INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

Fricative Phoneme Detection Using Deep Neural Networks and its Comparison to Traditional Methods

Metehan Yurt[1,†], *Pavan Kantharaju*[1], *Sascha Disch*[1], *Andreas Niedermeier*[1], *Alberto N. Escalante-B*[2], *Veniamin I. Morgenshtern*[3]

[1]Fraunhofer IIS, Erlangen, Germany
[2]WS Audiology, Erlangen, Germany
[3]Chair of Multimedia Communications and Signal Processing, Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, Germany

{metehan.yurt, pavan.kantharaju, sascha.disch, andreas.niedermeier}@iis.fraunhofer.de, alberto.escalante@wsa.com, veniamin.morgenshtern@fau.de

Abstract

Accurate phoneme detection and processing can enhance speech intelligibility in hearing aids and audio & speech codecs. As fricative phonemes have an important part of their energy concentrated in high frequency bands, frequency lowering algorithms are used in hearing aids to improve fricative intelligibility for people with high-frequency hearing loss. In traditional audio codecs, while processing speech in blocks, spectral smearing around fricative phoneme borders results in pre and post echo artifacts. Hence, detecting the fricative borders and adapting the processing accordingly could enhance the quality of speech. Until recently, phoneme detection and analysis were mostly done by extracting features specific to the class of phonemes. In this paper, we present a deep learning based fricative phoneme detection algorithm that exceeds the state-of-the-art fricative phoneme detection accuracy on the TIMIT speech corpus. Moreover, we compare our method to other approaches that employ classical signal processing for fricative detection and also evaluate it on the TIMIT files coded with AAC codec followed by bandwidth limitation. Reported results of our deep learning approach on original TIMIT files are reproducible and come with an easy to use code that could serve as a baseline for any future research on this topic[*].

Index Terms: Fricative phonemes, Phoneme detection, Machine learning, Deep neural networks, TIMIT speech corpus

1. Introduction

Since the advent of technology for processing and transmitting speech signals, engineers and linguists have been identifying ways to improve the intelligibility of speech. One approach has been to identify the different categories of phonemes in a given language and process them based on their signal characteristics using the classical signal processing methods. This has proven to be beneficial in applications such as digital coding of speech and signal processing in hearing aids [1]. However, this relies on signal characteristics to be very similar for phonemes of a particular class. With fricative phonemes, this can lead to difficulties in detecting voiced fricatives (such as /v/, /dh/, /z/ and /zh/) due to their slightly tonal structure as compared to the noise-like structure common for other fricative phonemes. With the recent achievements in deep learning, we observed significant improvements in the detection and classification accuracy of speech recognition and natural language processing. This motivated us to tackle the fricative detection problem using neural networks in order to further improve on the state of the art. In this paper, we improve our previous work [2] and compare our results with other fricative detection methods. Additionally, we show that our method is successful even on degraded and bandwidth limited speech signals.

[†] Some of the work was done during the master thesis at WS Audiology.
[*] https://github.com/yurtmete/FriDNN

1.1. Fricative phonemes

Phonemes are perceptually distinct units of sound in a language that help distinguish one word from another. Fricative phonemes belong to the group of unvoiced phonemes along with affricatives. While the spectrogram of a speech signal reveals a neat harmonic structure for voiced phonemes, the unvoiced phonemes are more noise like without any tonal structure. Due to their particular structure, the start and stop positions of fricatives in speech can be identified. In this paper, we use the standard TIMIT ARPABET phonetic codes[1] (/s/, /sh/, /f/, /th/, /z/, /zh/, /v/ and /dh/) when we refer to the fricative phonemes [4].

1.2. Comparison to alternative fricative detection methods in literature

S-Transform and Linear Prediction Coefficients (LPC) based features have been used with a threshold detector [5], achieving an 83.81% recall on fricatives. Combining LPC features with Spectral Weighting was investigated in [6] and an 85.3% recall on fricative detection was reported. Combining time domain and frequency domain features (zero-crossing rate (ZCR), first and second peak frequencies, and band energy ratio (BER)) was suggested in [7]. Using a linear discriminant classifier a 94.08% unweighted average recall (UAR) was achieved. However we believe that this result was optimistic as voiced fricative phonemes (i.e., /v/, /dh/, /z/ and /zh/) were excluded from the test set where /v/ and /dh/ are known to be hard to detect due to their vowel like structure. Additionally, the SA sentences were not excluded from the test set (see Section 2.3) and non-causal post processing was applied after the detection in [5], [6] and [7]. This makes their evaluation setup easier than ours since we do not apply non-causal post-processing to the detection and do not include SA sentences in the test set for our baseline signal processing and deep learning approaches (see Section 3.4 for a fair comparison.).

[1]See Table 2.1 in [3] for ARPABET to IPA mapping.

1.3. Alternative models

Deep Neural Networks (DNN) have been applied to the estimation of phoneme posterior probabilities recently. Mel-Frequency Cepstrum Coefficients (MFCC) were fed to single layer [8] and deep fully connected neural networks [9] and [10]. However, MFCC based features were calculated and cascaded for 9 to 11 frames (single frame was 25 ms long), where the posterior was calculated for the frame in the middle. For instance, taking 9 frames input resulted 52.5 ms delay ($(8 \times 10 + 25)/2$), assuming frames were 10 ms apart (10 ms is the typical shift among frames in MFCC based future calculation). Hence, the method relied on contextual information (looked into the future) and proved to be successful [11]. Same procedure was applied to Mel-frequency filterbank-based features and the total context duration was 310 ms, which means using 155 ms of future samples from the frame that the posterior is calculated for [12].

Recurrent Neural Networks (RNNs) have been considered for phoneme recognition tasks, and shown to be successful when they were used bidirectionally [13], [14] and [15]. However, that requires complete context of the frame being detected, that is, both previous and latter frames are needed. For this reason, all of the investigations mentioned so far that use fully connected networks and bidirectional RNNs may not be applicable for audio codecs and hearing aids, since they introduce significant delay.

We investigate 1D CNNs as they proved to be useful to extract problem specific low-level features from time domain audio samples, and led to better phoneme posterior probability estimation compared to MFCC features [16], [17] and [18]. The difference of our model is that it does not use fully connected layers and has smaller context size as compared to the previous investigations: the minimum context size was 250 ms among [16], [17] and [18].

2. Network architecture and training

2.1. Network architecture

We employ 1D CNN with residual connections [19]. The network takes mono speech signal segments in time domain at 16 kHz sampling frequency as inputs, and calculates the posterior probabilities for the event that the sample at the center of the segment belongs to three phoneme classes we define below in Section 2.3. Table 1 shows the structure of the model we used. The input stage of the network takes raw speech segments of 320 samples (20 ms). The input stage is followed by 2 convolutional stages. Stage 1 consists of 1 convolutional layer and stage 2 consists of 6 stacked convolutional layers. For each convolutional stage, the first number in the square brackets indicates the kernel size, the number after "/" indicates the stride value, and the last number indicates the number of filters used. The strides are only applied to the first convolutional layer of each stage for downsampling and the values for strides are found empirically. In stage 2 we use residual connections [19]. After the convolutional stages, 1D global average pooling is used. All of the layers have ReLU activation, except the three neurons at the output stage, which have the softmax activation. The total number of parameters in this model is 113 283.

2.2. TIMIT dataset

The TIMIT Speech Corpus consists of raw speech sentences (at 16 kHz sampling rate) and their corresponding time-aligned phonetic labels [4]. It contains a total of 6300 sentences, 10 sentences spoken by each of 630 speakers from 8 major dialect regions of the United States. We used the entire training set of the TIMIT dataset for training (462 speakers), and the core test set for testing (24 speakers). We created the validation set (50 speakers) following the methodology in [3].

Table 1: *Network architecture*

Stage	**Description**
Input	Raw speech segment with shape [320,1]
Convolutional Stage 1	[32/6, 48]
Convolutional Stage 2	[8/3, 48] x 6
Global Average	1D global average pooling
Output	Three-neuron softmax

2.3. Data preprocessing

In the TIMIT dataset, there are two sentences that are read by all the speakers; these are called SA sentences. We excluded them from all of the speakers in all sets (training, validation and test) to prevent a possible bias in the evaluation. Specifically, if SA sentences were not excluded, the model might conceivably memorize the phonemes in these sentences using the training set, and then make near perfect detection for these sentences in the test set. This would make the model appear to perform better than it does in reality. After the exclusion of SA sentences, there were 8 sentences per speaker left.

To keep the training and validation set balanced, we ensured an approximate 50:50 fricative-to-non-fricative sample ratio. To do this, we extracted 16 randomly placed 20 ms (320 in samples) long speech segments from each sentence: 8 segments labeled as fricative phoneme and 8 segments labeled as non-fricative phoneme. Recall that the label for a segment is determined based on the identity of the middle sample of the segment.

Non-fricative segments were also divided into two classes: voiced and unvoiced non-fricative segments. The former corresponded to actual speech samples and the latter corresponded to non-speech silence samples. Phonemes /h#/, /epi/, /pau/, /bcl/, /dcl/, /gcl/, /pcl/, /tcl/, /kcl/ were considered as unvoiced non-fricative segments. Note that the closure phonemes (/bcl/, /dcl/, /gcl/, /pcl/, /tcl/, /kcl/) were also included in unvoiced non-fricative segments.

For each epoch, our training set consisted of 59136 (462 speakers x 8 sentences x 16 random segments) speech segments placed randomly as explained above. From epoch to epoch the training set of 59136 segments was randomly regenerated from the training set of the TIMIT dataset.

We applied the same procedure to form our validation set, but we kept it fixed for different training runs and also for different epochs in same training run. Our validation samples consisted of 6400 (50 speakers x 8 sentences x 16 segments) speech segments.

For each segment from training, validation and test sets, we applied standard deviation normalization,

$$x = x/\sqrt{var(x)} \quad (1)$$

where x is the extracted segment with 320 samples and $var(x)$ is the variance of the segment.

2.4. Network training

We used categorical cross-entropy loss and Adam optimizer starting with a 0.001 learning rate. We halved the learning rate if the validation loss did not decrease within 10 epochs, and we applied early stopping if the loss computed on the validation set did not decrease within 40 epochs. For convolutional layers, we applied l2 regularization with the factor of 0.0001 to all the weights, and applied batch normalization after each linear filtering operation and before each non-linearity.

3. Evaluation and results

We evaluated the performance of our detection algorithm on the core test of the TIMIT dataset that contained 192 sentences with 24 different speakers (each speaker had 8 sentences after the exclusion of two SA sentences as explained in Section 2.3). The total duration of the test set was 583 seconds.

Figure 1 explains how we used the trained network to make inferences. The speech signal in time domain is displayed in blue in the top display. We take a sliding window that consists of 320 samples (20 ms), represented in black in the top display. The window is moved from left to right, one sample at a time. For each location of the sliding window, we apply the network to the speech segment marked by the window. This gives us the posterior probabilities for the event that the sample *at the center* of the sliding window belongs to the three classes (fricative, voiced non-fricative, and unvoiced non-fricative segments). Hence, our system incurs 10 ms delay, which as discussed in Section 1.3 is much shorter than the minimum delay of 52.5 ms among mentioned methods [8], [9], [10] and [12]. The posterior probabilities for fricative segments are displayed in blue in the bottom display.

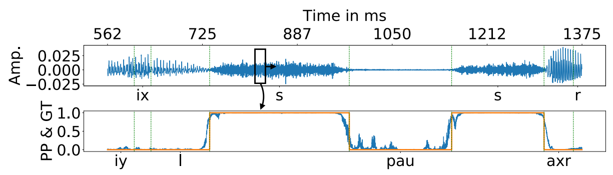

Figure 1: *Top: Time domain signal of the sentence SI756 by the speaker FELC0 in the TIMIT core test set in blue and input to the network in the black box. Amp. indicates the amplitude of the time domain signal. Bottom: Posterior probabilities (PP) for fricative segments calculated by the network in blue and the ground truth (GT) in orange. Phoneme labels are indicated in the bottom x-axes and time in ms is indicated in the top x-axis.*

We make the final decision based on the posterior probabilities produced by the network in the final three-neuron softmax layer. Table 2 (a) summarizes the overall performance on the core test set.

Note that in Table 2 voiced and unvoiced non-fricative segments are represented in non-fricative class and unweighted average is the arithmetic mean of the metrics over two classes. We present here the unweighted averages since the core test set is highly unbalanced (see number of samples for each class in Table 2).

We also break down the recall for fricatives in Table 2 (a) for unvoiced and voiced fricatives as well as each fricative phoneme separately. Recall for unvoiced fricatives is 92.76%, where /s/, /sh/, /f/, /th/ have the following recalls individually: 97.52, 95.31, 84.22, 66.30 in %. Recall for voiced fricatives is 86.89%, where /z/, /zh/, /v/, /dh/ have the following recalls individually: 96.39, 79.25, 78.89, 66.85 in %.

3.1. Performance on perceptually coded speech signals

Telecommunication applications such as VOIP telephony, internet radio etc., use audio and speech codecs to perceptually encode and transmit speech at a lower data rates as compared to the plain Pulse Code Modulated (PCM) signals. However, depending on bandwidth availability, encoding at lower bitrates lead to coding noise thereby lowering the speech intelligibility. Phoneme detection and enhancement methods can be used to enhance the speech intelligibility [20]. The network defined in Section 2.1 was evaluated with input that was perceptually coded. The well known AAC codec was used for this purpose [21]. The core test set was encoded using AAC-LC encoder at 32 kbps and decoded using the corresponding AAC-LC decoder. The trained network from the original TIMIT dataset as detailed in Section 2.1 was used as is without additional training.

The results in Table 2 (b) shows comparable accuracy in fricative phoneme detection on the perceptually coded speech to the accuracy in the uncoded case. This demonstrates that the neural network in Section 2.1 can be used to accurately detect fricatives even on perceptually coded speech.

3.2. Performance on bandwidth limited speech signals

The limited bandwidth of a narrow-band telephony also limits the speech intelligibility. There are several bandwidth enhancement techniques currently in use to improve the quality of speech. Processing of such bandlimited speech signals at phoneme levels can further improve speech intelligibility. We applied the trained neural network for fricative detection on the 4 kHz low pass filtered version of the core test set that were perceptually coded (as in Section 3.1) and achieved the results in Table 2 (c). Results before "/" indicates the performance without any further training on coded and bandlimited signals. As evident from the results, the network performs rather poorly. Therefore, the original TIMIT training and validation sets were perceptually coded and low pass filtered to 4 kHz bandwidth and the network was retrained. This led to a considerable improvement in detection accuracy (see results in Table 2 (c) after "/"). This shows that with an inclusive and well balanced training set, the network is capable of detecting fricatives even on bandlimited signals.

3.3. Comparison with a baseline—a signal processing based fricative detection algorithm

Classic signal processing techniques are popular for detecting and processing the fricative phonemes (see Section 1.2). This involves using the signal characteristics specific to the fricative phonemes: ZCR [22], Gradient Index (GI) [23] and high frequency band to low frequency BER [24]. While ZCR and GI are calculated in time domain, BER is a frequency domain feature. Fricatives, due to their noise like structure, display relatively high ZCR and GI as compared to non-fricatives. Also, with most of the energy of a fricative being present in the high frequencies, the high frequency band to low frequency BER is generally larger for fricatives. Figure 2 shows the features for a speech segment.

The features for fricative detection were implemented in MATLAB and the class conditional probability values for fricative and non-fricative class were obtained for each feature on the

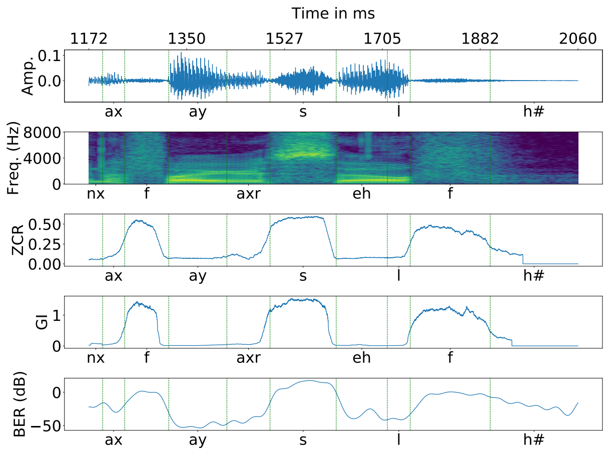

Figure 2: *Top to bottom: Time domain signal, spectrogram, ZCR, GI and BER values for the SI1669 sentence by the speaker MDAB0 in the TIMIT core test set.*

TIMIT training set. Principal Component Analysis (PCA) [25] was then used to represent the features accurately in a lower dimensional space before applying the Naive Bayesian classifier. The Maximum A-Posteriori (MAP) is calculated to classify the given samples into fricative and non-fricative classes. The algorithm was evaluated on the same core test set as used for our network from Section 2.1 and the performance is shown in Table 2 (d).

Table 2: *Precision, Recall and F1-Score in %. Note that there are* 7 920 188 *(495 sec) non-fricative and* 1 407 811 *(88 sec) fricative samples in the core test set.*

Class	**Precision**	**Recall**	**F1-Score**
(a) Deep learning model performance on original core test set			
non-fricative	98.34	94.57	96.42
fricative	74.88	91.00	82.16
unweighted average	86.61	92.79	89.29
(b) Deep learning model performance on the AAC coded core test set without re-training			
non-fricative	98.16	93.88	95.97
fricative	72.35	90.07	80.24
unweighted average	85.25	91.98	88.11
(c) Deep learning model performance on bandlimited (4 kHz) core test set without (before "/") and with (after "/") retraining			
non-fricative	87.96 / 97.86	98.56 / 92.99	92.96 / 95.37
fricative	74.87 / 69.20	24.11 / 88.57	36.47 / 77.70
unweighted average	81.42 / 83.53	61.34 / 90.78	64.72 / 86.53
(d) Performance of the baseline algorithm on original core test set			
non-fricative	96.47	87.96	92.02
fricative	54.75	81.91	65.63
unweighted average	75.61	84.94	78.83

3.4. Comparison to the state of the art

The best reported fricative detection performance was 94.08% UAR [7]. As discussed in Section 1.2, the voiced fricative phonemes were excluded from the test set and non-causal post-processing were applied to the detections. To reach a fair comparison we also applied the same procedure and achieved 95.89% UAR.

4. Computational considerations

Our deep learning approach requires around 4.2 million (4 177 170) FLOPs for a single inference. The baseline signal processing based algorithm requires about 50k FLOPs; a lot less than the deep learning model. This suggests that our baseline signal processing approach is more suitable for product-ready applications, whereas, our deep learning based approach can be used for applications such as audio restoration in offline mode etc., where detection accuracy is of essence as opposed to being real time capable.

Our network from Section 2.1 is currently non-optimized for complexity as the focus was to investigate if we could achieve better fricative detection accuracy than the state-of-the-art approaches using neural networks. However, there are many ways to make neural networks lightweight so that they become more suitable for low power devices: quantization aware training, post-training quantization of weights and biases, weight pruning [26], [27], knowledge transfer through distillation [28], and using more memory efficient convolutional layers [29], [30] to name a few. We plan to investigate these methods to optimize our network for complexity as part of our future work on this topic.

5. Conclusions

In this paper, we provide a state-of-the-art baseline for fricative phoneme detection using machine learning based on neural networks. The 1D CNN based fricative detection can achieve 92.79% UAR on the TIMIT core test set and we believe that this is the best performance reported on fricative detection for TIMIT dataset to date. We showcase the network performance on perceptually coded speech signals and also on band-limited speech signals. We further elaborate on alternate fricative detection methods and compare our neural network based fricative detection method against traditional methods. This paper demonstrates that the neural network based detection can outperform the traditional fricative detection methods and can also be applied to detect fricatives in real world speech signals. The results of our deep learning model on the original TIMIT files are fully reproducible and we make the complete source code available. We also include the trained model that was used in the evaluations in this paper[*].

6. Acknowledgments

This work has been supported by the Bavarian Ministry of Economic Affairs, Regional Development and Energy in the DSAI project. The authors would like to thank Dr. Christian Dittmar from Fraunhofer IIS for his devoted support.

[*] https://github.com/yurtmete/FriDNN

7. References

[1] J. D. Robinson, T. Baer, and B. C. Moore, "Using transposition to improve consonant discrimination and detection for listeners with severe high-frequency hearing loss: La utilización de la transposición para mejorar la discriminación consonántica y la detección en oyentes con hipoacusia severa para frecuencias agudas," *International Journal of Audiology*, vol. 46, no. 6, pp. 293–308, 2007.

[2] M. Yurt, A. N. Escalante-B., and V. I. Morgenshtern, "Fricative phoneme detection with zero delay," 2020. [Online]. Available: https://openreview.net/forum?id=BJxlmeBKwS

[3] A. K. Halberstadt, "Heterogeneous acoustic measurements and multiple classifiers for speech recognition," Ph.D. dissertation, Massachusetts Institute of Technology, 1999.

[4] 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, p. 27403, 1993.

[5] H. K. Vydana and A. K. Vuppala, "Detection of fricatives using S-transform," *The Journal of the Acoustical Society of America*, vol. 140, no. 5, pp. 3896–3907, 2016.

[6] H. K. Vydana and A. K. Vuppala, "Detection of fricative landmarks using spectral weighting: A temporal approach," *Circuits, Systems, and Signal Processing*, pp. 1–24, 2020.

[7] D. Ruinskiy and Y. Lavner, "A multistage algorithm for fricative spotting," in *XXII Annual Pacific Voice Conference (PVC)*. IEEE, 2014, pp. 1–5.

[8] S. M. Siniscalchi, D. Yu, L. Deng, and C.-H. Lee, "Exploiting deep neural networks for detection-based speech recognition," *Neurocomputing*, vol. 106, pp. 148–157, 2013.

[9] D. Yu, S. M. Siniscalchi, L. Deng, and C.-H. Lee, "Boosting attribute and phone estimation accuracies with deep neural networks for detection-based speech recognition," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2012, pp. 4169–4172.

[10] I.-F. Chen, S. M. Siniscalchi, and C.-H. Lee, "Attribute based lattice rescoring in spontaneous speech recognition," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 3325–3329.

[11] J. Pinto, B. Yegnanarayana, H. Hermansky, and M. Magimai-Doss, "Exploiting contextual information for improved phoneme recognition," in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2008, pp. 4449–4452.

[12] S. M. Siniscalchi, T. Svendsen, and C.-H. Lee, "A bottom-up modular search approach to large vocabulary continuous speech recognition," *IEEE transactions on audio, speech, and language processing*, vol. 21, no. 4, pp. 786–797, 2012.

[13] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural networks*, vol. 18, no. 5-6, pp. 602–610, 2005.

[14] J. C. Vásquez-Correa, P. Klumpp, J. R. Orozco-Arroyave, and E. Nöth, "Phonet: A tool based on gated recurrent neural networks to extract phonological posteriors from speech." in *INTERSPEECH*, 2019, pp. 549–553.

[15] J. Franke, M. Mueller, F. Hamlaoui, S. Stueker, and A. Waibel, "Phoneme boundary detection using deep bidirectional LSTMs," in *Speech Communication; 12. ITG Symposium*. VDE, 2016, pp. 1–5.

[16] D. Palaz, R. Collobert, and M. M. Doss, "Estimating phoneme class conditional probabilities from raw speech signal using convolutional neural networks," *arXiv preprint arXiv:1304.1018*, 2013.

[17] D. Palaz, M. M. Doss, and R. Collobert, "Convolutional neural networks-based continuous speech recognition using raw speech signal," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4295–4299.

[18] D. Palaz, M. Magimai-Doss, and R. Collobert, "End-to-end acoustic modeling using convolutional neural networks for hmm-based automatic speech recognition," *Speech Communication*, vol. 108, pp. 15–32, 2019.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[20] L. M. Arslan and J. N. L. Hansen, "Minimum cost based phoneme class detection for improved iterative speech enhancement," in *Proceedings of ICASSP '94. IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. ii, 1994, pp. II/45–II/48 vol.2.

[21] "Information technology Generic coding of moving pictures and associated audio information Part 7: Advanced Audio Coding (AAC)," International Organization for Standardization, Geneva, CH, Tech. Rep., 2006.

[22] S. Kay and R. Sudhaker, "A zero crossing-based spectrum analyzer," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 1, pp. 96–104, 1986.

[23] W. M. Bernd Iser, Gerhard Schmidt, *Bandwidth Extension of Speech Signals*. Springer Verlag, 2008.

[24] A. M. Kondoz, *Digital Speech - Coding for Low Bit Rate Communication Systems*. John Wiley and Sons, 2004.

[25] G. Dougherty, *Pattern recognition and classification: an introduction*. Springer Science & Business Media, 2012.

[26] Y. LeCun, J. S. Denker, S. A. Solla, R. E. Howard, and L. D. Jackel, "Optimal brain damage." in *NIPS*, vol. 2. Citeseer, 1989, pp. 598–605.

[27] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding," *arXiv preprint arXiv:1510.00149*, 2015.

[28] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[29] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size," *arXiv preprint arXiv:1602.07360*, 2016.

[30] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.