Robust Language Identification Using Convolutional Neural Network Features

Sriram Ganapathy[1], *Kyu Han*[1], *Samuel Thomas*[1], *Mohamed Omar*[1],
Maarten Van Segbroeck[2], *Shrikanth S. Narayanan*[2]

[1]IBM T.J. Watson Research Center, Yorktown Heights, NY, USA.
[2]Signal Analysis and Interpretation Laboratory, University of Southern California, Los Angeles, USA.
{ganapath,kjhan,sthomas,mkomar}@us.ibm.com, {maarten,shri}@sipi.usc.edu

Abstract

The language identification (LID) task in the Robust Automatic Transcription of Speech (RATS) program is challenging due to the noisy nature of the audio data collected over highly degraded radio communication channels as well as the use of short duration speech segments for testing. In this paper, we report the recent advances made in the RATS LID task by using bottleneck features from a convolutional neural network (CNN). The CNN, which is trained with labelled data from one of target languages, generates bottleneck features which are used in a Gaussian mixture model (GMM)-ivector LID system. The CNN bottleneck features provide substantial complimentary information to the conventional acoustic features even on languages not seen in its training. Using these bottleneck features in conjunction with acoustic features, we obtain significant improvements (average relative improvements of 25% in terms of equal error rate (EER) compared to the corresponding acoustic system) for the LID task. Furthermore, these improvements are consistent for various choices of acoustic features as well as speech segment durations.

Index Terms: Convolutional Neural Networks, Bottleneck Features, Language Identification.

1. Introduction

The DARPA Robust Automatic Transcription of Speech (RATS) [1] program targets the development of speech systems operating on highly distorted speech recorded over "degraded" radio channels. The data used here consists of recordings obtained from retransmitting a clean signal over eight different radio channel types, where each channel introduces a unique degradation mode specific to the device and modulation characteristics [1]. For the language identification (LID) task, the performance is degraded due to the short segment duration of the speech recordings in addition to the significant amount of channel noise. In this paper, we discuss the techniques developed to improve the LID system performance over the previous submission [2].

Traditionally, phoneme recognition followed by language modeling (PRLM) was one of the popular methods for automatic LID task [3, 4]. This approach uses a multilingual phoneme recognizer to generate phoneme sequences which are converted to language model (n-gram) features for the LID classifier. The success of this approach is dependent on the performance of the phoneme decoder. For relatively clean data with good phoneme recognition accuracies, the PRLM method provides good performance comparable to acoustic systems [5]. However, the performance of phoneme decoders and speech recognition systems is significantly degraded for the highly noisy data in the RATS corpus [6]. In the recent past, the use of multi-layer-perceptron (MLP) based posterior features were attemped for LID [7]. The Tandem features have shown promising results [8]. Motivated by this effort, we explore the use of convolutional neural network (CNN) based features for LID.

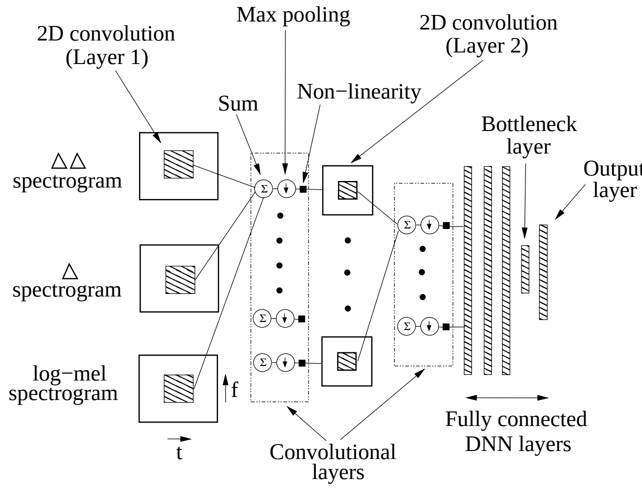

Figure 1: Architecture of a convolutional neural network containing one convolutional layer followed by deep neural network.

CNNs are variants of MLPs containing one or more convolutional layers and max pooling layers [9]. A convolutional layer consists of a set of weights which process a portion of the input signal. These weights are shared along the entire input space. The max pooling layer generates a lower resolution version of convolutional filter outputs by computing the maximum value of filter activations within a specified window. Recently, CNNs have shown promising results for various phoneme recognition and keyword spotting (KWS) tasks [10, 11].

In this paper, we develop a LID system using a CNN based phoneme recognizer trained on one of the target languages. The CNN is trained with *log-mel* spectrogram and contains a bottleneck (BN) layer before the output layer. For LID, the output of the BN layer from the trained CNN is used as feature representations for a Gaussian mixture model (GMM). The Gaussian mean supervector is converted to an ivector representa-

This research was funded by the Defense Advanced Research Projects Agency (DARPA) under Contract No. D11PC20192 under the RATS program.

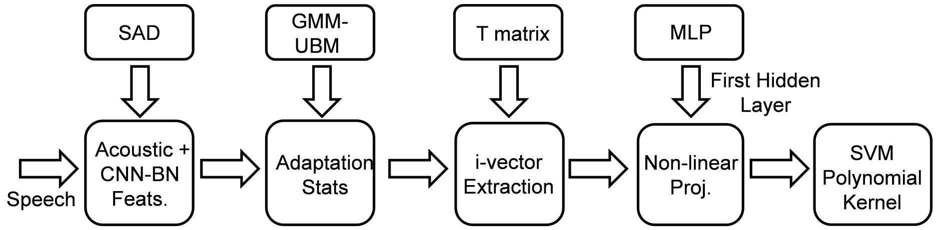

Figure 2: Block schematic of LID system using acoustic features appended with CNN bottleneck features.

tion [12] which is used to train language specific support vector machine (SVM) classifiers with higher order polynomial kernels [2]. We perform LID experiments on the RATS corpus for various speech segment durations. In these experiments, the additional information from the CNN BN layer provides significant improvements in the performance of the LID system (average relative improvements of 25% in EER compared to the corresponding acoustic feature based LID system).

The rest of the paper is organized as follows. In Sec. 2, we describe the CNN framework for phoneme recognition. Sec. 3 describes the application of CNN based features for LID. We analyze the characteristics of CNN-BN features for LID in Sec. 4. The LID experiments with CNN features are reported in Sec. 5. In Sec. 6, we conclude with a summary of the paper.

2. CNN Based Phoneme Recognition

The CNN models used in this paper are trained on noisy data provided under the RATS program for Arabic Levantine (ALV) and Farsi (FAS) KWS [11]. For each of these languages, about 300 hours of data, transmitted over 8 noisy channels, is available for acoustic modeling [1]. The CNNs are trained on 32 dimensional *log-mel* spectra augmented with Δ and $\Delta\Delta$s. The *log-mel* spectra are extracted by first applying *mel* scale integrators on power spectral estimates in short analysis windows (25 ms) of the signal followed by the *log* transform. Each frame of speech is also appended temporally with a context of 11 frames.

The block schematic of the CNN architechture is shown in Fig. 1. The CNNs use 2 convolutional layers with 512 hidden nodes. All the nodes in the first convolutional layer are processed with 9×9 filters that are two dimensionally (2D) convolved with the input representations. The output of the filters from the log, delta and double-delta streams are summed and are processed with the max-pooling operation, which downsamples the 2D representation along the spectral dimension. Here, three consecutive spectral values are replaced with their maximum value. The output of the max pooling is processed with sigmoidal non-linearity. The second convolutional layer has a similar set of 4×3 filters followed by max-pooling. The non-linear outputs from the second convolutional layer are then input to a fully connected deep neural network (DNN). We use three hidden layers with 2048 units, followed by a bottleneck layer with 25 activations before the final output layer (The number of BN activations was chosen to be small enough to facilitate the training of the LID system by concatenation with acoustic features). The networks are trained with the cross-entropy criterion.

The main advantage of CNNs for noisy and channel degraded speech comes from the use of local filters, weight sharing and max pooling. The use of local filters in CNNs which focus only on a few sub-bands, provides better robustness against channel distortions that are only present in parts of the spectrum. In such a case, the assumption is that the local filters which focus on relatively cleaner parts of the spectrum can still extract speech characteristics well enough to overcome any ambiguity arising from the noisy parts. The weight sharing and max pooling improve the robustness of the CNN to small frequency shifts. This is important because, for example, the formant locations for the same phoneme may appear on slightly different frequencies for different speakers or even for the same speaker due to linear frequency transpositions caused by the channel [1]. Furthermore, weight sharing of the filters helps in avoiding the issues with over-fitting and improves generalization due to the reduced number of trainable parameters.

3. LID system

The block schematic of the LID system [2] is shown in Fig. 2. The input signal is processed using Wiener filtering [13] and cepstral coefficients are derived which are referred to as acoustic features. We also use the CNN to generate bottleneck features of 25 dimensions which are concatenated with the acoustic features. A Gaussian mixture model-universal background model (GMM-UBM) with 1024 components is trained using the training and development portion of the LID data [1]. The zeroth and first order GMM statistics for each recording are obtained and these are used for training a factor analysis (FA) model [12]. We use 300 dimensional ivectors derived from the FA model to train a multi-layer perceptron (MLP) with one hidden layer. Once the MLP is trained, the nonlinear transformation from the ivectors to hidden layer outputs is alone retained and these hidden layer activations are used as inputs for SVM classification.

In our experiments, we use a SVM classifier with a higher order polynomial kernel for each target language of interest. For testing, the ivectors for the test utterance, processed by the MLP hidden layer, are used with each language dependent SVM to generate a score. A common threshold is applied to the score and the performance of the system is evaluated using equal error rate (EER) obtained from the detection error tradeoff (DET) curves.

4. Analyzing CNN-BN features for LID

In this section, we explore the usefulness of CNN-BN features for the LID system. We use a CNN trained on Arabic Levantine (ALV). The spectrographic representation of a portion of Arabic recording is shown in the left panel of Fig. 3. The posteriogram representation, which is the two dimensional plot of phoneme posteriors stacked along time, for this recording is shown in the bottom panel of Fig. 3. The similar plots for a Pashto (PUS) recording is shown in the right panel. Typically, a posteriogram

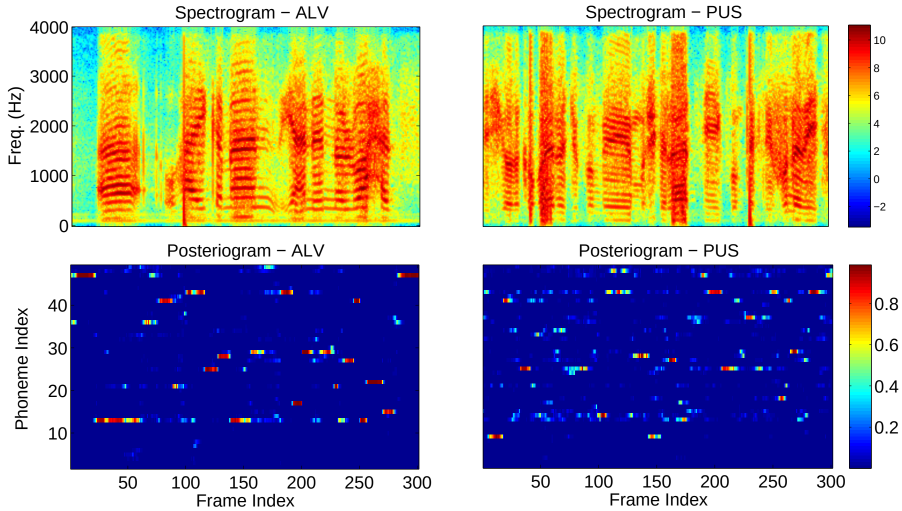

Figure 3: Comparison of spectrogram representation with posteriogram representation for a portion of Arabic and Pashto recording processed with Arabic CNN.

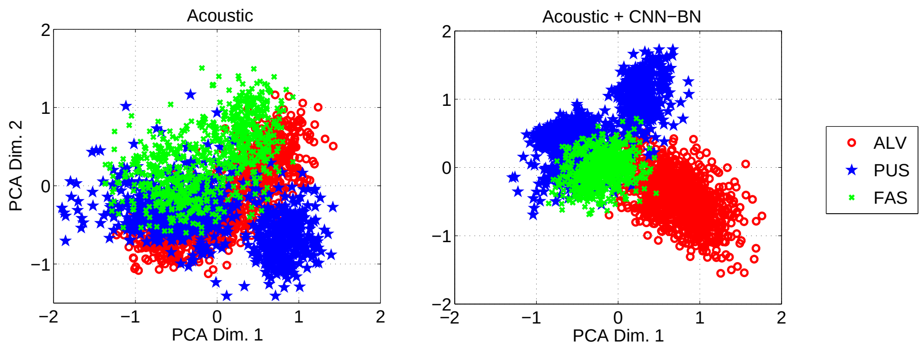

Figure 4: Scatter plot of first two dimensions of PCA projection for MLP hidden layer activations. The plot on the left uses acoustic features alone and the one on the right uses acoustic features with bottleneck features from ALV CNN.

with sharp activations indicates a good knowledge of the underlying phonetic content which could be useful for any application based on the posterior features. The posterior representation of ALV data is sharper and less noisy as the CNN is trained with ALV phonemes. Although the posteriogram for PUS data is noisy, there exists regions of the signal which generate sharp posteriors particularly for voiced regions. As seen in this figure, the information provided by the spectrogram and posteriogram streams are quite complimentary. The BN features used for LID experiments are a linear transformation of the posterior outputs except for a softmax operation.

For LID experiments, we concatenate the acoustic features with BN features and train the GMM based ivector model. In Fig. 4, we plot the first two principal components of the MLP hidden layer representation obtained using 30s recordings. The left panel shows a scatter plot for the LID system which uses acoustic features alone (in this case, power normalized cepstral coefficients [14]) and the right panel shows the same plot where the system was trained using a concatenation of acoustic and BN features. The scatter plot of the two significant PCA dimensions reveals that the fusion of acoustic and CNN-BN features improves the separation between the language classes considerably. This is desirable for improving the LID performance as the reduced overlap among language classes would result in a smaller number of false alarms for any given threshold.

5. Experiments

The development and test data for the LID experiments use the LDC releases of the Phase-I RATS LID development [1]. This consists of speech recordings from previous NIST-LRE clean recordings as well as other RATS clean recordings passed through eight (A-H) noisy communication channels. The training data contains about 270 hours of audio recorded over each radio channel. The five target languages are Arabic, Farsi, Dari, Pashto and Urdu. In addition to this, the database consists of several other imposter languages. In our experiments, the GMM-UBM is trained using $43,607$ recordings from the eight channels. The utterance level GMM statistics are used to train a factor analysis based ivector projection [12]. This model is trained with $33,672$ recordings of 120sec duration. The ivectors are used in a backend consisting of MLP hidden layer projection followed by a SVM training with the 12th order polynomial kernel (Sec. 3). We use 250k recordings of all durations for the MLP training and $82,398$ recordings for SVM training. The test data consists of two subsets - $52,789$ recordings from the eight noisy channels and four durations (120s,30s,10s and 3s) called the EVAL set as well as $9,899$ recordings from the DEV set.

In the initial set of experiments reported in Table 1, we use acoustic features based on power normalized cepstral coefficients (PNCC) [14]. The PNCC features are used to train the LID system (Sec. 3) with 250 dimensional (optimized for

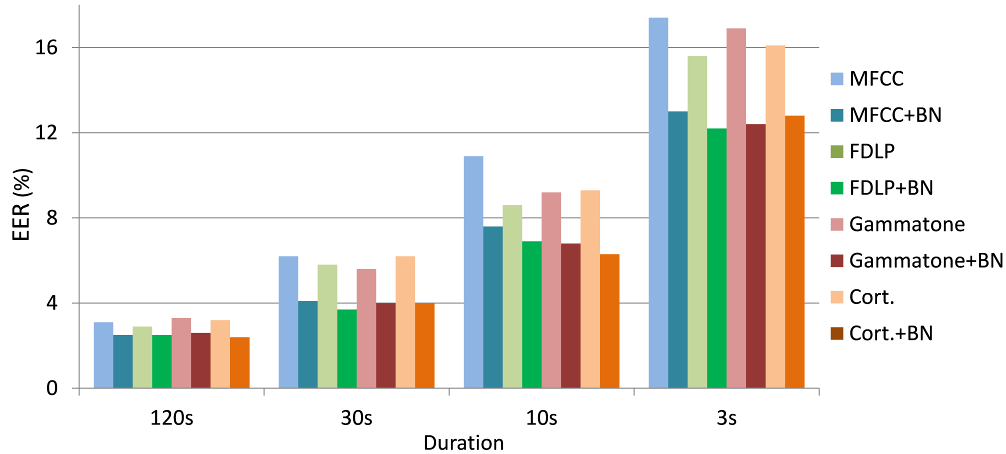

Figure 5: Performance of various of acoustic features with and without BN features for various speech segment durations of DEV set.

Table 1: Performance (EER %) of the LID system on the EVAL test set (DEV set in parentheses) for PNCC features with CNN features from ALV, FAS.

Feat.	120s	30s	10s	3s
PNCC	1.3 (3.1)	2.9 (5.2)	6.7 (8.5)	14.2 (15.6)
BN-ALV	1.3 (3.0)	2.3 (5.5)	5.9 (8.5)	15.3 (16.2)
BN-FAS	1.1 (2.6)	2.3 (4.8)	6.0 (8.3)	15.0 (15.0)
PNCC + BN-ALV	0.8 (2.7)	2.0 (4.3)	4.9 (6.6)	12.2 (11.7)
PNCC + BN-FAS	0.8 (2.8)	2.4 (3.6)	5.4 (6.6)	12.7 (11.1)

Table 2: Performance (EER %) of the LID systems on the DEV set using PNCC features with CNN features from ALV, FAS fused at various levels - feature, ivector and score.

Cond.	120s	30s	10s	3s
PNCC + ALV-BN Fusion				
Feat.	2.7	4.3	6.6	11.7
ivec	2.3	3.7	6.8	13.4
Score	2.6	3.8	6.4	12.5
PNCC + FAS-BN Fusion				
Feat.	2.8	3.6	6.6	11.1
ivec	2.3	4.0	6.1	12.8
Score	2.4	3.7	6.1	12.3

best performance [2]) ivectors followed by the SVM classifier. We experiment with the addition of CNN BN features generated from ALV-CNN as well as FAS-CNN to train the LID system with 300 dimensional ivectors. We also experiment with the use of BN features alone without any acoustic features with 200 dimensional ivectors. As seen in Table 1, the performance of the BN-FAS features are moderately better than the performance of the PNCC features. The use of BN features in addition with PNCC features provides significant improvement in performance for LID system for various test segment durations as well as the choice of test set. The BN features provide about 21% relative improvement in the EVAL set and about 25% in the DEV set.

The impact of BN features for various acoustic features is shown in Fig. 5. Here, we use a variety of feature processing techniques like mel frequency cepstral coefficients (MFCC) [15], frequency domain linear prediction (FDLP) [16], Gammatone [17] and cortical [18] features. In these experiments, the ALV-CNN based BN features are used and the results are reported on the DEV set for different speech segment durations. As seen in Fig. 5, the performance of all these features are improved by the use of BN features. The relative improvements are consistent even for short speech segment durations. These results illustrate that the bottleneck features based on CNN are both informative as well as complimentary to any choice of acoustic features for the LID task.

The results presented till now use the BN features in concatenation with the acoustic features. The final set of experiments, reported in Table 2, investigate the other methods of fusing the two streams, namely ivector fusion, where the ivectors from the two systems are used to jointly train the backend classifier as well as score fusion, where the scores from the two LID systems (acoustic and BN) are linearly combined with equal weighting. The feature fusion provides the best results although the ivector fusion provides good results for the 120s duration.

6. Summary

We have presented the application of convolutional neural network based phoneme recognition features for the LID task on the highly distorted radio channel data. The CNN BN features provide robust representations which are quite useful for the LID task by themselves. When the BN features are used in conjunction with acoustic features, significant improvements are obtained. These results are consistent for a variety of acoustic feature representations as well as the use of different target languages in CNN training. These experiments encourage us to pursue the use of multi-lingual CNNs in the future.

7. References

[1] K. Walker and S. Strassel, "The RATS Radio traffic collection system," in *Odyssey Speaker and Language Recognition Workshop*. ISCA, 2012.

[2] K. J. Han, S. Ganapathy, M. Li, M. Omar, and S. Narayanan, "TRAP Language identification system for RATS phase II evaluation," in *Interspeech*. ISCA, 2013.

[3] M. A. Zissman, "Comparison of four approaches to automatic language identification of telephone speech," *IEEE Transactions on Speech and Audio Processing*, vol. 4, no. 1, pp. 31, 1996.

[4] Jiri Navratil, "Spoken language recognition-a step toward multilinguality in speech processing," *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 6, pp. 678–685, 2001.

[5] N. Brummer, S. Cumani, O. Glembek, M. Karafiat, and P. Matejka, "Description and analysis of the Brno system for LRE2011," *Odyssey 2012-The Speaker and Language Recognition Workshop*, 2012.

[6] M. J. F. Gales and F. Flego, "Model-based approaches for degraded channel modelling in robust ASR.," in *Interspeech*. ISCA, 2012.

[7] Mhamed Faouzi BenZeghiba, Jean-Luc Gauvain, and Lori Lamel, "Phonotactic language recognition using mlp features.," in *Interspeech*, 2012.

[8] J. Ma, B. Zhang, S. Matsoukas, S. Mallidi, F. Li, and H. Hermansky, "Improvements in language identification on the RATS noisy speech corpus," 2013.

[9] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[10] O. Abdel-Hamid, A. Mohamed, H. Jiang, and G. Penn, "Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition," in *ICASSP*. IEEE, 2012, pp. 4277–4280.

[11] H. Soltau, H.K. Kuo, L. Mangu, G. Saon, and T. Beran, "Neural network acoustic models for the DARPA RATS program," in *Interspeech*, 2013.

[12] N. Dehak, P. A. Torres-Carrasquillo, D. Reynolds, and R. Dehak, "Language recognition via i-vectors and dimensionality reduction," in *Interspeech*, 2011.

[13] A. Adami and et al., "Qualcomm-ICSI-OGI features for ASR," in *Seventh International Conference on Spoken Language Processing*, 2002.

[14] C. Kim and R. Stern, "Power-normalized cepstral coefficients (PNCC) for robust speech recognition," *ICASSP*, 2012.

[15] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 28, no. 4, pp. 357–366, 1980.

[16] S. Thomas, S. Ganapathy, and H. Hermansky, "Recognition of reverberant speech using frequency domain linear prediction," *IEEE Signal Processing Letters*, vol. 15, pp. 681–684, 2008.

[17] R. Schluter, L. Bezrukov, H. Wagner, and H. Ney, "Gammatone features and feature combination for large vocabulary speech recognition," in *ICASSP*. IEEE, 2007.

[18] S. Nemala, K. Patil, and M. Elhilali, "A multistream feature framework based on bandpass modulation filtering for robust speech recognition," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 21, no. 2, pp. 416–426, 2012.