COMBINING SPARSE NMF WITH DEEP NEURAL NETWORK: A NEW CLASSIFICATION-BASED APPROACH FOR SPEECH ENHANCEMENT

Hung-Wei Tseng[1]*, Mingyi Hong*[2†]*, Zhi-Quan Luo*[1]

[1] Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455, USA
[2] Industrial and Manufacturing Systems Engineering, Iowa State University, Ames, IA 50011, USA

ABSTRACT

In this work, we consider enhancing a target speech from a single-channel noisy observation corrupted by non-stationary noises at low signal-to-noise ratios (SNRs). We take a classification-based approach, where the objective is to estimate an Ideal Binary Mask (IBM) that classifies each time-frequency (T-F) unit of the noisy observation into one of the two categories: speech-dominant unit or noise-dominant unit. The estimated mask is used to binary weight the noisy mixture to obtain the enhanced speech. In the proposed system, the sparse non-negative matrix factorization (NMF) is used to extract features from the noisy observation, followed by a Deep Neural Network (DNN) for classification. Compared with several existing classification-based systems, the proposed system uses minimal speech-specific domain knowledge, but is able to achieve better performance in certain low SNR regions. Moreover, the proposed system outperforms the traditional statistical method, especially in terms of improving the intelligibility.

Index Terms— Speech enhancement, deep neural network (DNN), non-negative matrix factorization (NMF), sparse coding

1. INTRODUCTION

Enhancing speech from a single-microphone noisy recording is an important task in many engineering systems, including hearing aids, mobile communication, and robust speech recognition. Two main metrics for evaluating enhancement algorithms are the quality and the intelligibility of the enhanced speech. In this paper, we propose a classification-based algorithm that improves both of these two metrics, under low input SNR (-5dB) and for three types of background noises: street noise, factory noise and babble noise. The considered scenarios are challenging – even normal-hearing listeners have less than 50% recognition rate [1].

Various speech enhancement algorithms have been proposed in the literature, including statistical-based methods such as the minimum mean-square error (MMSE) estimators [2], as well as the NMF-based algorithms [3–6]. Despite different approaches taken, all these algorithms aim to estimate a "soft gain" for producing the enhanced speech. In contrast, there are a few works that apply "hard gain" – those gain coefficients that only take value from 0 or 1 – and achieve surprisingly good enhancement results. For example, the authors of [7] have shown that using the Ideal Binary Mask (IBM) as the hard gain can significantly improve the intelligibility, compared with the traditional soft-gain based approaches [8]. In IBM, a binary mask is applied to the noisy mixture in the time-frequency (T-F) representation, so that the speech-dominant T-F units with high SNRs are kept and the rest of the T-F units are discarded. Despite its success, IBM is impractical because it requires access to *both* the actual noise and speech samples. However, only noisy mixture is accessible in the enhancement stage. To overcome this difficulty, several classification-based algorithms such as [1, 9–11] have been proposed. In these algorithms, features are extracted from the noisy mixture, and a classifier is applied for estimating IBM.

The success of all the classification-based algorithms mentioned above heavily depends on using sophisticated speech-specific features. For example, the Amplitude Modulation Spectrograms (AMS) are used in [1, 9]. A composite feature includes AMS, Relative Spectral Transform and Perceptual Linear Prediction (RASTA-PLP), Mel-Frequency Cepstral Coefficients (MFCC) and pitch-based features are used in [10]. Multi-Resolution Cochleagram (MRCG), a new feature that captures different resolutions of the T-F representation, is proposed in [11]. After the speech-specific features are extracted, different classifiers, such as the Bayesian classifier [1], the Support Vector Machine (SVM) [9], or the neural network [10, 11] can be applied.

In this paper, we propose a new algorithm for IBM-based speech enhancement. We demonstrate that superior enhancement performance can be achieved by using a simple feature extraction step followed by a DNN classifier, without requiring much speech-specific domain knowledge during the entire process. In particular, we use the standard Sparse NMF (SNMF) to extract features from the noisy mixture. This step is simple and easy to implement. Though SNMF has been applied to soft gain calculation before [3–6], this is the first time it is used for IBM estimation. The SNMF step is followed by a DNN classifier, which is a neural network with more than one hidden layers. Here we use DNN rather than SVM for classification because the former achieves a better performance in our experiment, which is in line with other reports [12, 13].

Notation: We use uppercase letters to denote matrices and lowercase letters to denote either vectors or scalars. X_n represents the n-th column of the matrix X, while the (k,n)-th entry is denoted by $X_{k,n}$. $\cdot/\cdot$, $\odot$ and $\geq$ denote entry-wise division, multiplication and "greater or equal to" respectively. $X^{\cdot\beta}$ means raising each entry of X to the power β. $\mathcal{R}^{K\times N}$ denotes the set of all $K\times N$ matrices with real entry values, while $\mathcal{R}_+^{K\times N}$ is defined similarly but with non-negative entries. $[A;B]$ denote the vertical concatenations of matrices A and B.

2. SYSTEM DESCRIPTION

We first provide an overview of the proposed system shown in Fig. 1. The details will be described in latter subsections. As a proof of concept, in this paper we consider a matched-noise condition, i.e., the noise types and the input SNRs in both the training and testing stages are the same. For each type of noise, a system like Fig. 1 is

*This work was carried out in part using computing resources at the University of Minnesota Supercomputing Institute (MSI).

†The second author performed this work while working as a research associate at University of Minnesota.

trained independently. The proposed system differs from [9–11] in how the feature extraction and the classification are performed.

In the training stage, time-domain signals are transformed into the T-F representation by passing through an *analysis front-end* and a *cochleagram* computation [14]. Then, we apply SNMF, a sparse variant of NMF originally designed for object recognition [15], to extract features. SNMF is divided into a *dictionary training* step and a *sparse coding* step. In the dictionary training step, a speech dictionary D^s and a noise dictionary D^v are learned from the speech cochleagram S and the noise cochleagram V, respectively. These dictionaries are used to extract features from the noisy cochleagram Y in the sparse coding step. The extracted features together with the true IBM are used to train the *DNN classifier*.

In the testing stage, noisy speech is first transformed into the cochleagram representation, and is then passed through the sparse coding block for feature extraction. Different from the training stage, the sparse coding here uses the dictionaries already trained in the training stage, and therefore does not require access to the actual speech and noise. Taking the extracted features as input, the trained DNN generates the estimated IBM. The *resynthesis back-end* takes the estimated IBM and produces the enhanced speech.

2.1. Analysis front-end, Cochleagram, IBM, and Resynthesis back-end

In the analysis front-end, audio files sampled at 16 kHz are passed through a 64-channel Gammatone filterbank with center frequencies spanning from 50 Hz to 8 kHz on the equivalent rectangular bandwidth rate scale. The output of each filter is divided into 20-ms segments with 10-ms overlap. Energy in each segment is calculated and then forms a T-F matrix called cochleagram. We use $Y \in \mathcal{R}_+^{64\times N}$ to denote the cochleaagram of the noisy speech, where N represents the total number of time frames. Let S and V denote the cochleagram of clean speech and noise, respectively.

The ideal binary mask matrix $B \in \mathcal{R}_+^{64\times N}$ is defined based on whether a T-F unit is either speech-dominant or noise-dominant:

$$B_{k,n} = \begin{cases} 1, & \text{if } \mathrm{SNR}_{k,n} \geq LC \\ 0, & \text{otherwise} \end{cases}$$

where $\mathrm{SNR}_{k,n} = \frac{S_{k,n}}{V_{k,n}}$ denotes the local SNR, and LC denotes the local SNR criterion which is set to -5 dB. Clearly, calculating IBM requires access to S and V, which can only be done in the training stage but not the testing stage.

In the resynthesis step, the estimated IBM (which will be explained later) is used to binarily weight the Gammatone filterbank output of each channel. The speech-dominant regions are kept intact while the noise-dominant regions are discarded. All the 64 weighted streams are then summed to produce the final enhanced speech. We use the implementation from Wang's group[1] for the analysis front-end, cochleagram calculation, and the resynthesis back-end.

2.2. Learning Speech and Noise Dictionary

We first present the idea and the computational algorithm for the dictionary learning step in SNMF, and then show how to specialize it to speech and noise. Dictionary learning factorizes a non-negative matrix $X \in \mathcal{R}_+^{K\times N}$ into the product of a dictionary $D \in \mathcal{R}_+^{K\times M}$ and a sparse code $G \in \mathcal{R}_+^{M\times N}$:

$$X \approx D \times G \tag{1}$$

where M denotes the size of the dictionary. Columns of D are called *atoms*. Since G is sparse, only a *few* atoms suffice to represent X.

[1] See the code available at `http://www.cse.ohio-state.edu/pnl/`.

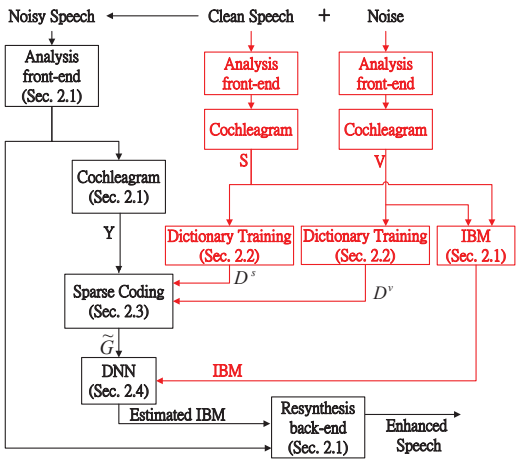


Fig. 1. Block diagram of the proposed system. Modules highlighted in red are those modules used only in the training stage; modules in black are those used in both training and testing stage.

In other words, the sparse code G necessarily contains important information about X. Computationally, factorization (1) is achieved by solving (2):

$$\min_{D\geq 0, G\geq 0} d_{\mathrm{IS}}(X \mid DG) + \lambda \sum_{m,n} \log(\epsilon + G_{m,n}) \tag{2}$$

where $d_{\mathrm{IS}}(\cdot \mid \cdot)$ denotes the Itakura-Satio (IS) divergence [16] between matrices A and B:

$$d_{\mathrm{IS}}(A \mid B) = \sum_{k,n} \left\{ \frac{A_{k,n}}{B_{k,n}} - \log \frac{A_{k,n}}{B_{k,n}} - 1 \right\}.$$

The second term in (2) is a sparsity-promoting regularization [17]. The larger the λ, the sparser the G. Moreover, ϵ is a small positive constant to make the term inside logarithm strictly positive. Similar to [16], (2) can be solved by:

$$D \leftarrow D \odot \left\{ \frac{\left[(DG)^{\cdot -2} \odot X\right] G^T}{(DG)^{\cdot -1} G^T} \right\}^{\cdot \frac{1}{2}} \tag{3}$$

$$G \leftarrow G \odot \left\{ \frac{D^T \left[(DG)^{\cdot -2} \odot X\right]}{D^T (DG)^{\cdot -1} + \frac{\lambda}{\epsilon + G}} \right\}^{\cdot \frac{1}{2}} \tag{4}$$

Besides being easy to implement, these update rules generate non-increasing objective values and converge theoretically [18].

A speech dictionary $D^s \in \mathcal{R}_+^{64\times M^s}$ is trained by substituting S for X in (2), while a noise dictionary $D^v \in \mathcal{R}_+^{64\times M^v}$ is learned by replacing X with V. Only one *universal* speech dictionary is trained, while one noise dictionary is trained for *each* noise type. The speech dictionary is expected to work well for *all* utterances regardless of the gender, dialects, and content. Therefore, we set $M^s > 64$ with $\lambda > 0$ in order to ensure the speech dictionary is comprehensive enough. As for noise dictionary, we use $M^v < 64$ with $\lambda = 0$ because D^v is responsible for characterizing only *one* noise type.

2.3. Sparse Coding

By assuming a linear model, i.e., $Y \approx S + V$, S and V can be estimated from Y via sparse coding:

$$\min_{G^s, G^v} d_{\mathrm{IS}}\left(Y \mid D^s G^s + D^v G^v\right) + \beta \sum_{m,n} \log\left(\epsilon + G^s_{m,n}\right) \tag{5}$$

where D^s and D^v are the previously learned dictionaries and ϵ is a small positive constant. Problem (5) can be efficiently solved by a similar update as in (4). By the ways that D^s and D^v are trained, we have $D^s G^s \approx S$ and $D^v G^v \approx V$. If both approximations are accurate, then the IBM can be reliably estimated by a simple thresholding:

$$\hat{B}_{k,n} = \begin{cases} 1, & \text{if } \frac{[D^s G_n^s]_k}{[D^v G_n^v]_n} \geq LC \\ 0, & \text{otherwise} \end{cases}, \quad (6)$$

which can be viewed as applying a simple linear classifier on the sparse code $G_n := [G_n^s; G_n^v] \in \mathcal{R}_+^{(M^s+M^v)\times 1}$ for each channel k. However, if the approximation is inaccurate, then more advanced classifiers such as SVM or DNN can improve the performance. To include more *temporal information*, we concatenate two adjacent frames to construct the final feature:

$$\tilde{G} := \left\{ \tilde{G}_n \in \mathcal{R}_+^{3(M^s+M^v)\times 1} \mid [G_{n-1}; G_n; G_{n+1}], \forall\, n \right\} \quad (7)$$

2.4. Deep Neural Network

We train a DNN for classification. For time frame n, it takes $\tilde{G}_n$ as the input and takes $[B_{1,n}; \cdots; B_{64,n}] \in \mathcal{R}_+^{64\times 1}$ as the label. The output of the network, the estimated IBM $[\hat{B}_{1,n}; \cdots; \hat{B}_{64,n}]$, is used to resynthesize the enhanced speech. In particular, we use a 5-layer Rectified Linear Unit (ReLU) [19] network, which uses a rectified activation function, $\max(0, z)$, as the non-linear layer. Compared with traditional sigmodial or hyperbolic tangent non-linearity, ReLU achieves a better performance without any unsupervised pretraining [20]. Training of the network is performed by minimizing the cross entropy loss over the training set with pure back propagation algorithm.

2.5. Rationale of the System Design

In our design, we perform IBM estimation in the cochleagram domain instead of the conventional spectrogram domain. There are two main reasons behind our choice of the working domain. First, the cochleagram only requires 64 channels to provide a fine enough frequency resolution for a 16 kHz speech, while the conventional spectrogram usually requires at least 256 channels. The channel reduction comes from the non-uniform frequency spacing in the cochleagram which mimics the human perception. From a classification point of view, estimating a mask with dimension 64 is easier and thus more desirable than a mask with dimension 256. Second, despite the smaller dimensionality, masking on the cochleagram domain still results in high quality enhanced speech [10], provided the mask is accurately estimated.

The SNMF is chosen for feature extraction, mainly due to its simplicity and effectiveness. Existing works use sophisticated features to find an accurate mask. The design and extraction of these features heavily rely on extensive domain knowledge and human fine tuning [1, 9–11]. In contrast, our SNMF based approach is conceptually much simple. However, despite the simplicity, the SNMF is capable of extracting the desired speech and noise information, even when the noise is non-stationary [3,6]. This desirable property is the basis for achieving an accurate IBM estimation.

The DNN is used for classification, mainly due to its huge success in various complicated machine learning tasks [12,13]. Certainly, other classifiers such as SVM can also be used, but experimentally we have found DNN to achieve the best result.

3. PERFORMANCE EVALUATION

3.1. Experiment Setting

The TIMIT data set [21] was used as the speech corpus. Three types of noises from NOISEX-92 [22], namely, street, factory, babble, are taken as noise sources. Audio files are resampled at 16 kHz. Noisy mixture is obtained by mixing a sentence with one type of noise at -5 dB. In the training stage, a universal speech dictionary is trained using 500 sentences from the "train" subset of TIMIT. For each noise type v, a separate noise dictionary D^v is trained using a randomly selected 30-second noise segment. Further, a DNN is trained using 4000 noisy sentences obtained by mixing utterances from the "train" subset and randomly selected noise segments of type v. In the testing stage, the "test" subset is combined with each of the three noise sources for performance evaluation.

To quantify the classification performance, we use the HIT-FA criterion, which has been shown to correlate well to human speech intelligibility [1]. Here HIT represents the percentage of correctly predicted speech-dominant T-F units while FA is the percentage of wrongly predicted noise-dominant T-F units. To assess the performance of the enhanced speech, we use the Perceptual Evaluation of Speech Quality score (PESQ) [23] for evaluating the quality and the Short-Time Objective Intelligibility measure (STOI) [24] for intelligibility. PESQ ranges from 0.5 to 4.5, while STOI takes its value between -1 and 1. Both measures are known to correlate well to human perception. The higher the value, the better the performance.

To evaluate the performance of DNN in our proposed approach, we compare two alternative classifiers: the simple thresholding (6) and the linear SVM (LSVM), assuming that all classifiers use the same set of features extracted from SNMF. Though kernel SVM generally achieves a better performance than LSVM, the high complexity of kernel SVM makes it prohibitive in our setup, where both the feature dimensionality and the number of samples are big. In SNMF + LSVM, while $\tilde{G}_n$ (7) remains as the feature, we train 64 LSVMs as the classifiers for the 64 channels [25]. Moreover, two other classification-based systems proposed in Kim *et al.* [1] and Chen *et al.* [11] are also compared. For Kim's system, we use the values reported in [9], and use the results reported in [11] (Table 1 and 2) for Chen's system. Both systems are compared because they also consider a matched-noise condition and use similar front-end and back-end structures.[2] Therefore, we can focus on the influence of feature extraction and classification. Further, to compare the quality and intelligibility of the enhanced speech, we implement a commonly-used statistical-based algorithm by using the MMSE algorithm [26] for noise tracking and the Log Short-Time Spectrum Amplitude (LSTSA) estimator [2] for gain calculation. For notational convenience, this algorithm is referred to as LSTSA.

3.2. Parameter Selection of the Proposed System

3.2.1. Parameters for Dictionary Training and Sparse Coding

The speech dictionary is trained by solving (2), where X is replaced by the clean speech cochleagram S. The dictionary size M^s is set to 512, λ is set to 0.01, and ϵ is set to 10^{-5}. We perform 500 iterations of the multiplicative updates (3) and (4), which takes about 1 hour when using a MATLAB implementation on a cluster computer.[3]

For each noise type v, the corresponding noise dictionary D^v is trained by solving problem (2), where X is replaced by the noise cochleagram. For different noise dictionaries, their sizes M^v are

[2] Chen's system uses a 32-channel Gammatone filterbank in the analysis front-end, and sets the LC value to -10 dB. [9] modifies Kim's system to use exactly the same front-end and back-end as ours.

[3] A Linux-based system with 8 Intel Sandy bridge E5-2670 processors (2.6 GHz) and 64 GB memory.

chosen according to the "complexity" of the corresponding noise sources. For example we use the largest number of atoms for the babble noise dictionary, as it is clearly the most complex one among all three types of noises. Other parameters for the noise dictionary training as well as the sparse coding are given in Table 1.

In (5), the regularization parameter β in each noise type is determined by first solving (5) for a large number of potential candidates[4] of β, and then picking the one with the best separation performance (or the smallest $d_{IS}(S \mid D^s G^s) + d_{IS}(V \mid D^v G^v)$). Interestingly, the resulting β makes intuitive sense: when M^v is large, the corresponding noise dictionary D^v is more likely to represent the speech. As a result smaller β should be used so that the speech dictionary can represent the speech source as much as possible.

Table 1. Parameters of noise dictionaries and sparse coding stage

	Street	Factory	Babble
M^v in (2)	5	15	25
ϵ in (2)	10^{-5}	10^{-5}	10^{-5}
β in (5)	0.15	0.04	0.02
ϵ in (5)	10^{-5}	10^{-5}	10^{-5}

3.2.2. Parameters for DNN

We use a 5 layer ReLU network [19] with $3 \times M$ neurons in the input layer, 1024 neurons in the 3 hidden layers, and 64 neurons in the output layer. Here, M denotes the total number of dictionary atoms. For example, the network used in babble noise has $3 \times (512 + 25)$ neurons in the input layer. The drop out probability [19] in the input layer is set to 0.2, and it is changed to 0.5 in the hidden and output layer. To avoid form diverging, we constraint the L_∞ norm of the network weight to be less than 0.1. 500 epoches is used for supervised training, which takes 30 hours to run on a cluster computer. [3]

3.3. The Results

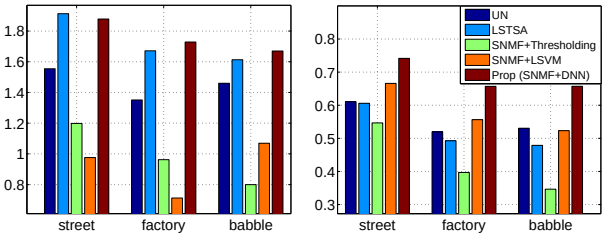
In Table 2, we present the classification performance under different noise conditions at -5 dB SNR. The "x" means there is no available data. The differences between these systems are the feature extraction step and/or the classification step. The front-end and back-end are the same. The first three rows of Table 2 show a monotonic performance improvement when the stronger classifier is used. When the SNR is as low as -5 dB, SNMF itself is unable to produce a reliable decomposition, and therefore SNMF+Thresholding cannot deliver satisfactory result. When a classifier is used, we see a better performance by using DNN than using LSVM. Also, it is clear that the proposed system outperforms the system proposed by Chen *et al.* and Kim *et al*. Our results also suggest that when the DNN is used as the classifier, a simple SNMF based feature extraction is sufficient for classification. Note that [10] also uses DNN for classification, but only the 0 dB scenario was considered, where most normal-hearing listeners have near perfect recognition rate.

Fig. 2 evaluates the quality and the intelligibility of the enhanced speech by different approaches. Audio samples are available at [27]. In terms of both PESQ and STOI, the proposed system outperforms the thresholding and the LSVM counterparts. This is consistent with the results in Table 2. Compared to LSTSA, the proposed approach achieves a comparable speech quality to LSTSA. This is very encouraging, as it shows that a classification-based system is capable of achieving high speech quality, although its primary target is to improve the speech intelligibility. Further, we observe from Fig. 2(a) that the advantage of the proposed approach over LSTSA is more pronounced when the noise becomes increasingly more non-stationary (from "street" to "babble"). We attribute this to the use of SNMF in the feature extraction step, as this technique is known to work well when the noise source contains distinct features, and is robust to the non-stationarity [3–6]. Fig. 2(b) compares the speech intelligibility of the enhanced speech. As expected, LSTSA is not able to improve the speech intelligibility, while the proposed system has a relatively significant improvement. This is consistent with what is known in the literature, that traditional enhancement methods cannot improve speech intelligibility [8], while a properly trained classification-based algorithm can [1, 9–11].

[4] $\beta \in [0.001, 0.01, 0.02, 0.04, 0.06, 0.08, 0.1, 0.12, 0.15, 0.2, 0.3, 0.5]$

Table 2. Classification results for different systems at -5 dB SNR under 3 noise types. Bold faced letters denote the best result.

		Street	Factory	Babble
Proposed (SNMF+DNN)	HIT	**86.5%**	**78.1%**	**75.8%**
	FA	**12.9%**	**9.0%**	**13.9%**
	HIT-FA	**73.6%**	**69.1%**	**61.9%**
SNMF+LSVM	HIT	73.9%	65.7%	58.8%
	FA	**12.9%**	11.4%	18.7%
	HIT-FA	61.0%	54.3%	40.1%
SNMF+Thresholding	HIT	59.4%	35.3%	43.2%
	FA	25.0%	17.5%	31.7%
	HIT-FA	34.4%	17.8%	11.5%
Chen *et al.* [11]	HIT	x	70%	62%
	FA	x	7%	13%
	HIT-FA	x	63%	49%
Kim *et al.* [1] (from Table 1 of [9])	HIT	x	57.3%	53.8%
	FA	x	26.7%	27.1%
	HIT-FA	x	30.6%	26.6%



(a) PESQ (b) STOI

Fig. 2. Performance comparison of the enhanced speech at -5 dB SNR under street, factory, and babble noises. "UN" denotes the unprocessed noisy mixture. "LSTSA" denotes using LSTSA [2] with MMSE noise variance tracking [26]. SNMF+Thresholding and SNMF+LSVM denotes the system that uses the same structure as the proposed system, but changes the classifiers from DNN to thresholding and LSVM, respectively. All values are averaged over 300 randomly selected sentences from the "test" data set.

4. CONCLUSIONS AND FUTURE WORKS

In this paper, we present a novel method for classification-based speech enhancement, which combines SNMF for feature extraction and DNN for classification. The proposed system uses only limited speech-specific knowledge, and is able to obtain a superior performance in terms of both quality and intelligibility under three different types of non-stationary noise sources at -5 dB SNR. In the current work, we only consider a matched-noise condition, in which both the training and testing stages use the same noise type under the same SNR. In future, we plan extending this work to unmatched-noise conditions. It is also interesting to consider joint optimizing both the NMF for feature extraction and the DNN for classification.

5. REFERENCES

[1] Gibak Kim, Yang Lu, Yi Hu, and Philipos C. Loizou, "An algorithm that improves speech intelligibility in noise for normal-hearing listeners," *The Journal of the Acoustical Society of America*, vol. 126, no. 3, 2009.

[2] Yariv Ephraim and David Malah, "Speech enhancement using a minimum mean-square error log-spectral amplitude estimator," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 33, no. 2, pp. 443 – 445, Apr. 1985.

[3] Mikkel N. Schmidt, Jan Larsen, and Fu-Tien Hsiao, "Wind noise reduction using non-negative sparse coding," in *IEEE Workshop on Machine Learning for Signal Processing*, Aug. 2007, pp. 431 –436.

[4] Gautham J. Mysore and Paris Smaragdis, "A non-negative approach to semi-supervised separation of speech from noise with the use of temporal dynamics," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2011, pp. 17–20.

[5] Nasser Mohammadiha, Paris Smaragdis, and Arne Leijon, "Supervised and unsupervised speech enhancement using nonnegative matrix factorization," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 10, pp. 2140–2151, Oct 2013.

[6] Hung-Wei Tseng, Srikanth Vishnubhotla, Mingyi Hong, Xiangfeng Wang, Jinjun Xiao, Zhi-Quan Luo, and Tao Zhang, "A single channel speech enhancement approach by combining statistical criterion and multi-frame sparse dictionary learning," in *Proc. Interspeech*, 2013.

[7] Douglas S. Brungart, Peter S. Chang, and DeLiang Wang Simpson, Brian D. Simpson, "Isolating the energetic component of speech-on-speech masking with ideal time-frequency segregation," *The Journal of the Acoustical Society of America*, vol. 120, no. 6, pp. 4007–4018, doi = http://dx.doi.org/10.1121/1.2363929, 2006.

[8] Yi Hu and Philipos C. Loizou, "A comparative intelligibility study of single-microphone noise reduction algorithms," *The Journal of the Acoustical Society of America*, vol. 122, no. 3, 2007.

[9] Kun Han and DeLiang Wang, "An svm based classification approach to speech separation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2011, pp. 4632–4635.

[10] Yuxuan Wang and DeLiang Wang, "Towards scaling up classification-based speech separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 7, pp. 1381–1390, July 2013.

[11] Jitong Chen, Yuxuan Wang, and DeLiang Wang, "A feature study for classification-based speech separation at very low signal-to-noise ratio," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014.

[12] Geoffrey Hinton., Li Deng, Dong Yu, George E. Dahl, Abdel rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N. Sainath, and Brian Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, Nov 2012.

[13] Yoshua Bengio, Aaron Courville, and Pascal Vincent, "Representation learning: A review and new perspectives," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 8, pp. 1798–1828, Aug 2013.

[14] DeLiang Wang and Guy J. Brown, *Computational Auditory Scene Analysis: Principles, Algorithms, and Applications*, Wiley-IEEE Press, 2006.

[15] Daniel D. Lee and H. Sebastian Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[16] Cédric Févotte and Jérôme Idier, "Algorithms for nonnegative matrix factorization with the β-divergence," *Neural Computation*, vol. 23, no. 9, pp. 2421–2456, 2011.

[17] Emmanuel J. Candés, Michael B. Wakin, and Stephen P. Boyd Boyd, "Enhancing sparsity by reweighted l1 minimization," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5-6, pp. 877–905, 2008.

[18] Meisam Razaviyayn, Mingyi Hong, and Zhi-Quan Luo, "A unified convergence analysis of block successive minimization methods for nonsmooth optimization," *SIAM Journal on Optimization*, vol. 23, no. 2, pp. 1126–1153, 2013.

[19] George E. Dahl, Tara N. Sainath, and Geoffrey E. Hinton, "Improving deep neural networks for lvcsr using rectified linear units and dropout," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, May 2013, pp. 8609–8613.

[20] Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proceedings of the ICML*, 2013.

[21] John Garofolo, Lori Lamel, William Fisher, Jonathan Fiscus, David Pallett, Nancy Dahlgren, and Victor Zue, "DARPA TIMIT acoustic phonetic continuous speech corpus," 1993.

[22] Andrew Varga and Herman J.M. Steeneken, "Assessment for automatic speech recognition: Ii. noisex-92: A database and an experiment to study the effect of additive noise on speech recognition systems," *Speech Communication*, vol. 12, no. 3, pp. 247 – 251, 1993.

[23] Antony W. Rix, John G. Beerends, Michael P. Hollier, and Andries P. Hekstra, "Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs," in *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2001. Proceedings.*, 2001, vol. 2, pp. 749 –752 vol.2.

[24] Cees H. Taal, Richard C. Hendriks, Richard Heusdens, and Jesper Jensen, "An algorithm for intelligibility prediction of time-frequency weighted noisy speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 2125–2136, Sept 2011.

[25] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin, "LIBLINEAR: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[26] Timo Gerkmann and Richard C. Hendriks, "Unbiased mmse-based noise power estimation with low complexity and low tracking delay," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 4, pp. 1383–1393, 2012.

[27] Hung-Wei Tseng, "Icassp'2015 companion website," `http://ohandyya.wix.com/bioinfo#!nmf-dnn-ibm/cqse`, 2014.