 297–302.

[14] Z. Tüske, P. Golik, R. Schlüter, and H. Ney, "Acoustic modeling with deep neural networks using raw time signal for lvcsr," in *Proc. of Interspeech*, Singapore, Sep. 2014, pp. 890–894.

[15] D. Palaz, R. Collobert, and M. Magimai. -Doss, "End-to-end Phoneme Sequence Recognition using Convolutional Neural Networks," *ArXiv e-prints*, Dec. 2013.

[16] D. Palaz, R. Collobert, and M. Magimai.-Doss, "Estimating phoneme class conditional probabilities from raw speech signal using convolutional neural networks," in *Proc. of Interspeech*, 2013.

[17] D. Palaz, M. Magimai.-Doss, and R. Collobert, "Convolutional neural networks-based continuous speech recognition using raw speech signal," in *Proc. of ICASSP*, April 2015.

[18] Y. LeCun, "Generalization and network design strategies," in *Connectionism in Perspective*, R. Pfeifer, Z. Schreter, F. Fogelman, and L. Steels, Eds. Zurich, Switzerland: Elsevier, 1989.

[19] J. Bridle, "Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition," in *Neuro-computing: Algorithms, Architectures and Applications*, 1990, pp. 227–236.

[20] L. Bottou, "Stochastic gradient learning in neural networks," in *Proceedings of Neuro-Nmes 91*. Nimes, France: EC2, 1991.

[21] K. F. Lee and H. W. Hon, "Speaker-independent phone recognition using hidden markov models," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37, no. 11, pp. 1641–1648, 1989.

[22] A. Varga and H. J. Steeneken, "Assessment for automatic speech recognition: Ii. noisex-92: A database and an experiment to study the effect of additive noise on speech recognition systems," *Speech communication*, vol. 12, no. 3, pp. 247–251, 1993.

[23] H.-G. Hirsch, "Fant-filtering and noise adding tool," 2005.

[24] H.-G. Hirsch and D. Pearce, "The Aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions," in *ASR2000-Automatic Speech Recognition: Challenges for the new Millenium ISCA Tutorial and Research Workshop (ITRW)*, 2000.

[25] S. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, "The htk book," *Cambridge University Engineering Department*, vol. 3, 2002.

[26] R. Collobert, K. Kavukcuoglu, and C. Farabet, "Torch7: A matlab-like environment for machine learning," in *BigLearn, NIPS Workshop*, 2011.

[27] H.-G. Hirsch and D. Pearce, "Applying the advanced ETSI frontend to the aurora-2 task," Tech. Rep., 2006, version 1.1.

PCT Exhibit 1019 - Page 5 of 5