Dithering Techniques in Automatic Recognition of Speech Corrupted by MP3 Compression: Analysis, Solutions and Experiments

Michal Borsky, Petr Mizera, Petr Pollak, Jan Nouza

PII: S0167-6393(16)30086-3
DOI: 10.1016/j.specom.2016.11.007
Reference: SPECOM 2420

To appear in: *Speech Communication*

Received date: 21 April 2016
Revised date: 6 October 2016
Accepted date: 23 November 2016

Please cite this article as: Michal Borsky, Petr Mizera, Petr Pollak, Jan Nouza, Dithering Techniques in Automatic Recognition of Speech Corrupted by MP3 Compression: Analysis, Solutions and Experiments, *Speech Communication* (2016), doi: 10.1016/j.specom.2016.11.007

Dithering Techniques in Automatic Recognition of Speech Corrupted by MP3 Compression: Analysis, Solutions and Experiments

Michal Borsky[1]*, Petr Mizera*[1]*, Petr Pollak*[1] *and Jan Nouza*[2]

[1]Faculty of Electrical Engineering, CTU in Prague, Czech Republic
[2]Institute of Information Technology & Electronics, TUL, Czech Republic
borskmic@fel.cvut.cz*,mizerpet@fel.cvut.cz,pollak@fel.cvut.cz,jan.nouza@tul.cz

Abstract

A large portion of the audio files distributed over the Internet or those stored in personal and corporate media archives are in a compressed form. There exist several compression techniques and algorithms but it is the MPEG Layer-3 (known as MP3) that has achieved a really wide popularity in general audio coding, and in speech, too. However, the algorithm is lossy in nature and introduces distortion into spectral and temporal characteristics of a signal. In this paper we study its impact on automatic speech recognition (ASR). We show that with decreasing MP3 bitrates the major source of ASR performance degradation is deep spectral valleys (i.e. bins with almost zero energy) caused by the masking effect of the MP3 algorithm. We demonstrate that these unnatural gaps in spectrum can be effectively compensated by adding a certain amount of noise to the distorted signal. We provide theoretical background for this approach where we show that the added noise affects mainly the spectral valleys. They are filled by the noise while the spectral bins with speech remain almost unchanged. This helps to restore a more natural shape of log spectrum and cepstrum, and consequently has a positive impact on ASR performance. In our previous work, we have proposed two types of the signal dithering (noise addition) technique, one applied globally, the other in a more selective way. In this paper, we offer a more detailed insight into their performance. We provide results from many experiments where we test them in various scenarios, using a large vocabulary continuous speech recognition (LVCSR) system, acoustic models based on gaussian-mixture model (GMM) as well as on deep-neural network (DNN), and multiple speech databases in three languages (Czech, English and German). Our results prove that both the proposed techniques, and the selective dithering method, in particular, yield consistent compensation of the negative impact of the MP3 compressed speech on ASR performance.

Index Terms: uniform dithering, spectrally selective dithering, MP3 compression, GMM-HMM, DNN-HMM

1. Introduction

Modern automatic speech recognition (ASR) systems are becoming an integral part of our lives. They assist us in various situations, e.g. as voice operated interfaces to home devices, personal assistants, dictation programs and programs for producing sub-titles, voice search tools, broadcast monitoring systems, etc. Usually, they perform well under quiet acoustic conditions and with high quality signals. On the other side, when recording is done in noisy environment or the speech signal is distorted on its way from a microphone to the ASR engine, their usability can drop significantly. While some types of signal degradation can be hardly avoided, there are distortions that have been introduced unintentionally, as a side effect of our needs to encode a signal effectively and to reduce data transmission flow or computer memory.

Viewed broadly, audio encoding is the process of creating a compact representation for an audio signal, with as little redundant information as possible. Most modern audio compression algorithms are based either on the source model of speech production or on the perceptual model of hearing. They led to the inventions of various coders, often with very specific and narrow areas of application. In general, the voice source encoders are better suited for speech and have been widely employed in telecommunications, while the perceptual coders are historically more popular in multimedia, namely for video and music storage and distribution.

The compression algorithm widely known under name MP3 belongs to the perceptual audio encoders that were developed primarily for the music industry, but it has seen successful use for speech encoding as well. The reasons are mainly historical as it was among the first encoders that allowed a high compression rate and provided good perceptual quality. Moreover, it appeared in the period of the rapid growth of the Internet and media sharing. It quickly established itself as the principal compression algorithm used for speech and music files in the commercial, and even more in the private, sphere. Only music professionals, phoneticians, and audiophiles still shun it; they do it in spite of various studies whose results proved that even expert listeners can hardly distinguish between original and encoded files for bitrates higher than 256kbps [1].

In case of a speech signal, people tended to use much lower bit rates because even highly compressed speech (containing audible distortions) was perceived by human listeners as intelligible. Recently, professional studios and many broadcasters are leaving the MP3 coding tools and prefer formats that are better suited for speech (e.g. Speex or FLAC). However, the amount of MP3 speech that is still being produced every day, together with the data that has been compressed and archived since 1990s when MP3 became popular, is large enough to be considered as a true challenge for research in the ASR domain.

The goal of this article is to analyze the contribution of two dithering techniques proposed by the authors whose aim is to reduce the negative impact of MP3 coding on the performance of a state-of-the-art ASR system operating with two types of acoustic models: a gaussian-mixture-model (GMM) and a deep neural network (DNN) one. The paper is organized as follows: Section 2 analyzes and summarizes the impact of MP3 compression on a speech signal in time and spectral domains and examines the artifacts it introduces. Section 3 reviews some of the algorithms commonly used for the adaptation of ASR sys-

tems to mismatched conditions. This is followed by the presentation of several compensation techniques designed specifically for MP3 compression by the authors of this paper and by others. Section 4 is devoted to experiments; it describes their setup, employed systems and data, and provides results achieved in many tests conducted with speech recordings in three languages (Czech, English and German). In the final section, we summarize the advantages of our approach, discuss the results and mention several practical applications.

2. MP3 Compression and ASR

MP3 refers to Layer 3 of the MPEG/audio algorithm which was the first international standard for a high-fidelity digital-audio compression algorithm adopted by ISO/IEC in 1992. The core principles of its encoding process are based on the perceptual limitations of human hearing which include temporal and spectral masking and the hearing threshold. Since the standard itself is defined as open, its specifications are available to anyone for free. On the other side, no precise specifications exist on how to implement the encoder - there are only suggestions, and any interested subject can implement its own version. This has created a situation when there are various encoders available; e.g. Lame [2], mp3Pro [3] and iTunes; each with different audio quality on its output [4].

Although the algorithm's designers made strong requirements on keeping high-fidelity for the output quality as measured in subjective listening tests, the encoding method is lossy by its nature and introduces audible distortions [5]. These can easily be identified using spectral analysis, and have been shown to influence also estimations of basic vocal characteristics such as pitch and formant frequencies [6]. And what is more important, they can have a significant impact on ASR systems, because their acoustic processing front-end heavily relies on spectral features.

2.1. Spectral Distortion of MP3 Speech

Several detailed studies dealing with distortions introduced by the MP3 coding have been published previously, e.g. [5] and [7]. Although their authors have generally been concerned with perceptually distinguishable artifacts, their conclusions can be extended to the field of ASR as well. There are two main factors that cause signal degradation: a) bandlimiting, and b) introduction of unnaturally deep spectral valleys (also called gaps), which are spectrum bins with a very low energy. Our previous works [8] and [9] have proven it, too.

The MP3 format actively narrows the spectral bandwidth as its bitrate decreases in order to improve the subjective quality after compression. This introduces two separate problems. The first is that higher frequencies carry a major portion of the information about certain phonetic units (namely unvoiced consonants) and this is lost during compression. As a result, the partial error rate for these units increases more rapidly than for voiced units, which in turn steeply moves up the overall error rate; see [10]. The second problem can potentially occur when a general purpose ASR is used to decode compressed speech. Because of a mismatch between training and testing conditions, a performance of an ASR system decreases. Although it might seem tempting to solve this problem by training the acoustic model (AM) on compressed signals, there are several drawbacks to this strategy. Each bitrate is assigned its own low-pass (LP) cut-off frequency, and thus this strategy would require training bitrate-specific (matched) AMs. Moreover, a bitrate signal detector would have to precede the standard ASR scheme. However, because of different implementations mentioned above, MP3 coded signals with the same bitrates can differ. Nouza et al. in [9] have shown that ASR accuracy achieved with two same(low) bitrates but different encoders may differ by more than 40%. This would suggest that the AMs would have to be not only bitrate-specific, but encoder-specific as well.

The application of a psychoacoustic model creates artifacts known as spectral valleys, which are easily identifiable in spectrograms as almost zero energy areas at low and middle frequencies, see Figure 1 for an example. The exact nature of this artifact is highly context-dependent, making it statistically impossible to predict the affected frequency bins in the spectral domain. This type of distortion often influences only some parts of speech (usually starts and ends of continuously uttered phrases) and some phonemes. It makes spectral features, and those that build on them (e.g. cepstral ones), less reliable. Our earlier work on that topic [9] theoretically analyzed the effects of spectral valleys on extracted features and showed that the MP3 coding displaced the positions of features in the cepstral domain and significantly increased their variances. The latter has a large impact on the dynamic (delta and delta-delta) coefficients. We did experiments with excluding these 1st and 2nd order parameters from the feature vectors but it did not help and word error rates were still considerably high.

2.2. Impact of Additive Noise on MP3 Speech Features

In our previously published papers [9] and [10], it has been shown that the problem of spectral valleys caused by MP3 compression can be lessened to a certain level by adding some small amount of noise to the corrupted speech signal. Usually, additive noise is considered as something that always corrupts speech signals and a lot of research has been done to find ways how to remove it. However, in case of signals with unnatural valleys in spectrum, additive noise can be a simple yet efficient mean to fill these gaps and to make the spectrum more natural. In the following text, we try to explain why it is possible and how it works.

Most state-of-the-art ASR systems use cepstral features extracted from the signal. Often it is Mel-Frequency Cepstral Coefficients (MFCC) or Perceptual Linear Prediction (PLP) parameters. Let us focus on the former, now. The procedure to compute MFCCS is well-known and rather straightforward. It includes two non-linear operations, which makes exact derivations more difficult, but it is still possible to get at least a picture of what is happening when a signal with added noise is parameterized. We shall follow the approach proposed by Pettersen in his thesis [11].[1]

The standard MFCC computation consists of five basic blocks (as shown in Fig. 2). Let us consider speech samples $s[n]$ to which noise samples $r[n]$ are added. The latter are supposed to be uniformly distributed values drawn from the interval $<-R, R>$. After the compound signal $x[n] = s[n] + r[n]$ passes the first block, we get its complex spectrum where both components are still additive, i.e. $X[k] = S[k] + R[k]$. When the magnitude square is computed (the first non-linearity), the resulting equation has a more complex form

[1] Let us note that the main motivation of Pettersen's work was to investigate new methods for noise compensation techniques in robust speech recognition, i.e. he aimed at removing or suppressing noise. Our approach is just opposite: We want to add some noise to speech signal distorted by MP3 compression in order to restore its corrupted spectral and cepstral features.

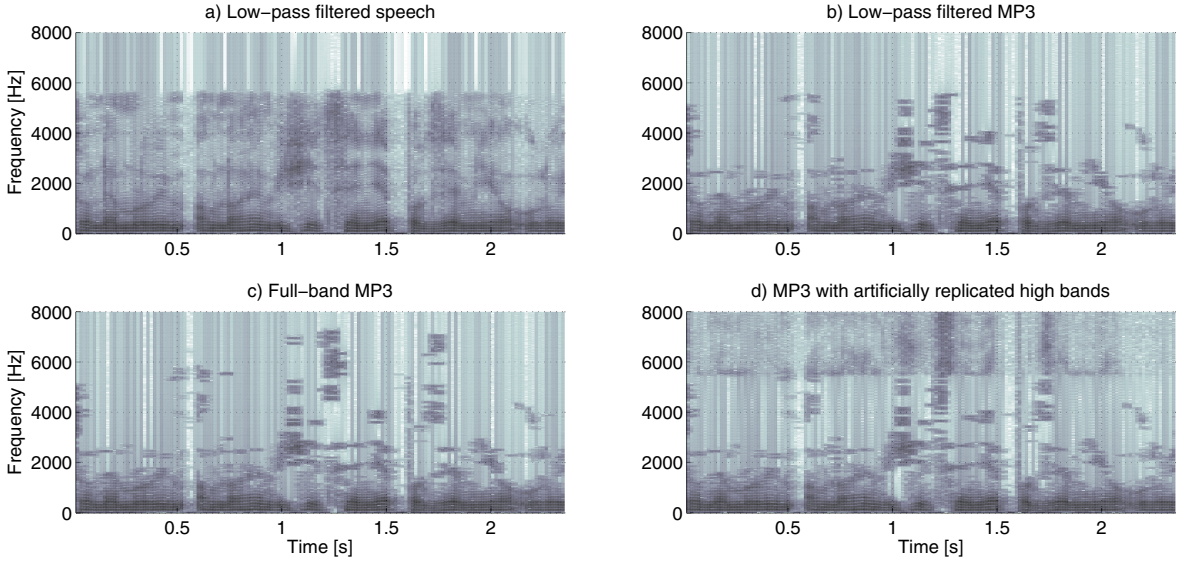

Figure 1: *Illustrative spectrograms for all the studied cases of removing and degrading the information in time-frequency domain introduced by a MP3 compression.*

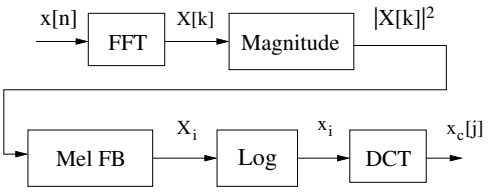

Figure 2: *Standard scheme of MFCC parameterization*

$$|X[k]|^2 = |S[k]|^2 + |R[k]|^2 + 2|S[k]||R[k]|\cos\phi[k] \quad (1)$$

where $\phi[k]$ is the angle between $S[k]$ and $R[k]$. In the next step, the sum of the weighted powers (squares) of individual DFT bins is computed for each Mel-shaped filter bank channel

$$X_i = \sum_k W_i[k]|X[k]|^2 \quad (2)$$

and we obtain the power in the i-th Mel channel assigned as X^i, while $W^i[k]$ are weighting coefficients that usually correspond to a triangle window. This is a linear operation, and so it can be applied to both of the original components and on the basis of eqs. (1) and (2) we can write

$$X_i = \sum_k W_i[k]|X[k]|^2 + \sum_k W_i[k]|X[k]|^2 + 2\sum_k W_i[k]|S[k]||R[k]|\cos\phi[k] \quad (3)$$

which can be simplified as

$$X_i = S_i + R_i + f_1(S_i, R_i, \phi_i). \quad (4)$$

where $f_1(.)$ is a slightly complex function representing the non-linear component. If we assume that $x_i = \log X_i$, $s_i = \log S_i$, and $r_i = \log R_i$, then eq. (4) can be rewritten as

$$\frac{X_i}{S_i} = 1 + \frac{R_i}{S_i} + \frac{f_1(S_i, R_i, \phi_i)}{S_i} \quad (5)$$

or

$$e^{x_i - s_i} = 1 + e^{r_i - s_i} + f_2(S_i, R_i, \phi_i, s_i). \quad (6)$$

where $f_2(.)$ is a non-linear function modeling the impact of the first non-linearity. In [11], Pettersen shows that when the second non-linear operation (logarithmic function) is applied, the logarithmic power in the i-th Mel channel has the following form, composed of several terms related to the contributions of the original speech and noise signals:

$$x_i = s_i + \log\left(1 + e^{r_i - s_i} + f_3(S_i, R_i, s_i, r_i)\right). \quad (7)$$

The last term in (7) is again a rather complex non-linear function $f_3(.)$. But it can be ignored in situations where either speech or noise dominates. In these two cases, the equation can be further simplified, and it can be shown that $x_i \approx s_i$ when speech dominates, while $x_i \approx r_i$ when noise dominates (for details, see page 18 in Pettersen's work [11]). The last step, the DCT computation, is just a linear transform of the values x_i into the MFCC coefficients $x_c[j]$.

Moreover, when we assume an uncorrelated dithering noise (which is the case for uniform dithering), the eq. (1) is simplified to

$$|X[k]|^2 = |S[k]|^2 + |R[k]|^2 \quad (8)$$

and consequently the eq. (7) to

$$x_i = s_i + \log\left(1 + e^{r_i - s_i}\right). \quad (9)$$

The above-mentioned dominance is in this case more significant, and small additive uncorrelated dithering should not distort the principal information related to the speech portion of the signal.

Let us return back to our task. From the above analysis, we may conclude that when we add noise to an MP3 signal, it will influence mainly the spectral bins with a very low energy, i.e.

the spectral valleys will be filled by noise, while the bins with speech will remain almost unaffected. This explains why the additive noise can play a positive role in this special situation. The resulting signal will have spectrum that is more natural, and hence also the MFCCs will better match the training conditions. This is in full agreement with our experimental results published in [8] and [9].

A similar contribution can be supposed also for PLP cepstral features because the PLP procedure utilizes very similar non-linear operations. (The only difference is that the second one is applied in an alternative way, via AR modeling.)

3. Techniques Improving MP3 ASR

All previous studies on the practical usability of MP3 recordings have concluded that an ASR system will work with a little degradation if a sufficiently high bitrate is used. Their authors agreed that the 24kbit bitrate represented the threshold below which accuracy started to drop rapidly [12], [13] and [14].

To build a system capable of precise MP3 speech recognition, one must either improve the quality of the speech features, adapt the acoustic model to the compression's characteristics, or use a more robust ASR architecture. Most published works on this subject have employed a solution that falls within these three categories. The main criterion by which these solutions can be distinguished is their usability in other similar situations. The first category comprises general-purpose compensation techniques such as feature normalization, AM adaptation, matched training and the use of a robust ASR architecture. The second category includes dithering-based algorithms which find little use outside the field of compressed speech recognition.

3.1. Matched Training

Section 2 points out that the key issue in matched training for real-life ASRs is the precise bitrate detection. This section summarizes recent works and further elaborates upon the complications associated with the process.

D'Alessandro et al. [15] have proven that precise bitrate differentiation can be based on feature vectors containing power spectral densities from narrow frequency bands of approximately 43 Hz and the use of a support vector machine classifier. Their classifier performed at 97% accuracy for bitrates down to 128kbps. The study's second significant conclusion was that a transcoding a lower bitrate into a higher one does not effect the detection accuracy.

A more recent work on double compression was performed by Qiao et al. [16], where the authors experimented with both the up-transcoding and down-transcoding scenarios. Their classifier's feature vector utilized quantized modified discrete cosine transform (MDCT) coefficients and was tested for bitrates in the range from 192kbps down to 64kbps. The results showed significant differences in detection accuracy, depending on whether the signal was down or up-transcoded. The classifier achieved 100% accuracy in the case of up-transcoding 64kbps→192kbps but only 61.8% accuracy for down-transcoding 192kbps→64kbps.

In both mentioned cases, the classification was performed for signals that were transcoded by the same coder. Yang et al. [17] followed up on the research into using different encoders in each step and found a small difference in detection accuracy. It is interesting to note that these results qualitatively corresponded with the findings in [9], but the results of using different encoders were not as significant.

When we take into account all problems involved in accurately detecting real compression rates in certain setups and a differing audio quality of various encoders, we must come to the conclusion that matched training for each bitrate is not a preferable solution for an every-day system. Nonetheless, we still investigated this option in the experimental part of this article in order to compare its performance with the unmatched conditions to see whether the potential improvement is worth the additional computational and design effort.

3.2. Acoustic Model Adaptation

AM adaptation is another common solution used to reduce the mismatch between training and testing data. The current GMM-HMM systems utilized in real-life applications use generally one of these two techniques: either maximum a posteriori (MAP) adaptation or some form of a linear transformation, e.g. feature Maximum Likelihood Linear Regression (fMLLR). The former is used in situations with large amounts of available adaptation data, and it is known to provide the best results. However, the latter is used more often. This is mostly due to its robustness against a lack of adaptation data. The fMLLR was the preferred technique also in this article.

3.3. Robust Architecture

The field of ASR has experienced a burst of development since the introduction of DNN and its derivatives a few years ago. The classic GMM-HMM architecture has been replaced by hybrid DNN-HMMs, which have proven to work much better in nearly every scenario, including the recognition of distorted speech; see [18] or [19]. Seps et al. [20] have shown that this architecture can also improve performance for the recognition of MP3 recordings by up to 30% relatively over the GMM-HMM approach.

3.4. Bandwidth Limitation

The first attempt to solve the above-mentioned bandwidth-limitation problem and to avoid the process in which all training data is compressed was conducted by Barras et al. [12]. They proposed and practically tested a parameterization scheme wherein the training data was filtered by a low-pass (LP) filter at different cut-off frequencies. A cut-off frequency was assigned to each bit-rate and the AMs were trained to better match the training and testing conditions. The compressed AMs were then tested on the compressed signals, and the results showed an absolute decrease in word error rate (*WER*) of about 1-2%.

3.5. Spectrally Selective Dithering

The idea of adding noise was first proposed in [9], where we proved that adding a certain amount of noise can improve recognition performance. The aim of this technique was to compensate for the problem of spectral valleys by "filling them in" and making compressed features more similar to the non-compressed ones. For the purpose of this study we included the results of this simple approach in our experimental part as well, and refer to it as *uniform dithering* (UD). In our subsequent work [8], we extended this approach by designing and implementing an algorithm called *spectrally selective dithering* (SSD) to automatically detect corrupted frequency bands and add a weighted amount of noise in the spectral domain in order to patch only the affected bands.

The algorithm's block scheme is presented in Figure 3. The principal idea behind it is to decompose the speech signal into

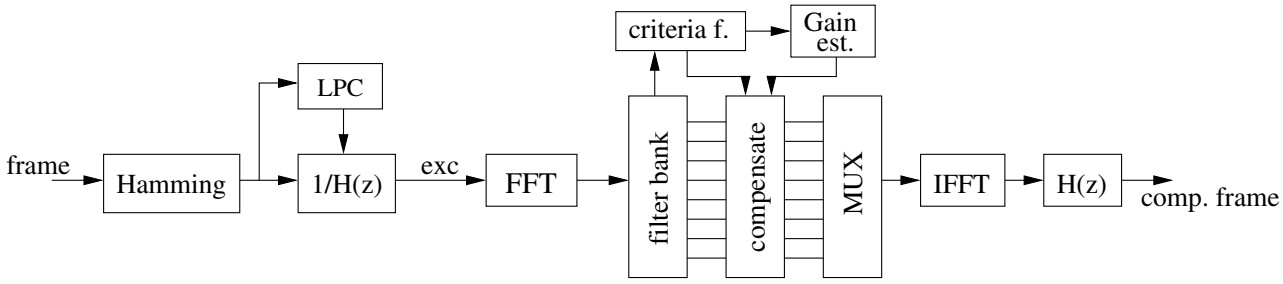

Figure 3: *Block diagram of the SSD compensation technique*

the spectral envelope represented by the LPC model and the residual signal (exc), detect the zeroed bands in the excitation signal, estimate the weight of the added noise from the unaffected bands, and repair the signal. It can be divided into two main parts: the zero-band detector and the gain estimation/compensation block.

In theory, the residual signal for the auto-regressive process approximates a random process with a Gaussian distribution and zero mean value. However, it turns out this is not true for compressed speech. Through spectral analysis it can be shown that the affected bands in the exc are quantized to zero as well. This has enabled us to employ a simple detection function, based on the smoothness of the spectral curve, to detect distorted bands. Let us assume a standard feature-extraction scheme for a short-time FFT weighted with a Hamming window, which is decomposed by an analysis LPC filter with the frequency response $1/H(z)$. The extracted exc can be then split into multiple frequency bands b, each containing the spectral components f_1 through f_2. The smoothness score for any particular band can then be defined as:

$$crit(b) = \sqrt{\sum_{f=f_1}^{f_2} (exc(f) - exc(f\text{-}1))^2}. \quad (10)$$

The most important parameter to optimize was the number of frequency bins in the band. When a band was too narrow, the detector returned too many false alarms. On the other hand, when a band was too wide, the frequency positions of zeroed bands became inaccurate. For the purpose of our experiments, we worked with 4 bins per band for $16KHz$ sampled signals. The gain of the added noise was then estimated as a global average from the unaffected bands. The compensated residual signal for a particular frame can be expressed in the form:

$$exc(b) = \begin{cases} exc(b) + G * noise, \\ exc(b) \end{cases} \quad (11)$$

The results showed that the application of this technique could bring even further error reductions in comparison to UD, as error rates dropped by absolute percentage of up to 1.8%.

4. Experiments

This section presents the series of MP3 speech recognition experiments for the purpose of evaluating the discussed compensation techniques in rather clear acoustic conditions. The first series of experiments focused on modeling MP3 distortions; the next one compared matched and mismatched training and assessed the contribution of dithering techniques for the GMM-HMM architecture. The final part dealt with experiments with DNN-HMM systems. Performance was evaluated using *WER* criteria and the experiments were performed using KALDI tool [21]. The dithering techniques were implemented either at the level of the feature extraction tool (uniform dithering within CtuCopy tool) or separately in the MATLAB environment (Spectrally Selective Dithering). The experiments were done for Czech, English and German in order to demonstrate that the compression affects different languages in the same manner.

4.1. Databases

The signals for the Czech experiments came from the SPEECON database [22] and private car speech database called as CZKCC. The selected utterances from both of these databases matched rather clear acoustic conditions, i.e. we used the SPEECON utterances recorded in an office environment and car speech data recorded in standing car with a switched-off engine. The training and the testing sets were obtained by randomly splitting the data in a 9:1 ratio. The full training set contained 72 hours and the testing set contained 2 hours of speech. Signals for the English experiments were from the WSJ database [23]. We used the full 81-hour train-si284 set plus the eval92 set as our test set. The data for German were taken from the GLOBALPHONE database [24]. The German results are presented on the eval set. Finally, the signals from all databases were recorded in acoustically clear conditions with a minimum of low-level environmental noise using 16kHz sampling frequency and 16 bit precision. This consistency in the sampling frequency and the bit-depth across the languages allowed us to use the same number of bins in a band for SSD. The final specifications of particular data sets for each language are summarized in Tab. 1.

4.2. Feature Extraction

The 13-dimensional PLP and MFCC features were extracted from the signals using CtuCopy [25] with the same 32/16*[ms]* setup and then normalized using Cepstral Mean Normalization (CMN). Five preceding and following vectors were spliced onto the central vector and then transformed via LDA/MLLT into the final 40 dimensional vector of base-line features. The compression was simulated using the Lame encoder [2]. The dithering was applied at the feature extraction level before any other normalization and transformation. The optimal value for uniform dithering was determined by manually increasing the power of added noise until a minimal error rate was achieved.

Table 1: *Summary of used setups for different languages*

	AM			LM		
lang.	train	test	phn.	voc.	n-gram	OOV
CZ	72h	2h	44	340k	2	1.2
ENG	81h	0.7h	39	125k	3	1.8
GER	14.9h	1.5h	41	38k	3	2.2

4.3. GMM-HMM Acoustic Models

The Czech AM was trained on 72 hours of speech using the Viterbi algorithm for context-dependent cross-word triphones. The starting phone set consisted of 44 monophones and a single silence model which also served as a garbage model for other non-speech events. The English AM was trained on 81 hours of speech and the starting phone set consisted of 39 monophones. The German AM was trained on 14.9 hours of speech for 41 monophones.

The quality of these AMs was later improved using speaker adaptive training (SAT), a combination of UBM + SGMM [26] and discriminative training using the *MPE* [27] criteria. SAT was based on fMMLR adaptation [28], which was also used to perform the final speaker-specific adaptation during the decoding step. The adaptation was performed in an unsupervised fashion in two steps. In the first pass, we used the baseline SI model to obtain the phonetic transcription from which the linear transformation matrices were estimated. During the second pass decoding, these transforms were applied to get the final decoded transcriptions. It must be noted, however, that fMMLR also served for channel adaptation in the case of mismatched recognition. This feature proved to be especially important for comparison of matched and mismatched conditions.

4.4. DNN-HMM Acoustic Model

The DNN-HMM hybrid system was built upon 40 dimensional base-line features which were later speaker-adapted by fMLLR. The transformation matrices were estimated during the SAT stage of GMM-HMM training. The DNN topology consisted of an input layer with 440 units (for the 40-dimensional fMLLR features with the context of 5 frames with mean and variance normalization), followed by 6 hidden layers with 2048 neurons per layer and the sigmoid activation function.

The process of building the DNN-HMM system began with the initialization of hidden layers that employed Restricted Boltzmann Machines (RBMs) and then added the output layer. The process continued with frame cross-entropy training and ended with sMBR sequence-discriminative training. More detailed information can be found in Kaldi recipe s5.

4.5. Dictionary and Language Modeling

A trigram LM with a 340k vocabulary was used for the Czech language. The LM [29] was created using the publicly available resources of the Czech National Corpus [30]. For the English experiments, we used the standard tri-gram LM available in WSJ corpora [30]. The German tri-gram LM was created using the Rapid Language Adaptation Toolkit (RLAT) [31] and can be downloaded from http://csl.uni-bremen.de/GlobalPhone/ for free. The complete information about the sizes of the test sets and the LMs used is summarized in Tab. 1.

4.6. Results and Their Discussion

The current ASR systems have found applications in many areas, not just dictation or broadcast transcription systems. Keyword spotting systems, for example, have begun to be widely deployed in the commercial sphere, too. Nonetheless, the recognition and understanding of natural human speech remains the ultimate goal. Therefore, all of the experiments in the article were run for a common LVCSR task with four specific setups and goals described, later in this text.

- ***MP3 Modeling***

The MP3 modeling part extended LP filtering and spectral-valleys experiments from our previous studies and included a new set of experiments to validate the generally accepted conclusions about perceived quality. Various papers on this topic have been presented in the past and often for different languages (e.g. English, Polish, Czech). For this reason we felt that an analysis for a single language (Czech) would be sufficient to demonstrate the outcomes of compression, as it is largely language independent. For further experimental purposes, we included two feature-extraction setups, PLPs and MFCCs, and used the GMM-HMM architecture.

- ***Matched and Mismatched Training***

The subsequent experiments compared ASR systems specifically trained on compressed speech with general AMs. To extend the experiments already performed, we included the English (ENG) and German (GER) languages alongside the standard Czech (CZ).

- ***Impact of Dithering***

The third part focused on different setups with the GMM-HMM architecture in mismatched conditions. We analyzed the contribution of an adaptation technique based on linear transformation, adding uniform noise and finally patching up the signals with the SSD algorithm.

- ***DNN***

Since the arrival of neural nets, deep learning has taken over in the ASR field for practically every task. Therefore, the final series of experiments was focused on the DNN architecture. We once again started with the matched vs. mismatched analysis and then moved on to the LVCSR task. The primary goals were to assess the potential of the proposed dithering techniques for DNN-HMM systems and to answer the question whether compensation techniques applied at the feature extraction level can improve the performance of neural network based systems.

4.6.1. Modeling of MP3 Distortion

The coefficients for the linear-phase LP FIR filter were estimated using a window method of a sufficiently high order to ensure a steep attenuation increase above the cut-off frequency. While for the 12kbps bitrate it is around 5.5kHz, the experiments in this paper went even further, in order to demonstrate that current ASR systems are reasonably resistant to bandwidth limitations. The original signals were recorded with a 16kHz sampling frequency, meaning f_{max} = 8kHz, but many real-life ASR applications run successfully over a telephone channel, which limits usable bandwidth down to 3.4kHz. Therefore we decided to stop our experiments at the 4kHz boundary.

The MFCC and PLP coefficients showed very similar results, $\Delta WER = 0.3\%$ for the full-band signals. The error rate curves followed the same trend which approximated an exponential function with a breakpoint around 6kHz. The only difference was in the rate of increase, which was steeper for the

MFCC features. It can safely be concluded that the substantial increase in *WER* for MP3 speech cannot be caused by low-pass filtering alone.

Table 2: *WER [%] for low-pass filtered speech*

f_{cut}	full	7kHz	6kHz	5kHz	4kHz
MFCC	14.6	14.7	15.1	16.5	19.5
PLP	14.3	14.3	14.3	15.3	17.5

On the other hand, the effects of spectral valleys were much harder to simulate without introducing other compression artifacts at the same time. In our previous article [8], we intentionally let spectral valleys distort only the lower parts of the bandwidths, since we assumed that the upper parts would be wiped out by the LP filter in any case. To achieve this goal, we first compressed the signals and then artificially added the upper bands using the uncompressed signals. We can thus look at these signals as if only the lower bands were compressed, with the upper bands containing all their original information. The error rates obtained were worse than for LP filtering (Δ*WER* up to 2%) but still much better than for fully compressed speech (Δ*WER* up to 9%).

The approach used for this article was slightly different. The signals were compressed at the corresponding bitrates, but the cut-off frequency was uniformly set at 8kHz. There were two main reasons for this decision. First, it is more natural for a user to simply compress audio and not worry about artificial bandwidth extension/reconstruction. Second, we wanted to find out if, by not limiting the bandwidth of MP3 speech, we could improve the system's performance. The primary argument for this approach was that even if high-frequency information was heavily distorted, it was still present and could therefore contribute to the overall performance. However, this procedure is generally considered suboptimal and is advised against by the music community. The common sense advises that the available bandwidth needs to be limited, otherwise it would introduce other unwanted artifacts. The argument against full-bandwidth MP3 is thus based on subjective listening tests and as such might not hold true for ASR. To illustrate the effects of these setups, LP filtered speech, normal MP3, full-band MP3, and artificially replicated high frequency, figure 1 contains sample spectrograms of a single signal for each setup.

Figure 4 shows the error rate for full-band and limited MP3 speech. While the MFCC features did contribute from removing the heavily distorted HF bands for lower bitrates, the situation was reversed for the PLP ones. The process did not affect signals with higher bitrates in any significant way. However, full-band MFCCs still did not outperform PLPs in any setup, which leads us to conclude that LP filtering is beneficial for MP3 ASR as well.

It can be speculated that the advantage of PLPs originates from their design, which emulates the behavior of the human hearing system. The Bark filter bank attenuates the higher frequencies, and the cube root transforms the intensity into perceived loudness. This particular knowledge is similarly, although in much greater detail, exploited in the psychoacoustic model of the MP3 encoder. The similarity was proven in the experiments with full-band features, where the impact of LP filtering on perceived audio quality was found to be equally present for recognition using PLPs.

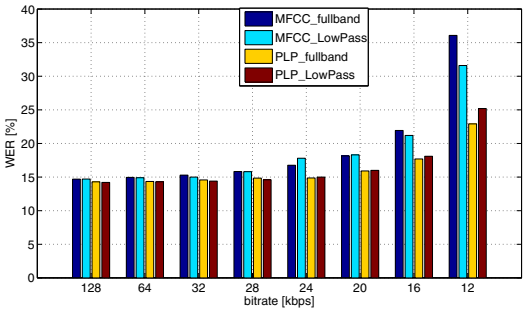

Figure 4: *Comparison of fullband and standard(LP-filtered) MP3*

Table 3: *WER*[%] *in matched & mismatched training, GMM*

bitrate	MFCC match	MFCC mismatch	PLP match	PLP mismatch
CZ				
clean	14.6	-	14.2	-
128kbps	15.7	14.7	14.5	14.2
64kbps	15.8	14.9	14.8	14.3
32kbps	16.5	15.0	15.0	14.4
28kbps	16.7	15.8	15.0	14.6
24kbps	17.1	17.8	15.0	15.0
20kbps	17.1	18.3	15.0	16.0
16kbps	19.7	21.2	16.8	18.1
12kbps	22.9	31.6	18.9	25.2
ENG				
clean	8.5	-	8.3	-
128kbps	8.7	10.0	8.7	9.4
64kbps	8.8	10.4	8.5	9.3
32kbps	8.7	10.9	8.7	9.8
28kbps	8.7	11.6	8.7	10.1
24kbps	9.3	12.3	8.9	11.0
20kbps	9.7	13.3	9.1	11.7
16kbps	11.1	16.4	9.9	13.8
12kbps	11.5	22.6	10.4	17.1
GE				
clean	18.7	-	18.8	-
128kbps	18.9	18.8	18.8	19.1
64kbps	19.1	19.1	18.8	19.1
32kbps	19.2	19.4	19.1	19.5
28kbps	19.9	19.9	19.0	19.9
24kbps	20.2	23.0	19.3	20.5
20kbps	21.0	26.2	20.1	21.7
16kbps	22.7	33.1	22.8	25.4
12kbps	26.6	46.0	23.3	42.6

4.6.2. Matched and Mismatched Training

The problems and potential gains of using bitrate-specific AMs were discussed at the beginning of this article. This section presents the results from our practical experiments with these models. The focus of this part was to compare mismatched and matched AMs. In general, the fMLLR has been shown to improve the performance of ASR systems deployed in different environmental conditions. Since this feature can be exploited naturally by utilizing the speaker-specific adaptation in mismatched conditions, we decided to include it into our setup. The

set of languages was expanded to include Czech, German and English, so as to prove the hypothesis that the negative effects of compression are language-independent to a large degree.

The results are summarized in Table 3, and there are several interesting things to see there. The English and German languages displayed the expected trend, where all the matched conditions outperformed the mismatched ones for virtually all bitrates. For low bitrates especially (<16kbps), the difference was significant, and the improvement from the use of matched AMs reached as high as 49% relatively. The second notable trend was a relatively smaller degradation of MFCCs over PLPs and a much slower increase in *WER* as a function of bitrate. It can be concluded that training bitrate-specific AMs is potentially a viable option for these languages, since the overall improvements in error rates were significant.

However, the Czech language displayed a different trend due to slightly more varying acoustic environment of used signals from the SPEECON and CZKCC databases as it was mentioned in Sec. 4.1. Although we have selected subparts with rather clear acoustic conditions, the recordings were not done in such uniform environment like in the case of WSJ for English and GlobalPhone for German. The advantage of using matched training was clear only for bitrates <= 24kbps. While the absolute *WER* over mismatched conditions was within 1%, and had a decreasing tendency, the higher bitrates still suffered from the process. The rest of the trends discussed above remained the same for Czech as well as for English and German.

4.6.3. Impact of Dithering

The experiments in this section started with speaker-independent models, which were gradually improved. Although results for speaker-adapted models were already presented in the previous section, they are mentioned here mainly for the purpose of consistency. The sequence of feature refinement was finished with applying either the uniform dithering or spectrally selective dithering. The acronyms for particular setups are listed below, and the results are presented in Table 4.

- GMM1 - speaker-independent
- GMM2 - speaker-dependent
- GMM3 - speaker-dependent + UD compensation
- GMM4 - speaker-dependent + SSD compensation

The initial error rates for clean speech differed for each language. The large amount of training data for English, along with its relatively simple grammatical structure, were the primary reasons why it achieved the best *WER* of 8.3%. The results for Czech were significantly worse, with *WER* of 14.2% and German scored last, with *WER* of 18.8%. The subpar performance for Czech can most likely be attributed to the lower complexity of used LM (bigram LM) and the vocabulary size which is still suboptimal for Czech as highly inflective language in comparison to English. Concerning German, the much worse results were clearly influenced by very small amount of training data available in GlobalPhone database. Despite of this, the overall contribution of the dithering techniques was clearly proven for all the studied languages and bitrates.

Aside from a single case (German and 12kbps), the SSD algorithm outperformed the UD technique by a slight margin for all studied languages and bitrates. The relative average improvement was highly language-specific, at 1.28% for Czech, 12.5% for English, and 1.73% for German. On the other hand, the relative improvement of using the system which employed the SSD compensated features (GMM4) over the non-compensated system (GMM2) was more consistent, with only Czech displaying a notably varying result. It was 15% for German, 13% for English and 1.8% for Czech. Based on these results, it can be concluded that SSD-compensated features are more similar to non-compressed features and are a preferable solution for MP3 recognition.

Figure 5 plots the results for SSD-compensated features. Although the baseline *WERs* differed greatly, all languages displayed the same general trend. The breakpoint for the exponential increase occurred around 20kbps. Based on these results, it can be concluded that MP3 compression is largely language-independent.

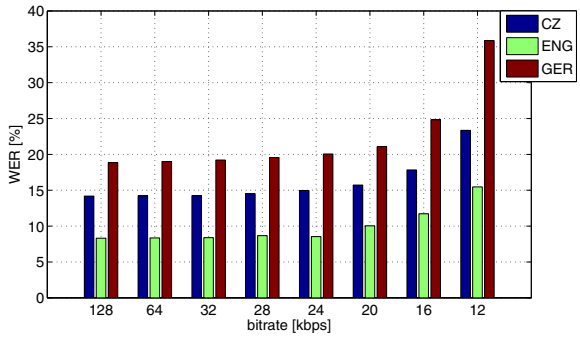

Figure 5: *Error for SSD-compensated features in GMM system and all languages*

4.6.4. DNN-HMM Architecture

Neural net based AMs have displayed a much greater robustness against adverse environmental conditions than their GMM predecessors. This feature has naturally raised a question whether the DNN systems can still benefit from any feature-level modification technique. To answer this question, we used the same experimental protocol and similar acronyms as before, resummarized for clarity in the points bellow. The only difference was the choice of the language: we focused solely on English. The reasons that led us to this decision were mainly based on the amount of data available. DNN systems are known to be data-hungry, and failing to provide sufficient data will result in a system that gives higher error rates than a comparable GMM system.

- DNN1 - speaker-independent
- DNN2 - speaker-dependent
- DNN3 - speaker-dependent + UD compensation
- DNN4 - speaker-dependent + SSD compensation

The comparison between matched and mismatched conditions is shown in Table 5. The qualitative trends previously observed in GMM systems held true for DNN as well. The matched training significantly outperformed the mismatched training, starting at 0% for 128kbps and ending at a relative 60.5% for 12kbps. Another comparison could be drawn against the matched GMM, where the overall difference between DNN and GMM systems was up to 2% and the neural nets proved to outperform the GMM architecture. However, the absolute ΔWER decreased with decreasing bitrate (only 0.1% for

Table 4: *WER [%] for MP3 speech and PLP*

	CZ				ENG				GER			
bitrate	GMM1	GMM2	GMM3	GMM4	GMM1	GMM2	GMM3	GMM4	GMM1	GMM2	GMM3	GMM4
128kbps	18.20	14.29	14.24	**14.18**	11.60	9.49	9.51	**8.32**	22.13	19.10	19.13	**18.85**
64kbps	18.43	14.35	14.27	**14.22**	11.82	9.37	9.65	**8.35**	22.33	19.12	19.32	**19.01**
32kbps	18.70	14.45	14.5	**14.25**	11.95	9.89	9.67	**8.39**	22.68	19.56	19.64	**19.21**
28kbps	19.21	14.64	14.64	**14.52**	12.74	10.14	9.98	**8.67**	23.54	19.94	19.95	**19.57**
24kbps	19.87	15.02	15.07	**14.97**	13.47	11.02	11.21	**9.54**	24.48	20.53	20.69	**20.05**
20kbps	21.27	16.02	15.88	**15.72**	14.23	11.70	11.70	**10.04**	26.89	21.72	21.78	**21.08**
16kbps	25.19	18.15	18.27	**17.83**	17.65	13.82	13.47	**11.72**	33.61	25.45	25.66	**24.83**
12kbps	34.73	25.20	24.04	**23.35**	25.16	17.12	16.39	**15.47**	51.02	42.68	**34.87**	35.86

12kbps), which led us to the conclusion that the constraints of current MP3 recognition are dependent more on bitrate and less on the choice of ASR architecture. This observation proved true for the mismatched system as well where the speech compression process used nearly quadrupled the error rate from 6.8% $\rightarrow$ 26.1%. In comparison, the error rate in GMM dropped from 8.3% to 17.1%. In other words, the clean-conditioned DNN outperformed the GMM, but the compressed GMM outperformed the DNN.

Table 5: *WER*[%] *in matched & mismatched training, DNN*

	PLP	
bitrate	match	mismatch
ENG		
clean	6.8	-
128kbps	7.0	7.0
64kbps	7.0	7.1
32kbps	7.2	7.4
28kbps	7.23	7.4
24kbps	7.6	8.5
20kbps	7.96	9.2
16kbps	8.2	11.9
12kbps	10.3	26.1

The final experiment involved the LVCSR task with the DNN system in mismatched conditions and with compensated features. The fMLLR adaptation provided the major portion of the improvement by up to a relative 42%. The application of a feature compensation technique, in the form of either UD or SSD, yielded mixed results. The former often worsened the actual recognition score and the latter brought only marginal improvements. The relative reduction in *WER* reached 2.9% on average, and the only notable improvement was observed for 12kbps, at 10.27%. For comparison, the relative improvement for the GMM system and English was 13%. These results indicate that modifying features by adding noise does not bring the same benefit for DNN systems as it does for GMM systems.

Figure 6 plots the results of GMM and DNN systems for all setups. The graphs illustrate the advantages of using neural nets for all bitrates aside from the very lowest.

5. Conclusion

This article analyzes the limits in automatic recognition of MP3 compressed speech using systems built on the common GMM-HMM and DNN-HMM architectures. It provides a detailed overview of the artifacts introduced by the MP3 algorithm and quantifies their negative impacts on a speech signal and an ASR system. Several possible compensation methods

Table 6: *WER [%] for MP3 speech and PLP*

	ENG			
bitrate	DNN1	DNN2	DNN3	DNN4
clean	8.33	**6.82**	7.11	6.93
128kbps	9.06	7.07	7.28	**6.84**
64kbps	9.57	7.16	7.30	**6.91**
32kbps	9.75	7.44	7.58	**7.25**
28kbps	9.94	7.46	7.78	**7.37**
24kbps	12.81	8.56	9.60	**8.49**
20kbps	14.42	9.29	10.31	**9.06**
16kbps	20.70	**11.94**	13.22	12.00
12kbps	38.61	26.17	**23.36**	23.50

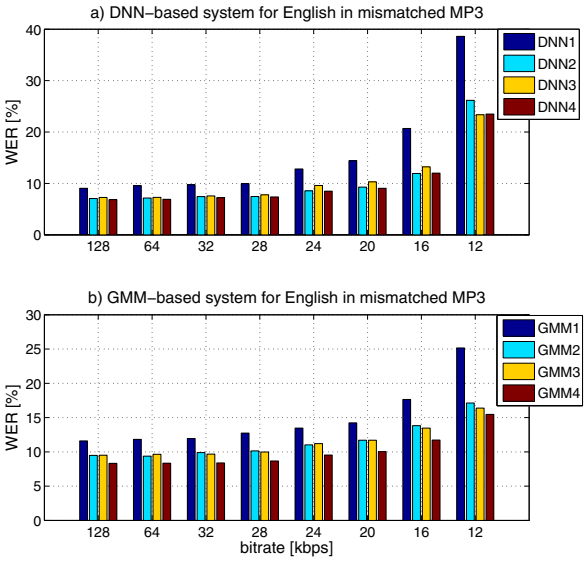

Figure 6: *Comparison of GMM and DNN systems for the English and all stages of AM refinement*

are described and reviewed. The first group include a matched training method and an acoustic model adaptation, i.e. the approaches commonly used for environmental adaptation. The second group includes methods based on uniform or spectrally selective noise dithering, which find their usage primarily (and almost exclusively) in this domain. Their main advantage is the fact that they do not require any prior information about the signal, about the compression type and rate, or about the ASR system.

The main conclusions from the initial set of experiments can be summarized into several points. The MP3 compression introduces two main artifacts that degrade speech: low-pass filtering and spectral valleys. We have tried to investigate the impact of these two MP3 specific distortions separately and it was shown that the latter one influenced ASR performance more strongly, mainly due to its influence to dynamic features used standardly in ASR. Therefore, we focused on solutions that aim at reducing the depth of these valleys.

Our experiments with uniform and spectrally selective dithering proved in practice the conclusions drawn from the theoretical analysis presented in Section 2.2. After compensation, cepstral features (both MFCC and PLP) yielded lower error rates. The selective dithering method was particularly effective for GMM-based systems, where significant reduction of WER values was observed for all languages and bitrates. The results achieved with the simpler, uniform dithering were not that unambiguous. Our explanation is that in this case, the positive effect from filling the unnaturally empty spectral bands was mixed with negative influence of added noise in remaining ones. The experiments were conducted with a state-of-the-art LVCSR on speech data from three languages.

The results presented in Section 4.6.3 prove that the SSD technique can consistently outperform both adapted features and uniform dithering at the same time. In all our experiments, the only notable exception was the 12kbps German test. Another question remains: why was the improvement for English so much higher than for other two languages? Interestingly, English was also the language with the lowest baseline *WER*. This observation was somewhat contradictory to our initial expectations, where we assumed that it would be German (with the worst initial *WER*) that would benefit the most.

We have investigated also the methods that try to detect MP3 bitrates first and then they employ matched AMs trained for each bitrate setting. The results presented in Section 3 show that it works, however, the main problem is the reliable detection, mainly in cases where different or even multiple encoders were used. The latter situation is not that rare as it may seem. It often occurs in broadcast archives where some speech segments were compressed for telephone transmission first and later recompressed for archivation.

The last type of experiments focused on comparing the impacts on two recently most often used ASR architectures, GMM and DNN. A complete set of tests for all proposed schemes was performed for English only, as it was the language with the largest amount of training data. The neural nets outperformed the GMMs in general, but proved to be more sensitive to the quality of the input data in the mismatched scenario and hence less successful in case of very low bitrates. However, our results might be influenced by the available amount of training data. It is possible that with more training data, the DNNs would be more robust even in these extreme situations.

The proposed compensation techniques have been already applied in practice. They have become essential for a large national project whose aim was to process and automatically transcribe (by means of a dedicated LVCSR system) a huge audio archive collected by the Czech Radio company [32]. The archive contained almost 250.000 audio documents with total duration about 100.000 hours. We received them in the MP3 format because at the time of the archive digitization (early 2000s) the storage space had been strictly limited. The sound quality of the archive files corresponded to the times and means of their recording, however, there were quite a lot of documents from the last two decades whose poor quality had been caused by improper signal manipulation and compression. It was, e.g., the files recorded in regional studios or abroad (by international correspondents) that had been compressed at the place of their origin, then sent through digital lines (with their own codecs) and recompressed again during the final digitization process. The quality of these audio files was very low, similar to that of an MP3 signal compressed at very low bitrate, and low was also the ASR system performance. After applying the techniques described in this paper, the transcription accuracy of these recordings increased significantly, by 10 or even more per cent.

In [9] we mention another possible application of the dithering methods. They can be utilized also for those audio files that had been recorded on devices (notebooks, smartphones) that have a 'noise suppression' option. It is known that the signal manipulation techniques that employ spectral subtraction in a too extensive way can harm the performance of ASR systems. A signal that passed through a 'denoising' procedure can exhibit similar artifacts as the compressed one, i.e. deep spectral valleys. In this situation, the dithering technique is helpful, too. In [9] we demonstrated it on a large database that had been recorded on a modern notebook with the 'noise suppression' option (non-intentionally) switched on. While the original data achieved 55.4 % WER (in a GMM-HMM system), after applying the simple uniform dithering method, the *WER* was reduced to 22.8 %. When we repeated the same experiment with the new and more robust DNN acoustic model, the *WER* reduction between the original and dithered signal was also evident (from 26.2 % to 19.4 %).

6. Acknowledgements

The research described in the paper was supported by CTU Grant SGS14/191/OHK3/3T/13 "Advanced Algorithms of Digital Signal Processing and their Applications" and by Technology Agency of the Czech Republic in project no. TA04010199 called "MultiLinMedia".

7. References

[1] D. Pan, "A tutorial on MPEG/audio compression," *MultiMedia, IEEE*, vol. 2, no. 2, pp. 60–74, 1995.

[2] R. Hegemann, A. Leidinger, and R. Brito. (2011) Lame. [Online]. Available: http://lame.sourceforge.net

[3] Coding technologies. mp3pro. [Online]. Available: http://www.mp3prozone.com

[4] R. Böhme and A. Westfeld, "Statistical characterisation of mp3 encoders for steganalysis," in *Proceedings of the 2004 Workshop on Multimedia and Security*, ser. MMSEC '04. ACM, 2004, pp. 25–34.

[5] C.-M. Liu, H.-W. Hsu, and W.-C. Lee, "Compression artifacts in perceptual audio coding," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 4, pp. 681–695, 2008.

[6] R. H. van Son, "A study of pitch, formant, and spectral estimation errors introduced by three lossy speech compression algorithms," in *Acta Acustica united with Acustica*, vol. 91, 2005, pp. 771–778.

[7] K. Brandenburg, "MP3 and AAC explained," 1999.

[8] M. Borsky, P. Mizera, and P. Pollak, "Spectrally selective dithering for distorted speech recognition," in *Proc. of INTERSPEECH2015*, 2015.

[9] J. Nouza, P. Cerva, and J. Silovsky, "Adding controlled amount of noise to improve recognition of compressed and spectrally distorted speech," in *Proc. of ICASSP*, 2013, pp. 8046–8050.

[10] M. Borsky, P. Pollak, and P. Mizera, "Advanced acoustic modelling techniques in mp3 speech recognition," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2015, no. 1, 2015. [Online]. Available: http://asmp.eurasipjournals.com/content/2015/1/20

[11] S. G. S. Pettersen, "Robust speech recognition in the presence of additive noise," Ph.D. dissertation, Norwegian University of Science and Technology, 2008.

[12] C. Barras, L. Lamel, and J. Gauvain, "Automatic transcription of compressed broadcast audio," in *Proc. of ICASSP*, 2001, pp. 265–268.

[13] L. Besacier, C. Bergamini, D. Vaufreydaz, and E. Castelli, "The effect of speech and audio compression on speech recognition performance," in *Proceedings of 2001 IEEE Fourth Workshop on Multimedia Signal Processing*, 2001, pp. 301–306.

[14] P. S. Ng and I. Sanches, "The influence of audio compression on speech recognition systems," in *SPECOM 2004 - Proceedings of Conference Speech and Computer*, 2004.

[15] B. D'Alessandro and Y. Q. Shi, "MP3 bit rate quality detection through frequency spectrum analysis," in *Proceedings of the 11th ACM Workshop on Multimedia and Security*, 2009, pp. 57–62.

[16] M. Qiao, A. Sung, and Q. Liu, "Improved detection of MP3 double compression using content-independent features," in *Signal Processing, Communication and Computing (ICSPCC), 2013 IEEE International Conference on*, 2013, pp. 1–4.

[17] R. Yang, Y.-Q. Shi, and J. Huang, "Defeating fake-quality mp3," in *Proceedings of the 11th ACM Workshop on Multimedia and Security*, ser. MMSEC '09. ACM, 2009, pp. 117–124.

[18] M. Seltzer, D. Yu, and Y. Wang, "An investigation of deep neural networks for noise robust speech recognition," in *Proc. of ICASSP*, 2013, pp. 7398–7402.

[19] T. Yoshioka, S. Karita, and T. Nakatani, "Far-field speech recognition using cnn-dnn-hmm with convolution in time," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, 2015, pp. 4360–4364.

[20] L. Seps, J. Malek, P. Cerva, and J. Nouza, "Investigation of deep neural networks for robust recognition of nonlinearly distorted speech," in *Proc. of INTERSPEECH*, 2014, pp. 363–367.

[21] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi Speech Recognition Toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. Hilton Waikoloa Village, Big Island, Hawaii, US: IEEE Signal Processing Society, 2011.

[22] P. Pollak and J. Cernocky, "Czech SPEECON adult database," Tech. Rep., Nov 2003, http://www.speechdat.org/speecon.

[23] D. B. Paul and J. M. Baker, "The design for the wall street journal-based csr corpus," in *DARPA Speech and Language Workshop*. Morgan Kaufmann Publishers, 1992.

[24] T. Schultz, N. Vu, and T. Schlippe, "Globalphone: A multilingual text amp; speech database in 20 languages," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, 2013, pp. 8126–8130.

[25] P. Fousek, P. Mizera, and P. Pollak. (2014) Ctucopy feature extraction tool. [Online]. Available: http://noel.feld.cvut.cz/speechlab

[26] D. Povey, L. Burget, M. Agarwal, P. Akyazi, K. Feng, A. Ghoshal, O. Glembek, N. Goel, M. Karafiat, A. Rastrow, R. Rose, P. Schwarz, and S. Thomas, "Subspace gaussian mixture models for speech recognition," in *Proc. of ICASSP*, 2010, pp. 4330–4333.

[27] D. Povey and P. Woodland, "Minimum phone error and i-smoothing for improved discriminative training," in *Proc. of ICASSP*, 2002, pp. 105–108.

[28] A. Ghoshal, D. Povey, M. Agarwal, P. Akyazi, L. Burget, K. Feng, O. Glembek, N. Goel, M. Karafiat, A. Rastrow, R. Rose, P. Schwarz, and S. Thomas, "A novel estimation of feature-space mllr for full-covariance models," in *Proc. of ICASSP*, 2010, pp. 4310–4313.

[29] V. Prochazka, P. Pollak, J. Zdansky, and J. Nouza, "Performance of Czech speech recognition with language models created from public resources," *Radioengineering*, vol. 20, pp. 1002–1008, 2011.

[30] "Ústav českého národního korpusu (Institute of Czech National Corpus) - SYN2006PUB," Prague, 2006, http://ucnk.ff.cuni.cz/ english/syn2006pub.php.

[31] T. Schultz, "Rapid language adaptation tools for multilingual speech processing," in *Automatic Speech Recognition Understanding, 2009. ASRU 2009. IEEE Workshop on*, 2009, pp. 51–51.

[32] J. Nouza, P. Cerva, J. Zdansky, K. Blavka, M. Bohac, J. Silovsky, J. Chaloupka, M. Kucharova, L. Seps, J. Malek, and M. Rott, "Speech-to-text technology to transcribe and disclose 100,000+ hours of bilingual documents from historical Czech and Czechoslovak radio archive," in *Proc. of INTERSPEECH*, 2014, pp. 964–968.