INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

Optimizing Voice Activity Detection for Noisy Conditions

Ruixi Lin, Charles Costello, Charles Jankowski, Vishwas Mruthyunjaya

CloudMinds Technology Inc., Santa Clara, CA 95054
`charles.jankowski@cloudminds.com`

Abstract

In this work, we focus our attention on how to improve Voice Activity Detection (VAD) in noisy conditions. We propose a Convolutional Neural Network (CNN) based model, as well as a Denoising Autoencoder (DAE), and experiment against acoustic features and their delta features in noise levels ranging from SNR 35 dB to 0 dB. The experiments compare and find the best model configuration for robust performance in noisy conditions. We observe that combining more expressive audio features with the use of DAEs improve accuracy, especially as noise increases. At 0 dB, the proposed model trained with the best feature set could achieve a lab test accuracy of 93.2% (averaged across all noise levels) and 88.6% inference accuracy on device.We also compress the neural network and deploy the inference model that is optimized for the app so that the average on-device CPU usage is reduced to 14% from 37%.

Index Terms: Voice Activity Detection, Convolutional Neural Networks, Denoising Autoencoders

1. Introduction

Voice Activity Detection (VAD) is a crucial first step in any speech processing system [1]. It determines when a speaker is talking to the system, and consequently which segments of audio the system should analyze. The primary difficulty in developing VAD systems is distinguishing between audio from the speaker and background noise. In the past, rule-based and machine learning approaches leveraging speech signal properties and acoustic features have been designed to build VAD modules. As the requirement of building more robust models to noise conditions are raised in recent years, deep learning techniques are being widely used to extract data-driven features from noisy waveforms or acoustic features. All these works emphasize the reliability of selected features, and evaluate the quality of models under various noisy conditions. To take a more rigorous look at the usefulness of the various feature aspects, including neural denoising techniques, acoustic features like log mel-filterbanks and Mel-Frequency Cepstral Coefficients (MFCCs), we propose a convolutional neural network based method along with a pre-trained Denoising Autoencoder (DAE), and focus on comparing how individual features and feature sets perform in different noise settings, then draw the best model and feature set, especially in high noise (SNR<20 dB) environments.

In section 2, we discuss related prior works on the problem of VAD, and summarize the aspects of the problem this work address. In section 3, we describe the training and test data sets. In section 4, we detail the proposed neural network models, both for denoising and classification. In section 5 and 6, we present experimental and on-device test findings and analyze the results. Finally, in section 7, we draw conclusions and provide an outlook on future work.

2. Prior Work

Traditional VADs such as ETSI AMR VAD Option 2 [2] and G.729B [3] have proposed to use parameters like frame energies of different frequency bands, SNR of background, channel, and frame noise, differential zero crossing rate, and thresholds at different boundaries of the parameter space for decision making. The problems of the parameters come with noise and lower SNRs. Data-driven approaches have been used in recent works. Deep Belief Network (DBN) [4] that extracts the underlying features through nonlinear hidden layers and connects with a linear classifier can obtain better VAD accuracies than G.729B. The use of acoustic features combined with SVM approaches have further shown improvements in noisy conditions [5, 6, 7]. Deep neural networks have been proved to capture temporal information, approaches that feed MFCC or Perceptual Linear Prediction (PLP) features to feed-forward deep neural networks (DNNs) and recurrent neural networks (RNNs) have been studied in [8, 9, 10, 11]. DNNs coupled with stacked denoising autoencoders, with greedy layerwise pre-training have also been proposed [12]. Multi-layer perceptrons (MLPs) trained using acoustic features have also been explored for this task in [13]. In addition, as a part of the DARPA RATS program, GMM and neural network techniques with selected feature combinations are designed to adapt to unseen noisy channels for performing speech activity detection[14, 15].

This work follows the prior works closely, but it brings forth a systematical analysis of how different feature sets allow for more robust performance of VAD in noisy conditions. Our particular contributions are as follows:

- Improving VAD performance by CNNs combining with DAEs, MFCCs or filterbanks, and their combinations in noisy conditions
- Providing a comparison of two optimization frameworks for VAD model deployment on device towards lower CPU usage

3. Training and Test Data

Our goal in this work is to develop a VAD system that is robust to noisy conditions. To this end, we chose to use the AISHELL 1 (AISHELL) Chinese Mandarin speech corpus [16] mainly, and manually labeled beginnings and ends of voice frames. The signal-to-noise ratio (SNR) of the original data set is 35 dB. To facilitate model testing in real-world like noise environments, additive background noises are added to the raw waveforms to simulate a variety of SNRs ranging from 0 dB to 20 dB. The noise waveforms were collected by our company from sites at the Mobile World Congress (MWC) Expo, recorded at sampling rate of 16kHz.

Ruixi Lin is now with National University of Singapore. Charles Costello is now with Plus One Robotics, Boulder, CO. Vishwas Mruthyunjaya is now with Aisera, Palo Alto, CA.

http://dx.doi.org/10.21437/Interspeech.2019-1776

Table 1: *DAE Architecture, where FS denoted the feature size*

Layer	Shape	Details
Input	$(21 \times FS)$	n/a
Encoder 1	(500)	ReLU
BatchNormalization	(500)	n/a
Encoder 2	(256)	ReLU
BatchNormalization	(500)	n/a
Decoder 1	(500)	None
Decoder 2	$(21 \times FS)$	None

AISHELL is 178 hours long, and covers 11 domains. The recording was done by 400 speakers from different accent areas in China. We create 4 noisy data sets at SNR 0, 5, 10, and 20 dB, and separate each into train, development, and test sets.

4. Technical Background

We develop a convolutional neural network (CNN) for VAD, and a front-end Denoising Autoencoder (DAE) [17] to remove background noise from input speeches. To further explain why this CNN as well as a DAE topology is selected, we would like to emphasize that the method is in line with the ideas of using neural networks to extract robust features. The hidden layers of the bottleneck DAE allows for learning low-level representation of the corrupted input distribution. The CNN follows closely with the DNNs for VAD, but its convolution and pooling operations are more adept at reducing input dimensions. In addition, the denoised input is fed for the training and inference of the CNN classifier. Therefore, the CNN with DAE would benefit from the ability of recovering corrupted signals and hence enhancing representation of features, thus providing robustness.

4.1. Denoising Autoencoder

In this work, we employ a 2-layer bottleneck network for the DAE, and set the encoding layers hidden unit sizes to 500 and 256. We use the *ReLU* activation function for the encoder, followed by batch normalization, and no activation function or normalization is applied for the decoder. The DAE is trained layer-wise using standard back-propagation. The objective function is root mean squared error between clean data $\{\mathbf{x}_i\}_{i=1}^{N}$ and decoded data $\{\tilde{\mathbf{x}}_i\}_{i=1}^{N}$, as defined as below.

$$\mathcal{J}(\theta, \theta') = \min_{\theta, \theta'} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(\theta, \theta'; \mathbf{x}^{(i)}, \tilde{\mathbf{x}}^{(i)}) \qquad (1)$$

where $\mathcal{L}(\cdot)$ represents the loss, θ and θ' denote the encoding weights and biases, and the decoding weights and biases respectively.

The training data consists of 32 hours of noisy data and clean data. The noisy data is a combined data set of SNRs 0, 5, 10, and 20 dB. For the clean counterpart, we use the original data which is very clean and has a SNR of 35 dB.

The model is pre-trained with MFCC features and Filter-Bank features. Each frame of features is concatenated with its left and right frames, by a window of size 21. The DAE architecture is summarized in Table 1, where FS denotes the feature size.

Table 2: *CNN Architecture, where FS denotes the feature size and BS denotes the training batch size*

Layer	Dimensions	Details
Input	$(BS, 21, FS, 1)$	n/a
Convolution	$(BS, 19, FS-2, 64)$	3x3
Max_Pooling	$(BS, 9, (FS-2)/2, 64)$	2x2
Dropout 1	$(BS, 9{\times}(FS-2)/2 \times 64)$	0.5
Flatten	$(BS, 9{\times}(FS-2)/2 \times 64)$	n/a
Dense 1	$(BS, 128)$	n/a
Dropout 2	$(BS, 128)$	0.5
Dense 2	$(BS, 2)$	n/a

Table 3: *Test accuracy (%) of CNNs using MFCC features on AISHELL; It is noted that Combined denotes using DAE and $\Delta, \Delta\Delta$*

SNR (dB)	**35**	**20**	**10**	**5**	**0**
MFCC	96.93	95.76	92.49	88.91	83.97
MFCC + DAE	97.00	95.67	92.89	90.50	86.03
MFCC, $\Delta, \Delta\Delta$	97.16	95.79	93.04	90.33	84.73
MFCC + Combined	**97.16**	**96.01**	**93.24**	**91.90**	**87.90**

4.2. CNN

We propose a frame-level CNN with frame-based input features (denoised by DAEs) and labels $\{\mathbf{u}_i, y_i\}_{i=1}^{N}$. As each input frame is windowed by its neighboring 10 left and right frames, forming a 21-frame windowed input, a 2D convolutional kernel would be natural to use to reduce input size. We apply (3, 3) convolutional filters, (2, 2) max pooling strided by (2, 2), dropout, flatten, reduce the flattened features to a fully-connected output, and then compute the logits. The network is trained in mini-batches using back-propagation, to minimize the sparse softmax cross entropy loss between labels $\{y_i\}_{i=1}^{N}$ and the argmax of the last layer logits, denoted by$\{y_i^{'}\}_{i=1}^{N}$. The loss function is defined below.

$$\mathcal{H}_{y'}(y) = -\sum_{i=1}^{N} y'_i log(y_i) \qquad (2)$$

Table 2 shows our CNN model architecture. In inference time, we apply a post-processing mechanism to CNN outputs for more accurate estimations. Details will be discussed in section 6. As before, FS denotes the feature size. BS denotes the training batch size.

5. Experiments and Results

5.1. Evaluation Metric

Our ground truth is manually labeled speech/non-speech frames, and we evaluate the test performance using frame-based accuracy. In the following sub-sections, test accuracy will be presented in tables. For notations in the table,

5.2. Feature Sets and Results

For VAD, we process the training waveforms with a 25 ms wide window, advance the waveform with a sliding window of 10 ms, and extract 13-dimensional MFCC features at 16 kHz sam-

Table 4: *Test accuracy (%) of CNNs using filterbank features on AISHELL; It is noted that Combined denotes using DAE and* $\Delta, \Delta\Delta$

SNR (dB)	35	20	10	5	0
FBank	95.81	92.26	88.11	83.92	78.13
Norm. FBank	96.32	93.46	88.65	87.63	84.04
FBank + DAE	95.71	93.20	89.02	84.63	79.74
Norm. FBank + DAE	95.67	93.24	89.18	85.04	82.80
FBank, $\Delta, \Delta\Delta$	96.43	92.81	88.92	86.63	73.77
Norm. FBank, $\Delta, \Delta\Delta$	**96.56**	**94.82**	**90.30**	**88.47**	**85.24**
FBank + Combined	95.04	90.21	88.71	83.88	80.62
Norm. FBank + Combined	95.85	92.83	89.82	85.78	82.25

Table 5: *A Comparison on test accuracy (%) of different approaches; It is noted that Combined denotes using DAE and* $\Delta, \Delta\Delta$

Data	Model	10 dB	5 dB	0 dB
AURORA2 (English)	G.729B	72.02	69.64	65.54
	SVM$_r$	85.21	80.94	74.26
	MK-SVM	85.38	82.30	75.59
	DBN	86.63	81.85	76.66
	DDNN	86.98	82.30	76.85
	MFCC + Combined	**87.68**	**86.02**	**78.35**
AISHELL (Chinese)	MFCC + Combined (16kHz)	93.24	91.90	87.90
	MFCC + Combined (8kHz)	93.64	92.53	92.52
	MFCC + Combined (16kHz(from 8kHz))	**96.14**	**94.19**	**93.67**

pling rate. Likewise, we convert the raw waveforms into 40-dimensional log mel-filterbank (filterbank) features. The filterbank features are then normalized to zero mean and unit variance per utterance based. Additionally, to explore whether more expressive features would help, we use Δ and $\Delta\Delta$ features together with their MFCC or filterbank features. Besides, the input features could be denoised using pre-trained DAE. Same operations are performed for development data and test data. We ran experiments 3 times for each feature set and recorded the average test accuracy.

Tables 3 and 4 show VAD test results of CNNs trained with different feature sets on the AISHELL dataset. In Table 3, the first row draws baseline accuracy results of using 13 MFCCs only, and unsurprisingly, the accuracy drops as noise level of the speeches increases. The second and third row illustrate that either the use of DAE or 39 MFCCs + Δ and $\Delta\Delta$ features would help improve the results, especially in noisier conditions. The last row adopts a combined approach of using both DAE and deltas, and the accuracy turns out to be better than all rows above.

Compared to the results with MFCCs, the results of filterbanks as shown in Table 4 look more interesting. First of all, 4 sets of additional experiments with normalized filterbank features are added to the 4 feature sets with filterbank features. It is clearly observed that normalized features works better then unnormalized features. Secondly, significant improvements provided by using deltas are found in both normalized and unnormalized filterbank features, with normalized filterbank + $\Delta, \Delta\Delta$ being the best accuracy feature configuration, as seen in row 6. What is a little unexpected is that DAE exhibits limited improvements on the normalized filterbank features, and the combined approach did not depict the most effective improvements. An explanation is that we kept the exact same DAE architecture for both MFCCs and filterbanks during training for fair comparisons, but for filterbank features, whose dimensionality is larger than MFCCs, a deeper autoencoder should be more preferable.

Above all, MFCCs generally outperform filterbanks on this VAD task despite of different feature schemes.

5.3. Comparisons to Contemporary Approaches

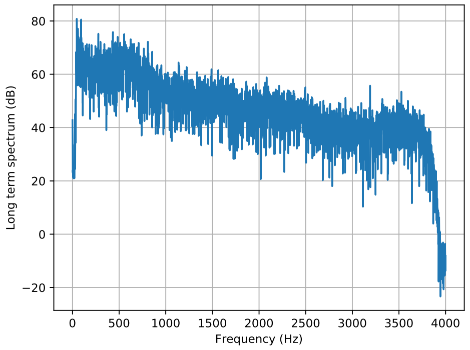

Figure 1: *Long-term spectrum (dB) of MWC noise*

We mainly compare frame-based VAD test accuracies of our best model on Mandarin AISHELL (in row 7 to 9 of Table 5) with previous approaches and more recent neural network methods on an English data set AURORA 2 (in row 1 to 5 of Table 5). The comparison on accuracy for SNR 10, 5, and 0 dB are recorded. Moreover, language difference could play an important role and render very different results when it comes to building models with acoustic features. The languages of AISHELL and AURORA 2 data differ, as a result we also run experiments on AURORA 2 and report the results in row 6 of Table 5 with the best model of our proposed method. In terms of the details of experiments for row 6, we follow the same choice of utterances and a similar train test split scheme as described in [4]. 4004 (1001 at each level) utterances at clean and three different SNR levels, added with airport noise, are used for train, development, and test at 10 dB, 5 dB, and 0 dB, with the proposed DAE and VAD methods. The utterances are derived from the AURORA 2 database, CD 3, Test Set B [18]. For the corresponding frame-based VAD ground truths, the generated reference VAD labels from [19] are used.

From the fair comparison point of view, we notice that sampling rate of AURORA 2 is 8 kHz, which differs from AISHELL (16 kHz). In an effort to make fairer comparisons, we chose to downsample AISHELL to 8 kHz to apply the same filtering as AURORA 2 and provide result comparisons, and then upsample to 16 kHz to perform additional experiments, as the whole framework is built in 16 kHz. Those results are presented in row 7 and 8 of Table 5, with original results shown in row 6. Notwithstanding the difference in sampling rates, AISHELL and AURORA 2 are essentially similar in speech qualities, variety of speakers, and more importantly noise types, where the MWC noise is similar to the airport noise added to AURORA 2 data. To illustrate this point, we plot the long-term spectrum of the MWC noise used in this work at 8 kHz sampling rate in Figure 1, to compare with the long-term spectrum of the airport noise used for experiments in row 1 to 5 presented in the AURORA 2 paper [18], and we found that the two spectrum plots look very similar.

Some highlights of the comparisons are drawn to our attention. The G.729B model and its VAD accuracy as reported in [4] delineates a baseline. Overall, neural network methods like DDNN outperform SVM based methods, which is discussed in [12]. At all three SNRs, the best model is the proposed CNN/MFCC + combined features model on both AURORA 2 and AISHELL data, and the accuracy increases by 2% to 4% especially at lower SNRs like 5 dB or 0 dB. An analysis for our model to outperform the DDNN where both models used denoising techniques is that, first of all, DDNN may suffer a slight performance degradation from the greedy layer-wise pretraining of a very deep stacked DAE, even though the denoising module is fine-tuning on the classification task, and secondly, the convolution and pooling of a CNN leads to a considerable merit over a DNN of handling combined features in the higher layers, especially when some amount of noise still exists in the input speech features, and this is better than fully-connected DNN which handles features in the lower layers [20]. The selection of speech features also contributes to a performance difference. MFCC, Δ, $\Delta\Delta$ features are helpful in extracting the dynamics of how MFCCs change over time, which were not used by the DDNN. Another important finding lies in the language difference of data. As the results suggest, VAD of AISHELL could be an easier task compared to that of AURORA 2, where AISHELL results exhibit a roughly 10% higher accuracy score compared to AURORA 2 results. Therefore, the high VAD accuracy on AISHELL from row 7 to 9 is a combined effort of both the proposed model and the data. To further study the potential influence of languages on VAD, we will focus on cross testing for models trained with multilingual data sets in the future work. Moreover, an interesting side finding from row 7 to 9 is that, as the signals are downsampled and then upsampled, the accuracy goes up instead of going down as expected due to a loss of higher band information. This could be explained by the fact that the low-pass filters provide a smoothing effect, which consequently reduced frame-by-frame errors.

6. On-device CPU Usage and Accuracy

From the system integration and deployment point of view, we benchmark the CNN/MFCC VAD model performance on Android devices. We emphasize on lowering CPU usage of the VAD app by means of neural network compression, and measure the CPU usage and accuracy on two mobile phones, either with Qualcomm Snapdragon 820 or 350 chipsets (Kryo CPU, Adreno 530 GPU, Hexagon DSP).

6.1. Implementation

For optimized on-device app deployment, we select two neural network compression frameworks to compress and deploy our models, including TensorFlow Mobile (TFM) [21] and Qualcomm Snapdragon Neural Processing Engine (SNPE) SDK [22]. The main idea of the app using either TFM or SNPE modules is to produce an estimate of when speech is present, smooths those estimates with averaging, then thresholds that average to come up with a crude speech/non-speech estimate. Specifically, the module consists of a recorder and a detector, where the recorder uses a bytebuffer to store 10 × 160 frames (16 kHz samples/sec and 10 ms frame rate) of 100 ms of waveform, calculate MFCCs and form 21-frame windows, and send that to the detector. The delay of the detector is thus approximately 210 ms. The softmax score (from 0 to 1) every 10 ms is smoothed by a moving average. The resulting average is then compared against a confidence threshold to come up with a binary estimate of speech/nonspeech.

Table 6: *CPU usage with accuracy in brackets*

Snapdragon	820	835
TFM	40% (89.04%)	34% (87.25%)
SNPE	23% (88.30%)	15% (89.98%)

6.2. CPU Usage

CPU usages are depicted in Table 6, with accuracy written in brackets. For testing under different noise conditions, the averages of CPU usage of all levels are recorded. Using these frameworks, our model achieves an average of 28% CPU usage on-phone, where using TFM (or TF Lite, a default way for model optimization in TensorFlow) would result in an average of 37% CPU usage across the two Snapdragon chip versions, and using SNPE would obtain an average of 19% CPU usage. Furthermore, we observe that SNPE is a more designated platform for reducing CPU usage on these Snapdragon based devices, and using SNPE could achieve an average reduction of 18% (37% - 19%) of CPU usage compared to using TFM. Meanwhile, averaging the 4 CPU usages shown in table, we obtained the average on-device inference accuracy of 88.6%, which is in line with our experimental results. Unfortunately, we did not get a latest Snapdragon 845 device for experiments, but we believe that with more advanced hardware, our model can be optimized to achieve even lower CPU. What is more, this model could also run on GPU and DSP, and the compressed model can be further quantized within the two frameworks. All of these will be explored in future works.

7. Conclusions

We have drawn comparisons on a CNN based VAD model using different feature sets in noisy conditions on multiple languages. We have observed that using a CNN adding DAE with MFCC, Δ, and $\Delta\Delta$ features is most helpful for improving VAD performance in high noise. With the considerable amount of parameters used in the network, deploying the model on device could face high CPU usage. To tackle the problem of high CPU usage, we have optimized the inference model with neural network compression frameworks. As for future work, we will further our analysis by interpreting the results using end-to-end metrics like utterance FA/FR, adding statistical evidence for scores which were not provided and discussed due to space limitations, design post-processing methods to improve the frame-by-frame VAD accuracy, and experiment with optimization methods on the model size for mobile deployments. Moreover, we would also like to study how the CNN and DAE model with different feature sets would perform on unseen noise types.

8. References

[1] N. S. K. J. Sohn and W. Sung, "A statistical model-based voice activity detection," in *IEEE SIGNAL PROCESSING LETTERS*, vol. 6, NO. 1, 1999.

[2] "Digital cellular telecommunications system (phase 2+); voice activity detector (vad) for adaptive multi rate (amr) speech traffic channel; general description," 1999.

[3] ITU, "Itu-t recommendation g. 729 annex b: A silence compression scheme for use with g. 729 optimized for v. 70 digital simultaneous voice and data applications," in *Int. Telecommun. Union, volume=35, no. 9*, 2000, pp. 64–73.

[4] X. L. Zhang and J. Wu, "Deep belief networks based voice activity detection," in *IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING*, vol. 21, no. 4, 2013.

[5] J. H. C. J. W. Shin and N. S. Kim, "Voice activity detection based on statistical models and machine learning approaches," in *Comput. Speech Lang*, vol. 24, no. 3, 2010, pp. 515–530.

[6] J. Wu and X. L. Zhang, "Efficient multiple kernel support vector machine based voice activity detection," in *IEEE Signal Process. Lett.*, vol. 18, no. 8, 2011, pp. 466–499.

[7] X. L. Zhang and J. Wu, "Linearithmic time sparse and convex maximum margin clustering," in *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 6, 2012, pp. 1669–1692.

[8] G. Ferroni, R. Bonfigli, E. Principi, S. Squartini, and F. Piazza, "Neural networks based methods for voice activity detection in a multi-room domestic environment," in *Proc. of EVALITA as part of XIII AI*IA Symposium on Artificial Intelligence*, vol. 2, 2014.

[9] F. Vesperini, P. Vecchiotti, E. Principi, S. Squartini, and F. Piazza, "Deep neural networks for multi-room voice activity detection: Advancements and comparative evaluation," in *2016 International Joint Conference on Neural Networks (IJCNN)*, 2016.

[10] T. Hughes and K. Mierle, "Recurrent neural networks for voice activity detection," in *ICASSP*, 2013.

[11] I. Ariav, D. Dov, and I. Cohen, "A deep architecture for audio-visual voice activity detection in the presence of transients," in *Signal Processing*, vol. 142, 2018, pp. 69–74.

[12] X. L. Zhang and J. Wu, "Denoising deep neural networks based voice activity detection," in *arXiv:1303.0663v1*, 2013.

[13] J. Dines, J. Vepa, and T. Hain, "The segmentation of multi-channel meeting recordings for automatic speech recognition," in *INTERSPEECH*, 2006.

[14] G. Saon, S. Thomas, H. Soltau, S. Ganapathy, and B. Kinsbury, "The ibm speech activity detection system for the darpa rats program," in *INTERSPEECH*, 2013.

[15] S. Thomas, S. Ganapathy, G. Saon, and H. Soltau, "Analyzing convolutional neural networks for speech activity detection in mismatched acoustic conditions," in *ICASSP*, 2014.

[16] H. Bu, J. Du, X. Na, B. Wu, and H. Zheng, "Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline," in *Oriental COCOSDA 2017, submitted*, 2017.

[17] P. Vincent, H. Larochelle, Y. Bengio, and P. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 1096–1103.

[18] H. Hirsch and D. Pearce, "The aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions," in *ISCA ITRW ASR2000*, 2000. [Online]. Available: http://dnt.kr.hsnr.de/aurora/download/asr2000_final_footer.pdf

[19] "Low-complexity variable frame rate analysis for speech recognition and voice activity detectionlow-complexity variable frame rate analysis for speech recognition and voice activity detection processing," vol. 4, no. 5, 2010, pp. 798–807.

[20] O. Abdel-Hamid, A. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," in *IEEE/ACM TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING*, vol. 22, no. 10, 2014.

[21] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, R. Jozefowicz, Y. Jia, L. Kaiser, M. Kudlur, J. Levenberg, D. Man, M. Schuster, R. Monga, S. Moore, D. Murray, C. Olah, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Vigas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "Tensorflow: Large-scale machine learning on heterogeneous systems," 2015.

[22] Qualcomm, "Snapdragon neural processing engine now available on qualcomm developer network," 2017.