*INTERSPEECH 2016*
September 8–12, 2016, San Francisco, USA

# Combining Mask Estimates for Single Channel Audio Source Separation using Deep Neural Networks

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

Centre for Vision, Speech and Signal Processing, University of Surrey, Guildford, UK.
{grais, g.roma, andrew.simpson, m.plumbley}@surrey.ac.uk

## Abstract

Deep neural networks (DNNs) are usually used for single channel source separation to predict either soft or binary time frequency masks. The masks are used to separate the sources from the mixed signal. Binary masks produce separated sources with more distortion and less interference than soft masks. In this paper, we propose to use another DNN to combine the estimates of binary and soft masks to achieve the advantages and avoid the disadvantages of using each mask individually. We aim to achieve separated sources with low distortion and low interference between each other. Our experimental results show that combining the estimates of binary and soft masks using DNN achieves lower distortion than using each estimate individually and achieves as low interference as the binary mask.

**Index Terms**: Combining estimates, deep neural networks, single channel source separation, neural network ensembles, deep learning

## 1. Introduction

Deep neural networks (DNNs) have been used recently to tackle the single channel audio source separation (SCSS) problem [1, 2, 3, 4]. The input of the DNN is the mixed signal and the output is a time-frequency mask. The mask is then used to separate the sources by scaling the mixed signal according to the contribution of each source in the mixed signal. DNNs can be trained to predict different types of masks [5]. The most used masks are binary and soft masks [3, 5, 6]. The DNN that predicts a binary mask achieves separated sources with less interference than the DNN that predicts a soft mask. On the other hand, the separated sources by a soft mask are less distorted than the separated sources using a binary mask [7, 8, 9].

The combination of many different predictors might yield better predictions than the best single predictor [10, 11]. There are several methods of combining different neural networks' predictions, where every neural network is trained individually [12, 13, 14]. This combination is done, for example by majority voting in classification problems or by weighted combinations in regression problems [11, 13, 15]. Finding the optimal weights for combining the predictors to achieve better prediction than using a single predictor is a challenging problem [11]. To do this, simple averaging of the different predictions [15] or solving a least squares problem to find the optimal weights are sometimes used [11, 14].

In this paper, we introduce a new technique for combining estimates using deep neural networks (DNNs) to combine the predictions of two different DNNs for the single channel source separation problem (SCSS). The first predictor (DNN1) is used to predict a binary mask from the mixed signal and the second predictor (DNN2) is used to predict a soft mask. The third DNN (DNN3) is used to find a good combination of the two predictors to achieve the advantages and avoid the disadvantages of each predictor individually. Binary masks achieve separation with lower interference between the separated sources than soft masks [7, 16], while soft masks achieve separated signals with lower distortion than binary masks [2, 3, 7]. The main goal of combining the predictions of the two DNNs is to achieve separation for the sources with both low interference and low distortion. The reason for using a single DNN for combining all the predicted sources from soft and binary masks is to include the missing data from one separated source that appears in the estimates of the other separated sources in the combination process. Combining all the separated sources using one DNN increases the possibility of remixing the separated sources. To decrease this possibility, we train DNN3 discriminativly [3, 17]. To avoid the final estimates of the sources from being biased estimates, the data set that is used to train the third DNN is different than the data set that is used to train each predictor [11]. Note that, if DNN3 has only linear activation functions, training DNN3 (finding the weights) may be seen as finding the best weighted linear combination of the predictors that can approximate the target sources. So, using DNN3 with many hidden layers and nonlinear activation functions gives a better chance of forming non-linear complex combinations of the predictors than just using the average or weighted linear combinations of their predictions. DNNs are usually good at modelling the complex structure of the data in many machine learning tasks [18, 19].

The contributions of this paper are: using a DNN to find the best combination of different DNNs' estimates (DNNs ensembles using another DNN). To the best of our knowledge, this is the first time the neural network "ensembles" process has been done by another DNN to find the best combinations of the different predictors. The DNN for combining the predictors (DNN3) is trained discriminatively to decrease the interference between the final estimates of the sources.

This paper is organized as follows: In Section 2 a mathematical formulation of the SCSS problem is given. Section 3 shows using DNNs to predict binary and soft masks to separate the sources from the mixed signal. In Sections 4 and 5, we present our proposed approach of discriminatively combining the predicted sources using DNN3. The experimental results and the conclusion of this paper are presented in Sections 6 and 7.

## 2. Problem formulation of SCSS

The main concern of the SCSS problem is to separate one or more sources from their single mixture. For simplicity, we assume the number of sources to be separated in this paper is two. Let us assume that the mixed signal $y(t)$ is a mixture of two sources $s_1(t)$ and $s_2(t)$ as $y(t) = s_1(t) + s_2(t)$. This



http://dx.doi.org/10.21437/Interspeech.2016-216

problem is usually solved in the short time Fourier transform (STFT) domain [20, 21], where $Y(n,f)$ is the STFT of $y(t)$, $n$ and $f$ represent the frame index and the frequency index respectively. This problem can be formulated as $Y(n,f) = S_1(n,f) + S_2(n,f)$, where $S_1(n,f)$ and $S_2(n,f)$ are the unknown STFTs of the sources in the observed mixed signal $Y(n,f)$. The estimates for the sources $\hat{S}_1(n,f)$ and $\hat{S}_2(n,f)$ are usually found by predicting a time-frequency mask $M(n,f)$ that scales the mixed signal according to the contribution of each source in the mixed signal as in [1, 5, 16] as follows:

$$\begin{aligned} \hat{S}_1(n,f) &= M(n,f)Y(n,f) \\ \hat{S}_2(n,f) &= (1 - M(n,f))\, Y(n,f) \end{aligned} \tag{1}$$

where the mask $M(n,f)$ takes real values either between zero and one (soft mask), or takes only zero and one values as a binary mask. The main goal here is to predict a mask $M(n,f)$ that achieves good separation for the sources. In this framework, the magnitude spectrum of the measured signal is usually approximated as the sum of the magnitude spectra of the estimated sources [20, 22] as follows: $|Y(n,f)| \approx \left|\hat{S}_1(n,f)\right| + \left|\hat{S}_2(n,f)\right|$. The magnitude spectrograms and the mask can be written in matrix form as $\mathbf{Y}(n,f)$, $\mathbf{S}(n,f)$, and $\mathbf{M}(n,f)$, where each column $n$ represents a spectral frame and each row $f$ represents a frequency index.

## 3. Training DNNs to predict binary and soft masks

Two DNNs are trained to predict binary and soft masks from the mixed signal using the same set of the training data. Given the magnitude spectrogram of the first set of the training mixed signals $\mathbf{X}_{\text{tr}}^{(1)}$, which is the sum of the sources $\mathbf{S}_{\text{tr1}}^{(1)}$ and $\mathbf{S}_{\text{tr2}}^{(1)}$, the DNNs are trained to minimize the following cost function:

$$C_{\mathbf{M}} = \sum_{n,f} \left(\mathbf{Z}_{\mathbf{M}}\,(n,f) - \mathbf{M}_{\mathbf{r}}\,(n,f)\right)^2 \tag{2}$$

where $\mathbf{Z}_{\mathbf{M}}$ is the output of the last layer of the DNNs and $\mathbf{M}_{\mathbf{r}}$ is the reference mask. For DNN1, $\mathbf{M}_{\mathbf{r}}$ is a binary mask ($\mathbf{M}_{\mathbf{b}_1}$) and for DNN2, $\mathbf{M}_{\mathbf{r}}$ is a soft mask ($\mathbf{M}_{\mathbf{s}_1}$). The binary mask is defined as follows:

$$\mathbf{M}_{\mathbf{b}_1}(n,f) = \begin{cases} 1 & \text{if } \mathbf{S}_{\text{tr1}}^{(1)}(n,f) \geq \mathbf{S}_{\text{tr2}}^{(1)}(n,f) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

and the soft mask is defined as:

$$\mathbf{M}_{\mathbf{s}_1}(n,f) = \frac{\mathbf{S}_{\text{tr1}}^{(1)}(n,f)}{\mathbf{S}_{\text{tr1}}^{(1)}(n,f) + \mathbf{S}_{\text{tr2}}^{(1)}(n,f)} \tag{4}$$

where the subscript tr$i$ in $\mathbf{S}_{\text{tr}i}^{(1)}$ indicates the training data of source number $i$ and the superscript 1 indicates the first set of the training data. In this training stage, the clean sources $\mathbf{S}_{\text{tr1}}^{(1)}$ and $\mathbf{S}_{\text{tr2}}^{(1)}$ are available.

## 4. Training DNN3 to combine the predictions of the binary and soft masks

For estimating the optimal combination of different predictors (DNN1 and DNN2), it is usually preferred to learn the parameters of the combining-model (DNN3) using different set of training data than the used data for training each predictor individually [11, 13]. This avoids the final estimates from being biased estimates [11]. Thus, we use different set of training data to train DNN3. The magnitude spectrogram $\mathbf{X}_{\text{tr}}^{(2)}$ of the mixed signal of the second set of the training data is applied to both trained DNNs to predict binary and soft masks as $\mathbf{Z}_{\mathbf{b}}$ and $\mathbf{Z}_{\mathbf{s}}$ respectively. The estimates of the sources using the binary mask (DNN1) are computed as follows:

$$\hat{\mathbf{S}}_{\mathbf{b}_{\text{tr1}}}^{(2)} = \mathbf{X}_{\text{tr}}^{(2)} \odot \mathbf{Z}_{\mathbf{b}} \quad \text{and} \quad \hat{\mathbf{S}}_{\mathbf{b}_{\text{tr2}}}^{(2)} = \mathbf{X}_{\text{tr}}^{(2)} \odot (\mathbf{1} - \mathbf{Z}_{\mathbf{b}}) \tag{5}$$

where $\odot$ denotes an element-wise multiplication and $\mathbf{1}$ is a matrix of ones. Also the estimates of the sources using the soft mask (the output of DNN2) are computed as follows:

$$\hat{\mathbf{S}}_{\mathbf{s}_{\text{tr1}}}^{(2)} = \mathbf{X}_{\text{tr}}^{(2)} \odot \mathbf{Z}_{\mathbf{s}} \quad \text{and} \quad \hat{\mathbf{S}}_{\mathbf{s}_{\text{tr2}}}^{(2)} = \mathbf{X}_{\text{tr}}^{(2)} \odot (\mathbf{1} - \mathbf{Z}_{\mathbf{s}}). \tag{6}$$

Here each source is estimated twice from DNN1 and DNN2. The interference between the initial estimates of the separated sources using the binary mask is usually less than the interference between the separated sources using the soft mask. Also the distortion of the estimated sources of using the soft mask is usually less than the distortion of using the binary mask. We need to combine the estimates to achieve low distortion and low interference between the separated sources. The idea here is to train DNN3 to combine the estimated sources to find the final estimates of the sources that are better than the estimates of DNN1 and DNN2 individually. All the four estimates in Eqs. (5) and (6) are concatenated and fed to the third DNN (DNN3) as shown in Fig.1. Given the reference sources for the second set of the training data $\mathbf{S}_{\text{tr1}}^{(2)}$ and $\mathbf{S}_{\text{tr2}}^{(2)}$, DNN3 is trained by minimizing the following cost function:

$$\begin{aligned} C_{\mathbf{c}} = &\sum_{n,f} \left(\mathbf{Q}\,(n,f) - \mathbf{V}\,(n,f)\right)^2 \\ &- \lambda_1 \sum_{n,f} \left(\mathbf{Q}_1\,(n,f) - \mathbf{S}_{\text{tr2}}^{(2)}\,(n,f)\right)^2 \\ &- \lambda_2 \sum_{n,f} \left(\mathbf{Q}_2\,(n,f) - \mathbf{S}_{\text{tr1}}^{(2)}\,(n,f)\right)^2 \end{aligned} \tag{7}$$

where $\lambda_1, \lambda_2$ are regularization parameters, $\mathbf{V}$ is the concatenation of the reference signals $\mathbf{V} = \left[\mathbf{S}_{\text{tr1}}^{(2)} : \mathbf{S}_{\text{tr2}}^{(2)}\right]$, $\mathbf{Q}$ is the actual output of DNN3 which is a concatenation of two outputs $\mathbf{Q} = [\mathbf{Q}_1 : \mathbf{Q}_2]$. The output $\mathbf{Q}_1$ is the set of the output nodes of DNN3 that correspond to the normalized reference output of the first source $\mathbf{S}_{\text{tr1}}^{(2)}$, while the output $\mathbf{Q}_2$ is for the second source $\mathbf{S}_{\text{tr2}}^{(2)}$. The normalization is done by dividing each frame by its Euclidean norm. This normalization is important since the activation function in the output layer is a sigmoid function that takes values between zero and one. The first term in the cost function in Eq. (7) minimizes the difference between the outputs of DNN3 and their corresponding reference signals. The second and third terms of the cost function maximize the dissimilarity/differences between the estimated DNN3 outputs of the different sources, which is considered as "discriminative learning". The cost function in Eq. (7) aims to prevent each set of the outputs of DNN3 from representing the other set. This helps in decreasing the possibility of remixing the separated sources in DNN3 and also achieves better separation for the estimated sources [3]. Note that, DNN1 and DNN2 are trained to predict masks in their output layers, while DNN3 is trained to predict normalized magnitude spectrograms of the sources in its output layer.

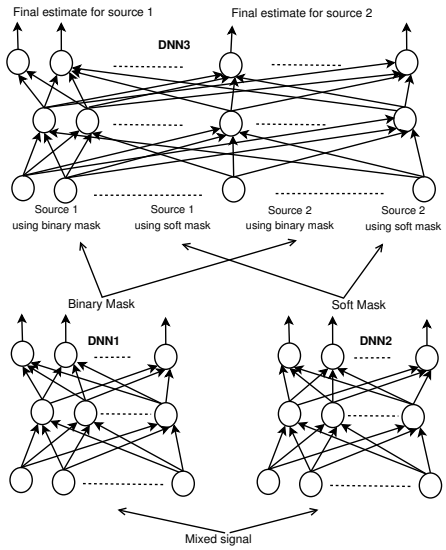


Figure 1: The overview of the proposed approach of using a single DNN to combine the predictions of two different DNNs for SCSS. The DNNs at the bottom (DNN1, DNN2) are used to predict binary and soft masks for separation. The DNN at the top (DNN3) is used to combine the predictions of DNN1 and DNN2.

## 5. Testing DNN1, DNN2, and DNN3

Given an observed mixed signal $y(t)$ of the test data that needs to be separated, the magnitude spectrogram $\mathbf{Y}$ is calculated. Then $\mathbf{Y}$ is fed to DNN1 and DNN2 to predict binary and soft masks $\mathbf{Z}_{\mathrm{b}_{\mathrm{ts1}}}$ and $\mathbf{Z}_{\mathrm{s}_{\mathrm{ts1}}}$ respectively. The masks $\mathbf{Z}_{\mathrm{b}_{\mathrm{ts1}}}$ and $\mathbf{Z}_{\mathrm{s}_{\mathrm{ts1}}}$ are then used to find initial estimates for the sources as $\tilde{\mathbf{S}}_{\mathrm{b}_{\mathrm{ts1}}}$ and $\tilde{\mathbf{S}}_{\mathrm{b}_{\mathrm{ts2}}}$ for the binary mask as follows:

$$\tilde{\mathbf{S}}_{\mathrm{b}_{\mathrm{ts1}}} = \mathbf{Z}_{\mathrm{b}_{\mathrm{ts1}}} \odot \mathbf{Y} \quad \text{and} \quad \tilde{\mathbf{S}}_{\mathrm{b}_{\mathrm{ts2}}} = \left(\mathbf{1} - \mathbf{Z}_{\mathrm{b}_{\mathrm{ts1}}}\right) \odot \mathbf{Y} \quad (8)$$

and $\tilde{\mathbf{S}}_{\mathrm{s}_{\mathrm{ts1}}}$ and $\tilde{\mathbf{S}}_{\mathrm{s}_{\mathrm{ts2}}}$ for the soft mask as follows:

$$\tilde{\mathbf{S}}_{\mathrm{s}_{\mathrm{ts1}}} = \mathbf{Z}_{\mathrm{s}_{\mathrm{ts1}}} \odot \mathbf{Y} \quad \text{and} \quad \tilde{\mathbf{S}}_{\mathrm{s}_{\mathrm{ts2}}} = \left(\mathbf{1} - \mathbf{Z}_{\mathrm{s}_{\mathrm{ts1}}}\right) \odot \mathbf{Y}. \quad (9)$$

The estimates for the sources are concatenated and fed to the third DNN as shown in Fig.1. The source signals can have any values, but the output nodes of DNN3 are composed of sigmoid activation functions that take values between zero and one. To ensure that we do not lose the scale information of the sources, the Euclidean norm (gain) of all frames in the spectrograms of the estimated sources $\tilde{\mathbf{S}}_{\mathrm{b}_{\mathrm{ts1}}}$, $\tilde{\mathbf{S}}_{\mathrm{b}_{\mathrm{ts2}}}$, $\tilde{\mathbf{S}}_{\mathrm{s}_{\mathrm{ts1}}}$ and $\tilde{\mathbf{S}}_{\mathrm{s}_{\mathrm{ts2}}}$ are computed as follows: for the binary mask case $\boldsymbol{\alpha}_{\mathrm{b}_{\mathrm{ts1}}} = \left[\alpha_{\mathrm{b}_{1,1}}, ..., \alpha_{\mathrm{b}_{N,1}}\right]$ for the first source and $\boldsymbol{\alpha}_{\mathrm{b}_{\mathrm{ts2}}} = \left[\alpha_{\mathrm{b}_{1,2}}, ..., \alpha_{\mathrm{b}_{N,2}}\right]$ for the second source, and for the soft mask case as $\boldsymbol{\alpha}_{\mathrm{s}_{\mathrm{ts1}}} = \left[\alpha_{\mathrm{s}_{1,1}}, ..., \alpha_{\mathrm{s}_{N,1}}\right]$ for the first source and $\boldsymbol{\alpha}_{\mathrm{s}_{\mathrm{ts2}}} = \left[\alpha_{\mathrm{s}_{1,2}}, ..., \alpha_{\mathrm{s}_{N,2}}\right]$ for the second source, where $N$ is the number of frames in each source. The final scale for each frame in each source is the average of the scales of the hard and soft mask estimates as $\boldsymbol{\alpha}_{\mathrm{ts1}} = [\alpha_{1,1}, ..., \alpha_{N,1}]$ and $\boldsymbol{\alpha}_{\mathrm{ts2}} = [\alpha_{1,2}, ..., \alpha_{N,2}]$, where $\alpha_{n,i} = (\alpha_{\mathrm{b}_{n,i}} + \alpha_{\mathrm{s}_{n,i}})/2$. Each $\alpha_{n,i}$ here is considered as an estimate of the scale of its corresponding frame $n$ in source $i$. The scales $\alpha_{n,i}$ are then saved to be used later. The four initial estimates in Eqs. (8) and (9) are concatenated and fed to DNN3 as shown in Fig.1. The output of DNN3 is $\hat{\mathbf{S}} = \left[\hat{\mathbf{S}}_{\mathrm{ts1}} : \hat{\mathbf{S}}_{\mathrm{ts2}}\right]$. The values of the outputs of DNN3 are between zero and one. The magnitude spectrograms and their scales are then used to build the final mask as follows:

$$\mathbf{M}(n, f) = \frac{\boldsymbol{\alpha}_{\mathrm{ts1}}(n)\hat{\mathbf{S}}_{\mathrm{ts1}}(n, f)}{\boldsymbol{\alpha}_{\mathrm{ts1}}(n)\hat{\mathbf{S}}_{\mathrm{ts1}}(n, f) + \boldsymbol{\alpha}_{\mathrm{ts2}}(n)\hat{\mathbf{S}}_{\mathrm{ts2}}(n, f)} \quad (10)$$

where the multiplication $\boldsymbol{\alpha}_{\mathrm{ts1}}(n)\hat{\mathbf{S}}_{\mathrm{ts1}}(n, f)$ means that every frame (vector) in $\hat{\mathbf{S}}_{\mathrm{ts1}}$ is multiplied (scaled) with its corresponding gain entry in $\boldsymbol{\alpha}_{\mathrm{ts1}}$. The scaling using $\alpha$s here helps in using DNN3 with bounded outputs between zero and one without the need to train DNN3 over all possible scales of the source signals. The final estimate of the magnitude spectrogram of each source is computed as

$$\hat{\mathbf{S}}_1 = \mathbf{M} \odot \mathbf{Y} \quad \text{and} \quad \hat{\mathbf{S}}_2 = (\mathbf{1} - \mathbf{M}) \odot \mathbf{Y} \quad (11)$$

where $\mathbf{M}$ is computed using Eq. (10). The time domain estimates for the source signals $\hat{s}_1(t)$ and $\hat{s}_2(t)$ are computed using the inverse STFT of $\hat{\mathbf{S}}_1$ and $\hat{\mathbf{S}}_2$ respectively with the phase angle of the STFT of the mixed signal.

## 6. Experiments and Discussion

We tested our proposed algorithm on a collection of speech (vocal) and music (accompaniment) signals from the SiSEC 2015 dataset [23]. The dataset has 100 songs. Each song represents a mixture of vocal, bass, drums, and other musical instruments. We used our proposed algorithm to separate each song into vocal and accompaniment signals. The first 35 songs were used to train DNN1 and DNN2 for separation as shown in Section 3. The second 35 songs were used to train DNN3 for combining the estimates of DNN1 and DNN2 as shown in Section 4. The remaining 30 songs were used for testing. The data was sampled at 44.1kHz sampling rate. The magnitude spectrograms for the data were calculated using the STFT: A Hanning window with 2048 points length and overlap interval of 512 was used and the FFT was taken at 2048 points, the first 1025 FFT points only were used since the conjugate of the remaining 1024 points are involved in the first points.

For DNN1 and DNN2, the number of nodes in each hidden layer was 1025 with three hidden layers. The length of the input for DNN3 is 4100, which is the length of the combination of the four estimates of DNN1 and DNN2 (4*1025) as shown in Fig.1. The number of nodes in the output layer of DNN3 is 2050, which is the length of the concatenation of the two separated sources (2*1025). For DNN3, we used three hidden layers with 4100 nodes in each hidden layer. Sigmoid nonlinearity was used at each node, including the output nodes for all DNNs in this paper. The parameters for 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 algorithm using Theano [24, 25]. For the regularization parameters $\lambda_1$ and $\lambda_2$ in Eq. (7), we tested with different values as $\lambda_1 = \lambda_2 = \lambda$ as shown in the Figures below.

We compared using DNN3 for combining the estimates of DNN1 and DNN2 with combining the estimates using the simple average of the estimates of DNN1 and DNN2.

Performance measurements of the proposed separation approach were done using the signal to distortion ratio (SDR) and the signal to interference ratio (SIR) [26]. The SDR values

are usually considered as the overall performance evaluation for any source separation approach and the SIR indicates the success of the approach in separating the mixed signals [26]. High SIR values mean low interference between the separated sources and high SDR values mean low distortion.

Figures 2 and 3 show the box-plots of the average SDR and SIR between the vocal and accompaniment signals over the 30 songs in the test set. To plot these figures, the average SDR and SIR values between the vocal and accompaniment for each song were calculated for each model. Model1 in the figures show the results of using the binary mask only (DNN1) for separating the mixed signal. Model2 shows the results of using only the soft mask (DNN2) for separation. Model3 shows the results of combining the estimates of DNN1 and DNN2 by taking the average between their estimates. Model4 to model6 show the results of combining the estimates of DNN1 and DNN2 using DNN3 with different values for the regularization parameters $\lambda_1 = \lambda_2 = \lambda$ in Eq. (7). The red lines in the box-plots represent the median over the results of the 30 songs. The

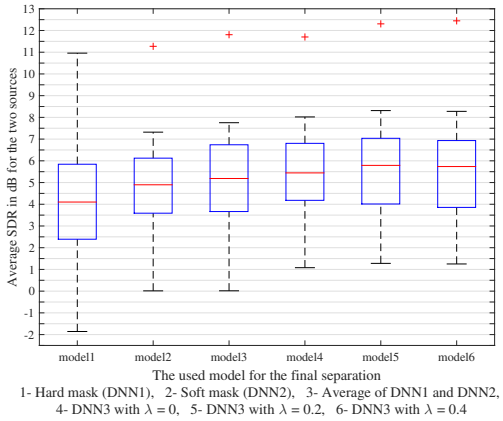


Figure 2: The average SDR of the vocal and accompaniment signals.

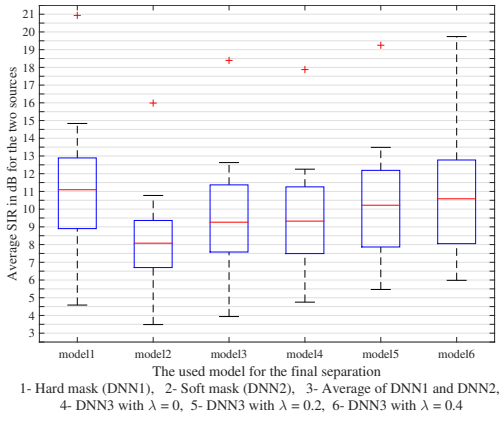


Figure 3: The average SIR of the vocal and accompaniment signals.

data shown in Figures 2 and 3 were also analysed using non-parametric statistical methods [27] to determine the significance of the differences between the shown models. For SDR in Fig.2, there is a main effect of model ($P < 0.01, df = 5, \chi^2 = 90.4$, Friedman test [28]). There are no significant differences between model2 and model3, model4 and model6, and model5 and model6 ($P > 0.05$, Wilcoxon signed-rank test [29], Bonferroni corrected [30]). There are significant differences between all the other pairs of models ($P < 0.01$, Wilcoxon signed-rank test, Bonferroni corrected). This means the following: using soft mask (model2) is significantly better than the binary mask (model1), combining the estimates of DNN1 and DNN2 by taking their average (model3) is significantly better than using the binary mask (model1) but not significantly better than the soft mask (model2), combining the estimates of DNN1 and DNN2 using DNN3 (model4, model5, and model6) achieves significant improvements over each estimate individually (model1 and model2) and also over the average of the estimates (model3). Model5 gives the highest median values (the red line inside the box-plot) compared to all other models.

For SIR in Fig.3, there is a main effect of model ($P < 0.01, df = 5, \chi^2 = 126.23$, Friedman test). There are no significant differences between model1 and model5, model1 and model6, and model3 and model4 ($P > 0.05$, Wilcoxon signed-rank test, Bonferroni corrected). This means that there is no evidence of significant differences in the separation performance between the binary mask model (model1) and model5 and model6. There are significant differences between all the other pairs of models ($P < 0.01$, Wilcoxon signed-rank test, Bonferroni corrected). This means the following: using binary mask (model1) is significantly better than the soft mask (model2) and all other models (except model5 and model6), all the combinations of model1 and model2 are significantly better than using the soft mask (model2).

From the above analysis of the results, model5 and model6 achieve significant SDR improvements compared to all other models and they also achieve good separation (high SIR values) that is not significantly different than the binary mask. Thus, model5 and model6 retain the advantage of using the binary mask by achieving low interference between the separated sources and achieve even less distortion (better SDR) than the soft mask.

The implementation for this paper is available at: `http://cvssp.org/projects/maruss/combiningdnn/`

## 7. Conclusions

In this work, we proposed a new approach of combining mask estimates of two different deep neural networks using third deep neural network (DNN). We used the proposed combining approach to achieve the advantages of two different source separation estimates using binary and soft masks, where the first estimate (binary mask) achieves better separation with less interference than the soft mask, but with distorted sources, while the second estimate (soft mask) achieves less separation, but with less distortion than the binary mask. Combining the estimates of binary and soft masks using another DNN achieves less distortion than each estimate individually and as good separation as the binary mask. This means that the new method of combining estimates using a DNN that is trained discriminatively achieves better results than each estimator individually.

## 8. Acknowledgements

This work is supported by grants EP/L027119/1 and EP/L027119/2 from the UK Engineering and Physical Sciences Research Council (EPSRC).

# 9. References


[1] 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.

[2] 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.

[3] 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.

[4] E. M. Grais, M. U. Sen, and H. Erdogan, "Deep neural networks for single channel source separation," in *Proc. ICASSP*, 2014, pp. 3734–3738.

[5] 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.

[6] G. Roma, A. J. Simpson, E. M. Grais, and M. D. Plumbley, "Remixing musical audio on the web using source separation," in *Proc. the 2nd Web Audio Conference (WAC)*, 2016.

[7] E. M. Grais and H. Erdogan, "Single channel speech music separation using nonnegative matrix factorization and spectral masks," in *Proc. DSP*, 2011.

[8] A. Narayanan and D. Wang, "Ideal ratio mask estimation using deep neural networks for robust speech recognition," in *Proc. ICASSP*, 2013, pp. 7092–7096.

[9] 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.

[10] A. Sharkey, *Combining Artificial Neural Nets*. Springer, 1999.

[11] M. LeBlanc and R. Tibshirani, "Combining estimates in regression and classification," *Journal of the American Statistical Association*, vol. 91, no. 436, pp. 1641–1650, 1996.

[12] L. K. Hansen and P. Salamon, "Neural network ensembles," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 12, no. 10, pp. 993–1001, 1990.

[13] Y. Shimshoni and N. Intrator, "Classification of seismic signals by integrating ensembles of neural networks," *IEEE Trans. on Signal Processing*, vol. 46, no. 5, pp. 1194–1201, 1998.

[14] A. Krogh and J. Vedelsby, "Neural network ensembles, cross validation, and active learning," in *Advances Neural Inform*, 1995, pp. 231–238.

[15] E. M. Grais, G. Roma, A. J. R. Simpson, and M. D. Plumbley, "Single channel audio source separation using deep neural network ensembles," in *the 140th audio engineering society convention*, 2016.

[16] A. J. R. Simpson, G. Roma, and M. D. Plumbley, "Deep Karaoke: extracting vocals from musical mixtures using a convolutional deep neural network," in *Proc. LVA/ICA*, 2015, pp. 429–436.

[17] P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Deep learning for monaural speech separation," in *Proc. ICASSP*, 2014, pp. 1562–1566.

[18] 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.

[19] Y. Bengio, "Learning deep architectures for AI," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[20] 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.

[21] E. M. Grais and H. Erdogan, "Hidden Markov models as priors for regularized nonnegative matrix factorization in single-channel source separation," in *Proc. InterSpeech*, 2012.

[22] A. Ozerov, C. Fevotte, and M. Charbit, "Factorial scaled hidden Markov model for polyphonic audio representation and source separation," in *Proc. WASPAA*, pp. 121–124.

[23] 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.

[24] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. Goodfellow, A. Bergeron, N. Bouchard, D. Warde-Farley, and Y. Bengio, "Theano: new features and speed improvements," in *Deep learning and unsupervised feature learning NIPS workshop*, 2012.

[25] 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.

[26] 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.

[27] 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.

[28] M. Friedman, "A Comparison of alternative tests of significance for the problem of m rankings," *The Annals of Mathematical Statistics*, vol. 11, no. 1, pp. 86–92, 1940.

[29] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945.

[30] H. Y. and A. C. Tamhane, *Multiple Comparison Procedures*. John Wiley and Sons, 1987.