Knowledge Distillation in Acoustic Scene Classification

JEE-WEON JUNG, (Member, IEEE), HEE-SOO HEO, (Member, IEEE), HYE-JIN SHIM, (Member, IEEE), AND HA-JIN YU, (Member, IEEE)

1School of Computer Science, University of Seoul, Seoul 02504, South Korea
2Naver Corporation, Seongnam 13561, South Korea

Corresponding author: Ha-Jin Yu (hjyu@uos.ac.kr)

This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT and Future Planning under Grant 2020R1A2C1007081.

ABSTRACT Common acoustic properties that different classes share degrade the performance of acoustic scene classification systems. This results in a phenomenon where a few confusing pairs of acoustic scenes dominate a significant proportion of all misclassified audio segments. In this article, we propose adopting a knowledge distillation framework that trains deep neural networks using soft labels. Soft labels, extracted from another pre-trained deep neural network, are used to reflect the similarity between different classes that share similar acoustic properties. We also propose utilizing specialist models to provide additional soft labels. Each specialist model in this study refers to a deep neural network that concentrates on discriminating a single pair of acoustic scenes that are frequently misclassified. Self multi-head attention is explored for training specialist deep neural networks to further concentrate on target pairs of classes. The goal of this article is to train a single deep neural network that demonstrates performance equivalent to, or higher than, an ensemble of multiple models, by distilling the knowledge from several models. Diverse experiments conducted using the detection and classification of acoustic scenes and events 2019 task 1-a dataset demonstrate that the knowledge distillation framework is effective in acoustic scene classification. Specialist models successfully decrease the number of misclassified audio segments in the target classes. The final single model with the proposed method that is trained by the proposed knowledge distillation from several models, including specialists trained using an attention mechanism, shows a classification accuracy of 77.63%, higher than an ensemble of the baseline and multiple specialists.

INDEX TERMS Acoustic scene classification, common properties among different acoustic scenes, deep neural networks, frequently misclassified pair, knowledge distillation, specialist models.

I. INTRODUCTION

Acoustic scene classification (ASC) is a multi-class classification task that classifies an input audio segment into one of the pre-defined acoustic scenes. With the recent prosperity in deep neural networks (DNNs), research interest in the ASC with DNNs is increasing where recent state-of-the-art systems comprise DNNs [1]–[4]. For studies in ASC task, the detection and classification of acoustic scenes and events (DCASE) community is providing a common platform including datasets released annually for researches to study and report results [5]–[8]. Through previous challenges, a variety of studies have been made in feature exploration [9]–[12], DNN architectures [13]–[18], data augmentation [19]–[21], and ensembles [22], [23].

Common acoustic properties that reside among different acoustic scenes cause performance degradation of the ASC task [24]. It is hypothesized that the learning of a DNN can be disturbed if audio segments are labelled differently even though they share acoustic properties. For example, a babbling sound can exist in both ‘airport’ and ‘shopping_mall’ audio segments. However, audio segments that contain the same babbling sound are labelled as airport and shopping_mall depending on the location of the recording. One phenomenon that these common acoustic properties evoke is that a few pairs of acoustic scenes occupy the
The majority of misclassified audio segments. Fig. 1 demonstrates the impact of a few dominantly misclassified pairs of acoustic scenes by illustrating a cumulative bar graph. It shows that among 45 pairs, a combination of 10 acoustic scenes, the top five most confused pairs accounted for more than half of the total errors, and the top ten pairs accounted for more than 80% of the total misclassifications. The confusion matrix of the baseline DNN in Fig. 2 additionally confirms this phenomenon. It shows that pairs such as ‘airport-shopping_mall’, ‘public_square-street_pedestrian’, and ‘metro-metro_station’ which share similar acoustic configurations, dominates the misclassifications.

In this study, we adopt a knowledge distillation (KD) framework [25], [26] from the perspective of extracting soft labels that reflect the similarity between defined acoustic scenes, which was first introduced to the ASC task in authors’ previous work [24]. The KD framework sequentially trains two DNNs, where the first is trained using conventional ground truth labels (i.e. one-hot hard labels), combined with a categorical cross-entropy (CCE) loss function, and the second is trained with soft labels that are the output of the first DNN. Refer to Section III for further details regarding the KD framework. The experimental results show that the adoption of the KD framework successfully lowers the misclassification of pairs of acoustic scenes that are frequently misclassified.

However, these pairs are still occupy the majority of misclassified audio segments. To further mitigate this phenomenon, we take the approach of training several specialist DNNs where each specialist model concentrates on discriminating a specific target pair of acoustic scenes that are frequently misclassified [27]. We propose training with biased mini-batches to train the specialist DNNs. We also explore the attention mechanism to further concentrate on the target pairs of acoustic scenes. The purpose of this study using the specialist DNNs is to develop a single DNN that successfully distilled the knowledge of multiple models needed to identify pairs of acoustic scenes that are frequently confused. Specifically, we aim for a distilled single DNN to outperform an ensemble of several specialist models including the baseline DNN. Fig. 2 illustrates three confusion matrices from the baseline and two proposed systems: KD and specialist-based KD (SKD). It demonstrates that the proposed methods reduce the number of misclassified audio segments among frequently confusing pairs of acoustic scenes.

In this study, we present an overall adaptation of the KD framework for the ASC task with hypothesis and problematic phenomena in which our hypothesis evokes [24], [27]. The novelty of this article can be summarized as follows:

1) A modified three-phase training framework by combining previous studies.
2) Compare the effect of combinations of various loss functions, used in the proposed framework.
3) A criterion for model selection and application of multi-head attention mechanism for specialist DNNs.
4) A method to provide superiority to the teacher DNN’s input.
5) The effectiveness of using different number of specialist DNNs for the SKD.

The rest of this article is organized as follows. Section II describes the overall system used in this study. Section III describes the KD framework. The proposed specialist models are introduced in Section IV. Section V addresses specialist knowledge distillation. Experiments and results are in Section VI and the paper is concluded.

II. SYSTEM DESCRIPTION

In this section, we describe two components of the ASC system that we use throughout this article. The system comprises a front-end DNN for feature extraction and a support vector machine (SVM) for back-end classification. Table 1 shows the classification accuracy of our implementation of the baseline DNN in this study compared to a previous system [27], which demonstrated the second highest accuracy in the DCASE 2019 challenge among those using raw waveform as input.

| System       | Input Feature | # Param | Acc (%) |
|--------------|---------------|---------|---------|
| Jung et al. [27] | Raw waveform  | 636k    | 73.71   |
| Ours         | Raw waveform  | 636k    | 74.45   |

A. FRONT-END FEATURE EXTRACTION

Recent studies on the ASC task mainly utilize DNNs for feature extraction combined with other back-end classifiers, or use them in an end-to-end fashion. With the recent prosperity in deep learning, these systems succeeded in extracting fixed dimensional representation (also referred to as feature or code) with minimal pre-processing of input audio segments. In this study, we use the DNN architecture from [27],

J.-W. Jung et al.: Knowledge Distillation in ASC

VOLUME 8, 2020

166871
FIGURE 2. Illustration of confusion matrices of the baseline DNN and the two proposed systems, knowledge distillation (KD) and specialist-based knowledge distillation (SKD), evaluated using the DCASE 2019 task1-a test set. Demonstration of the importance of common acoustic properties among different classes can be observed through the confusion matrix of the baseline. The number of misclassified segments in the top three most confusing pairs of acoustic scenes shows that both proposed systems not only improve overall classification accuracy, but also reduce the number of misclassified segments among frequently misclassified pairs of acoustic scenes.

with stereo raw waveform inputs and representation vector outputs.

Table 2 describes the DNN architecture used in this article. The DNN is directly fed by raw waveforms and comprises convolutional residual blocks [29], [30] followed by a global max pooling and a global average pooling layer, a fully-connected layer, and an output layer. Convolutional blocks with residual connections extract a frame-level representation, which is then aggregated using a global max pooling and a global average pooling layer. Concatenation of the two global pooling layers is connected to the fully-connected layer whose output is used as the representation vector.

The system is trained using a categorical cross-entropy (CCE) with the ground truth (hard label) that can be described as:

$$L_{CCE} = -\frac{1}{N} \sum_{n=1}^{N} \sum_{c=1}^{C} y_{n,c} \cdot \log(P(o_c|x_n)),$$

where $N$ refers to the size of the mini-batch, $C$ is the number of classes, $y_{n,c}$ refers to a $C$ dimensional one-hot vector (if audio segment $x_n$ belongs to class $c$, only $y_{n,c}$ would be 1 and other values are all 0), and $P(o_c|x_n)$ is the output of the DNN output layer’s $c^{th}$ node with softmax activation applied when audio segment $x_N$ is input. Note that after training is complete, the output layer is removed. Other detailed explanations are dealt in Section VI.

B. BACK-END CLASSIFICATION

Conducting the ASC task in end-to-end fashion by combining front-end feature extraction and back-end classification with a single DNN is known to performs well. However, when conducting an ensemble of multiple DNNs, the softmax activated outputs of a DNN may not be appropriate to convey the probabilities or the concept of confidence, leading to poor calibration [31]. Therefore, we train a support vector machine (SVM) for back-end classification. Two kernels, Gaussian and sigmoid, are explored. Unless mentioned explicitly, all system performance in this article is reported using a SVM for back-end classification.
III. KNOWLEDGE DISTILLATION

A. OVERVIEW

Knowledge distillation was first devised in two studies conducted by Hinton et al. [25] and Li et al. [26] and was introduced for the ASC task in [24]. Although the two papers convey similar methodologies, Hinton et al. focused on the transfer of knowledge between DNNs, therefore, referred to as knowledge distillation. Li et al., on the other hand, focused on the fact that one DNN is used to train the other DNN and therefore referred to as teacher-student learning. Throughout this study, we use the terminology knowledge distillation (KD) framework and refer to the two DNNs as the teacher DNN and the student DNN.

Fig. 3 illustrates the process pipeline of the KD framework. In the KD framework, the teacher DNN (i.e. baseline) is trained in advance using the conventional scheme of hard label (one-hot vector) combined with the CCE loss function described in (1). After training the teacher DNN, the weight parameters of the teacher DNN are frozen. Then, the student DNN is trained using the Kullbeck-Leibler (KL) divergence loss function with the soft labels generated by the teacher DNN. The training of the student DNN can be described as:

\[
\mathcal{L}_{\text{KD}}(P_T, P_S) = \frac{1}{N} \sum_{n=1}^{N} \sum_{c=1}^{C} P_T(o_c|x_n) \cdot \log \left( \frac{P_T(o_c|x_n)}{P_S(o_c|x_n)} \right), \tag{2}
\]

where \(P_T(o_c|x_n)\) and \(P_S(o_c|x_n)\) refer to the teacher DNN and the student DNN output layer’s \(c^{th}\) node with softmax activation applied when audio segment \(x_n\) is input, and other notations follow those in (1).

B. FURTHER OBJECTIVE FUNCTIONS

In this sub-section, we further hypothesize two possible phenomena that can arise when applying the KD framework, and we propose two additional objective functions to supplement the KD framework. The first problematic scenario is that because the student DNN is only trained using the output of the teacher DNN, if the teacher DNN is overfitted and provides wrong soft labels, then the model may collapse. To find a solution for this, we conduct comparative experiments by adding the CCE loss in (1) using hard labels when applying the KD framework.

The second problem is that the distillation of knowledge is indirect because the representation of the last hidden layer is fed to the back-end SVM, instead of the output layer of the teacher. Therefore, we hypothesize that comparing the representation of the last hidden layer between the teacher DNN and the student DNN can be a more direct approach.

We use two distance metrics, as done in [32], mean squared error (MSE) and cosine distance described as:

\[
\mathcal{L}_{\text{MSE}} = \frac{1}{N} \sum_{n=1}^{N} \sum_{m=1}^{M} (P_T(e_m|x_n) - P_S(e_m|x_n))^2, \tag{3}\]

\[
\mathcal{L}_{\text{COS}} = -\frac{1}{N} \sum_{n=1}^{N} \sqrt{\frac{\sum_{m=1}^{M} P_T(e_m|x_n) \cdot P_S(e_m|x_n)}{\sum_{m=1}^{M} P_T(e_m|x_n)^2} \cdot \sum_{m=1}^{M} P_S(e_m|x_n)^2}, \tag{4}\]

where \(M\) refers to the dimensionality of the last hidden layer, and \(e_m\) is the \(m^{th}\) output of the last hidden layer. Note that in practice, we use \(-1 \times \text{cosine similarity}\) (also referred to as the cosine proximity loss) for cosine distance. Application of the two additional objective functions can be described as:

\[
\mathcal{L} = \mathcal{L}_{\text{KD}} + \mathcal{L}_{\text{CCE}} + \mathcal{L}_{\text{CODE}}, \tag{5}\]

where \(\mathcal{L}_{\text{CODE}}\) is either \(\mathcal{L}_{\text{MSE}}\) or \(\mathcal{L}_{\text{COS}}\) defined in (3) and (4), respectively. We investigate the effect of each component, expanding [33] which only considered \(\mathcal{L}_{\text{CODE}}\) and \(\mathcal{L}_{\text{KD}}\). The experimental results in Table 3 demonstrate that \(\mathcal{L}_{\text{KD}} + \mathcal{L}_{\text{MSE}}\) shows the best performance.

**TABLE 3.** Analysis of the effect of using various components to calculate the loss function. Weight parameters of all models except the baseline are initialized using those of the baseline in Table 1. (✓: included in loss function, ◐: excluded from loss function).

| Component                  | Hard label (one-hot) | Soft label (output layer) | Code (embedding) | Acc  |
|----------------------------|----------------------|---------------------------|------------------|------|
| Baseline                   | ✓                    | ◐                         | ◐                | 74.45|
| Original TS                | ◐                    | ✓                         | ◐                | 75.11|
| Code                       | -                    | -                         | Cosine           | 75.37|
|                            | -                    | -                         | MSE              | 75.26|
| Hard+Soft                  | ✓                    | ✓                         | -                | 75.60|
| Hard+Code                  | ✓                    | ✓                         | Cosine           | 75.20|
|                            | ✓                    | ✓                         | MSE              | 75.72|
| Soft+Code                  | ◐                    | ✓                         | Cosine           | 75.66|
|                            | ◐                    | ✓                         | MSE              | 75.89|
| All                        | ✓                    | ✓                         | Cosine           | 75.58|
|                            | ✓                    | ✓                         | MSE              | 75.07|
In this study, we propose to utilize multiple audio segments within an acoustic scene, recorded in different locations to extract soft labels. The ASC task defines rather abstract acoustic scenes such as a park or an airport, thus, the acoustic properties can widely vary depending on the recording locations. For example, a recording of a park where kids are playing tennis and where birds are chirping will contain different acoustic properties but both will be labelled park. Therefore, we hypothesize that using multiple audio segments recorded at different locations will enable the extraction of soft labels that are less dependent on recording location. Concretely, we utilize two approaches to set the superiority of the teacher DNN using multiple segments recorded at different locations. For example, a recording of a park where kids are playing tennis and where birds are chirping will contain different acoustic properties but both will be labelled ‘park’. Therefore, we hypothesize that using multiple audio segments recorded at different locations will enable the extraction of soft labels that are less dependent on recording location. Concretely, we utilize two approaches to set the superiority of the teacher DNN using multiple segments recorded at different locations. First, instead of making a specialist DNN to classify a specific category with fewer classes, because there were too many categories. For example, the general model classifies 15000 classes and a specialist was set to classify 150 classes among the 15000 classes. We adopted the concept of specialist models to the ASC task with two modifications [27].

In this research, we show that adapting the KD framework to the ASC task is effective not only by improving the overall classification accuracy, but also by lowering the number of misclassified audio segments in pairs of acoustic scenes that are frequently misclassified. The middle confusion matrix in Fig. 2 illustrates the confusion matrix of the model described in the previous section with KD training. It shows that the misclassification in majority of pairs, including top-3 most confusing pairs, decreased with few exceptions (e.g. ‘park’-‘public_square’). Despite the improvement, the confusion matrix still has the same problem of a few pairs frequently being misclassified. To further mitigate the confusion between the acoustic scenes we introduce specialist models for the confusing pairs [27].

In [25], a specialist DNN was proposed to classify more specific categories with fewer classes, because there were too many categories. For example, the general model classifies 15000 classes and a specialist was set to classify 150 classes among the 15000 classes. We adopted the concept of specialist models to the ASC task with two modifications [27]. First, instead of making a specialist DNN to classify a subset of the classes, we train specialist DNNs to classify entire classes. This approach enables both direct score-level ensemble and knowledge distillation from multiple specialist models to a single DNN, introduced in Section V. Second, we construct biased mini-batches focusing on a pair of confusing two classes. Specifically, we construct biased mini-batches by composing half of them with recordings from the two target confusing classes, and the other half with the rest of the classes. All the specialist DNNs in this article are initialized using the weight parameters of the baseline system.

We also propose a criterion for model selection for the specialist models. Various methods have been proposed for general model selection such as early stopping. Overall accuracy, which is widely used for model selection may not be an appropriate metric for the specialist models. Specialist models are not designed to classify all classes well; rather, they are trained to give additional discriminative power for the target pairs of acoustic scenes. To achieve such a goal, we propose to use the number of misclassified target pairs of two classes as a criterion. The number of misclassified segments by different specialist DNNs are described in Table 5, and performances based on using different numbers of specialist

### Table 4: Knowledge distillation from a teacher DNN to a student DNN with various temperature and teacher DNN input durations for extracting soft labels.

| Type of superiority | Duration of teacher DNN input | teacher DNN T | student DNN T | student DNN T | student DNN T |
|---------------------|-------------------------------|---------------|---------------|---------------|---------------|
| -                   | 10 seconds                    | 72.46         | 71.94         | 72.59         | 72.27         | 72.52         | 72.13         | 72.75         | 72.96         | 73.49         |
| Concat              | 20 seconds                    | 75.89         | 75.51         | 75.34         | 75.22         | 75.49         | 75.26         | 75.51         | 74.99         | 75.55         |
|                     | 30 seconds                    | 75.72         | 75.53         | 75.45         | 75.05         | 75.49         | 75.53         | 74.78         | 75.14         | 75.20         |
| Each                | 20 seconds                    | 75.32         | 75.66         | 75.97         | 75.11         | 75.57         | 75.51         | 75.26         | 75.28         | 75.57         |
|                     | 30 seconds                    | 76.27         | 75.62         | 75.51         | 75.68         | 75.68         | 75.30         | 75.20         | 74.97         | 75.26         |

Temperature [25] is the parameter that decides the extent of smoothness of the output after applying the softmax function. It can be interpreted as a sort of label smoothing, and is widely used in tasks of other domains that utilize the KD framework. The DNN’s softmax output with temperature applied can be described as:

\[
o_i = \frac{\exp(z_i / T)}{\sum_c \exp(z_c / T)},
\]

where \(o_i\) refers to the output node for class \(i\) with softmax applied, \(z_i\) refers to the logit node for class \(i\) before softmax is applied, and \(T\) is the temperature. Table 4 shows the results of comparative experiments with varying temperatures.

### IV. SPECIALIST MODELS

#### A. OVERVIEW

In this research, we show that adapting the KD framework to the ASC task is effective not only by improving the overall classification accuracy, but also by lowering the number of misclassified audio segments in pairs of acoustic scenes that are frequently misclassified. The middle confusion matrix in Fig. 2 illustrates the confusion matrix of the model described in the previous section with KD training. It shows that the misclassification in majority of pairs, including top-3 most confusing pairs, decreased with few exceptions (e.g. ‘park’-‘public_square’). Despite the improvement, the confusion matrix still has the same problem of a few pairs frequently being misclassified. To further mitigate the confusion between the acoustic scenes we introduce specialist models for the confusing pairs [27].

In [25], a specialist DNN was proposed to classify more specific categories with fewer classes, because there were too many categories. For example, the general model classifies 15000 classes and a specialist was set to classify 150 classes among the 15000 classes. We adopted the concept of specialist models to the ASC task with two modifications [27]. First, instead of making a specialist DNN to classify a subset of the classes, we train specialist DNNs to classify entire classes. This approach enables both direct score-level ensemble and knowledge distillation from multiple specialist models to a single DNN, introduced in Section V. Second, we construct biased mini-batches focusing on a pair of confusing two classes. Specifically, we construct biased mini-batches by composing half of them with recordings from the two target confusing classes, and the other half with the rest of the classes. All the specialist DNNs in this article are initialized using the weight parameters of the baseline system.
be used to calculate attention weights, e.g. phonetic information, can be used to derive attention weights for speaker recognition. Weights for attention mechanism can also be derived using the given representation itself, referred to as a self-attention mechanism.

Multi-head attention separates a representation space into multiple subspaces and concatenates them after applying softmax activation to each subspace. Each subspace is referred to as a head. Multi-head attention assigns an attention weight for each head; therefore, it is expected that each subspace can be adequately emphasized. In this study, we propose adopting self multi-head attention [36] to further let a specialist DNN focus on discriminating a target pair of confusing acoustic scenes. Biased mini-batches make each specialist DNN focus on a target pair of acoustic scenes, however, the DNN architecture remains the same. We assume that adding a self multi-head attention to the DNN would facilitate specialist DNN to focus on a target pair of scenes with a few additional parameters by emphasizing relevant information and de-emphasizing others.

Specifically, we apply this mechanism to the feature map of the last convolutional layer (‘Res7’ in Table 2), before global pooling, similar to the method proposed in [38]. Given a feature map of 1-dimensional CNN, $h = [h_1, h_2, \ldots, h_T]$, $h_t \in \mathbb{R}^F$, the representation space is first separated into multiple subspaces according to the number of heads:

$$h = \begin{bmatrix} h_{11} & h_{12} & \cdots \\ \vdots & \vdots & \cdots \\ h_{K1} & \cdots & h_{KT} \end{bmatrix}, \quad h_{lt} \in \mathbb{R}^{F/K}$$

(7)

where $F$ is the number of filters, $t$ is the time index, and $k$ is the head index, and $K$ is the number of heads. We derive attention weights for each subspace by first averaging a given representation by the time axis ($h_{lt}'$), conducting dot product with weight $u_l$, and then applying the softmax activation:

$$h_{lt}' = \frac{1}{T} \sum_t h_{lt}.$$  

(8)

$$w_{kl} = \frac{\exp(h_{lt}' \cdot u_l)}{\sum_{k'}^{F/K} \exp(h_{lt}' \cdot u_{l_{k'}})},$$

(9)

where $w_{kl}$ refers to the attention weight for $l^{th}$ filter in head $k$, $u_l$ is a trainable vector, $u_l \in \mathbb{R}^{F/K}$, and $w_{k} = [w_{k1}, w_{k2}, \ldots, w_{k,F/K}]$. Finally, we derive the multi-head attention applied feature map $b$:

$$b = \begin{bmatrix} b_{11} & b_{12} & \cdots \\ \vdots & \vdots & \cdots \\ b_{K1} & \cdots & b_{KT} \end{bmatrix}, \quad b_{kl} = h_{lt}' w_{kl}.$$  

(10)

### B. MULTI-HEAD ATTENTION-BASED SPECIALIST MODELS

Attention mechanisms demonstrate successful performance across a number of tasks [15], [35]–[38]. It emphasizes and de-emphasizes parts of a given representation by re-scaling hidden representations with attention weights. Attention weights has a total value of 1 through a softmax activation function, which ensures that some parts are emphasized and other parts are de-emphasized. Additional information can be used to calculate attention weights, e.g. phonetic information, can be used to derive attention weights for speaker recognition. Weights for attention mechanism can also be derived using the given representation itself, referred to as a self-attention mechanism.

Multi-head attention separates a representation space into multiple subspaces and concatenates them after applying softmax activation to each subspace. Each subspace is referred to as a head. Multi-head attention assigns an attention weight for each head; therefore, it is expected that each subspace can be adequately emphasized. In this study, we propose adopting self multi-head attention [36] to further let a specialist DNN focus on discriminating a target pair of confusing acoustic scenes. Biased mini-batches make each specialist DNN focus on a target pair of acoustic scenes, however, the DNN architecture remains the same. We assume that adding a self multi-head attention to the DNN would facilitate specialist DNN to focus on a target pair of scenes with a few additional parameters by emphasizing relevant information and de-emphasizing others.

Specifically, we apply this mechanism to the feature map of the last convolutional layer (‘Res7’ in Table 2), before global pooling, similar to the method proposed in [38]. Given a feature map of 1-dimensional CNN, $h = [h_1, h_2, \ldots, h_T]$, $h_t \in \mathbb{R}^F$, the representation space is first separated into multiple subspaces according to the number of heads:

$$h = \begin{bmatrix} h_{11} & h_{12} & \cdots \\ \vdots & \vdots & \cdots \\ h_{K1} & \cdots & h_{KT} \end{bmatrix}, \quad h_{lt} \in \mathbb{R}^{F/K}$$

(7)

where $F$ is the number of filters, $t$ is the time index, and $k$ is the head index, and $K$ is the number of heads. We derive attention weights for each subspace by first averaging a given representation by the time axis ($h_{lt}'$), conducting dot product with weight $u_l$, and then applying the softmax activation:

$$h_{lt}' = \frac{1}{T} \sum_t h_{lt}.$$  

(8)

$$w_{kl} = \frac{\exp(h_{lt}' \cdot u_l)}{\sum_{k'}^{F/K} \exp(h_{lt}' \cdot u_{l_{k'}})},$$

(9)

where $w_{kl}$ refers to the attention weight for $l^{th}$ filter in head $k$, $u_l$ is a trainable vector, $u_l \in \mathbb{R}^{F/K}$, and $w_{k} = [w_{k1}, w_{k2}, \ldots, w_{k,F/K}]$. Finally, we derive the multi-head attention applied feature map $b$:

$$b = \begin{bmatrix} b_{11} & b_{12} & \cdots \\ \vdots & \vdots & \cdots \\ b_{K1} & \cdots & b_{KT} \end{bmatrix}, \quad b_{kl} = h_{lt}' w_{kl}.$$  

(10)

### V. SPECIALIST KNOWLEDGE DISTILLATION

The knowledge distillation framework enables training a student DNN with the soft labels extracted from a teacher DNN. With specialist models, we can make an ensemble of multiple specialist models and a baseline model. Although an ensemble brings additional performance improvement, it requires
multiplicative computational costs and thus impractical for real-world scenarios. In this study, we propose to combine two frameworks and distill the knowledge from multiple models, specialist models and a baseline model, into a single DNN, which we refer to as the specialist knowledge distillation (SKD) framework. Our objective is to develop a single DNN that show equivalent to, or better performance than an ensemble of multiple specialist DNNs and a baseline. Fig. 4 illustrates the overall process pipeline of the proposed SKD framework for the ASC task.

Distillation of the proposed framework (phase 3 in Fig. 4) is addressed as a summation of knowledge distillation in (2) with multiple soft labels:

$$\mathcal{L}_{SKD} = \sum_{s \in Sp} \mathcal{L}_{KD}(s, P_s),$$

(11)

where $Sp$ is the group of DNNs used to extract soft labels, $P_T$ is the teacher DNN (baseline), $P_{sp}$ is the specialist DNN, and $k$ refers to concentrating on the $k^{th}$ most confused pair of acoustic scenes. Experimental results in Table 7 show that the SKD framework successfully incorporates the teacher DNN and multiple specialist DNNs into a single DNN.

**TABLE 7.** Comparison of accuracy when conducting a score-sum ensemble and the proposed specialist knowledge distillation using the baseline and multiple specialist DNNs. Best performing configurations from Table 6 are reported.

|                | Ensemble | SKD |
|----------------|----------|-----|
| Biased mini-batch | 76.58    | 76.37 |
| Multi-head attention | 75.93    | 77.63 |

**VI. EXPERIMENTS AND RESULTS**

All experiments in this article were conducted using PyTorch [39], a deep learning library for Python. Codes for reproduction is available.²

²https://github.com/Jungjee/dcass2019specialistkd

**A. DATASET**

The DCASE 2019 task 1-a dataset was used for all experiments. It comprises 14400 segments totaling about 40 hours from one of the 10 defined acoustic scenes recorded in 12 European cities. All segments are recorded at a sampling rate of 48 kHz, 24-bit resolution, in stereo. Each segment has a duration of 10 seconds where the original recording of five to six minutes was split by organizers. We use the train set in the fold-1 configuration for model training and test set in fold-1 configuration for model evaluation following the literature.³

**B. EXPERIMENTAL CONFIGURATIONS**

We directly input raw waveforms with pre-emphasis applied to both channels. All 479999 samples consisting of 10 seconds at a 48kHz sampling rate are input to the DNN. Residual blocks comprise a 1d convolution layer with a kernel length of 3, batch normalization (BN) [40], leaky rectified unit (ReLU) activation [41], and a max pooling of size 3 following the work found in [27]. We conduct both global max pooling and global average pooling to the output of the last residual block, and then concatenate the two outputs. Code representation has a dimensionality of 64. The rest of the configuration follow the description in Table 2. In our research, all DNNs, the baseline, specialist DNNs, and the final single model have identical architecture, resulting in the same number of weight parameters, with an exception in multi-head attention specialist DNNs.

We use the AMSGrad optimizer [42], [43] with a learning rate of 0.001. We adopt mix-up [44] with $\alpha = 0.1$ after first five epochs. Weight decay with $\lambda = 0.001$ is applied to all DNNs. We use a mini-batch size of 24 for all experiments. Center loss is additionally applied for training all DNNs [45]. Note that we fix all DNNs throughout this article to use raw waveform as input to measure the sole effect of the proposed methods. The KD framework itself has been validated to be effective for different input features (e.g. Mel-spectrogram) in previous works [24], [27].

**C. RESULTS**

Table 3 describes comparison results with different composition of loss functions for knowledge distillation. Here, columns depict whether a specific component is included in the loss function and rows depict the performance for each combination. Soft label refers to the original knowledge distillation loss function, which uses the KL-divergence between the student DNN’s output and the teacher DNN’s output (i.e. $\mathcal{L}_{KD}$). Hard label refers to a conventional CCE loss function where the purpose is to complement in case the soft label is wrong (i.e. $\mathcal{L}_{CCE}$). Code refers to comparing the teacher DNN and the student DNN’s code representation with a defined loss function (i.e. $\mathcal{L}_{MSE}$ or $\mathcal{L}_{COS}$). In this study,

³Performance on the competition evaluation set cannot be reported because labels are not publicly available
we conduct experiments using cosine similarity and MSE following the work in [32]. Results show that in all cases, knowledge distillation improves performance regardless of distillation points and the kind of objective functions used. Usage of hard labels is not necessary for successful distillation. For the objective function used to compare the code representation of the teacher and the student DNN, MSE demonstrates higher accuracy than cosine similarity. The best result is achieved by using soft label and code together, without hard labels, demonstrating an accuracy of 75.89%. Hereafter, we conduct all knowledge distillation, including specialist knowledge distillation, using both soft label and MSE for code distillation.

Table 4 shows the results of knowledge distillation under different durations of teacher DNN input to extract the soft label, the types of methods to extract soft labels from longer durations, and various temperatures for the teacher DNN and the student DNN. The first row shows the result of the conventional knowledge distillation framework, and the second to fifth rows show the results of providing additional superiority to the soft labels extracted from the teacher DNN by using multiple segments from the same class. Under ‘Type of superiority’, ‘Concat’ refers to inputting concatenated segments to the teacher DNN, and ‘Each’ refers to inputting each segment to the teacher DNN and averaging soft labels from different segments to derive the final soft label. Different columns show the effect of using temperature for knowledge distillation where a higher temperature leads to a smoother soft label.

Results show that regardless of different temperatures and types of methods of superiority, the proposed scheme of using multiple segments for extracting soft labels improves performance. Using three segments showed a slightly further improvement compared to using two segments in most cases. When using multiple segments to extract a soft label, averaging the soft labels showed higher performance than inputting concatenated segments to the teacher DNN in most cases. Adopting the concept of temperature was effective with single segment-based soft labels. However, when using soft labels extracted from multiple segments, setting both the teacher DNN and the student DNN’s temperature to 1 showed the best performance. The best accuracy, 76.27% is achieved by using three segments to extract a soft label, and averaging the soft labels, without giving explicit temperatures for knowledge distillation. Hereafter, we use this configuration for all knowledge distillation from multiple teacher DNNs including specialist DNNs to a single student DNN.

Table 5 describes the number of misclassified segments by the baseline, the specialist DNNs with biased mini-batch construction, and the specialist DNNs that additionally used multi-head attention. We use both biased mini-batch and multi-head attention schemes for ‘Attention’ because biased mini-batch construction is needed to make a specialist DNNs focus on a specific pair of classes. Results show that six of the 10 multi-head attention-based specialist DNNs outperform the baseline and biased mini-batch based specialist DNNs. We conclude that specialist DNNs can effectively reduce the number of misclassifications of the target confusing pairs and that using attention mechanism for specialist DNNs show further improvement for the most confusing pairs.

Table 6 shows the results from conducting specialist knowledge distillation using two specialist frameworks with differing numbers of specialist DNNs. Specialist DNNs trained with biased mini-batch composition and attention show different tendencies overall. When using biased mini-batch composition, using more specialist DNNs demonstrates the best performance where an accuracy of 76.37% is achieved with seven specialist DNNs. On the other hand, using only a few specialist DNNs demonstrated the best performance when both biased mini-batch and multi-head attention is used to train the specialist DNNs. Accuracy of 77.63% is achieved with only one specialist DNN.

Additionally, the results in Table 6 shows a tendency consistent with the results in Table 5. Specialists trained with the multi-head attention framework show relatively more effectiveness for the top confusing pairs. Accordingly, specialist knowledge distillation using attention-based specialist DNNs require only one, two, or three specialist DNNs for the best performance. From these results, we conclude that using the proposed multi-head attention framework for training specialist DNNs in addition to biased mini-batch construction is effective. It not only shows better performance, but also requires fewer specialist DNNs, minimizing the overhead in the overall process pipeline.

Table 7 describes comparison results between a score-sum ensemble and the proposed SKD. For the ensemble, we report the best performance among using different numbers of specialists empirically. Results show that the combination of training specialists with multi-head attention and conducting SKD outperforms all other configurations. Based on this result, we conclude that the SKD successfully distills the knowledge of each specialist DNNs into a single DNN.

Lastly, Table 8 addresses the comparison of state-of-the-art systems in the ASC task. Three systems from the DCASE 2019 task 1-a are used for comparison which ranked the 1st, 3rd, and 5th, respectively where we report best single system performance that use Mel-spectrograms extracted from both channels as input [2]–[4]. Note that we could not find single system performance of the 2nd and 4th ranking systems’ performance on the task 1-a fold 1 test set. Result demonstrates

**TABLE 8.** Comparison with state-of-the-art systems on the DCASE 2019 task1-a challenge. Three systems selected for comparison adopt two-channel Mel-spectrogram as input feature, without any ensemble technique (= Not reported).

| System      | # Param | Acc  |
|-------------|---------|------|
| Chen et al. [2] | -  | 77.56 |
| Seo et al. [3] | -  | 76.00 |
| Huang et al. [4] | 25088k | 77.87 |
| Ours       | 636k   | 77.63 |
the effectiveness of the proposed approach. The proposed system using SKD shows the second best performance with 0.24% lower than that of Huang et al.’s work, but requires almost forty times less number of parameters.

VII. CONCLUSION

In this research, we adapted a knowledge distillation framework for the ASC task with two purposes. One was to mitigate the misclassification of a few pairs of confusing acoustic scenes using soft labels that reflect the similarity of acoustic properties between different classes. The other purpose was to develop a single DNN that demonstrates performance comparable to a score-level ensemble of multiple DNNs by conducting distillation from multiple DNNs into a single DNN. We proposed various techniques, configuring superiority for soft labels with multiple segments and temperatures to make knowledge distillation successful in the ASC task. We also adapted the scheme of applying specialist DNNs to the ASC task by configuring each specialist DNN to concentrate on discriminating a target pair of acoustic scenes. We also proposed the techniques using biased mini-batch construction and the multi-head attention specialist. Utilizing various proposed methods, a single DNN that is fed by raw waveforms demonstrated an accuracy of 77.63% on the test set of fold-1 configuration from the DCASE task 1-a dataset.

REFERENCES

[1] K. Koutini, H. Eghbal-Zadeh, and G. Widmer, “Receptive-field-regularized CNN variants for acoustic scene classification,” in Proc. Detection Classification Acoustic Scenes Events Workshop (DCASE), New York, NY, USA, 2019, pp. 124–128.

[2] H. Chen, Z. Liu, Z. Liu, P. Zhang, and Y. Yan, “Integrating the data augmentation scheme with various classifiers for acoustic scene modeling,” DCASE Challenge, NYU Tandon School Eng., New York, NY, USA, Tech. Rep., Jun. 2019. [Online]. Available: http://dcase.community/challenge2019/task-asc-scene-classification-results-a

[3] S. Hyeri and P. Jihwan, “Acoustic scene classification using various pre-processed features and convolutional neural networks,” DCASE Challenge, Tech. Rep., Jun. 2019.

[4] J. Huang, P. L. Meyer, H. Lu, H. C. Marrari, and J. D. Hoy, “Acoustic scene classification using deep learning-based ensemble averaging,” DCASE Challenge, Tech. Rep., Jun. 2019.

[5] T. Virtanen, A. Mesarios, T. Heittola, A. Diment, E. Vincent, B. Benetos, and B. M. Elizalde, in Proc. Detection Classification Acoustic Scenes Events Workshop (DCASE), 2017. [Online]. Available: http://dcasecommunity/workshop2017/proceedings

[6] A. Mesaros, T. Heittola, and T. Virtanen, “A multi-device dataset for urban acoustic scene classification,” in Proc. Detection Classification Acoustic Scenes Events Workshop (DCASE), Nov. 2018, pp. 9–13.

[7] D. M. Plumbley, C. Kroos, P. J. Bello, G. Richard, P. W. D. Ellis, and A. Mesaros, in Proc. Detection Classification Acoustic Scenes Events Workshop (DCASE), 2018. [Online]. Available: http://dcasecommunity/workshop2018/proceedings

[8] M. Mandel, J. Salamon, and P. W. D. Ellis, in Proc. Detection Classification Acoustic Scenes Events Workshop (DCASE), New York, NY, USA, Oct. 2019. [Online]. Available: http://dcasecommunity/ workshop2019/proceedings

[9] M. C. Green and D. Murphy, “Acoustic scene classification using spatial features,” in Proc. Detection Classification Acoustic Scenes Events Workshop (DCASE), Nov. 2017, pp. 42–45.

[10] J.-W. Jung, H.-S. Heo, I.-H. Yang, S.-H. Yoon, H.-J. Shim, and H.-J. Yu, “DNN-based audio scene classification for DCASE2017: Dual input features, balancing cost, and stochastic data duplication,” in Proc. Detection Classification Acoustic Scenes Events Workshop (DCASE), Nov. 2017, pp. 59–63.

[11] J. Lee, J. Park, S. Kum, Y. Jeong, and J. Nam, “Combining multi-scale features using sample-level deep convolutional neural networks for weakly supervised sound event detection,” in Proc. Detection Classification Acoustic Scenes Events Workshop (DCASE), Nov. 2017, pp. 69–73.

[12] T. Zhang, J. Liang, and B. Ding, “Acoustic scene classification using deep CNN with fine-resolution feature,” Expert Syst. Appl., vol. 143, Apr. 2020, Art. no. 113067.

[13] H. Zeinali, L. Burget, and J. H. Cernocky, “Convolutional neural networks and x-vector embedding for DCASE2018 acoustic scene classification challenge,” in Proc. Detection Classification Acoustic Scenes Events 2018 Workshop (DCASE), Nov. 2018, pp. 202–206.

[14] J. Cho, S. Yun, H. Park, J. Eum, and K. Hwang, “Acoustic scene classification based on a large-margin factorized CNN,” in Proc. Detection Classification Acoustic Scenes Events Workshop (DCASE), New York, NY, USA, 2019, pp. 45–49.

[15] Z. Ren, Q. Kong, K. Qian, M. Plumbley, and B. Schuller, “Attention-based convolutional neural networks for acoustic scene classification,” in Proc. Detection Classification Acoustic Scenes Events Workshop (DCASE), Nov. 2018, pp. 39–43.

[16] J. Naranjo-Alcacar, S. Perez-Castanos, P. Zuccarello, and M. Cobos, “Acoustic scene classification with squeeze-excitation residual networks,” IEEE Access, vol. 8, pp. 112287–112296, 2020.

[17] H.-J. Shim, J.-W. Jung, J.-H. Kim, and H.-J. Yu, “Capturing scattered discriminative information using a deep architecture in acoustic scene classification,” 2020, arXiv:2007.04631. [Online]. Available: http://arxiv.org/abs/2007.04631

[18] M. D. McDonnell and W. Gao, “Acoustic scene classification using deep residual networks with large hidden units,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), May 2020, pp. 141–145.

[19] S. Mun, S. Park, D. K. Han, and H. Ko, “Generative adversarial network based acoustic scene training set augmentation and selection using SVM hyper-plane,” in Proc. Detection Classification Acoustic Scenes Events Workshop (DCASE), Nov. 2017, pp. 93–102.

[20] T. Inoue, P. Vinayakshin, S. Wang, D. Wood, A. Munawar, B. J. Ko, N. Greco, and R. Tachibana, “Shuffling and mixing data augmentation for environmental sound classification,” in Proc. Detection Classification Acoustic Scenes Events Workshop (DCASE), New York, NY, USA, 2019, pp. 109–113.

[21] H. Chen, Z. Liu, Z. Liu, and P. Zhang, “ACGAN-based data augmentation integrated with long-term scalogram for acoustic scene classification,” 2020, arXiv:2005.13146. [Online]. Available: http://arxiv.org/abs/2005.13146

[22] T. Nguyen and F. Pernkopf, “Acoustic scene classification using a convolutional neural network ensemble and nearest neighbor filters,” in Proc. Detection Classification Acoustic Scenes Events Workshop (DCASE), Nov. 2018, pp. 34–38.

[23] J.-W. Jung, H.-S. Heo, H.-J. Shim, and H.-J. Yu, “DNN based multi-level feature ensemble for acoustic scene classification,” in Proc. Detection Classification Acoustic Scenes Events Workshop (DCASE), Nov. 2018, pp. 113–117.

[24] H.-S. Heo, J.-W. Jung, H.-J. Shim, and H.-J. Yu, “Acoustic scene classification using teacher-student learning with soft-labels,” in Proc. Interspeech, 2019, pp. 614–618.

[25] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” Mar. 2015, arXiv:1503.02531.

[26] B. Gitlow, “Fusion of audio streams for acoustic scene classification,” in Proc. Detection Classification Acoustic Scenes Events Workshop (DCASE), 2018, pp. 1910–1914.

[27] H. Zeinali, L. Burget, and J. H. Cernocky, “Convolutional neural networks for acoustic scene classification,” 2020, arXiv:2005.13146. [Online]. Available: http://arxiv.org/abs/2005.13146

[28] T. Nguyen and F. Pernkopf, “Acoustic scene classification using a convolutional neural network ensemble and nearest neighbor filters,” in Proc. Detection Classification Acoustic Scenes Events Workshop (DCASE), New York, NY, USA, 2019, pp. 109–113.

[29] J. Naranjo-Alcacar, S. Perez-Castanos, P. Zuccarello, and M. Cobos, “Acoustic scene classification with squeeze-excitation residual networks,” IEEE Access, vol. 8, pp. 112287–112296, 2020.

[30] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. Interspeech, 2014, pp. 1910–1914.

[31] J.-W. Jung, H. Heo, H.-J. Shim, and H.-J. Yu, “Distilling the knowledge of specialist deep neural networks in acoustic scene classification,” in Proc. Detection Classification Acoustic Scenes Events Workshop (DCASE), 2019, pp. 114–118.

[32] J.-W. Jung, H.-S. Heo, I.-H. Yang, H.-J. Shim, and H.-J. Yu, “Avoiding speaker overfitting in end-to-end DNNs using raw waveform for text-independent speaker verification,” in Proc. Interspeech, Sep. 2018, pp. 23–24.

[33] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2016, pp. 770–778.

[34] K. He, X. Zhang, S. Ren, and J. Sun, “Identity mappings in deep residual networks,” in Proc. Eur. Conf. Comput. Vis. Springer, 2016, pp. 630–645.
[31] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, “On calibration of modern neural networks,” in Proc. Int. Conf. Mach. Learn., 2017, pp. 1321–1330.

[32] J.-W. Jung, H.-S. Heo, H.-J. Shim, and H.-J. Yu, “Short utterance compensation in speaker verification via cosine-based teacher-student learning of speaker embeddings,” in Proc. IEEE Autom. Speech Recognit. Understand. Workshop (ASRU), Dec. 2019, pp. 335–341.

[33] H.-S. Heo, J.-W. Jung, I.-H. Yang, S.-H. Yoon, H.-J. Shim, and H.-J. Yu, “End-to-End losses based on speaker basis vectors and all-speaker hard negative mining for speaker verification,” in Proc. Interspeech, Sep. 2019, pp. 4035–4039.

[34] J. Kim, M. El-Khamy, and J. Lee, “Bridgenets: Student-teacher transfer learning based on recursive neural networks and its application to distant speech recognition,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), Apr. 2018, pp. 5719–5723.

[35] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” 2015, arXiv:1409.0473. [Online]. Available: https://arxiv.org/abs/1409.0473

[36] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in Proc. Adv. Neural Inf. Process. Syst., 2017, pp. 5998–6008.

[37] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, “Listen, attend and spell: A neural network for large vocabulary conversational speech recognition,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), Mar. 2016, pp. 4960–4964.

[38] M. India, P. Safari, and I. Hernando, “Self multi-head attention for speaker recognition,” in Proc. Interspeech, Sep. 2019, pp. 4305–4309.

[39] A. Paszke et al., “Pytorch: An imperative style, high-performance deep learning library,” in Proc. Adv. Neural Inf. Process. Syst., 2019, pp. 8024–8035.

[40] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in Proc. Int. Conf. Mach. Learn., 2015, pp. 448–456.

[41] A. L. Maas, A. Y. Hannun, and A. Y. Ng, “Rectifier nonlinearities improve neural network acoustic models,” in Proc. ICML Workshop Deep Learn. Audio, Speech Lang. Process., 2013, pp. 3–8.

[42] S. J. Reddi, S. Kale, and S. Kumar, “On the convergence of Adam and beyond,” 2018, arXiv:1904.09237. [Online]. Available: https://arxiv.org/abs/1904.09237

[43] P. T. Tran and L. T. Phong, “On the convergence proof of AMSGrad and a new version,” IEEE Access, vol. 7, pp. 61706–61716, 2019.

[44] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, “mixup: Beyond empirical risk minimization,” 2018, arXiv:1710.09412. [Online]. Available: https://arxiv.org/abs/1710.09412

[45] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, “A discriminative feature learning approach for deep face recognition,” in Proc. Eur. Conf. Comput. Vis. Springer, 2016, pp. 499–515.

JEE-WEON JUNG (Member, IEEE) received the B.S. degree in computer science and the B.A. degree in business administration from the University of Seoul, South Korea, in 2017. He is currently pursuing the Ph.D. degree. His research interests include speaker recognition, acoustic scene classification, audio spoofing detection, and deep learning.

HEE-SOO HEO (Member, IEEE) received the B.S. and Ph.D. degrees in computer science from the University of Seoul, South Korea, in 2013 and 2019, respectively. Since 2020, he has been an AI Researcher at Naver Corporation. His research interests include machine learning, speaker recognition, acoustic scene classification, and image recognition.

HYE-JIN SHIM (Member, IEEE) received the B.A. degree in international trade and the M.S. degree in computer science, in 2017 and 2019, respectively. She is currently pursuing the Ph.D. degree. Her research interests include speaker recognition, audio spoofing detection, acoustic scene classification, and deep learning.

HA-JIN YU (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in computer science from KAIST, South Korea, in 1990, 1992, and 1997, respectively. From 1997 to 2000, he was a Senior Researcher at LG Electronics. From 2000 to 2002, he was the Director at SL2 Ltd. Since 2002, he has been a Professor with the Department of Computer Science, University of Seoul. His research interests include the speech and speaker recognition, acoustic scene classification, and machine learning. He is an Editor of the Journal of the Korean Society of Speech Sciences.