Multiple-Embedding Separation Networks: Sound Class-Specific Feature Extraction for Universal Sound Separation

Hokuto Munakata *, Ryu Takeda * and Kazunori Komatani *
* Osaka University, The Institute of Scientific and Industrial Research (SANKEN) Osaka, Japan
E-mail: h_munakata@ei.sanken.osaka-u.ac.jp, {rtakeda, komatani}@sanken.osaka-u.ac.jp

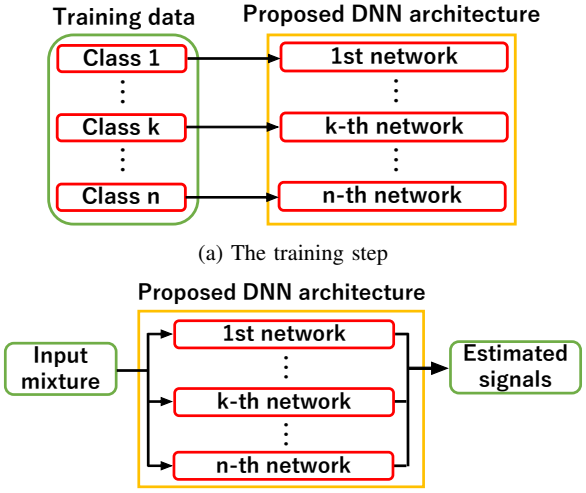
***Abstract*—We propose a novel deep neural network (DNN) architecture for universal sound separation. This task aims to separate monaural mixture signals containing various sounds into the corresponding source sound signals (e.g., speech, barking of a dog, etc.). Previous studies used a speech separation network and made mixtures for datasets by sampling dry source signals randomly from a database. These methods did not use sound class labels, although they are available during the training step. We propose Multiple-Embedding Separation Networks, an architecture using sound class labels in a training step. This architecture contains multiple feature-extraction networks that are further specialized for each sound class. Each network is trained together with the class labels to capture characteristics as class-specific embeddings. We evaluated the performance of our proposed method and a method commonly used for universal sound separation as a baseline. This evaluation adopted a dataset containing six classes. As a result, the proposed method outperformed the baseline in terms of the average separation performance by 0.22 dB, especially for speech mixtures by 2.28 dB. We found a complex relationship between the amounts of the data of the dry source signal of each class and the separation performance.**

I. Introduction

A. Motivation

When observing a sound signal in an actual environment, some source signals may be overlapped as a mixture signal and interfere with each other. Our global goal is to separate such a mixture with various sounds (e.g., speech, barking of a dog, etc.) into corresponding sources in a monaural condition. This task is known as "universal sound separation" [1]–[3]. This technology will be applied in various systems using sound information. For example, it will improve the selective ability of assistive hearing devices, the performance of sound classification, and the efficiency of editing video or sound data.

In universal sound separation, deep neural network (DNN) methods have been applied. These methods achieved higher performance than NMF-based methods (e.g., [4], [5]) in monaural speech separation due to their high expressiveness. A separation network applying DNN extracts a sound feature from the input mixture and estimates masks to separate. Such networks need a large amount of training data, namely, pairs of the sources and the mixture. Kavalelov attempted to make a dataset from the Pro Sound Effects Library database containing encyclopedic samples of movie production recordings including, for example, sounds of animal calls, creaking doors, and musical instruments [1]. In addition, some researchers have approached this task with a dataset made in the same way [3], [6]. However, these works did not use the sound class labels explicitly during the training step, although they were given the labels, or the sound data were divided into chunks based on their sound characteristics in a database.



(a) The training step

(b) The estimation step

Fig. 1: The main concept of proposed method. In the training step, each network is trained together with the class. In the estimation step, all networks are used.

In this paper, we propose a novel deep neural network architecture called Multiple-Embedding Separation Networks. Figure 1 shows the main concept of our proposed method. By explicitly mapping networks to the class labels, our DNN architecture aims to learn information about the characteristics of the sound directly from the class labels. The architecture has multiple feature extraction networks called Embedding Networks. Each Embedding Network is trained together with the class labels to capture the characteristics as class-specific embeddings. As an initial learning, the networks trained individually using the dataset are divided by the sound class labels for a few epochs.

We evaluated the separation performance of our proposed method and TDCN++ [1] as a baseline method using a dataset

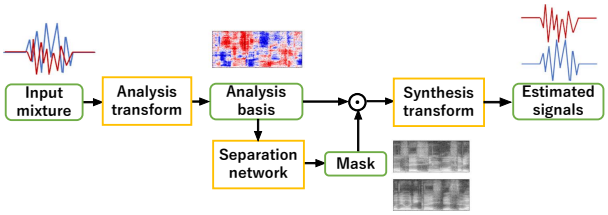


Fig. 2: An overview of the DNN-based speech separation model. DNN is used for a separation network.

containing six classes. To make analysis easier, the mixtures of the training data consist of combinations of the same class. As a result, the proposed method outperformed the baseline in terms of the average separation performance by 0.22 dB, especially for speech mixtures by 2.28 dB.

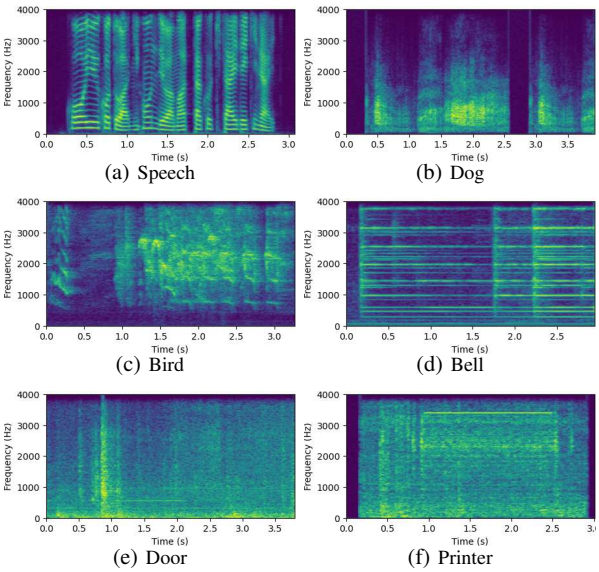
Our specific contributions are as follows:

1) We proposed a novel network architecture that improves the separation performance of the sound class with a large amount of data of the dry source signal.
2) We found that the separation performance for the class with a small amount of data of the dry source signal is supported by the learning results of the class with a large amount of the data.
3) We obtained the results that could lead to a deeper understanding of how networks capture characteristic of the sound data when dealing with multi-classes.

B. Related works

Various DNN-based methods have been proposed, and succeeded in monaural speech separation [7]–[15]. These methods have been developed through modeling the sound properly. The earliest proposed method, Deep Clustering [7] estimates separation masks using k-means in embeddings space. Deep Attractor Network provides a more sophisticated separation network by using an attractor vector as a perceptual effect in human speech perception [9]. TasNet replaced Short-Time Fourier Transform (STFT) with Encoder-Decoder transform [11]. Conv-TasNet replaced the LSTM layer of TasNet with the Temporal Convolutional Network (TCN) inspired by WaveNet [16]. Dual-Path RNN modeled extremely long sequences of sound with intra- and inter-chunk approaches. When training or evaluating models in speech separation, almost all cases use a benchmark dataset, such as WSJ0-2mix [7] or WHAM! [17]. These datasets include over 50 hours of the pairs of the sources and the mixture.

The main problem with these methods is their massive dataset requirements. If a method is applied to a task with only a small amount of the training data, it does not achieve adequate performance. In the monaural condition, the task is extremely difficult without the training data, unlike multi-channel conditions, such as blind source separation based on ICA or multi-channel NMF [18], [19]. There are a few methods that do not learn the separation task directly [15], but they still require a large amount of training data.



(a) Speech (b) Dog (c) Bird (d) Bell (e) Door (f) Printer

Fig. 3: Examples of data used in universal sound separation. Horizontal axis and vertical axis represent time bins and frequency bins respectively.

Less prior information on the characteristics of the sound is given in universal sound separation. The class labels provide one type of the sparse information available, so we should use them effectively.

II. Preliminary

A. Monaural Speech Separation with DNN

The purpose of monaural speech separation is to estimate C source sound signals $\mathbf{s}_1, \mathbf{s}_2, ...\mathbf{s}_C \in \mathbb{R}^{1\times T}$ from the observed mixture signal $\mathbf{s}_{mix} = \sum_{i=1}^{C} \mathbf{s}_i$, where T is the length of signals. With a DNN, the task is formulated as

$$[\hat{\mathbf{s}}_1, \hat{\mathbf{s}}_2, ...\hat{\mathbf{s}}_C] = \mathbf{f}_\theta(\mathbf{s}_{mix}), \tag{1}$$

where $\hat{\mathbf{s}}_1, \hat{\mathbf{s}}_2, ...\hat{\mathbf{s}}_C$ are estimated signals and $\mathbf{f}_\theta(\cdot)$ is a non-linear transformation of DNN with parameters represented by θ.

Figure 2 shows a conventional speech separation model. It is composed of three parts: an analysis transform, a separation network, and a synthesis transform. First, the input mixture is transformed to an analysis basis $\mathbf{w} \in \mathbb{C}^{N\times L}$ by an analysis transform, namely, Short-Time Fourier Transform (STFT) or Encoder Transform [11]. This transform is formulated as

$$\mathbf{w} = U(\mathbf{s}_{mix}), \tag{2}$$

where $U(\cdot)$ represents the analysis transform, N is the number of the analysis basis and L is length of segment. Second, the separation network estimates masks as many times as the sources from $\mathbf{w}$. The mask $\mathbf{m}_i \in \{m|m \in \mathbb{R}, 0 \leq m \leq 1\}^{N\times L}$ represents the dominance of the i-th sources in each element of $\mathbf{w}$. This estimation is formulated as

$$[\mathbf{m}_1, \mathbf{m}_2, ...\mathbf{m}_C] = M(\mathbf{w}), \tag{3}$$

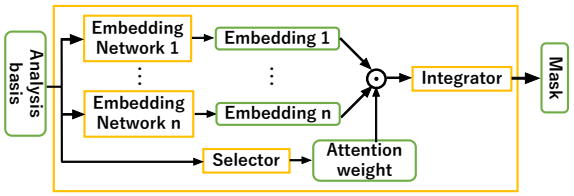


Fig. 4: The separation network of proposed method

where $M(\cdot)$ represents mask estimation transform by the separation network. Then, the masked analysis basis is formulated as $\mathbf{w} \odot \mathbf{m}_i$, where $\odot$ represents the Hadamard product. Third, the masked analysis basis is transformed into estimated signals by a synthesis transform, namely, Inverse STFT (iSTFT) or Decoder Transform. This transform is formulated as

$$\hat{\mathbf{s}}_i = V(\mathbf{w} \odot \mathbf{m}_i), \tag{4}$$

where $V(\cdot)$ represents the synthesis transform.

The training data consists of pairs of sources and the mixture. When making the dataset, C dry source signals are taken from the database randomly as the source sound signals and summed into the mixture signal.

B. Universal Sound Separation

Universal sound separation is formulated in the same way as other approaches to monaural speech separation, although characteristics of the data are different. Figure 3 shows examples of the data represented as a log-power spectrogram. Aside from the speech, the examples are the same data used in the previous studies [1], [2]. Each sound has a very different time-frequency structure, for example, the spectrogram of the speech has a clear harmonic structure, although one of the creaking of doors is blurred in the frequency direction.

The data of sources used in previous studies have labels or the sound data are divided into some chunks by their characteristics in database. Even the labels are not given, the sound classification method, such as [20]–[22] gives them to each sources. However, previous studies used these labels only for excluding background sound. Namely, the sounds of barking of a dog and the sounds of printing papers belonged to the same class.

III. Multiple-Embedding Separation Networks

Our architecture has multiple feature extraction networks called Embedding Networks. Each network is trained together with the class label to capture the characteristics as class-specific embeddings. In this section, we introduce setting about the sound class labels, details of architecture, and an initial training.

A. Sound Class Label

When making the dataset, we assign the sound class labels to all sources. The dataset are divided into classes by these labels for the initial training. In addition to this, the labels are useful for equalizing the amounts of the training data of each class when making the dataset. If the amounts of the training data for each sound class is imbalanced, DNN is trained only for the classes with a large amount of data. Equalizing the amounts of the training data between classes leads to more training for classes with a small amount of data.

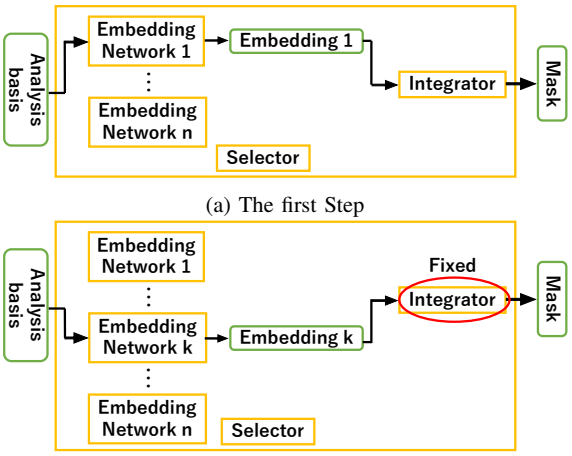


Fig. 5: Each step of the initial training. The first step is to do for Integrator. In this step, Embedding network is chosen randomly because the parameters of the trained is reset after the step. In the second step, each Embedding Network is trained. During the training, the parameters except Embedding Networks, such as Integrator are fixed.

B. Network Architecture

Our network is based on TDCN++ [1] but has a unique separation network. Figure 4 shows the structure of the separation network. This network is composed of three parts: Embedding Networks, Selector and Integrator.

First, Q Embedding Networks extract the sound feature by transforming the analysis basis into embeddings $\mathbf{e}_1, \mathbf{e}_2, ...\mathbf{e}_Q$, which are formulated as

$$\mathbf{e}_j = E_j(\mathbf{w}) \in \mathbb{R}^{N \times L}, \tag{5}$$

where $E_1(\cdot), E_2(\cdot), ...E_Q(\cdot)$ are the transforms of the corresponding Embedding Networks. Each embedding is class-specific. Second, Selector output an attention weight. This architecture is inspired by [23]. The attention weight $\mathbf{a}$ represents which embedding is important for the separation, which is formulated as

$$\mathbf{a} = S(\mathbf{w}) \in \{a | a \in \mathbb{R},\ 0 \leq a \leq 1\}^Q, \tag{6}$$

where $S(\cdot)$ represents the transform of Selector. a_j is the j-th element of $\mathbf{a}$. It satisfies $\sum_j a_j = 1$. Third, Integrator estimates the separation mask as the output of the separation network formulated as follows.

$$[\mathbf{m}_1, \mathbf{m}_2, ...\mathbf{m}_C] = I(\sum_j a_j \cdot \mathbf{e}_j), \tag{7}$$

TABLE I: The amounts of data of the dry source signal for each class.

	Speech	Dog	Bird	Door	Bell	Printer
hours	36.2	1.4	8.3	28.0	4.9	16.5

TABLE II: The separation performance for combinations of the same class in terms of SI-SDRi (dB). Average represents the average SI-SDRi for six classes. The bottom two were trained on the same datasets.

	Speech	Dog	Bird	Door	Bell	Printer	Average
baseline (all combinations)	3.62	8.48	5.20	10.30	6.84	2.66	6.18
baseline (combinations of the same class)	4.84	**8.60**	**5.69**	12.04	**7.98**	**3.69**	7.14
Our proposed	**7.12**	8.48	5.38	**12.64**	7.26	3.28	**7.36**

where $I(\cdot)$ represents the estimation transform by Integrator. The embeddings are summed up and further transformed into the mask.

Except to output of each part, we adopted TCN proposed as a replacement for RNNs in various sequence modeling tasks. TCN is also adopted by TDCN++, but we do not adopt a longer-range skip-residual connection. The output of Embedding Networks is raw output of TCN. The output dimension of Selector is Q. To match the dimension, 1-D Convolutional block make the output of TCN smaller into $Q \times L$. As a representative, the element of the end of the time bin is extracted, and Softmax function transform it into the attention weight. Integrator is the shallow version of the separation network of TDCN++

C. Initial Training

If we trained total networks without the class labels from beginning, there is a possibility that only one Embedding Network is trained and the other Embedding Networks are not. In this case, each Embedding Network is class-unspecific and cannot perform at full potential. To avoid this, the initial training using the class labels is needed.

The initial training consists of two steps. Figure 5 shows each step of the initial training. First, Integrator and non-separation network are trained using all datasets for a few epochs. The input analysis basis is received by only one Embedding Network. The mask estimation of Integrator is formulated as follows.

$$[\mathbf{m}_1, \mathbf{m}_2, ...\mathbf{m}_C] = I(\mathbf{e}_1). \tag{8}$$

After this step, the parameters of the trained Embedding Network are reset. Second, each Embedding Network is trained individually using the different training divided into classes for a few epochs. In this step, the parameters except Embedding Networks are fixed. The mask estimation of Integrator is formulated as follows.

$$[\mathbf{m}_1, \mathbf{m}_2, ...\mathbf{m}_C] = I(\mathbf{e}_j),\ j = 1, ...Q. \tag{9}$$

After this step, the total networks are trained using all data.

IV. Experiment

To evaluate proposed method, we compared the difference of the performance between our proposed method and TDCN++ [1] as a baseline method using a dataset including six sound classes.

A. Datasets

In order to simplify the initial training of the proposed method and the analysis of separation performance, we made a dataset of six sound classes and fixed the number of sources that could overlap as mixture to two. For the six classes, we chose the sound of speech, barking of a dog, humming of a bird, creaking of a door, ringing of bells, and operation of a printer. We called each class "Speech," "Dog," "Bird," "Door," "Bell," and "Printer," respectively. The sound data of speech are from the Japanese Newspaper Article Sentences Read Speech Corpus (JNAS) [1]. It includes 177 hours of people of various ages and genders reading a newspaper aloud. The data of the other classes are from the Pro Sound Effects Library database [2] used in the previous studies [1], [2]. These data were preprocessed, such as the use of a repetition and a clipping. The repetition was carried out in the same manner of the previous study [1]. Longer signals were clipped to 10 seconds. Table I shows the amounts of data of the dry source signal. In advance, we divided the data into training, validation, and test datasets at a ratio of 10:1:1. All data were down-sampled to 8000 Hz.

If the number of Embedding Networks matched that of all combinations of the sound classes, the number of parameters would be extremely large. For this reason, we made training and validation datasets consisting of combinations of only the same classes, for example, mixtures containing only Speech-Speech. The training datasets of the classes have mixtures ranging from 5.3 to 6.7 hours, and the validation datasets of the classes have mixtures of 0.5 to 0.6 hours. The amounts of the training data of each class are equalized, namely, the total amounts of the training and the validation datasets are 34.4 hours and 3.4 hours, respectively. On the other hand, as reference, we also made training and validation datasets consisting of all combinations of classes for the baseline. The training datasets of the classes have mixtures of 1.2 to 1.3 hours, while the validation datasets of the classes have mixtures of 0.1 to 0.2 hours. The amounts of the training and

[1] http://research.nii.ac.jp/src/JNAS.html
[2] https://www.prosoundeffects.com/

the validation datasets of each class are also equalized, namely the total amounts of the training and the validation datasets are 29.6 hours and hours, respectively. The test dataset was made of all combinations of the classes. The mixture amounts of the combinations ranged from 0.4 to 0.6 hours.

B. Training and Evaluation Configuration

The number of Embedding Networks was six to match the number of the classes. The initial training was done for five epochs through the first and the second steps. All models were implemented in Pytorch and trained using Adam [24] with a batch size of 1 on a triple NVIDIA GeForce RTX 2080 paralleled GPU. The learning ratio was 3.0×10^{-4}. When the validation loss did not decrease for ten epochs consecutively, training was finished. The loss function was the negative scale-invariant signal-to-distortion ratio (SI-SDR) [25]. SI-SDR is given by

$$\text{SI-SDR}(\mathbf{s}, \hat{\mathbf{s}}) = 10 \log_{10} \frac{||\alpha \mathbf{s}||^2}{||\alpha \mathbf{s} - \hat{\mathbf{s}}||^2}, \tag{10}$$

where $\alpha = \langle \mathbf{s}, \hat{\mathbf{s}} \rangle / ||\mathbf{s}||^2$, $\mathbf{s}$ is the source sound signal and $\hat{\mathbf{s}}$ is the estimated signal. In the training step, utterance-level permutation invariant training [26] was applied. The hyperparameter settings for the network refer to the best parameters of Conv-TasNet, except to Encoder-Decoder used as analysis-synthesis transform and the number of 1-D Conv repetitions of TCN. The parameters of the Encoder-Decoder are the same as those in a previous study [2]. The number of 1-D Conv repetitions for Embedding Networks, Selector, and Integrator was one, although for the baseline it was two due to the depth of our proposed method matching that of the baseline. In the evaluating the step, model performance was measured by SI-SDR improvement (SI-SDRi), which is the improvement in SI-SDR of the estimated signals from the raw mixture.

C. Results

Table II shows the performance of the same-class combinations. Our proposed method outperformed the baseline in average performance. The performance for Speech and Door improved by 2.28 dB and 0.60 dB, although the performance of the others degraded by up to 0.72 dB. Table III shows the performance of the combinations of different classes. The baseline trained with all combinations achieved the highest performance among all of the combinations. In comparing the proposed method and the baseline, the performance of the proposed method for non-speech combinations outperformed or underperformed the baseline by $+0.80$ dB to -0.97 dB, but all combinations of Speech-Other degraded.

Figure 6 shows the average value of the attention weight of Selector for the test dataset of the combinations of the same class. Embedding Networks 1 to 6 were trained for Speech, Dog, Bird, Door, Bell, and Printer in that order. The darker component of the six contributes more greatly to the separation.

TABLE III: The separation performance for combinations of the different classes in terms of SI-SDRi (dB). The component of the matrix represents the separation performance corresponding to each combination of the test data. The order of the performance is the same as in Table II

	Speech	Dog	Bird	Door	Bell
Dog	8.71 6.68 6.66				
Bird	12.53 8.79 7.06	12.74 10.44 10.85			
Door	14.42 9.22 8.00	15.14 12.23 12.65	9.06 8.47 8.95		
Bell	10.87 7.36 5.83	9.40 8.45 8.91	11.94 9.42 8.82	14.19 10.60 11.20	
Printer	8.92 4.84 4.79	7.73 6.01 6.10	7.95 6.09 5.21	11.25 7.87 7.61	6.52 6.46 5.49

D. Discussion

The average value of the attention weight of Speech was extremely biased, and the performance of the proposed method outperformed the baseline significantly. This shows that the proposed method is able to extract class-specific features correctly. Comparing Table I and Table II, the proposed method improved the performance for the class with a large amount of data of the dry source signal. This relationship indicates that the class-specific feature extraction is effective for the class with a large amount of data of the dry source signal.

The average value of the attention weight of Dog was most biased toward Embedding Networks 1 and 6, which were trained for Speech and Printer. In Table I, Speech and Printer each has a large amount of data of the dry source signal, but Dog does not. This indicates that the separation performance for the class with a small amount of data of the dry source signal is supported by the learning result of the class with a large amount of data. In the other words, the network learns the basic features that are common to the sound data by training a class with a large amount of data. This tendency of the learning result is explained in terms of the data augmentation techniques, which encourage capturing essential features by increasing the amount of data increasing data.

In Table III, the proposed method was not effective for the combinations not used in training. Consequently, we should train the proposed architecture with all combinations for the universal sound separation task. The number of model parameters of the proposed method is proportional to the number of Embedding Networks. If the number of Embedding Networks matched the number of all combinations, the model would need numerous parameters, thus requiring unrealistic calculation costs. To avoid this problem, merging classes with similar sound characteristics into a single class would be effective.

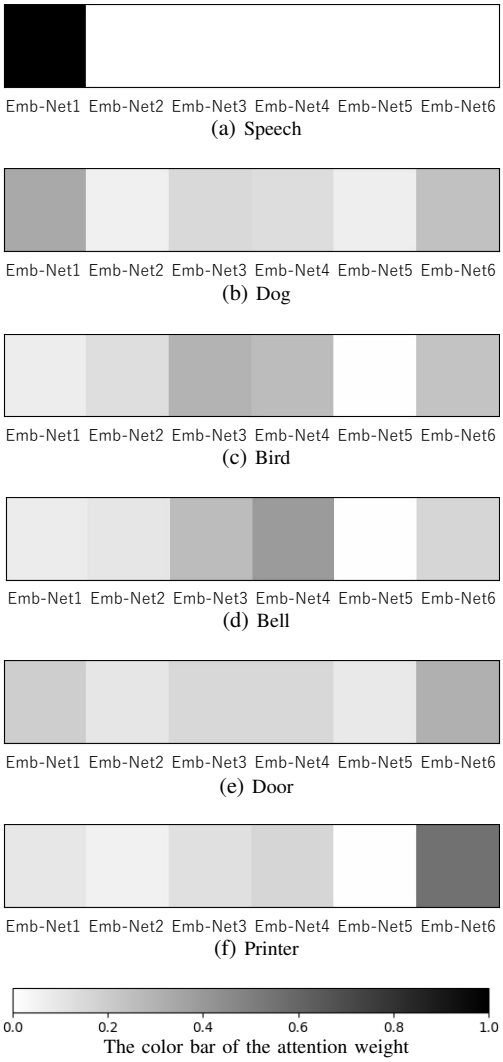


Fig. 6: The average value of the attention weight. (a) to (f) correspond to the test dataset of each class. Emb-Net 1 to 6 were trained for Speech, Dog, Bird, Door, Bell, and Printer in that order. The color bar shows the value of each attention weight.

V. Conclusions

In this paper, we introduced a novel DNN architecture, Multiple-Embedding Separation Networks. As a result of our experiment, we found that the proposed method improved the performance for a class with a large amount of data of the dry source signal. In future work, we will study more appropriate sound class labels. Our proposed method is not able to handle all class combinations as of this moment. If all combinations were labeled individually, a combinatorial explosion would occur. This not only increases the number of networks required, but also leads to the subdivision of classes which decreases the amount of data for each class. For this reason, we need the labels that can be assigned to all combinations and that are not too large. To solve this problem, we will adopt a sound classification approach to integrate classes or combinations of classes that have similar features. By using the output or the internal representation of these methods, we will get more appropriate sound class labels improving the separation performance.

References

[1] I. Kavalerov, S. Wisdom, H. Erdogan, B. Patton, K. Wilson, J. L. Roux, and J. R. Hershey, "Universal Sound Separation," in *Proc. of WASPAA*, Oct. 2019, pp. 175–179.

[2] E. Tzinis, S. Wisdom, J. R. Hershey, A. Jansen, and D. P. W. Ellis, "Improving Universal Sound Separation Using Sound Classification," in *Proc. of ICASSP*, May 2020, pp. 96–100.

[3] S. Wisdom, H. Erdogan, D. Ellis, R. Serizel, N. Turpault, E. Fonseca, J. Salamon, P. Seetharaman, and J. Hershey, "What's All the FUSS About Free Universal Sound Separation Data?" *arXiv preprint arXiv:2011.00803*, 2020.

[4] R. Jaiswal, D. FitzGerald, D. Barry, E. Coyle, and S. Rickard, "Clustering NMF basis functions using Shifted NMF for monaural sound source separation," in *Proc. of ICASSP*, May 2011, pp. 245–248.

[5] K. Yoshii, R. Tomioka, D. Mochihashi, and M. Goto, "Beyond NMF: Time-Domain Audio Source Separation without Phase Reconstruction," in *Proc. of ISMIR*, Nov. 2013, pp. 369–374.

[6] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal *et al.*, "Audio Set: An ontology and human-labeled dataset for audio events," in *Proc. of ICASSP*, May 2017, pp. 776–780.

[7] J. R. Hershey, Z. Chen, J. L. Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *Proc. of ICASSP*, May 2016, pp. 31–35.

[8] Y. Isik, J. L. Roux, Z. Chen, S. Watanabe, and J. R. Hersheye, "Single-Channel Multi-Speaker Separation using Deep Clustering," in *Proc. of INTERSPEECH*, Sep. 2016, pp. 545–549.

[9] Z. Chen, Y. Luo, and N. Mesgarani, "Deep attractor network for single-microphone speaker separation," in *Proc. of ICASSP*, Mar. 2017, pp. 246–250.

[10] Y. Luo and N. Mesgarani, "Conv-TasNet: Surpassing Ideal Time-Frequency Magnitude Masking for Speech Separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 8, pp. 1256–1266, Aug. 2019.

[11] L. Yi and N. Mesgarani, "TaSNet: Time-Domain Audio Separation Network for Real-Time, Single-Channel Speech Separation," in *Proc. of ICASSP*, Apr. 2018, pp. 696–700.

[12] Y. Luo, Z. Chen, and T. Yoshioka, "Dual-Path RNN: Efficient Long Sequence Modeling for Time-Domain Single-Channel Speech Separation," in *Proc. of ICASSP*, May 2020, pp. 46–50.

[13] C. Subakan, M. Ravanelli, S. Cornell, M. Bronzi, and J. Zhong, "Attention is All You Need in Speech Separation," in *Proc. of ICASSP*, June 2021, pp. 21–25.

[14] D. Wang and J. Chen, "Supervised Speech Separation Based on Deep Learning: An Overview," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 10, pp. 1702–1726, Oct. 2018.

[15] V. Narayanaswamy, J. J. Thiagarajan, R. Anirudh, and A. Spanias, "Unsupervised Audio Source Separation using Generative Priors," in *Proc. of INTERSPEECH*, Oct. 2020, pp. 2657–2661.

[16] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves *et al.*, "Wavenet: A generative model for raw audio," in *Proc. of 9th ISCA Speech Synthesis Workshop*, Sep. 2016, p. 125.

[17] G. Wichern, J. Antognini, M. Flynn, L. R. Zhu, E. McQuinn, D. Crow, E. Manilow, and J. L. Roux, "WHAM!: Extending Speech Separation to Noisy Environments," in *Proc. of INTERSPEECH*, Oct. 2019, pp. 1368–1372.

[18] D. Kitamura, N. Ono, H. Sawada, H. Kameoka, and H. Saruwatari, "Determined Blind Source Separation Unifying Independent Vector Analysis and Nonnegative Matrix Factorization," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 9, pp. 1626–1641, Sep. 2016.

[19] H. Sawada, H. Kameoka, S. Araki, and N. Ueda, "Multichannel Extensions of Non-Negative Matrix Factorization With Complex-Valued Data," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 5, pp. 971–982, May 2013.

[20] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "Panns: Large-scale pretrained audio neural networks for audio pattern recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2880–2894, Oct. 2020.

[21] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore *et al.*, "CNN Architectures for Large-scale Audio Classification," in *Proc. of ICASSP*, May 2017, pp. 131–135.

[22] A. Jansen, M. Plakal, R. Pandya, D. P. W. Ellis, S. Hershey, J. Liu *et al.*, "Unsupervised Learning of Semantic Audio Representations," in *Proc. of ICASSP*, Apr. 2018, pp. 126–130.

[23] R. Gu, L. Chen, S.-X. Zhang, J. Zheng, Y. Xu, M. Yu3 *et al.*, "Neural Spatial Filter: Target Speaker Speech Separation Assisted with Directional Information," in *Proc. of INTERSPEECH*, Sept. 2019, pp. 4290–4294.

[24] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[25] J. L. Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, "SDR - half-baked or well done?" in *Proc. of ICASSP*, May 2019, pp. 626—-630.

[26] M. Kolbæk, D. Yu, Z.-H. Tan, and J. Jensen, "Multitalker speech separation with utterance-level permutation invariant training of deep recurrent neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 10, pp. 1901–1913, Oct. 2017.