JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 38, 499-515 (2022)
DOI: 10.6688/JISE.202205_38(3).0001

Monaural Instrument Sound Segregation by Stacked Recurrent Neural Network

WEN-HSING LAI[1] AND SIOU-LIN WANG[2]
[1]*Department of Computer and Communication Engineering*
[2]*Ph.D. Program in Engineering Science and Technology, College of Engineering*
National Kaohsiung University of Science and Technology
Kaohsiung, 824 Taiwan
E-mail: {lwh; 0015901}@nkust.edu.tw

A stacked recurrent neural network (sRNN) with gated recurrent units (GRUs) and jointly optimized soft time-frequency mask was proposed for extracting target musical instrument sounds from a mixture of instrumental sound. The sRNN model stacks and links multiple simple recurrent neural networks (RNNs), which makes sRNN an excellent model with temporal dynamic behavior and real deepness. The GRU improves the gate foundations of long short-term memory and reduces the operating time. Experiments were conducted to test the proposed method. A musical dataset collected from real instrumental music was used for training and testing; electric guitar and drum sounds were the target sounds. Objective and subjective assessment scores obtained for the proposed method were compared with those obtained for two models, namely Wave-U-Net and SH-4stack, and a conventional RNN model. The results indicated that electric guitar and drum sounds can be successfully extracted through the proposed method.

Keywords: electric guitar, drums, sound separation, stacked recurrent neural network, gated recurrent unit, time-frequency mask

1. INTRODUCTION

Musical instruments are indispensable elements in music. Although many audio source separation methods have been developed, they mainly extract vocal tracks, making singing separation the mainstream method. Fewer studies have analyzed the separation of musical instruments. Extracting specific musical instruments from a mixture signal can be achieved through different audio source separation methods.

A typical audio source separation system can be split into supervised and unsupervised learning. Supervised learning uses machine training to label and find common traits in data groups and learns them to generate a model as the basis for separating input data. Examples of supervised learning include deep neural network (DNN) [1], deep recurrent neural network (DRNN) [2-5], supervised non-negative matrix factorization (NMF) [6-8], classification and regression tree (CART) [9], and computational auditory scene analysis (CASA) [10, 11], *etc.*. Supervised learning systems can reinforce models by training large volumes of data and increase training times to improve the generated results.

Unsupervised learning involves providing input data to the machine and mining internal structures and types in the data by using appropriate data processing as well as logic or algorithms without machine learning. The mined data can immediately be sorted and segmented based on information in the data. Examples of unsupervised learning include *k*-

Received February 1, 2021; revised April 6, 2021; accepted May 12, 2021.
Communicated by Po-Chyi Su.

means clusters, Gaussian mixed models (GMM) [12-14], robust principal component analysis (RPCA) [15-17], and repeating pattern extraction techniques (REPET) [18, 19], *etc.* The advantage of unsupervised learning is that it does not require to do the time-consuming data labeling and training. However, the separation result cannot be improved by increasing the amount of training data, either.

Several instrument sound separation techniques are available and can be divided into four categories; CASA-based, decomposition-based, model-based, and neural network-based techniques. Most such techniques are also applied in singing voice separation. One example of a CASA-based system is that proposed in [20], which uses pitch-based CASA and a time-frequency masking method. In addition, NMF [21-26] is one of the most widely used decomposition-based methods for separation. However, these methods commonly encounter difficulties at the clustering step and are best suited to percussive instruments [27]. Instead of spectral decomposition, researchers in [28] studied time-domain decomposition.

Model-based methods such as the sinusoidal [29], average harmonic structure [27, 30, 31], and harmonic and inharmonic models [32, 33] usually entail the establishment of generative models of music signals to facilitate separation. Neural networks [34] constitute another favored method for musical instrument sound separation; examples of such networks include multilayer perceptron [35] and fuzzy neural networks [36]. Using a DNN is the current state-of-the-art method [37-39].

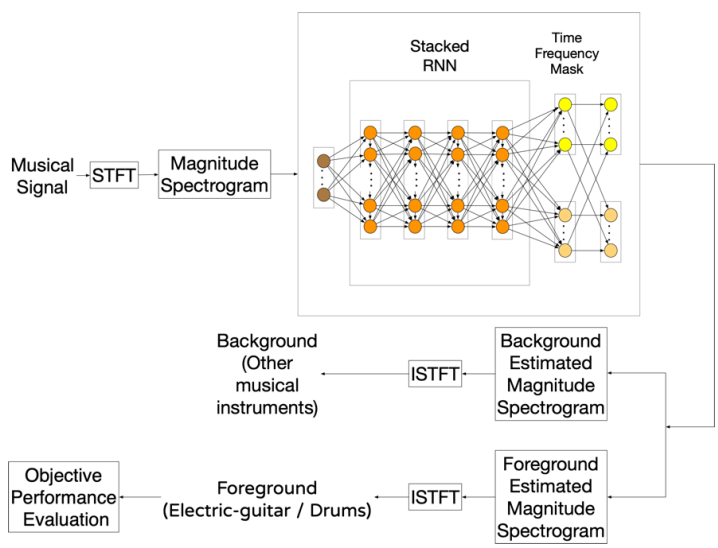


Fig. 1. Musical instrument sound separation framework.

In this research, we employed a supervised learning system that executes basic musical instrument separation through a stacked recurrent neural network (sRNN) [40] to extract the primary extraction target, namely electric guitar and drum sounds, from among multiple instrumental sounds.

As depicted in Fig. 1, the first step involves converting the musical signal into a magnitude spectrogram through short-time Fourier transform (STFT). Subsequently, the sRNN

is used to train and generate the musical instrument sound separation model. The foreground magnitude spectrogram successfully extracted from the musical signal is converted into a separate waveform through inverse STFT (ISTFT). Finally, this waveform is compared with the original target sound to perform an objective and subjective performance evaluation.

2. PROPOSED METHOD

Because of its temporal dynamic behavior, a recurrent neural network (RNN) is often employed in audio signal processing. However, an RNN cannot be considered a deep learning model because with a simple RNN, hierarchizing the input of the current time step is difficult. An sRNN is a method of stacking and linking multiple simple RNNs; thus, an sRNN exhibits depth. Therefore, we proposed a system entailing the use of an sRNN to extract the primary target, electric guitar or drum sound, from among multiple instrumental sounds. Gated recurrent units (GRUs), which improve the gate foundations of long short-term memory (LSTM) and simplify LSTM parameters to reduce operating time, are used as the cells of the recurrent hidden layer in the sRNN architecture. A soft time-frequency mask is jointly optimized and generated to separate the mixture. The sRNN, GRU, and soft time-frequency mask are described in the following sections.

2.1 Stacked Recurrent Neural Network

Neural networks simulate the working principle of nerve cells in the brain to make node-to-node connections and use parameter vectors as the weighted value for controlling these connections, achieving machine self-learning. A simple RNN use paths that form loops for data with time sequences to infinitely cycle through closed circulations and simultaneously remember and update to generate the newest analysis results. The schematic of the simple RNN and the expanded network structure are presented in Fig. 2.

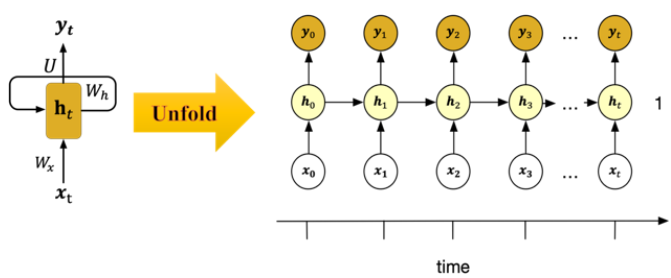


Fig. 2. The simple RNN operation structure.

According to Fig. 2, assuming that a model is built using a sequence with a length of T, after it is expanded, the model can become a T-time-stage neural network. In the network, the hidden state is the neural network memory where the result of the previous step is stored.

As presented in Eq. (1), the hidden state in time stage t, h_t, is derived from input x_t and h_{t-1}. W_x is the weight matrix from the input state to the hidden state, and W_h is the

weight matrix of the hidden state of time $t-1$. In addition, f is typically the *tanh* function or improves in performance by using rectified linear units.

$$h_t = f(W_x x_t + W_h h_{t-1}) \tag{1}$$

Finally, the predicted output y_t is calculated, where the weight matrix is U. The calculations are presented in Eq. (2).

$$y_t = softmax(U h_t) \tag{2}$$

Typically, to minimize the error between the trained output and the real answer, accurate results can be obtained through the simple RNN training; however, the simple RNN has two weaknesses. First, because of the over simplistic structure of the simple RNN, the hierarchical processing of the input cannot be conducted in the current time step. The simple RNN structure can be expanded, such as the sRNN [41] approach used in this research, which stacks multiple recurrent hidden layers to provide multilevel information through multiple time scales.

$$\begin{aligned} h_t^l &= f_h(h_t^{l-1}, h_{t-1}^l) \\ &= \varnothing_l(U^l h_{t-1}^l + W^l h_t^{l-1}) \end{aligned} \tag{3}$$

As presented in Eq. (3), h_t^l is the hidden state of the lth layer in time stage t and is obtained by calculating state transition function f_h of h_t^{l-1} and h_{t-1}^l. In the equation, $\varnothing_l(\cdot)$ is an element-wise nonlinear function, W^l is the weight matrix in the lth hidden layer, and U^l is the weight matrix of the recurrent connection in the lth hidden layer.

Another weakness is encountering gradient exploding or gradient vanishing during training. This results in the simple RNN being unable to normally process large amounts of data and difficulties in capturing and learning long-term dependence between time series data. But if LSTM or GRU models are used in the hidden state, gradient exploding or gradient vanishing can be drastically prevented through their gating mechanisms.

2.2 Gated Recurrent Unit

This research used gated recurrent unit (GRU) as the cells of the recurrent hidden layer for the sRNN architecture. GRU improve the gate foundations of LSTM and simplify the LSTM parameters to reduce operation time.

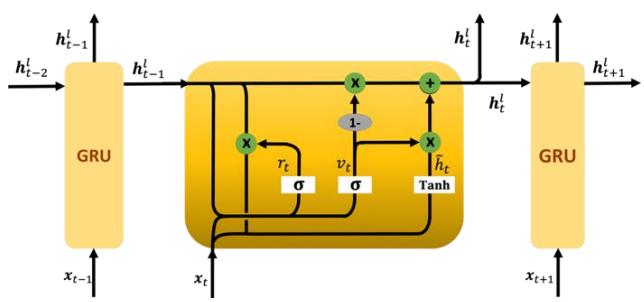


Fig. 3. The GRU operation structure.

Fig. 3 presents how the GRU operation structure can be presented as Eqs. (4)-(7):

$$v_t = \sigma(W_v \cdot [h_{t-1}, x_t]), \tag{4}$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]), \tag{5}$$

$$\tilde{h}_t = tanh(W \cdot [r_t * h_{t-1}, x_t]), \tag{6}$$

$$h_t = (1 - v_t) \odot h_{t-1} + v_t \odot \tilde{h}_t. \tag{7}$$

In the Fig. 3, GRU structure, × is pointwise multiplication, and + is pointwise addition. The two nodes σ and *tanh* have specific weights, and mapping conversion is performed inside these nodes. The mapping conversion results are presented through the "1−" node.

GRU use two gates: reset gate r_t and the gate of the update hidden state v_t. The goal of r_t is to "ignore" the status of past hidden states – for instance, when $r = 0$, as presented in Eq. (6), the $\tilde{h}_t$ of the candidate hidden layer is directly determined by input x, thereby "ignoring" past hidden states.

v_t is the aggregate of the forget gate and input gate in LSTM. In Eq. (7), $(1 - v_t) \odot h_{t-1}$ is the data in past hidden states that must be forgotten and deleted by the forget gate. $v_t \odot \tilde{h}_t$ is equivalent to the input gate and adds weight v_t to $\tilde{h}_t$, where the new data is located, to generate the newest hidden state h_t.

Furthermore, the rectified linear unit is added in the output state for activation. This can again prevent the gradient from worsening during gradient descent to normally process large amounts of data and can add multi-level networks to increase the precision of training outcomes.

2.3 Time-Frequency Mask

In prevalent audio source separation methods, the weight distribution of each segment within the mixture signal in the foreground and background is calculated. The generated matrix is the time-frequency mask, which can be divided into two types: spectrograms in which the foreground and background of each segment of data that can be explicitly represented by 0 or 1 are binary hard masks, and spectrograms in which the foreground and background of each segment of information that are matched to weight values between 0 and 1 are soft masks. The soft mask formula is presented in Eq. (8).

$$S_{mask}(m,n) = \frac{|F(m,n)|}{|F(m,n)| + gain * |B(m,n)|} \tag{8}$$

F is the foreground matrix, which is the spectrogram of the electric guitar/drums sound. B is the background matrix or the spectrogram of the other musical instruments. Within $m = 1, \ldots, n_1$ and $n = 1, \ldots, n_2$, n_1 and n_2 represent frequency and time in the spectrogram, respectively, and the gain value is correlated with the signal-to-noise ratio (SNR) of musical signals. Because the musical signals used in this research were all mixed with SNR 0dB, the gain value used was 1.0.

3. EXPERIMENTS

To characterize musical signals, the waveform of the musical signals is first converted

into a spectrogram through STFT. The waveform output is then converted using ISTFT from the estimated magnitude spectrogram generated by the trained model. In this research, input waveforms were processed using 1024-point STFT with 50% overlap.

3.1 Datasets

In this research, the MedleyDB [42] dataset was used to perform the separation experiment. MedleyDB was distributed by Bittner et al. in 2014 and includes 122 multi-tracks in total, of which 105 are full length tracks, that are approximately 3-5 minutes long. The dataset also includes 17 excerpt recording songs that are 1 min or shorter.

The genre of the tracks in the dataset include rock, pop, classical music, jazz, fusion, world music, musical theater, and singer-songwriter songs that involve the drum set, electric bass, piano, electric guitar, acoustic guitar, auxiliary percussion, double bass, violin, cello, flute, and mandolin. In addition, some tracks have male or female vocals, and some tracks were generated using a synthesizer.

Each multi-track in the dataset contained individual tracks for vocals and different instruments, with the entire dataset composed of 49 songs with electric guitar and drum tracks. Songs were split into 20-s clips. From the 49 songs, 80 clips of 8 songs (26 min and 31 s in total) were selected for training. Thirty clips of four other songs (9 min and 49 s in total) were selected for testing. All eight training songs and four testing songs contained electric guitar and drum tracks as well as different instruments such as the bass, piano, violin, cello, and trombone.

In our experiment, we simulated the separation of electric guitar and drum sounds from their monaural mixture. Hence, monaural remixes of the real song tracks with both electric guitars and drums were used. Musical mixture signals comprised electric guitar and drums as the main sounds and other instruments (not including vocals) as the accompanying sounds; they were all mixed with an SNR of 0 dB. Table 1 details the experimental parameters.

Table 1. The experimental parameters.

Name	Value
Sampling rate	44100 Hz
bits/sample	16 bits
Window size	1024
Hop size	512

3.2 Objective Performance Evaluation

In this research, the Blind Source Separation Evaluation (BSS_EVAL) toolbox [43] developed by C. Févotte was used to perform objective performance evaluation for the experiment results. The BSS_EVAL toolbox is a performance evaluation toolbox that was developed to evaluate blind source separation algorithms. It can calculate the distance between the estimated results and the target audio. The BSS_EVAL toolbox was used in this research to generate the signal-to-distortion ratio (SDR), signals-to-artifact ratio (SAR) and signal-to-interference ratio (SIR) of the experiment results and to calculate the normalized SDR (NSDR), global SDR (GSDR), global SAR (GSAR), global SIR (GSIR), and global NSDR (GNSDR). The formula is as follows:

$$\hat{s}(t) = s_{target}(t) + e_{interf}(t) + e_{artif}(t). \tag{9}$$

As presented in Eq. (9), $\hat{s}(t)$ and $s_{target}(t)$ represent the estimated and ideal target signal. In addition, $e_{interf}(t)$ represents background tracks that remain in the foreground that were erroneously determined as the foreground, and $e_{artif}(t)$ represents the foreground that remains in the background that was erroneously determined as the background. Therefore, $\hat{s}(t)$ can be broken into $s_{target}(t)$, $e_{interf}(t)$, and $e_{artif}(t)$.

The signals SDR, SAR, and SIR can be calculated using Eqs. (10)-(12):

$$\mathrm{SDR}(\hat{v}, v) = 10\log_{10}\frac{\| s_{target} \|^2}{\| e_{interf} + e_{artif} \|^2}, \tag{10}$$

$$\mathrm{SAR}(\hat{v}, v) = 10\log_{10}\frac{\| s_{target} + e_{interf} \|^2}{\| e_{artif} \|^2}, \tag{11}$$

$$\mathrm{SIR}(\hat{v}, v) = 10\log_{10}\frac{\| s_{target} \|^2}{\| e_{interf} \|^2}. \tag{12}$$

In these equations, $\hat{v}$ represents the estimated electric guitar/drums sound, with v representing the ideal clean electric guitar/drums sound. The $\|\cdot\|$ in Eqs. (10)-(12) are the Euclidean norm.

$$\mathrm{NSDR}(\hat{v}, v, x) = \mathrm{SDR}(\hat{v}, v) - \mathrm{SDR}(x, v) \tag{13}$$

The NSDR is the result of normalizing the SDR. Calculating the NSDR score can determine whether the SDR of the estimated musical signal improves against the SDR of the ideal musical signal. The calculation is presented in Eq. (13), with x representing the mixed musical signal, which is used primarily to evaluation differences in the SDR between the musical signal and separated foreground.

However, the means of the SDR, SIR, SAR, and NSDR do not take the test clip length into consideration. To be fair, the weighted means of the SDR, SIR, SAR, and NSDR of all test clips weighted by their length are used as the overall performance measure. Global SDR (GSDR), Global SIR (GSIR), Global SAR (GSAR), and Global NSDR (GNSDR), were accordingly obtained:

$$\mathrm{GSDR} = \frac{\sum_k \omega_k \mathrm{SDR}}{\sum_k \omega_k}, \tag{14}$$

$$\mathrm{GSIR} = \frac{\sum_k \omega_k \mathrm{SIR}}{\sum_k \omega_k}, \tag{15}$$

$$\mathrm{GSAR} = \frac{\sum_k \omega_k \mathrm{SAR}}{\sum_k \omega_k}, \tag{16}$$

$$\mathrm{GNSDR} = \frac{\sum_k \omega_k \mathrm{NSDR}(\hat{v}_k, v_k, x_k)}{\sum_k \omega_k}. \tag{17}$$

As shown in Eqs. (14)-(17), ω is the number of seconds in the musical signal. The global average value was obtained by multiplying the obtained SDR, SIR, SAR, and NSDR scores with the number of seconds in the musical signal before adding the results together

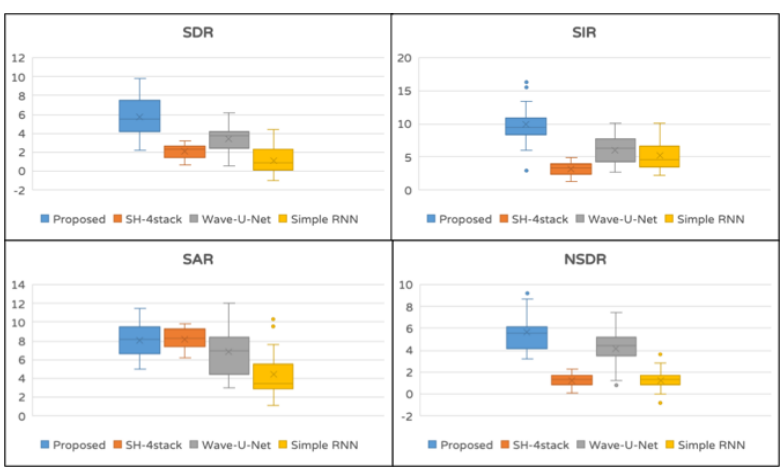
to obtain the average.

According to Eqs. (10)-(17), higher SDR, SAR, SIR, NSDR, GSDR, GSIR, GSAR, and GNSDR values indicate more effective separation performance. Research on the exact correlation of these measures with subjective ratings or on what value constitutes successful separation is still lacking. Different datasets or experimental setups can also influence the results, including the use of natural or generated recordings, the number of sources, the presence or absence of additional information, instantaneous or convolutive mixing, and instrument type. However, the objective value can be assessed using the results of the 2018 community-based Signal Separation Evaluation Campaign (SiSEC 2018) [44]. When additional data were used, the estimated median SDR, SIR, and SAR scores determined for the most effective separation method were as follows: 7, 16, and 7 dB for vocals, respectively; 5, 10, and 7 dB for bass, respectively; and 5, 9, and 6 dB for other sound, respectively. When additional data were not used, the estimated median SDR, SIR, and SAR scores were as follows: 6, 10, and 6 dB for vocals, respectively; 4, 9, and 6 dB for bass respectively; and 4, 6, and 5 dB for other sound, respectively.

Table 2. The SDR, SAR, SIR and NSDR scores of the different hyper-parameter settings.

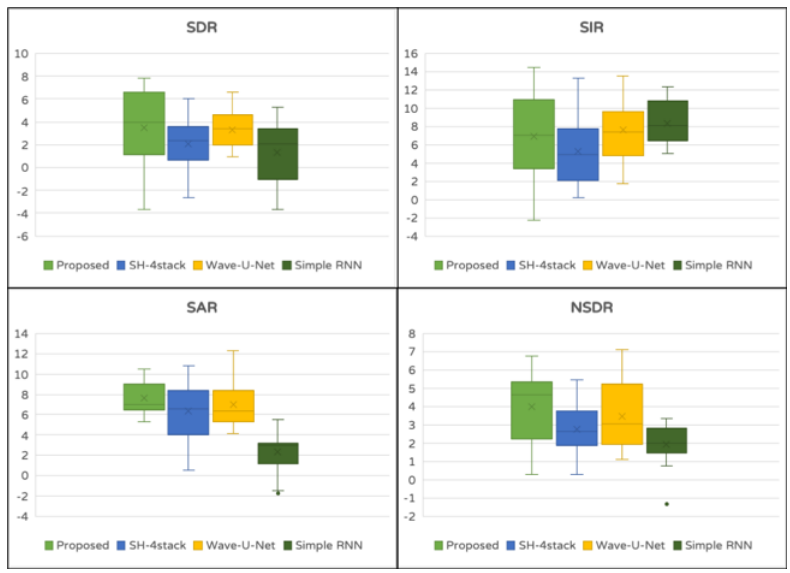
Name	SDR	SIR	SAR	NSDR
3 layers (500 cells)	4.57	9.26	7.15	4.65
3 layers (1000 cells)	**5.66**	10.13	**7.90**	**5.52**
4 layers (500 cells)	5.21	**10.35**	7.53	5.29

Experiments were conducted to evaluate the separation performance of the proposed method. First, different hyperparameters were tested; Table 2 presents the SDR, SIR, SAR, and NSDR results obtained for 3 layers with 500 cells per layer, 3 layers with 1000 cells per layer, and 4 layers with 500 cells per layer. A soft mask was used for all tests. The best scores were obtained for the 3 layers with 1000 cells per layer (except for SIR score). Therefore, these layers were used in the subsequent experiments.



(a) Electric guitar.

Fig. 4 (a). Box plots of SDR, SIR, SAR, and NSDR scores for the musical signal clips in the test set.



(b) Drums.

Fig. 4 (b). Box plots of SDR, SIR, SAR, and NSDR scores for the musical signal clips in the test set.

To ensure a balanced comparison, we used the CNN-based method, SH-4stack [45] and Wave-U-Net [46] models, and simple RNN model on MedleyDB for simulation. In the Wave-U-Net model, the U-Net architecture is changed into a one-dimensional time-domain system for end-to-end source separation. The SH-4stack model is based on a stacked hourglass network with multiple hourglass modules, with each module outputting a mask for each audio source. The masks are multiplied by the input spectrogram to generate the predicted spectrogram used for audio source separation. The simple RNN is structured using 1000 cells; unlike the proposed method, it does not use LSTM or GRU or possess a multilayer architecture.

Fig. 4 presents box plots of the distribution and average values of the SDR, SIR, SAR, and NSDR scores for the musical signal clips in the test set; Fig. 4 (a) details the electric guitar separation results, and Fig. 4 (b) presents the drum sound separation results. The comparative results are listed in Table 3. Our method demonstrated superior separation scores in most tests, indicating that the sRNN can successfully extract electric guitar and drum signals from other musical signals.

Table 3. Comparison with Wave-U-Net, SH-4stack, and Simple RNN on MedleyDB. The scores of (a) electric guitar for training; (b) electric guitar for testing; (c) drums for training; and (d) drums for testing.

(a)

Method	SDR	SIR	SAR	NSDR	GSDR	GSIR	GSAR	GNSDR
Wave-U-Net	5.96	10.50	8.67	6.39	6.05	10.64	8.79	6.43
SH-4stack	3.77	6.32	8.47	5.09	3.80	6.35	8.47	5.13
Simple RNN	2.08	8.64	3.77	4.09	2.09	8.80	3.80	4.11
Proposed	**6.82**	**12.66**	**8.91**	**7.68**	**6.84**	**12.73**	**8.96**	**7.69**

(b)

Method	SDR	SIR	SAR	NSDR	GSDR	GSIR	GSAR	GNSDR
Wave-U-Net	3.38	6.07	6.90	4.19	3.41	6.07	6.97	4.23
SH-4stack	2.14	3.22	**8.21**	1.27	2.17	3.26	**8.25**	1.27
Simple RNN	1.05	5.20	4.38	1.24	1.09	5.27	4.38	1.25
Proposed	**5.66**	**10.13**	7.90	**5.52**	**5.65**	**10.35**	7.95	**5.53**

(c)

Method	SDR	SIR	SAR	NSDR	GSDR	GSIR	GSAR	GNSDR
Wave-U-Net	5.51	**14.26**	8.42	6.82	5.53	**14.27**	8.43	6.84
SH-4stack	2.43	7.47	4.79	5.69	2.44	7.47	4.80	5.71
Simple RNN	2.58	7.13	5.27	4.06	2.64	7.13	5.37	4.08
Proposed	**5.99**	11.29	**9.75**	**6.92**	**6.01**	11.3	**9.77**	**6.94**

(d)

Method	SDR	SIR	SAR	NSDR	GSDR	GSIR	GSAR	GNSDR
Wave-U-Net	3.26	7.50	6.96	3.18	3.28	7.56	6.97	3.29
SH-4stack	2.03	5.29	6.29	2.78	2.07	5.31	6.32	2.79
Simple RNN	1.24	**7.93**	2.19	1.83	1.26	**7.93**	2.23	1.84
Proposed	**3.59**	7.11	**7.61**	**3.95**	**3.63**	7.13	**7.65**	**3.97**

A comparison with other methods is presented in Table 4. However, a direct comparison of their results was difficult because of limitations imposed by diverse dataset they used and because of limitations caused by the other methods' varying number of sources and different target and background musical instruments. Some of the methods require additional information such as musical scores. Accordingly, this study excluded methods involving sounds mixed from fewer musical instruments [21, 25, 37, 38], those involving synthetic or Musical Instrument Digital Interface (MIDI)-generated datasets [27, 29, 39], and those requiring additional information such as musical scores [28, 45, 47]; the remaining methods were used for comparison. Accordingly, the comparison revealed that our method outperformed the other methods (Table 4).

Table 4. The list of existing monaural instrument separation methods.

		Target	Background	SIR	SAR	SDR
multiple instruments/ real data/ score free	Proposed	guitar	multiple instrument	**10.13**	**7.90**	**5.66**
	[31]	main instrument	accompaniment	8.1	5.2	2.7
2-3 Instruments mixed	[21]	piano, clarinet, flute, trombone	piano, clarinet, flute, trombone	2.4–23.5	5.8–12.7	0.6–12.3
	[25]	trumpet, flute, piano	guitar, flute, piano	–	–	5.32–7.92
	[37]	violin	horn, piano	15.6	6.75	6.11
	[38]	piano	guitar	–	–	4.1–5.0
Synthetic or MIDI	[27]	organ	piccolo	25.1	12.1	11.8
	[29]	clarinet, flute, violin, trumpet	clarinet, flute, violin, trumpet	36.6	11.8	11.7
	[39]	other	vocal, drum, bass	–	–	2.35–2.58

Table 4. (Cont'd) The list of existing monaural instrument separation methods.

		Target	Background	SIR	SAR	SDR
Score informed	[28]	violin	clarinet, saxophone, bassoon	11.92–23.87	5.88–10.72	5.41–10.49
	[45]	bass, drums, other, vocals	bass, drums, other, vocals	–	–	1.77–5.16
	[47]	drum	17 instruments	4.3–10.7	0.9–11.5	–0.8 –5.5

3.3 Subjective Performance Evaluation

In this research, the SDR, SIR, SAR, NSDR, GSDR, GSIR, GSAR, and GNSDR measures were employed to objectively evaluate the quality of our proposed method. However, these objective measures only determine whether the extracted audio closely matches the original audio. Therefore, an additional subjective evaluation was performed. For this evaluation, 10 listeners were invited to participate in the experiment. The mean opinion score (MOS) and comparison mean opinion score (CMOS) were used as the subjective measures.

The MOS and CMOS are both widely used in audio and video analysis; the listeners' subjective perceptions as well as ratings and personal preferences are used for evaluation. The MOS ranges from 1 ("bad") to 5 ("excellent"). The CMOS is based on Annex E of the International Telecommunication Union-T Recommendation P.800 [48], and its total value ranges from −3 ("much worse") to +3 ("much better"). To reduce the rating difficulty for the listeners, the assessment levels were simplified into three categories in our experiment: "worse," "equal," and "better."

The MOS results are presented in Table 5, where Tables 5 (a) and (b) detail the results for electric guitar sound separation and drum sound separation, respectively. Each listener was assigned three electric guitar and three drum sounds for each method (a total of 12 guitar and 12 drum sounds) for evaluation. As presented in Table 5 (a), the MOS for the simple RNN was between 2.27 and 3, with an average score of 2.64, indicating poor to fair ratings. The MOS for the SH-4stack model was between 2.6 and 3.13, with an average score of 2.92, indicating poor to fair ratings. The MOS for the Wave-U-Net model was between 2.93 and 3.8, with an average score of 3.35, indicating fair to good ratings (closer to fair). The MOS for our proposed method was between 3.73 and 4.13, with an average score of 3.96, indicating fair to good ratings (closer to good).

As shown in Table 5 (b), the MOS for the simple RNN was between 2.33 and 2.87, with an average score of 2.58, indicating poor to fair ratings. The MOS for the SH-4stack model was between 2.73 and 3.13, with an average score of 2.95, indicating poor to fair ratings. The MOS for the Wave-U-Net model was between 2.93 and 3.27, with an average score of 3.11, indicating fair to good ratings (closer to fair). The MOS for our proposed method was between 3.47 and 4, with an average score of 3.78, indicating fair to good ratings (closer to good).

In CMOS, the following six pair-sets of listening evaluations were performed: our proposed method was evaluated against the SH-4stack, Wave-U-Net, and Simple RNN models; Wave-U-Net was evaluated against SH-4stack and simple RNN models, and the SH-4stack model was evaluated against the simple RNN model. Each listener was assigned three pairs in each evaluation pair for the electric guitar and drum sounds separately.

Table 5. The MOS scores of (a) electric guitar and (b) drums for Wave-U-Net, SH-4stack, Simple RNN and the proposed method.

(a)

Listener	Wave-U-Net	SH-4stack	Simple RNN	Proposed
Listener 1	3.47	2.93	2.53	3.73
Listener 2	3.2	2.8	2.6	3.93
Listener 3	3.27	2.6	2.6	3.93
Listener 4	3	3.07	2.67	3.93
Listener 5	3.67	2.73	2.53	3.87
Listener 6	3.2	3.07	2.67	4.07
Listener 7	3.73	3.13	2.27	4
Listener 8	2.93	2.73	2.73	4.07
Listener 9	3.8	3.13	2.8	4
Listener 10	3.27	3	3	4.13
Average Score	3.35	2.92	2.64	**3.96**

(b)

Listener	Wave-U-Net	SH-4stack	Simple RNN	Proposed
Listener 1	3.2	2.87	2.6	3.93
Listener 2	3.13	2.73	2.47	3.87
Listener 3	3.07	2.93	2.4	3.47
Listener 4	3	3.07	2.67	3.6
Listener 5	3.27	2.73	2.53	3.53
Listener 6	3.07	3.07	2.33	3.93
Listener 7	3.27	3.13	2.53	4
Listener 8	2.93	2.8	2.8	3.73
Listener 9	2.93	3.13	2.87	3.87
Listener 10	3.27	3.0	2.6	3.87
Average Score	3.11	2.95	2.58	**3.78**

The percentage distribution of CMOS is presented in Table 6, where Tables 6 (a) and (b) show the results obtained for electric guitar sound separation and drum sound separation, respectively. The percentage ratio in the table represents the percentage ratio of preference. For example, in Table 6 (a), when our proposed method was compared with the SH-4stack model, our method accounted for 46% of the preferences and SH-4stack for 27%; the remaining 27% was considered as equal.

In conclusion, the results in Tables 5 and 6 demonstrate that the proposed method is superior to Wave-U-Net, SH-4stack, and simple RNN models in terms of the MOS and CMOS.

Table 6. The percentage distribution of CMOS of (a) electric guitar and (b) drums for SH-4stack, Wave-U-NET, the simple RNN and the proposed method.

(a)

Testing Pair	Wave-U-Net	SH-4stack	Simple RNN	Proposed	Equal
Proposed: SH-4stack	–	27%	–	46%	27%
Proposed: Wave-U-Net	27%	–	–	43%	30%
Proposed: Simple RNN	–	–	4%	93%	3%
Wave-U-Net: SH-4stack	40%	33%	–	–	27%
Wave-U-Net : Simple RNN	87%	–	6%	–	7%
SH-4stack : Simple RNN	–	53%	37%	–	10%

(b)

Testing Pair	Wave-U-Net	SH-4stack	Simple RNN	Proposed	Equal
Proposed : SH-4stack	–	20%	–	73%	7%
Proposed : Wave-U-Net	20%	–	–	60%	20%
Proposed : Simple RNN	–	–	4%	93%	3%
Wave-U-Net : SH-4stack	46%	17%	–	–	37%
Wave-U-Net : Simple RNN	60%	–	20%	–	20%
SH-4stack : Simple RNN	–	50%	20%	–	30%

4. CONCLUSIONS

In this study, we proposed an sRNN model using GRUs and jointly optimized soft mask for extracting musical instrument sounds from mixed signals derived from multiple musical instruments. The generated electric guitar and drum extraction were compared with the results from several existing methods. Although the results did not constitute perfect extractions, the objective and subjective performance evaluation scores demonstrate the feasibility of the proposed approach. If the neural network structure is reinforced and large amounts of data are added for training, additional types of musical instrument sounds can be extracted. The model can assist in musical instrument sound recognition and music information retrieval in the future.

ACKNOWLEDGMENT

The authors acknowledge Messrs Fong-Kai Jhan, Zih-Jie Lin, Sheng-Hong Huang, and Yu-Ci Sie for the assistance of the experiment.

REFERENCES

1. Y.-S. Lee, C.-Y. Wang, S.-F. Wang, J.-C. Wang, and C.-H. Wu, "Fully complex deep neural network for phase-incorporating monaural source separation," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2017, pp. 281-285.
2. P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Singing-voice separation from monaural recordings using deep recurrent neural networks," in *Proceedings of the 15th International Society for Music Information Retrieval Conference*, 2014, pp. 477-482.
3. J. Sebastian and H. A. Murthy, "Group delay based music source separation using deep recurrent Neural Networks," in *Proceedings of International Conference on Signal Processing and Communications*, 2016, pp. 1-5.
4. W. Yuan, S. Wang, X. Li, M. Unoki, and W. Wang, "Proximal deep recurrent neural network for monaural singing voice separation," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019, pp. 286-290.
5. S. Yang and W.-Q. Zhang, "Singing voice separation based on deep regression neural network," in *Proceedings of IEEE International Symposium on Signal Processing and*

Information Technology, 2019, pp. 1-5.
6. J. Byun and J. W. Shin, "Initialization for NMF-based audio source separation using priors on encoding vectors," *China Communications*, Vol. 16, 2019, pp. 177-186.
7. S. Koundinya and A. Karmakar, "Homotopy optimisation based NMF for audio source separation," *IET Signal Processing*, Vol. 12, 2018, pp. 1099-1106.
8. K. Zen, M. Suzuki, H. Sato, S. Oyama, and M. Kurihara, "Monophonic sound source separation by non-negative sparse autoencoders," in *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, 2014, pp. 3623-3626.
9. P. Silva, "Classification, segmentation and chronological prediction of cinematic sound," in *Proceedings of the 11th International Conference on Machine Learning and Applications*, 2012, pp. 369-374.
10. Q. Kong, Y. Wang, X. Song, Y. Cao, W. Wang, and M. D. Plumbley, "Source separation with weakly labelled data: An approach to computational auditory scene analysis," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2020, pp. 101-105.
11. Y. Li and D. Wang, "Separation of singing voice from music accompaniment for monaural recordings," *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 15, 2007, pp. 1475-1487.
12. Y. Bando, Y. Sasaki, and K. Yoshii, "Deep Bayesian unsupervised source separation based on a complex Gaussian mixture model," in *Proceedings of IEEE 29th International Workshop on Machine Learning for Signal Processing*, 2019, pp. 1-6.
13. T. T. H. Duong, N. Q. K. Duong, P. C. Nguyen, and C. Q. Nguyen, "Gaussian modeling-based multichannel audio source separation exploiting generic source spectral model," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, Vol. 27, 2019, pp. 32-43.
14. P. Magron and T. Virtanen, "Bayesian anisotropic gaussian model for audio source separation," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2018, pp. 166-170.
15. P.-S. Huang, S. D. Chen, P. Smaragdis, and M. Hasegawa-Johnson, "Singing-voice separation from monaural recordings using robust principal component analysis," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2012, pp. 57-60.
16. C. Zhu, D. Huang, S. Zhou, Y. Chen, J. Lin, and D. Jiang, "A robust unsupervised method for the single channel speech separation," in *Proceedings of the 15th International Conference on Computational Intelligence and Security*, 2019, pp. 387-390.
17. Y. Ikemiya, K. Itoyama, and K. Yoshii, "Singing voice separation and vocal F0 estimation based on mutual combination of robust principal component analysis and subharmonic summation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, Vol. 24, 2016, pp. 2084-2095.
18. Z. Rafii and B. Pardo, "A simple music/voice separation method based on the extraction of the repeating musical structure," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2011, pp. 221-224.
19. Z. Rafii and B. Pardo, "REpeating pattern extraction technique (REPET): A simple method for music/voice separation," *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 21, 2013, pp. 73-84.
20. Y. Li and D. Wang, "Musical sound separation using pitch-based labeling and binary

time-frequency masking," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2008, pp. 173-176.
21. D. Kitamura, H. Saruwatari, K. Shikano, K. Kondo, and Y. Takahashi, "Music signal separation by supervised nonnegative matrix factorization with basis deformation," in *Proceedings of the 18th International Conference on Digital Signal Processing*, 2013, pp. 1-6.
22. S. Kırbız and B. Günsel, "Perceptually weighted non-negative matrix factorization for blind single-channel music source separation," in *Proceedings of the 21st International Conference on Pattern Recognition*, 2012, pp. 226-229.
23. S. Yazawa, M. Hamanaka, and T. Utsuro, "Novel approach to separation of musical signal sources by NMF," in *Proceedings of the 12th International Conference on Signal Processing*, 2014, pp. 610-615.
24. A. Hayashi, H. Kameoka, T. Matsubayashi, and H. Sawada, "Non-negative periodic component analysis for music source separation," in *Proceedings of Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, 2016, pp. 1-9.
25. B. Essaid and N. Batel, "New method based on single-channel separation algorithm using gammatone filterbank for cochlear implants," in *Proceedings of International Conference on Applied Smart Systems*, 2018, pp. 1-4.
26. T. Virtanen, "Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria," *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 15, 2007, pp. 1066-1074.
27. Z. Duan, Y. Zhang, C. Zhang, and Z. Shi, "Unsupervised single-channel music source separation by average harmonic structure modeling," *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 16, 2008, pp. 766-778.
28. P.-K. Jao, L. Su, Y.-H. Yang, and B. Wohlberg, "Monaural music source separation using convolutional sparse coding," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, Vol. 24, 2016, pp. 2158-2170.
29. Y. Li, J. Woodruff, and D. Wang, "Monaural musical sound separation based on pitch and common amplitude modulation," *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 17, 2009, pp. 1361-1371.
30. A. Klapuri, T. Virtanen, and T. Heittola, "Sound source separation in monaural music signals using excitation-filter model and EM algorithm," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2010, pp. 5510-5513.
31. J.-L. Durrieu, G. Richard, and B. David, "An iterative approach to monaural musical mixture de-soloing," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2009, pp. 105-108.
32. K. Itoyama, M. Goto, K. Komatani, T. Ogata, and H. G. Okuno, "Integration and adaptation of harmonic and inharmonic models for separating polyphonic musical signals," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2007, pp. I-57-I-60.
33. K. Itoyama, M. Goto, K. Komatani, T. Ogata, and H. G. Okuno, "Simultaneous processing of sound source separation and musical instrument identification using bayesian spectral modeling," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2011, pp. 3816-3819.

34. R. J. Cant, C. S. Langensiepen, and W. Metcalf, "Mask optimisation for neural network monaural source separation," in *Proceedings of UKSim-AMSS 19th International Conference on Computer Modelling Simulation*, 2017, pp. 116-121.
35. K. Youssef and P.-Y. Woo, "Instrument sound separation in songs," in *Proceedings of IEEE International Conference on Electro/Information Technology*, 2008, pp. 442-447.
36. S. D. Teddy and E. M.-K. Lai, "Model-based approach to separating instrumental music from single track recordings," in *Proceedings of the 8th International Conference on Control, Automation, Robotics and Vision Conference*, Vol. 3, 2004, pp. 1808-1813.
37. S. Uhlich, F. Giron, and Y. Mitsufuji, "Deep neural network based instrument extraction from music," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2015, pp. 2135-2139.
38. E. Manilow, P. Seetharaman, and B. Pardo, "Simultaneous separation and transcription of mixtures with multiple polyphonic and percussive instruments," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2020, pp. 771-775.
39. P. Seetharaman, G. Wichern, S. Venkataramani, and J. L. Roux, "Class-conditional embeddings for music source separation," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019, pp. 301-305.
40. W.-H. Lai and S.-L. Wang, "Separation of electric sound based on stacked recurrent neural network," in *Proceedings of International Conference on Technologies and Applications of Artificial Intelligence*, 2020, pp. 52-55.
41. R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio, "How to construct deep recurrent neural networks," *arXiv Preprint*, 2014, arXiv:1312.6026.
42. R. Bittner, J. Salamon, M. Tierney, M. Mauch, C. Cannam, and J. Bello, "MedleyDB: A multitrack dataset for annotation-intensive MIR research," in *Proceedings of the 15th International Society for Music Information Retrieval Conference*, 2014, pp. 155-160.
43. E. Vincent, R. Gribonval, and C. Fevotte, "Performance measurement in blind audio source separation," *IEEE Transactions on Audio, Speech and Language Processing*, Vol. 14, 2006, pp. 1462-1469.
44. F.-R. Stöter, A. Liutkus, and N. Ito, "The 2018 signal separation evaluation campaign," *arXiv Preprint*, 2020, arXiv:1804.06267.
45. S. Park, T. Kim, K. Lee, and N. Kwak, "Music source separation using stacked hourglass networks," *arXiv*, 2018, No. arXiv:1805.08559.
46. D. Stoller, S. Ewert, and S. Dixon, "Wave-U-Net: A multi-scale neural network for end-to-end audio source separation," *arXiv Preprint*, 2018, No. arXiv:1806.03185.
47. O. Gillet and G. Richard, "Transcription and separation of drum signals from polyphonic music," *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 16, 2008, pp. 529-540.
48. International Telecommunication Union, "P.800: Methods for subjective determination of transmission quality," Recommendation P.800 (08/96), 1996.

Wen-Hsing Lai (賴玟杏) received the Ph.D. degree in Communication Engineering from National Chiao Tung University, Taiwan. She is an Associate Professor at the Department and Graduate Institute of Computer and Communication Engineering, National Kaohsiung University of Science and Technology, Taiwan. Her major research area is audio and speech processing.

Siou-Lin Wang (王秀琳) received the B.S. degree in Department of Computer Science and Information Engineering from Shu-Te University, Taiwan and the M.S. degree in Department of Computer and Communication Engineering from National Kaohsiung First University of Science and Technology, Taiwan. She is currently pursuing the Ph.D. degree in Ph.D. Program in Engineering Science and Technology, College of Engineering from National Kaohsiung University of Science and Technology, Taiwan.