CONVOLUTIONAL NEURAL NETWORK PRE-TRAINED WITH PROJECTION MATRICES ON LINEAR DISCRIMINANT ANALYSIS

Takashi Fukuda, Osamu Ichikawa, and Ryuki Tachibana

IBM Watson Multimodal, IBM Japan Ltd.
19-21, Nihonbashi Hakozaki-cho, Chuo-ku, Tokyo 103-8510, JAPAN
E-mail:{fukuda1, ichikaw, ryuki}@jp.ibm.com

ABSTRACT

Recently, the hybrid architecture of a neural network (NN) and a hidden Markov model (HMM) has shown significant improvement on automatic speech recognition (ASR) over the conventional Gaussian mixture model (GMM)-based system. The convolutional neural network (CNN), a successful NN-based system, can represent local spectral variations spanning the time-frequency space. Meanwhile, spectro-temporal features have been widely studied to make ASR more robust. Typically, the spectro-temporal features are extracted from acoustic spectral patterns using a 2D filtering process. Convolutional layers in CNN that have various local windows can also be regarded as an efficient feature extractor to capture 2D spectral variations. In a standard procedure, the local windows in CNN are initialized randomly before the pre-training and are iteratively updated with a back propagation algorithm in the pre-training and fine-tuning steps. In this paper, we explore using projection matrices composed of eigenvectors estimated by linear discriminant analysis (LDA) objective function as initial weights for the first convolutional layer in CNN. From analysis of the local windows trained by the proposed method, we can see the eigenvectors of LDA has desirable properties as initial weights of CNN. The proposed method yielded a 8.1% relative improvement compared to CNN with local weights initialized randomly.

Index Terms— CNN, LDA, eigenvector, local window

1. INTRODUCTION

Deep neural networks (DNNs) have recently achieved tremendous success for automatic speech recognition (ASR) tasks. They are usually combined with hidden Markov model (HMM) and compute the output probabilities instead of the conventional Gaussian mixture model (GMM). A common CNN topology was proposed by LeCun *et al.* [1, 2] and has been widely used in image recognition and computer vision fields [3]. Sainath *et al.* applied CNN to large vocabulary continuous speech recognition (LVCSR) and searched for the appropriate architecture to make CNN more effective compared to DNNs [4, 5]. CNN used for ASR is typically comprised of one or more convolutional layers often with a subsampling step and then followed by several fully connected layers as in a standard multilayer neural network. It is known that CNN reduces spectral variations caused by speaker characteristics, speaking styles, and acoustic environments compared to fully connected DNN alone, which has been extensively used for acoustic models. The architecture of CNN is designed to take advantage of the 2D structure of input features spanning time-frequency plane. This is achieved with local connections (local windows) and tied weights followed by some form of pooling, that generates translation-invariant features. In general, CNN works with far fewer parameters than DNN does. Since spectral representations of speech have strong correlations, modeling local correlations with CNN has been shown to be beneficial [4].

Spectro-temporal features are also a well researched topic, especially for noise-robust ASR. Most of the techniques are based on modulation spectra and Gabor filtering for time-frequency spectral patterns [6, 7, 8, 9, 10]. They were created to represent certain stimuli to which the neurons of the mammalian auditory cortex are sensitive. These stimuli consist of both spectral and temporal modulation frequencies [11]. One area of research focus for spectro-temporal features is how long-term temporal information is exploited for ASR [12, 13, 14, 15]. Although this trend stems from recent findings on the human auditory system, how to effectively integrate short-term temporal variation with ASR is still an important research topic. In addition to these techniques inspired by the auditory system, linear discriminant analysis (LDA) and heteroscedastic discriminant analysis (HDA), both of which are based on machine learning theory, are also used to represent temporal information of spectra [16]. LDA-based sprectro-temporal representation has been shown to be beneficial in the literature. In a typical example of using LDA for a feature extraction, a supervector is made from several consecutive frames and transformed into a new feature space that is more suitable for ASR by multiplying the supervector by the projection matrix estimated from the training data.

Although both CNN and spectro-temporal features play an important role in making ASR robust, in terms of feature extraction, these techniques do not sufficiently generalize acoustic variations spanning the time-frequency space. This paper proposes a method to apply projection matrices estimated with the LDA objective function as initial weights for a first convolutional layer of CNN to effectively utilize the classification capability that LDA originally has. We expect the synergetic effect between CNN and LDA because CNN is the algorithm directly focusing on errors of phone classification while LDA is based on the distributions of phones. Conventionally, the first n eigenvectors obtained by LDA that have large eigenvalues are considered as filters that extract the important variational components of spectra. We present through experiments that replacing initial weights with eigenvectors from LDA helps construct better local filters and improves the overall accuracy on ASR. This paper also compares the local weights estimated by CNN and the eigenvectors of LDA, and shows that the eigenvectors of LDA can effectively function as initial weights of CNN.

2. PROPOSED METHOD

2.1. CNN

This section briefly summarizes a standard CNN architecture focusing on the key points of comparison related to our proposed method.

978-1-4799-9988-0/16/$31.00 ©2016 IEEE

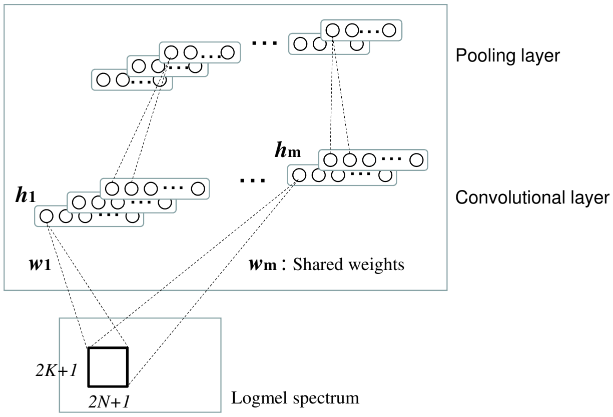


Fig. 1. CNN architecture.

Most of the notations in this section are based on the paper written by Sainath *et al* [4].

A typical CNN architecture for ASR is shown in Figure 1. Unlike DNN, each hidden activation $\boldsymbol{h}_m$ in convolutional layers is computed by multiplying the small local input $\boldsymbol{V}$ by weights $\boldsymbol{W} = [\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_m, \cdots, \boldsymbol{w}_M]$, adding a bias b_m, and applying a nonlinear function in that layer, where $\boldsymbol{w}_m$ is a connection weight $w_m(i, j)$ consisting of $2K + 1$ bands by $2N + 1$ frames centered at the current band of the frame, $v(n, k)$

$$h_m(n,k) = f\left(\sum_{i=-N}^{N} \sum_{j=-K}^{K} w_m(i,j) \cdot v(n+i, k+j) + b_m \right), \quad (1)$$

where $h_m(n, k)$ represents the neuron of the m^{th} feature map and f is a nonlinear function, typically a sigmoid function. M is the number of connection weights. The weights $\boldsymbol{W}$ are then shared across the entire input space, as indicated in the figure. Each local connection (local window) is considered as a filter to extract important spectro-temporal features from input features. After computing the hidden units, each map is subsampled with mean or max pooling to remove variability in the hidden units (i.e. convolutional band activations), that exist due to gender, speaking style, channel distortion, etc. This paper uses the max pooling that receives activations from r convolutional bands and outputs the maximum of the activations from these bands. A fully connected network is then added after the max-pooling layer to integrate the pooling features.

2.2. Linear Discriminant Analysis

LDA is a well-known feature extraction technique to make ASR more robust. The basic idea of LDA is to find a projection of the data where the variance between the classes is large compared to the variance within the classes. This can be stated formally as finding a projection matrix $\boldsymbol{\theta}$ that maximizes the objective function

$$J(\boldsymbol{\theta}) = \frac{det(\boldsymbol{\theta}^T \boldsymbol{\Sigma}_b \boldsymbol{\theta})}{det(\boldsymbol{\theta}^T \boldsymbol{\Sigma}_w \boldsymbol{\theta})}, \quad (2)$$

where $\boldsymbol{\Sigma}_b$ is the between-class covariance matrix and $\boldsymbol{\Sigma}_w$ is the shared within-class covariance matrix. The solution to this maximization problem is to take the first p eigenvectors of the matrix $\boldsymbol{\Sigma}_w^{-1} \boldsymbol{\Sigma}_b$ for a p dimensional projection. As with CNN, the eigenvectors that compose the projection matrix are regarded as filters that represent important variations on spectral patterns. In LDA, there are several choices for the estimation process depending on the class definitions. The typical choice is to use phonemes as classes [17]. Using classes corresponding to HMM states (leaves) is another choice that has been shown to improve the performance [18, 19]. In this paper, we use the HMM state levels as the LDA class definitions. Both CNN and LDA are based on supervised training, so the resultant filters are dependent on the characteristics of training data and objective function.

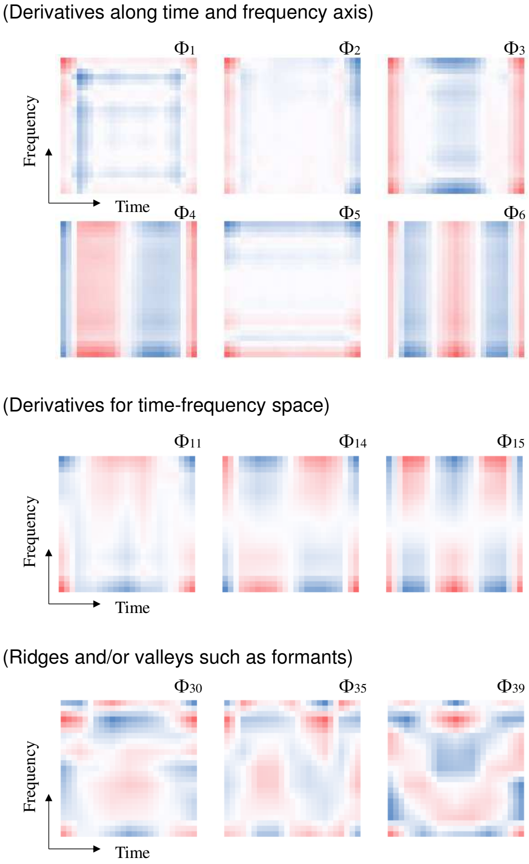
2.3. Connection Weights Initialized by LDA Eigenvectors

This paper proposes using LDA eigenvectors as initial local connection weights in CNN. First, LDA eigenvectors are estimated with training data by using the objective function described in Section 2.2 and then initial weights for the first convolutional layer in CNN are partially or fully replaced by the eigenvectors of LDA. What we need here is to make supervectors for LDA with the same size as the local windows in CNN, that is, $2K + 1$ bands by $2N + 1$ frames. $D - 2K$ supervectors can be created for the current frame, where D is the number of dimensions of input feature. Note that the number of dimensions of the supervector is smaller than that of a supervector used in a standard LDA-based feature projection [20] because the supervector in our proposed method is created with the size of CNN local windows. By solving the maximization problem on LDA, eigenvectors $\boldsymbol{\theta} = [\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \cdots, \boldsymbol{\theta}_p, \cdots, \boldsymbol{\theta}_{(2K+1)\times(2N+1)}]$ are obtained. First p eigenvectors that have large eigenvalues are used to create initial shared connection weights $\hat{\boldsymbol{W}} = [\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \cdots, \boldsymbol{\theta}_p, \boldsymbol{w}_{p+1}, \boldsymbol{w}_{p+2}, \cdots, \boldsymbol{w}_M]$ for the first convolutional layer. The connection weights in CNN initialized with LDA eigenvectors are updated with the usual pretraining and fine-tuning steps. Although weights after second convolutional layers could be initialized with a scheme similar to that of the proposed method, this paper focuses only on the initialization of the first convolutional layer. Since the proposed method is applied to the initial values of local windows, there is no need to change the architecture of CNN or the network training scheme.

2.4. Analysis of LDA Eigenvectors

This section discusses the characteristics of eigenvectors obtained by LDA, which are used as initial weights in CNN. Figure 2 shows the upper nine eigenvectors of 9×9 blocks ($N = 4, K = 4$) on a log Mel-filterbank sequence extracted by LDA. Other distinctive eigenvectors are also depicted in the figure. Speech data used here is the training data set ('Lecture Set') described in Section 3.1. The number of dimensions of log Mel-filterbank coefficients is 40. In the figure, red and blue colors represent positive and negative values, respectively. Each eigenvector image was smoothed by linearly interpolating an element of the eigenvector to make a comparison visually easy.

From a filter-operational point of view, $\boldsymbol{\theta}_1$ is considered to be a smoothing operator and as such is neutral characteristics among all of eigenvectors. It is stated in the literature that $\boldsymbol{\theta}_1$ has generally no effect on feature extraction for ASR. Filters $\boldsymbol{\theta}_2$, $\boldsymbol{\theta}_3$, $\boldsymbol{\theta}_4$, and $\boldsymbol{\theta}_6$ are considered to correspond to from first-order to fourth-order derivative operators with respect only to the time axis. $\boldsymbol{\theta}_5$ is regarded as the first-order derivative operator along the frequency axis only. These filters from $\boldsymbol{\theta}_2$ to $\boldsymbol{\theta}_6$ capture spectral variation individually along the time or the frequency axis while filters $\boldsymbol{\theta}_{11}$, $\boldsymbol{\theta}_{14}$, and $\boldsymbol{\theta}_{15}$ repre-


(Derivatives along time and frequency axis)

(Derivatives for time-frequency space)

(Ridges and/or valleys such as formants)

Fig. 2. Eigenvectors obtained by LDA.

sent 2D spectral variations spanning both time and frequency axes. In contrast, filters $\boldsymbol{\theta}_{30}$, $\boldsymbol{\theta}_{35}$, and $\boldsymbol{\theta}_{39}$ are subspaces that represent ridges and/or valleys (dynamic features) around formants on spectral patterns. Considering the remaining eigenvectors obtained by LDA, filters with large eigenvalues tend to capture comprehensive spectro-temporal variations such as first- and second-order derivatives. In contrast, eigenvectors with low eigenvalues represent relatively detailed variations on speech such as higher-order derivatives. Although LDA is a method based on the metric of class distribution, many of the resultant filters have a regular pattern reflecting principal components of speech. A comparison of the LDA and CNN filters is given in Section 4.

3. EXPERIMENT

3.1. Speech Data

The experiments presented in this paper are all based on speaker-independent models. We present the results on two in-house test sets in Japanese. These data sets focus on spontaneous speech and reverberation. The sampling frequency of both data sets is 16 kHz. The data sets are summarized as follows.

Lecture Data Set:
This data set is made up of lectures recorded at a university and consists of 83 hours of utterances by 147 speakers for training. The recorded speech data was manually transcribed. Test data is composed of the lectures of three speakers and consists of 1.6 hours in total and approximately 3K unique words. Subjects of the lectures, for example, pertain to politics and psychology, ranging from 7.5 minutes to 45 minutes per lecture. A microphone (pin mic) was attached close to lecturer's chest, so there is little ambient noise and reverberation. The speaking style is spontaneous.

Farfield Data Set:
This data set was recorded in a quiet meeting room (not an anechoic chamber) with microphone distances of 50 and 100 cm. The data set includes little ambient noise but does reverberation. Training data consists of 55 hours of speech including a 'command and control' utterance, an address, a personal name, a phone number, and a short messaging task, all of which are a read-speech style. Test data is comprised of 2,000 sentences of the same contents for each microphone distance and is uttered by 5 male and 5 female speakers.

3.2. CNN Architecture

The frontend acoustic features are 40-dimensional log Mel-filterbank coefficients computed from 25-ms frames with a 10-frame shift and are fed to CNN. The sampling frequency was 16 kHz. Speaker-level mean and variance normalization for the static features, where the statistics were calculated only on the speech regions of the data, were used throughout the CNN training step. CNN is combined with HMM to form a hybrid CNN/HMM system.

The first convolutional layer has 128 hidden units, the second has 256 hidden units, and the following five fully connected layers have 1024 hidden units each. In our experiments, the first p initial local weights in 128 windows of the first convolutional layer were replaced with LDA-based eigenvectors. LDA was performed with the same training data set as CNN. For the baseline CNN, all of 128 local windows were initialized randomly. Only the initialization step for local windows before pre-training is different between the baseline and the proposed method. The size of the local windows is 9×9 for the first convolutional layer and 3×4 for the second convolutional layer. We used max-pooling in frequency only (not time) for the pooling layer. This was shown to be optimal for ASR [21]. The pooling size in these experiments is 3 for both first and second pooling layers. Sigmoid function is used as the nonlinear activation function for hidden units. The softmax layer has 5000 output targets corresponding to the context-dependent phonemes obtained by growing a phonetic decision tree with a quinphone cross-word context.

The training data of the CNN was fully randomized at the frame level and the network was trained using stochastic gradient descent on minibatches of 250 frames with a cross-entropy criterion. Prior to the cross-entropy training of the full network, layer-wise discriminative pre-training was used for the fully connected layers (DNN layers) by running one cross-entropy sweep over the training data for the intermediate networks that had been obtained by adding one hidden layer at a time. The cross-entropy training converged after 15 iterations.

3.3. Results

The experimental results for each test set are provided in Tables 1 and 2. In the tables, we changed the number of weights p initialized with LDA eigenvectors. Notations of 'mp001', 'mp005', and 'mp006' in Table 1 indicate speaker ID in the test set. Tables 1 and 2 also include the results when 40-dimensional LDA features were input to CNN (Baseline CNN-LDA). The LDA features were extracted

Table 1. Results with lecture data set.

CNN	%CER (Character Error Rate)			
	mp001	mp005	mp006	AVERAGE
Baseline CNN-logMel	14.1	21.7	12.3	16.0
Baseline CNN-LDA	15.4	23.8	13.0	17.4
Proposed CNN-logMel (p=32)	13.6	21.2	12.1	15.6
Proposed CNN-logMel (p=64)	13.3	20.6	10.2	14.7
Proposed CNN-logMel (p=81)	13.1	20.6	12.0	15.2

Table 2. Results with farfield data set.

CNN	%CER (Character Error Rate)		
	50cm	100cm	AVERAGE
Baseline CNN-logMel	16.3	18.4	17.4
Baseline CNN-LDA	17.4	20.5	19.0
Proposed CNN-logMel (p=32)	15.5	17.6	16.6
Proposed CNN-logMel (p=64)	15.1	16.9	16.0
Proposed CNN-logMel (p=81)	15.4	17.3	16.4

by projecting 117-dimensional PLP features composed of consecutive 9-frame PLP with 13-dimensions down to 40 dimensions. Note that the LDA features for Baseline CNN-LDA were not used as initial values of local windows but were only used as the input features to the CNN.

From Table 1 for the lecture set, we can see that our proposed method outperformed the baseline CNN for all of the speakers. In our experiments, accuracy was best when 64 eigenvectors of LDA were used as initial weights for CNN, and the relative improvement against the baseline was 8.1% on average. Next, we consider the farfield test set shown in Table 2. We can see a similar trend as the lecture set, with the proposed method performing significantly better than the baseline. For this test set, our proposed method achieved 8.0% error reduction when the number of eigenvectors p was again set to 64.

4. DISCUSSION

We compare the convolutional filters obtained by LDA alone, CNN alone, and CNN initialized with LDA eigenvectors. Figure 3 picks up three characteristic filters $\boldsymbol{\theta}_3$, $\boldsymbol{\theta}_{15}$, and $\boldsymbol{\theta}_{30}$ from Figure 2. The filters we discuss here were all obtained with the training data of the lecture set discussed in Section 3. We first examine filters by LDA and CNN alone. The middle part of the figure shows three filters of the first convolutional layer of the baseline CNN for the ASR experiments in Section 3, obtained after pre-training and fine-tuning. Unlike the eigenvalues in LDA, there is no metric to measure the importance of local windows in CNN. Thus, we show the filters that are most similar to LDA filters shown above, in cosine similarity distance between each LDA-based filter of $\boldsymbol{\theta}_3, \boldsymbol{\theta}_{15}, \boldsymbol{\theta}_{30}$ and all filters in CNN. As shown in the figure, the shape of the filters estimated by CNN roughly resembles that from LDA, but what the filters capture is quite different. For example, $\boldsymbol{\theta}_3$ in LDA is regarded as the second-order time derivative filter, but the most similar one from CNN is not obviously a simple time derivative filter but a more complicated one.

Next we address the filters obtained by our proposed method (bottom images in Figure 3). We can see the shapes of the filters initialized by LDA and then optimized by CNN training are transformed from the original shapes optimized by LDA. This suggests that the local filters estimated by our proposed method better fit the training data compared to LDA alone and also have the classification capability that LDA originally has. In addition, as already shown in the previous section, we see the best ASR accuracy when the number of eigenvectors p was set to 64. This means that the half size of the first convolutional layer is derived from LDA and the other half is purely calculated by iterations of CNN training starting from random values. In other words, the resultant convolutional layer is considered to have two kinds of features, and hence the hybrid usage of filters from LDA and CNN provides complementariness at the feature level similarly to complementariness obtained by feature combination techniques.

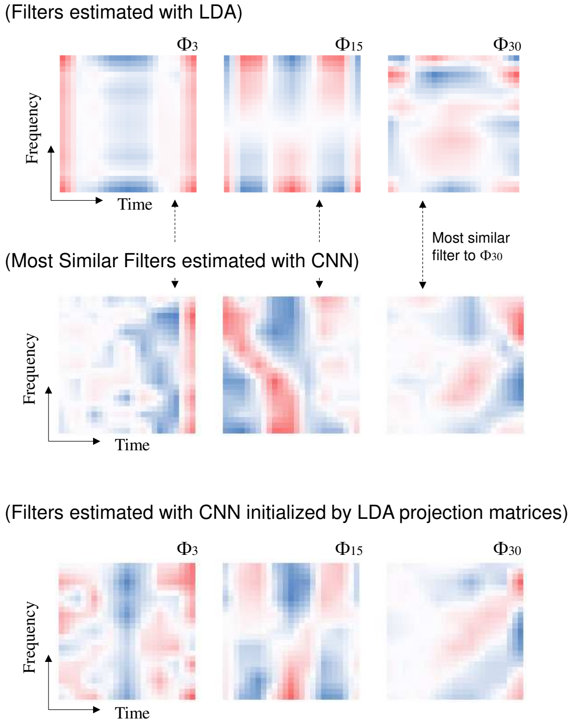


Fig. 3. Comparisons of local filters obtained by LDA, CNN, and CNN initialized with LDA eigenvectors.

5. CONCLUSION

This paper proposed initializing a part of local connection weights in the first convolutional layer of CNN by using eigenvectors estimated from the LDA objective function. Eigenvectors are obtained from supervectors created by the same size as local windows in CNN. Because LDA is designed to have discriminative characteristics, replacing about half of initial weights by the eigenvalues of LDA can provide an additional discriminative capability to CNN. The proposed initialization technique showed gains of up to 8.1% relative for the lecture-style data set, and 8.0% relative for the farfield data set over the standard CNN initialized with random values. We also compared filters obtained by LDA and CNN, and found that the filters obtained from CNN initialized by LDA eigenvectors have an intermediate property between LDA and CNN, and provide complementariness in the form of spectro-temporal feature extraction. In the future, we will investigate factorial connection weight initialization by partitioning training data.

6. REFERENCES

[1] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient based Learning Applied to Document Recognition," *Proceedings of the IEEE*, vol. 86, pp. 2278-2324, 1998.

[2] Y. LeCun, F. Huang, and L. Bottou, "Learning Methods for Generic Object Recognition with Invariance to Pose and Lighting," *Proc. CVPR*, vol. 2, pp. 97-104, 2004.

[3] S. Lawrence, "Face recognition: A Convolutional Neural Network Approach," *IEEE Transaction on Neural Networks*, vol. 8, no. 1, pp. 98-113, 1997.

[4] T. N. Sainath, A. Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep Convolutional Neural Networks for LVCSR", *Proc. ICASSP*, pp. 8614-8618, 2013.

[5] T. N. Sainath, B. Kingsbury, A. Mohamed, G. Dahl, G. Saon, H. Soltau, T. Beran, A. Aravkin, and B. Ramabhadran, "Improvements to Deep Convolutional Neural Networks for LVCSR," *Proc. ASRU*, pp.315-320, 2013.

[6] N. Mesgarani, S. Thomas, and H. Hermansky, "A Multistream Multiresolution Framework for Phoneme Recognition", *Proc. Interspeech*, pp. 318-321, 2010.

[7] H. Hermansky, P. Fousek, "Multi-resolution RASTA filtering for TANDEM-based ASR," *Proc. Interspeech*, pp.361-364, 2005.

[8] M. R. Schadler, B. T. Meyer, and B. Kollmeier, "Spectro-temporal modulation subspace-spanning filter bank features for robust automatic speech recognition," *Acoust. Soc. Am. 131(5)*, pp. 4134-4151, 2012.

[9] M. Kleinschmidt, "Localized spectro-temporal features for automatic speech recognition," *Proc. Eurospeech*, pp. 2573-2576, 2003.

[10] T. Fukuda, O. Ichikawa, and M. Nishimura, "Long-term spectro-temporal and static harmonic features for voice activity detection," *IEEE Journal of Selected Topics in Signal Processing*, Vol. 4, No. 5, pp. 834-844, 2010.

[11] N. Mesgarani, D. Stephen, S. Shamma, "Representation of Phonemes in Primary Auditory Cortex: How the Brain Analyzes Speech," *Proc. IEEE ICASSP*, pp. 765-768, 2007.

[12] R. Drullman, J. M. Festen, and R. Plomp, "Effect of temporal envelope smearing on speech perception," *J. Acoust. Soc. Amer.*, Vol. 95, pp. 1053-1064, 1994.

[13] R. Drullman, J. M. Festen, and R. Plomp, "Effect of reducing slow temporal modulations on speech perception," *J. Acoust. Soc. Amer.*, Vol. 95, pp. 2670-2680, 1994.

[14] N. Kanedera, T. Arai, H .Hermansky, and M. Pavel, "On the relative importance of various components of the modulation spectrum for automatic speech recognition," *Speech Communication*, vol. 28, no. 1, pp. 43-55, 1999.

[15] D. Poeppel, "The analysis of speech in different temporal integration windows: cerebral lateralization as asymmetric sampling in time," *Speech Communication*, Vol. 41, pp. 245-255, 2003.

[16] G. Saon, M. Padmanabhan, R. Gopinath, and S. Chen, "Maximum likelihood discriminant feature spaces," *Proc. ICASSP*, pp. 1129-1132, 2000.

[17] S. S. Kajarekar, B. Yegnanarayana, and H. Hermansky, "A study of two dimensional linear discriminants for ASR," *Proc. ICASSP*, pp. 137-140, 2001.

[18] K. Beulen, L. Welling, and H. Ney, "Experiments with linear feature extraction in speech recognition," *Proc. Eurospeech*, pp. 1415-1418, 1995.

[19] R. H. Umbach and H. Ney, "Linear discriminant analysis for improved large vocabulary continuous speech recognition," *Proc. ICASSP*, pp. 13-16, 1992.

[20] H. Soltau, G. Saon, B. Kingsbury, H. K. J. Kuo, L. Mangu, D. Povey, and A. Emami, "Advances in Arabic speech transcription at IBM Under the DARPA GALE program," *IEEE Trans. Audio, Speech, and Language processing*, Vol. 17, No. 5, pp. 884-894, 2009.

[21] O. Abdel-Hamid, A. Mohamed, H. Jiang, and G. Penn, "Applying Convolutional Neural Network Concepts to Hybrid NN-HMM Model for Speech Recognition," *Proc. ICASSP*, pp. 4277-4280, 2012.