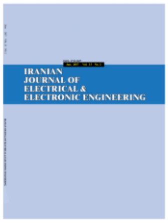
IUST Iran University of Science and Technology

Iranian Journal of Electrical and Electronic Engineering

Journal Homepage: ijeee.iust.ac.ir

Research Paper



An Information-Theoretic Discussion of Convolutional Bottleneck Features for Robust Speech Recognition

B. Nasersharif*(C.A.) and N. Naderi*

Abstract: Convolutional Neural Networks (CNNs) have been shown their performance in speech recognition systems for extracting features, and also acoustic modeling. In addition, CNNs have been used for robust speech recognition and competitive results have been reported. Convolutive Bottleneck Network (CBN) is a kind of CNNs which has a bottleneck layer among its fully connected layers. The bottleneck features extracted by CBNs contain discriminative and rich context information. In this paper, we discuss these bottleneck features from an information theory viewpoint and use them as robust features for noisy speech recognition. In the proposed method, CBN inputs are the noisy logarithm of Mel filter bank energies (LMFBs) in a number of neighbor frames and its outputs are corresponding phone labels. In such a system, we showed that the mutual information between the bottleneck layer and labels are higher than the mutual information between noisy input features and labels. Thus, the bottleneck features are a denoised compressed form of input features which are more representative than input features for discriminating phone classes. Experimental results on the Aurora2 database show that bottleneck features extracted by CBN outperform some conventional speech features and also robust features extracted by CNN.

Keywords: Bottleneck Features, Convolutional Neural Network, Convolutive Bottleneck Network, Mutual Information, Robust Speech Recognition.

1 Introduction

IN recent years, Deep Neural Networks (DNNs) have occupied a very important role in extending Automatic Speech Recognition (ASR) systems where they are used for acoustic modeling, feature extraction and transformation, and also constructing end to end ASR systems. Different kinds of DNNs have been used for speech processing: deep Belief Network (DBN), deep Auto-encoder, deep Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), and Recurrent Neural Network (RNN) [1–5]. DBNs and auto-encoders have been utilized for speech enhancement and also robust speech feature extraction [1–3], [6, 7]. Furthermore, DBNs have been applied to acoustic modeling accompanied by hidden Markov models (HMMs) [8]. However, LSTM and RNN have been directly employed for acoustic modeling [9].

Convolutional neural networks are a class of neural networks consisting of alternating convolution and pooling layers [10,11]. CNNs have been extensively used as acoustic models along with HMMs, in which, HMM state observation likelihoods are estimated using CNNs. In this case, CNN inputs are speech spectrogram [10–14] or raw speech signal [15–17].

However, CNN has been utilized as a feature extractor by many researchers. For instance, CNN and DBN have been applied to Large-Scale ASR task as feature extractors [18]. In [19, 20], very deep CNN (up to ten layers) is used for robust speech recognition using broader padding in order to keep the feature size compatible for adding more CNN layers. In our other research [21], CNN has been used as a robust feature extractor from noisy speech spectrogram in two ways: fixed resolution and multiresolution convolution filters.

Iranian Journal of Electrical and Electronic Engineering, 2021.
Paper first received 12 August 2019, revised 10 October 2020, and accepted 17 October 2020.

* The authors are with the Department of Computer Engineering, K. N. Toosi University of Technology, Tehran, Iran.
E-mails: b.nasersharif@kntu.ac.ir and navid.naderi@email.kntu.ac.ir.
Corresponding Author: B. Nasersharif.

https://doi.org/10.22068/IJEEE.17.2.1563

In addition, CNNs have been successfully used in other fields. In [22], the authors applied CNN to Language Identification as a feature extractor and also classifier. Moreover, features for speech activity detection in noisy conditions have been extracted using CNN [23]. Also, CNN has been effectively used in the presence of additive noise for image processing [24] and also other image processing applications such as facial key point detection and face recognition [25]. The combination of CNN, Long Short-Term Memory (LSTM), and DNN has been also reported for speech recognition and feature extraction [26, 27].

Convolutive Bottleneck Network (CBN) can be considered as a kind of CNNs, where its fully connected layers include a narrow bottleneck layer. Bottleneck layer outputs of a trained CBN have usually been used as features. An output layer with the *SoftMax* activation function with the aim of classification is usually added on the top of fully connected layers of CBN [28–30].

CBNs have also been used for feature extraction. For example, in [28, 31, 32], disorder dependent features have been extracted using CBNs for dysarthric speech recognition. Additionally, in [29], CBN features are used for bi-directional Generalize Variable Parameter HMMs (GVP-HMMs). Moreover, CBNs have been used in Audio-Visual speech recognition as a feature extractor [33]. Also, Large Vocabulary Conversational Speech Recognition is done using CBNs in [34].

Recently, researchers have discussed DNNs from an information-theoretic view and information bottleneck principle [35,36]. DNN training has been evaluated using mutual information between the layers and input/output variables. In [37], layers of an auto-encoder are evaluated from an information theory point of view where the information plane[1] of the network during different epochs are discussed. Also, the direct effect of training data size on the layers in the information plane has been shown as well as the importance of the symmetric architecture of such networks. Information-theoretic concepts have also been used in order to understand the inner organization of CNN using an extended *Rényi* α-entropy function for convolution layers. Also, the effect of increasing the number of layers has been discussed, and it has been shown that adding more convolution-pooling layers may increase the generalization power of CNNs, but the extra numbers of layers could tend to information loss [38].

In this paper, we discussed the bottleneck layer in CNN from an information-theoretic view and show that this layer is a good representation for noisy speech since the mutual information between this layer and phone label is greater than the mutual information between noisy speech features and phone labels. Based on this fact, we propose to use the CBN bottleneck layer for extracting robust speech features, where CBN inputs are the noisy logarithm of Mel filter bank energies (LMFBs) and its outputs are corresponding phone labels.

The remainder of this paper is organized as follows. Section 2 introduces the proposed method and mutual information analysis of the proposed method. Section 3 includes our experimental results. Finally, the conclusion is given in Section 4.

2 Proposed Method

In CBNs, the bottleneck layer represents a condensed feature vector obtained from a large input feature map. It is expected that the bottleneck layer can aggregate the propagated information and extract fundamental features included in an input map [39, 40]. The method for converting a high dimensional feature vector (X) into a compressed low dimensional feature vector, called bottleneck, is desirable when it preserves the maximum possible information of the original feature vector according to the desired label (L). So, in the next sub-section, we describe the computation method of the aforementioned information.

2.1 Mutual Information Analysis for CBN

In this paper, we try to obtain proof, based on information theory for the usefulness of the bottleneck features extracted by CBN. In any feedforward DNN including CNN, a data processing inequality has been proposed and validated based on Markov property as (1) [38]:

$$I(X,T_1) > I(X,T_2) > \ldots > I(X,T_N) \tag{1}$$

where $I(X, T_i)$ denotes the mutual information between network input (X) and the i-th layer of DNN (T_i).

By rewriting (1) for CNN, we have:

$$I(X,C_1) > \ldots > I(X,C_M) > I(X,T_1) > \ldots > I(X,T_N) \tag{2}$$

where C_i and T_i denote i-th convolution and i-th fully connected layer respectively [38]. However, in CBN we expect that:

$$I(X,T_{\lceil \frac{N}{2} \rceil}) \geq I\left(X,T_i\right), \quad \forall i < \left\lceil \frac{N}{2} \right\rceil \tag{3}$$

where N is the number of fully connected layers and therefore $T_{\lceil \frac{N}{2} \rceil}$ is the bottleneck layer.

We computed the mentioned mutual information in order to show the correctness of (3) as in [41]. According to the Shannon definition, the mutual information of a pair of variables $Z = \{z_1, z_2, \ldots, z_n\}$ and $G = \{g_1, g_2, \ldots, g_n\}$ is expressed as below [41]:

$$I(Z,G) = H(Z) + H(G) - H(Z,G) \tag{4}$$

[1] The plane of mutual information values that each layer preserves on the input and output variables” [37].

where $H(Z)$ and $H(G)$ denote entropies of Z and G, respectively, and $H(Z, G)$ indicates their joint entropy calculated as (5) [41]:

$$H(Z,G) = H\left(\frac{Z \circ G}{tr(Z \circ G)}\right) \tag{5}$$

where ∘ denotes Hadamard product. Using a Gaussian kernel $k_{\sigma\sqrt{2}}(a, b) = \exp(-\frac{1}{2\sigma^2}\|a-b\|^2)$, the entropy of Z can be estimated as (6) [41]:

$$\hat{H}(Z) = -\log \frac{1}{n^2}\sum_{i,j=1}^{n} k_{\sigma\sqrt{2}}(z_i, z_j) \tag{6}$$

With the computation of the mutual information as noted in (4)–(6), we analyze the usefulness of the proposed features.

2.2 Extracting Robust Bottleneck Features from CBN

As mentioned earlier, speech spectrogram or raw speech signal can be used as inputs of CNN for extracting informative features from them. We have used CNN for learning robust filter banks in our previous work where it outperformed Mel Filter bank and also DBN [21]. Since CBN inherits the advantages of both CNN and bottleneck layer, we expect that bottleneck features be more robust to noise. We want to obtain maximum information from noisy input features about phone labels which possibly improve the noisy speech recognition rate. We think that the bottleneck features extracted from an appropriately trained CBN can compress important information from a mass of noisy features because of convolution and pooling layers while they also include rich context information according to the classification layer.

Our proposed architecture for CBN has been shown in Fig. 1. LMFBs in 11 successive frames are fed into the CBN where the output is the phone label (one of 18 phones existed in the Aurora2 [42] database) corresponding to the center frame. As a result, robust contextual features can be encoded in the bottleneck layer. The extracted bottleneck features are used for training and testing a GMM-HMM system for ASR.

2.2.1 Convolution and Pooling Layers

We determine our CBN structure based on previous studies and our experiments [10, 28, 33] as in Fig. 1. We have used 1-4 convolution (and pooling) layers for our structures.

The convolution filter and pooling sizes have been selected according to previous studies. However, we should adjust these sizes to find the best parameters for our dataset and practice [18, 28, 29]. The number of neurons in each convolution (and pooling) layer depends on the application and dataset. Thus, this is another parameter that should be adjusted.

2.2.2 Fully Connected Layers

As can be seen from Fig. 1, the structures of fully connected layers in our CBN are symmetric, where the number of neurons in the layers after and before the bottleneck layer are equal symmetrically. The number of fully connected layers is an odd number and more than two, e.g. three or five. The number of fully connected layers and the number of neurons in each layer are parameters which should be adjusted. Thus, the number of bottleneck features is also a parameter.

3 Experiments and Results

3.1 Experimental Setup

Experiments have been performed on the Aurora2

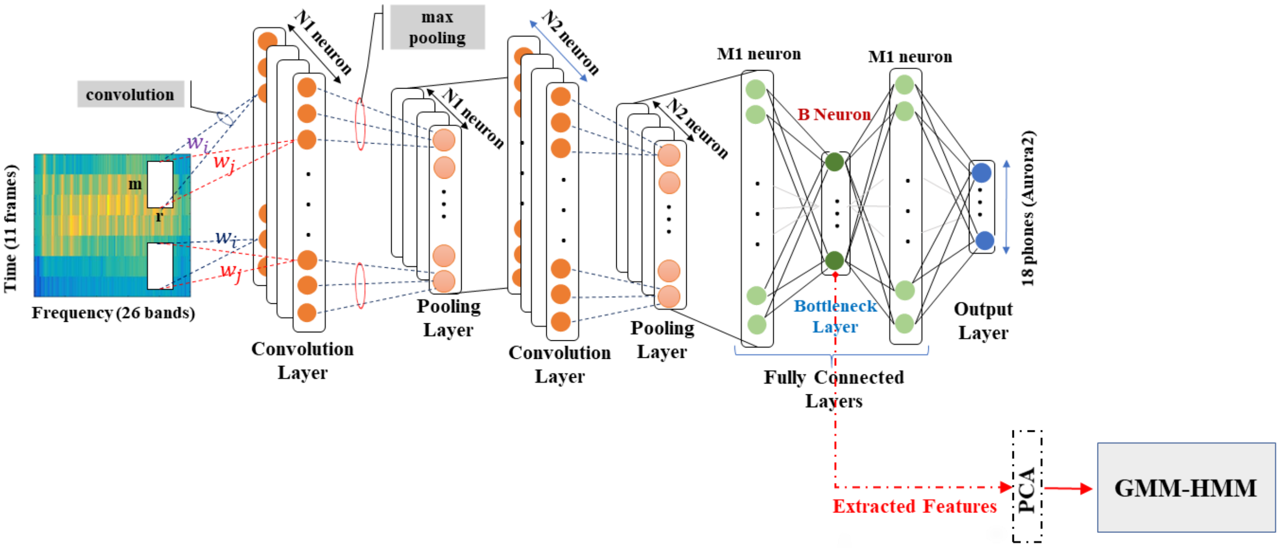


Fig. 1 The proposed CBN architecture.

database. The frame size is 25-ms with 15-ms overlap. The number of Mel filters is equal to 26. We used HMMs with 16 states, each containing 3 Gaussian mixtures for acoustic modeling, trained on the clean speech. CBN systems have been built using CNTK [43].

We used Aurora2 multi-condition noisy set for training CBN where its labels are provided using corresponding phone classes. Due to the positive and negative values in the CBN input, we have used the hyperbolic tangent (*TanH*) activation function. We have used stochastic gradient descent and mini-batch size of 50-500 for CBN training where we used momentum term after 5th epoch.

The learning rates in the first three epochs for convolution, hidden and output layers are equal to 1, 0.1, and 0.001, respectively. The learning rate decreases until 0.3% of the primitive one for 4th–40th epochs. The number of epochs has been selected equal to 40 which has been shown the best performance in our experiments.

3.2 Number of Neurons in the Bottleneck Layer

The number of neurons in the bottleneck layer (the number of bottleneck features) is the most important parameter in CBN. Thus, we have tested the different numbers of neurons in the bottleneck layer while the other parameters have kept fixed.

According to the previous studies, around 30 bottleneck features in CBN is a reasonable choice [28]. The recognition results for the different number of bottleneck features have been shown in Table 1. In this experiment, we use one convolution (and pooling) layer with 120 neurons in which the convolution filter size is equal to 11×7, and the pooling size is equal to 1×2. One fully connected layer with 500 neurons has been included before and after the bottleneck layer. As can be seen from Table 1, 50 neurons in the bottleneck layer produce the best recognition accuracy.

3.3 Number of Neurons in Convolution Layer

The number of neurons in the convolution layer should be tuned manually. We have conducted experiments on the number of neurons in the convolution layer and the results are shown in Table 2. These results indicate that using 150 convolution neurons tends to the best result.

Table 1 Average of recognition accuracy (on SNR 0-20DB and all noise types) for different bottleneck feature sizes.

Bottleneck	Average recognition accuracy			
features	A	B	C	AVG
30	77.03	74.12	77.03	75.76
50	78.11	75.72	76.64	76.82
100	76	73.01	75.63	74.88
150	77.19	74.74	76.07	76

3.4 Fully Connected Layers Results

In order to evaluate the effect of the fully connected layers, we change the number of hidden layers and neurons per each layer where the bottleneck layer is fixed (50 neurons in the layer). The other parameters are the same as Sections 3.2–3.3.

The results of different fully connected architectures for CBN have been shown in Table 3. It seems that three hidden layers including the bottleneck in the middle have the best performance for extracting the CBN bottleneck features.

3.5 Convolution and Pooling Layers Results

Since convolution and pooling layers perform as a low-level filter stage and extract the lower-level information to be fed to fully connected layers, its parameters can affect the system performance. Consequently, we used more convolution layers for our experiments. Convolution filter sizes have been selected based on the number of convolution-pooling layers. With increasing the number of convolution layers, we selected lower convolution filter sizes. For saving spatial information, we have utilized max-pooling only once in each structure.

The top-rated 1-4 convolution-pooling layers and configurations are listed in Table 4 and the average results of word recognition accuracy over different noise levels (-5dB–20dB) are shown in Fig. 2. The results of average recognition accuracy over different noise sets are listed in Table 5. One fully connected layer with 500 neurons has been included before and after the bottleneck layer, where the bottleneck layer has 50 neurons.

As can be seen from Table 5, when we increase the number of convolution-pooling layers to 3, the recognition accuracy improves. But, when we use 4

Table 2 Average of recognition accuracy (on SNR 0-20DB and all noise types) for the different number of neurons in the convolution layer.

#Convolution	Average recognition accuracy			
layer neurons	A	B	C	AVG
100	73.28	69.35	71.62	71.42
120	78.11	75.72	76.64	76.82
150	78.83	76.98	84.87	80.23
200	77.52	74.69	75.94	76.05
250	72.59	68.84	71.37	70.93

Table 3 Average of recognition accuracy (on SNR 0-20DB and all noise types) for different fully connected layers structures.

FC architecture	Average recognition accuracy			
	A	B	C	AVG
500-50-500	78.83	76.98	84.87	80.23
300-50-300	77.04	73.12	76.38	75.51
700-50-700	78.01	75.39	76.30	76.57
500-500-50-500-500	79.11	76.14	76.17	77.14

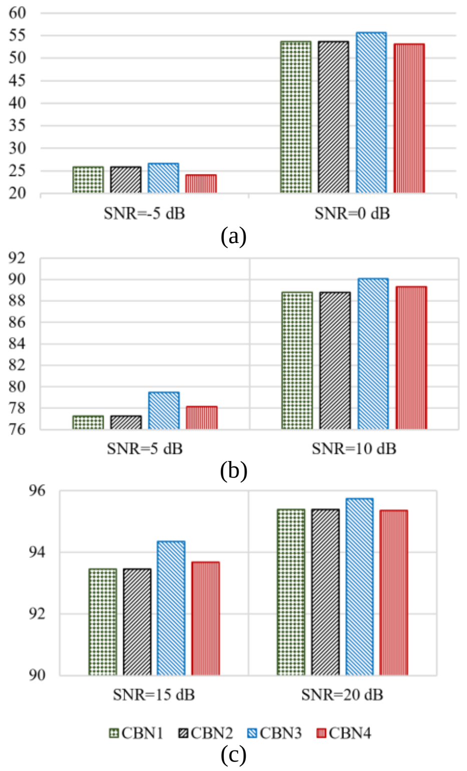
convolution layers, the recognition accuracy degrades. It can be due to the lack of training data for this deep network. The results of Fig. 2 indicate that three convolution-pooling layers outperform other CBN structures.

Table 4 The structures of CBNs, where lines in each row of ‘Structure’ column show the corresponding convolution-pooling layers, in which # shows the number of convolution-pooling neurons, C shows convolution filter size, and P shows pooling size.

Name	Layer	Structure
CBN1	1	#150 - C:11×7 - P:1×2
CBN2	1	#50 - C:7×5 - P:1×1
	2	#150 - C:5×3 - P:1×2
CBN3	1	#50 - C:5×3 - P:1×1
	2	#50 - C:5×3 - P:1×1
	3	#50 - C:3×3 - P:1×2
CBN4	1	#50 - C:5×3 - P:1×1
	2	#50 - C:3×3 - P:1×2
	3	#150 - C:3×3 - P:1×1
	4	#150 - C:3×3 - P:1×2

Table 5 Average of recognition accuracy (on SNR 0-20DB and all noise types) for 1-4 convolution and pooling layers.

Structure	Average recognition accuracy			
	A	B	C	AVG
CBN1	78.83	76.98	84.87	80.23
CBN2	82.64	81.20	80.88	81.57
CBN3	83.8	82.74	82.27	82.94
CBN4	82.83	81.65	80.71	81.73



(c)

Fig. 2 Averages of word recognition accuracy over different noise levels for CBNs with fully connected structure of 500-50-500; a) High level of noise, b) Medium level of noise, and c) low level of noise.

3.6 Discussion of Denoising Bottleneck Layer Based on Mutual Information

In order to show the effectiveness of bottleneck features extracted from CBN, we consider the mutual information as:

- $I(T_{n,\lceil \frac{N}{2} \rceil}, L)$: The mutual information between the bottleneck layer $T_{n,\lceil \frac{N}{2} \rceil}$ (where CBN input is noisy) and Label L.
- $I(X_n, L)$: The mutual information between noisy Input X_n and Label L.
- $I(X_c, L)$: The mutual information between clean input X_c and Label L.

For the best network structure obtained from experiments, we compared the average of $I(T_{n,\lceil \frac{N}{2} \rceil}, L)$ with the average of $I(X_n, L)$ and $I(X_c, L)$ over 5000 randomly selected frames in Fig. 3.

As can be seen from the figure, $I(X_c, L)$ is significantly higher than the $I(X_n, L)$, which is expected. On the other hand, $I(T_{n,\lceil \frac{N}{2} \rceil}, L)$ is higher than $I(X_n, L)$ where it is lower than $I(X_c, L)$. This is reasonable because we cannot perfectly reconstruct the clean features, but we can decrease noise effects on features using the bottleneck layer $T_{n,\lceil \frac{N}{2} \rceil}$. However, maximizing the mutual information between compressed features and phone labels as well as maximizing compression at the same time seems feasible since:

$$\text{dimension}\left(T_{n,\lceil \frac{N}{2} \rceil}, L\right) < \text{dimension}\left(X_n\right) \text{ and}$$

$$I\left(T_{n,\lceil \frac{N}{2} \rceil}, L\right) > I\left(X_n, L\right) \qquad (7)$$

This property is desirable since it shows that $T_{n,\lceil \frac{N}{2} \rceil}$ has sufficient information about the phone label. It also

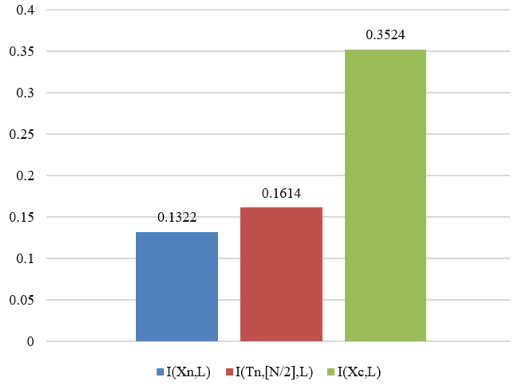


Fig. 3 Average of the mutual information between noisy input and target, the bottleneck features obtained from noisy input and target, clean input and target for 5000 frames.

shows that noise effects on X_n have been decreased in $T_{n,\left\lceil \frac{N}{2} \right\rceil}$ which means that $T_{n,\left\lceil \frac{N}{2} \right\rceil}$ is a denoised compressed form of X_n.

3.7 Discussion of CBN Training Based on Mutual Information

In this section, we discuss the training of the best CBN structure and the role of the bottleneck layer from the view of mutual information.

The mutual information of the best structure of CBN obtained from Sections 4.2 to 4.5 is analyzed in this section.

First of all, data processing inequality for CBN is validated in Fig. 4, as mentioned in Section 2.1 (where C_i and T_i have been defined in Section 2.1).

Furthermore, the mutual information between the bottleneck and label is more than the mutual information between input and label (0.2543 > 0.2315) which indicates a suitable representation of the bottleneck layer.

In addition to the best network structure, data processing inequality is validated for CBN with 4 convolution-pooling layers shown in Fig. 5.

As can be seen from Fig. 5, data processing inequality for this CBN structure is not validated. Also, in this case, the mutual information between the bottleneck and label is less than the mutual information between input and the label (0.2123 < 0.2315).

3.8 Final Discussions

In order to evaluate the overall system, we compare the best result of the previous sections with the results of the most known conventional speech features such as LMFB, LMFB+CMVN, MFCC, MFCC+CMVN, and robust features extracted using CNN and DBN, which are listed in Table 6.

For a fair comparison, the CNN and DBN inputs include 11 consecutive frames. The used CNN has three convolution-pooling layers and three hidden layers. The used DBN has two hidden layers with 2048 neurons per layer. DBN and CNN inputs are noisy speech spectrograms and their outputs correspond to denoised LMFBs.

As can be seen from Table 6, the proposed features (CBN) and also proposed futures followed by PCA (CBN+PCA) outperform other features except features obtained by ETSI AEF. This performance can be due to the fact that the convolution and pooling layers construct robust features, while the bottleneck layer uses the phone class labels and makes the rich context robust features.

On the other hand, ETSI AFE uses Voice Activity Detection (VAD), noise estimation, two-pass Wiener filter-based noise suppression, and blind feature equalization techniques. The noise estimation is updated using non-speech frames attained from VAD [45]. Thus, it uses many preprocessing such as VAD and noise estimation techniques which our method does not consider. Therefore, it is logical that ETSI AFE outperforms our proposed method.

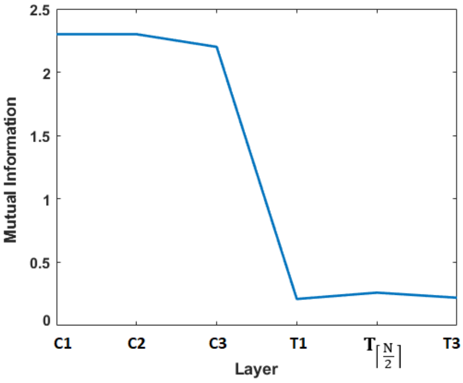


Fig. 4 The mutual information between input and each layer of a well-trained CBN.

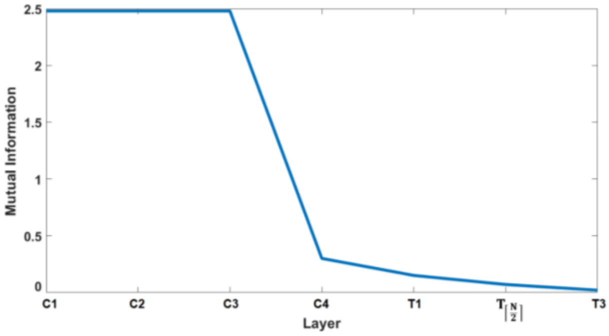


Fig. 5 The mutual information between input and each layer of a fair trained CBN.

Table 6 Average of recognition accuracy (on SNR 0-20DB and all noise types) for different speech feature extraction methods.

Method	Average recognition accuracy			
	A	B	C	AVG
LMFB	19.59	23.62	13.49	18.90
LMFB+CMVN	32.84	36	33.13	33.99
MFCC	63.06	66.66	60.18	63.30
MFCC+CMVN	76.52	79.88	70.61	75.67
CNN [21]	67.57	67.12	60.79	65.16
ETSI AFE [44]	86.69	85.57	82.81	85.02
CBN	83.8	82.74	82.27	82.94
CBN+PCA	84.25	83.28	82.76	83.43

4 Conclusion

In this paper, we discussed the information content of the Convolutive Bottleneck Network. We showed that the mutual information between noisy input features and the phone class label is lower than the mutual information between the bottleneck layer and the same labels.

Thus, we proposed to use bottleneck features extracted by CBN as robust features for noisy speech recognition. In the proposed method, noisy LMFBs in a number of successive frames are CBN inputs and corresponding phone labels are its outputs. We examined various structures for CBN including different numbers of convolution, pooling and fully connected layers and also the number of neurons in each layer, especially fully connected and bottleneck layers, and the network sensitivity to such parameters is confirmed. The experimental results show that CBN using three convolution-pooling pairs and three hidden layers outperforms CNN and has an acceptable performance.

In addition, we discussed this best structure based on the mutual information between input and each layer in the training phase where data processing inequality for input and each layer based on Markov property has been validated. Due to this property, the mutual information of noisy input and bottleneck has been higher than the mutual information of noisy input and output layer. Hence, the bottleneck layer can extract promising information about phone class labels from noisy input features.

From Real-Time Factor (RTF) viewpoint, our feature extraction method is about 3 times slower than real-time.

Conflict of Interest

The authors declare that they have no conflict of interest.

Acknowledgments

We would like to thank Dr. Ahmad Akbari and Audio and Speech Processing Lab (ASPL) of Iran University of Science and Technology (IUST) for supporting this research.

References

[1] F. Baniardalan, A. Akbari, and B. Nasersharif, "Bottleneck features extraction and denoising in the sub-band level using deep auto-encoders for speech recognition.," in *First International Conference on Signal Processing and Intelligent Systems*, pp. 58–63, 2015.

[2] M. Gholamipour and B. Nasersharif, "Robust MFCC extraction using deep belief network.," in *First International Conference on Signal Processing and Intelligent Systems*, pp. 70–75, 2015.

[3] K. Han, Y. He, D. Bagchi, E. Fosler-Lussier, and D. Wang, "Deep neural network based spectral feature mapping for robust speech recognition," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[4] T. Gao, J. Du, L. Dai, and C. Lee, "Densely connected progressive learning for LSTM-based speech enhancement," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* pp. 5054–5058, 2018.

[5] I. Ariav, D. Dov, and I. Cohen, "A deep architecture for audio-visual voice activity detection in the presence of transients," *Signal Processing,* Vol. 142, pp. 69–74, 2018.

[6] J. Du, Q. Wang, T. Gao, Y. Xu, L. R. Dai, and C. H. Lee, "Robust speech recognition with speech enhanced deep neural networks," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[7] X. Feng, Y. Zhang, and J. Glass, "Speech feature denoising and dereverberation via deep autoencoders for noisy reverberant speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* pp. 1759–1763, 2014.

[8] A. R. Mohamed, G. E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE Trans. Audio, Speech and Language Processing*, Vol. 20, No. 1, pp. 14–22, 2012.

[9] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Interspeech,* 2014.

[10] T. N. Sainath, A. R. Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep convolutional neural networks for LVCSR," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* pp. 8614–8618, 2013.

[11] O. Abdel-Hamid, L. Deng, and D. Yu, "Exploring convolutional neural network structures and optimization techniques for speech recognition," in *Interspeech,* pp. 1173–1175, 2013.

[12] O. Abdel-Hamid, A. R. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, Vol. 22, No. 10, pp. 1533–1545, 2014.

[13] 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 *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* pp. 4277–4280, 2012.

[14] J. T. Huang, J. Li, and Y. Gong, "An analysis of convolutional neural networks for speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* pp. 4989–4993, 2015.

[15] D. Palaz, R. Collobert, and M. M. Doss, "Estimating phoneme class conditional probabilities from raw speech signal using convolutional neural networks," in *Interspeech*, 2013.

[16] D. Palaz, M. M. Doss, and R. Collobert, "Convolutional neural networks-based continuous speech recognition using raw speech signal," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4295–4299, 2015.

[17] D. Palaz, M. Magimai.-Doss, and R. Collobert, "Analysis of CNN-based speech recognition system using raw speech as input," in *Interspeech*, pp. 11–15, 2015.

[18] 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*, Vol. 64, pp. 39–48, 2015.

[19] Y. Qian and P. C. Woodland, "Very deep convolutional neural networks for robust speech recognition," in *IEEE on Spoken Language Technology Workshop (SLT)*, pp. 481–488, 2016.

[20] T. Yoshioka, K. Ohnishi, F. Fang, and T. Nakatani, "Noise robust speech recognition using recent developments in neural networks for computer vision," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5730–5734, 2016.

[21] N. Naderi and B. Nasersharif, "Multiresolution convolutional neural network for robust speech recognition," in *Iranian Conference on Electrical Engineering (ICEE)*, pp. 1459–1464, 2017.

[22] A. Lozano-Diez, R. Zazo-Candil, J. Gonzalez-Dominguez, D. T. Toledano, and J. Gonzalez-Rodriguez, "An end-to-end approach to language identification in short utterances using convolutional neural networks," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[23] S. Thomas, S. Ganapathy, G. Saon, and H. Soltau, "Analyzing convolutional neural networks for speech activity detection in mismatched acoustic conditions," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2519–2523, 2014.

[24] R. Yeh, M. Hasegawa-Johnson, and M. N. Do, "Stable and symmetric filter convolutional neural network," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2652– 2656, 2016.

[25] S. Wu, J. Xu, S. Zhu, and H. Guo, "A deep residual convolutional neural network for facial keypoint detection with missing labels," *Signal Processing*, Vol. 144, pp. 384–391, 2018.

[26] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4580–4584, 2015.

[27] T. N. Sainath, R. J. Weiss, A. Senior, K. W. Wilson, and O. Vinyals, "Learning the speech front-end with raw waveform CLDNNS," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[28] Y. Takashima, T. Nakashika, T. Takiguchi, and Y. Ariki, "Feature extraction using pre-trained convolutive bottleneck nets for dysarthric speech recognition," in *23rd European Signal Processing Conference (EUSIPCO)*, pp. 1411–1415, 2015.

[29] R. Su, X. Liu, and L. Wang, "Convolutional neural network bottleneck features for bi-directional generalized variable parameter HMMS," in *International Conference on Information and Automation*, pp. 1126–1131, 2016.

[30] F. Grézl, M. Karafiát, S. Kontár, and J. Cernocky, "Probabilistic and bottle-neck features for LVCSR of meetings," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vol. 4, pp. IV–757–IV–760, 2007.

[31] T. Nakashika, T. Yoshioka, T. Takiguchi, Y. Ariki, S. Duffner, and C. Garcia, "Dysarthric speech recognition using a convolutive bottleneck network," in *12th International Conference on Signal Processing (ICSP)*, pp. 505–509, 2014.

[32] T. Nakashika, T. Yoshioka, T. Takiguchi, Y. Ariki, S. Duffner, and C. Garcia, "Convolutive bottleneck network with dropout for dysarthric speech recognition," *Transactions on Machine Learning and Artificial Intelligence*, Vol. 2, pp. 1–15, 2014.

[33] Y. Takashima, Y. Kakihara, R. Aihara, T. Takiguchi, Y. Ariki, N. Mitani, K. Omori, and K. Nakazono, "Audio-visual speech recognition using convolutive bottleneck networks for a person with severe hearing loss," *IPSJ Transactions on Computer Vision and Applications*, Vol. 7, pp. 64–68, 2015.

[34] K. Veselý, M. Karafiát, and F. Grézl, "Convolutive bottleneck network features for LVCSR," in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 42–47, 2011.

[35] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," *arXiv:physics/0004057*, 2000.

[36] N. Tishby and N. Zaslavsky, "Deep learning and the information bottleneck principle," in *IEEE Information Theory Workshop (ITW)*, pp. 1–5, 2015.

[37] R. Shwartz-Ziv and N. Tishby, “Opening the black box of deep neural networks via information,” *arXiv:1703.00810*, 2017.

[38] S. Yu, R. Jenssen, and J. C. Principe, “Understanding convolutional neural network training with information theory,” *arXiv:1804.06537*, 2018.

[39] Y. Bengio, “Learning deep architectures for AI,” *Foundations and trends in Machine Learning*, Vol. 2, No. 1, pp. 1–127, 2009.

[40] C. Sui, R. Togneri, and M. Bennamoun, “Extracting deep bottleneck features for visual speech recognition,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1518–1522, 2015.

[41] L. G. S. Giraldo, M. Rao, and J. C. Principe, “Measures of entropy from data using infinitely divisible kernels,” *IEEE Transactions on Information Theory*, Vol. 61, No. 1, pp. 535–548, 2015.

[42] H. G. Hirsch and D. Pearce, “The aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions,” in *Automatic Speech Recognition: Challenges for the new Millenium ISCA Tutorial and Research Workshop (ITRW)*, 2000.

[43] D. Yu, A. Eversole, M. Seltzer, K. Yao, Z. Huang, B. Guenter, O. Kuchaiev, Y. Zhang, F. Seide, and H. Wang, “An introduction to computational networks and the computational network toolkit,” *Microsoft Technical Report MSR-TR-2014–112*, 2014.

[44] A. Fazel and S. Chakrabartty, “Sparse auditory reproducing kernel (SPARK) features for noise-robust speech recognition,” in *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 20, No. 4, pp. 1362–1371, May 2012.

[45] D. Macho, L. Mauuary, B. Noé, Y. M. Cheng, D. Ealey, D. Jouvet, H. Kelleher, D. Pearce, and F. Saadoun, “Evaluation of a noise-robust DSR front-end on Aurora databases,” in *Seventh International Conference on Spoken Language Processing*, Sep. 2002.

B. Nasersharif received the B.Sc. degree in Hardware Engineering from the Amirkabir University of Technology (AUT), Tehran, Iran, in 1997. He received his M.Sc. and Ph.D. degrees in Computer Engineering (Artificial Intelligence) from Iran University of Science and Technology (IUST), Tehran, Iran, in 2001 and 2007 respectively. He is an Assistant Professor in the Computer Engineering Department, K. N. Toosi University of Technology from September 2011 until now. His research interests include speech processing, deep learning for speech processing, and pattern recognition.

N. Naderi received the B.Sc. degree in Computer Science from Chamran University, Ahvaz, Iran, in 2014, and the master’s degree in Artificial Intelligence from K. N. Toosi University of Technology, Tehran, Iran, in 2017. Where he is currently a Ph.D. candidate. His research interests include deep learning for speech processing and speech enhancement.