ZERO-MEAN CONVOLUTIONAL NETWORK WITH DATA AUGMENTATION FOR SOUND LEVEL INVARIANT SINGING VOICE SEPARATION

Kin Wah Edward Lin and Masataka Goto

National Institute of Advanced Industrial Science and Technology (AIST), Japan
{edward.lin, m.goto}@aist.go.jp

ABSTRACT

We address an issue of separating singing voices from polyphonic music signals regardless of sound level variance of the mixture input. Using a standard separation quality assessment tool *BSS Eval 4.0*, we found that the separation quality of a singing voice separation (SVS) system based on a dilatable Convolutional Neural Network (CNN) decreases under different sound levels. Even if this SVS system is comparable to state-of-the-art SVS systems, it is vulnerable to the issue of sound level variance. We therefore investigate four methods of making the CNN-based SVS system invariant to different sound levels — two types of data augmentation, frame normalization, and zero-mean convolution. By testing all 15 combinations of the four methods, we found that all combinations can improve the sound level invariance and analyzed the best combinations. To the best of our knowledge, this is the first SVS work systematically investigating sound level variance.

Index Terms— singing voice separation, sound level invariance, Convolutional Neural Network (CNN), zero-mean convolutions, data augmentation

1. INTRODUCTION

Given the audio signal of a musical piece with a singer and instrumental accompaniment, a Singing Voice Separation (SVS) system automatically separates the voice and its accompaniment. There are many potential applications of SVS systems: melody extraction/annotation [1], singing skill evaluation [2], automatic lyrics recognition [3,4], automatic lyrics alignment [5, 6], singer identification [7, 8] and style visualization [9]. Since SVS is important, various SVS-related articles [10–17] have been published and two evaluation frameworks MIREX and SiSEC [18] have been established.

Despite the good performance of the state-of-the-art SVS systems, they may be vulnerable to variations of sound level in the mixture inputs (i.e., different input volumes and different mixing balances). Stoller et al. [19] investigated the sound level of datasets commonly used for SVS and Singing Voice Detection (SVD) [20]. They found that datasets are different from each other in term of the balance (mixing ratio) between the singing voice and its accompaniment, and also the entire sound level of them. They concluded that SVS models trained on a single dataset may not generalize enough but did not propose solutions to mitigate this issue. Schlüter et al. [20], on the other hand, showed that the state-of-the-art SVD systems based on Convolutional Neural Network (CNN) are vulnerable to the sound level variance issue and proposed a simple but useful technique called first-layer *zero-mean convolutions* to mitigate this issue. In this paper, we further investigate this issue with a focus on SVS rather than SVD.

This work was supported in part by JST ACCEL Grant Number JPMJAC1602, Japan.

		Music Accompaniment					Music Accompaniment				
	dB	-6	-3	0	3	6	-6	-3	0	+3	+6
Singing Voice	-6	**10.55**	6.83	3.84	1.70	0.19	**15.54**	18.24	21.49	25.41	30.01
	-3	14.65	**10.55**	6.83	3.84	1.70	13.82	**15.54**	18.24	21.49	25.41
	0	19.25	14.65	**10.55**	6.83	3.84	12.60	13.82	**15.54**	18.24	21.49
	+3	24.18	19.25	14.65	**10.55**	6.83	11.63	12.60	13.82	**15.54**	18.24
	+6	28.78	24.18	19.25	14.65	**10.55**	10.37	11.63	12.60	13.82	**15.54**
		(a) Singing Voice Quality					**(b) Accompaniment Quality**				

Table 1. Separation quality (dB) of Ideal Ratio Mask (IRM).

1.1. Illustration of Sound Level Invariance Problem

We propose a task of separating singing voices under different sound levels of the input sound mixtures. First, we show that the upper-bound baseline with an *Ideal Ratio Mask (IRM)* [21], which is the ratio of the ground-truth source spectrogram to the mixture spectrogram, is sound level invariant (i.e., separation quality does not decrease under different sound levels). We adjust the sound levels and balance of the singing voice and its accompaniment in the standard SVS dataset called *DSD100* [22] at gains of ± 3 and ± 6 dB (see Section 2.2 for adjusting procedures). Together with the original sound level, we test 25 sets of different sound level mixture inputs by using *BSS Eval 4.0* [18], a standard quality assessment tool used in evaluating SVS systems (see Section 3.2 for evaluation procedures). The separation qualities of the baseline with an IRM are shown in Table 1. We found that sets with the same sound level difference between the singing voice and its accompaniment have the same separation qualities. For example, the 0-dB difference sets (-6,-6), (-3,-3), (0,0), (+3,+3), and (+6,+6) have the same separation quality at diagonal bold values. The 3-dB difference sets (-3,-6), (0,-3), (+3,0), and (+6,+3) also have the same separation

		Music Accompaniment					Music Accompaniment				
	dB	-6	-3	0	+3	+6	-6	-3	0	+3	+6
Singing Voice	-6	**4.35**	1.33	-2.47	-10.98	-19.76	**10.23**	13.11	14.31	11.98	9.59
	-3	8.27	**5.10**	1.03	-5.79	-13.81	8.06	**10.68**	12.02	11.10	9.53
	0	11.74	8.89	**5.21**	0.65	-7.97	5.36	8.28	**10.24**	10.55	9.42
	+3	13.72	11.84	8.86	**4.07**	-2.46	1.68	5.32	8.21	**9.29**	9.06
	+6	14.75	13.58	11.39	7.95	**2.91**	-3.77	1.23	5.04	7.30	**8.19**
		(a) Singing Voice Quality					**(b) Accompaniment Quality**				

Table 2. Separation quality (dB) of the CNN baseline.

quality. This ideal invariance with IRM is the goal of enabling SVS systems to be sound level invariant. We can also conclude that this tool and procedure is reliable enough to investigate the issue of sound level variance.

Second, we show that a simple dilatable Convolutional Neural Network (CNN) for SVS (see Section 2.1) is vulnerable to this sound level variance issue. We carry out the same procedures mentioned above to evaluate this CNN baseline. Table 2 shows the separation qualities of the CNN baseline. The results in Table 2 are far from the ideal invariant results in Table 1: sets with the same sound level difference between the singing voice and its accompaniment have different separation qualities. For example, although the singing voice quality at (0,0) is 5.21 dB, the quality at (6,6) drops to 2.91 dB. These dB drops suggest that the state-of-the-art SVS systems trained with deep learning would also be vulnerable to the sound level variance issue. Note that our CNN baseline without data augmentation and model blending [23] achieved good separation qualities for the singing voice and its accompaniment at 5.21 dB and 10.24 dB, respectively, at the original sound level (0,0). Our CNN is thus comparable to the state-of-the-art system [23], which uses data augmentation and model blending to achieve the best results (5.59 dB, 11.40 dB) in SiSEC 2016 [22].

1.2. Research Approach

We investigate four methods of improving the CNN baseline to achieve sound level invariance. We first need to quantify how much invariance each CNN-based SVS system has. Due to large computational costs, it is impractical to evaluate all possible sound level differences between the singing voice and its accompaniment. As a workaround, this paper focuses on the 0-dB sound level difference corresponding to the diagonal bold values in the above tables. Since we want those values to be closer, we simply evaluate their differences by calculating the Average Squared Differences (ASD) among all 10 pairs of those values (i.e., the average of square differences of (-6,-6)-(-3,-3) pair, (-6,-6)-(0,0) pair, (-6,-6)-(+3,+3) pair, ..., (+3,+3)-(+6,+6) pair). The smaller the ASD is, the more ideal the sound level invariance is. Moreover, we also need to make sure the proposed sound-level-invariant SVS system is still able to maintain a good separation quality. We therefore evaluate the separation quality maintenance by calculating both the average and the standard deviation of those values. Based on these quantitative evaluation measures, we evaluate all combinations of the four methods to find out the best combinations and examine the effectiveness of each method.

2. METHODS

2.1. CNN Baseline

We adopt our previous CNN [24] proposed for SVS. Its performance is further improved to maintain its competitiveness with the current state-of-the-art systems. Given stereo mixture signals, we apply Short-Time Fourier Transform (STFT) on each mixture signal to obtain the magnitude and phase spectrograms (with Hann windowing size of 1024 samples, hop size of 256 samples, sampling rate of 22.05 kHz). Our CNN takes a 25-continuous-frames excerpt with their frequency capped at 8 kHz (keeping 372 bins) as the network input X. As the signals are stereo, the size of X is then $(2{\times}372{\times}25)$. The input is then followed by 2 convolutional layers, each of which has 64 $(3{\times}3)$ filters with no padding; a non-overlap $(3{\times}3)$ max-pooling; 2 convolutional layers again, each of which has 64 $(3{\times}3)$ filters with no padding; and a non-overlap $(3{\times}3)$ max-pooling again. The excerpt size is now reduced to $(64{\times}39{\times}1)$. By applying 64 $(39{\times}1)$ filters, the excerpt can be further processed by 3 fully-convolutional dense layers [25] each of which has 744 units and has 50% dropout [26] applied to their inputs. The activation function in each convolutional and dense layer is a rectified linear function, except for the final layer, which uses a sigmoid function. The network output Y is reshaped to $(2{\times}372{\times}1)$ and is assumed to be a mask for separating the singing voice of the central frame of X.

2.2. Data Augmentation

Uhlich et al. [23] propose 3 types of data augmentation, namely Sound Level Adjusting (SLA), Left/Right Channel Swapping (CS), and Instrument Tracks Chunking and Mixing. As we are concerned here with the sound level variance issue, we adopt only SLA and CS and leave the other for future work. We adjust the sound level as follows. We first calculate the magnitude spectrograms of the ground truth singing voice and its accompaniment as described in Section 2.1. Then we scale these magnitude spectrograms at gains of ± 3 and ± 6 dB. As *BSS Eval 4.0* [18] needs the signals of the ground truth singing voice and its accompaniment for the evaluation, we calculate the inverse STFT (iSTFT) of the scaled magnitude spectrograms using their corresponding phase spectrogram to synthesize these ground truth signals. As our CNN needs the spectrogram excerpt of the mixture signal as the input, we can simply first add these synthesized singing voice and its accompaniment signals to form their mixture signals, and then we calculate the magnitude spectrograms of these mixture signals as described in Section 2.1. As the gains are applied to the signals expressed as floating-point numbers, positive gains cannot result in clipping.

2.3. Frame Normalization

As we assume the network output is a mask for separating the singing voice of the central frame of the mixture input, we carry out frame normalization instead of batch normalization [27] or instance normalization [28]. We use Welford's

algorithm [29] to update the normalization parameters. First, we set the frame mean $\bar{x}$ to be a zero vector of size 372; one for each channel. Then we subtract each frame from each spectrogram excerpt in a training mini-batch with $\bar{x}$. Then we update the frame mean $\bar{x}$ and variance $\hat{x}$ by calculating the mean and variance of these subtracted frames respectively. While we test our CNN, we do not update $\bar{x}$ and $\hat{x}$. This normalization is expected to remove sound level information and still maintain the harmonic structure. Finally, the normalized network input is concatenated [30] with the original network input as we also strive to maintain the separation quality.

2.4. First Layer Zero Mean Convolution

Schlüter et al. [20] mathematically show that any cross-correlation with a zero-mean filter will remove a global offset from a log-magnitude spectrogram. They also empirically show when the input is log-magnitude mel spectrogram and the learnable filter of the first convolutional layer is zero-mean, then the global offset could be accountable for the sound level variance in the context of SVD. Since we did not succeed in training a CNN using log-magnitude spectrogram in the context of SVS, we only adopt their technique to magnitude spectrogram. Although the global offset cannot be theoretically canceled in our case, as this paper was greatly inspired by Schlüter et al. [20], we would still like to see how effective this zero-mean filter is used with non-log magnitude spectrogram in the context of SVS.

3. EXPERIMENTS

3.1. Training

The DSD100 dataset [22] is a public dataset created for evaluating SVS systems. 100 songs feature different artists and genres. 50 songs are for the development (Dev) set and the other 50 songs are for a test (Test) set. We leave a more advanced dataset called MUSDB18 [18] for future work.

Networks are initialized by random orthogonal matrices [31] and are trained with mini-batches of B=207 excerpts ($\sim$ 60.08 secs). The Tensorflow[1] ADAM [32] optimizer with its default values is used to minimize the loss function below:

$$\frac{1}{BF}\sum_{b=0}^{B-1}\sum_{f=0}^{F-1}\left\{\begin{array}{l}\Big(V[b,0,f,0]-Y[b,0,f,0]\times X[b,0,f,12]\Big)^2+\\ \Big(V[b,1,f,0]-Y[b,1,f,0]\times X[b,1,f,12]\Big)^2\end{array}\right\} \quad (1)$$

where F=372 is the number of bins, V is the magnitude spectrogram of the ground truth singing voice, Y is the network output and X is the network input. V, Y and X are expressed in the [Batch, Left/Right Channel, Frequency, Time] format. Note that we need only the central frame of X to calculate the above loss function. All 16 CNNs are trained with all possible spectrogram excerpts found in the Dev set. In case of SLA, we can store the augmented datasets in advance, so that we can simply randomly pick the excerpts from these datasets.

[1] https://www.tensorflow.org/

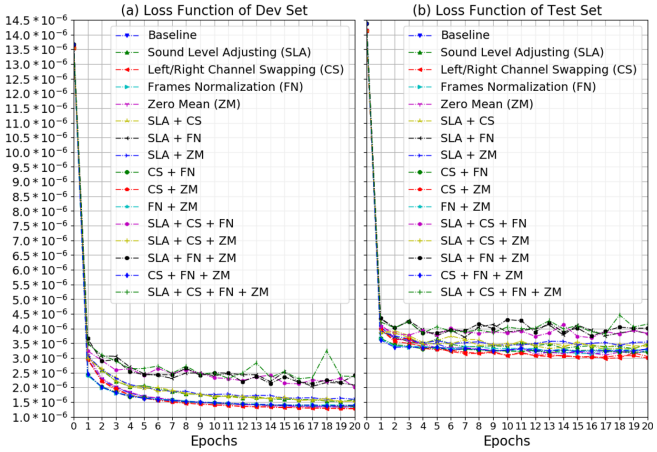

Fig. 1. Loss functions for each method at 0 dB gain.

This speeds up the training process because we do not need to augment the data on-the-fly. Given the 3 hrs 35 mins of songs in the Dev set, we have 5415 updates in each epoch. We train each CNN with 20 epochs. At each epoch, all excerpts are shuffled. Figure 1 shows the loss functions of each CNN at the original sound level. We see that all 16 CNNs are properly trained without over-fitting and under-fitting. Note that, for each CNN, the loss values at each epoch are the average of the loss values of each song. We are able to calculate such values because we dilate our CNNs as described in [33] to accept an arbitrary number of frames. Although we calculate and record the loss value after each epoch, this does not affect much the training time. The training time for each CNN is within 5 hrs using a single GPU. It is much faster than the training time of the latest system [16], which takes 79 hrs.

3.2. Evaluation

We calculate the iSTFT of the element-wise multiplication of Y and X with the mixture phase spectrogram to synthesize the separated signals. As our CNN is dilatable, we can synthesize the signals per song. It takes about 30 mins for 50 songs. We use *BSS Eval 4.0* [18] instead of *BSS Eval 3.0* because it is much faster and has similar performance [18]. As the rejections of the interference, noise and artifacts are assumed to be equally important [34], all measurement values are in terms of Source-to-Distortion Ratio (SDR). Each song is evaluated based on the average SDR of a set of 30 sec music clips with 15 sec overlap. Following the evaluation scheme of SiSEC 2016 [22], we also exclude the clips which are smaller than 30 sec and yield Not-a-Number (NaN) SDR values for the singing voice under the IRM scheme. The NaN values mostly occur at the start and end of the song, where there is no singing voice. Each method is evaluated based on the median of the Test song SDRs.

3.3. Results

Based on the calculation method described in Section 1, the sound level variances and separation quality maintenance of each method are shown in Table 3. First, we see all methods achieve an ASD smaller than the CNN baseline does,

					S. Lvl. Variances		Sep. Quality Maintenance			
Meth	S.Lvl.	Ch.	Fr.	Z.	Voice	Accomp.	Voice		Accomp.	
-od	Adj.	Swap	Nor.	M.	ASD	ASD	Ave.	S.D.	Ave.	S.D
IRM					0.000	0.000	10.548	0.000	15.542	0.000
(1)	✓				0.090	**0.071**	4.834	0.212	10.267	**0.188**
(2)		✓			1.637	1.839	4.391	0.905	9.966	0.959
(3)			✓		1.099	1.498	4.418	0.741	9.828	0.865
(4)				✓	1.212	1.808	4.340	0.778	9.810	0.951
(5)	✓	✓			**0.074**	0.131	**4.946**	**0.192**	10.307	0.256
(6)	✓		✓		0.576	0.565	4.321	0.537	9.838	0.532
(7)	✓			✓	0.443	0.172	4.746	0.470	10.249	0.293
(8)		✓	✓		1.429	1.621	4.366	0.845	9.776	0.900
(9)		✓		✓	1.393	1.571	4.555	0.835	10.078	0.886
(10)			✓	✓	1.189	1.419	4.399	0.771	9.834	0.842
(11)	✓	✓	✓		0.389	0.296	4.305	0.441	9.749	0.385
(12)	✓	✓		✓	0.227	0.125	4.932	0.337	**10.532**	0.250
(13)	✓		✓	✓	1.057	0.752	4.139	0.727	9.777	0.613
(14)		✓	✓	✓	0.769	0.929	4.542	0.620	9.946	0.682
(15)	✓	✓	✓	✓	0.919	0.651	4.087	0.678	9.576	0.571
CNN					1.727	1.991	4.329	0.929	9.727	0.998

Table 3. Sound level variances and separation quality maintenance of each method at gains of ± 3 and ± 6 dB. The smaller the ASD, the smaller (better) the variances. The larger the average and the smaller the standard deviation, the better the maintenance.

		Music Accompaniment					Music Accompaniment				
	dB	-6	-3	0	+3	+6	-6	-3	0	+3	+6
Singing Voice	-6	**4.43**	1.29	-0.66	-3.26	-8.85	**10.16**	13.03	16.45	19.48	20.88
	-3	7.97	**4.78**	1.55	-0.77	-4.39	7.45	**10.42**	13.16	16.33	18.67
	0	10.40	8.17	**5.00**	1.54	-1.17	4.12	7.66	**10.58**	13.20	15.73
	+3	13.02	11.04	8.41	**5.21**	1.61	0.03	4.43	7.89	**10.79**	13.07
	+6	14.33	13.40	11.13	8.59	**5.25**	-4.91	0.34	4.81	8.25	**10.71**
		(a) Singing Voice Quality					**(b) Accompaniment Quality**				

Table 4. Separation quality (dB) of the method (12).

	Sound Level Variances		Separation Quality Maintenance			
	Singing Voice	Music Accomp.	Singing Voice		Music Accomp.	
	ASD	ASD	Ave.	S.D.	Ave.	S.D.
IRM	0.0000	0.0000	10.5479	0.0000	15.5417	0.0000
(5)	**0.0148**	0.0053	5.3247	**0.0862**	10.7862	0.0516
(12)	0.0308	**0.0018**	**5.3363**	0.1242	**10.8523**	**0.0296**
CNN	0.2065	0.2807	5.0280	0.3214	10.1896	0.3747

Table 5. Sound level variances and separation quality maintenance of method (5) and (12) at gains of $\pm 1, \pm 2$ dB. At 0 dB gain, method (12) achieves 5.34 dB for the singing voice and 10.87 dB for its accompaniment.

suggesting all methods can improve the sound level invariance of deep-learning-based SVS systems. Next, we see that method (5) achieves the best variances and maintenance for the singing voice; and method (12) achieves the second best variances and best maintenance for the music accompaniment. Method (12) has slightly lower singing voice maintenance than method (5). The separation quality of method (12) is shown in Table 4. Method (12) has much better invariances improvement than the CNN baseline shown in Table 2. Since ± 3 and ± 6 would be too loud or too soft, we further carry out the same experiment for methods (5) and (12) with the gains of ± 1 and ± 2, and the result is shown in Table 5. Now, the result of Table 5 contradicts with Table 3 as method (12) has slightly better singing voice maintenance than method (5).

In this way, we confirmed that methods (5) and (12) are more effective than the other combinations. However, we cannot conclude that method (12) with zero-mean convolution technique is better than method (5) without it, though we used "zero-mean convolutional network" in the title of this paper. This is because the original zero-mean convolution technique [20] was designed for log-magnitude mel spectrogram and, theoretically speaking, the global offset (i.e., sound level variance) can be removed only when it is applied to log-magnitude (mel) spectrogram. In the future we plan to investigate how to successfully train a CNN with log-magnitude spectrogram so that the potential of the zero-mean convolution technique can be unleashed.

Another dissatisfying aspect of this evaluation is that the range and the interval of the sound level adjustment need to be set in advance. This introduces a bias in the evaluation. For example, as we know which ranges of sound levels we test, there is a strong bias towards SLA method. Moreover, beside average squared differences, other candidates like the sum of squared difference may also be suitable for measuring the sound level invariance. As future work, we investigate the relationships between the sound level invariance and the network input and architecture so that we do not have to set the range and interval of the sound level adjustment in advance.

We can investigate the usefulness of each four method by further examining Table 3 from the viewpoint of ablation tests. By comparing methods (9) and (12), for example, we can see that the sound level variances and the maintenance become worse if we remove the SLA method. After carrying out all similar comparisons, we have the following observations. (i) Although each of four methods improves the sound level variances and maintenance, the combination of all of them is not the best. This indicates each method has their own way to tackle the sound level invariance issue, and they may interfere with each other. (ii) SLA is the most effective method. (iii) Although CS is not directly related to sound level invariance, its data augmentation doubles the dataset size and thus improves the model generalization. (iv) FN contradicts with SLA very much. However, if we do not know how to adjust the sound level properly, FN could be useful. (v) ZM slightly interferes with SLA, but it works well with CS to maintain separation quality of the music accompaniment.

4. CONCLUSION

We investigated how sound level variances of the mixture input affect the SVS system. By giving systematical evaluation procedures, we not only showed that the current state-of-the-art systems are vulnerable to this sound level variance issue, but also showed that all combinations of all four methods eased this issue, and the combination of sound level adjusting and left/right channel swapping can be used with zero-mean convolution, but should not be used with frame normalization.

5. REFERENCES

[1] J. Salamon, R. Bittner, J. Bonada, J.J Bosch, G. Emilia, and J.P. Bello, "An analysis/synthesis framework for automatic f0 annotation of multitrack datasets," in *ISMIR*, 2017.

[2] K.W.E. Lin, H. Anderson, M.H.M. Hamzeen, and S. Lui, "Implementation and evaluation of real-time interactive user interface design in self-learning singing pitch training apps," in *ICMC and SMC*, 2014.

[3] A. Mesaros and T. Virtanen, "Automatic recognition of lyrics in singing," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2010, pp. 4, 2010.

[4] M. McVicar, D. PW Ellis, and M. Goto, "Leveraging repetition for improved automatic lyric transcription in popular music," in *ICASSP*, 2014.

[5] M. Mauch, H. Fujihara, and M. Goto, "Lyrics-to-audio alignment and phrase-level segmentation using incomplete internet-style chord annotations," in *SMC*, 2010.

[6] H. Fujihara, M. Goto, J. Ogata, and H.G. Okuno, "Lyricsynchronizer: Automatic synchronization system between musical audio signals and lyrics," *J-STSP, 5-6*, pp. 1252–1261, 2011.

[7] K.W.E. Lin, T. Feng, N. Agus, C. So, and S. Lui, "Modelling mutual information between voiceprint and optimal number of mel-frequency cepstral coefficients in voice discrimination," in *ICMLA*, 2014.

[8] H. Fujihara, M. Goto, T. Kitahara, and H.G. Okuno, "A modeling of singing voice robust to accompaniment sounds and its application to singer identification and vocal-timbre-similarity-based music information retrieval," *IEEE Trans on Audio, Speech and Language Processing*, 2010.

[9] K.W.E. Lin, H. Anderson, N. Agus, C. So, and S. Lui, "Visualising singing style under common musical events using pitch-dynamics trajectories and modified traclus clustering," in *ICMLA*, 2014.

[10] P.S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Singing-voice separation from monaural recordings using deep recurrent neural networks," in *ISMIR*, 2014.

[11] S. Uhlich, F. Giron, and Y. Mitsufuji, "Deep neural network based instrument extraction from music," in *ICASSP*, 2015.

[12] Z.C. Fan, J.S.R. Jang, and C.L. Lu, "Singing voice separation and pitch extraction from monaural polyphonic audio music via dnn and adaptive pitch tracking," in *IEEE BigMM*, 2016.

[13] Y. Luo, Z. Chen, J. R. Hershey, J. Le Roux, and N. Mesgarani, "Deep clustering and conventional networks for music separation: Stronger together," in *ICASSP*, 2017.

[14] A. Jansson, E. Humphrey, N. Montecchio, R. Bittner, A. Kumar, and T. Weyde, "Singing voice separation with deep u-net convolutional networks," in *ISMIR*, 2017.

[15] D. Stoller, S. Ewert, and S. Dixon, "Wave-u-net: A multi-scale neural network for end-to-end audio source separation," in *ISMIR*, 2017.

[16] N. Takahashi and Y. Mitsufuji, "Multi-scale multi-band densenets for audio source separation," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2017.

[17] Z. Rafii, A. Liutkus, F-R Stoter, S.I. Mimilakis, D. FitzGerald, and B. Pardo, "An overview of lead and accompaniment separation in music," *IEEE/ACM Trans. Audio, Speech, Language Process*, vol. 26, no. 8, pp. 1307–1335, 2018.

[18] F-R Stöter, A. Liutkus, and N. Ito, "The 2018 signal separation evaluation campaign," in *LVA/ICA*. Springer, 2018.

[19] Daniel Stoller, Sebastian Ewert, and Simon Dixon, "Jointly detecting and separating singing voice: A multi-task approach," in *LVA/ICA*. Springer, 2018.

[20] Jan Schlüter and Bernhard Lehner, "Zero-mean convolutions for level-invariant singing voice detection," in *ISMIR*, 2018.

[21] Y. Wang, A. Narayanan, and D.L. 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.

[22] A. Liutkus, F-R Stöter, Z. Rafii, D. Kitamura, B. Rivet, N. Ito, N. Ono, and J. Fontecave, "The 2016 signal separation evaluation campaign," in *LVA/ICA*. Springer, 2017.

[23] S. Uhlich, M. Porcu, F. Giron, M. Enenkl, T. Kemp, N. Takahashi, and Y. Mitsufuji, "Improving music source separation based on deep neural networks through data augmentation and network blending," in *ICASSP*, 2017, pp. 261–265.

[24] K.W.E Lin, B. B.T., E. Koh, S. Lui, and D. Herremans, "Singing voice separation using a deep convolutional neural network trained by ideal binary mask and cross entropy," *Neural Computing and Applications*, Dec 2018.

[25] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *CVPR*, 2015.

[26] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *JMLR*, vol. 15, no. 1, pp. 1929–1958, 2014.

[27] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv:1502.03167*, 2015.

[28] D. Ulyanov, A. Vedaldi, and V.S. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv:1607.08022*, 2016.

[29] R.F. Ling, "Comparison of several algorithms for computing sample means and variances," *JASA, 69-348*, pp. 859–866, 1974.

[30] G. Huang, Z. Liu, L. Van Der Maaten, and K.Q. Weinberger, "Densely connected convolutional networks," in *CVPR*, 2017.

[31] A.M. Saxe, J.L. McClelland, and S. Ganguli, "Exact solutions to the nonlinear dynamics of learning in deep linear neural networks," *arXiv:1312.6120*, 2013.

[32] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv:1412.6980*, 2014.

[33] T. Sercu and V. Goel, "Dense prediction on sequences with time-dilated convolutions for speech recognition," *arXiv:1312.6120*, 2016.

[34] 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. 1462–1469, 2006.