Deep Clustering and Conventional Networks for Music Separation: Strong Together

Luo, Y.; Chen, Z.; Hershey, J.R.; Le Roux, J.; Mesgarani, N.

Abstract

Deep clustering is the first method to handle general audio separation scenarios with multiple sources of the same type and an arbitrary number of sources, performing impressively in speaker-independent speech separation tasks. However, little is known about its effectiveness in other challenging situations such as music source separation. Contrary to conventional networks that directly estimate the source signals, deep clustering generates an embedding for each time-frequency bin, and separates sources by clustering the bins in the embedding space. We show that deep clustering outperforms conventional networks on a singing voice separation task, in both matched and mismatched conditions, even though conventional networks have the advantage of end-to-end training for best signal approximation, presumably because its more flexible objective engenders better regularization. Since the strengths of deep clustering and conventional network architectures appear complementary, we explore combining them in a single hybrid network trained via an approach akin to multi-task learning. Remarkably, the combination significantly outperforms either of its components.

DEEP CLUSTERING AND CONVENTIONAL NETWORKS FOR MUSIC SEPARATION: STRONGER TOGETHER

Yi Luo[†]   Zhuo Chen[†]   John R. Hershey[‡]   Jonathan Le Roux[‡]   Nima Mesgarani[†]

[†] Department of Electrical Engineering, Columbia University, New York, NY
[‡]Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA

ABSTRACT

Deep clustering is the first method to handle general audio separation scenarios with multiple sources of the same type and an arbitrary number of sources, performing impressively in speaker-independent speech separation tasks. However, little is known about its effectiveness in other challenging situations such as music source separation. Contrary to conventional networks that directly estimate the source signals, deep clustering generates an embedding for each time-frequency bin, and separates sources by clustering the bins in the embedding space. We show that deep clustering outperforms conventional networks on a singing voice separation task, in both matched and mismatched conditions, even though conventional networks have the advantage of end-to-end training for best signal approximation, presumably because its more flexible objective engenders better regularization. Since the strengths of deep clustering and conventional network architectures appear complementary, we explore combining them in a single hybrid network trained via an approach akin to multi-task learning. Remarkably, the combination significantly outperforms either of its components.

Index Terms— Deep clustering, Singing voice separation, Music separation, Deep learning

1. INTRODUCTION

Monaural music source separation has been the focus of many research efforts for over a decade. This task aims at separating a music recording into several tracks where each track corresponds to a single instrument. A related goal is to design algorithms that can separate vocals and accompaniment, where all the instruments are considered as one source. Music source separation algorithms have been successfully used for predominant pitch tracking [1], accompaniment generation for Karaoke systems [2], or singer identification [3].

Despite these advances, a system that can successfully generalize to different music datasets has thus far remained unachievable, due to the tremendous variability of music recordings, for example in terms of genre or types of instruments used. Unsupervised methods, such as those based on computational auditory scene analysis (CASA) [4], source/filter modeling [5], or low-rank and sparse modeling [6], have difficulty in capturing the dynamics of the vocals and instruments, while supervised methods, such as those based on nonnegative matrix factorization (NMF) [7], F0-based estimation [8], or Bayesian modeling [9], suffer from generalization and processing speed issues.

Recently, deep learning has found many successful applications in audio source separation. Conventional regression-based networks try to infer the source signals directly, often by inferring time-frequency (T-F) masks to be applied to the T-F representation of the

mixture so as to recover the original sources. These mask-inference networks have been shown to produce superior results compared to the traditional approaches in singing voice separation [10]. These networks are a natural choice when the sources can be characterized as belonging to distinct classes.

Another promising approach designed for more general situations is the so-called deep clustering framework [11]. Deep clustering has been applied very successfully to the task of single-channel speaker-independent speech separation [11]. Because it uses of pairwise affinities as separation criterion, deep clustering can handle mixtures with multiple sources from the same type, and an arbitrary number of sources. Such difficult conditions are endemic to music separation.

In this study, we explore the use of both deep clustering and conventional mask-inference networks to separate the singing voice from the accompaniment, grouping all the instruments as one source and the vocals as another. The singing voice separation task that we consider here is amenable to class based separation, and would not seem to require the extra flexibility in terms of source types and number of sources that deep clustering would provide. However, in addition to opening up the potential to apply to more general settings, the additional flexibility of deep clustering may have some benefits in terms of regularization. Whereas conventional mask-inference approaches only focus on increasing the separation between sources, the deep clustering objective also reduces within-source variance in the internal representation, which could be beneficial for generalization. In recent work it has been shown that forcing deep network activations to cluster well can improve the resulting test performance [12]. To investigate these potential benefits, we develop a two-headed "Chimera" network with both a deep clustering head and a mask-inference head attached to the same network body. Each head has its own objective, but the whole hybrid network is trained in a joint fashion akin to multi-task training. Our findings show that the addition of the deep clustering criterion greatly improves upon the performance of the mask-inference network.

2. MODEL DESCRIPTION

2.1. Deep clustering

Deep clustering operates according to the assumption that the T-F representation of the mixed signal can be partitioned into multiple sets, depending on which source is dominant (i.e., its power is the largest among all sources) in a given bin. A deep clustering network takes features of the acoustic signal as input, and assigns a D-dimensional embedding to each T-F bin. The network is trained to encourage the embeddings of T-F bins dominated by the same source to be similar to each other, and the embeddings of T-F bins

dominated by different sources to be different. Note that the concept of "source" shall be defined according to the task at hand: for example, one speaker per source for speaker separation, all vocals in one source versus all other instruments in another source for singing voice separation, etc. A T-F mask for separating each source can then be estimated by clustering the T-F embeddings [13].

The training target is derived from a label indicator matrix $\mathbf{Y} \in \mathbb{R}^{TF \times C}$, where T denotes the number of frames, F the feature dimension, and C the number of sources in the input mixture $\mathbf{x}$, such that $Y_{i,j} = 1$ if T-F bin $i = (t, f)$ is dominated by source j, and $Y_{i,j} = 0$ otherwise. We can construct a binary affinity matrix $\mathbf{A} = \mathbf{Y}\mathbf{Y}^T$, which represents the assignment of the sources in a permutation independent way: $A_{i,j} = 1$ if i and j are dominated by the same source, and $A_{i,j} = 0$ if they are not. The network estimates an embedding matrix $\mathbf{V} \in \mathbb{R}^{TF \times D}$, where D is the embedding dimension. The corresponding estimated affinity matrix is then defined as $\hat{\mathbf{A}} = \mathbf{V}\mathbf{V}^T$. The cost function for the network is

$$\mathcal{L}_{\mathrm{DC}} = ||\hat{\mathbf{A}} - \mathbf{A}||_F^2 = ||\mathbf{V}\mathbf{V}^T - \mathbf{Y}\mathbf{Y}^T||_F^2. \qquad (1)$$

Although the matrices $\mathbf{A}$ and $\hat{\mathbf{A}}$ are typically very large, their low-rank structure can be exploited to decrease the computational complexity [11].

At test time, a clustering algorithm such as K-means is applied to the embeddings V to generate a cluster assignment matrix, which is used as a binary T-F mask applied to the mixture to estimate the T-F representation of each source.

2.2. Multi-task learning and Chimera networks

Whereas the deep clustering objective function has been shown to enable the training of neural networks for challenging source separation problems, a disadvantage of deep clustering is that the post-clustering process needed to generate the mask and recover the sources is not part of the original objective function. On the other hand, for mask-inference networks, the objective function minimized during training is directly related to the signal recovery quality. We seek to combine the benefits of both approaches in a strategy reminiscent of multi-task learning, except that here both approaches address the same separation task.

In [11] and [13], the typical structure of a deep clustering network is to have multiple stacked recurrent layers (e.g., BLSTMs) yielding an N-dimensional vector at the top layer, followed by a fully-connected linear layer. For each frame t, this layer outputs a D-dimensional vector for each of the F frequencies, resulting in a $F \times D$ representation $\mathbf{Z}_t$. To form the embeddings, $\mathbf{Z}$ then passes through a $\mathtt{tanh}$ non-linearity, and unit-length normalization independently for each T-F bin. Concatenating across time results in the $TF \times D$ embedding matrix $\mathbf{V}$ as used in Eq. 1.

We extend this architecture in order to create a two-headed network, which we refer as "Chimera" network, with one head outputting embeddings as in a deep clustering network, and the other head outputting a soft mask, as in a mask-inference network. The new mask-inference head is obtained starting with $\mathbf{Z}$, and passing it through F fully-connected $D \times C$ mask estimation layers (e.g., softmax), one for each frequency, resulting in C masks $\mathbf{M}^{(c)}$, one for each source. The structure of the Chimera network is illustrated in Figure 1.

The body of the network, up to the layer outputting $\mathbf{Z}$, can be trained with each head separately. For the deep clustering head, we use the objective $\mathcal{L}_{\mathrm{DC}}$. For the mask-inference head, we can use a classical magnitude spectrum approximation (MSA) objective [6,
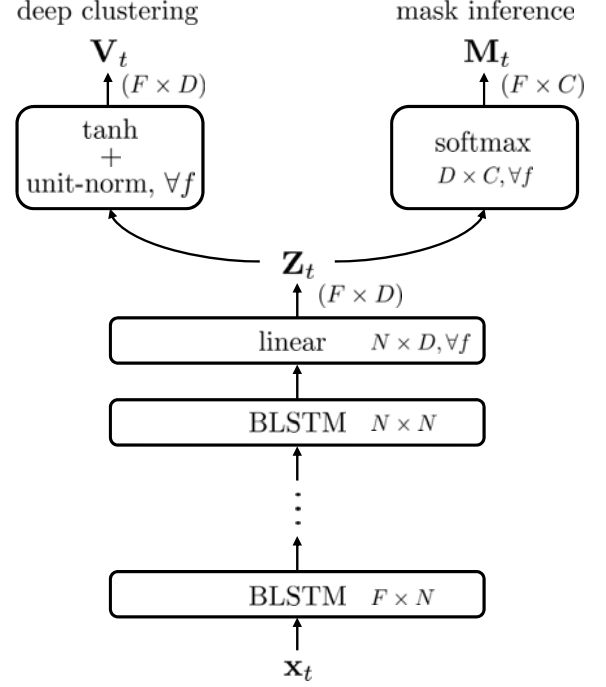

Fig. 1. Structure of the Chimera network.

14, 15], defined as:

$$\mathcal{L}_{\mathrm{MSA}} = \sum_c ||\mathbf{R}^{(c)} - \mathbf{M}^{(c)} \odot \mathbf{S}||_2^2, \qquad (2)$$

where $\mathbf{R}^{(c)}$ denotes the magnitude of the T-F representation for the c-th clean source and $\mathbf{S}$ that of the mixture. Although this objective function makes sense intuitively, one caveat is that the mixture magnitude $\mathbf{S}$ may be smaller than that of a given source $\mathbf{R}^{(c)}$ due to destructive interference. In this case, $\mathbf{M}^{(c)}$, which is between 0 and 1, cannot bring the estimate close to $\mathbf{R}^{(c)}$. As a remedy, we consider an alternative objective, denoted as masked magnitude spectrum approximation (mMSA), which approximates $\mathbf{R}^{(c)}$ as the output of a masking operation on the mixture using a reference mask $\mathbf{O}^{(c)}$, such that $\mathbf{O}^{(c)} \odot \mathbf{S} \approx \mathbf{R}^{(c)}$, for source c:

$$\mathcal{L}_{\mathrm{mMSA}} = \sum_c ||(\mathbf{O}^{(c)} - \mathbf{M}^{(c)}) \odot \mathbf{S}||_2^2. \qquad (3)$$

Note that this is equivalent to a weighted mask approximation objective, using the mixture magnitude as the weights.

We can also define a global objective for the whole network as

$$\mathcal{L}_{\mathrm{CHI}} = \alpha \frac{\mathcal{L}_{\mathrm{DC}}}{TF} + (1 - \alpha)\mathcal{L}_{\mathrm{MI}} \qquad (4)$$

where $\alpha \in [0, 1]$ controls the importance between the two objectives, and the objective $\mathcal{L}_{\mathrm{MI}}$ for the mask inference head is either $\mathcal{L}_{\mathrm{MSA}}$ or $\mathcal{L}_{\mathrm{mMSA}}$. Note that here we divide $\mathcal{L}_{\mathrm{DC}}$ by TF because the objective for deep clustering calculates the pair-wise loss for each of the $(TF)^2$ pairs of T-F bins, while the spectrum approximation objective calculates end-to-end loss on the TF time-frequency bins. For $\alpha = 1$, only the deep clustering head gets trained together with the body, resulting in a deep clustering network. For $\alpha = 0$, only the mask-inference head gets trained together with the body, resulting in a mask-inference network.

At test time, if both heads have been trained, either can be used. The mask-inference head directly outputs the T-F masks, while the deep clustering head outputs embeddings on which we perform clustering using, e.g., K-means.

3. EVALUATION AND DISCUSSION

3.1. Datasets

For training and evaluation purposes, we built a remixed version of the DSD100 dataset for SiSEC [16], which we refer to as DSD100-remix. For evaluation only, we also report results on two other datasets: the hidden iKala dataset for the MIREX submission, and the public iKala dataset for our newly proposed models.

The DSD100 dataset includes synthesized mixtures and the corresponding original sources from 100 professionally produced and mixed songs. To build the training and validation sets of DSD100-remix, we use the DSD100 development set (50 songs). We design a simple energy-level-based detector [17] to remove silent parts in both the vocal and accompaniment tracks, so that the vocals and accompaniment fully overlap in the generated mixtures. After that, we downsample the tracks from 44.1 kHz to 16 kHz to reduce computational cost, and then randomly mix the vocals and accompaniment together at 0 dB SNR, creating a 15 h training set and a 0.5 h validation set. We build the evaluation set of DSD100-remix from the DSD100 test set using a similar procedure, generating 50 pieces (one for each song) of fully-overlapped recordings with 30 seconds length each.

The input feature we use is calculated by the short-time Fourier transform (STFT) with 512-point window size and 128-point hop size. We use a 150-dimension mel-filterbank to reduce the input feature dimension. First-order delta of the mel-filterbank spectrogram is concatenated into the input feature. We used the ideal binary mask calculated on the mel-filterbank spectrogram as the target Y matrix.

3.2. System architecture

The Chimera network's body is comprised of 4 bi-directional long-short term memory (BLSTM) layers with 500 hidden units in each layer, followed by a linear fully-connected layer with a D-dimension vector output for each of the frame's $F = 150$ T-F bins. Here, we use $D = 20$ because it produced the best performance in a speech separation task [11]. In the mask-inference head, we set $C = 2$ for the singing voice separation task, and use softmax as the nonlinearity. We use the rmsprop algorithm [18] as optimizer and select the network with the lowest loss on the validation set.

At test time, we split the signal into fixed-length segments, on which we run the network independently. We also tried running the network on the full input feature sequence, as in [11], but this lead to worse performance, probably due to the mismatch in context size between the training and test time. The mask-inference head of the network directly generates T-F masks. For deep clustering, the masks are obtained by applying K-means on the embeddings for the whole signal. We apply the mask for each source to the mel-filterbank spectrogram of the input, and recover the source using an inverse mel-filterbank transform and inverse-STFT with the mixture phase, followed by upsampling.

3.3. Results for the MIREX submission

We first report on the system submitted to the Singing Voice Separation task of the Music Information Retrieval Evaluation eXchange

Table 1. Evaluation metrics for different systems in MIREX 2014-2016 on the hidden iKala dataset. V denotes vocals and M music.

	GNSDR		GSIR		GSAR	
	V	M	V	M	V	M
DC	6.3	11.2	14.5	25.2	10.1	7.3
MC2 [19]	5.3	9.7	10.5	19.8	11.2	6.1
MC3 [19]	5.5	9.8	10.8	19.6	11.2	6.3
FJ1 [1]	6.8	10.1	13.3	11.2	11.5	10.0
FJ2 [1]	6.3	9.9	13.7	11.7	10.6	9.1
IIY1 [23]	4.2	7.8	15.5	12.4	7.7	5.4
IIY2 [23]	4.5	7.9	13.3	14.3	8.6	5.0

(MIREX 2016) [19]. That system only contains the deep clustering part, which corresponds to $\alpha = 1$ in the hybrid system. In the MIREX system, dropout layers with probability 0.2 were added between each feed-forward connection, and sequence-wise batch normalization [20] was applied in the input-to-hidden transformation in each BLSTM layer. Similarly to [13], we also applied a curriculum learning strategy [21], where we first train the network on segments of 100 frames, then train on segments of 500 frames. As distinguishing between vocals and accompaniment was part of the task, we used a crude rule-based approach: the mask whose total number of nonzero entries in the low frequency range (< 200 Hz) is more than a half is used as the accompaniment mask, and the other as the vocals mask.

The hidden iKala dataset has been used as the evaluation dataset throughout MIREX 2014-2016, so we can report, as shown in Table 1, the results from the past three years, comparing the best two systems in each year's competition to our submitted system for 2016. The official MIREX results are reported in terms of global normalized SDR (GNSDR), global SIR (GSIR), global SAR (GSAR) [22].

Due to time limitations at the time of the MIREX submission, we submitted a system that we had trained using the DSD100-remix dataset described in Section 3.1. However, as mentioned in the MIREX description, the DSD100 dataset is different from both the hidden and public parts of the iKala dataset [22]. Nonetheless, our system not only won the 1st place in MIREX 2016 but also outperformed the best systems from past years, even without training on the better-matched public iKala dataset, showing the efficacy of deep clustering for robust music separation. Note that the hidden iKala dataset is unavailable to the public, and it is thus unfortunately impossible to evaluate here what the performance of our system would be when trained on the public iKala data.

3.4. Results for the proposed hybrid system

We now turn to the results using the Chimera networks. During the training phase, we use 100 frames of input features to form fixed duration segments. We train the Chimera network in three different regimes: a pure deep clustering regime (DC, $\alpha = 1$), a pure mask-inference regime (MI, $\alpha = 0$), and a hybrid regime (CHI$_\alpha$, $0 < \alpha < 1$). All networks are trained from random initialization, and no training tricks mentioned above for the MIREX system are added. We report results on the DSD100-remix test set, which is matched to the training data, and the public iKala dataset, which is not.

By design, deep clustering provides one output for each source, and the sequence of the separation result is random. Therefore, the scores are computed by using the best permutation between ref-

Table 2. SDRi (dB) on the DSD100-remix and the public iKala datasets. The suffix after CHI_α denotes which head of the Chimera network is used for generating the masks.

	DSD100-remix		iKala	
	V	M	V	M
DC	4.9	7.2	6.1	10.0
MI	4.8	6.7	5.2	8.9
$CHI_{0.1}$-DC	4.8	7.2	6.0	9.7
$CHI_{0.1}$-MI	**5.5**	**7.8**	**6.4**	**10.5**
$CHI_{0.5}$-DC	4.7	7.1	5.9	9.9
$CHI_{0.5}$-MI	**5.5**	**7.8**	6.3	**10.5**

Table 3. SDRi (dB) on the DSD100-remix and the public iKala datasets with various objectives in the MI head and embedding dimensions D.

$\mathcal{L}_{MI}$	D	DSD100-remix		iKala	
		V	M	V	M
MSA	20	**5.5**	7.8	6.4	10.5
mMSA	20	5.4	7.8	6.5	10.7
mMSA	10	**5.5**	**7.9**	**6.6**	**10.8**

erences and estimates at the file level. Table 2 shows the results with the MSA objective in the MI head. We compute the source-to-distortion ratio (SDR), defined as scale-invariant SNR [13], for each test example, and report the length-weighted average over each test set of the improvement of SDR in the estimate with respect to that in the mixture (SDRi).

As can be seen in the results, MI performs competitively with DC on DSD100-remix, however DC performs significantly better on the public iKala data. This shows the better generalization and robustness of the deep clustering method in cases where the test and training set are not matched. The best performance is achieved by CHI_α-MI, the MI head of the Chimera network. Interestingly, the performance of the DC head does not change significantly for the values of α tested. This suggests that joint training with the deep clustering objective allows the body of the network to learn a more powerful representation than using the mask-inference objective alone; this representation is then best exploited by the mask-inference head thanks to its signal approximation objective.

We now look at the influence of the objective used in the MI head. For the mMSA objective, we use the Wiener like mask [15] since it is shown to have best performance among oracle masks computed from source magnitudes. As shown in Table 3, training a hybrid $CHI_{0.1}$ network using the mMSA objective leads to slightly better MI performance overall compared to MSA. We also consider varying the embedding dimension D, and find that reducing it from $D = 20$ to $D = 10$ leads to further improvements. Because the output of the linear layer $\mathbf{Z}_t$ has dimension $F \times D$, decreasing D also leaves room to increase the number of frequency bins F.

Table 4 shows the results for various input features. We design various features by varying the sampling rate, the window/hop size in the STFT, and the dimension of the mel-frequency filterbanks. All networks are trained in the same hybrid regime as above, with the mMSA objective in the MI head and an embedding dimension $D = 10$. For simplicity, we do not concatenate first-order deltas into

Table 4. SDRi (dB) on the DSD100-remix and the public iKala datasets with various input features.

	DSD100-remix		iKala	
	V	M	V	M
16k-1024-256-mel150	5.5	7.9	6.6	10.6
16k-1024-256-mel200	5.5	7.9	6.9	10.9
22k-1024-256-mel200	5.9	7.9	7.2	10.7
22k-2048-512-mel300	**6.1**	**8.1**	**7.4**	**11.0**

the input feature. We can learn from the results that higher sampling rate, larger STFT window size STFT, and more mel-frequency bins result in better performance.
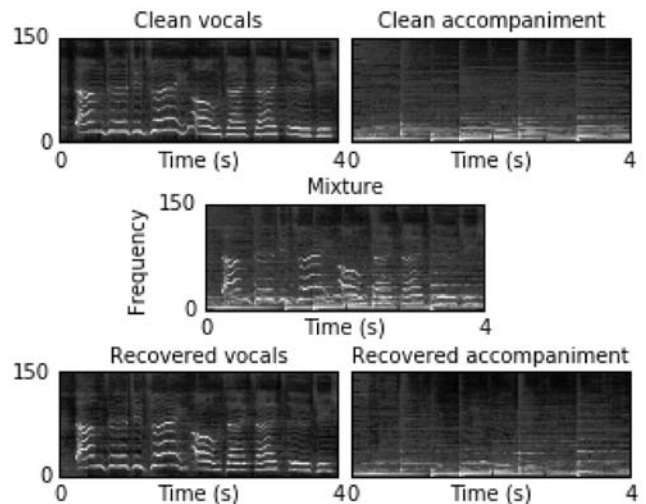

Fig. 2. Example of separation results for a 4-second excerpt from file 45378_chorus in the public iKala dataset.

4. CONCLUSION

In this paper, we investigated the effectiveness of a deep clustering model on the task of singing voice separation. Although deep clustering was originally designed for separating speech mixtures, we showed that this framework is also suitable for separating sources in music signals. Moreover, by jointly optimizing deep clustering with a classical mask-inference network, the new hybrid network outperformed both the plain deep clustering network and the mask-inference network. Experimental results confirmed the robustness of the hybrid approach in mismatched conditions.

Audio examples are available at [24].

5. ACKNOWLEDGEMENT

6. REFERENCES

[1] Zhe-Cheng Fan, Jyh-Shing Roger Jang, and Chung-Li Lu, "Singing voice separation and pitch extraction from monaural polyphonic audio music via DNN and adaptive pitch tracking," in *Proc. IEEE International Conference on Multimedia Big Data (BigMM)*. IEEE, 2016, pp. 178–185.

[2] Hideyuki Tachibana, Yu Mizuno, Nobutaka Ono, and Shigeki Sagayama, "A real-time audio-to-audio karaoke generation system for monaural recordings based on singing voice suppression and key conversion techniques," *Journal of Information Processing*, vol. 24, no. 3, pp. 470–482, 2016.

[3] Adam L. Berenzweig, Daniel P.W. Ellis, and Steve Lawrence, "Using voice segments to improve artist classification of music," in *Proc. AES 22nd International Conference: Virtual, Synthetic, and Entertainment Audio*. Audio Engineering Society, 2002.

[4] Yipeng Li and DeLiang Wang, "Separation of singing voice from music accompaniment for monaural recordings," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 4, pp. 1475–1487, 2007.

[5] Jean-Louis Durrieu, Gaël Richard, Bertrand David, and Cédric Févotte, "Source/filter model for unsupervised main melody extraction from polyphonic audio signals," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 3, pp. 564–575, 2010.

[6] Po-Sen Huang, Scott Deeann Chen, Paris Smaragdis, and Mark Hasegawa-Johnson, "Singing-voice separation from monaural recordings using robust principal component analysis," in *Proc. ICASSP*, 2012, pp. 57–60.

[7] Pablo Sprechmann, Alexander M Bronstein, and Guillermo Sapiro, "Real-time online singing voice separation from monaural recordings using robust low-rank modeling," in *IS-MIR*, 2012, pp. 67–72.

[8] Chao-Ling Hsu and Jyh-Shing Roger Jang, "On the improvement of singing voice separation for monaural recordings using the MIR-1K dataset," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 2, pp. 310–319, 2010.

[9] Po-Kai Yang, Chung-Chien Hsu, and Jen-Tzung Chien, "Bayesian singing-voice separation," in *Proc. ISMIR*, 2014, pp. 507–512.

[10] Po-Sen Huang, Minje Kim, Mark Hasegawa-Johnson, and Paris Smaragdis, "Singing-voice separation from monaural recordings using deep recurrent neural networks," in *Proc. IS-MIR*, 2014, pp. 477–482.

[11] John R. Hershey, Zhuo Chen, Jonathan Le Roux, and Shinji Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *Proc. ICASSP*, 2016, pp. 31–35.

[12] Renjie Liao, Alex Schwing, Richard Zemel, and Raquel Urtasun, "Learning deep parsimonious representations," in *Advances in Neural Information Processing Systems*, 2016, pp. 5076–5084.

[13] Yusuf Isik, Jonathan Le Roux, Zhuo Chen, Shinji Watanabe, and John R. Hershey, "Single-channel multi-speaker separation using deep clustering," in *Proc. Interspeech*, 2016.

[14] Felix Weninger, Jonathan Le Roux, John R. Hershey, and Björn Schuller, "Discriminatively trained recurrent neural networks for single-channel speech separation," in *Proc. IEEE GlobalSIP 2014 Symposium on Machine Learning Applications in Speech Processing*, Dec. 2014.

[15] Hakan Erdogan, John R. Hershey, Shinji Watanabe, and Jonathan Le Roux, "Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks," in *Proc. ICASSP*, 2015, pp. 708–712.

[16] "SiSEC 2016 Professionally produced music recordings task," https://sisec.inria.fr/home/2016-professionally-produced-music-recordings/, accessed: 2016-09-11.

[17] Javier Ramirez, Juan Manuel Górriz, and José Carlos Segura, *Voice activity detection. fundamentals and speech recognition system robustness*, INTECH Open Access Publisher, 2007.

[18] Tijmen Tieleman and Geoffrey Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural Networks for Machine Learning*, vol. 4, no. 2, 2012.

[19] "MIREX 2016: Singing Voice Separation Results," http://www.music-ir.org/mirex/wiki/2016:Singing_Voice_Separation_Results, accessed: 2016-09-11.

[20] César Laurent, Gabriel Pereyra, Philémon Brakel, Ying Zhang, and Yoshua Bengio, "Batch normalized recurrent neural networks," *arXiv preprint arXiv:1510.01378*, 2015.

[21] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston, "Curriculum learning," in *Proc. ICML*, 2009, pp. 41–48.

[22] "MIREX 2016: Singing Voice Separation Task," http://www.music-ir.org/mirex/wiki/2016:Singing_Voice_Separation, accessed: 2016-09-11.

[23] Yukara Ikemiya, Katsutoshi Itoyama, and Kazuyoshi Yoshii, "Singing voice separation and vocal f0 estimation based on mutual combination of robust principal component analysis and subharmonic summation," *arXiv preprint arXiv:1604.00192*, 2016.

[24] "Audio examples for chimera network," http://naplab.ee.columbia.edu/ivs.html.