Neural Network-Based Speaker Identification for Low-Power Devices

Skanda Koppula

Supervisors: Dr. James Glass, Professor Anantha Chandrakasan

I. Introduction

Consumer devices using speech interfaces are growing in number and complexity. Small devices such as wearables, personal assistants, robots, and phones boast extensive voice-controlled interfaces that transact identity-specific data and are linked to personal profiles. Misuse of speech interfaces to maliciously execute illegitimate commands on such devices has been repeatedly demonstrated [1], [2]. In a particularly egregious example reported by popular press, a crafted TV commercial was able to activate voice-controlled personal assistants and deliver a voice command to execute an online purchase [3]. There is a strong motivation for speech interfaces linked to user profiles and private data to be simultaneously capable of speaker identification.

In the task of speaker identification (SID), a device learns the speech patterns unique to each of its owners, developing a model to distinguish between these identities and identities not in its set of owners (‘the universe). The device can subsequently use this model to perform forward inference, and identify whether an input voice command is from an authorized or a malicious party. Text-dependent SID refers to models trained to recognize persons speaking a specific keyword (e.g. ‘OK Google). In contrast, text-independent models are able to distinguish a speaker for any spoken input.

Voice interfaces are common in small devices, where speech is an simple, intuitive avenue for user-device interaction. This presents a challenge: small devices are constrained in their power consumption, and relatedly, their memory capacity. Traditional SID algorithms consume on the upwards of hundreds of megabytes for model storage alone, well exceeding the limitations of many IoT devices.

This work focuses on developing an inference architecture for text-independent speaker identification that is applicable in scenarios constrained by power and latency, as is often the case in

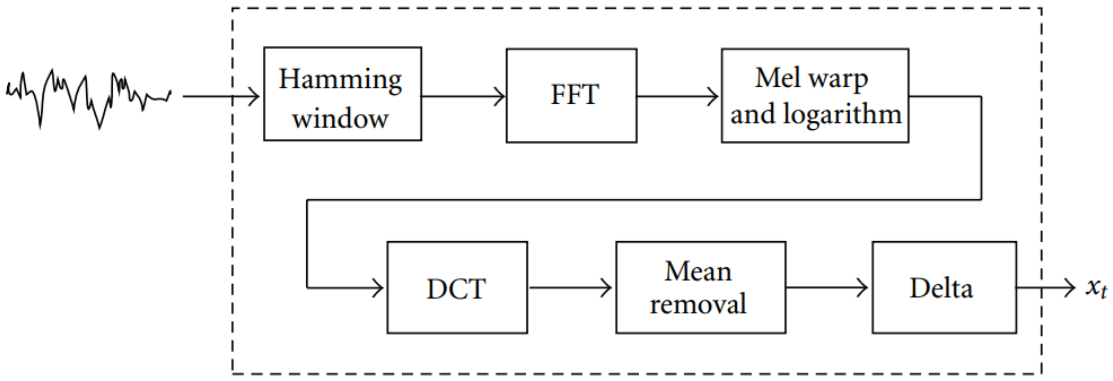

Fig. 1. Example DSP pipeline for MFCC feature extraction. (1) Hamming windowing is first applied to avoid spectral leakage during the FFT (2) FFT is applied to capture frequency information (3) Mel filterbank is applied to the log-spectrum (4) Discrete Cosine Transform is used to extract real-number coefficients describing the envelope of the spectrum (5) The coefficients are mean-normalized (6) Optionally, the rate of change of coefficients (deltas) are appended to the coefficients to capture dynamic, contextual behavior of the time-varying signal. [5]

small devices. We will design new methods for speaker identification that achieve state-of-the-art accuracy but exhibit much smaller model sizes, to fit the constraints imposed by low-memory and low-power accelerators. We also aim to demonstrate the workability and comparative efficiency of this approach by implementation of our work in an FPGA.

In this proposal, we will first overview the background of speaker identification and neural network based speech inference (Section II). Second, we will detail the thesis's main objectives and salient subtasks (Section III). Then we will then discuss related research and how this thesis builds on those prior works (Section IV). We conclude with a thesis timeline (Section V).

II. Preliminaries

A. *Front-End: Feature Extraction and VAD*

SID systems rely on pre-processing the raw waveform to capture salient information about the power spectrum of the utterance. Commonly used in SID are Mel-Frequency Cepstral Coefficients (MFCCs), features that capture the envelope of the frequency spectrum of an input waveform. To capture information emulating human perception of a waveform, MFCC extraction applies a Mel filterbank to the log-frequency spectrum; a Mel filter emulates filtering of frequencies based on frequency filtering performed by the human cochlea [4]. Figure 1 details the DSP pipeline for extracting MFCC features from an input waveform.

Non-voiced parts of the input signal provide no information for SID classification and serve to add noise and confuse backend classifiers. Nearly all text-independent SID systems apply voice-activity detection (VAD) to filter out segments of the waveform for which no human speech is detected. Pre-processing with VAD has been shown to produce marked improvements in automatic speaker recognition systems (ASR) and SID accuracy [6]. Additionally, in the context of ASR and SID hardware accelerators, integrating VAD allows designers to power-gate the speech classifier circuits, lowering the devices total power draw. An example of this was demonstrated for ASR by Price et al. in 2016 [7].

There are three main kinds of VAD filters: energy-based, harmonicity-based, and modulation-frequency based. In low SNR conditions, triggering activation when the signals energy exceeds some threshold noise-floor is often sufficient to indicate the presence of speech. Two more robust methods of voice detection leverage the acoustic periodicity of human speech. Harmonicity-based (H) VAD measures the harmonics periodicity of the input signal. Modulation frequency (MF) based VAD measures the temporal rate of change of energy across different frequency bands. These measurements are fed into an upstream neural network classifier trained to detect voice activity with these inputs. A comparison of these approaches can be found in [7]. We intend on re-using RTL source from [8] and [7] to implement a VAD that uses all three methods.

B. Traditional Back-End: Gaussian Mixture Models and i-vectors

After feature vector extraction and VAD, SID systems rely on one of many algorithmic backends to drive classification of MFCCs. State-of-the-art systems frequently use of Gaussian Mixture Models (GMMs) due to its ability to mimic a wide variety of MFCC distributions. While we will not be using GMMs in this work, we briefly describe the method to familiarize an interested reader with the algorithmic benchmarks to which we will compare against. As we will discuss in Section IV, most attempts at low-resource SID have used GMM-based backends.

In a GMM-UBM system, the universe of all speakers is first modeled using a GMM (the universal background model, ‘UBM). To learn the parameters of the UBM, expectation-maximization (EM) algorithm is used to fit the UBM to a public corpora of speech data covering thousands of speakers [9]. When the model wishes to learn a new ‘enrollee speaker, it shifts the UBM to better fit the enrollees input utterances using a few iterations of EM, forming a speaker-specific model. During evaluation, the system compares the likelihood of an input test utterance under the UBM versus under the speaker specific model [10].

State-of-the-art systems modify this simple evaluation procedure slightly. A ‘test-speaker GMM is trained using the input test utterance and a few iterations of EM. The distance between a speaker-specific model and the test-speaker model are compared using the concatenation of the Gaussian Mixture means (‘supervectors). Smaller cosine distances between supervectors indicates greater speaker similarity [11].

Unfortunately, storing, training, and comparing supervectors is computationally intensive and works poorly in practice: for a 2048-mixture GMM of 45 dimensions, the supervector dimensionality is extremely large: $45 \times 2048 = 92160$. Thus, a low-dimensionality intermediate representation of the supervector, the *i-vector*, was formed to capture the difference between a UBM and a speaker-specific model in a low-dimensional subspace. For brevity, we leave a complete description of more optimized i-vector training, using sufficient statistics, to [12]. It is important to know, that transforming MFCCs to i-vectors, and classifying i-vectors using PLDA is currently state of the art [13] [1]

Unfortunately, i-vector algorithms are not particularly amiable for low-resource SID evaluation. For example, with the standard i-vector size of 200, the i-vector extraction model requires 145 MB of storage [2]. Even reducing the precision from 32-bit float to 8-bit (at an 8

C. Contemporary Back-End: Neural Networks

Neural networks demonstrate promise as an alternative back-end MFCC classifier. In addition to achieving state-of-the-art in image recognition, language translation, and drug discovery [18]–[20], end-to-end neural networks have demonstrated record-breaking performance in a number of speech related tasks: speech recognition, language recognition, and text-dependent speaker identification [21]–[23]. In this work, we intend to use recurrent and feedforward neural network architectures as our SID backend model.

Particularly attractive about a deep neural networks (DNNs) is the powerful ability to compress model size while experiencing little to no loss in accuracy. Through a combination of pruning, quantization, and parameter encoding, compression ratios of up 510x have been demonstrated and implemented in FPGA for image recognition networks [24]. In speech recognition, 20x

[1]An alternative approach to train i-vectors is to use sufficient statistics of neural network posteriors instead of GMM statistics. This has been shown to result in modest accuracy gains. We leave the details to [14].

[2]These models were trained using the popular speech toolkit Kaldi using publically available training scripts on the RSR and SRE corpora [15]–[17]

reduction in sizes have been achieved (from 51 MB to 3.7 MB) using weight-matrix factorization [25]. In both these cases, no loss of accuracy was experienced. Additionally, architecture level optimizations (redesigning the network structure) provide another avenue for model compression; we demonstrate that this affords an addition 10x reduction in our speaker identification network size. Unlike i-vector models, which have resisted attempts to reduce their memory bandwidth and computational overhead, neural networks show promise as a candidate for a model that would fit in SRAM [26].

III. Problem Statement and Tasks

The goal of this work is to develop an digital inference architecture for text-independent speaker identification that is applicable in scenarios constrained by power, memory, and latency. We plan to demonstrate our algorithms and architecture in FPGA. By using various model optimizations, we aim to achieve a system that is comparable to or exceeds i-vector SID systems in recognition accuracy.

The core of this work is two-fold. First, we will design and train a low-depth, small footprint SID network that can be efficiently evaluated on resource-constrained devices (such as a low-power accelerator). Secondly, we will be developing generalizable hardware designs to perform inference on the network we develop.

In the process, there are a number of subtasks that the project will need to hit:

1) As noted by [7] and [26], one challenge of low-power network inference is keeping the network size small enough to fit in SRAM; of primary focus is model size, for which we will need to design and benchmark a competitive SID network less than 8 MB. For this we will experiment with new network architecture designs (e.g. using long short-term memory networks), edge pruning [26], and weight quantization [27]. One particularly promising network architecture design we found during experimentation is the use of attention to direct a small recurrent network to focus on the most informative parts of an input signal [28] (Figure 3).
2) We will study the effects of forcing fixed-point, integer arithmetic on the recognition accuracy of our SID network. The signed floating-point parameters typically used in networks complicates digital logic, so we seek to operate in a fixed-point, integer world. In small experiments, our rounded model experienced drops in recognition rates from 3-5%, requiring us to retrain the network in the fixed-point parameter domain. To compensate for

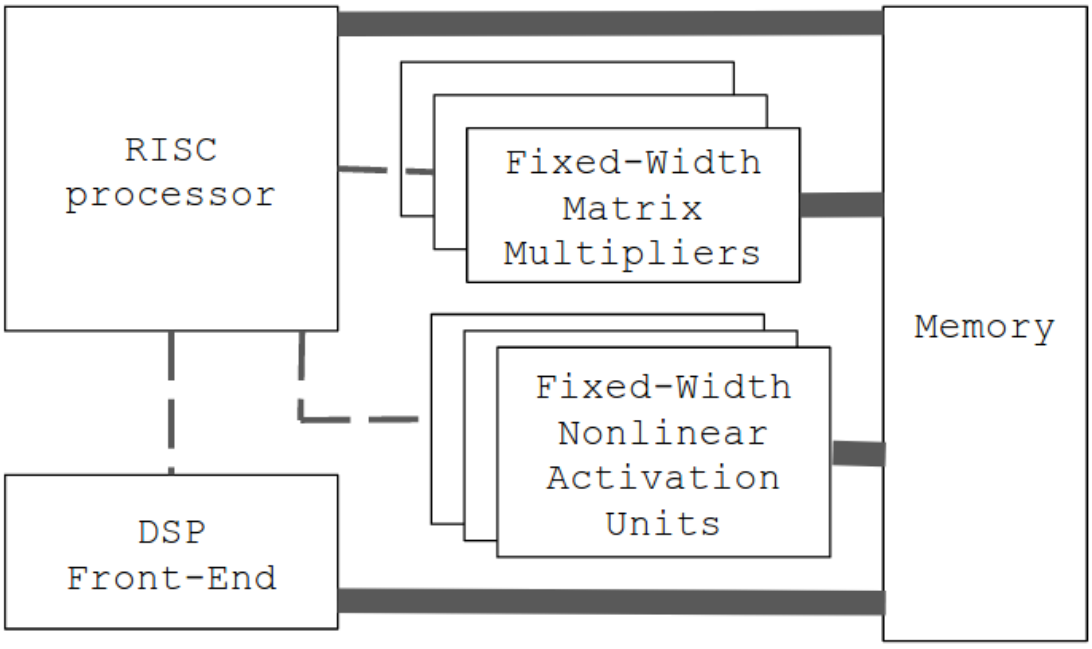

Fig. 2. Inference architecture for re-programmable networks for SID. Thick edges between components represent data lines, dashed lines represent control signal lines.

accuracy losses, we will also experiment in applying Dense-Sparse-Dense training methods [29].

3) Our goal is for our digital architecture to be generalizable; if we decide to change the SID network architecture post-synthesis and placement, we should be able to do so. This would involve writing a small compiler to translate network architectures to RISC instructions that execute on a RISC processor that orchestrates the correct sequence of commands to other components in the design (e.g. matrix multipliers, non-linear activation units, etc.). Figure 2 illustrates the target digital architecture. As is currently implemented, the network architecture is hard-coded in RTL.
4) Some effort has been made to integrate phonetic information into the SID i-vector systems. Such work attempts to normalize out intra-speaker phonetic variation, and solely capture inter-speaker variation [22], [30]. Additions such as bottleneck features and DNN-based sufficient statistics have been proposed to the basic i-vector system allowing for modest gains in accuracy. We have considered applying the same phoneme-integration techniques to our network model; the cost of this would be increased model size. We intend to

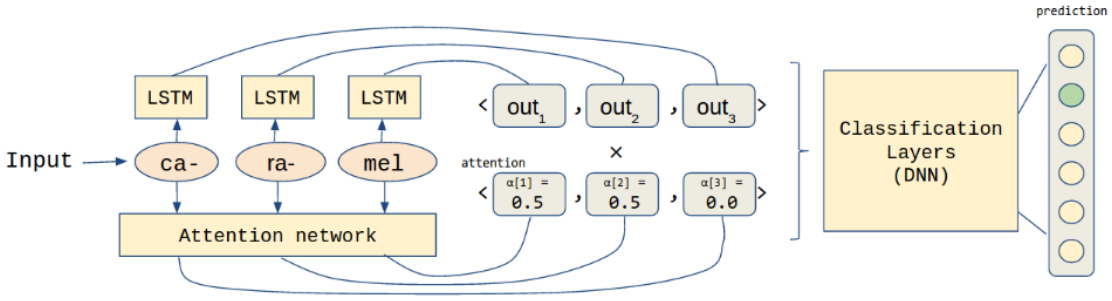

Fig. 3. Network architecture with small recurrent LSTM module and an attention mechanism. Based on various SID benchmarks, weve found this topology to have a promising model complexity-accuracy ratio.

run small benchmarks to understand the model complexity-performance ratio of such an addition.

5) Last but not least, we will need to demonstrate designs for front-end processing components (for which there is prior work [7], [8], [31]), and integrate this component into our network evaluation designs.

IV. Prior Work

A. Neural Network Based SID

Several groups have studied the application of end-to-end neural networks for SID. For text-dependent SID, [23] and [32] have demonstrated network architectures that exceed the accuracy of i-vector systems. These networks, however, exceed 70 MB in size, and not suitable for direct adaptation [3]

Prior art on neural-network based text-independent SID is a bit more sparse. [33] describes a Siamese network for text-independent speaker verification (the one-vs-universe case of speaker identification). The authors report mixed accuracy benchmarks as compared to i-vector systems on a proprietary dataset. In this work, we aim to achieve across the board accuracy gains against the i-vector system on standard open SID datasets.

[3]Interestingly, both these models have been deployed to consumer electronics, in the Google Home personal assistant and Windows phones, respectively. The devices do not do inference locally; rather, they communicate with cloud services that perform forward evaluation through the network for them.

B. SID on Resource-Constrained Devices

[34] implements a very basic SID system in FPGA. The core recognition algorithm uses the the cosine distance between downsampled MFCC feature vectors. Accuracy of the system was not benchmarked. [35] replaces the cosine distance classifier with an SVM, again reporting no accuracy benchmarks. Both works implement their own DSP front-end as the crux of the papers.

More recently, [5] implements a complete GMM/UBM text-independent SID system on FPGA. They suffer a 5% accuracy reduction because of losses due to fixed-point integer approximations in the GMM mean and covariance matrices.

To the best of our knowledge, besides general purpose network evaluators such as EIE and Eyeriss, there has been no previous work that use perform neural-network based SID on resource-constrained devices [36], [37]. Both EIE and Eyeriss are much larger in area than what is required for SID, and lack a SID front-end.

V. TIMELINE

Previous and ongoing work:

1) Collect and pre-process YOHO, SRE, and RSR datasets. Become familiar with Kaldi (September 2016, Done)
2) Train GMM/UBM and i-vector models on SRE and RSR for comparison purposes (October 2016, Done)
3) Implement PLDA and neural network backend with i-vector model. Survey literature on end-to-end DNN approaches to SID and gauge feasibility. Research the costs and benefits of building i-vector vs. neural network SID for low-power. (November 2016, Done)
4) Implement first feedforward network. Test on YOHO datasets. Debug the extremely poor accuracies. Explore using AWS GPU machines for training (December 2016, Done)
5) Demonstrate a working SID demo on an ARM embedded device using neural network/i-vector fusion backend. Get setup on SLS GPU machines. (January 2017, Done)
6) Create network compression scripts that translate weights to 8-bit integers by using a quantization encoding scheme (February 2017, Done)
7) Continue attempts to make model smaller. Implement LSTM/recurrent networks as alternative to less-accurate feedforward networks. Debug poor accuracies. Review Bluespec, obtain Xilinx SoC. (March 2017, Done)

8) Various fixes to LSTM network. Implement attention mechanism on hunch. Make designs for matrix multiplier, ReLU in BSV (April 2017, Done)
9) Convert BSV to Vivado HLS in order load modules as IP into Vivado to program Xilinx SoC. Wire multipliers and activations together to obtain simple neural network architecture in hardware, on the Xilinx FPGA. (Early May 2017, Done)
10) Read manuals to figure out how to feed in audio data over UART from computer to the ARM chip on the Xilinx SoC, and then subsequently to the FPGA. Continue expanding simple network in Vivado, write test bench. Convert previously trained network weights to 8-bit integer using my scripts and benchmark. (May 2017, Ongoing)

Future work:

1) Demonstrate SID on Xilinx SoC, using previous weeks work (hardcoded network topology, front-end on computer) (early June 2017, Ongoing)
2) Benchmark small footprint attention network SID on more datasets, comparison i-vector vs. attention network, write-up for ICASSP publication. (June 2017, Ongoing)
3) Read and familiarize self with Michaels front-end DSP RTL source (VAD and feature extraction). Plan out how to integrate (July 2017)
4) Obtain documentation about RISC processor design from Chiraag. Familiarize self with how to execute programs on processor. Think about procedure for converting a Tensorflow network topology into program on RISC processor. (August 2017)
5) Implement and test front-end DSP RTL on Xilinx SoC (September 2017)
6) Implement and test general purpose matrix multiplier modules and activation modules (Figure 2), based on previous Vivado HLS work. Prepare for ICASSP submission. (October 2017)
7) Implement and test neural network to RISC instructions compiler (November 2017)
8) Piece together module, debug and test, and demo adjustable neural network based SID on Xilinx FPGA (December 2017)
9) Work with Chiraag on hardware accelerated encrypted-DNN evaluation (January 2018)
10) Port designs from encrypted DNN to SID designs to protect input and the speaker model (February, March 2018)
11) Explore other applications of the FPGA designs: running language ID and/or KWS on the same chip, using the same front-end, but loading different model (April, May 2018)

REFERENCES

[1] S. Price, "Radio broadcast hacks listeners' amazon echo devices," Mar 2016. [Online]. Available: http://www.upi.com/Odd_News/2016/03/16/Radio-broadcast-hacks-listeners-Amazon-Echo-devices/8711458115390/

[2] B. Feldman, "Npr segment on amazon echo accidentally activates robot army of speakers." [Online]. Available: http://nymag.com/selectall/2016/03/npr-segment-on-amazon-echo-messes-with-devices.html

[3] "Amazon echos activated by tv comment," Jan 2017. [Online]. Available: http://www.bbc.com/news/technology-38553643

[4] K. Prahallad, "Topic: Spectrogram, cepstrum and mel-frequency analysis." [Online]. Available: http://www.speech.cs.cmu.edu/15-492/slides/03_mfcc.pdf

[5] P. Ehkan, T. Allen, and S. F. Quigley, "Fpga implementation for gmm-based speaker identification," *International Journal of Reconfigurable Computing*, vol. 2011, p. 18, 2011.

[6] E. Chuangsuwanich and J. Glass, "Robust voice activity detector for real world applications using harmonicity and modulation frequency," 2011.

[7] M. Price, J. Glass, and A. P. Chandrakasan, "A scalable speech recognizer with deep-neural-network acoustic models and voice-activated power gating," *2017 IEEE International Solid-State Circuits Conference (ISSCC)*, 2017.

[8] ——, "A 6 mw, 5,000-word real-time speech recognizer using wfst models," *IEEE Journal of Solid-State Circuits*, vol. 50, no. 1, pp. 102–112, 2015.

[9] D. Reynolds, "Universal background models," *Encyclopedia of biometrics*, pp. 1547–1550, 2015.

[10] D. Povey, S. M. Chu, and B. Varadarajan, "Universal background model based speech recognition," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*. IEEE, 2008, pp. 4561–4564.

[11] W. M. Campbell, D. E. Sturim, D. A. Reynolds, and A. Solomonoff, "Svm based speaker verification using a gmm supervector kernel and nap variability compensation," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 1. IEEE, 2006, pp. I–I.

[12] S. Shum, "Low-dimensional speech representation based on factor analysis and its applications," 2011.

[13] P. Kenny, T. Stafylakis, P. Ouellet, M. J. Alam, and P. Dumouchel, "Plda for speaker verification with utterances of arbitrary duration," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 7649–7653.

[14] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1695–1699.

[15] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. EPFL-CONF-192584. IEEE Signal Processing Society, 2011.

[16] C. S. Greenberg, A. F. Martin, L. Brandschain, J. P. Campbell, C. Cieri, G. R. Doddington, and J. J. Godfrey, "Human assisted speaker recognition in nist sre10." in *Odyssey*, 2010, p. 32.

[17] A. Larcher, K. A. Lee, B. Ma, and H. Li, "Text-dependent speaker verification: Classifiers, databases and rsr2015," *Speech Communication*, vol. 60, pp. 56–77, 2014.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[19] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[20] J. Ma, R. P. Sheridan, A. Liaw, G. E. Dahl, and V. Svetnik, "Deep neural nets as a method for quantitative structure–activity relationships," *Journal of chemical information and modeling*, vol. 55, no. 2, pp. 263–274, 2015.

[21] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[22] F. Richardson, D. Reynolds, and N. Dehak, "Deep neural network approaches to speaker and language recognition," *IEEE Signal Processing Letters*, vol. 22, no. 10, pp. 1671–1675, 2015.

[23] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 4052–4056.

[24] 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.

[25] Y. Wang, J. Li, and Y. Gong, "Small-footprint high-performance deep neural network-based speech recognition using split-vq," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4984–4988.

[26] 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.

[27] Google. (2017, apr) How to quantize neural networks with tensorflow. [Online]. Available: https://www.tensorflow.org/performance/quantization

[28] V. Mnih, N. Heess, A. Graves *et al.*, "Recurrent models of visual attention," in *Advances in neural information processing systems*, 2014, pp. 2204–2212.

[29] S. Han, J. Pool, S. Narang, H. Mao, S. Tang, E. Elsen, B. Catanzaro, J. Tran, and W. J. Dally, "Dsd: Regularizing deep neural networks with dense-sparse-dense training flow," *arXiv preprint arXiv:1607.04381*, 2016.

[30] Z. Tang, L. Li, and D. Wang, "Multi-task recurrent model for speech and speaker recognition," *arXiv preprint arXiv:1603.09643*, 2016.

[31] J.-C. Wang, J.-F. Wang, and Y.-S. Weng, "Chip design of mfcc extraction for speech recognition," *INTEGRATION, the VLSI journal*, vol. 32, no. 1, pp. 111–131, 2002.

[32] S.-X. Zhang, Z. Chen, Y. Zhao, J. Li, and Y. Gong, "End-to-end attention based text-dependent speaker verification," *arXiv preprint arXiv:1701.00562*, 2017.

[33] D. Snyder, P. Ghahremani, D. Povey, D. Garcia-Romero, Y. Carmiel, and S. Khudanpur, "Deep neural network-based speaker embeddings for end-to-end speaker verification."

[34] G. Sarkar and G. Saha, "Real time implementation of speaker identification system with frame picking algorithm," *Procedia Computer Science*, vol. 2, pp. 173–180, 2010.

[35] R. Ramos-Lara, M. López-García, E. Cantó-Navarro, and L. Puente-Rodriguez, "Svm speaker verification system based on a low-cost fpga," in *Field Programmable Logic and Applications, 2009. FPL 2009. International Conference on*. IEEE, 2009, pp. 582–586.

[36] S. Han, X. Liu, H. Mao, J. Pu, A. Pedram, M. A. Horowitz, and W. J. Dally, "Eie: efficient inference engine on compressed deep neural network," in *Proceedings of the 43rd International Symposium on Computer Architecture*. IEEE Press, 2016, pp. 243–254.

[37] Y.-H. Chen, T. Krishna, J. S. Emer, and V. Sze, "Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks," *IEEE Journal of Solid-State Circuits*, 2016.