LETTER

# Encoding Detection and Bit Rate Classification of AMR-Coded Speech Based on Deep Neural Network

**Seong-Hyeon SHIN**[†], **Woo-Jin JANG**[†], **Ho-Won YUN**[†], ***Nonmembers***, ***and*** **Hochong PARK**[†a)], ***Member***

**SUMMARY** A method for encoding detection and bit rate classification of AMR-coded speech is proposed. For each texture frame, 184 features consisting of the short-term and long-term temporal statistics of speech parameters are extracted, which can effectively measure the amount of distortion due to AMR. The deep neural network then classifies the bit rate of speech after analyzing the extracted features. It is confirmed that the proposed features provide better performance than the conventional spectral features designed for bit rate classification of coded audio.

***key words:*** *bit rate, speech codec, AMR, deep neural network, feature vector*

## 1. Introduction

In digital media communication and storage, a signal is typically processed by a codec in order to reduce the number of bits. Accordingly, there exist many signal variations due to various coding schemes. In order to investigate the coding history of the signal, it is desired to detect whether a given signal is an original or a coded signal, and to classify the bit rate of the coded signal [1]–[4]. The encoding detection and bit rate classification of coded signal can be used for detecting signal splicing and fake-quality file, and for blind assessment of signal quality [3].

Many methods for encoding detection and bit rate classification of audio signal have been reported [1]–[4]. They use high frequency information and modified discrete cosine transform (MDCT) coefficients as core features, because audio codecs generally quantize the spectral information in MDCT domain. These methods, however, cannot be applied directly to speech signal because of the different coding schemes between audio and speech codecs; the speech codec quantizes the speech parameters and excitation signal [5]. Therefore, it is required to develop new features optimized for the bit rate classification of coded speech.

In this letter, we propose a method for encoding detection and bit rate classification of coded speech based on speech-specific features, in order to investigate the coding history of speech. We consider only the Adaptive Multi Rate (AMR) speech codec, since it is the most common codec in speech communications [5]. The proposed method inputs an AMR-coded speech signal without any coding information, and classifies its bit rate into nine classes, where one class corresponds to the original and the remaining eight classes correspond to eight different bit rates of AMR.

Based on the operation of AMR, we first develop features specific to the bit rate classification of AMR-coded speech, which consist of the short-term and long-term temporal statistics of speech parameters. We then design two classifiers based on deep neural network (DNN) [6]. Finally, we measure the performance of the proposed classifiers and compare it with that of the methods using the spectral features designed for bit rate classification of coded audio.

Manuscript received July 12, 2017.
Manuscript revised September 27, 2017.
Manuscript publicized October 20, 2017.

[†]The authors are with the Department of Electronics Engineering, Kwangwoon University, Seoul, Korea.
a) E-mail: hcpark@kw.ac.kr

DOI: 10.1587/transinf.2017EDL8155

## 2. Proposed Method of Bit Rate Classification

### 2.1 Features for Bit Rate Classifier

A key step in designing a classifier is to develop the features that can effectively extract the core information that distinguishes the different classes. Figure 1 shows the overall procedure for computing the proposed feature vector. At the first stage, the frame-based speech parameters are extracted, where the sampling frequency is 8 kHz and the frame length is 20 ms. Synchronization between the parameter frame and the AMR frame is not conducted.

Since the spectral envelope of speech is distorted due to codec, the frame-based linear predictive (LP) coefficients and Mel-frequency cepstral coefficients (MFCCs) of the input are first computed in order to model the amount of distortion. The order of LP analysis is 10, and the number of LP coefficients and MFCCs is 10 and 12, respectively, where the first component of MFCCs is not included. A frame-based zero crossing rate (ZCR) of input is also computed. Then, a 23-dimensional vector $\boldsymbol{U}_f$ consisting of LP coefficients, MFCCs, and ZCR is determined as shown in Fig. 1, where $f$ is a frame index and the subscripts of [ ] refer to the number of rows.

Next, the frame-wise LP-residual of input, also referred to as a speech excitation, is computed using a 10-th order LP filter. Ideally, the speech excitation has a flat spectral envelope, but when it is incorrectly modeled due to a limited number of quantization bits, its flatness is degraded. Hence, in order to independently model the spectral flatness of speech excitation, the LP coefficients, MFCCs, and ZCR of the LP-residual are computed and are denoted by another 23-dimensional vector, $\boldsymbol{V}_f$. For each frame, $\boldsymbol{U}_f$ and $\boldsymbol{V}_f$ are merged into a 46-dimensional vector $\boldsymbol{Z}_f$, as shown in Fig. 1.

In order to analyze the temporal characteristics of signal distortion due to coding, the desired bit rate classifier

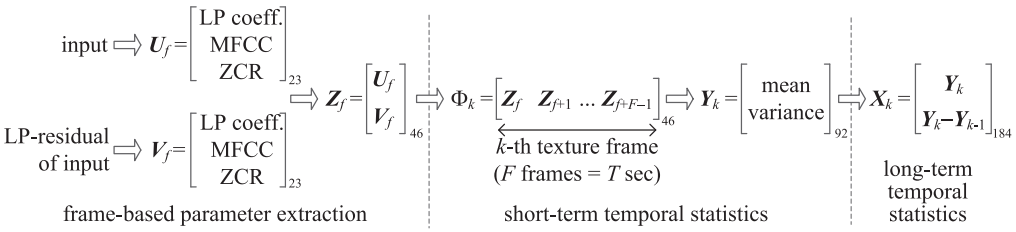


**Fig. 1** Procedure for computing the proposed feature vector $\boldsymbol{X}_k$ for each texture frame.

needs to model the temporal characteristics of the features. A neural network with recurrent states, which is called a recurrent neural network, can model the temporal characteristics inside the network [7], but has difficulty in accurately controlling the scale of temporal analysis and in providing multiple time scales. Therefore, we adopt a strategy to extract the features representing the temporal properties of the signal outside the neural network and then to input these features into a feed-forward neural network [6].

To achieve this, at the second stage, the frame-based vectors, $\boldsymbol{Z}_f$s, are aggregated into a texture frame consisting of $F$ frames, which corresponds to $T = (F \times 0.02)$ seconds. As shown in Fig. 1, $\boldsymbol{Z}_f$s for the $k$-th texture frame are denoted by a $46 \times F$ matrix, $\boldsymbol{\Phi}_k$. Then, the row-wise mean and row-wise variance of $\boldsymbol{\Phi}_k$ are computed, resulting in a 92-dimensional vector, $\boldsymbol{Y}_k$. $\boldsymbol{Y}_k$ corresponds to the short-term temporal properties because it represents the temporal statistics inside the $T$-sec-long texture frame.

Finally, at the third stage, the difference in the $\boldsymbol{Y}_k$ value between the adjacent texture frames, $\boldsymbol{Y}_k - \boldsymbol{Y}_{k-1}$, is computed. It corresponds to long-term temporal properties because it represents the inter-texture-frame properties. Therefore, as shown in Fig. 1, a 184-dimensional feature vector $\boldsymbol{X}_k$ for the $k$-th texture frame is determined. For each texture frame, the proposed classifier inputs $X_k$ and outputs the bit rate class, thereby working as a bit rate classifier every $T$ seconds.

In the proposed method, LP coefficients are used instead of line spectral pairs (LSPs) because LP coefficients are more sensitive to changes in the spectral envelope than LSPs. Then, small distortion of the spectral envelope causes a large change in LP coefficients, which makes them perform bit rate classification better than LSPs. The pitch information, which is one of the major speech features, is not included in the proposed feature vector, because the variation of pitch value between different bit rates is not significant.

### 2.2 Deep Neural Network for Bit Rate Classifier

The proposed neural network has three hidden layers with 180, 45, and 30 neurons each, and uses a sigmoid activation function [6]. When training the neural network, a dropout is used to prevent overfitting [8].

We develop two classifiers based on two different DNNs that perform different tasks for feature analysis, as shown in Fig. 2. One classifier, denoted by CL-P, is based on a DNN that outputs a class probability. This DNN, denoted by DNN-P, has nine output neurons and is trained to map the input feature vector to a 9-dimensional probability vector over nine classes at the output layer. In testing, DNN-P computes the probability of each class at the output layer and the class with the highest probability is selected as the output bit rate class.

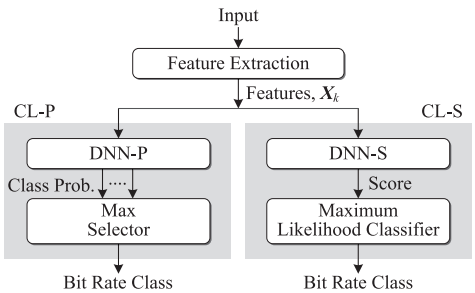


**Fig. 2** Overall structure of two proposed classifiers based on two different DNNs.

The other classifier, denoted by CL-S, is based on a DNN that outputs a single score. This DNN, denoted by DNN-S, has one output neuron and is trained to map the input feature vector to nine uniformly-spaced discrete scores between 0.0 and 1.0, corresponding to each class, at the output layer. A maximum likelihood classifier is then designed after estimating the conditional probability density function, $prob(S|C)$, of the score $S$ at the DNN-S output for each class $C$ using the training data. In testing, DNN-S computes a score $S$ at the DNN-S output, and the class $C$ with the highest $prob(S|C)$ is selected as the output bit rate class. Both networks, DNN-P and DNN-S, use the same features and have the same structure, except for the number of output neurons.

## 3. Performance Evaluation

For performance evaluation, we use the Texas Instruments and Massachusetts Institute of Technology (TIMIT) speech database (DB) [9], which consists of English sentences of about 320 minutes. Each sentence in TIMIT is coded using AMR with eight different bit rates, resulting in the final DB consisting of the original and coded sentences of about $320 \times 9 = 2880$ minutes. After deleting the mute periods, 70%, 15%, and 15% of the resulting DB are randomly selected for training, validation, and testing, respectively.

The performance of classifier varies with the texture frame length $T$, because $T$ defines the scale of temporal

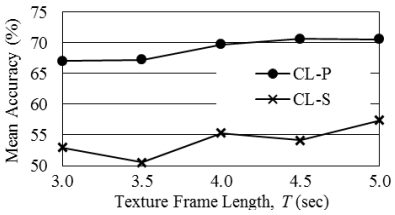


**Fig. 3** Mean accuracy of CL-P and CL-S as a function of $T$.

**Table 1** Confusion matrix of the proposed 9-class classifier for $T = 4$.

| | CL-P (MA = 69.7%) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| | orig. | 12.2k | 10.2k | 7.95k | 7.4k | 6.7k | 5.9k | 5.15k | 4.75k |
| 0 | 99.8 | - | 0.2 | - | - | - | - | - | - |
| 1 | - | 65.3 | 9.3 | 0.3 | 23.3 | - | 1.8 | - | - |
| 2 | - | 10.5 | 84.7 | 1.3 | 2.9 | 0.2 | 0.4 | - | - |
| 3 | - | 0.5 | 1.6 | 58.2 | 2.2 | 29.1 | 7.3 | - | 1.1 |
| 4 | - | 1.6 | 2.9 | 0.6 | 89.8 | - | 5.1 | - | - |
| 5 | - | - | 0.9 | 26.5 | 0.9 | 50.4 | 18.0 | 0.4 | 2.9 |
| 6 | - | 0.2 | - | 3.6 | 3.6 | 21.5 | 68.4 | 1.6 | 1.1 |
| 7 | - | 0.2 | 0.2 | 0.5 | 24.4 | 2.0 | 3.4 | 26.4 | 42.9 |
| 8 | - | - | - | 0.5 | - | 1.5 | - | 13.8 | 84.2 |
| | CL-S (MA = 55.3%) | | | | | | | | |
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 0 | 99.8 | 0.2 | - | - | - | - | - | - | - |
| 1 | - | 56.4 | 18.0 | 4.4 | 12.0 | 5.6 | 3.6 | - | - |
| 2 | - | 11.6 | 79.5 | 7.6 | 0.9 | 0.4 | - | - | - |
| 3 | - | - | 6.7 | 33.5 | 36.0 | 15.1 | 7.4 | 1.1 | 0.2 |
| 4 | - | - | 4.4 | 21.8 | 52.4 | 15.4 | 6.0 | - | - |
| 5 | - | - | 1.6 | 13.1 | 30.9 | 19.8 | 32.0 | 1.7 | 0.9 |
| 6 | - | - | - | 1.3 | 15.3 | 21.2 | 59.3 | 2.2 | 0.7 |
| 7 | - | 0.2 | 0.7 | 7.5 | 13.4 | 4.9 | 7.5 | 22.0 | 43.8 |
| 8 | - | - | - | - | 0.4 | 0.4 | 3.1 | 20.7 | 75.3 |

statistics. Figure 3 shows the mean accuracy (MA) of the two proposed classifiers over all nine classes as a function of $T$. Based on the results in Fig. 3, we select $T = 4$ as the final operating point, because both short texture frame and high MA are desired for real applications.

Table 1 shows the performance of CL-P and CL-S for $T = 4$ in more detail in the form of confusion matrices, where the rows correspond to the true class and the columns correspond to the estimated class. The class index $C = 0$ is the original and $1 \le C \le 8$ corresponds to 12.2 ∼ 4.75 kbps in the order of descending bit rates. The MA, which corresponds to the descending diagonal average of confusion matrix, of CL-P and CL-S is 69.7% and 55.3%, respectively. For both classifiers, the original class is almost perfectly classified, which means that the proposed classifiers can discriminate between the original and the coded speech.

To the best of our knowledge, studies on the bit rate classification of AMR-coded speech have not yet been reported. Hence, a direct comparison between the proposed and the previous methods is impossible. We therefore conduct an indirect comparison using the conventional spectral features originally designed for bit rate classification of coded audio, such as MDCT features and discrete Fourier transform (DFT) features. We input MDCT features and DFT features to DNN-P in Fig. 2, respectively, and train each DNN-P for bit rate classifier. In this way, the performance difference by different features, while using the same DNN-based classifier, can be measured.

**Table 2** Confusion matrix of 9-class classifier using MDCT and DFT features for $T = 4$.

| | Classifier using MDCT features (MA = 32.4%) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| | orig. | 12.2k | 10.2k | 7.95k | 7.4k | 6.7k | 5.9k | 5.15k | 4.75k |
| 0 | 99.8 | - | 0.2 | - | - | - | - | - | - |
| 1 | - | 18.3 | 34.0 | 17.1 | 2.4 | 13.5 | 10.0 | - | 4.7 |
| 2 | - | 22.4 | 34.5 | 13.6 | 2.7 | 8.9 | 11.3 | 0.2 | 6.4 |
| 3 | - | 12.7 | 16.4 | 20.5 | 2.4 | 17.3 | 19.6 | 0.2 | 10.9 |
| 4 | - | 15.1 | 16.4 | 19.2 | 3.3 | 16.4 | 16.0 | 0.3 | 13.3 |
| 5 | - | 8.7 | 9.5 | 15.6 | 1.6 | 17.1 | 26.2 | 0.2 | 21.1 |
| 6 | - | 7.1 | 7.3 | 12.0 | 1.8 | 17.4 | 29.3 | - | 25.1 |
| 7 | - | 5.3 | 3.1 | 8.5 | 0.9 | 10.2 | 16.4 | - | 55.6 |
| 8 | - | 0.5 | 0.7 | 2.2 | 0.2 | 10.0 | 17.3 | 0.6 | 68.5 |
| | Classifier using DFT features (MA = 47.7%) | | | | | | | | |
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 0 | 100.0 | - | - | - | - | - | - | - | - |
| 1 | - | 56.7 | 20.3 | 7.5 | 7.3 | 4.4 | 3.3 | 0.5 | - |
| 2 | - | 15.5 | 60.0 | 8.4 | 8.5 | 5.4 | 2.0 | - | 0.2 |
| 3 | - | 6.6 | 16.0 | 25.1 | 12.7 | 23.6 | 12.9 | 1.6 | 1.5 |
| 4 | - | 12.0 | 16.9 | 24.7 | 15.1 | 18.4 | 10.7 | 1.3 | 0.9 |
| 5 | - | 2.0 | 7.1 | 16.5 | 9.1 | 26.9 | 30.5 | 2.6 | 5.3 |
| 6 | - | 0.9 | 2.2 | 8.4 | 6.2 | 22.7 | 46.3 | 2.0 | 11.3 |
| 7 | - | 2.4 | 4.2 | 5.7 | 5.8 | 6.7 | 6.7 | 5.8 | 62.7 |
| 8 | - | - | - | - | - | 0.2 | 3.1 | 3.4 | 93.3 |

MDCT features are computed as described in [3] after applying the necessary change due to different sampling frequency. MDCT is applied to each 40ms-frame with 50% overlap, resulting in 160 MDCT coefficients for each 20ms-frame. Then, as in [3], the absolute average of each MDCT coefficient over $T$ seconds is computed. In addition, as in [3], 12 MFCCs and their first and second derivatives are computed and averaged over $T$ seconds, resulting in additional $12 \times 3 = 36$ features. In this way, a 196 (= 160 + 36)-dimensional MDCT feature vector is obtained. Another 196-dimensional feature vector based on DFT is computed in the same way as the computation of MDCT features, after replacing MDCT with DFT.

In order verify our computation of the conventional features, we first apply the MDCT features to the bit rate classifier of AAC (advanced audio coding)-coded audio and confirm that the MA of 92.3%, similar to [3], is achieved. Next, we apply the MDCT and DFT features to the bit rate classifier of AMR-coded speech for $T = 4$, and obtain the confusion matrices shown in Table 2. MDCT and DFT features can discriminate between the original and the coded speech, but they are not capable of properly classifying the bit rates because they cannot detect the slight distortion differences between different bit rates. The results in Tables 1 and 2 confirm that the proposed speech-specific featrues provide better performance in bit rate classifiction of coded speech on average than the conventional MDCT and DFT features.

The proposed feature vector $\boldsymbol{X}_k$ consists of various types of parameters: short-term and long-term statistics of LP coefficients, MFCCs, and ZCR derived from the input and another set derived from the LP-residual of input. We then evaluate the individual importance of each type of feature in order to confirm that all elements in $\boldsymbol{X}_k$ are required to provide good performance. After deleting a subset of features in interest from $\boldsymbol{X}_k$, we re-train the network using the modified $\boldsymbol{X}_k$ and measure its performance. The amount of

**Table 3** Mean accuracy of CL-P after deleting subset of features for $T = 4$.

| Deleted feature subset | MA (%) | MA decrease (% point) | Deleted feature subset | MA (%) | MA decrease (% point) |
|---|---|---|---|---|---|
| None(0) | 69.7 | - | LP coeff(80) | 59.6 | 10.1 |
| LP-resid(92) | 53.9 | 15.8 | ZCR(8) | 59.9 | 9.8 |
| Mean(92) | 54.0 | 15.7 | Long-term(92) | 63.1 | 6.6 |
| Variance(92) | 58.0 | 11.7 | MFCC(96) | 65.7 | 4.0 |

**Table 4** Confusion matrix of 4-class classifier for $T = 4$.

| | CL-P (MA = 91.2%) | | | | CL-S (MA = 90.7%) | | | |
|---|---|---|---|---|---|---|---|---|
| | *ORIG* | *H* | *M* | *L* | *ORIG* | *H* | *M* | *L* |
| *ORIG* | 100.0 | - | - | - | 99.6 | 0.4 | - | - |
| *H* | - | 85.6 | 14.4 | - | - | 85.9 | 14.0 | 0.1 |
| *M* | - | 2.3 | 96.4 | 1.3 | - | 4.7 | 92.3 | 3.0 |
| *L* | - | 0.3 | 17.1 | 82.6 | - | 0.5 | 14.7 | 84.8 |
| | MDCT features (MA = 66.6%) | | | | DFT features (MA = 82.6%) | | | |
| | *ORIG* | *H* | *M* | *L* | *ORIG* | *H* | *M* | *L* |
| *ORIG* | 100.0 | - | - | - | 100.0 | - | - | - |
| *H* | - | 36.3 | 60.1 | 3.6 | - | 62.7 | 37.0 | 0.3 |
| *M* | - | 11.1 | 77.0 | 11.9 | - | 7.4 | 87.4 | 5.2 |
| *L* | - | 1.5 | 45.3 | 53.2 | - | 2.4 | 17.3 | 80.3 |

performance decrease then becomes a measure of the importance of the corresponding feature subset.

Table 3 shows the MA of CL-P when each subset of features is deleted, where the number in the parenthesis shows the number of deleted features. For example, 'LP-resid(92)' means that, when deleting 92 features derived from the LP-residual of input, the MA decreases by 15.8% points, which confirms the necessity for the LP-residual features. From Table 3, therefore, we can conclude that there is no redundancy in the proposed features and thus no subset of features can be deleted to reduce the number of features while maintaining the same performance.

The difference between the signal distortions in adjacent bit rates is very small, especially in low rates. Therefore, the bit rate classification into nine classes might be too strict. We therefore re-define four classes of bit rates: *ORIG* (original) class corresponding to $C = 0$; *H* (high) class corresponding to $C = 1$ and 2; *M* (mid) class corresponding to $C = 3$, 4, 5, and 6; and *L* (low) class corresponding to $C = 7$ and 8. We then design two 4-class classifiers based on DNN-P and DNN-S, as shown in Fig. 2, by re-training the DNNs for 4-class classification. Table 4 shows the confusion matrices of the resulting two 4-class classifiers, CL-P and CL-S, for $T = 4$. Both have relatively high accuracy compared to the 9-class classifier. CL-P is still superior to CL-S on average, but the difference in performance between the two classifiers decreases, compared to the 9-class classifier, due to the fewer classes.

We also design a 4-class classifier based on DNN-P using MFCC and DFT features, and the performance is given in Table 4. The proposed speech-specific features still provide better performance than the conventional MFCC and DFT features.

## 4. Conclusion

A method is proposed for a bit rate classification of AMR-coded speech. We develop new features specific to coded speech that can effectively distinguish the different bit rates of speech, which include the short-term and long-term temporal statistics of LP coefficients, MFCCs, and ZCR. In this way, for each texture frame, a 184-dimensional feature vector is defined. We design two bit rate classifiers based on DNN that analyze the feature vector in a different way. It is confirmed that the bit rate classifier using the proposed features has better performance than that using MDCT and DFT features designed for bit rate classification of coded audio.

## Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (NRF-2016R1D1A1B03930923).

### References

[1] B. D'Alessandro and Y.Q. Shi, "MP3 bit rate quality detection through frequency spectrum analysis," Proc. 11th ACM Workshop on Multimedia and Security, pp.57–61, 2009.
[2] T. Bianchi, A. De Rosa, M. Fontani, G. Rocciolo, and A. Piva, "Detection and classification of double compressed MP3 audio tracks," Proc. 1st ACM Workshop on Information Hiding and Multimedia Security, pp.159–164, 2013.
[3] D. Luo, W. Luo, R. Yang, and J. Huang, "Identifying compression history of wave audio and its applications," ACM Trans. Multimedia Computing, Communications and Applications, vol.10, no.3, pp.30:1–30:19, 2014.
[4] D. Seichter, L. Cuccovillo, and P. Aichroth, "AAC encoding detection and bitrate estimation using a convolutional neural network," Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, pp.2069–2073, 2016.
[5] 3GPP TS 26.090, "Adaptive multi-rate (AMR) speech codec," June 2002.
[6] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol.521, no.7553, pp.436–444, 2015.
[7] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio, "How to construct deep recurrent neural networks," arXiv preprint arXiv:1312.6026, 2013.
[8] N. Srivastava, G. Hinton, A. Krizhevsky, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," J. Machine Learning Research, vol.15, no.1, pp.1929–1958, June 2014.
[9] J.S. Garofolo, L.F. Lamel, W.M. Fisher, J.G. Fiscus, and D.S. Pallett, "DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1," NASA STI/Recon Technical Report N, vol.93, 1993.