https://doi.org/10.20965/jrm.2017.p0037

Paper:

Sound Source Localization Using Deep Learning Models

Nelson Yalta*, Kazuhiro Nakadai, and Tetsuya Ogata***

*Intermedia Art and Science Department, Waseda University
3-4-1 Ohkubo, Shinjuku, Tokyo 169-8555, Japan
E-mail: {nelson.yalta@ruri., ogata@}waseda.jp
**Honda Research Institute Japan Co., Ltd.
8-1 Honcho, Wako, Saitama 351-0188, Japan
E-mail: nakadai@jp.honda-ri.com
[Received July 22, 2016; accepted December 28, 2016]

This study proposes the use of a deep neural network to localize a sound source using an array of microphones in a reverberant environment. During the last few years, applications based on deep neural networks have performed various tasks such as image classification or speech recognition to levels that exceed even human capabilities. In our study, we employ deep residual networks, which have recently shown remarkable performance in image classification tasks even when the training period is shorter than that of other models. Deep residual networks are used to process audio input similar to multiple signal classification (MUSIC) methods. We show that with end-to-end training and generic preprocessing, the performance of deep residual networks not only surpasses the block level accuracy of linear models on nearly clean environments but also shows robustness to challenging conditions by exploiting the time delay on power information.

Keywords: sound source localization, deep learning, deep residual networks

1. Introduction

Speech is one of the most important means of communication between humans. To participate in a conversation, humans usually require information about the source of the speech. Regardless of the number of sources, humans look in the direction of the source(s) to continue interacting. In audio processing, the search for the location of the source (i.e., sound source localization (SSL)) is a generic problem formulated as part of the *cocktail party effect*, which humans can naturally solve by obtaining the source location and then filtering the desired speech information.

Machines and robots have become part of everyday life. Thus, for natural interactions with humans, robots should have auditory functions [1] that can be used for SSL, whereby recognizing sound locations and detecting sound events are necessary. Environmental factors such as background noise, sound sources in motion, and room reverberations change dynamically in the real world [2]. In addition, the number of microphones and the acoustic properties of a robot complicate SSL.

Conventional methods to implement and improve the performance of SSL include subspace-based methods such as multiple signal classification (MUSIC) [3]. In these methods, the localization process employs representations of the energy of signals as well as the time difference of the arrival of those signals. These representations are called steering vectors and can be obtained in the frequency domain by means of measurements [4] or physical models [5]. While MUSIC uses only sound information, other methods employ audio-visual data for tracking sound sources in real time [6]. One study revealed that when using both visual data and the pitch extracted from a binaural auditory system, a conversation between multiple sources can be tracked [6]. However, a binaural system can be used only for two-dimensional localization. By contrast, a microphone array is used for an SSL task in three dimensions [7]. In that study, the time delay of arrival was used to estimate the source location, but the system worked only when the source was located within 3 to 5 m of the array. The use of a pitch-cluster-map for sound identification was introduced in [8], in which the sound source localization was based on delay and sum beam forming using a microphone array of 32 channels, and for multiple sources main-lobe fitting was employed. The system could obtain the location of not only speech events but also non-voice sounds. MUSIC was implemented in [4, 9] based on standard eigenvalue decomposition (SEVD-MUSIC) and generalized eigenvalue decomposition (GEVD-MUSIC). These implementations revealed that performing SSL was possible even in noisy environments that incurred a relatively low computation cost.

Recently, deep neural networks (DNN) and deep convolutional neural networks (DCNN) approaches have led to major breakthroughs in different signal processing fields such as speech recognition [10, 11], natural language processing [12] and computer vision [13]. Many improvements have occurred in the field of computer vision using deep learning. Deep residual networks

(DRNs) [14] have shown the best performance on the ImageNet Large Scale Visual Recognition Challenge 2015.

In this study, we refined a method for performing SSL tasks by *replacing* the MUSIC method with deep learning (DL), which in some cases requires a pre-calculation of the environment, and by maintaining a *robust* implementation in challenging environments without a major increase in the *learning cost* (e.g., amount of memory usage and computational time for the optimization process).

The different implementations of MUSIC have performed real-time SSL in noisy environments. However, the performance of these methods depends on the number of microphones used for the task and diminishes with increased signal-to-noise ratio (SNR). In addition, to improve the performance of SSL using GEVD-MUSIC, the correlation matrix for noise must be pre-calculated for known noises. However, in case of *unknown noises*, the performance of GEVD-MUSIC may drop to the same accuracy level as that of SEVD-MUSIC, thus showing low *robustness* at a low SNR.

A DNN-based model used in a flexible arrangement of a microphone array was introduced in [15] to implement SSL tasks. In that study, the power and phase information of the audio was exploited to improve SSL. The use of both power and phase information is more important for a multiple-channel audio source because the location calculation is affected by noise and reflections. However, implementing a model with greater input information requires a larger number of parameters to increase the processing time and *learning cost*. Moreover, DCNNs have performed better than DNNs in speech signal processing tasks. In addition, because of the shared weights, the number of parameters to be trained can be reduced. Even with a lack of sound reflections and when using only the delay information between channels, the study in [16] showed that obtaining the direction of arrival (DOA) is possible using a DCNN. However, using a single frame increases the difficulty of finding the source location because of the lack of phase information in environments having a high noise level or a greater number of reflections. Therefore, for a time-delay based model, using multiple frames is recommended to improve the performance in noisy environments [17]. Furthermore, although DCNN approaches perform well at classification tasks [13, 18], the slow learning process, the long training time because of possible fine-tuning requirements, and the fact that the DCNN model may not be used in other tasks means that DCNN approaches are unsuitable for performing a flexible task such as SSL.

In this study, we focus on the following three aspects of DL implementation for SSL tasks:

1. Replacement: A subspace-based method such as MUSIC, which is commonly used for SSL, requires a steering vector from a multiple-channel signal to process the localization. This method can be replaced with a DCNN model by using the power information from frequency-domain frames of a multiple-channel audio as input and by providing the location of the sound source as target. For this *replacement*, the model does not require information about the environment or the input noise.
2. Robustness: Real-world environments change dynamically and their noises affect audio signals. Thus, DCNNs should perform accurate SSL at different SNR levels.
3. Learning efficiency: The performance of a DL model also depends on the training time. However, a larger training set does not ensure a good performance because of problems that arise in the training process such as overfitting and accuracy training saturation [14].

The remainder of this paper is organized as follows. We first present our proposed methods for sound source separation and then explain residual learning and its use for SSL. We next describe the implementation and training of DRNs, and demonstrate the effectiveness of DRNs experimentally. Finally, we present the conclusion of our study and offer suggestions for future research.

2. Proposed Method

2.1. Deep Convolutional Neural Network

We implement DCNN to replace MUSIC using a real-world impulse response and end-to-end training to perform the SSL task. Our DCNN uses only the power information from several frequency-domain frames of a multiple-channel audio stream to obtain the location of a sound source, and adding environmental information is not required to perform the task. During the last decade, factors such as the commercialization of high-performance low-cost graphics processing units (GPU), development of improved optimization algorithms, and availability of open-source libraries have enabled the employment of DL for classification or recognition problems. DL, which attempts to model high-level abstractions, uses artificial neural networks with multiple hidden layers of units between the input and output layers to model complex nonlinear representations. This structure is known as DNN. A principal characteristic of a DNN is its ability to self-organize sensory features from extensive training data.

Recently, *convolutional layers*, which employ a different layer configuration, were implemented in networks and were successful at several signal processing tasks [10–13]. The configuration neurons of convolutional layers, instead of being fully connected, are formed by means of spatially arranged features maps. A configuration that uses one or two convolutional layers inside a neural network is known as a convolutional neural network (CNN). A DCNN, defined previously, uses several convolutional layers. The CNN proposed in [19] employed a typical configuration consisting of:
a) convolutional layers, in which *feature maps* were obtained by *convolving* the input x with a kernel filter W,

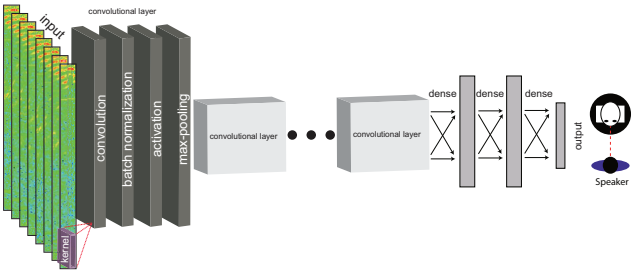

Fig. 1. DCNN structure.

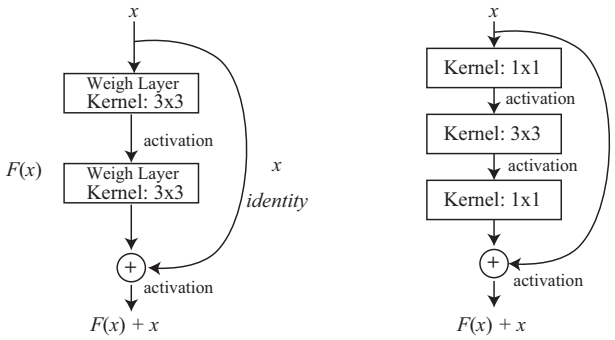

Fig. 2. Residual learning. Left: a building block. Right: a bottleneck.

and then applying a nonlinear function δ [20]. Note that the k-th feature map is determined by:

$$h^k_{i,j} = \delta\left(\left(W^k * x\right)_{i,j} + b_k\right), \quad \ldots\ldots\ldots \quad (1)$$

where W^k represents the weights and b_k is the bias at the k-th feature map.
b) sub-sampling layers known as pooling layers, which in most cases consist of a MaxPooling (max) layer, formulated as:

$$y^k_{i,j} = \max^k_{m,n\in[0,p]}\left(h^k_{i+m,j+n}\right), \quad \ldots\ldots\ldots \quad (2)$$

which, from a $p \times p$ pooling region, obtains the maximum value for the i-th, j-th position on the k-th feature map.
c) fully connected layers.

These CNNs have been used in different tasks such as image classification [13], in which they have performed better than standard feedforward neural networks because of their fewer connections and parameters as well as the ability to make proper assumptions about the nature of images. CNNs have also been tested on speech tasks such as raw speech detection [21], acoustic models [11, 22], and speech recognition [23, 24], in which the features of CNN can correctly reduce the word error rate.

Stochastic gradient decent (SGD) optimization and its variants such as momentum [25] and AdaGrad [26], have been shown to perform effective training on deep networks. However, the deeper the network is, the more difficult the training becomes because small changes to network parameters amplify outputs and increase the cost (i.e., loss) function error. Thus, to train deeper models, a mechanism called *batch normalization* was introduced in [27]. This mechanism not only accelerates the training process and allows training of deeper DNNs or DCNNs, but it also enables training with higher learning rates without divergence risks. The development of new frameworks has allowed an implemented flexible convolutional layer to be divided into sublayers [28], in which adding batch normalization and implementing different activations are possible, as shown in **Fig. 1**.

However, DCNN models such as those used in image classification tasks [13] barely achieve an acceptable accuracy level on SSL tasks because of background noise, moving sources, and room reverberations. To obtain a *robust* implementation, in our study we add convolutional layers trained with residual learning.

2.2. Residual Learning

DCNN models trained with residual learning not only perform better but also can speed up the learning process and improve the loss convergence in the training. Recent research has revealed that the network depth is critical during challenging tasks. Deeper models are not only useful in classification tasks such as ImageNet dataset classification but also greatly benefit other non-trivial vision computing tasks. *The deeper a network is, the easier the task becomes* [14]. Vanishing or exploding gradients presents an obstacle for implementing *very deep networks*, but this problem has been addressed by normalizing intermediate layers or normalizing initialization, thus allowing training convergence with the use of SGD optimization. With increasing network depth, accuracy training saturates and then degrades rapidly, thus exposing this as a degradation problem. Deep residual networks (DRN) were introduced in [14] to address this degradation. Residual learning can be denoted as:

$$F(x) := H(x) - x, \quad \ldots\ldots\ldots\ldots \quad (3)$$

where $H(x)$ is the desired underlying mapping. Residual learning allows the layers to fit a residual mapping $F(x)$ instead of hoping that each few stacked layer directly fits the desired underlying map. The original mapping is then reformulated as:

$$H(x) = F(x) + x. \quad \ldots\ldots\ldots\ldots \quad (4)$$

This formulation can be implemented by a feedforward neural network using shortcut connections (**Fig. 2**). A shortcut is presented as an identity mapping, which skips one or more layers. In addition, because it neither computationally complex nor requires additional parameters, it can be trained end-to-end using a common library [28].

To decrease the learning cost, a deeper bottleneck was also introduced in [14]. Here, a stack of three layers was used for each residual function F; therefore, to reduce and restore the dimensions, 1×1, 3×3, and 1×1 kernel for each of the three convolutional layers were used.

2.3. Nonlinear Activation Functions

Novel nonlinearity activation functions have been implemented to improve the accuracy performance of DC-

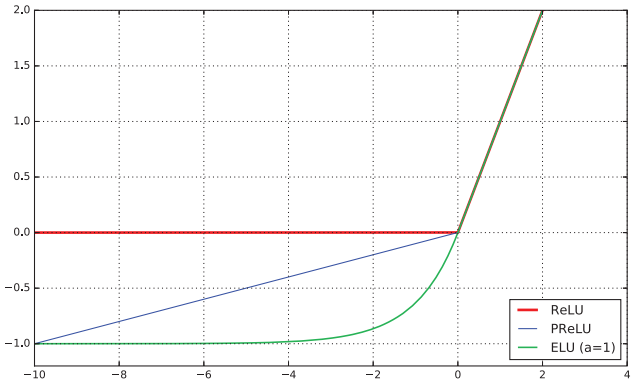

Fig. 3. Activation function output.

NNs. Even in challenging conditions with higher noise, models implemented with the novel activation functions perform better than conventional methods.

The introduction of novel nonlinear activation functions has improved the accuracy and performance of DCNNs (**Fig. 3**) [18, 29], thus allowing DCNNs to surpass human classification performance. In contrast to conventional sigmoid-like activations, rectifier neurons (e.g., rectified linear unit (ReLU)) nonlinearities have been used with considerable success for computer vision [13] and sound tasks [30, 31]. The ReLU activation function, which is defined as:

$$f(x) = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}, \quad \cdots\cdots\cdots \quad (5)$$

has led to better solutions. However, because the activation outputs are non-negative, the mean activation is greater than zero. Thus, the function is not centered and slows down learning [32]. Therefore, to speed up learning and improve performance, a generalization of ReLU called *parametric rectified linear unit* (PReLU) was introduced in [18]. This study claimed that using PReLU in a DCNN can surpass human performance. PReLU is defined as:

$$f(x_i) = \begin{cases} x_i & \text{if } x_i \geq 0 \\ a_i x_i & \text{if } x_i < 0 \end{cases}, \quad \cdots\cdots\cdots \quad (6)$$

where a_i is a coefficient that controls the slope of the negative part and can be a learnable parameter or fixed value. PReLU can improve model accuracy by including negative values in the activation output, with minimal overfitting risk at negligible additional computational cost. Nevertheless, PReLU does not ensure a robust deactivation state. Exponential linear unit (ELU) [29] was introduced as a nonlinearity that can improve learning characteristics compared with other linear activation functions. ELU is defined as:

$$f(x) = \begin{cases} x & \text{if } x \geq 0 \\ a(exp(x) - 1) & \text{if } x < 0 \end{cases}, \quad \cdots\cdots \quad (7)$$

where *exp* represents the exponential function. A network implemented with ELU activations considerably accelerates learning and show more robust generalization performance than ReLUs and PReLUs.

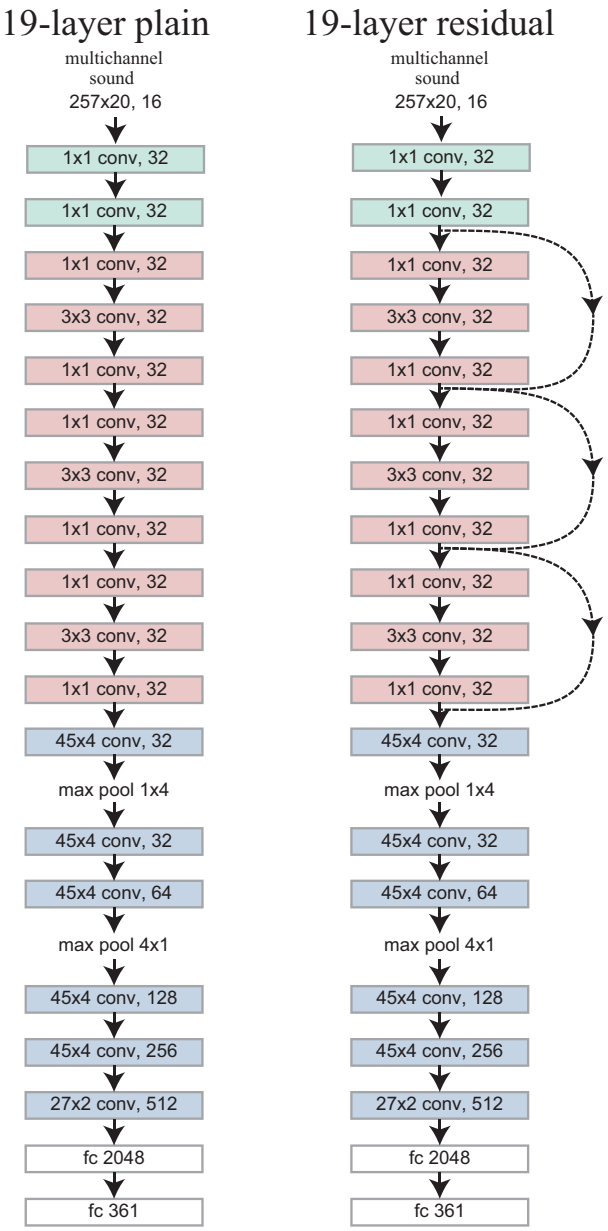

Fig. 4. Network architecture. Left: plain network. Right: residual network. The dotted line shortcut represents a 1×1 convolutional layer.

2.4. Model Architecture

DCNNs and convolutional layers trained for residual learning can improve the performance and robustness of a plain network without additional learning costs. Therefore, we implement models (**Fig. 4**) with 1–4 residual blocks, named ResNet 1–4, respectively (see **Table 1**), and train them by means of supervised learning. After the input, two 1×1 kernel convolutional layers with TanH and ELU are stacked and then connected to the residual blocks. We also evaluate larger models using only a convolutional layer after the input. According to [14], a

Table 1. ResNet architecture for SSL.

Layer Name	Output Size	ResNet1 19 Layers	ResNet2 28 Layers	ResNet3 36 Layers	ResNet4 45 Layers
conv1_x	257×20	1×1, 32 TanH 1×1, 32 ELU	1×1, 32 TanH 1×1, 32 ELU	1×1, 32 TanH	1×1, 32 TanH
conv2_x	257×20	$\begin{bmatrix} 1\times1,32 \\ 3\times3,32 \\ 1\times1,32 \end{bmatrix} \times3 \times1$ *ELU*	$\begin{bmatrix} 1\times1,32 \\ 3\times3,32 \\ 1\times1,32 \end{bmatrix} \times3 \times2$ *TanH/ELU*	$\begin{bmatrix} 1\times1,32 \\ 3\times3,32 \\ 1\times1,32 \end{bmatrix} \times3 \times3$ *ELU*	$\begin{bmatrix} 1\times1,32 \\ 3\times3,32 \\ 1\times1,32 \end{bmatrix} \times3 \times4$ *ELU*
conv3	213×17	45×4, 32, stride 1			
conv4	169×11	1×4 max pool, stride 1 45×4, 64, stride 1			
conv5	125×8	45×4, 64, stride 1			
conv6	78×5	4×1 max pool, stride 1 45×4, 128, stride 1			
conv7	34×2	45×4, 256, stride 1			
conv8	8×1	27×2, 512, stride 1			
	1×1	2048-d fc, ELU			
	1×1	361-d fc, SoftMax			

deeper bottleneck can accelerate up learning. Therefore, we use three bottlenecks as a residual block, in which each bottleneck contains a set of three layers (**Fig. 2** right). We then add a plain network with six convolutional layers having an empirical-sized kernel, which we evaluated prior to conducting experiments. The empirical kernel size is set to 45×4 dims for the first five layers and then a 27×2 kernel convolution layer is stacked. With this configuration, a ResNet with one residual block is implemented with 19 layers.

Each convolutional layer is followed by a batch normalization [27] layer, and an ELU [29] activation is used for the rest of the network. A TanH activation is evaluated on the first residual block of the ResNet with two residual blocks. A max-pooling layer with a kernel size of 1×4 is used after the first 45×4 kernel convolutional layer, and another max-pooling layer with a kernel size of 4×1 is used after the third 45×4 convolutional layer. To evaluate the performance of residual learning, we also train a plain network (PlainNet) with the same number of layers and iterations as those in ResNet1.

3. Experiments

3.1. Data Preparation and Preprocessing

For training and evaluation, we used the Acoustic Society of Japan-Japanese Newspaper Article Sentence (ASJ-JNAS) corpora, which include Japanese utterances of different lengths from 216 speakers for training and those from 42 speakers for evaluation. To prepare the training dataset, we used impulse responses from a HEARBO robot (**Fig. 5** left) equipped with a microphone array of 8 and 16 channels obtained from a 4×7 m room with 200 ms of reverberation every 5°. A clean single-channel utterance was convoluted with the multiple-channel impulse response of a random angle and white noise was added to each channel in a range between clean data and -30 dB of SNR. The noise added to each channels was different to simulate a real environmental noise. The input data for the network was prepared using short-time Fourier transform (STFT), and a label pointing to the angle or silence was set as the target. The input was preprocessed according the following steps:

- A normalized N-channel audio with a 16-kHz sampling rate was transformed into STFT features, thus extracting a frame with a length of 400 samples (25 ms) and a hop of 160 samples (10 ms). The length and hop values were based on previous research on speech tasks [22].
- From the STFT, we used only the power information. The power was normalized on the W frequency bin axis, in a range between 0.1 and 0.9.
- Because one STFT frame from the audio contains corrupt information due to noise, we stacked H frames. Thus, the input of the network became a $N \times W \times H$ dimensional file.

Regarding the output, we prepared the angle label as follows (**Fig. 6**):

- From the clean audio used to prepare the input multiple-channel audio, an STFT file with the same dimensions as those of the input but with only one channel ($1 \times W \times H$) was obtained.
- From this file, we evaluated the root mean squared (RMS) power of all frames.

Fig. 5. Array of microphones. Left: HEARBO robot. Right: microcone.

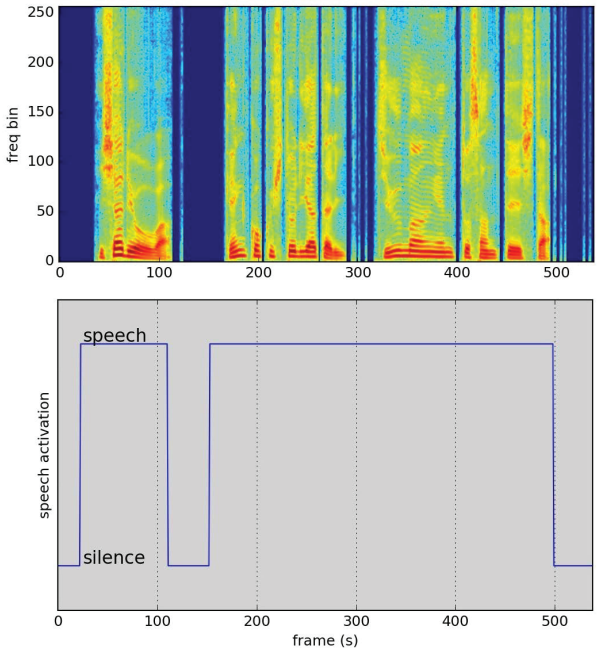

Fig. 6. Output preprocessing. Top: STFT clean audio. Bottom: speech-angle activation label.

- If several frames F have a RMS value of -120, the target label pointed to the silence label index. Otherwise, the target label was set to the angle used to prepare the input audio.

In addition, to evaluate the performance of residual learning, we trained a model using the impulse responses from a microcone (**Fig. 5** right) in an environment with many reflections (**Fig. 7**). The microcone is a microphone array of seven channels and was manufactured by BIAMP.

3.2. Training Method

For our experiments, we used audio data employing STFT to prepare multiple training files, the input for which was an array having dimensions of 7/8/16 $\times 257 \times$ 20 (i.e., audio channels, frequency bins, frames); the corresponding label output had an integer value. From each audio file, a random number of prepared files were selected. To maintain a uniform distribution between the angle targets, 33750 files were selected for each angle. We thus prepared a dataset having 2430000 files. For the distribution, we mixed the files with sound and no-sound information. Simultaneously, a noise with a random SNR level was added to each audio file. The SNR values of the input audio were selected from Clean, 30, 10, 5, 0, -5, -10, -20, and -30 dB.

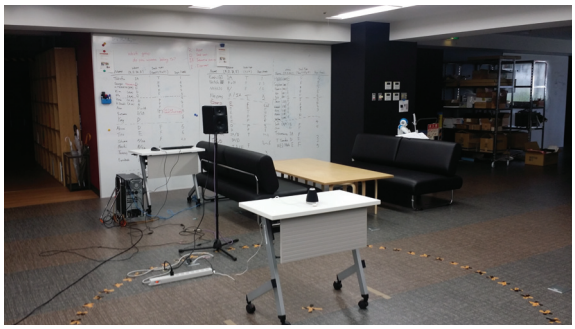

Fig. 7. Microcone impulse response room. *Reverberation time* was approximately 500 ms.

We trained six end-to-end networks using ADAM solver [33] and SoftMax with cross entropy as the loss function. No finetune or previous training was used for the experiments. The initial alpha for the solver was set to 10^{-4}, and the target-label was set from 0 to 359°. The output of the network was set to 361 dimensions, where the 0 to 359 outputs were activated with their respective angles labels, and the 360th output was activated when the input data contained no-audio information. The value of F for evaluating the input no-audio data was empirically set to 13 frames. As a result, the network not only could locate the sound source angle, but it also ensured that the network activated a correct output in the absence of sound information. For the experiments, we did not use the dropout function on the fully connected layers. We trained shorter models for five epochs and larger models for two epochs, using a mini batch of only 70 files. Each epoch required approximately 24 h using a GPU NVIDIA Titan X.

3.3. Evaluation Criteria

We evaluated the networks using two difference measures. First, we evaluated the accuracy of an audio file despite the number of frames generated from it. In this *block accuracy* evaluation, we forwarded the inputs file that was preprocessed from an audio file, and then stacked the outputs. From the outputs, we evaluated the median angle with respect to the target angle of the file using a confusion matrix, and then calculated the mean accuracy of each network based on a corresponding SNR level. We did not consider whether the input file contains sound or

no-sound information. SEVD-MUSIC was implemented on HARK [34] and its result was used as a reference.

Second, we evaluated on a point-to-point basis each file obtained from an audio file. We not only evaluated the accuracy of the angle output, but the correct activation of the silence index at the absence of sound information in the input. For the evaluation, we used the detection rate as well as the accuracy rate formulation proposed in [35]. These were formulated and modified, respectively, as:

$$C = \left(\frac{N - D - S}{R} \right) \times 100\%, \quad \ldots\ldots\ldots \quad (8)$$

$$A = \left(\frac{N - D - S - I}{R} \right) \times 100\%, \quad \ldots\ldots\ldots \quad (9)$$

where C is the correct detection rate, A is the correct accuracy rate, N is the correct source positions, which is the number of sources located inside a fixed range of their correct positions, and S is the source position replacement or incorrect position, which is the number of sources located further from fixed range of their correct position. I is the number of incorrect insertions or ghost detections, D is the number of incorrect speech deletions or misdetections, and R is the number of frames of reference for the evaluation. In this evaluation, we compared the results between models.

For both evaluations, 50 audio files (25 from females, and 25 from males) were used and tested at intervals of 5° from 0° to 359°. The same white noise was added to all channels in an SNR range from the clean data up to −35 dB. Thus, the noise evaluated was different from that used in the training process. We checked robustness against the speaker, SNR, as well as the efficiency of shorter/longer training on deeper networks.

4. Results

In this section, we present the result of using DL models for SSL tasks. The models showed robust performance in challenging environments without additional learning cost. **Table 2** lists the configuration parameters for training and evaluation.

4.1. Training Process

Figure 8 presents the graphs from the training process of the models. We observed that a training using residual networks not only had a fast convergence compared to plain networks (**Fig. 8** top), but also that the learning loss was less. However, stacking additional residual blocks did not affect the learning process, neither speeding up the loss convergence (**Fig. 8** center) nor improving the initial accuracy of the training dataset (**Fig. 8** bottom).

4.2. Block Level Accuracy

We evaluated the block level accuracy for all networks at different SNR levels and compared it with SEVD-MUSIC as a reference. The confusion matrix for the evaluation of our experiments is shown in **Fig. 9**. Note that the block accuracy of the median angle was evaluated with a tolerance of ±2.5°.

Table 2. Experimental configuration.

Parameters	Value
Number of sources:	0 or 1
Sampling frequency:	16 kHz
Frame length and shift:	25 ms & 10 ms
Frames per input file:	20 frames
SNR:	45 dB ∼ −35 dB
Microcone reverberation time:	@500 ms
Microcone source distance:	1.5 m
Number of Microphones:	7
HEARBO reverberation time:	200 ms
Number of Microphones:	8 & 16
HEARBO source distance@8mic:	1 m & 1.5 m
HEARBO source distance@16mic:	1.5 m
Training Noise Signal:	Multiple Channel white noise
Training Speakers:	216 (male & female)
Test Noise Signal:	Same white noise
Test Speakers:	46 (male & female)

Table 3 shows the results of the block accuracy rate of all systems evaluated for HEARBO robot. The first three columns show comparisons when using a 10-layer PlainNet at different distances and using 8 and 16 channels. We observed that the accuracy improved when the number of channels increased. In addition, we could achieve a better performance when the distance to the array was closer. However, the accuracy was not better than that of SEVD-MUSIC. We then observed that larger deep-learning-based SSL performed better than did SEVD-MUSIC. However, we also determined that not only does residual learning perform better than do plain networks, but that stack additional residual blocks is not required, whereas a different activation function is needed to obtain a good performance. ResNet1 showed better performance on SNR levels higher than 0, and an acceptable performance remained until −10 dB. The performance then diminished quickly.

The last two columns of **Table 3** show the results of the block accuracy rate for models trained for two epochs. The trained models performed better compared to SEVD-MUSIC, and the best accuracy between models alternated based on the SNR level. Until an SNR level of −5 dB was reached, ResNet3 showed better results than that of the ResNet4. On lower SNR levels, ResNet4 showed a slightly better result. However, both of these models, even when were deeper than others, did not perform better compared to a model trained for additional epochs.

For a better observation of the network behavior, we present the confusion matrices of ResNet4 in **Fig. 9**. We observed that on clean environments, the model more accurately predicted the angle (**Fig. 9** top). However, we observed that at an SNR level of −35 dB, the model predicted nearly all the angles as silences. In this case, the models showed that when determining whether the input

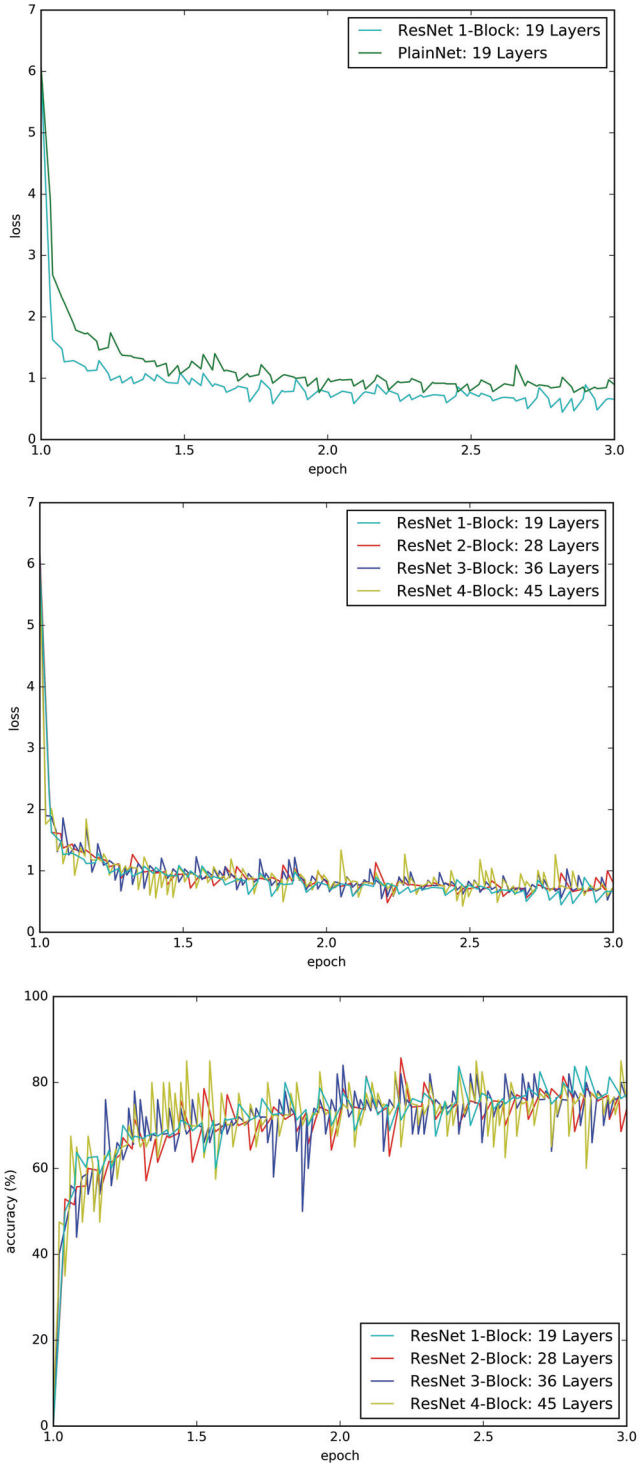

Fig. 8. Training graphs. Top: PlainNet and ResNet1 training loss. Center: ResNets training loss. Bottom: ResNets training accuracy.

has sound or no-sound information was not possible because of the noise level, the model fixed the output to the silence index, offering stability on silence or very noisy environments.

Table 4 shows the comparison of SEVD-MUSIC and a DCNN model that uses an array of microphones of 7 channels in an environment having greater reflections.

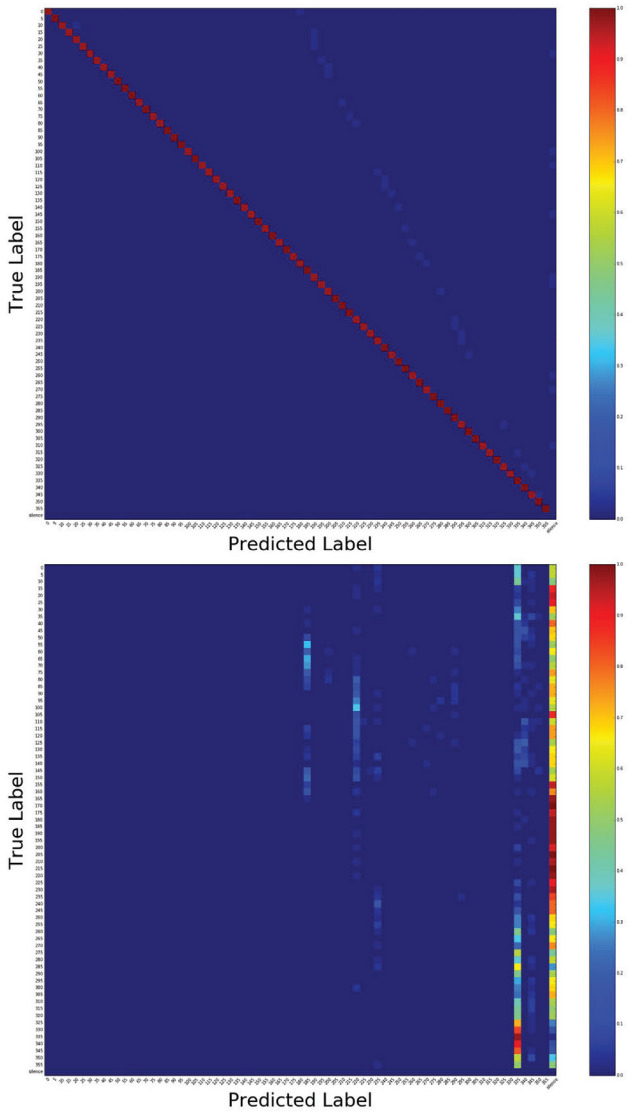

Fig. 9. ResNet4 angle prediction confusion matrix. Top: Clean. Bottom: −35 dB SNR.

The table indicates that SEVD-MUSIC performed better than ResNet1. However, with a reduced number of channels and the use of power information, ResNet1 could calculate the DOA as effectively as does the SEVD-MUSIC.

4.3. Point-to-Point Evaluation

We evaluated the detection and accuracy rate performance of the models using the evaluation based on the formulation presented in [35]. In this evaluation, we compared the output of the models to the true angle prepared in advance. Here, we considered not only the angle of the source, but whether the models could detect sound or silence information at the input. The fixed range for a correct position was set to $\pm 20^\circ$.

Table 5 shows the detection rate of the models and **Table 6** shows their accuracy rates. We observed that ResNet1 had a better result than others and performed well on the accuracy rate until −10 dB (i.e., it rose to

Table 3. HEARBO block accuracy.

SNR (dB)	PlainNet 8Ch 10 Layers /1.0m	PlainNet 8Ch 10 Layers /1.5m	PlainNet 16Ch 10 Layers /1.5m	SEVD-MUSIC 16 Channels	PlainNet	ResNet1	ResNet2 TanH/ELU	ResNet3	ResNet4
−35	0.00	0.00	0.08	1.39	0.03	2.45	1.94	1.39	1.47
−30	0.00	0.00	0.08	1.39	0.17	3.50	2.67	1.47	1.92
−25	0.00	0.00	0.08	1.39	1.06	6.94	5.33	2.08	3.33
−20	0.06	0.00	0.08	1.39	4.86	17.19	13.86	4.81	7.78
−15	0.06	1.42	2.42	1.39	19.14	36.28	31.94	11.44	20.72
−10	0.06	1.94	5.58	3.64	35.78	60.31	53.03	31.75	31.78
−5	2.94	7.92	14.31	16.42	55.53	80.06	71.14	55.86	44.69
0	14.19	28.78	34.17	40.11	70.58	90.08	82.08	75.58	58.89
5	37.00	54.14	68.89	66.75	80.17	95.25	88.97	86.89	74.92
10	63.89	80.89	87.89	84.67	88.33	97.53	93.11	93.22	88.17
15	91.00	91.04	93.61	90.69	94.28	98.83	96.44	97.42	96.03
30	98.56	98.77	98.92	96.61	99.58	99.61	99.61	99.69	99.72
45	98.28	98.67	98.89	98.51	99.58	99.53	99.58	99.58	99.78
Clean	98.53	98.61	98.72	98.53	99.14	99.14	99.36	99.25	98.89

Table 4. Microcone (array of 7 microphones) block accuracy.

SNR (dB)	SEVD-MUSIC	ResNet1
−35	**1.39**	1.39
−30	**1.39**	1.39
−25	**1.39**	1.39
−20	**1.42**	1.39
−15	**2.64**	2.06
−10	**9.10**	6.42
−5	**21.16**	16.94
0	**36.12**	33.61
5	**50.99**	49.69
10	**66.75**	64.61
15	**81.26**	78.11
30	**97.42**	95.36
45	**98.01**	97.39
Clean	**98.17**	97.50

Table 5. HEARBO detection rate.

SNR (dB)	Plain Network	ResNet1	ResNet2 TanH/ELU	ResNet3	ResNet4
−35	−86.13	13.16	−8.26	−10.59	−47.06
−30	−80.72	15.92	−3.43	−8.92	−19.47
−25	−71.65	21.11	8.75	−1.08	0.72
−20	−48.6	32.64	28.64	−4.02	14.29
−15	0.18	53.23	49.37	7.18	27.59
−10	34.45	68.95	59.53	36.74	27.65
−5	51.93	74.93	64.16	55.00	25.23
0	61.73	76.26	66.8	63.04	32.04
5	67.69	76.19	70.02	68.93	46.94
10	71.99	76.75	72.66	73.58	61.84
15	75.55	78.74	75.08	76.98	72.17
30	81.21	82.82	82.44	83.26	83.03
45	84.22	84.94	85.54	85.24	85.33
Clean	84.14	84.75	85.23	85.11	84.66

greater than 50% and then fell). These results are very similar to those of the block accuracy evaluation, which can be used as a reference for future works.

Most of the models showed high value negative accuracy at lower SNR levels. Thus, the networks generated replacement of the location or the detected ghost sources.

5. Discussion

5.1. Reverberation Rooms

When distant microphones were used to capture an audio stream, the signals were affected by environmental noise and the room's reverberation. This *observation model* can be expressed as follows [9]:

$$x(\omega) = D(\omega)s(\omega) + n(\omega). \quad \ldots\ldots\ldots (10)$$

where $x(\omega)$ is an observed microphone signal vector with M observations at frequency ω denoted $[x_1(\omega)x_2(\omega)\cdots x_M(\omega)]^T$; $D(\omega)$ is a transfer function matrix between the array of microphones and a sound source; $s(\omega)$ is a clean speech signal vector with N sources at frequency ω denoted as $[s_1(\omega)s_2(\omega)\cdots s_N(\omega)]^T$, where T represents a transpose operator; and $n(\omega)$ is a noise vector with diffuse and dynamically changing colored noise. Note that $n(\omega)$ is statistically independent of $s(\omega)$.

Table 3 suggests that the DL model also uses the reverberation of the room or that having an input with a similar or higher time dimension could improve the SSL. We compared HEARBO arrays with eight microphones at different distance using the same reverberation time (**Table 3**). In this case, the accuracy of the performance was reduced when the source was closer. However, **Fig. 10**

Table 6. HEARBO accuracy rate.

SNR (dB)	PlainNet	ResNet1	ResNet2 TanH/ELU	ResNet3	ResNet4
−35	−99.40	−0.05	−21.47	−17.24	−60.15
−30	−93.91	2.83	−16.69	−16.03	−32.75
−25	−84.83	7.97	−4.35	−12.15	−12.46
−20	−61.85	19.46	15.38	−4.30	1.08
−15	−13.04	40.05	36.14	−2.69	14.3
−10	21.18	55.81	46.15	23.60	14.58
−5	38.68	61.73	50.88	41.70	11.98
0	48.43	62.96	53.53	49.75	18.81
5	54.51	63.07	56.80	55.80	33.72
10	58.74	63.51	59.45	60.42	48.63
15	62.40	65.50	61.94	63.77	59.06
30	68.06	69.60	69.24	70.08	69.82
45	71.04	71.70	72.30	72.04	72.11
Clean	70.89	71.53	72.01	71.90	71.46

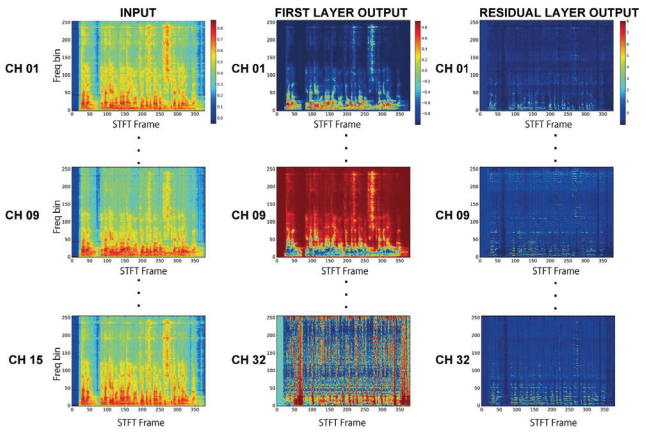

Fig. 10. ResNet1 output features. Left: input data. Center: conv1 output. Right: conv2 output.

shows that DL models used most power from the direct sound and compared the level of that sound between the channels rather than measuring the power of reverberations at each channel. **Fig. 10** presents the features from the *conv1* and the residual block *conv2* of ResNet1 and shows that after training, the model boosted the power from the direct sound. In addition, after *conv2*, the additional information (i.e., reverberation or noise) was nearly reduced to zero.

After comparing the model with the others given in **Table 4**, where the reverberation time of the former is greater (approximately 500 ms), we observed that the performance of the model also decreased. In this case, it is possible that the model required a longer window or more stacked frames at the input to obtain a similar length to match the reverberation time. However, increasing the hyperparameters of the DL models becomes necessary, thereby increasing both the learning process the real-time processing.

5.2. Processing Time

Processing time is a major parameter to consider if we want to implement a DCNN-based SSL system in real-time. However, using a GPU for evaluation reduces the processing time at levels that is possible to employ DCNN-based systems in real-time applications. During our evaluation, the processing time increased with the number of files used in the forwarding stage. We evaluated the time from the moment the data was transferred from the central processing unit (CPU) to the GPU up to the moment that the model returned the result to the CPU. The average time for all models did not exceed 10 ms per file batch.

We also evaluated the processing time on a CPU Intel Core i7@3.2 GHz. From the moment that we inputted the data up to the moment we obtained the result, the processing time did not exceed 350 ms for a real-time evaluation using the 19-layered ResNet model.

5.3. Untrained Conditions

We showed that the DCNN can adapt to different levels of SNR and localize the target. The localization accuracy was not affected when the noise was different from that used in the training. However, we observed that during the test, for unknown conditions such as higher or lower SNR used in the training, the model's accuracy diminished. Thus, using different (i.e., a wide range of) levels of SNR during the training phase is required to maintain good accuracy performance.

6. Conclusion

In this study, we proposed an SSL method that use a DRN to exploit the time delay between channels to predict the DOA. *Substituting* MUSIC for a DRN model using ELU activations for SSL not only reduced the *learning cost* but made the performance *more robust* with untrained noises at lower SNR. We showed that not only can a DCNN generate better block accuracy than can SEVD-MUSIC in noisy environments, but that a DRN with ELU performs better than a DCNN plain network that has the same number of layers. The performance of the networks did not diminish even when a different noise was used (i.e., from that used by the networks during training). However, stacking additional residual layers did not improve either the convergence of the loss nor the accuracy rate on deeper models during a short training period. DRN models can have an acceptable performance during a shorter training period, even when models are *very deep*. However, to obtain a robust performance in challenging environments, the models require additional learning cost.

We plan to study residual learning in the application of multiple sources for SSL and improve the accuracy on environments with greater reflections and fewer microphones. In addition, we plan to implement sound source separation tasks using the employed models for SSL as future works.

Acknowledgements

The work has been supported by MEXT Grant-in-Aid for Scientific Research (A) 15H01710.

References:

[1] K. Nakadai, T. Lourens, H. G. Okuno, and H. Kitano, "Active audition for humanoid," Proc. of the National Conf. on Artificial Intelligence, pp. 832-839, 2000.
[2] K. Nakadai, H. Nakajima, Y. Hasegawa, and H. Tsujino, "Sound source separation of moving speakers for robot audition," Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), pp. 3685-3688, 2009.
[3] R. O. Schmidt, "Multiple emitter location and signal parameter estimation," IEEE Trans. on Antennas and Propagation, Vol.34, No.3, pp. 276-280, 1986.
[4] K. Nakamura, K. Nakadai, F. Asano, Y. Hasegawa, and H. Tsujino, "Intelligent Sound Source Localization for Dynamic Environments," 2009 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 664-669, 2009.
[5] B. D. Rao and K. V. S. Hari, "Performance Analysis of Root-Music," IEEE Trans. on Acoustics, Speech, and Signal Processing, Vol.37, No.12, pp. 1939-1949, 1989.
[6] K. Nakadai, K. Hidai, H. G. Okuno, H. Mizoguchi, and H. Kitano, "Real-time Auditory and Visual Multiple-speaker Tracking For Human-robot Interaction," J. of Robotics and Mechatronics, Vol.14, No.5, pp. 479-489, 2002.
[7] J. Valin, J. Rouat, and L. Dominic, "Robust Sound Source Localization Using a Microphone Array on a Mobile Robot," Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), pp. 1228-1233, 2003.
[8] Y. Sasaki, M. Kaneyoshi, and S. Kagami, "Pitch-Cluster-Map Based Daily Sound Recognition for Mobile Robot Audition," J. of Robotics and Mechatronics, Vol.22, No.3, 2010.
[9] K. Nakadai, G. Ince, K. Nakamura, and H. Nakajima, "Robot audition for dynamic environments," 2012 IEEE Int. Conf. on Signal Processing, Communications and Computing (ICSPCC 2012), pp. 125-130, 2012.
[10] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep Neural Networks for Acoustic Modeling in Speech Recognition," IEEE Signal Processing Magazine, pp. 82-97, November 2012.
[11] J. Platt and L. Deng, "Ensemble deep learning for speech recognition," Proc. Interspeech, 2014.
[12] R. Socher, C. C. Lin, A. Y. Ng, and C. D. Manning, "Parsing natural scenes and natural language with recursive neural networks," Proc. of the 26th Int. Conf. on Machine Learning (ICML), pp. 129-136, 2011.
[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," Advances in Neural Information Processing Systems 25 (NIPS2012), pp. 1-9, 2012.
[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," Arxiv.Org, Vol.7, No.3, pp. 171-180, 2015.
[15] R. Takeda and K. Komatani, "Sound Source Localization based on Deep Neural Networks with directional activate function exploiting phase information," ICASSP, pp. 405-409, 2016.
[16] T. Hirvonen, "Classification of spatial audio location and content using convolutional neural networks," Audio Engineering Society Convention 138, 2015.
[17] D. Pavlidi, A. Grif, M. Puigt, and A. Mouchtaris, "Real-Time Multiple Sound Source Localization and Counting Using a Circular Microphone Array," IEEE Trans. on Audio, Speech and Language Processing, Vol.21, No.10, pp. 2193-2206, 2013.
[18] K. He, X. Zhang, S. Ren, and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," arXiv preprint, pp. 1-11, 2015.
[19] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, B. León, Y. Bengio, P. Haffner, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-Based Learning Applied to Document Recognition," Proc. of IEEE, Vol.86, No.11, p. 86, 1998.
[20] Y. Lecun and M. A. Ranzato, "Deep Learning Tutorial," Proc. of 30th Int. Conf. on Machine Learning (ICML), 2013.
[21] C. J. C. B. John Platt, S. Sukittanon, A. C. Surendran, J. C. Platt, C. J. C. Burges, and B. Look, "Convolutional Networks for Speech Detection," Int. Speech Communication Association, pp. 2-5, 2004.
[22] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, Long Short-Term Memory, fully connected Deep Neural Networks," Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), pp. 4580-4584, 2015.
[23] O. Abdel-hamid, H. Jiang, and G. Penn, "Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition," 2012 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), pp. 4277-4280, 2012.
[24] T. N. Sainath, B. Kingsbury, G. Saon, H. Soltau, A.-r. Mohamed, G. Dahl, and B. Ramabhadran, "Deep Convolutional Neural Networks for Large-scale Speech Tasks," Neural Networks, Vol.64, pp. 39-48, 2015.
[25] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," Jmlr W&Cp, Vol.28, No.2010, pp. 1139-1147, 2013.
[26] J. Duchi, E. Hazan, and Y. Singer, "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization," J. of Machine Learning Research, Vol.12, pp. 2121-2159, 2011.
[27] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," arXiv, 2015.
[28] S. Tokui, "Introduction to Chainer: A Flexible Framework for Deep Learning," PFI/PFN Weekly Seminar, 2015.
[29] D. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)," Under review of ICLR2016 (1997), pp. 1-13, 2015.
[30] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," ICML Workshop on Deep Learning for Audio, Speech and Language Processing, Vol.28, 2013.
[31] G. E. Dahl, T. N. Sainath, and G. E. Hinton, "Improving deep neural networks for LVCSR using rectified linear units and dropout," Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), pp. 8609-8613, 2013.
[32] Y. Lecun, I. Kanter, and S. A. Solla, "Eigenvalues of covariance matrices: application to neural-network learning," Physical Review Letters, Vol.66, No.18, pp. 2396-2399, 1991.
[33] D. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," arXiv:1412.6980 [cs], pp. 1-15, 2014.
[34] K. Nakadai, T. Takahashi, H. G. Okuno, H. Nakajima, Y. Hasegawa, and H. Tsujino, "Design and Implementation of Robot Audition System 'HARK' – Open Source Software for Listening to Three Simultaneous Speakers," Advanced Robotics, Vol.24, No.5-6, pp. 739-761, 2010.
[35] T. Takahashi, K. Nakadai, C. H. Ishii, E. Jani, and H. G. Okuno, "Study of sound source localization, sound source detection of a real environment," The 29th Annual Conf. of the Robotics Society of Japan, 2011.

Name:
Nelson Yalta

Affiliation:
Master Course Student, Intermedia Art and Science Department, Waseda University

Address:
3-4-1 Okubo, Shinjuku-ku, Tokyo 169-8555, Japan

Brief Biographical History:

2009 Received Bachelor in Electronic Engineering, Universidad Privada Antenor Orrego
2014- Embassy Recommendation, MEXT Scholarship awarded
2014- Research Student, Waseda University
2015- Master Course, Intermedia Art and Science Department, Waseda University
2015- Graduate Program for Embodiment Informatics, Waseda University

Main Works:

• "Multiple Input Audio Denoising Using Deep Neural Networks," The 33rd Annual Conf. of the Robotics Society of Japan, September 2015.
• "Sound Source Localization Using Deep Residual Networks," The 34th Annual Conf. of the Robotics Society of Japan, September 2016.

Membership in Academic Societies:

• The Robotics Society of Japan (RSJ)
• The Institute of Electrical and Electronic Engineers (IEEE)

Name:
Kazuhiro Nakadai

Affiliation:
Honda Research Institute Japan Co., Ltd.
Tokyo Institute of Technology

Address:
8-1 Honcho, Wako-shi, Saitama 351-0188, Japan
2-12-1-W30 Ookayama, Meguro-ku, Tokyo 152-8552, Japan

Brief Biographical History:
1995 Received M.E. from The University of Tokyo
1995-1999 Engineer, Nippon Telegraph and Telephone and NTT Comware
1999-2003 Researcher, Kitano Symbiotic Systems Project, ERATO, JST
2003 Received Ph.D. from The University of Tokyo
2003-2009 Senior Researcher, Honda Research Institute Japan Co., Ltd.
2006-2010 Visiting Associate Professor, Tokyo Institute of Technology
2010- Principal Researcher, Honda Research Institute Japan Co., Ltd.
2011- Visiting Professor, Tokyo Institute of Technology
2011- Visiting Professor, Waseda University

Main Works:
• K. Nakamura, K. Nakadai, H. and G. Okuno, "A real-time super-resolution robot audition system that improves the robustness of simultaneous speech recognition," Advanced Robotics, Vol.27, Issue 12, pp. 933-945, 2013 (Received Best Paper Award).
• H. Miura, T. Yoshida, K. Nakamura, and K. Nakadai, "SLAM-based Online Calibration for Asynchronous Microphone Array," Advanced Robotics, Vol.26, No.17, pp. 1941-1965, 2012.
• R. Takeda, K. Nakadai, T. Takahashi, T. Ogata, and H. G. Okuno, "Efficient Blind Dereverberation and Echo Cancellation based on Independent Component Analysis for Actual Acoustic Signals," Neural Computation, Vol.24, No.1, pp. 234-272, 2012.
• K. Nakadai, T. Takahashi, H. G. Okuno et al., "Design and Implementation of Robot Audition System "HARK"," Advanced Robotics, Vol.24, No.5-6, pp. 739-761, 2010.
• K. Nakadai, D. Matsuura, H. G. Okuno, and H. Tsujino, "Improvement of recognition of simultaneous speech signals using AV integration and scattering theory for humanoid robots," Speech Communication, Vol.44, pp. 97-112, 2004.

Membership in Academic Societies:
• The Robotics Society of Japan (RSJ)
• The Japanese Society for Artificial Intelligence (JSAI)
• The Acoustic Society of Japan (ASJ)
• Information Processing Society of Japan (IPSJ)
• Human Interface Society (HIS)
• International Speech and Communication Association (ISCA)
• The Institute of Electrical and Electronics Engineers (IEEE)

Name:
Tetsuya Ogata

Affiliation:
Professor, Faculty of Science and Engineering, Waseda University

Address:
3-4-1 Okubo, Shinjuku-ku, Tokyo 169-8555, Japan

Brief Biographical History:
1997- Research Fellow, The Japanese Society for the Promotion of Science (JSPS)
1999- Research Associate, Waseda University
2001- Research Scientist, Brain Science Institute, RIKEN
2003- Lecturer, Kyoto University
2005- Associate Professor, Kyoto University
2012- Professor, Waseda University

Main Works:
• "Repeatable Folding Task by Humanoid Robot Worker using Deep Learning," IEEE Robotics and Automation Letters (RA-L), Vol.2, No.2, pp. 397-403, 2016.
• "Dynamical Integration of Language and Behavior in a Recurrent Neural Network for Human-Robot Interaction," Frontiers in Neurorobotics, July 15, 2016.
• "Multimodal Integration Learning of Robot Behavior using Deep Neural Networks," Robotics and Autonomous Systems, Vol.62, No.6, pp. 721-736, 2014.

Membership in Academic Societies:
• The Robotics Society of Japan (RSJ)
• The Japanese Society for Artificial Intelligence (JSAI)
• The Japan Society of Mechanical Engineers (JSME)
• Information Processing Society of Japan (IPSJ)
• The Society of Instrument and Control Engineers (SICE)
• Society of Biomechanisms Japan
• The Institute of Electrical and Electronic Engineers (IEEE)