Two Stage Single Channel Audio Source Separation using Deep Neural Networks

Emad M. Grais, Gerard Roma, Andrew J. R. Simpson, and Mark D. Plumbley, *Fellow, IEEE*

Abstract—Most single channel audio source separation (SCASS) approaches produce separated sources accompanied by interference from other sources and other distortions. To tackle this problem, we propose to separate the sources in two stages. In the first stage, the sources are separated from the mixed signal. In the second stage, the interference between the separated sources and the distortions are reduced using deep neural networks (DNNs). We propose two methods that use DNNs to improve the quality of the separated sources in the second stage. In the first method, each separated source is improved individually using its own trained DNN, while in the second method all the separated sources are improved together using a single DNN. To further improve the quality of the separated sources, the DNNs in the second stage are trained discriminatively to further decrease the interference and the distortions of the separated sources. Our experimental results show that using two stages of separation improves the quality of the separated signals by decreasing the interference between the separated sources and distortions compared to separating the sources using a single stage of separation.

Index Terms—Single channel audio source separation, deep neural networks, audio enhancement, discriminative learning.

I. INTRODUCTION

The main aim of single channel audio source separation (SCASS) is to separate different audio sources from their single mixture [1]–[4]. Nonnegative matrix factorization (NMF) [5] is one of the most popular approaches for SCASS [1], [6], [7]. NMF is usually used in SCASS to decompose the spectrogram of the mixed signal into weighted linear combinations of trained dictionaries of the sources in the mixed signal. The estimate of each source is then computed from the decomposition results.

Deep neural networks (DNNs) have also been used recently for SCASS [8]–[11]. Mostly, there are two main approaches that use DNNs for SCASS. In the first approach, the DNNs are used to map the features of the mixed signal into the features of the sources directly [10]–[12]. The second approach for using DNNs for SCASS is to map the mixed signal into spectral masks that explain the contribution of each source in the mixed signal [8], [9], [13]. The masks take bounded values between zero and one, representing the ratios between the sources in the mixed signal. The advantage of the first approach is that the DNNs are trained using the reference sources directly, which

makes the training of the DNN less sensitive to the variation of the mixing ratio of the sources in the mixed signals [14]. The advantage of the second approach is that the DNNs are trained to produce bounded values (masks between zero and one) rather than training the DNNs to predict the sources which can take any real values [13].

Most of the approaches that have been used for SCASS produce separated sources with interference from the other sources and distortions [15], [16]. To tackle this problem, in [15]–[21] the separated sources were considered as distorted signals that need to be enhanced. After separating the sources from the mixed signal, NMF was used in [15] and Gaussian mixture models (GMMs) were used in [16] to enhance each separated source individually. In [17]–[21], source separation was used to separate speech signals from background noise (speech denoising) in two stages. In the first stage the speech signals were separated/denoised from the background noise and in the second stage the quality of the denoised speech signals was improved. In [17], [18] DNNs were used in two stages, in the first stage DNNs were used to remove the noise followed by other DNNs for: channel compensation in [17], and incorporating contextual information in [18]. In [19]–[21] different combinations of NMF and DNN-masks were used in different stages to improve the perceptual quality of the denoised speech signals. In [14], [22]–[24] many stages of DNNs were used for different audio signal processing applications.

In this paper we propose to use DNNs to enhance the quality of the separated audio sources. We propose to separate the sources in two stages. In the first stage, the sources are separated from the mixed signal. In the second stage, the separated sources are enhanced using DNNs to decrease the interference between the separated sources and distortions. We call the first stage the *separation stage* and the second stage the *enhancement stage*. We propose two methods of using DNNs to enhance the separated sources in the second stage. The first proposed method is to enhance each separated source individually using its own trained DNN. Enhancing sources individually may work well when each source does not contain information from the other sources. In SCASS, each separated source often contains remaining signals from the other sources [16]. To consider the information of each source that appears in the other separated sources, we also propose a second method of enhancement where all the separated sources are enhanced together using a single DNN. For both methods, the DNNs in the second stage are trained discriminatively to maximize the differences between the predicted sources which decreases the interference between the separated sources and

This work is supported by grants EP/L027119/1 and EP/L027119/2 from the U.K. Engineering and Physical Sciences Research Council (EPSRC). (Corresponding author: Emad M. Grais.) The authors are with Centre for Vision, Speech and Signal Processing, University of Surrey, Guildford GU2 7XH, U.K. (e-mail: grais@surrey.ac.uk, m.plumbley@surrey.ac.uk). G. Roma is now with Georgia Institute of Technology, USA (e-mail: gerard.roma@gatech.edu). A. J. R. Simpson now runs an independent AI lab.

also the distortions [10], [11], [25]. To show that our proposed enhancement methods can be applied after various source separation techniques, we apply the proposed enhancement approach to two different baseline SCASS techniques: NMF and DNNs. In the first (separation) stage, either NMF or a DNN is used to separate the mixed signal. In the second (enhancement) stage, DNNs are used to enhance the separated sources. We also study the case of using NMF for enhancement in the second stage inspired by [15].

The contributions of this paper are as follows. (1) Using different DNNs for a two stage source separation and enhancement approach by introducing two new methods of applying the enhancement to the separated sources: using a single DNN to enhance all the separated sources together so that the missing parts of each source that appear in the other separated sources can still be used for enhancement; the second method is to enhance each source independently. (2) Training the DNNs in the two stages on different target outputs for SCASS [26], where the DNN in the first stage is learned to predict spectral masks and in the second stage the DNNs are used to predict the sources directly while keeping the outputs of the DNNs bounded in both stages. This means we train the whole system (separation and enhancement) on different representations for the training data. We also train the models in the enhancement stage over a different set of training data than the training data for the models in the first stage. This allows the models in the enhancement stage to be trained to deal with difficult scenarios when the models in the separation stage have to separate mixtures that are different than those used for training the models for separation, which usually yields separated signals with poor quality. (3) Using discriminative learning (regularized cost functions) to train the DNNs in the enhancement stage to further minimize the interference between the separated sources.

This paper is organized as follows. In Section II, the mathematical formulation of the SCASS problem is introduced. Section III-A provides an overview of using NMF for SCASS. Sections III-B and III-C give an overview of using DNNs for source separation. The main contributions of this paper of using DNNs to enhance the separated sources are presented in Section IV. In the remaining sections, the experiments, results, and the conclusion are presented.

II. PROBLEM FORMULATION OF SCASS

Given a mixture of I audio sources as $y(t) = \sum_{i=1}^I s_i(t)$, the aim of the SCASS is to find estimates $\hat{s}_i(t)$ for the sources $s_i(t)$, $\forall i$ from the mixed signal $y(t)$. The estimate $\hat{S}_i(n, f)$ for source i in the short time Fourier transform (STFT) domain can be found by predicting a time-frequency mask $M_i(n, f)$ that scales the mixed signal according to the contribution of source i in the mixed signal as follows [8], [13]:

$$\hat{S}_i(n, f) = M_i(n, f) \times Y(n, f) \quad (1)$$

where $Y(n, f)$ is the STFT of the observed mixed signal $y(t)$, n and f are the time and frequency indices respectively. $M_i(n, f)$ takes real values between zero and one. The main goal here is to predict masks $M_i(n, f)$, $\forall i$ that separate the

sources from the mixed signal. For the rest of this paper, we denote the magnitude spectrograms and the masks in a matrix form as $\mathbf{Y}$, $\hat{\mathbf{S}}_i$, and $\mathbf{M}_i$.

III. SINGLE STAGE SCASS METHODS

Many methods have been proposed to tackle the SCASS problem [1]–[4]. In this paper we focus on the most common approaches for SCASS which are: nonnegative matrix factorization [1], [7] and deep neural networks [8], [10], [11].

A. Nonnegative matrix factorization for SCASS

Nonnegative matrix factorization (NMF) [5] is one of the most popular approaches for SCASS [1]. NMF is used to decompose a nonnegative matrix $\mathbf{V}$ into a multiplication of a nonnegative basis matrix (dictionary) $\mathbf{B}$ and a nonnegative gain matrix $\mathbf{G}$ as $\mathbf{V} \approx \mathbf{B}\mathbf{G}$. The solution for $\mathbf{B}$ and $\mathbf{G}$ can be found by minimizing a cost function such as the following generalized Kullback-Leibler (KL) divergence cost function [5]:

$$\min_{\mathbf{B}, \mathbf{G}} \sum_{a,b} \left(V_{a,b} \log \frac{V_{a,b}}{(\mathbf{B}\mathbf{G})_{a,b}} - V_{a,b} + (\mathbf{B}\mathbf{G})_{a,b} \right) \quad (2)$$

where a and b are the row and column indices of the matrices. The solutions of the basis and gain matrices for Eq. (2) can be computed by the following multiplicative update rules for the basis and gain matrices respectively [5]:

$$\mathbf{B} \leftarrow \mathbf{B} \odot \frac{(\mathbf{V} \mathbf{G}^T)}{\mathbf{1} \mathbf{G}^T}, \quad \mathbf{G} \leftarrow \mathbf{G} \odot \frac{\mathbf{B}^T (\mathbf{V})}{\mathbf{B}^T \mathbf{1}} \quad (3)$$

where $\mathbf{1}$ is a matrix of ones, the operation $\odot$ is Hadamard (element-wise) multiplication, and all divisions are element-wise operations [5]. The matrices $\mathbf{B}$ and $\mathbf{G}$ are typically initialized by positive random numbers and then updated iteratively using the update rules in Eq. (3).

NMF is often used for SCASS in two phases: a training phase and a separation phase [27], [28]. In the training phase, the magnitude spectrograms of a set of training data for the sources $\mathbf{S}_{\text{tri}}$ are decomposed into basis and gain matrices as follows:

$$\mathbf{S}_{\text{tri}} \approx \mathbf{B}_{\text{tri}} \mathbf{G}_{\text{tri}}, \quad i = 1, 2, \dots, I \quad (4)$$

where the subscript tri indicates the training data of source i . The update rules in Eq. (3) are used to compute the basis and gain matrices in Eq. (4). The basis matrices $\mathbf{B}_{\text{tri}}$ are used as trained dictionaries to represent the sources. In the separation phase, the magnitude spectrogram $\mathbf{Y}$ of the mixed signal is decomposed into a weighted linear combination of the trained dictionaries as:

$$\mathbf{Y} \approx [\mathbf{B}_{\text{tri}1}, \dots, \mathbf{B}_{\text{tri}I}] \mathbf{G}_{\text{ts}}, \quad \text{where } \mathbf{G}_{\text{ts}} = \begin{bmatrix} \mathbf{G}_{\text{ts}1} \\ \vdots \\ \mathbf{G}_{\text{ts}I} \end{bmatrix} \quad (5)$$

and “ts” denotes the test/separation phase. The only unknown here is the gain matrix $\mathbf{G}_{\text{ts}}$ since the trained dictionaries $\mathbf{B}_{\text{tri}}$ are fixed. The gain update rule in Eq. (3) is used to compute $\mathbf{G}_{\text{ts}}$. Then the estimates of the sources are computed as follows:

$$\tilde{\mathbf{S}}_i = \mathbf{M}_{\text{NMF}i} \odot \mathbf{Y}, \quad \forall i \quad \text{where } \mathbf{M}_{\text{NMF}i} = \frac{\mathbf{B}_{\text{tri}} \mathbf{G}_{\text{ts}i}}{\sum_{i=1}^I \mathbf{B}_{\text{tri}} \mathbf{G}_{\text{ts}i}} \quad (6)$$

and division is also an element-wise operation.

B. DNNs for SCASS to predict the sources

The deep neural networks (DNNs) that are used in this paper are multi-layer feed forward neural networks. DNNs are good in modelling the structure of the data in many machine learning tasks [29], [30], and they have been used successfully to tackle the SCASS problem [8], [12], [13]. In a similar way to NMF, DNNs are usually used for SCASS in two phases: a training phase and a separation phase [8], [13]. In the training phase, the training data is used to train a DNN to map the mixed signal into the reference sources. Since the sources can take values that are not limited to a small range, the DNN in this case has to be trained over a wide dynamic range of values for each source [10]–[12]. In the separation stage the trained DNNs are used to map the mixed signal into the sources.

C. DNNs for SCASS to predict masks

Another method of using DNNs in SCASS is to train the DNNs to predict spectral masks. The masks are then used to separate the sources by scaling the mixed signal according to the contribution of each source in the mixed signal. Variety of masks have been introduced in [8], [13]. In this paper, for simplicity, we only consider the soft masks [8], [13]. Given the magnitude spectrograms $\mathbf{S}_{tri}, \forall i$ of a set of training data for the sources, the DNN is trained to predict reference soft masks $\mathbf{M}_{tri}, \forall i$ as [8]:

$$\mathbf{M}_{tri} = \frac{\mathbf{S}_{tri}}{\sum_{i=1}^I \mathbf{S}_{tri}}, \forall i \quad (7)$$

where the division here is done element-wise. The reference/target output of the DNN for all sources is formed by stacking the reference masks for all sources as

$$\mathbf{M}_{tr} = [\mathbf{M}_{tr1}, \dots, \mathbf{M}_{tri}, \dots, \mathbf{M}_{trI}]. \quad (8)$$

The input of the DNN is the magnitude spectrogram $\mathbf{X}_{tr}$ of the mixed signals of the training data which is formulated as

$$\mathbf{X}_{tr} = \sum_{i=1}^I \mathbf{S}_{tri}.$$

The DNN is trained by minimizing a certain cost function. Many cost functions have been tested before and the simple mean square error cost function seems to work well [31]. In this work, the DNN is trained to minimize the following cost function [8]:

$$C_1 = \sum_{n,f} (\mathbf{Z}_{tr}(n, f) - \mathbf{M}_{tr}(n, f))^2 \quad (9)$$

where $\mathbf{Z}_{tr}$ is the actual output of the final layer of the DNN and $\mathbf{M}_{tr} \in [0, 1]$ is the reference mask which is computed from Eqs. (7) and (8). The activation functions of the output layer for the DNN are often sigmoid functions, thus $\mathbf{Z}_{tr} \in [0, 1]$.

In the separation phase, the frames of the magnitude spectrogram of the mixed signal are fed to the trained DNN to produce a mask $\mathbf{Z}_{ts}$ which is composed of a concatenation of different masks for the sources as $\mathbf{Z}_{ts} = [\mathbf{Z}_{ts1}, \dots, \mathbf{Z}_{tsi}, \dots, \mathbf{Z}_{tsI}]$. The estimated masks $\mathbf{Z}_{tsi}, \forall i$ are then used to separate the sources similar to Eq. (1).

IV. TWO STAGE SCASS APPROACHES

The signals separated by either NMF or DNN are often distorted and each separated source may include unwanted signals (interference) from the other sources [15], [16]. To tackle this problem, we propose to use a second stage of separation (enhancement stage) based on DNNs to improve the quality of the separated sources. The inputs of the DNNs in the second stage are the separated sources from the first stage and the outputs are the enhanced separated sources with less interference and distortions than the outputs of the first stage.

Signal enhancement has previously been done by training a model for each signal and each signal is then enhanced using its own trained model [16], [32]. In this paper, we train a DNN model for each source and each separated source is enhanced individually using its own trained model. Fig. 1 shows one of our proposed two-stage source separation approaches for SCASS, where a DNN is used for separation. The DNN in the first stage (DNN-A) in Fig. 1 is used to map the mixed signal into a mask that is then used to separate the sources from the mixed signal as shown in Section III-C, then each trained DNN in the second stage (DNN-B₁ to DNN-B_I) is used to enhance its corresponding separated source individually.

Enhancing sources individually may work well in the case when there is no interaction between the sources. In the SCASS problem, all the separated sources are estimated from the same mixture. Each separated source often contains remaining signals (information) from the other sources. When each separated source is enhanced individually, the remaining signals of each source that appear in the other separated sources are not used in enhancing its own source. To consider the information of each source that appears in the other separated sources, we propose another method of enhancement based on enhancing all the separated sources together using a single trained model for all sources. Fig. 2 shows our second proposed enhancement approach where all the separated sources are enhanced using a single DNN (DNN-C). The DNNs in the second stage (DNN-B₁ to DNN-B_I, or DNN-C) in Figs. 1 and 2 are trained to map the distorted separated sources into their corresponding clean sources. In the case of using NMF for separation, DNN-A in Figs. 1 and 2 is replaced by NMF. Also in the case of using NMF for enhancement, the DNNs in the second stage are replaced by NMF.

A. Training the DNNs for separation and enhancement

The DNN in the separation stage (DNN-A) in Figs. 1 and 2 is used for separation by predicting masks as shown in Section III-C. The reason for not using DNN-A to predict the sources directly as shown in Section III-B is to avoid training it on a wide range of the possible values that the sources can take. As shown in Section III-C, to train DNN-A in the first stage, training mixed signals are used as the input for DNN-A and their corresponding reference masks as shown in Eqs. 7 and 8 are used as reference outputs for DNN-A. Usually the trained DNN-A is used to separate mixed signals that are different than the signals in the training data [8], [10]–[12], and since

Fig. 1. The overview of the proposed approach of using DNN-A for source separation and DNN-B₁, ..., DNN-B_i, ..., DNN-B_I to enhance the separated sources individually.

Fig. 2. The overview of the proposed approach of using DNN-A for source separation and a single DNN (DNN-C) for enhancement.

the main idea of using the DNNs in the second stage is to enhance the output separated sources of DNN-A, hence, it is undesirable to train DNN-A and the DNNs in the second stage using the same dataset. Therefore, we divide the available training data into two sets. The first set is used to train DNN-A and the second set is used to train the DNNs in the second stage. To generate the training data to train the DNNs in the second stage in Figs. 1 and 2, the trained DNN-A is used to separate mixed signals from the second set of the training data. The mixed signal of this set of training data is formulated as

$$\mathbf{X}_{\text{tr}}^{(2)} = \sum_{i=1}^I \mathbf{S}_{\text{tri}}^{(2)}$$

where $\mathbf{X}_{\text{tr}}^{(2)}$ is the magnitude spectrogram of the mixed signal in the second set of the training data, the superscript “(2)” indicates that the second set of the training data is used in this stage. The frames of $\mathbf{X}_{\text{tr}}^{(2)}$ are fed as inputs to DNN-A, which then produces mask $\mathbf{Z}_{\text{tr}}^{(2)}$ which is a concatenation of masks for many sources as $\mathbf{Z}_{\text{tr}}^{(2)} = [\mathbf{Z}_{\text{tr}1}^{(2)}, \dots, \mathbf{Z}_{\text{tri}}^{(2)}, \dots, \mathbf{Z}_{\text{tr}I}^{(2)}]$. The estimated masks are used to separate the sources as follows:

$$\tilde{\mathbf{S}}_{\text{tri}}^{(2)} = \mathbf{Z}_{\text{tri}}^{(2)} \odot \mathbf{X}_{\text{tr}}^{(2)}, \forall i. \quad (10)$$

In our first proposed method of enhancement, each DNN (DNN-B₁ to DNN-B_I) in the second stage should be trained to enhance its corresponding separated source only as shown in Fig. 1. In this method, the missing data for each source i that appears in the other separated sources will not be used

to train its own DNN-B_i. In the second proposed method of enhancement, the DNN in the second stage (DNN-C) should be trained to enhance all the separated sources ($\tilde{\mathbf{S}}_{\text{tri}}^{(2)}, \forall i$) together as shown in Fig. 2. For both methods, all the DNNs in the second stage are trained using the separated signals $\tilde{\mathbf{S}}_{\text{tri}}^{(2)}, \forall i$ as inputs and their corresponding reference/clean signals $\mathbf{S}_{\text{tri}}^{(2)}, \forall i$ as outputs. Each frame of the spectrogram $\mathbf{S}_{\text{tri}}^{(2)}$ of the reference source i is normalized to have a unit Euclidean-norm. As we will show later, this normalization allows us to train the DNNs in the second stage in both proposed methods of enhancement to produce bounded values in its output layers without any need to train them over a wide range of values that the sources can have. Since the reference normalized signals have values between zero and one, we choose the activation functions of the output layer of the DNNs in the second stage to be sigmoid functions. To decrease the interference between the separated sources, the learning of the DNNs in the enhancement stage is guided/regularized to maximize the dissimilarity between the estimated signals for different sources [10], [25].

The DNNs for the first enhancement method in Fig. 1 are trained to minimize the following cost function for all sources:

$$C_B = \sum_{n,f} \left(\mathbf{R}_{\text{tri}}^{(2)}(n, f) - \mathbf{S}_{\text{tri}}^{(2)}(n, f) \right)^2 - \lambda \sum_{i \neq j} \sum_{n,f} \left(\mathbf{R}_{\text{tri}}^{(2)}(n, f) - \mathbf{S}_{\text{tr}j}^{(2)}(n, f) \right)^2, \forall i \quad (11)$$

where λ is a regularization parameter and $\mathbf{R}_{\text{tri}}^{(2)}$ is the actual output of DNN-B_i in the second stage. In the second method of enhancement which is shown in Fig. 2, the input and the output for DNN-C are the concatenation of the separated signals computed from Eq. (10) as $\mathbf{U} = [\tilde{\mathbf{S}}_{\text{tr}1}^{(2)}, \dots, \tilde{\mathbf{S}}_{\text{tr}I}^{(2)}]$ and their reference signals $\mathbf{V} = [\mathbf{S}_{\text{tr}1}^{(2)}, \dots, \mathbf{S}_{\text{tr}I}^{(2)}]$ respectively. DNN-C is trained to produce in its output layer the concatenation of the normalized reference signals $\mathbf{V} = [\mathbf{S}_{\text{tr}1}^{(2)}, \dots, \mathbf{S}_{\text{tr}I}^{(2)}]$.

DNN-C in this case is trained to minimize the following cost function:

$$C_C = \sum_{n,f} (\mathbf{Q}(n, f) - \mathbf{V}(n, f))^2 - \lambda \sum_{i \neq j} \sum_{n,f} \left(\mathbf{Q}_i(n, f) - \mathbf{S}_{\text{tr}j}^{(2)}(n, f) \right)^2, \forall i \quad (12)$$

where $\mathbf{Q}$ is the actual output of the DNN which is a concatenation of different sources as $\mathbf{Q} = [\mathbf{Q}_1, \dots, \mathbf{Q}_i, \dots, \mathbf{Q}_I]$. The output $\mathbf{Q}_i$ is the set of DNN-C output nodes that correspond to the normalized reference output of the i^{th} source $\mathbf{S}_{\text{tri}}^{(2)}$.

The first term in the cost functions in Eqs. (11) and (12) minimize the difference between the outputs of the DNNs and their corresponding reference signals. The second term in Eqs. (11) and (12) maximizes the differences between the estimated DNN outputs of different sources which decreases the possibility of each set of the outputs of the DNNs from representing the other set. This helps in achieving good separation for the estimated sources [10], [25]. Note that, DNN-A is trained to

predict a mask in its output layer, while DNN-B₁ to DNN-B_I, and DNN-C are trained to predict normalized magnitude spectrograms of the sources in their output layers. This means the outputs of the DNNs in both stages are bounded between zero and one.

B. Testing the trained DNNs for separation and enhancement

Given the magnitude spectrogram $\mathbf{Y}$ of the mixed signal, the trained DNN-A is first used to separate the mixed signals. The frames of $\mathbf{Y}$ are fed to DNN-A to produce concatenated masks in its output layer as $\tilde{\mathbf{Z}}_{\text{ts}} = [\tilde{\mathbf{Z}}_{\text{ts}1}, \dots, \tilde{\mathbf{Z}}_{\text{ts}i}, \dots, \tilde{\mathbf{Z}}_{\text{ts}I}]$. The output masks are then used to compute initial estimates for the magnitude spectra of the sources as follows:

$$\tilde{\mathbf{S}}_{\text{ts}i} = \mathbf{Z}_{\text{ts}i} \odot \mathbf{Y}, \quad \forall i. \quad (13)$$

The initial estimates for the sources $\tilde{\mathbf{S}}_{\text{ts}i}, \forall i$ are usually distorted [15], [16], and need to be enhanced by either DNN-B₁ to DNN-B_I or DNN-C in the second stage. The source signals can have any values but the DNNs in the second stage are trained to produce bounded values and also the output nodes of the DNNs in the second stage are composed of sigmoid activation functions that gives values between zero and one. To retain the scale information between the sources, the Euclidean norm (gain) of each frame in the spectrograms of the initially estimated sources $\tilde{\mathbf{S}}_{\text{ts}1}$ to $\tilde{\mathbf{S}}_{\text{ts}I}$ are computed as $\alpha_{\text{ts}i} = [\alpha_{1,i}, \dots, \alpha_{n,i}, \dots, \alpha_{N,i}]$, $\forall i$, and saved to be used later, where N is the number of frames in each source. The initially separated sources are then fed to the DNNs in the second stage either as shown in Fig. 1 for DNN-B₁ to DNN-B_I or the separated sources are concatenated as $\tilde{\mathbf{S}}_{\text{ts}} = [\tilde{\mathbf{S}}_{\text{ts}1} : \tilde{\mathbf{S}}_{\text{ts}I}]$, and then fed to DNN-C in the second stage as shown in Fig. 2. The outputs of the DNNs in the second stage are $\hat{\mathbf{S}}_{\text{ts}} = [\hat{\mathbf{S}}_{\text{ts}1} : \hat{\mathbf{S}}_{\text{ts}I}]$. The values of the outputs of each DNN in the second stage are between zero and one. Then each output frame and its corresponding gain $\alpha_{n,i}$ in $\alpha_{\text{ts}i}$, $\forall i$ are used to build a mask as follows:

$$\mathbf{M}_{\text{ts}i} = \frac{\alpha_{\text{ts}i} \otimes \hat{\mathbf{S}}_{\text{ts}i}}{\sum_{i=1}^I \alpha_{\text{ts}i} \otimes \hat{\mathbf{S}}_{\text{ts}i}}, \quad \forall i \quad (14)$$

where the division here is also done element-wise, and the multiplication $\alpha_{\text{ts}i} \otimes \hat{\mathbf{S}}_{\text{ts}i}$ means that each frame in $\hat{\mathbf{S}}_{\text{ts}i}$ is multiplied (scaled) by its corresponding gain entry in $\alpha_{\text{ts}i}$. The scaling by $\alpha_{n,i}$ here helps in using the DNNs in the second stage with bounded outputs between zero and one without the need to train the DNNs in the second stage over all possible scales of the source signals. Each $\alpha_{n,i}$ in $\alpha_{\text{ts}i}$ is considered as an estimate for the scale of its corresponding frame n in source i . The final enhanced estimate of the magnitude spectrogram of each source is computed as

$$\hat{\mathbf{S}}_i = \mathbf{M}_{\text{ts}i} \odot \mathbf{Y}. \quad (15)$$

The time domain estimates for source $\hat{s}_i(t)$ is computed using inverse STFT of $\hat{\mathbf{S}}_i$ and the phase angle of the STFT of the mixed signal.

C. NMF for separation and DNNs for enhancement

Here we show how to use DNNs to enhance the sources separated by NMF. NMF is used to separate the mixed signal in the first stage as shown in Section III-A and DNNs are used to enhance the separated sources in the second stage. The sources separated by NMF in Eq. (6) are considered to be distorted and need to be enhanced [16]. The procedures for enhancing the separated sources here are the same as in Sections IV-A and IV-B. In this case, the DNNs in the second stage (similar to DNN-B₁ to DNN-B_I and DNN-C in the case of using DNN-A for separation) are trained using sources separated by NMF. Also, NMF in the separation stage and the DNNs in the enhancement stage are trained on different training datasets. We denote NMF that is used for separation as NMF-A and the DNNs that are used to enhance the sources separated by NMF as NMF-DNN-B₁ to NMF-DNN-B_I for the first method of enhancement and NMF-DNN-C for the second method of enhancement.

D. DNN for separation and NMF for enhancement

In [15], NMF was used to enhance the quality of speech signals separated from different background noise. To compare the performance of using DNNs for enhancement with using NMF for enhancement, we also use NMF to enhance the separated sources. In this work, we slightly modify the work in [15] to enhance all the separated sources in a similar manner to the way the DNNs are used for enhancement here. The clean sources in the second training dataset $\mathbf{S}_{\text{tri}}^{(2)}$, $\forall i$ are used to train basis vectors for the sources as in Eq. (4) using the update rules in Eq. (3). NMF is then used to enhance the separated sources by projecting the separated spectrograms $\tilde{\mathbf{S}}_{\text{ts}i}$ of each source i in Eq. (13) into the trained basis matrices as follows:

$$\tilde{\mathbf{S}}_{\text{ts}i} \approx \mathbf{B}_{\text{tri}}^{(2)} \mathbf{G}_{\text{ts}i}, \quad \forall i \quad (16)$$

where $\mathbf{B}_{\text{tri}}^{(2)}$ is the trained basis matrix for the second set of the training data of source i . The gain update rule in Eq. (3) is used to find solutions for $\mathbf{G}_{\text{ts}i}$, $\forall i$. The decomposition results are then used to build a mask and find the final estimate for each source similar to Eq. (6), which is similar to the solutions of the DNNs in Eqs. (14) and (15). In the case of using NMF for enhancement, we do not need to worry about calculating the scale of each source ($\alpha_{\text{ts}i}$ in Eq. (14)) because this is already calculated implicitly in the gain matrices $\mathbf{G}_{\text{ts}i}$, $\forall i$ in Eq. (16). We denote NMF that is used for enhancement as NMF-B.

E. NMF for separation and enhancement

For completeness, we study the case of using NMF-A for separation followed by NMF-B for enhancement. This means, the sources separated by NMF-A in Eq. (6) are then enhanced using NMF-B similarly to Eq. (16). Note that the basis matrices for NMF-A are trained using the first set of the training data and the basis matrices for NMF-B are trained on the second set of the training data.

V. EXPERIMENTS AND DISCUSSION

We applied our proposed two stage SCASS approaches to separate vocal and music sources from various songs from the dataset of SiSEC-2015-MUS-task [33]. The dataset has 100 stereo songs for different genders of singers and different type of music instruments. The average length of each song is around four minutes. Some of the given sources in this dataset are mono (single channel) and other are stereo (two channels) recordings. To use this data for SCASS, we converted the songs from stereo to mono by taking the average between the two channels for each song. We used our proposed algorithms to separate each mono song into mono vocal and music (accompaniment) signals. The accompaniment signals tend to have higher energy than the vocal signals in most of the songs in this dataset [33]. The data was sampled at a 44.1kHz sampling rate. Magnitude spectrograms for the data were calculated using the STFT: a Hanning window with length 2048 samples and a shift of 512 samples were used and the FFT was taken at 2048 points, the first 1025 FFT points only were used as features for the data.

The first 35 songs from the dataset were used to train NMF-A and DNN-A models for separation. Since the number of sources here is two ($I = 2$), DNN-A was trained to predict a single mask M_{vocal} and the second mask is computed as $1 - M_{\text{vocal}}$, where 1 is a matrix of ones. The models for the NMF-A case are $B_{\text{tr1}}^{(1)}$ and $B_{\text{tr2}}^{(1)}$ as shown in Section III-A. For the case of using a DNN for separation, DNN-A was trained as shown in Section III-C.

The next 35 songs of the SiSEC dataset were used to train DNN-B₁, DNN-B₂, and DNN-C in the second stage for enhancement as shown in Section IV-A and also the basis matrices $B_{\text{tr1}}^{(2)}$ and $B_{\text{tr2}}^{(2)}$ as shown in Sections IV-D and IV-E. The next 10 songs were used as a development set to choose the parameters of the DNNs in the two stages. The remaining 20 songs were used for testing.

For the parameters of NMF, as in [15], we trained 80 basis vectors for each source in both separation and enhancement stages. For choosing the parameters of the DNNs, we have tried different values for the parameters of the DNNs (number of hidden layers, number of nodes in each layer, ..., etc) on the development set and we have obtained reasonable results using the following selections for the parameters of the DNNs in the separation and enhancement stages: in the separation stage (DNN-A), the number of nodes in each hidden layer was 1025 with three hidden layers. In the second stage (enhancement), for DNN-B₁ and DNN-B₂, the number of nodes in each hidden layer was 2050 with three hidden layers. For DNN-C, the size of the input and output layers is 2050 which is the length of the concatenation of the two sources 2×1025 , and three hidden layers with 4100 nodes in each hidden layer were used. As in [13], [34] the sigmoid function was used as an activation function at each node including the output nodes for all the DNNs. We also tried rectified linear activation function in the hidden layers but the difference with sigmoid was not significant. The values of the weights and bias of all the DNNs were initialized randomly. We used 200 epochs for backpropagation training for each DNN. Stochastic gradient

descent was used with batch size 100 frames and learning rate 0.1. We implemented our proposed algorithms using Theano [35]. For the regularization parameter λ in Eqs. (11) and (12), we tested with different values as we will show later.

Performance of the separation and enhancement algorithms was measured using the signal to distortion ratio (SDR), signal to interference ratio (SIR), and signal to artefact ratio (SAR) [36]. SIR indicates how much remaining signals from one source appear in the other separated sources (interference). SAR indicates the artefacts caused by the separation algorithm to the estimated separated sources. The SDR values evaluate the overall quality of the separated sources based on the artefacts and the interference between the separated sources. Achieving high SDR, SIR, and SAR values indicates good separation performance.

The average SDR, SIR, and SAR values of the separated vocal and accompaniment (acc) signals for the 30 test songs are reported in Figs. 3 to 8. For example, the SDR values shown in Fig. 3 are calculated as $(\text{SDR}_{\text{vocal}} + \text{SDR}_{\text{acc}})/2$ for each song and each model. Figs. 3 to 5, show the box-plot of the results of using NMF for source separation (NMF-A) and different approaches for enhancement. Figs. 6 to 8 show the results of using DNN-A for separation and different models for enhancement. In these figures, the definition of each model is shown in Table I. In this table: models N and D are for using NMF-A and DNN-A respectively for separation only without enhancement; the models start with letter ‘‘N’’ are for using NMF-A for separation; the models start with letter ‘‘D’’ are for using DNN-A for separation; models NN and DN are for using NMF-B for enhancement to enhance the separated sources from NMF-A and DNN-A respectively; the models with letter ‘‘B’’ are for enhancing the separated sources individually using DNNs (NMF-DNN-B_i or DNN-B_i); the models with letter ‘‘C’’ are for enhancing the separated sources together (NMF-DNN-C or DNN-C); the numbers 0, 2, and 4 in some of the model names are related to the values of the regularization parameter λ in Eqs. 11 and 12 as shown in Table I.

Models =>	N	NN	NB0	NB2	NB4	NC0	NC2	NC4
Separation	NMF-A							
Enhancement	-	NMF-B	NMF-DNN-B $\lambda = 0$ $\lambda = 0.2$ $\lambda = 0.4$			NMF-DNN-C $\lambda = 0$ $\lambda = 0.2$ $\lambda = 0.4$		
Models =>	D	DN	DB0	DB2	DB4	DC0	DC2	DC4
Separation	DNN-A							
Enhancement	-	NMF-B	DNN-B $\lambda = 0$ $\lambda = 0.2$ $\lambda = 0.4$			DNN-C $\lambda = 0$ $\lambda = 0.2$ $\lambda = 0.4$		

TABLE I
THE DEFINITION OF EACH MODEL IN FIGS.3 TO 8. THIS SHOWS WHICH MODEL IS USED FOR SEPARATION AND WHICH MODEL IS USED FOR ENHANCEMENT IN EACH CASE.

As can be seen from Figs. 3 to 5, using the enhancement stage (models NN to NC4) increases the values of SDR, SIR, and SAR of the separated sources. This means the enhancement stage has improved the quality of the separated signals. We can also see that using DNNs in models NB0 to NC4 (NMF-DNN-B₁, NMF-DNN-B₂, and NMF-DNN-C) for enhancement gives significantly better SDR and SIR results than model NN (NMF-B). To compare between the proposed

two methods of using DNNs for enhancement, we can look at the SDR, SIR, and SAR values for models NB0 and NC0 where $\lambda = 0$. Model NC0, where a single DNN (NMF-DNN-C) is used to enhance all the separated sources together gives better SDR and SAR values than model NB0 (where each separated source is enhanced individually using its own DNN). These may be due to most of the available information for each source is used in model NC0 while in model NB0 some of the information for each source that appears in the other separated sources is not considered in enhancing its own source. Increasing λ decreases the overlap possibility of the estimated sources in the spectrogram domain of the separated sources, which decreases the interference between the separated sources (high SIR values). Many audio sources usually occupy the same regions in the spectrogram domain with certain level, and strongly forcing the estimated sources to not totally overlap in the spectrogram domain (by increasing λ) means that each source will lose parts of its information in its estimated signal, which leads to increase the artefacts in the separated sources as shown in Fig. 5 where the SAR decreases with increasing λ . We stopped at $\lambda = 0.4$ since the SDR values in most cases start to decrease.

Fig. 3. The box-plot of the average SDR of the vocal and accompaniment signals of using NMF-A for separation and different models for enhancement. N = NMF-A, NN = NMF-A $\Rightarrow$ NMF-B, NB = NMF-A $\Rightarrow$ NMF-DNN-B, NC = NMF-A $\Rightarrow$ NMF-DNN-C.

Fig. 4. The box-plot of the average SIR of the vocal and accompaniment signals for using NMF-A for separation and different models for enhancement. N = NMF-A, NN = NMF-A $\Rightarrow$ NMF-B, NB = NMF-A $\Rightarrow$ NMF-DNN-B, NC = NMF-A $\Rightarrow$ NMF-DNN-C.

The data shown in Figs. 3 to 5 were analysed using non-parametric statistical methods [37] to determine the significance of the effects of enhancing the separated signals in the second stage. The decision to employ non-parametric tests

Fig. 5. The average SAR of the vocal and accompaniment signals for using NMF-A for separation and different models for enhancement. N = NMF-A, NN = NMF-A $\Rightarrow$ NMF-B, NB = NMF-A $\Rightarrow$ NMF-DNN-B, NC = NMF-A $\Rightarrow$ NMF-DNN-C.

was based on the distributions of the pairwise comparisons being non-normally distributed [37]. A pair of models are statistically significant difference (SSD) if $p < 0.05$, Wilcoxon signed-rank test [38] and Bonferroni corrected [39]. Table II, parts “NMF-SDR, NMF-SIR, and NMF-SAR” show the SSD between each pair of models in Figs. 3 to 5. In this table, we denote the models in the rows as better and SSD than the models in the columns by “+”, the cases with worse and SSD as “-”, and the cases without SSD as “0”. For example, model NC4 is SSD and better than most other models in SIR, and model NC0 is SSD and better than all other models in SAR. Note that, the term SSD should be interpreted in the statistical sense - so, we are not claiming that the magnitude of the difference is “significant” (loosely meaning: particularly big) but, rather, we are claiming that the difference is “statistically significant” (meaning reliable, independent of the actual magnitude of the difference). As can be seen from Table II and Figs. 3 to 5, model N is SSD and worse than all other models for SDR, SIR, and SAR, which means there is improvements due to using the second stage of enhancement compared to using NMF-A only for separation without enhancement (model N). We can also see that model NC0 is SSD and better than model NB0 in SDR and SAR and not SSD in SIR, which means that using a single DNN (NMF-DNN-C) to enhance all the separated sources together is better than enhancing each separated source individually using its own DNN (NMF-DNN-B₁ and NMF-DNN-B₂). Also, we can see improvements in SDR, SIR and most SAR values between the proposed enhancement methods using DNNs (models NB0 to NC4) compared to the enhancement method in [15] using NMF-B (model NN). This means the proposed enhancement methods using DNNs are SSD and better than using NMF for enhancement. The regularization parameter λ in models NB0 to NC4 has significant impact on the results, and can be used as a trade-off parameter between achieving high SIR values verses SAR and vice versa.

For the case of using DNN-A for separation in Figs 6 to 8, comparing model D in Figs. 6 to 8 with model N in Figs. 3 to 5, we can see the difference between using NMF-A (model N) and DNN-A (model D) for source separation. DNN works much better than NMF for source separation. The data

[illegible]

TABLE II

THE SIGNIFICANT DIFFERENCES STATISTICALLY (SSD) BETWEEN EACH PAIR OF MODELS IN FIGS. 3 TO 8. “+” OR “-” IN EACH CELL AT A CERTAIN ROW AND COLUMN MEANS THAT THE MODEL IN THIS ROW IS SSD AND BETTER OR WORSE RESPECTIVELY THAN THE MODEL IN THIS COLUMN. “0” MEANS NO EVIDENCE FOR SSD BETWEEN THE MODELS. NMF-SDR/SIR/SAR MEANS USING NMF-A FOR SEPARATION, AND DNN-SDR/SIR/SAR MEANS USING DNN-A FOR SEPARATION

Fig. 6. The box-plot of the average SDR of the vocal and accompaniment signals of using the DNN-A for separation and different models for enhancement. D = DNN-A, DN = DNN-A $\Rightarrow$ NMF-B, DB = DNN-A $\Rightarrow$ DNN-B, DC = DNN-A $\Rightarrow$ DNN-C.

Fig. 8. The box-plot of the average SAR of the vocal and accompaniment signals for using DNN-A for separation and different models for enhancement. D = DNN-A, DN = DNN-A $\Rightarrow$ NMF-B, DB = DNN-A $\Rightarrow$ DNN-B, DC = DNN-A $\Rightarrow$ DNN-C.

Fig. 7. The box-plot of the average SIR of the vocal and accompaniment signals for using DNN-A for separation and different models for enhancement. D = DNN-A, DN = DNN-A $\Rightarrow$ NMF-B, DB = DNN-A $\Rightarrow$ DNN-B, DC = DNN-A $\Rightarrow$ DNN-C.

shown in Figs. 6 to 8 were also analysed using the same non-parametric statistical methods as mentioned before.

Table II, parts “DNN-SDR, DNN-SIR, and DNN-SAR” show the SSD between each pair of models in Figs. 6 to 8. From this table and Figs. 6 to 8, we can see that all the used enhancement methods (models DN to DC4) improved the SDR and SIR results compared to using separation only (model D). For SAR, we can see that using one model to enhance all the separated sources (DC0) is SSD and better than all other models. Model DC4 is SSD and better than all other models in SIR. All the other models (DB0, DB2, DC2, and DN)

Fig. 9. The box-plot of SDR, SIR, and SAR of different methods of separation and enhancement. “Big” denotes using one big DNN with the number of layers and hidden units equal to the sum of the layers and hidden units of DNN-A and DNN-C networks and it was also trained using the first 70 songs. “Mask2” denotes using DNN-A for separation in the first stage and the neural network in the second stage is used to estimate spectral mask instead of estimating the sources directly. “DC0” denotes using DNN-A for separation and DNN-C with $\lambda = 0$ for enhancement as shown in Figs. 6 to 8.

except DC4 and DB4 are not worse than model D for SAR. Comparing between the two proposed enhancement methods using either DNN-B₁, DNN-B₂ or DNN-C (DB0 and DC0), we can see that there is no SSD between the two proposed enhancement methods in SDR and SIR values, but model DC0 is SSD and better than model DB0 in the SAR value.

Fig. 10. The box-plot of the SDR values of the mixed signal (denoted as “MIX” in the figure), the separated vocal, bass, drum, and other instrument sources using only DNN for separation without enhancement (denoted as “D”) and separation followed by DNN-C for enhancement with $\lambda = 0$ (denoted as “DC0”).

Fig. 11. The box-plot of the SIR values of the mixed signal (denoted as “MIX” in the figure), the separated vocal, bass, drum, and other instrument sources using only DNN for separation without enhancement (denoted as “D”) and separation followed by DNN-C for enhancement with $\lambda = 0$ (denoted as “DC0”).

In the following sections we show different cases for the second stage of enhancement. We will consider the method DC0 in previous sections as our ground truth of one of the proposed enhancement methods in this work because it gives reasonable SDR, SIR and SAR values.

First, instead of using DNN-A for separation and DNN-C for enhancement, we studied the case of using a big DNN that has a number of layers and hidden units equal to the sum of the numbers of layers and hidden units in DNN-A and DNN-C for separation. The big DNN in this case was trained over the first 70 songs of the dataset, which means the size of the big DNN and the amount of training data used for training it is equivalent to the sum of the training data and DNN-A and DNN-C parameters in the two stage case. To train the big DNN efficiently, we initialized the parameters of each hidden layer with the parameters of the corresponding hidden layer from DNN-A and DNN-C. The big DNN was then trained to predict masks in its output layer. Second, instead of using DNN-A to predict masks and DNN-C to predict the sources in its output layer, we studied the case where the two DNNs predict masks in their output layers. Fig. 9 shows a comparison between using a big DNN (denoted as “big” in Fig. 9) for separation, using two DNNs for separation and enhancement while the outputs of both DNNs are masks (denoted as “mask2” in Fig. 9), and the case of using DNN-A to estimate masks for

Fig. 12. The box-plot of the SAR values of the separated vocal, bass, drum, and other instrument sources using only DNN for separation without enhancement (denoted as “D”) and separation followed by DNN-C for enhancement with $\lambda = 0$ (denoted as “DC0”).

separation followed by DNN-C that estimates sources in the outputs for enhancements with $\lambda = 0$ (denoted as “DC0” in Fig. 9). As can be seen from Fig. 9, DC0 achieves slight improvements in SDR and significant improvements in SAR values than the other methods (big and Mask2).

We also studied the case of using the proposed source separation and enhancement approaches to separate the four sources: vocals, bass, drums, and other instruments from each song in the dataset. In this case we only show the case of using DNN-A for separation and DNN-C with $\lambda = 0$ for enhancement. Here we trained the DNN in the first stage to estimate four masks for the four sources, and the DNN in the second stage was trained to enhance the four separated sources together with $\lambda = 0$. Figs. 10 to 12 show the SDR, SIR, and SAR values respectively for the separated vocals, bass, drums, and other instruments from the mixed signal. The SDR and SIR values of the mixed signal with respect to each source are also shown in Figs. 10 and 11 as “MIX” (SAR for the mix is almost infinity). The results for using the DNN for separation without enhancement is denoted as “D” and using a DNN for separation followed by DNN-C with $\lambda = 0$ for enhancement is denoted as “DC0” in Figs. 10 to 12. As can be seen in Figs. 10 and 11, using DNN-A followed by DNN-C gives better results than using DNN-A only. The difference in the SDR and SIR values of each pair of models in Figs. 10 and 11 is SSD. The difference in SAR values of each pair of models in Fig. 12 is SSD except for the SAR between models D and DC0 for the bass.

From the shown results we can conclude that using DNNs as a second stage of enhancement improves the quality of the separated sources by decreasing the interference (achieving high SIR values) between the separated sources and the distortions (achieving high SDR values). Using discriminative learning with $\lambda \neq 0$ in Eqs. (11) to (12) to train the DNNs to maximize the differences between the separated sources can further decrease the interference between the separated sources and the distortions. Using a single DNN to enhance all the separated sources together gives slightly better results than enhancing each source separately. Using DNNs for enhancement works much better than the proposed work in [15] which uses NMF to enhance the separated sources. Training

the DNNs in the two stages over different representations for the training data (masks in the first stage and sources in the second stage) works better than training the DNNs in the two stages using masks only.

VI. CONCLUSION

In this work, we proposed two stage source separation approaches using DNNs to enhance the separated sources. We applied the proposed enhancement approaches on signals that were separated by either NMF or DNN. Two new methods of using DNNs to enhance the separated sources were proposed in this work. The first method was to enhance the separated signal for each source individually using its own trained DNN. The second method of enhancement was to use a single DNN to enhance all the separated sources together. For both proposed methods of enhancement, a discriminative training was used to train the DNNs in the enhancement stage. The experimental results show that the proposed enhancement methods using DNNs achieve separated sources with low distortion and interference. We also compared our proposed enhancement methods using DNNs with using NMF for enhancement. Our experimental results show that using DNNs for enhancement works better than using NMF. In our future work, we will investigate using more than two stages to further enhance the quality of the separated sources.

REFERENCES

- [1] T. Virtanen, "Monaural sound source separation by non-negative matrix factorization with temporal continuity and sparseness criteria," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 15, pp. 1066–1074, Mar. 2007.
- [2] P. Smaragdis, "Convolutional speech bases and their application to supervised speech separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 1, 2007.
- [3] E. M. Grais and H. Erdogan, "Single channel speech-music separation using matching pursuit and spectral masks," in *IEEE Conference on Signal Processing and Communications Applications (SIU)*, pp. 323–326.
- [4] P. Smaragdis, M. Shashanka, and B. Raj, "A sparse non-parametric approach for single channel separation of known sounds," in *Neural Information Processing Systems, Vancouver, BC, Canada*, Dec. 2009.
- [5] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," *Advances in NIPS*, vol. 13, pp. 556–562, 2001.
- [6] E. M. Grais and H. Erdogan, "Spectro-temporal post-smoothing in NMF based single-channel source separation," in *Proc. EUSIPCO*, 2012.
- [7] —, "Source separation using regularized NMF with MMSE estimates under GMM priors with online learning for the uncertainties," *Digital Signal Processing*, vol. 29, pp. 20–34, 2014.
- [8] F. Weninger, J. R. Hershey, J. L. Roux, and B. Schuller, "Discriminatively trained recurrent neural networks for single-channel speech separation," in *Proc. GlobalSIP*, 2014, pp. 577–581.
- [9] E. M. Grais, G. Roma, A. J. R. Simpson, and M. D. Plumbley, "Single channel audio source separation using deep neural network ensembles," in *Proc. 140th Audio Engineering Society Convention*, 2016.
- [10] P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Singing-Voice separation from monaural recordings using deep recurrent neural networks," in *Proc. ISMIR*, 2014, pp. 477–482.
- [11] —, "Deep learning for monaural speech separation," in *Proc. ICASSP*, 2014, pp. 1562–1566.
- [12] E. M. Grais, M. U. Sen, and H. Erdogan, "Deep neural networks for single channel source separation," in *Proc. ICASSP*, 2014, pp. 3734–3738.
- [13] H. Erdogan, J. Hershey, S. Watanabe, and J. L. Roux, "Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks," in *Proc. ICASSP*, 2015, pp. 708–712.
- [14] X.-L. Zhang and D. Wang, "Deep ensemble learning for monaural speech separation," *IEEE/ACM Trans. on audio, speech, and language processing*, vol. 24, no. 5, pp. 967–977, 2016.
- [15] D. Williamson, Y. Wang, and D. Wang, "A two-stage approach for improving the perceptual quality of separated speech," in *Proc. ICASSP*, 2014, pp. 7034–7038.
- [16] E. M. Grais and H. Erdogan, "Spectro-temporal post-enhancement using MMSE estimation in NMF based single-channel source separation," in *Proc. InterSpeech*, 2013.
- [17] A. Narayanan and D. Wang, "Investigation of speech separation as a front-end for noise robust speech recognition," *IEEE/ACM Trans. on audio, speech, and language processing*, vol. 22, no. 4, pp. 826–835, 2014.
- [18] E. W. Healy, S. E. Yoho, Y. Wang, and D. Wang, "An algorithm to improve speech recognition in noise for hearing-impaired listeners," *Journal of the Acoustical Society of America*, vol. 134, no. 4, pp. 3029–3038, 2013.
- [19] D. S. Williamson, Y. Wang, and D. Wang, "Estimating nonnegative matrix model activations with deep neural networks to increase perceptual speech quality," *Journal of the Acoustical Society of America*, vol. 138, no. 3, pp. 1399–1407, 2015.
- [20] —, "Reconstruction techniques for improving the perceptual quality of binary masked speech," *Journal of the Acoustical Society of America*, vol. 136, no. 2, pp. 892–902, 2014.
- [21] Y. Wang and D. Wang, "A structure-preserving training target for supervised speech separation," in *Proc. ICASSP*, 2014, pp. 6148–6152.
- [22] X.-L. Zhang and J. Wu, "Deep belief networks based voice activity detection," *IEEE Trans. on audio, speech, and language processing*, vol. 21, no. 4, pp. 697–710, 2013.
- [23] X.-L. Zhang and D. Wang, "Boosting contextual information for deep neural network based voice activity detection," *IEEE/ACM Trans. on audio, speech, and language processing*, vol. 24, no. 2, pp. 252–264, 2016.
- [24] E. M. Grais, G. Roma, A. J. R. Simpson, and M. D. Plumbley, "Combining mask estimates for single channel audio source separation using deep neural networks," in *Proc. InterSpeech*, 2016.
- [25] H. P. S., M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Joint optimization of masks and deep recurrent neural networks for monaural source separation," *IEEE/ACM Trans. Audio Speech and Language Processing*, vol. 23, no. 12, pp. 2136 – 2147, 2015.
- [26] Y. Wang, A. Narayanan, and D. Wang, "On training targets for supervised speech separation," *IEEE/ACM Trans. on audio, speech, and language processing*, vol. 22, no. 12, pp. 1849–1858, 2014.
- [27] E. M. Grais and H. Erdogan, "Single channel speech music separation using nonnegative matrix factorization with sliding window and spectral masks," in *Proc. InterSpeech*, 2011.
- [28] M. Helen and T. Virtanen, "Separation of drums from polyphonic music using nonnegative matrix factorization and support vector machine," in *EUSIPCO*, 2005.
- [29] G. Hinton, L. Deng, D. Yu, G. Dahl, A. R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modelling in speech recognition," *Signal Processing Magazine*, 2012.
- [30] Y. Bengio, "Learning deep architectures for AI," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.
- [31] A. A. Nugraha, A. Liutkus, and E. Vincent, "Multichannel audio source separation with deep neural networks," *IEEE/ACM Trans. on audio, speech, and language processing*, vol. 24, no. 9, pp. 1652–1664, 2016.
- [32] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "A regression approach to speech enhancement based on deep neural networks," *IEEE/ACM Trans. on audio, speech, and language processing*, vol. 23, no. 1, pp. 7–19, 2015.
- [33] N. Ono, Z. Rafii, D. Kitamura, N. Ito, and A. Liutkus, "The 2015 signal separation evaluation campaign," in *Proc. LVA/ICA*, 2015, pp. 387–395.
- [34] X. Xiao, S. Watanabe, H. Erdogan, L. Lu, J. Hershey, M. L. Seltzer, G. Chen, Y. Zhang, M. Mandel, and D. Yu, "Deep beamforming networks for multi-channel speech recognition," in *Proc. ICASSP*, 2016, pp. 5745–5749.
- [35] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio, "Theano: a CPU and GPU math expression compiler," in *Proc. the Python for Scientific Computing Conference (SciPy)*, 2010.
- [36] E. Vincent, R. Gribonval, and C. Fevotte, "Performance measurement in blind audio source separation," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 14, no. 4, pp. 1462–69, Jul. 2006.
- [37] A. J. R. Simpson, G. Roma, E. M. Grais, R. Mason, C. Hummersone, A. Liutkus, and M. D. Plumbley, "Evaluation of audio source separation

models using hypothesis-driven non-parametric statistical methods,” in *Proc. Eusipco*, 2016.

- [38] F. Wilcoxon, “Individual comparisons by ranking methods,” *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945.
- [39] H. Y. and A. C. Tamhane, *Multiple Comparison Procedures*. John Wiley and Sons, 1987.

Emad M. Grais is a research fellow at the Centre for Vision, Speech and Signal Processing, University of Surrey, U.K. Previously he was a postdoctoral fellow at Inria in Nancy-France and Sabanci University in Istanbul-Turkey. He received his B.Sc. degree in Communication and Electronics Engineering from Helwan University, Egypt, in 2004, and Ph.D. degree in Electronics Engineering from Sabanci University, Turkey in 2013. His research interests are single channel audio source separation, speech enhancement, and using machine learning techniques such

as deep learning and matrix factorization for audio signal processing.

Gerard Roma is a postdoctoral fellow at the school of Literature, Media and Communication of the Georgia Institute of Technology (GeorgiaTech, US). Before this, he was a research fellow at the Centre for Vision, Speech and Signal Processing (CVSSP), University of Surrey (UK). He received his degree in Philosophy from Universitat Autònoma de Barcelona (UAB) in 1997. After working in industry as a software developer, he received his M.Sc. (2008) and Ph.D. (2015) in Information and Communication Technologies from Universitat Pompeu Fabra

(UPF) in Barcelona, Spain. His research interests include sound analysis and retrieval, audio source separation and interactive music systems.

Andrew J. R. Simpson Formerly a Research Fellow at the University of Surrey, Dr Andrew J.R. Simpson has published widely on deep neural networks and now runs an independent AI lab.

Mark D. Plumbley (S'88-M'90-SM'12-F'15) received the B.A.(Hons.) degree in electrical sciences and the Ph.D. degree in neural networks from University of Cambridge, Cambridge, U.K., in 1984 and 1991, respectively. From 1991 to 2001, he was a Lecturer with Kings College London, London, U.K., before moving to Queen Mary University of London, London, in 2002, later becoming Director of the Centre for Digital Music. In 2015, he joined the Centre for Vision, Speech and Signal Processing, University of Surrey, Guildford, U.K., as Professor

of Signal Processing. His research interests include automatic analysis of sounds and music, including acoustic scene analysis, audio source separation, and automatic music transcription, using methods such as deep learning, matrix factorization, and sparse representations. He is a Member of the IEEE Signal Processing Society Technical Committee on Signal Processing Theory and Methods.