Speaker Adaptation of Convolutional Neural Network using Speaker Specific Subspace Vectors of SGMM

Murali Karthick Baskar
Brno University of Technology
7 PUBLICATIONS **3** CITATIONS
SEE PROFILE

Speaker Adaptation of Convolutional Neural Network using Speaker Specific Subspace Vectors of SGMM

Murali Karthick B, Prateek Kolhar, S Umesh

Dept. of Electrical Engineering

ee13s010@ee.iitm.ac.in, prateek.kolhar@gmail.com, umeshs@ee.iitm.ac.in

Abstract

The recent success of convolutional neural network (CNN) in speech recognition is due to its ability to capture translational variance in spectral features while performing discrimination. The CNN architecture requires correlated features as input and thus fMLLR transform which is estimated in de-correlated feature space fails to give significant improvement. In this paper, we propose two methods for extracting speaker adapted features in a correlated space using SGMMs. First, we estimate fMLLR transforms for correlated features by full covariance Gaussians using SGMM approach. Second, we augment speaker specific subspace vectors with acoustic features to provide speaker information in CNN models. Finally we propose a bottleneck - joint CNN/DNN framework to exploit the effects of both (fMLLR+ivectors) and (SGMM-fMLLR+speaker vectors) features. Experiments on TIMIT task show that our proposed features give 5.7 % relative improvement over the log-mel features. Furthermore experiments on switchboard task show that the bottleneck - joint CNN/DNN model achieves 12.2 % relative improvement over baseline joint CNN/DNN framework.

Index Terms: speech recognition, DNN, CNN, fMLLR, SGMM, subspace vectors

1. Introduction

Automatic speech recognition (ASR) systems have been through various changes in the way speech is modeled. The recent advent of deep neural networks (DNN) outperformed the conventional Gaussian mixture model - hidden Markov model (GMM-HMM) system [1]. Speaker Adaptation of features using feature space maximum likelihood regression (fMLLR) [2] and speaker identity vectors (i-vectors) [3] in DNN give significant improvements in recognition performance. The ability of DNN to model de-correlated features with i-vectors helps it to perform better speaker normalization. Further improvements in neural network were obtained by using convolutional neural network (CNN) [4] as it models the correlation along time and frequency in local segments of the input features.

The CNN architecture [5] can normalize speaker variance and stay immune to channel distortions, speaking styles, etc. The CNN consists of one or more pairs of convolution and max-pooling layers and the final layer as a softmax layer that produces the posterior probabilities. Each hidden activation is computed by multiplying localized inputs against the weights. These weights are shared across the entire input space, hence act like a filter. The filters help the CNN to model the correlation among the neighboring points in the input. The shared weights captures the translational variance to remove variability in the hidden units. Since CNNs try to model correlations they require input features such as log-mel filter bank (log-mel) which are locally correlated in time and frequency.

The improvement methods for the CNN model investigated in [6], show that vocal tract length normalization (VTLN) [7] over log-mel features improves the performance of CNN while fMLLR transform doesn't provide much improvement. One such method is multiscale CNN/DNN approach which showed better recognition performance. In this method, convolutional layers and fully connected layers are trained with VTLN warped log mel features with their outputs fed to a final DNN. Although, these methods showed improvement with VTLN warped features, fMLLR transformed log-mel features failed to improve due to the de-correlation effect of diagonal covariance Gaussian model used during estimation. Further study in [6], show that CNN model trained with features containing both fMLLR and i-vectors do not show as much improvement as seen in DNN. It also shows that the i-vectors do not satisfy the locality property and are more challenging to integrate with CNN.

To alleviate the above issues of speaker adaptation in CNN models, we propose three effective techniques: First, we modify the diagonal covariance transform to full-covariance fMLLR transform - using SGMM to preserve the correlation in the log-mel features. Second, in place of the i-vectors, we propose to use speaker specific subspace vectors of SGMM to train CNN model. Finally, we introduce a new framework in which a bottleneck layer placed after CNN layers and fully connected layers helps to extract features of relatively lesser dimension. These extracted features from both layers are then combined to train a final DNN model. The motivation behind using the bottleneck features is that it is more effective compared to directly providing the hidden layer outputs as features as suggested in [8].

To substantiate our hypothesis we conducted experiments on TIMIT and large vocabulary task using 30 hours switchboard corpus. The evaluation results showed that our proposed modifications in input features and model framework gave significant improvement to CNN. Our paper is structured as follows: In section 2, we have discussed about the prior works in modified CNN framework and speaker adaptation of CNN. Section 3 gives the overview of subspace Gaussian mixture model and how fMLLR transforms, speaker vectors are estimated. In section 4 we describe the architecture of bottleneck - joint CNN/DNN model. Section 5 gives the experimental setup and the parameters used in our system. In section 6 investigation is done over the proposed adaptation approaches on input features provided to CNN followed by its performance in our proposed framework. In section 7 the results are tabulated and discussed for LVSCR tasks and finally in 8 we conclude by showcasing the importance of our proposed modifications.

2. Related Work

The past work on applying speaker adaptation techniques to CNN has been proposed by [6]. The authors in [6] suggested a simple technique to apply fMLLR to CNN based input features. In this method, input features are re-correlated by multiplying the inverse of semi-tied covariance matrix with fMLLR transform matrix for training CNN model. However, these methods give relatively very less improvements showing that re-correlating the log mel features is not beneficial. In contrast to this approach, we have shown a simple way to compute fMLLR transforms for full-covariance matrix using SGMM.

The DNN and CNN need to be fed with separate type of features namely fMLLR and log-mel features for better m odeling. This problem has been discussed and handled in [9] by using a joint CNN/DNN scheme, where joint training of CNN and DNN is done with different feature sets. In this approach CNN and DNN are trained separately with VTLN+log-mel features and fMLLR features respectively. The output of these two models are then fed as input to a final DNN. Even though this method acheived better results, the number of parameters to be estimated are huge. This paper tries to overcome this problem by placing a bottleneck in both DNN and CNN before feeding it into final DNN.

3. Subspace Gaussian Mixture Model (SGMM)

SGMM [10] is an acoustic modeling technique whose mean and mixture weights are estimated by projecting a globally shared parameter towards a context-dependent state using low dimensional subspace vector. The important property of SGMM is that it is relatively compact and thus each specific state can be modeled with full-covariance Gaussian mixtures with lesser number of parameters. In SGMM, speaker adaptation is incorporated in the form of fMLLR transform approach and speaker specific subspace vectors.

3.1. SGMM-fMLLR Transforms

In [10] it is shown that SGMM is an effective model to estimate full-covariance fMLLR transforms which is an extension of the work done in [11]. A transform matrix $\mathbf{W}$ of dimension $d \times d+1$ is estimated, together with diagonal matrix $\mathbf{D}$ corresponding to eigen values in LDA computation. The fMLLR transformed feature vector $\widehat{\mathbf{x}}$ is computed as: $\widehat{\mathbf{x}} = \mathbf{W}\mathbf{x}$, where $\mathbf{W}^{(s)} = \left[\mathbf{A}^{(s)}; \mathbf{b}^{(s)}\right]$, $\mathbf{A}$ is square transform matrix and $\mathbf{b}$ represents bias.

3.2. Speaker Specific Subspace Vectors

The speaker vectors of SGMM [10] are estimated for each speaker separately after fixing the model parameters. These vectors are similar to the low-dimensional vector in joint-factor analysis but are well behaved and can be easily estimated with lesser number of parameters. The speaker vectors perform speaker adaptation in SGMM by moving the mean parameter closer to speaker space. The speaker subspace vectors $\hat{\mathbf{v}}_s = \mathbf{H}^{(s)^{-1}}\mathbf{y}^{(s)}$ are estimated using $\mathbf{H}^{(s)}$, $\mathbf{y}^{(s)}\gamma_i\,(s)$ and $\mathbf{H}^{(spk)}$.

$$\mathbf{H}^{(s)} = \sum_i \gamma_i\,(s)\,\mathbf{H}_i^{(spk)},\ \mathbf{H}^{(spk)} = \mathbf{N}_i^T \Sigma_i^{-1} \mathbf{N}_i$$

Here $\gamma_i\,(s)$ denotes the posterior probability of speaker s, $\mathbf{N}_i$ denotes speaker projection subspace matrix, $\mathbf{y}^{(s)}$ is an accumulation paramter and Σ_i represents full-covariance matrix for mixture component i. The resulting low-dimensional vector contains complete speaker information, since the estimation process includes a subspace matrix that projects down all the speaker characteristics to the low dimensional form.

4. Bottleneck - Joint CNN/DNN Framework

Bottleneck - Joint CNN/DNN approach is an extension of the existing joint CNN/DNN framework implemented by [9]. In this framework the CNN and DNN models are trained separately with their appropriate features and then their outputs are concatenated to train a final DNN model. This method exploits the benefits of CNN specific features and DNN specific features together. We propose to improve on this framework to reduce the input dimension by adding bottleneck layers to the CNN and DNN as in figure 1. This modification reduces the number of parameters and also provides significant improvement in performance.

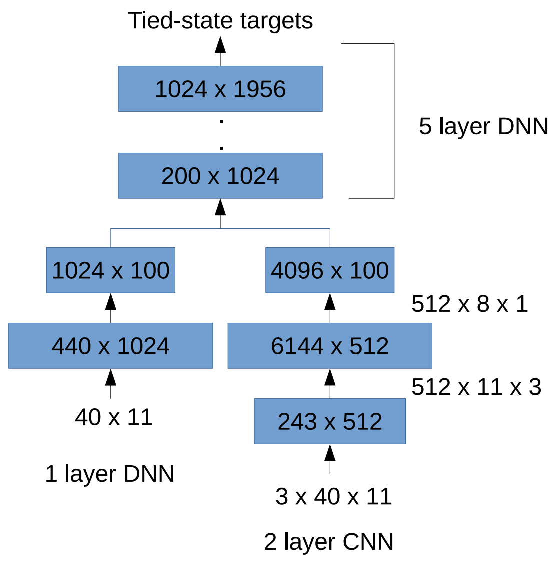

Figure 1: Block Diagram of proposed Bottleneck - Joint CNN/DNN Architecture

5. Experimental Setup

All our experiments are performed using Kaldi [12] speech recognition toolkit. We conducted preliminary experiments using TIMIT [13] dataset for evaluating our proposed features and bottleneck - joint CNN/DNN framework. The phone recognition experiments on TIMIT are done with train data containing 3696 utterances, test set of 192 utterances by using a bi-phone language model extracted from the training set for decoding. Experiments on large vocabulary (LVCSR) task are done with switchboard corpus (SWBD) [14] having 30 hours speech as a train set and 4k utterances for development as test set to show the consistency and performance of our approach. A tri-gram language model built using ˜700-hours of Fishers transcripts and Switchboard-1 (SW-1) transcripts are used for decoding.

A 13 dimensional Mel-frequency cepstral coefficients (MFCC) [15] along with delta and acceleration coefficients are initially extracted from the speech data. Linear discriminant analysis (LDA) [16] followed by maximum likelihood linear transform (MLLT) [17] and fMLLR transformation [2] are then applied over these MFCC features to build a GMM-HMM system. 40-dimensional log-mel filter bank features with their delta and acceleration coefficients are extracted to build an SGMM. A full covariance universal background model (UBM) [18] is then transformed to different context dependent tied-states of SGMM using subspace vectors.

Table 1: Number of tied-states, Gaussian mixtures, network layers and nodes, substates of SGMM used for TIMIT and 30 hours of SWBD corpus

Datasets	GMM-HMM		SGMM		DNN	
	ts	Gs	ts/ss	Gs	hl	ns
TIMIT	1956	15021	5827/9649	400	5	1024
SWBD	4167	90108	3697/400006	700	6	1024

hl-hidden layers, *ns*-nodes, *ts*-tied-states, *Gs*-Gaussian mixtures and *ss*-Substates

5.0.1. DNN and CNN Training

KALDI [12] recipe for TIMIT and SWBD is used to perform the DNN and CNN experiments. The fMLLR features spliced over a context of 11 frames are fed as input to build a generatively pre-trained DNN system. The number of output targets in DNN is kept same as the number of context-dependent tied-states of GMM-HMM system. We use 5 hidden layers each containing 1024 units with sigmoid activation function and output layer with softmax function. The parameter details of GMM-HMM, SGMM and DNN are shown in table 1.

In [4], it is shown that CNN requires correlated features such as log-mel. Thus 40-dimensional log-mel filter bank features with their delta and acceleration coefficients are used in all our CNN experiments. The log-mel features are spliced over a temporal context width of 11 frames. We use two CNN layers in our experiments, each with a separate configuration [6]. An overlapping context window of 9×9 is applied over the input features in the first layer. 3 outputs from the first convolutional layer are then pooled to the max-pooling layer. In the second CNN layer an overlapping window of 4×3 is applied to obtain 8×1 windowed input for convolutional layer. These two CNN layers are trained using frame level cross-entropy training. The method to incorporate speaker vectors into the CNN model is described in section 6.2. A common CNN configuration shown in table 2 is used for both TIMIT and SWBD experiments.

Table 2: CNN configuration for both TIMIT and SWBD experiments. Here Win denotes Windows and Pool denotes pooling.

CNN Layer	Feature Space				Temporal Space			
	Size		Shift		Size		Shift	
	Win	Pool	Win	Pool	Win	Pool	Win	Pool
1	9	3	1	3	9	1	1	1
2	4	1	1	1	3	1	1	1

The number of nodes in each convolutional layer is 512. The input dimensions of first layer is 3 times 9×9 window which is 243 dimensions (243×512). The second layer contains $512 \times 4 \times 3$ =6144 (6144×512) dimensions.

6. Analysis

In this section we analyze the effect of our three proposed methods in the CNN framework. First, we apply SGMM based fMLLR transform to apply speaker adaptation while preserving the correlation in the log-mel features. Secondly, we augment the speaker specific subspace vectors of SGMM with the speaker normalized input features to provide additional speaker information. Further we compare the performance of our bottleneck - joint CNN/DNN framework with the existing joint - CNN/DNN framework. The experiments to analyze the effect of the proposed speaker adapted features is done using TIMIT dataset. The model contains two CNN layers and an output layer in these CNN experiments.

6.1. Effect of SGMM-fMLLR Features

The SGMM-fMLLR transform matrix (40×41) is estimated from SGMM and then applied over the log-mel features to obtain SGMM-fMLLR (40×1) features. The intuition behind applying fMLLR transforms estimated using SGMM over log-mel features are: 1) it can generate full-covariance transform matrix with lesser number of parameters as mentioned in [11], 2) it helps to transform the log-mel features by preserving the correlations and locality. These features of SGMM-fMLLR transform provide better feature representation that help CNN to perform better speaker normalization.

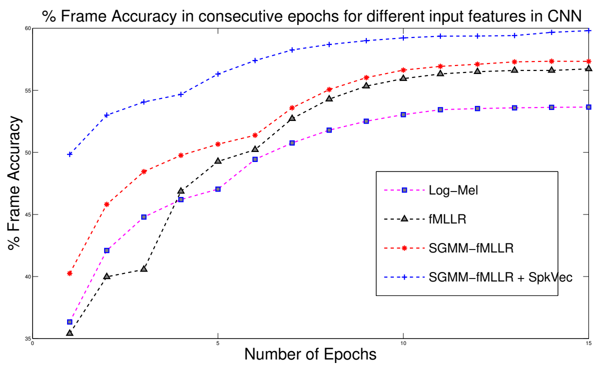

Figure 2: % Frame Accuracy on cross validation data for various epochs for TIMIT dataset

To validate our hypothesis, we can observe from figure 2 the frame accuracy improves for our proposed features. The frame accuracy denotes the classification power of CNN and varies for different features. From the figure, it is noted that the accuracy for fMLLR and SGMM-fMLLR features has less variation. Although, the fMLLR features has gained better accuracy over log-mel features the recognition rate of fMLLR features is inferior compared to log-mel features. The similar pattern is reflected in table 3 where conventional fMLLR features degrade the CNN performance while SGMM-fMLLR features gain relative improvements. So further experiments on CNN with speaker vectors are done only using SGMM-fMLLR features and log-mel features.

Table 3: Performance Comparison of SGMM-fMLLR features with basic fMLLR features. Here R.I represents relative improvement

Feature Type	% PER	% R.I.
Log-mel	21.2	-
fMLLR	22.3	- 4.9
SGMM-fMLLR	21.0	0.9

6.2. Effect of Speaker Specific Subspace Vectors

The SGMM speaker vectors are a well structured representation of the speaker's phonetic information in lower dimensions. Since these vectors act as mean adapting bias towards target speakers in SGMM, we hypothesize that they also can help CNN perform speaker adaptation more effectively. In this method we extract subspace vector of $S = 40$ subspace dimensions for each speaker. In CNN, the feature augmentation is done as shown in [4]. A 40 dimensional speaker vector is concatenated in the direction of frequency when a 9×9 overlapping context window is applied over the input feature matrix resulting in a new window size of 9×49 similar to the method used for i-vectors in [6]. The same speaker vectors are augmented for all context window shifts and the weights are shared across the shifts.

Table 4: Performance comparison of i-vectors and speaker vectors as augmented features in CNN

Feature Type	% PER	% R.I.
Log-mel + i-vectors	20.9	1.4
SGMM-fMLLR + i-vectors	20.7	2.3
Log-mel + spk. vectors	20.4	2.8
SGMM-fMLLR + spk. vectors	19.8	5.7

Here we compare the performance of speaker vectors and ivectors as augmented features with log-mel and SGMM-fMLLR features. From the table 4 we see that the speaker vectors along with log-mel features performed better than i-vectors. The table also shows that using speaker vectors along with SGMM-fMLLR features further improves the performance. The speaker vectors are more efficient in training CNN than i-vectors. The important reason behind this is that, the ivectors is of higher dimensions and hence needs huge number of parameters to be estimated. In case of speaker vectors, the low dimensionality helps in helps in robust estimation of weights in CNN with lesser paramters for performing better speaker adaptation.

6.3. Effect of Bottleneck-Joint CNN/DNN Framework

In our experiments we have used the joint CNN/DNN [9] as a baseline model. Here we have used the (SGMM-fMLLR + speaker vectors) as CNN specific features and (fMLLR + i-vectors) as DNN specific features. Table 5 compares the recognition performance of our proposed joint CNN/DNN model with bottleneck framework with the existing joint CNN/DNN technique for TIMIT dataset. The bottleneck layer helps to perform better classification over feature frames and thus the model show improved performance compared to joint CNN/DNN without having bottleneck layer. The table 5 also shows that the number of parameters involved while training our proposed model reduces by approximately 14 % when compared to the model without bottleneck layer. Also from the table we can observe that bottleneck - joint CNN/DNN gave 2.5 % absolute improvement over the baseline model.

7. Results on LVCSR Task

In this section we compare the results of joint CNN/DNN model with bottleneck-joint CNN/DNN using our proposed features using 30 hours of switchboard task. The results show that our proposed model relatively works well with SGMM-fMLLR+speaker vector features and give an average 10.7 % relative improvement over baseline joint CNN/DNN model. The results in 6 shows that as the data size increases the performance of our proposed framework gave 12% relative improvement while for TIMIT task it gave 8.3 % relative improvement over the baseline system. In case of our proposed speaker adaptation methods, the relative improvement is more for SGMM-fMLLR with switchboard while for speaker vectors the same range of improvement is obtained as in TIMIT dataset.

Table 5: Performance comparison of our proposed Bottleneck-Joint CNN/DNN with Joint CNN/DNN on TIMIT dataset. Here Bnk denotes bottleneck and # Prms represents number of parameters

Feature Type		Joint CNN/DNN		% R.I.
DNN	CNN	Without Bnk % WER (# Prms)	With Bnk % WER (# Prms)	
fMLLR + i-vectors	Log-mel	20.3 (13.4)	18.9 (11.0)	6.9
	+SGMM-fMLLR	19.5 (21.7)	18.1 (18.5)	7.2
	+ Spk. Vectors	17.9 (29.4)	16.4 (25.6)	8.3

Table 6: Performance comparison of our proposed Bottleneck-Joint CNN/DNN with Joint CNN/DNN for SWBD corpus

Feature Type		Joint CNN/DNN		% Rel. Imp
DNN	CNN	Without Bnk % WER	With Bnk % WER	
fMLLR + i-vectors	Log-mel	24.9	22.8	8.4
	+SGMM-fMLLR	23.4	20.7	11.5
	+ Spk. Vectors	22.1	19.4	12.2

8. Conclusion

In this paper we have proposed two simple yet effective techniques for incorporating speaker adaptive features in CNN domain. In the CNN architecture fMLLR transform approach using SGMM and the speaker specific subspace vectors are observed as better alternatives to conventional fMLLR and i-vectors. The addition of bottleneck layer in joint CNN/DNN model gives an improvement in all our experiments and also helps in reducing the number of parameters. The experimental results show that our proposed modification to the features give 5.7 % and 13.1 % relative improvement over the log-mel filterbank features for TIMIT and switchboard corpus respectively. A 12.2 % relative improvement is obtained for joint CNN/DNN with bottleneck over basic joint CNN/DNN model in switchboard task. The above improvements in the CNN framework are observed as a support for our hypothesis that SGMM-fMLLR and speaker vectors play a important role as input features for CNN.

9. References

[1] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks.,"

in *Interspeech*, pp. 437–440, 2011.

[2] M. J. Gales, “Maximum likelihood linear transformations for hmm-based speech recognition,” *Computer speech & language*, vol. 12, no. 2, pp. 75–98, 1998.

[3] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, “Front-end factor analysis for speaker verification,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 4, pp. 788–798, 2011.

[4] O. Abdel-Hamid, A.-r. Mohamed, H. Jiang, and G. Penn, “Applying convolutional neural networks concepts to hybrid nn-hmm model for speech recognition,” in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pp. 4277–4280, IEEE, 2012.

[5] Y. LeCun and Y. Bengio, “Convolutional networks for images, speech, and time series,” *The handbook of brain theory and neural networks*, vol. 3361, p. 310, 1995.

[6] T. N. Sainath, B. Kingsbury, G. Saon, H. Soltau, A.-r. Mohamed, G. Dahl, and B. Ramabhadran, “Deep convolutional neural networks for large-scale speech tasks,” *Neural Networks*, 2014.

[7] D. Kim, S. Umesh, M. Gales, T. Hain, and P. Woodland, “Using vtln for broadcast news transcription,” in *Proceedings of ICSLP*, 2004.

[8] F. Grezl, M. Karafiát, S. Kontár, and J. Cernockỳ, “Probabilistic and bottle-neck features for lvcsr of meetings.,” in *Proc. ICASSP, 2007*, pp. 757–760, IEEE, 2007.

[9] H. Soltau, G. Saon, and T. N. Sainath, “Joint training of convolutional and non-convolutional neural networks,” *to Proc. ICASSP*, 2014.

[10] D. Povey, L. Burget, M. Agarwal, P. Akyazi, F. Kai, A. Ghoshal, O. Glembek, N. Goel, M. Karafiát, A. Rastrow, R. C. Rose, P. Schwarz, and S. Thomas, “The subspace Gaussian mixture model-a structured model for speech recognition,” *Computer Speech & Language*, vol. 25, pp. 404–439, Apr. 2011.

[11] D. Povey and G. Saon, “Feature and model space speaker adaptation with full covariance gaussians.,” in *INTERSPEECH*, 2006.

[12] 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 *Proc. ASRU, 2011*, pp. 1–4, IEEE, 2011.

[13] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett, “Darpa timit acoustic-phonetic continous speech corpus cd-rom. nist speech disc 1-1.1,” *NASA STI/Recon Technical Report N*, vol. 93, p. 27403, 1993.

[14] J. J. Godfrey, E. C. Holliman, and J. McDaniel, “Switchboard: Telephone speech corpus for research and development,” in *Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on*, vol. 1, pp. 517–520, IEEE, 1992.

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

[16] N. Kumar and A. G. Andreou, “Heteroscedastic discriminant analysis and reduced rank hmms for improved speech recognition,” *Speech communication*, vol. 26, no. 4, pp. 283–297, 1998.

[17] G. Saon, M. Padmanabhan, R. Gopinath, and S. Chen, “Maximum likelihood discriminant feature spaces,” in *Acoustics, Speech, and Signal Processing, 2000. ICASSP’00. Proceedings. 2000 IEEE International Conference on*, vol. 2, pp. II1129–II1132, IEEE, 2000.

[18] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, “Speaker verification using adapted gaussian mixture models,” *Digital signal processing*, vol. 10, no. 1, pp. 19–41, 2000.