Deep Scattering Spectra with Deep Neural Networks for LVCSR Tasks

Tara N. Sainath[1], *Vijayaditya Peddinti*[2], *Brian Kingsbury*[1],
Petr Fousek[1], *Bhuvana Ramabhadran*[1], *David Nahamoo*[1]

[1]IBM T. J. Watson Research Center, Yorktown Heights, NY 10598, U.S.A
[2]Center for Language and Speech Processing, Johns Hopkins University, MD 21218, U.S.A

tsainath@us.ibm.com, vijay.p@jhu.edu, bedk@us.ibm.com,
petr_fousek@cz.ibm.com, {bhuvana, nahamoo}@us.ibm.com

Abstract

Log-mel filterbank features, which are commonly used features for CNNs, can remove higher-resolution information from the speech signal. A novel technique, known as Deep Scattering Spectrum (DSS), addresses this issue and looks to preserve this information. DSS features have shown promise on TIMIT, both for classification and recognition. In this paper, we extend the use of DSS features for LVCSR tasks. First, we explore the optimal multi-resolution time and frequency scattering operations for LVCSR tasks. Next, we explore techniques to reduce the dimension of the DSS features. We also incorporate speaker adaptation techniques into the DSS features. Results on a 50 and 430 hour English Broadcast News task show that the DSS features provide between a 4-7% relative improvement in WER over log-mel features, within a state-of-the-art CNN framework which incorporates speaker-adaptation and sequence training. Finally, we show that DSS features are similar to multi-resolution log-mel + MFCCs, and similar improvements can be obtained with this representation.

1. Introduction

A good feature representation for any pattern recognition task is one that preserves detail in the signal, while remaining stable and invariant to non-informative distortions. While conventional speech features, such as log-mel [1], PLP [2] and RASTA [3], are all designed to be deformation stable, they remove important higher-order information from the speech signal [4]. While better estimation techniques can be designed to preserve higher resolution detail [5], even these high resolution representations are processed using short term smoothing operators for deformation stability [6]. In short, designing an appropriate feature representation is challenging.

Deep scattering networks (DSN) [7] have recently been introduced to address some of the above challenges. DSNs take *a raw-signal* and generate a contractive representation, which preserves signal energy, while ensuring Lipschitz continuity to deformations ([7] and [8]). A scattering representation includes log-mel like features (first-order scatter) together with higher-order features that can preserve greater detail in the speech signal [4]. The representation generated by these networks, called Deep Scattering Spectrum (DSS), is locally translation invariant and stable to time varying deformations [4].

DSS features first showed promise in speech on TIMIT [9] for phonetic classification [4] and then recognition [10]. As there is only so much that can be learned from small-vocabulary tasks like TIMIT [11], in this paper, we extend the use of DSS features for large vocabulary continuous speech recognition (LVCSR). In our exploration of DSS features for LVCSR, we introduce many novelties in this paper. First, we explore the use of time and frequency scattering representations, which was only explored for classification [4] but not recognition [10]. Within this, we provide a deep analysis of how to properly do multi-resolution time and frequency scatter. Second, as the feature dimension for multi-resolution time and frequency scatter can be large (i.e., thousands), we introduce dimensionality reduction techniques to significantly reduce the feature dimension. Third, LVCSR systems incorporate speaker adaptation techniques [11], including VTLN [12], fMLLR [13] and i-vectors [14], which we also explore using in the DSS framework. While neural networks are ideal for high-dimensional time-frequency features, previous work has typically been done using shallow networks [15]. To our knowledge, this is one of the first papers exploring the benefit of time-frequency features using a state-of-the art deep CNN, encompassing speaker-adaptation and sequence-training [16].

The use of representations which filter in both time and frequency has certainly been explored before, including techniques such as PLP2 [17] and spectro-temporal modulation features [15, 18, 19]. The main difference between these representations and DSS features is the incorporation of 2nd-order scatter, and the operation to get frequency scatter. To be complete, we compare DSS features to other time-and-frequency feature representations, and analyze how important frequency and 2nd-order scatter are to the DSS framework.

Results with DSS features are first explored on a 50-hr English Broadcast News (BN) task [16]. We find that multi-resolution time+frequency DSS features achieves a 4% relative improvement over log-mel features. Next, we extend the use of DSS features to a larger 430-hr English BN task, where we observe gains of 7% relative, showing that this technique scales to larger tasks. Finally, we show that DSS features are similar to multi-resolution log-mel + MFCCs, and similar improvements can be obtained with this representation.

2. Deep Scattering Spectra with DNNs

2.1. DSS Features

In this section, we describe the DSS representation [4], including time scatter, frequency scatter, and multi-resolution scatter.

2.1.1. Time Scatter

As shown in [4], log-mel features can be approximated by convolving in time a signal x with a wavelet filterbank (ψ_{λ_1}), taking the modulus ($|.|$), and then applying a low-pass filter ($\phi(t)$). This feature representation can be written as $|x * \psi_{\lambda_1}| * \phi(t)$.

Typically, for a log-mel representation, the time of this averaging filter $\phi(t)$ is chosen to be ~ 25 ms and ψ_{λ_1} is a constant-Q filter-bank with Q=8. In this paper, following the terminology in [4], first-order scatter features are referred to as S_1.

While time averaging provides features which are locally invariant to small translations and distortions, it also leads to loss of higher-order information in the speech signal, such as attacks and bursts [4]. To recover this lost information another decomposition of the sub-band signals is performed using a second wavelet filter-bank(ψ_{λ_2}). This second decomposition captures the information in the sub-band signal, $|x * \psi_{\lambda_1}|$, left out by the averaging filter $\phi(t)$. The decomposed sub-band signals $|x*\psi_{\lambda_1}|*\psi_{\lambda_2}$, are once again passed through the low-pass filter $\phi(t)$ to extract stable features. The second order scatter is computed using a constant-Q filter-bank with $Q = 1$. Each of the decompositions $||x * \psi_{\lambda_1}| * \psi_{\lambda_2}| * \phi(t)$, has a limited number of non-zero coefficients, due to the band-limited nature of the signals $|x * \psi_{\lambda_1}|$. Typically, only first and second order scatter is used for speech [4, 10]. Again, following the terminology of [4], the second order scatter is referred to as S_2. To ensure that the higher order scatter just depends on the amplitude modulation component of the speech signal, the higher order scatter is normalized by the lower order scatter, i.e. $(\frac{S_2}{S_1})$.

The above description is known as time-scatter, as the wavelet convolution is applied to the time domain signal only. Next, we will describe frequency scatter, which was used in [4] for phonetic classification but not in [10] for recognition.

2.1.2. Frequency Scatter

Frequency scatter can be seen as a way of removing variability in the frequency signal, for example due to translations of formants created from different speaking styles. A very simple type of frequency averaging is to apply a discrete cosine transform (DCT) to a log-mel representation and perform cepstral truncation, which is common when generating MFCCs.

When applying frequency scatter in the DSS framework, the same time-scattering operation performed in time is now performed in the frequency domain on the S_1 and S_2 features. Specifically, frequency scattering features are created by iteratively applying wavelet transform and modulus operators, followed by a low-pass filter to the time-scatter features S_i, $|S_i * \psi^{fr}_{\lambda_1}| * \phi^{fr}(t)$. All frequency-scattering features are produced using wavelets with $Q = 1$. Similar to [4], we only compute first-order frequency scatter.

2.1.3. Multi-Resolution Scatter

The first-order time-scattering operating described in Section 2.1.1, is performed using a wavelet with $Q = 8$. To capture different spectral and temporal dynamics, wavelets with different Q factors can be used, an operation known as multi-resolution time scatter. Frequency and second-order scatter are calculated on each first-order time scatter S_1 generated with filterbank Q.

2.2. Neural Network Architecture

As discussed in [10], first-order time scatter features preserve locality in frequency, and thus they can be modeled by CNNs [20]. The second order time scatter, which is the decomposition of amplitude modulations in each sub-band of the first-order filter-bank ($|x * \psi_{\lambda_1}|$), preserves the locality of information, for a given sub-band λ_1. However, each of these sub-band decompositions has limited number of non-zero coefficients [10], and thus trying to model this with a CNN would be difficult as the resulting CNN filter size would be quite small in frequency. As a result, a DNN is better for 2nd order scatter. Following a similar analogy, DNNs are also more effective for frequency scatter.

To model DSS features, a joint CNN/DNN architecture is used [21]. The first order scatter for each Q is input into separate convolutional layers. All second-order time scatter and first-and-second order frequency scatter are fed as input into a fully connected layer. The output of this fully connected layer is then connected to the first fully connected layer of the CNN.

3. Exploration of DSS Features for LVCSR

3.1. Experimental Setup

We perform preliminary experiments to analyze DSS features on a 50-hr English BN task [16]. Results reported on 100 speakers from the EARS `dev04f` set. Similar to the architecture in [21], the CNN layer has 256 hidden units and the DNN layers have 1,024 hidden units, followed by 3,000 output targets. The joint CNN/DNN architecture has a similar number of hidden units, with the extra DNN layer also having 1,024 hidden units. We use rectified linear units (ReLU) for the non-linearity, which was shown in [10] to be better for DSS features. Unless otherwise noted, all DNNs are trained with cross-entropy, and results are reported in a hybrid DNN/HMM setup.

3.2. Analysis of Scattering Operations

3.2.1. Time Scatter

First, we analyze results using only time scatter, with a $Q = 8$ filterbank for the first-order scatter, which is similar to a log-mel representation. Table 1 shows adding second-order S_2 features gives small improvements, similar to what was observed for TIMIT [10]. One hypothesis is that since the scatter is computed over a small window for speech, i.e. 25-ms, second-order scatter is not beneficial for speech compared to audio tasks which use a much larger window [4]. Using a larger window for speech would smooth out important fine-level phonetic details.

Feature	WER
log-mel baseline	15.9
S_1, time	16.0
S_1+S_2, time	15.9

Table 1: Results for Time Scatter

3.2.2. Time and Frequency Scatter

Next, we analyze using time and frequency scatter, which can be a way of smoothing out the signal in time and removing frequency variations. This is similar in spirit to pooling in CNNs and VTLN-warping, which both aim to reduce frequency variations. In our experiments, first-order time-scatter uses a $Q = 8$ filterbank, while frequency-scatter uses a $Q = 1$ filterbank, as does second-order scatter. Table 2 shows that gains can be achieved with time-and-frequency scatter for recognition, even within a CNN framework, showing the complementarity of frequency scatter to CNN pooling.

3.2.3. Multi-Resolution Time and Frequency Scatter

Multi-resolution scatter applies a combination of filterbanks with different Q factors, in order to capture different resolutions of spectral and temporal dynamics in the signal. Multi-resolution scatter was shown to be helpful for both phonetic

Feature	WER
log-mel baseline	15.9
S_1, time	16.0
S_1+S_2, time	15.9
S_1+S_2, time+frequency	**15.5**

Table 2: Results for Time and Frequency Scatter

classification [4] and recognition [10]. However, one of the details these papers did not provide is a complete analysis of the recognition performance when combining different filterbanks, particularly when both time and frequency scatter are done.

Table 3 details the WER for different filterbanks which are complementary in their resolution for time scatter, namely $Q = 1, 4, 8, 13$. Note, the table only shows the different Q for time-scatter only, as frequency scatter is still done with a $Q = 1$ filterbank, though is affected by the time scatter resolution as it is performed on the output of the time scatter features.

Notice that using multiple filter-banks helps over using a single filterbank for time scatter. However, when the $Q = 1$ filterbank is used with another filterbank (i..e $Q = 8$ or $Q = 13$) for time and frequency scatter, additional improvements are not seen. One hypothesis is that because the features produced with a $Q = 1$ filterbank are small in dimension (i.e, 10), further smoothing with frequency scatter is not beneficial. The best results are obtained by using doing time and frequency scatter with higher resolutions, such as $Q = 4, 13$ or $Q = 8, 13$. This detailed analysis shows us that the $Q = 1, 8$ chosen for phonetic classification in [4] or $Q = 1, 4, 13$ chosen for phonetic recognition in [10] might not necessarily be the best choice for LVCSR. Overall, multi-resolution time and frequency scatter provides a WER of 15.1%, a 6% relative improvement in WER compared to log-mel features with a WER of 15.9%.

Feature	WER Time Scat.	WER Time+Freq. Scat.
log-mel baseline	15.9	15.9
S_1+S_2 (Q=1)	20.5	25.0
S_1+S_2 (Q=4)	16.2	16.3
S_1+S_2 (Q=8)	15.9	15.5
S_1+S_2 (Q=13)	16.1	15.7
S_1+S_2 (Q=1,8)	15.7	15.5
S_1+S_2 (Q=1,13)	15.5	15.5
S_1+S_2 (Q=4,13)	15.6	**15.1**
S_1+S_2 (Q=8,13)	15.3	**15.1**
S_1+S_2 (Q=1,4,13)	15.7	-

Table 3: Results for Multi-Q Time and Frequency Scatter

3.3. Dimensionality Reduction

Table 3 shows that the $Q = 4, 13$ filterbanks provide the best WER. However, this comes at the cost of a large increase in feature dimension. In this section, we highlight various strategies to reduce feature dimension for the S_1 frequency (S_1, f) and S_2 stream, fed into the DNN layer, as well as the S_1 time (S_1, t) stream, fed into the CNN layer.

3.3.1. DNN Dimensionality Reduction

Since a DNN does not require features to have correlations in time and frequency, unlike a CNN, standard dimensionality reduction techniques such as Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) [22], which have previously been explored for high-dimensional spectro-temporal features [19], can be applied.

The eigenvalues computed from applying a PCA to a combination of the S_1f+S_2 streams show that most of the variance of the data is explained by keeping 128 features. Table 4 shows that using first 128 features after applying a PCA, we can reduce parameters of the network from 26.5M to 14.1M, and take a small 0.1% increase in WER. Furthermore, increasing the features to 256 does not help. Finally, applying an LDA, which is often better at separating classes compared to PCA [22], we do not take any hit in WER, and can reduce parameters by 47%.

Feature	WER	Params
Baseline S_1, tf+S_2, tf (Q=4,13)	15.1	26.5M
S_1, tf+ **pca128**(S_1, f,S_2)	15.2	14.1M
S_1, tf+ **pca256**(S_1, f,S_2)	15.2	15.5M
S_1, tf+ **lda128**(S_1f,S_2)	**15.1**	**14.1M**

Table 4: Results with Dimensionality Reduction, DNN stream

3.3.2. CNN Dimensionality Reduction

Since features into the CNN layer must have time and frequency locality, we cannot apply PCA or LDA. Instead, we look at putting a linear bottleneck layer after the CNN streams, which is a reasonable place to remove locality as these features are further passed into the DNN layer. Table 5 shows the WER with different bottleneck sizes for the CNN layer. We can use a bottleneck size of 256, without taking any hit in WER. Combining the CNN and DNN dimensionality reduction schemes, we can reduce network size by approximately 60%, from 26.5M to 10.8M parameters, without any loss in accuracy.

Feature	WER	Params
Baseline S_1, tf+S_2, tf (Q=4,13)	15.1	26.5M
S_1, tf+ lda128(S_1f,S_2)	15.1	14.1M
S_1, tf,**bn=128**+ lda128(S_1f,S_2)	15.4	10.0M
S_1, tf,**bn=256**+ lda128(S_1f,S_2)	**15.1**	**10.8M**

Table 5: Results with Dimensionality Reduction, CNN stream

3.4. Scattering Features in a State-of-the-art System

Oftentimes we see that gains demonstrated on an speaker-independent (SI) system disappear once speaker adaptation and discriminative training are incorporated [11]. In this section, we demonstrate the value of DSS features after incorporating speaker adaptation and sequence training [16].

3.4.1. Speaker Adaptation

LVCSR systems typically apply VTLN to log-mel features [20]. First-order scatter is a similar operation to log-mel, with the exception that first-order scatter is computed using a Gabor rather than mel filterbank [4]. To apply VTLN to first-order scatter features, we compute a set of warped Gabor filterbanks, and estimate the optimal warp factor for each speaker via maximum likelihood, exactly as is done for mel filterbanks [12]. For a given speaker, the first-order scattering features are computed using the warped Gabor filters. Warping the Gabor filters changes the center-frequency and bandwidth of each filter, as well as the low-pass filter ψ, and thus the dimension of S_2 changes for each speaker since we preserve the non-zero S_2 coefficients. To have a constant S_2 dimension across utterances and speakers, we compute S_2 from the unwarped Gabor filters.

Table 6 shows the WER with and without VTLN for different time and frequency scattering operations. Notice that across

the board, VTLN provides improvements with all scattering operations. Furthermore, VTLN is complementary to frequency scatter, and offers gains on top of this.

Feature	WER no VTLN	WER with VTLN
log-mel	15.9	15.4
S_1+S_2, time+freq, Q=8	15.5	15.0
S_1+S_2, time+freq, Q=4,13	**15.1**	**14.7**

Table 6: Scattering Results with VTLN warping

In addition to VTLN, fMLLR [13] and i-vectors [14] are also commonly used speaker adapted features. As fMLLR and i-vector features do not obey locality in frequency, they can be incorporated as additional features in to the DNN stream, as in [23]. Table 7 shows that the DSS features show a 4% relative improvement even after incorporating fMLLR+i-vectors.

Feature	WER
log-mel +fMLLR+ivectors	13.9
S_1+S_2, time+freq, Q=4,13	**13.4**

Table 7: Scattering Results with VTLN, fMLLR+ivectors

3.4.2. Sequence Training

Since speech recognition is a sequence problem, WER of neural networks can be improved using a sequence-level training criterion [16] after cross-entropy training has finished. We apply sequence training to the networks trained with speaker-adapted log-mel and DSS features. Table 8 shows that even after sequence training, DSS features provide a 4% relative improvement in WER over log-mel features.

Feature	WER
log-mel	12.5
S_1+S_2, time+freq, Q=4,13	**12.0**

Table 8: Scattering Results after Sequence Training

3.5. Comparison With DNNs

As described in [4], DSS features are generated through a cascade of wavelet transforms and modulus nonlinearities, and thus have similar structure to a CNN, though it involves no learning. Given this property, we analyzed if a CNN model was needed with DSS features, or a DNN alone could be sufficient. Both experiments use DSS+fMLLR+ivector features. Table 9 shows that the joint CNN/DNN model is better than the DNN alone. This indicates that the convolutional structure learned by DSS features is complementary to the CNN, and both are needed for optimal recognition performance.

Feature	WER
joint CNN/DNN	**13.4**
DNN	14.2

Table 9: DNN vs. CNN/DNN

3.6. Results on 430-hr BN

We also explore scalability of the proposed techniques on 430 hours of English Broadcast News [16]. Results are reported on the full DARPA EARS dev04f set. The baseline deep CNN system is trained with speaker-adapted log-mel features, using the architecture described in Section 3.1. The speaker-adapted DSS features use a joint CNN/DNN architecture. Both networks have 5,999 output targets. Table 10 shows the results after sequence training. Even on a larger data set, scattering features provide a 7% relative improvement in WER, showing that this technique scales to larger tasks.

Feature	WER
log-mel	14.2
m1+m2, time+freq, mulitQ	13.2

Table 10: Results, 430 hrs BN

4. Can We Use Existing Methods?

While the application of DSS features to speech recognition is novel, it also raises the question: what other feature representations that are well studied in speech are similar to DSS features and could perhaps achieve similar results? For example, multi-resolution first-order time-scatter S_1 features are similar to generating log-mel features with different constant-Q filterbanks. The novelty of DSS features is the second-order S_2 features. In addition, frequency scatter can be seen as similar to applying a DCT, with the only difference being that frequency-scattering in the DSS framework applies a constant-Q filterbank, followed by a demodulation and low-pass filter operation.

To answer these questions, we compare the benefits of DSS features with log-mel and MFCCs, which represent operations similar to time first-order time and frequency scatter. Note, that a similar comparison could also have been done with other time+frequency feature representations, such as PLP2 [17] features, though log-mel + MFCC was chosen for simplicity. For these experiments, all DSS, log-mel and MFCC features are VTLN-warped. Log-mel features are fed into the CNN stream, while MFCCs are fed into the DNN stream.

Table 11 first compares S_1 time scatter to log-mel features, confirming that they have a similar WER. A small additional benefit is gained when adding in S_2 features. Next, notice that when incorporating time and frequency scatter with a $Q = 8$ filterbank, the WER is similar to log-mel + MFCC, and the gains from second-order S_2 features disappear. Finally, when using multi-resolution time and frequency scatter, log-mel+MFCC matches the performance of the DSS features. This shows us that while the gains coming from DSS features over log-mel seem to be consistent across tasks, similar results can be achieved with existing methods.

Feature	WER
log-mel, Q=8	15.4
S_1, time scatter, Q=8	15.4
S_1 +S_2 time scatter, Q=8	15.2
$S_1 + S_2$ time+freq scatter, Q=8	15.0
log-mel+mfcc, Q=8	15.0
$S_1 + S_2$ time+freq scatter, Q=4,13	14.7
log-mel + mfcc, Q=4,13	14.6

Table 11: Scattering vs. Log-Mel + MFCC

5. Conclusions

In this paper, we explored the use of DSS features for LVCSR tasks. We found that on both a 50 and 430-hr BN task, DSS features offered a 4-5% relative improvement compared to speaker-adapted log-mel features. However, we show that DSS features are similar to multi-resolution log-mel + MFCCs, and similar improvements can be obtained with this representation.

6. References

[1] S. Davis and P. Mermelstein, "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences ," *IEEE Transactions on Speech and Audio Processing*, vol. 28, no. 4, pp. 357 – 366, 1980.

[2] H. Hermansky, "Perceptual linear predictive (plp) analysis of speech," *The Journal of the Acoustical Society of America*, vol. 87, p. 1738, 1990.

[3] H. Hermansky and N. Morgan, "RASTA Processing of Speech," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 4, pp. 578 – 589, 1994.

[4] J. Andén and S. Mallat, "Deep scattering spectrum," *IEEE Transactions on Signal Processing*, 2014.

[5] M. Athineos and D. Ellis, "Autoregressive modeling of temporal envelopes," *IEEE Transactions on Signal Processing*, vol. 55, no. 11, pp. 5237–5245, 2007.

[6] S. Ganapthy, "Signal analysis using autoregressive models of amplitude modulation," Ph.D. dissertation, Johns Hopkins University, 7 2012.

[7] S. Mallat, "Deep learning by scattering," *Computing Resource Repository (CoRR)*, vol. abs/1306.5532, 2013.

[8] J. Bruna, S. Mallat *et al.*, "Invariant scattering convolution networks," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 8, pp. 1872–1886, 2013.

[9] L. Lamel, R. Kassel, and S. Seneff, "Speech Database Development: Design and Analysis of the Acoustic-Phonetic Corpus," in *Proc. of the DARPA Speech Recognition Workshop*, 1986.

[10] V. Peddinti, T. N. Sainath, S. Maymon, B. Ramabhadran, D. Nahamoo, and V. Goel, "Deep scattering spectrum with deep neural networks," in *to appear in Proc. of ICASSP*, 2014, pp. 361–364.

[11] T. N. Sainath, B. Ramabhadran, M. Picheny, D. Nahamoo, and D. Kanevsky, "Exemplar-Based Sparse Representation Features: From TIMIT to LVCSR," *IEEE Transactions on Speech and Audio Processing*, vol. 19, no. 8, pp. 2598–2613, 2011.

[12] L. Lee and R. C. Rose, "Speaker Normalization using Efficient Frequency Warping Procedures," in *Proc. ICASSP*, 1996.

[13] M. Gales, "Maximum likelihood linear transformations for HMM-based Speech Recognition," *Computer Speech and Language*, vol. 12, no. 2, pp. 75–98, 1998.

[14] G. Saon, H. Soltau, M. Picheny, and D. Nahamoo, "Speaker Adaptation of Neural Network Acoustic Models Using I-Vectors," in *Proc. ASRU*, 2013.

[15] S. Zhao and N. Morgan, "Multi-stream Spectro-Temporal Features for Robust Seech Recognition," in *Proc. Interspeech*, 2008.

[16] B. Kingsbury, "Lattice-Based Optimization of Sequence Classification Criteria for Neural-Network Acoustic Modeling," in *Proc. ICASSP*, 2009.

[17] M. Athineos, H. Hermansky, and D. Ellis, "PLP2: Autoregressive Modeling of Auditory-like 2-D Spectro-Temporal Patterns," in *ISCA Tutorial and Research Workshop on Statistical and Perceptual Audio Processing*, 2004.

[18] M. Kleinschmidt, "Localized Spectro-Temporal Features for Automatic Speech Recognition," in *Proc. Interspeech*, 2003.

[19] N. Mesgarani, M. Slaney, and S. A. Shamma, "Discrimination of Speech From Non-speech Based on Multiscale Spectro-Temporal Modulations," *TSALP*, vol. 14, no. 3, pp. 920–930, 2006.

[20] T. N. Sainath, A. Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep Convolutional Neural Networks for LVCSR," in *Proc. ICASSP*, 2013.

[21] T. N. Sainath, B. Kingsbury, A. Mohamed, G. Dahl, G. Saon, H. Soltau, T. Beran, A. Y. Aravkin, and B. Ramabhadran, "Improvements to Deep Convolutional Neural Networks for LVCSR," in *Proc. ASRU*, 2013.

[22] R. A. Fisher, "The use of Multiple Measurements in Taxonomic Problems," *Annals of Eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[23] H. Soltau, G. Saon, and T. N. Sainath, "Joint Training of Convolutional and Non-Convolutional Neural Networks," in *to appear in to Proc. ICASSP*, 2014.