ROBUST AUTOMATIC RECOGNITION OF SPEECH WITH BACKGROUND MUSIC

Jiri Malek, Jindrich Zdansky and Petr Cerva

Faculty of Mechatronics, Informatics, and Interdisciplinary Studies, Technical University of Liberec,
Studentská 2, 461 17 Liberec, Czech Republic.

jiri.malek@tul.cz

ABSTRACT

This paper addresses the task of Automatic Speech Recognition (ASR) with music in the background, where the accuracy of recognition may deteriorate significantly. To improve the robustness of ASR in this task, e.g. for broadcast news transcription or subtitles creation, we adopt two approaches: 1) multi-condition training of the acoustic models and 2) denoising autoencoders followed by acoustic model training on the preprocessed data. In the latter case, two types of autoencoders are considered: the fully connected and the convolutional network.

Presented experimental results show that all the investigated techniques are able to improve the recognition of speech distorted by music significantly. For example, in the case of artificial mixtures of speech and electronic music (low Signal-to-Noise Ratio (SNR) of 0 dB), we achieved absolute improvement of accuracy by 35.8%. For real-world broadcast news and a high SNR (about 10 dB), we achieved improvement by 2.4%. The important advantage of the studied approaches is that they do not deteriorate the accuracy in scenarios with clean speech (the decrease is about 1%).

Index Terms— Robust recognition, background music, feature enhancement, denoising autoencoder, multi-condition training.

1. INTRODUCTION

Nowadays, the research in automatic speech recognition (ASR) is focused on robustness of the performance with respect to difficult environmental conditions. These include, e.g., distant microphones, concurrent speech or background interference. In some applications, such as online $24/7$ monitoring of broadcast media, one of the most often encountered background interferences is music.

Two basic approaches exist which introduce the robustness to background interference into ASR. The first approach consists in utilization of the *multi-condition training* of the acoustic models. Here, the distorted speech signals are included in the training set, i.e., the model incorporates knowledge on the possible interference. The disadvantage here is the difficulty of including all possible noise types in the training data, which are later encountered within test environments [1]. Considering environmental noise, this approach was reported to obtain high performance in [2]. Besides additive noise, this technique was demonstrated to be beneficial for reverberated speech in [3, 4].

The other approach is to perform input speech (or feature) preprocessing, in order to separate the speech from the interference. The ASR is performed on the enhanced signal / features. An efficient speech separation can be achieved using *denoising autoencoders*, such as those proposed for environmental noises in [5]. The benefits of autoencoders for ASR was shown in [6]. Here, the car and factory noises were considered. The performance of multichannel autoencoders was demonstrated on the Chime-2 challenge [7] datasets in [8].

The front-end preprocessing usually introduces distortions into enhanced data, which are not observed by the acoustic model trained on the clean data. To mitigate, the enhancement is usually applied on both training and test data and a new acoustic model is trained on the enhanced dataset. This is shown for environmental noises, e.g., in [9].

When comparing the two above-mentioned approaches, some studies get superior results using front-end denoising [10], while others favor the multi-condition training [2].

Focusing specifically on separation of background music, Nonnegative Matrix Factorization (NMF, [11, 12, 13]) is often utilized. A direct application to robust ASR was proposed in [14], introducing a probabilistic approach based on a catalog of prepared music samples. The utilization of autoencoders for music-robust ASR was proposed in [15]. That paper compares utilization of the fully connected and convolutional networks. It demonstrates that the autoencoder is capable of learning features to discriminate between music and speech. Moreover, the method is shown to be largely language-independent.

Relation to prior work: The above mentioned techniques are usually employed in the context of environmental background noise. In this paper, we specifically focus on background music. We extend the analysis of the denoising autoencoders from [15] and compare the autoencoders directly to multi-condition training [2]. We aim to determine more suitable approach for music-robust ASR. Compared to [15], where autoencoders were trained for a specific musical piece, we train more general models using broad range of artificial mixtures of speech and various music. Considered genres range from classical music to electronic tunes. We study the robustness of the models with respect to unseen test conditions (varied music genres and energy of background music) and confirm the functionality on real-world radio broadcast shows.

2. PROBLEM FORMULATION AND DATA DESCRIPTION

We focus on robustness of ASR to music present in the background of the speech. All of the considered training data are generated artificially, by summation of the speech and music signal. We analyze different scenarios, with respect to average Signal-to-Noise Ratio (SNR) and the included music genres.

We consider a Large Vocabulary Continuous Speech Recognition (LVCSR) task. Due to the data most readily available to us, we focus on Czech language, without any loss of generality to the investigated problems. Our training set consists of 132 hours of Czech speech.

This work was supported by the Technology Agency of the Czech Republic (Project No. TA04010199).

Table 1. Setup of the training set for multi-style acoustic models and respective autoencoders

Dataset (genre)	N	SNR levels	Music styles included
Piano 1	3	clean, $10, 5, 0$	Classical piano
Piano 2	7	clean, $10, 5, 0, -5, -10, -15, -20$	Classical piano
Electronic	3	clean, $10, 5, 0$	Ambient, dance, down-tempo, chillout or idm

Table 2. Setup of the artificially generated test sets

Dataset (genre)	SNR levels	Music styles included
Clean	clean	None
Test:Piano	$10, 0, -10, -20$	Classical piano
Test:Violin	$10, 0, -10, -20$	Piano and violin compositions
Test:Electro	$10, 5, 0, -5$	Ambient, dance, down-tempo, chillout or idm

The music we utilize in generation of the training dataset originates in a database of free music tracks at the Free Music Archive [16]. We use the *Piano* tracks (duration 33 minutes) and a broad set of *Electronic* music (667 minutes). The latter set consists of genres such as ambient, dance, down-tempo, chillout or idm. The piano music provides the easier scenario; the music covers partly different frequency bands than the speech, with only a single instrument present. The mixtures are intelligible even for very low SNR. As a more difficult scenario, we select the electronic music, because it resembles the background music of the TV shows.

3. PROPOSED ROBUSTNESS-INTRODUCING TECHNIQUES

We consider two techniques: 1) the multi-condition training of the acoustic model; and 2) the removal of background music using a denoising autoencoder and subsequent acoustic model training on the processed data. We consider two types of autoencoders: a fully connected and the convolutional network.

The configuration of hyper-parameters for all acoustic models corresponds to the best performance in preliminary experiments with undistorted data. The configuration for autoencoders was selected based on preliminary experiments with a fully connected network on dataset Piano 1 (see Table 1).

3.1. General acoustic model structure

Apart from the training data, the acoustic models for both approaches are similar, based on Hidden Markov Model - Deep Neural Network (HMM-DNN) hybrid architecture [17]. The underlying Gaussian Mixture Model is trained as context dependent, speaker independent and contains 2219 physical states.

For feature extraction, filter bank coefficients [18] are computed using 25 ms frames of signal with frame shifts of 10 ms. To normalize the features, Cepstral Mean Subtraction ([19], CMS) with a floating window of 1 s is employed. The input for DNN consists of 11 consecutive feature vectors, 5 preceding and 5 following the current frame.

The Torch library [20] is used for the DNN training, which has a fixed duration of 20 epochs. The networks are fully connected and have feed-forward structure with 5 hidden layers. The activation function is ReLU . Each hidden layer consists of 768 units. The mini-batch size is 1024 input vectors and the learning rate is 0.08.

As our *baseline model*, we consider a single-style acoustic model, trained on an undistorted instance of the training dataset.

3.2. Multi-condition training of acoustic model

To train the multi-condition model, we prepare each dataset in the following way. We select N desired SNR levels (details are provided in Table 1). Subsequently, we split the speech corpus into $N + 1$ parts. The first part is left undistorted. To all other parts we add corresponding music, scaled to the predefined average SNR level. The average SNR is computed per one file of speech recordings, which usually corresponds to about two sentences (about 20 words).

We study three different multi-style models, based on Piano and Electronic music in the background of the training speech; details are provided in Table 1. The two piano-based training sets differ in energy levels of the noise; we aim to study influence of unseen noise-intensity conditions. In the experiments, we will denote the multi-condition models by notation MC:Train set, e.g., MC:Piano 1.

3.3. Fully connected feed-forward denoising autoencoder

Our fully connected denoising autoencoder is a feed-forward deep neural network, where all neurons in the lower hidden layer are connected to all neurons in the higher layer. It accepts distorted features at its input layer. The output is an estimate of clean speech features. During the training stage, the autoencoder requires pairs of corrupted and undistorted utterances. In this work, the undistorted data consists of 132 hours of training Czech speech (similar to acoustic model training) and its distorted counterpart is generated artificially, in a manner described in Section 3.2

The autoencoder is trained using the filter bank features (similar to acoustic model training). The training minimizes the mean square distance between the distorted input and the clean target. This criterion is sensitive to scaling, thus we normalize both training and test data (each feature separately) to zero mean and unitary variance. The same normalization values are utilized later in the test phase.

Our autoencoder is constituted of three hidden layers, with 1024 neurons in each layer. We use the ReLU activation function, a learning rate of 0.03 and a batch size of 512 samples. The training is always stopped after 20 epochs.

We denote models trained on the data processed using a fully connected autoencoder by the notation AE:Train set, e.g., AE:Piano 1; the setups are summarized in Table 1.

3.4. Convolutional denoising autoencoder

The convolutional autoencoder represents another network topology, in which the neurons in the higher hidden layer have connections to only several neurons in the lower layer. This model has been

proposed for acoustic modeling and feature extraction in ASR context in [21, 22]. Its advantages over a fully connected network include: easier modeling of translational variance within speech signals, which exist due to different speaking styles [23], and modeling of local correlations within spectral representations of the speech.

We denote models trained on data processed by convolutional autoencoder by the notation CAE:Train set, e.g., CAE:Piano 1; the setups are summarized in Table 1.

The input feature vectors, targets, the training dataset, the activation functions and optimizing criterion remain the same as for the AE. The topology of the two autoencoders differ in two aspects: 1) the input layer; and 2) the substitution of the first hidden layer of the AE by two convolutional layers in CAE (the number of hidden units remains constant).

The input of CAE consists of 11 feature maps, which correspond to 11 following frames in the input feature vector. Each feature map is 39 elements long (number of filter bank features for a single frame). The convolutional kernel in both layers is of size 5×1 (i.e., the weights are shared in frequency only). Between the convolutional layers, there is a max-pooling layer; we use max-pooling by factor of 3. The first hidden layer has 13 feature maps (i.e., 13×39 hidden units) and the second one 39 (i.e., 39×13 hidden units).

4. EXPERIMENTS

We report the results of our experiments via recognition accuracy [%]; all improvements are stated as absolute.

4.1. Description of the test set

We consider two types of data involved in our experiments: 1) The artificially generated data; and 2) the real-world speech recordings with music in the background.

The generated datasets share common test speech recordings. The set has a duration of 2 hours and 44 minutes (13622 words) and it consists of dictated texts, recorded in a silent environment via close-talk microphone. To the speech, we add piano tracks (8 minutes), piano and violin compositions (2 hours and 24 minutes) and electronic music (40 minutes) with various SNR levels. We concatenate the available music as is necessary, to create background for the whole test-speech set. For each scenario with a specific music type and SNR level, we replicate the whole test dataset. Details of the resulting datasets are summarized in Table 2. The piano and violin compositions represent mismatched training-test conditions for all variants of acoustic models. For Test:Electro dataset, the very low SNR levels are omitted, because the scenario is too complicated then (unintelligible even for human listener).

The real-world dataset was created by the authors solely for the purposes of this paper and consists of 17 minutes and 22 seconds of speech (2222 words), recorded from a digital broadcast of a local radio station (Radiožurnál [24]). The speech comes from several summaries, which are given at the beginning of the news program. A track of electronic music is present in the background. We estimate the average SNR level at about 10 dB.

4.2. Employed recognition engine

We use our own ASR system; its core is formed by a one-pass speech decoder performing a time-synchronous Viterbi search.

The linguistic part of the system consists of a lexicon and a language model. We use two types of language models: 1) A model originating from newspaper texts for the scenarios with simulated data; and 2) A model originating from broadcast transcriptions for the scenario with real-world data.

The lexicon contains 550k entries (word forms and multi-word collocations) that were observed most frequently in the corpora covering newspaper texts. The employed Language Model (LM) is based on N-grams. Due to the very large vocabulary size, the system uses bigrams. Our supplementary experiments showed that the bigram structure of the language model results in the best ASR performance with reasonable computational demands.

4.3. Matched training-test conditions

Here, we discuss performance achieved in scenarios with music genres and SNR levels available during training. See Tables 3 and 4, numbers styled in bold italics.

The baseline model achieves recognition accuracy of 85.0% on undistorted data. For this case, the robust models achieve comparable results (degradation by $0.1 - 1.1$ %), i.e., the robustness on distorted data is not achieved at the cost of worse performance on clean speech.

Within the *Test:Piano,* the accuracy of the baseline model deteriorates with increasing amounts of added background music. The decrease is 16.9% for the SNR level at 0 dB. All considered robust techniques achieve much lower degradation ($1.3 - 2.2\%$). Comparing MC models and AE/CAE models, their results are comparable. The performance of more general models Piano 2 (trained on a broader range of SNR levels) is comparable to the more specific Piano 1.

In the *Test:Electro* scenario, the accuracy of the baseline model deteriorates even more noticeably. The decrease is 46.1% for the SNR level at 0 dB. The robust techniques are able to improve this result by up to 35.8%. The model MC:Electronic achieves significantly better results than AE:Electronic and CAE:Electronic, especially at lower SNR levels (9.1% and 12.3% at SNR 0 dB, respectively). We hypothesize that the autoencoders require more training data, when complex multi-instrumental music is considered. Considering higher number of hidden units in AE/CAE for this scenario, a complimentary experiment (omitted due to lack of space) showed some increase in the accuracy, but not up to the levels of MC.

Table 3. Accuracy [%] achieved on the Test:Piano dataset. The numbers styled in bold italics denote matched train-test conditions; normal font denotes unseen SNR levels.

Model	**SNR levels**				
	clean	10	0	-10	-20
Baseline	***85.0***	82.0	68.1	41.4	16.4
MC:Piano 1	***84.9***	***84.5***	***83.5***	77.7	52.3
AE:Piano 1	***84.8***	***84.6***	***83.4***	77.2	52.6
CAE:Piano 1	***84.8***	***84.5***	***83.3***	77.9	54.4
MC:Piano 2	***84.8***	***84.3***	***83.7***	***81.6***	***72.3***
AE:Piano 2	***83.8***	***83.5***	***82.8***	***79.3***	***67.6***
CAE:Piano 2	***83.9***	***83.7***	***82.8***	***80.1***	***70.3***

4.4. Mismatched training-test conditions

This section discusses scenarios, in which the systems were exposed to data with unseen SNR levels (negative SNR in Tables 3 and 4), and unobserved music genres (piano and violin compositions in Table 5).

Within the *Test:Piano,* the accuracy baseline model falls to 16.4%. All Piano 1 models are able to partly improve by up to

Table 4. Accuracy [%] achieved on the Test:Electro dataset. The numbers styled in bold italics denote matched train-test conditions; normal font denotes unseen SNR levels.

Model	SNR levels				
	clean	10	5	0	-5
Baseline	***85.0***	78.9	65.1	38.9	18.4
MC:Electronic	***84.8***	***83.6***	***81.6***	***74.7***	53.1
AE:Electronic	***84.5***	***82.3***	***78.6***	***65.6***	38.4
CAE:Electronic	***84.1***	***81.9***	***77.8***	***62.4***	36.2

38% (SNR level −20 dB). The CAE:Electronic achieves the highest accuracy for very low SNR. The access to data with negative SNR levels (i.e., the Piano 2 models) during training improves the results considerably, improving the baseline performance by up to 55.9%.

In the *Test:Electro* scenario, the baseline model performs poorly, below 18.4% accuracy. Even the robust techniques are only partially able to compensate the difficult acoustic conditions, achieving 53.1% accuracy, for SNR level −5 dB. The MC:Electronic performs substantially better than both autoencoder models (by up to 14.7%). This corroborates the lower performance of autoencoders in more difficult scenarios.

The results achieved on *Test:Violin* demonstrate that the studied techniques are functional on unobserved music genres and improve their accuracy over the baseline recognizer (up to 24.3% at a SNR level of 0 dB). The MC models are more robust with respect to unseen music genres than the AE/CAE models.

Considering the positive SNR levels, the best results are achieved using MC:Electronic model, trained on the broadest spectrum of music genres. This indicates that for the sake of scenarios with mismatched training-test conditions, it is beneficial to include a broad range of genres in the training set. In the case of negative SNR levels, the best performance is achieved by MC:Piano2, which had access to negative SNR levels during training (but not the music genre). This indicates that the benefits of adding broad spectrum SNR levels in the training set are preserved even for unobserved music genres during tests.

Table 5. Accuracy [%] achieved on the Test:Violin dataset. Bold italics denotes matched train-test conditions; normal font denotes unseen music genre and/or SNR levels.

Model	SNR levels				
	clean	10	0	-10	-20
Baseline	***85.0***	76.2	46.8	18.2	5.7
MC:Piano 1	***84.9***	83.0	69.4	41.4	15.6
AE:Piano 1	***84.8***	82.1	64.8	37.2	14.0
CAE:Piano 1	***84.8***	82.0	65.8	37.9	14.3
MC:Piano 2	***84.8***	81.4	68.4	44.2	21.5
AE:Piano 2	***83.8***	80.5	63.9	38.7	16.6
CAE:Piano 2	***86.9***	81.1	66.4	40.9	18.9
MC:Electronic	***84.8***	83.5	71.1	39.0	13.5
AE:Electronic	***84.5***	81.4	62.2	31.7	9.9
CAE:Electronic	***84.1***	80.7	60.5	30.3	9.1

4.5. Real-world test set

The testing on our real-world dataset can be considered to be under mismatched training-test conditions. The included music is of a genre similar (but not identical) to music samples in the Electronic dataset. We estimate the SNR level of these recordings to be about 10 dB. The robust techniques are able to improve over the baseline recognizer by about 2.4%, which corresponds to the improvement in the simpler Test:Piano scenario at SNR level 10 dB.

Table 6. Accuracy [%] achieved on the Real-world dataset (mismatched training-test conditions; unseen music genre).

Model	
Baseline	83.7
MC:Elect 1	86.1
AE:Elect 1	85.8
CAE:Elect 1	86.1

5. CONCLUSIONS AND DISCUSSION

From the results stated above, we draw the following conclusions: 1) Both studied techniques are able to compensate for the performance decrease (caused by interfering music) encountered by a single-style baseline model. 2) The accuracy achieved by both techniques is comparable for matched train-test conditions and simpler background music. 3) The multi-condition models exhibit superior accuracy for mismatched training-test scenarios (with unseen music genre and/or SNR level) and for more complex background music. We hypothesize that more complicated scenarios will require more data to train the autoencoders. This holds for the Electronic training dataset (which consists of a broad spectrum of music genres) and also for the Piano 2 dataset (which contains a broad range of SNR levels). 4) Comparing both autoencoder topologies, the fully connected one achieves a higher performance compared to convolutional one in more difficult scenarios. The convolutional autoencoder exhibits a higher performance for simpler scenarios and lower SNR levels. 5) In accordance with literature, the models trained with broader range of music genres are more robust in mismatched train-test conditions. 6) The access to a broader range of SNR levels during the training helps in scenarios with similar SNR levels and unseen music genres.

The comparison of autoencoders is partly in contrast with the results presented in [15], where the performance of the convolutional autoencoder was superior to a fully connected case for all considered scenarios. We argue that this could be caused by: 1) the lower number of hidden units in our CAE compared to [15] (which we keep equal to number of neurons in AE). Our complimentary experiment confirmed that CAE benefits from the increased number of neurons significantly more than AE. 2) We consider more general target music (in [15], the autoencoders are trained for a specific "song", we train with respect to a general genre).

Concerning computational demands, the multi-condition training is less demanding, requiring training/utilization of only a single network. The advantage of autoencoders dwells in the simplicity of obtaining large amount of data for training, because there is no need for reference texts labeled manually. This fact inspires our future work, in which we will study the size of the datasets required for efficient training of the studied techniques and the benefits of increasing/decreasing that size. We expect that the training of an acoustic model on data preprocessed by the autoencoder requires a smaller (labeled) dataset in comparison with full multi-conditional training. Moreover, the autoencoder can be trained in a multilingual fashion [15], serving as a preprocessing tool for several language-specific acoustic models.

6. REFERENCES

[1] Jinyu Li, Li Deng, Yifan Gong, and Reinhold Haeb-Umbach, "An overview of noise-robust automatic speech recognition," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 22, no. 4, pp. 745–777, 2014.

[2] Michael L Seltzer, Dong Yu, and Yongqiang Wang, "An investigation of deep neural networks for noise robust speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 7398–7402.

[3] Keizo Kinoshita, Marc Delcroix, Takashi Yoshioka, Takeshi Nakatani, Armin Sehr, Walter Kellermann, and Roland Maas, "The reverb challenge: Acommon evaluation framework for dereverberation and recognition of reverberant speech," in *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2013 IEEE Workshop on*. IEEE, 2013, pp. 1–4.

[4] "Reverb challenge [online]," http://reverb2014.dereverberation.com/, Accessed: 2016-29-08.

[5] Yong Xu, Jun Du, Li-Rong Dai, and Chin-Hui Lee, "An experimental study on speech enhancement based on deep neural networks," *Signal Processing Letters, IEEE*, vol. 21, no. 1, pp. 65–68, 2014.

[6] Xugang Lu, Yu Tsao, Shigeki Matsuda, and Chiori Hori, "Speech enhancement based on deep denoising autoencoder.," in *INTERSPEECH*, 2013, pp. 436–440.

[7] Emmanuel Vincent, Jon Barker, Shigetaka Watanabe, Jonathan Le Roux, Francesco Nesta, and Marco Matassoni, "The second chimespeech separation and recognition challenge: Datasets, tasks and baselines," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 126–130.

[8] Shoko Araki, Tomoki Hayashi, Marc Delcroix, Masakiyo Fujimoto, Kazuya Takeda, and Tomohiro Nakatani, "Exploring multi-channel features for denoising-autoencoder-based speech enhancement," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 116–120.

[9] Jon Barker, Ricard Marxer, Emmanuel Vincent, and Shinji Watanabe, "The thirdchime'speech separation and recognition challenge: Dataset, task and baselines," in *2015 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU 2015)*, 2015.

[10] Marc Delcroix, Yotaro Kubo, Tomohiro Nakatani, and Atsushi Nakamura, "Is speech enhancement pre-processing still relevant when using deep neural networks for acoustic modeling?," in *INTERSPEECH*. 2013, pp. 2992–2996, ISCA.

[11] Angkana Chanrungutai and Chotirat Ann Ratanamahatana, "Singing voice separation for mono-channel music using non-negative matrix factorization," in *Advanced Technologies for Communications, 2008. ATC 2008. International Conference on*. IEEE, 2008, pp. 243–246.

[12] Emad M Grais and Hakan Erdogan, "Single channel speech music separation using nonnegative matrix factorization and spectral masks," in *Digital Signal Processing (DSP), 2011 17th International Conference on*. IEEE, 2011, pp. 1–6.

[13] Pablo Sprechmann, Alexander M Bronstein, and Guillermo Sapiro, "Real-time online singing voice separation from monaural recordings using robust low-rank modeling.," in *ISMIR*, 2012, pp. 67–72.

[14] Cemil Demir, Murat Saraclar, and Ali Taylan Cemgil, "Single-channel speech-music separation for robust asr with mixture models," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 21, no. 4, pp. 725–736, 2013.

[15] Mengyuan Zhao, Dong Wang, Zhiyong Zhang, and Xuewei Zhang, "Music removal by convolutional denoising autoencoder in speech recognition," in *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*. IEEE, 2015, pp. 338–341.

[16] "Free music archive [online]," http://freemusicarchive.org/, Accessed: 2016-08-29.

[17] George Dahl, Yu Dong, Li Deng, and Alex Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 1, pp. 30 –42, jan. 2012.

[18] Steve Young, "The htk hidden markov model toolkit: Design and philosophy," *Entropic Cambridge Research Laboratory, Ltd*, vol. 2, pp. 2–44, 1994.

[19] Richard J Mammone, Xiaoyu Zhang, and Ravi P Ramachandran, "Robust speaker recognition: A feature-based approach," *Signal Processing Magazine, IEEE*, vol. 13, no. 5, pp. 58, 1996.

[20] "Torch - a scientific computing framework for luajit [online]," http://torch.ch, Accessed: 2016-08-29.

[21] Tara N Sainath, Abdel-rahman Mohamed, Brian Kingsbury, and Bhuvana Ramabhadran, "Deep convolutional neural networks for lvcsr," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 8614–8618.

[22] Yajie Miao and Florian Metze, "Improving language-universal feature extraction with deep maxout and convolutional neural networks," 2014.

[23] Yann LeCun and Yoshua Bengio, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, pp. 1995, 1995.

[24] "Cesky rozhlas - radio station radiozurnal [online]," http://www.rozhlas.cz/radiozurnal/, Accessed: 2016-08-29.