SINGING-VOICE SEPARATION FROM MONAURAL RECORDINGS USING DEEP RECURRENT NEURAL NETWORKS

Po-Sen Huang[†], Minje Kim[‡], Mark Hasegawa-Johnson[†], Paris Smaragdis[†‡§]
[†]Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, USA
[‡]Department of Computer Science, University of Illinois at Urbana-Champaign, USA
[§]Adobe Research, USA
{huang146, minje, jhasegaw, paris}@illinois.edu

ABSTRACT

Monaural source separation is important for many real world applications. It is challenging since only single channel information is available. In this paper, we explore using deep recurrent neural networks for singing voice separation from monaural recordings in a supervised setting. Deep recurrent neural networks with different temporal connections are explored. We propose jointly optimizing the networks for multiple source signals by including the separation step as a nonlinear operation in the last layer. Different discriminative training objectives are further explored to enhance the source to interference ratio. Our proposed system achieves the state-of-the-art performance, 2.30∼2.48 dB GNSDR gain and 4.32∼5.42 dB GSIR gain compared to previous models, on the MIR-1K dataset.

1. INTRODUCTION

Monaural source separation is important for several real-world applications. For example, the accuracy of automatic speech recognition (ASR) can be improved by separating noise from speech signals [10]. The accuracy of chord recognition and pitch estimation can be improved by separating singing voice from music [7]. However, current state-of-the-art results are still far behind human capability. The problem of monaural source separation is even more challenging since only single channel information is available.

In this paper, we focus on singing voice separation from monaural recordings. Recently, several approaches have been proposed to utilize the assumption of the low rank and sparsity of the music and speech signals, respectively [7, 13, 16, 17]. However, this strong assumption may not always be true. For example, the drum sounds may lie in the sparse subspace instead of being low rank. In addition, all these models can be viewed as linear transformations in the spectral domain.

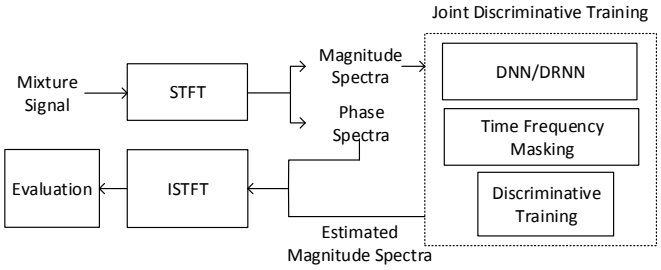


Figure 1. Proposed framework.

With the recent development of deep learning, without imposing additional constraints, we can further extend the model expressibility by using multiple nonlinear layers and learn the optimal hidden representations from data. In this paper, we explore the use of deep recurrent neural networks for singing voice separation from monaural recordings in a supervised setting. We explore different deep recurrent neural network architectures along with the joint optimization of the network and a soft masking function. Moreover, different training objectives are explored to optimize the networks. The proposed framework is shown in Figure 1.

The organization of this paper is as follows: Section 2 discusses the relation to previous work. Section 3 introduces the proposed methods, including the deep recurrent neural networks, joint optimization of deep learning models and a soft time-frequency masking function, and different training objectives. Section 4 presents the experimental setting and results using the MIR-1K dateset. We conclude the paper in Section 5.

2. RELATION TO PREVIOUS WORK

Several previous approaches utilize the constraints of low rank and sparsity of the music and speech signals, respectively, for singing voice separation tasks [7, 13, 16, 17]. Such strong assumption for the signals might not always be true. Furthermore, in the separation stage, these models can be viewed as a single-layer linear network, predicting the clean spectra via a linear transform. To further improve the expressibility of these linear models, in this paper, we use deep learning models to learn the representations from

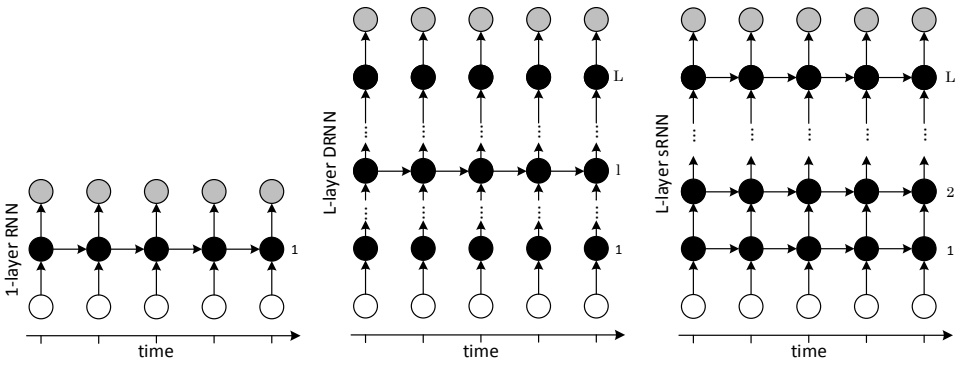


Figure 2. Deep Recurrent Neural Networks (DRNNs) architectures: Arrows represent connection matrices. Black, white, and grey circles represent input frames, hidden states, and output frames, respectively. (Left): standard recurrent neural networks; (Middle): L intermediate layer DRNN with recurrent connection at the l-th layer. (Right): L intermediate layer DRNN with recurrent connections at all levels (called stacked RNN).

data, without enforcing low rank and sparsity constraints.

By exploring deep architectures, deep learning approaches are able to discover the hidden structures and features at different levels of abstraction from data [5]. Deep learning methods have been applied to a variety of applications and yielded many state of the art results [2, 4, 8]. Recently, deep learning techniques have been applied to related tasks such as speech enhancement and ideal binary mask estimation [1, 9–11, 15].

In the ideal binary mask estimation task, Narayanan and Wang [11] and Wang and Wang [15] proposed a two-stage framework using deep neural networks. In the first stage, the authors use d neural networks to predict each output dimension separately, where d is the target feature dimension; in the second stage, a classifier (one layer perceptron or an SVM) is used for refining the prediction given the output from the first stage. However, the proposed framework is not scalable when the output dimension is high. For example, if we want to use spectra as targets, we would have 513 dimensions for a 1024-point FFT. It is less desirable to train such large number of neural networks. In addition, there are many redundancies between the neural networks in neighboring frequencies. In our approach, we propose a general framework that can jointly predict all feature dimensions at the same time using one neural network. Furthermore, since the outputs of the prediction are often smoothed out by time-frequency masking functions, we explore jointly training the masking function with the networks.

Maas et al. proposed using a deep RNN for robust automatic speech recognition tasks [10]. Given a noisy signal $\mathbf{x}$, the authors apply a DRNN to learn the clean speech $\mathbf{y}$. In the source separation scenario, we found that modeling one target source in the denoising framework is suboptimal compared to the framework that models all sources. In addition, we can use the information and constraints from different prediction outputs to further perform masking and discriminative training.

3. PROPOSED METHODS

3.1 Deep Recurrent Neural Networks

To capture the contextual information among audio signals, one way is to concatenate neighboring features together as input features to the deep neural network. However, the number of parameters increases rapidly according to the input dimension. Hence, the size of the concatenating window is limited. A recurrent neural network (RNN) can be considered as a DNN with indefinitely many layers, which introduce the memory from previous time steps. The potential weakness for RNNs is that RNNs lack hierarchical processing of the input at the current time step. To further provide the hierarchical information through multiple time scales, deep recurrent neural networks (DRNNs) are explored [3, 12]. DRNNs can be explored in different schemes as shown in Figure 2. The left of Figure 2 is a standard RNN, folded out in time. The middle of Figure 2 is an L intermediate layer DRNN with temporal connection at the l-th layer. The right of Figure 2 is an L intermediate layer DRNN with full temporal connections (called stacked RNN (sRNN) in [12]).

Formally, we can define different schemes of DRNNs as follows. Suppose there is an L intermediate layer DRNN with the recurrent connection at the l-th layer, the l-th hidden activation at time t is defined as:

$$\begin{aligned}\mathbf{h}_t^l &= f_h(\mathbf{x}_t, \mathbf{h}_{t-1}^l) \\ &= \phi_l\left(\mathbf{U}^l\mathbf{h}_{t-1}^l + \mathbf{W}^l\phi_{l-1}\left(\mathbf{W}^{l-1}\left(\dots\phi_1\left(\mathbf{W}^1\mathbf{x}_t\right)\right)\right)\right),\end{aligned} \tag{1}$$

and the output, $\mathbf{y}_t$, can be defined as:

$$\begin{aligned}\mathbf{y}_t &= f_o(\mathbf{h}_t^l) \\ &= \mathbf{W}^L\phi_{L-1}\left(\mathbf{W}^{L-1}\left(\dots\phi_l\left(\mathbf{W}^l\mathbf{h}_t^l\right)\right)\right),\end{aligned} \tag{2}$$

where $\mathbf{x}_t$ is the input to the network at time t, ϕ_l is an element-wise nonlinear function, $\mathbf{W}^l$ is the weight matrix

for the l-th layer, and $\mathbf{U}^l$ is the weight matrix for the recurrent connection at the l-th layer. The output layer is a linear layer.

The stacked RNNs have multiple levels of transition functions, defined as:

$$\begin{aligned}\mathbf{h}_t^l &= f_h(\mathbf{h}_t^{l-1}, \mathbf{h}_{t-1}^l) \\ &= \phi_l(\mathbf{U}^l \mathbf{h}_{t-1}^l + \mathbf{W}^l \mathbf{h}_t^{l-1}),\end{aligned} \tag{3}$$

where $\mathbf{h}_t^l$ is the hidden state of the l-th layer at time t. $\mathbf{U}^l$ and $\mathbf{W}^l$ are the weight matrices for the hidden activation at time $t-1$ and the lower level activation $\mathbf{h}_t^{l-1}$, respectively. When $l = 1$, the hidden activation is computed using $\mathbf{h}_t^0 = \mathbf{x}_t$.

Function $\phi_l(\cdot)$ is a nonlinear function, and we empirically found that using the rectified linear unit $f(\mathbf{x}) = \max(0, \mathbf{x})$ [2] performs better compared to using a sigmoid or tanh function. For a DNN, the temporal weight matrix $\mathbf{U}^l$ is a zero matrix.

3.2 Model Architecture

At time t, the training input, $\mathbf{x}_t$, of the network is the concatenation of features from a mixture within a window. We use magnitude spectra as features in this paper. The output targets, $\mathbf{y}_{\mathbf{1}_t}$ and $\mathbf{y}_{\mathbf{2}_t}$, and output predictions, $\hat{\mathbf{y}}_{\mathbf{1}_t}$ and $\hat{\mathbf{y}}_{\mathbf{2}_t}$, of the network are the magnitude spectra of different sources.

Since our goal is to separate one of the sources from a mixture, instead of learning one of the sources as the target, we adapt the framework from [9] to model all different sources simultaneously. Figure 3 shows an example of the architecture.

Moreover, we find it useful to further smooth the source separation results with a time-frequency masking technique, for example, binary time-frequency masking or soft time-frequency masking [7, 9]. The time-frequency masking function enforces the constraint that the sum of the prediction results is equal to the original mixture.

Given the input features, $\mathbf{x}_t$, from the mixture, we obtain the output predictions $\hat{\mathbf{y}}_{\mathbf{1}_t}$ and $\hat{\mathbf{y}}_{\mathbf{2}_t}$ through the network. The soft time-frequency mask $\mathbf{m}_t$ is defined as follows:

$$\mathbf{m}_t(f) = \frac{|\hat{\mathbf{y}}_{\mathbf{1}_t}(f)|}{|\hat{\mathbf{y}}_{\mathbf{1}_t}(f)| + |\hat{\mathbf{y}}_{\mathbf{2}_t}(f)|}, \tag{4}$$

where $f \in \{1, \ldots, F\}$ represents different frequencies.

Once a time-frequency mask $\mathbf{m}_t$ is computed, it is applied to the magnitude spectra $\mathbf{z}_t$ of the mixture signals to obtain the estimated separation spectra $\hat{\mathbf{s}}_{\mathbf{1}_t}$ and $\hat{\mathbf{s}}_{\mathbf{2}_t}$, which correspond to sources 1 and 2, as follows:

$$\begin{aligned}\hat{\mathbf{s}}_{\mathbf{1}_t}(f) &= \mathbf{m}_t(f)\mathbf{z}_t(f) \\ \hat{\mathbf{s}}_{\mathbf{2}_t}(f) &= (1 - \mathbf{m}_t(f))\, \mathbf{z}_t(f),\end{aligned} \tag{5}$$

where $f \in \{1, \ldots, F\}$ represents different frequencies.

The time-frequency masking function can be viewed as a layer in the neural network as well. Instead of training the network and applying the time-frequency masking to the results separately, we can jointly train the deep learning models with the time-frequency masking functions. We add an extra layer to the original output of the neural network as follows:

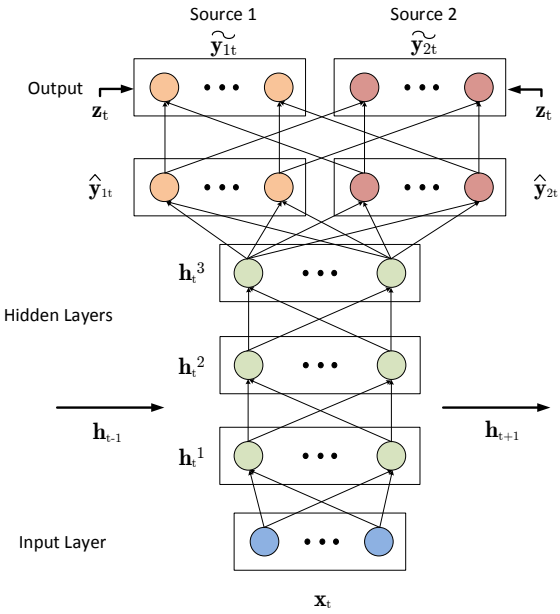


Figure 3. Proposed neural network architecture.

$$\begin{aligned}\tilde{\mathbf{y}}_{\mathbf{1}_t} &= \frac{|\hat{\mathbf{y}}_{\mathbf{1}_t}|}{|\hat{\mathbf{y}}_{\mathbf{1}_t}| + |\hat{\mathbf{y}}_{\mathbf{2}_t}|} \odot \mathbf{z}_t \\ \tilde{\mathbf{y}}_{\mathbf{2}_t} &= \frac{|\hat{\mathbf{y}}_{\mathbf{2}_t}|}{|\hat{\mathbf{y}}_{\mathbf{1}_t}| + |\hat{\mathbf{y}}_{\mathbf{2}_t}|} \odot \mathbf{z}_t,\end{aligned} \tag{6}$$

where the operator $\odot$ is the element-wise multiplication (Hadamard product). In this way, we can integrate the constraints to the network and optimize the network with the masking function jointly. Note that although this extra layer is a deterministic layer, the network weights are optimized for the error metric between and among $\tilde{\mathbf{y}}_{\mathbf{1}_t}$, $\tilde{\mathbf{y}}_{\mathbf{2}_t}$ and $\mathbf{y}_{\mathbf{1}_t}$, $\mathbf{y}_{\mathbf{2}_t}$, using back-propagation. To further smooth the predictions, we can apply masking functions to $\tilde{\mathbf{y}}_{\mathbf{1}_t}$ and $\tilde{\mathbf{y}}_{\mathbf{2}_t}$, as in Eqs. (4) and (5), to get the estimated separation spectra $\tilde{\mathbf{s}}_{\mathbf{1}_t}$ and $\tilde{\mathbf{s}}_{\mathbf{2}_t}$. The time domain signals are reconstructed based on the inverse short time Fourier transform (ISTFT) of the estimated magnitude spectra along with the original mixture phase spectra.

3.3 Training Objectives

Given the output predictions $\hat{\mathbf{y}}_{\mathbf{1}_t}$ and $\hat{\mathbf{y}}_{\mathbf{2}_t}$ (or $\tilde{\mathbf{y}}_{\mathbf{1}_t}$ and $\tilde{\mathbf{y}}_{\mathbf{2}_t}$) of the original sources $\mathbf{y}_{\mathbf{1}_t}$ and $\mathbf{y}_{\mathbf{2}_t}$, we explore optimizing neural network parameters by minimizing the squared error and the generalized Kullback-Leibler (KL) divergence criteria, as follows:

$$J_{MSE} = ||\hat{\mathbf{y}}_{\mathbf{1}_t} - \mathbf{y}_{\mathbf{1}_t}||_2^2 + ||\hat{\mathbf{y}}_{\mathbf{2}_t} - \mathbf{y}_{\mathbf{2}_t}||_2^2 \tag{7}$$

and

$$J_{KL} = D(\mathbf{y}_{\mathbf{1}_t}||\hat{\mathbf{y}}_{\mathbf{1}_t}) + D(\mathbf{y}_{\mathbf{2}_t}||\hat{\mathbf{y}}_{\mathbf{2}_t}), \tag{8}$$

where the measure $D(A||B)$ is defined as:

$$D(A||B) = \sum_i \left(A_i \log \frac{A_i}{B_i} - A_i + B_i \right). \tag{9}$$

$D(\cdot\|\cdot)$ reduces to the KL divergence when $\sum_i A_i = \sum_i B_i = 1$, so that A and B can be regarded as probability distributions.

Furthermore, minimizing Eqs. (7) and (8) is for increasing the similarity between the predictions and the targets. Since one of the goals in source separation problems is to have high signal to interference ratio (SIR), we explore discriminative objective functions that not only increase the similarity between the prediction and its target, but also decrease the similarity between the prediction and the targets of other sources, as follows:

$$||\hat{\mathbf{y}}_{\mathbf{1}_t}-\mathbf{y}_{\mathbf{1}_t}||_2^2-\gamma||\hat{\mathbf{y}}_{\mathbf{1}_t}-\mathbf{y}_{\mathbf{2}_t}||_2^2+||\hat{\mathbf{y}}_{\mathbf{2}_t}-\mathbf{y}_{\mathbf{2}_t}||_2^2-\gamma||\hat{\mathbf{y}}_{\mathbf{2}_t}-\mathbf{y}_{\mathbf{1}_t}||_2^2 \tag{10}$$

and

$$D(\mathbf{y}_{\mathbf{1}_t}||\hat{\mathbf{y}}_{\mathbf{1}_t})-\gamma D(\mathbf{y}_{\mathbf{1}_t}||\hat{\mathbf{y}}_{\mathbf{2}_t})+D(\mathbf{y}_{\mathbf{2}_t}||\hat{\mathbf{y}}_{\mathbf{2}_t})-\gamma D(\mathbf{y}_{\mathbf{2}_t}||\hat{\mathbf{y}}_{\mathbf{1}_t}), \tag{11}$$

where γ is a constant chosen by the performance on the development set.

4. EXPERIMENTS

4.1 Setting

Our system is evaluated using the MIR-1K dataset [6]. [1] A thousand song clips are encoded with a sample rate of 16 KHz, with durations from 4 to 13 seconds. The clips were extracted from 110 Chinese karaoke songs performed by both male and female amateurs. There are manual annotations of the pitch contours, lyrics, indices and types for unvoiced frames, and the indices of the vocal and non-vocal frames. Note that each clip contains the singing voice and the background music in different channels. Only the singing voice and background music are used in our experiments.

Following the evaluation framework in [13, 17], we use 175 clips sung by one male and one female singer ('abjones' and 'amy') as the training and development set. [2] The remaining 825 clips of 17 singers are used for testing. For each clip, we mixed the singing voice and the background music with equal energy (i.e. 0 dB SNR). The goal is to separate the singing voice from the background music.

To quantitatively evaluate source separation results, we use Source to Interference Ratio (SIR), Source to Artifacts Ratio (SAR), and Source to Distortion Ratio (SDR) by BSS-EVAL 3.0 metrics [14]. The Normalized SDR (NSDR) is defined as:

$$\text{NSDR}(\hat{\mathbf{v}}, \mathbf{v}, \mathbf{x}) = \text{SDR}(\hat{\mathbf{v}}, \mathbf{v}) - \text{SDR}(\mathbf{x}, \mathbf{v}), \tag{12}$$

where $\hat{\mathbf{v}}$ is the resynthesized singing voice, $\mathbf{v}$ is the original clean singing voice, and $\mathbf{x}$ is the mixture. NSDR is for estimating the improvement of the SDR between the preprocessed mixture $\mathbf{x}$ and the separated singing voice $\hat{\mathbf{v}}$. We report the overall performance via Global NSDR (GNSDR), Global SIR (GSIR), and Global SAR (GSAR), which are the weighted means of the NSDRs, SIRs, SARs, respectively, over all test clips weighted by their length. Higher values of SDR, SAR, and SIR represent better separation quality. The suppression of the interfering source is reflected in SIR. The artifacts introduced by the separation process are reflected in SAR. The overall performance is reflected in SDR.

For training the network, in order to increase the variety of training samples, we circularly shift (in the time domain) the singing voice signals and mix them with the background music.

In the experiments, we use magnitude spectra as input features to the neural network. The spectral representation is extracted using a 1024-point short time Fourier transform (STFT) with 50% overlap. Empirically, we found that using log-mel filterbank features or log power spectrum provide worse performance.

For our proposed neural networks, we optimize our models by back-propagating the gradients with respect to the training objectives. The limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) algorithm is used to train the models from random initialization. We set the maximum epoch to 400 and select the best model according to the development set. The sound examples and more details of this work are available online. [3]

4.2 Experimental Results

In this section, we compare different deep learning models from several aspects, including the effect of different input context sizes, the effect of different circular shift steps, the effect of different output formats, the effect of different deep recurrent neural network structures, and the effect of the discriminative training objectives.

For simplicity, unless mentioned explicitly, we report the results using 3 hidden layers of 1000 hidden units neural networks with the mean squared error criterion, joint masking training, and 10K samples as the circular shift step size using features with a context window size of 3 frames. We denote the DRNN-k as the DRNN with the recurrent connection at the k-th hidden layer. We select the models based on the GNSDR results on the development set.

First, we explore the case of using single frame features, and the cases of concatenating neighboring 1 and 2 frames as features (context window sizes 1, 3, and 5, respectively). Table 1 reports the results using DNNs with context window sizes $1, 3$, and 5. We can observe that concatenating neighboring 1 frame provides better results compared with the other cases. Hence, we fix the context window size to be 3 in the following experiments.

Table 2 shows the difference between different circular shift step sizes for deep neural networks. We explore the cases without circular shift and the circular shift with a step size of {50K, 25K, 10K} samples. We can observe that the separation performance improves when the number of training samples increases (i.e. the step size of circular

[1] https://sites.google.com/site/unvoicedsoundseparation/mir-1k

[2] Four clips, abjones_5_08, abjones_5_09, amy_9_08, amy_9_09, are used as the development set for adjusting hyper-parameters.

[3] https://sites.google.com/site/deeplearningsourceseparation/

Model (context window size)	GNSDR	GSIR	GSAR
DNN (1)	6.63	10.81	9.77
DNN (3)	6.93	10.99	10.15
DNN (5)	6.84	10.80	10.18

Table 1. Results with input features concatenated from different context window sizes.

Model (circular shift step size)	GNSDR	GSIR	GSAR
DNN (no shift)	6.30	9.97	9.99
DNN (50,000)	6.62	10.46	10.07
DNN (25,000)	6.86	11.01	10.00
DNN (10,000)	6.93	10.99	10.15

Table 2. Results with different circular shift step sizes.

Model (num. of output sources, joint mask)	GNSDR	GSIR	GSAR
DNN (1, no)	5.64	8.87	9.73
DNN (2, no)	6.44	9.08	11.26
DNN (2, yes)	6.93	10.99	10.15

Table 3. Deep neural network output layer comparison using single source as a target and using two sources as targets (with and without joint mask training). In the "joint mask" training, the network training objective is computed after time-frequency masking.

shift decreases). Since the improvement is relatively small when we further increase the number of training samples, we fix the circular shift size to be 10K samples.

Table 3 presents the results with different output layer formats. We compare using single source as a target (row 1) and using two sources as targets in the output layer (row 2 and row 3). We observe that modeling two sources simultaneously provides better performance. Comparing row 2 and row 3 in Table 3, we observe that using the joint mask training further improves the results.

Table 4 presents the results of different deep recurrent neural network architectures (DNN, DRNN with different recurrent connections, and sRNN) and the results of different objective functions. We can observe that the models with the generalized KL divergence provide higher GSARs, but lower GSIRs, compared to the models with the mean squared error objective. Both objective functions provide similar GNSDRs. For different network architectures, we can observe that DRNN with recurrent connection at the second hidden layer provides the best results. In addition, all the DRNN models achieve better results compared to DNN models by utilizing temporal information.

Table 5 presents the results of different deep recurrent neural network architectures (DNN, DRNN with different recurrent connections, and sRNN) with and without discriminative training. We can observe that discriminative training improves GSIR, but decreases GSAR. Overall, GNSDR is slightly improved.

Model (objective)	GNSDR	GSIR	GSAR
DNN (MSE)	6.93	10.99	10.15
DRNN-1 (MSE)	7.11	11.74	9.93
DRNN-2 (MSE)	7.27	11.98	9.99
DRNN-3 (MSE)	7.14	11.48	10.15
sRNN (MSE)	7.09	11.72	9.88
DNN (KL)	7.06	11.34	10.07
DRNN-1 (KL)	7.09	11.48	10.05
DRNN-2 (KL)	7.27	11.35	10.47
DRNN-3 (KL)	7.10	11.14	10.34
sRNN (KL)	7.16	11.50	10.11

Table 4. The results of different architectures and different objective functions. The "MSE" denotes the mean squared error and the "KL" denotes the generalized KL divergence criterion.

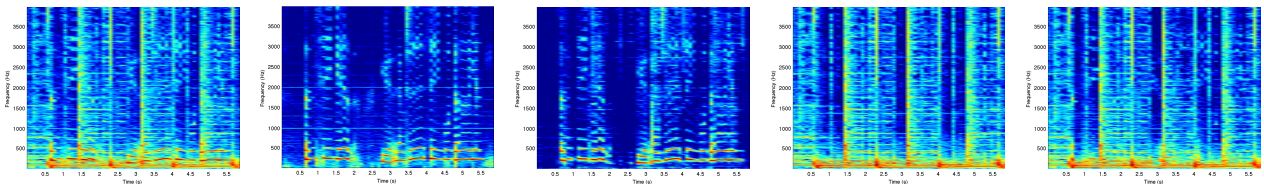
Model	GNSDR	GSIR	GSAR
DNN	6.93	10.99	10.15
DRNN-1	7.11	11.74	9.93
DRNN-2	7.27	11.98	9.99
DRNN-3	7.14	11.48	10.15
sRNN	7.09	11.72	9.88
DNN + discrim	7.09	12.11	9.67
DRNN-1 + discrim	7.21	12.76	9.56
DRNN-2 + discrim	7.45	13.08	9.68
DRNN-3 + discrim	7.09	11.69	10.00
sRNN + discrim	7.15	12.79	9.39

Table 5. The comparison for the effect of discriminative training using different architectures. The "discrim" denotes the models with discriminative training.

Finally, we compare our best results with other previous work under the same setting. Table 6 shows the results with unsupervised and supervised settings. Our proposed models achieve 2.30∼2.48 dB GNSDR gain, 4.32∼5.42 dB GSIR gain with similar GSAR performance, compared with the RNMF model [13]. An example of the separation results is shown in Figure 4.

5. CONCLUSION AND FUTURE WORK

In this paper, we explore using deep learning models for singing voice separation from monaural recordings. Specifically, we explore different deep learning architectures, including deep neural networks and deep recurrent neural networks. We further enhance the results by jointly optimizing a soft mask function with the networks and exploring the discriminative training criteria. Overall, our proposed models achieve 2.30∼2.48 dB GNSDR gain and 4.32∼5.42 dB GSIR gain, compared to the previous proposed methods, while maintaining similar GSARs. Our proposed models can also be applied to many other applications such as main melody extraction.



(a) Mixutre (b) Clean vocal (c) Recovered vocal (d) Clean music (e) Recovered music

Figure 4. (a) The mixture (singing voice and music accompaniment) magnitude spectrogram (in log scale) for the clip Ani_1_01 in MIR-1K; (b) (d) The groundtruth spectrograms for the two sources; (c) (e) The separation results from our proposed model (DRNN-2 + discrim).

Unsupervised			
Model	GNSDR	GSIR	GSAR
RPCA [7]	3.15	4.43	11.09
RPCAh [16]	3.25	4.52	11.10
RPCAh + FASST [16]	3.84	6.22	9.19
Supervised			
Model	GNSDR	GSIR	GSAR
MLRR [17]	3.85	5.63	10.70
RNMF [13]	4.97	7.66	10.03
DRNN-2	**7.27**	**11.98**	**9.99**
DRNN-2 + discrim	**7.45**	**13.08**	**9.68**

Table 6. Comparison between our models and previous proposed approaches. The "discrim" denotes the models with discriminative training.

6. ACKNOWLEDGEMENT

We thank the authors in [13] for providing their trained model for comparison. This research was supported by U.S. ARL and ARO under grant number W911NF-09-1-0383. This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number ACI-1053575.

7. REFERENCES

[1] N. Boulanger-Lewandowski, G. Mysore, and M. Hoffman. Exploiting long-term temporal dependencies in NMF using recurrent neural networks with application to source separation. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014.

[2] X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. In *JMLR W&CP: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2011)*, 2011.

[3] M. Hermans and B. Schrauwen. Training and analysing deep recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 190–198, 2013.

[4] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury. Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal Processing Magazine*, 29:82–97, Nov. 2012.

[5] G. Hinton and R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504 – 507, 2006.

[6] C.-L. Hsu and J.-S.R. Jang. On the improvement of singing voice separation for monaural recordings using the MIR-1K dataset. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(2):310 –319, Feb. 2010.

[7] P.-S. Huang, S. D. Chen, P. Smaragdis, and M. Hasegawa-Johnson. Singing-voice separation from monaural recordings using robust principal component analysis. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 57–60, 2012.

[8] P.-S. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. Heck. Learning deep structured semantic models for web search using clickthrough data. In *ACM International Conference on Information and Knowledge Management (CIKM)*, 2013.

[9] P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis. Deep learning for monaural speech separation. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014.

[10] A. L. Maas, Q. V Le, T. M O'Neil, O. Vinyals, P. Nguyen, and A. Y. Ng. Recurrent neural networks for noise reduction in robust ASR. In *INTERSPEECH*, 2012.

[11] A. Narayanan and D. Wang. Ideal ratio mask estimation using deep neural networks for robust speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 2013.

[12] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio. How to construct deep recurrent neural networks. In *International Conference on Learning Representations*, 2014.

[13] P. Sprechmann, A. Bronstein, and G. Sapiro. Real-time online singing voice separation from monaural recordings using robust low-rank modeling. In *Proceedings of the 13th International Society for Music Information Retrieval Conference*, 2012.

[14] E. Vincent, R. Gribonval, and C. Fevotte. Performance measurement in blind audio source separation. *Audio, Speech, and Language Processing, IEEE Transactions on*, 14(4):1462 –1469, July 2006.

[15] Y. Wang and D. Wang. Towards scaling up classification-based speech separation. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(7):1381–1390, 2013.

[16] Y.-H. Yang. On sparse and low-rank matrix decomposition for singing voice separation. In *ACM Multimedia*, 2012.

[17] Y.-H. Yang. Low-rank representation of both singing voice and music accompaniment via learned dictionaries. In *Proceedings of the 14th International Society for Music Information Retrieval Conference*, November 4-8 2013.