IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 8, 2019, accepted July 19, 2019, date of publication July 29, 2019, date of current version August 14, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2931656

An Adversarial Feature Distillation Method for Audio Classification

LIANG GAO, HAIBO MI, BOQING ZHU, DAWEI FENG, YICONG LI, AND YUXING PENG
National Key Laboratory of Parallel and Distributed Processing, National University of Defense Technology, Changsha 410073, China
Corresponding author: Haibo Mi (haibo_mihb@126.com)

This work was supported by the National Key Research and Development Program of China under Grant 2016YFB1000101.

ABSTRACT The audio classification task aims to discriminate between different audio signal types. In this task, deep neural networks have achieved better performance than the traditional shallow architecture-based machine-learning method. However, deep neural networks often require huge computational and storage requirements that hinder the deployment in embedded devices. In this paper, we proposed a distillation method which transfers knowledge from well-trained networks to a small network, and the method can compress model size while improving audio classification precision. The contributions of the proposed method are two folds: a multi-level feature distillation method was proposed and an adversarial learning strategy was employed to improve the knowledge transfer. The extensive experiments are conducted on three audio classification tasks, audio scene classification, general audio tagging, and speech command recognition. The experimental results demonstrate that: the small network can provide better performance while achieves the calculated amount of floating-point operations per second (FLOPS) compression ratio of 76:1 and parameters compression ratio of 3:1.

INDEX TERMS Convolutional neural networks, audio tagging, knowledge distillation, model compression.

I. INTRODUCTION

Hearing is one of the most important sensory and information source for humans, while ''machine listening'' is very challenge for machines. The automatic processing and analysis algorithms of audio have broad application prospects. Nevertheless, how to fully utilize audio data is still far from been solved, as acoustical events may occur at any time and accompanied by much random background noise. Sustainable efforts have been made to improve the performance of automatic models.

In the previous work, building ''bag of audio words'' [1] had been applied for audio classification. In addition, K-means [2] and spectral clustering methods [3] are used in unsupervised music retrieval. Shao *et al.* [4] proposed to measure the audio similarity by hidden Markov models [5] for audio clustering. Kumar and Raj [6] used a support vector machine (SVM) based multi-instance learning system for audio tagging and acoustic event detection. But these methods could not learn contextual features and utilize the correlation between different events. Some researchers depend on deep learning to solve tasks such as acoustic scene classification, event detection, and feature learning. The classifier based on convolutional neural networks (CNN) [7], [8] shown their remarkable capabilities in multiple acoustic tasks. However, due to insufficient label data and many noise data, the performance of acoustic networks is still not satisfactory. While some methods can further enhance the model, they can cause an increase in model storage and computational complexity. For example, building deeper or wider networks [9], [10] are helpful to improve accuracy. In addition, the ensemble method which aggregated knowledge of multi models [11], [12] is effective to get high-performance networks.

This article focuses on the problem of how to achieve model performance improvements under limited resources. The acoustic models are typically deployed in embedded devices, small computers, and mobile devices, where lacking computational power and storage [10]. The ensemble network and deeper network could not be applied in the industry because of large network redundancy, slow calculation speed, and high resource occupancy [13], [14]. There had been many works to improve the accuracy with limited resources, but hardly in acoustic scenarios. Model compression has

The associate editor coordinating the review of this manuscript and approving it for publication was Shiqing Zhang.

become a hot research issue which aims to achieve better performance in a small model. This paper proposes an effective model compression method for acoustic scenes.

To exert the advantages of deep learning in dealing with acoustic tasks, the problem of high resource occupation on the acoustic network has to be solved. In this paper, we proposed a CNN based method which transfers knowledge through feature maps to achieve the purpose of model compression and accuracy improving.

For model compression, knowledge distillation is an effective strategy which transfers knowledge of well-trained complex networks (teacher) to small networks (student). In 2014, Hinton proposed the traditional knowledge distillation [15], [16] method, in which the student network improves its performance by mimicking the soft labels of a well-trained teacher network. The knowledge distillation method has a wide range of applications in model compression [17] and semi-supervised tasks [18].

However, soft labels discard the details of the network about feature learning. In CNN structure, usually the convolutional layers as the feature extractor, and the fully connected layers composition the classifier. The feature map of the convolutional layer has more detailed information than the soft labels. That means transferring knowledge through feature maps would be a better choice. We recommend that the student network should simulates from the feature maps of a well-trained teacher network. And simulating from multi-level feature maps of teacher network helps to learn the complete features. Similarity measure functions such as Kullback Leibler (KL) divergence and Mean Squared Error (MSE) could be used to measure the differences between feature maps of teachers and students.

Due to the differences in network structure, data channel, and other factors, feature maps of student and teacher are of different size. In order to calculate feature similarity, the feature maps need to be re-interpreted into simpler terms of what the student network will understand more easily. In our method, adaptive pooling layers are used to re-interpret features maps. In addition, we introduced a discriminator to identify features from teacher or student, the generative adversarial strategy [19] force the target model to learn the features' expression of teacher network, which would strengthen the knowledge transfer. The discriminator judges whether the student has well learned the information implied in feature maps of the teacher, and optimizes the student by backpropagation algorithms. With the adversarial multi-level feature distillation, the small model improves its generalization ability.

We conducted experiments on three audio classification tasks, audio scene classification, general audio tagging, and speech command recognition. The results confirm that our method effectively improved the accuracy of models and achieved better performance of model compression. Compared with the general supervised learning method, our method achieved higher performance of mAP@3 92.54% on VGGNet while the mAP@3 of baseline is 91.5% in general audio tagging task. And the network trained with our method gets similar performance to the ensemble model, while the compression ratio of FLOPS is 76:1 the compression ratio of parameters is 3:1.

In summary, the proposed method has following advantages:

1. Our distillation method improves the performance of the CNN in audio tagging tasks.

2. Our method has better performance in model compression, no matter distillation from large models to small models or from multiple models to single models. Our approach reduces the network's resource requirements and complexity while keeping similar performance.

3. We improved the knowledge distillation method by combining the adversarial learning strategy and feature distillation in multi-level feature maps.

The organizational structure of this paper is as follows, the related work are introduced in the section II, the section III elaborates our method, the section IV is experiment and analysis, and our conclusion is given in the section V.

II. RELATED WORK

A. MODEL COMPRESSION

The large occupying of memory and calculations hinder the deep learning's promotion in embedded devices. In order to improve the computational efficiency of deep learning, some model compression methods have been proposed. Courbariaux et al. approached the binarized neural network method [20] which increases the computational speed and reduces network capacity by binarizing the weight and activation of the network. Hu et al. utilize network trimming method [21] to compress the network, removing neurons which are non-informative [22] would reduce network complexity without much precision loss. However, the shortage of network trimming is too much manually needed to adjust the pruning threshold in multiple iteration training. And binarized neural networks are limited by the original network structure, the method performs poorly in large models. Knowledge distillation method [16] transfers information from large networks a to small network to build smaller but effective networks.

B. KNOWLEDGE DISTILLATION

The knowledge distillation [23], [24] method is a teacher-student structure. The target network (student) learns from the well-trained network (teacher), mimicking the teacher's outputs to improve accuracy. In knowledge distillation, the knowledge of the well-trained network transferred to the small network, thus solving the problem of excessive redundancy and precision stagnation of the network to a certain extent. The tradition knowledge distillation (KD) [16] process uses the soft label information of the teacher network to induce the training of the student network. The feature distillation (FD) [25], [26] method, student network mimics the attention maps or the feature maps of teacher networks.

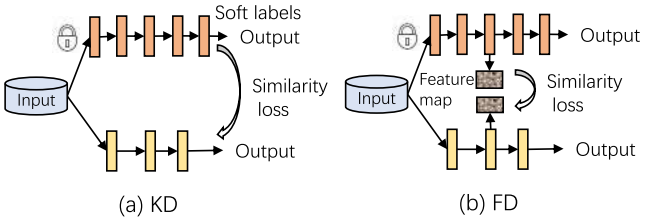


FIGURE 1. The structure of (a) traditional knowledge distillation and (b) feature distillation.

1) TRADITIONAL KNOWLEDGE DISTILLATION (KD)

The traditional knowledge distillation focuses on the class-level distribution of predicted probabilities (soft labels), which corresponds to the soften class-level probability scores. In soft label based knowledge distillation as Figure 1(a) shows, a small network with simple structure simulates the soft label distribution of a stronger teacher network with complex learning capabilities which provide more detailed knowledge than the onehot label.

2) FEATURE DISTILLATION (FD)

Jangho et al. proposed using feature-based distillation [25] instead of soft label based distillation, the structure shown in Figure 1(b). The feature maps of networks contain more information about their structure and details of samples compared with soft labels. In the process of feature distillation, using reconstruction losses to characterize differences in feature maps between teacher and student. The intermediate feature map of the trained network is the selective expression of specific parts of the sample. Distillation based on feature layers paid more attention to detailed sample information, such as the shape of samples and the color of samples. Direct learning from the teacher's feature maps is more straightforward for student network which can reduce the loss of knowledge that maps features to soft tags or attentio maps.

3) KNOWLEDGE TRANSFER WITH ADVERSARIAL LEARNING STRATEGY

Heo *et al.* [27] uses the adversarial learning strategy to induce student network training by using generated samples to effectively improve the ability of students to identify decision boundaries. Generative adversarial networks (GAN) [28]–[30] produces fairly good output through mutual confrontation learning of two modules in the framework: the generative model and the discriminative model. With the help of the adversarial strategy to distinguish the boundary between learning classes (in discriminator), the generator are able to produce more realistic false samples. In the knowledge distillation, the students have similar mechanisms to the generator in GAN, which imitates the teacher network's outputs. Adding the adversarial loss in the process of feature distillation would strengthen the ability of feature learning.

The previous distillation methods mainly focused on the computer vision field, lacking research on acoustic data. We use multi-level feature distillation to obtain more detailed information than single-level feature distillation, and the adversarial loss was added to monitor the knowledge transfer process.

III. METHODOLOGY

The overview structure of our distillation method shown in Figure 2. The framework of our structure consists of two branches, a branch of teacher network that had been well-trained and a branch of student network training from scratch. The teacher network provides multi-level features' instructional knowledge to guide the student's training process. We combine the adversarial learning mechanism and multi-level feature distillation to motivate target network to learn from trained networks. For each input, the target network imitates the trained model from multi-level features by reducing the knowledge difference between the feature of student and teacher. The adversarial units are used to enhance the feature distillation. Our framework strengthens

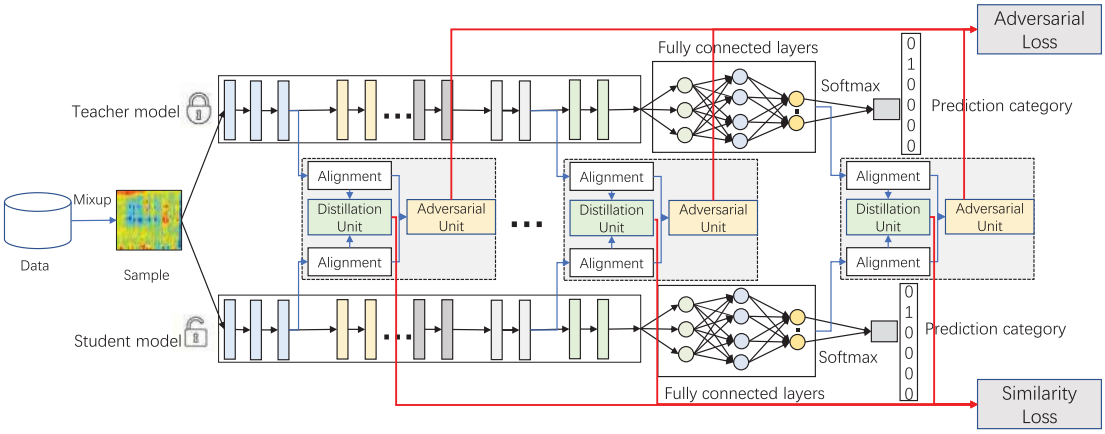


FIGURE 2. The overview of our approached method. The teacher and student get the same input which has adopted mixup operation. The teacher network's parameters is fixed. At the selected layers, teacher and student produce features, minimizing their similarity loss and Discriminator loss to optimize the student network.

the knowledge transfer from teachers to the student, which making the student stronger and compressing the model size. The procedure of our approach involves two steps, (1) pre-training of the teacher networks with general supervised learning method, (2) transferring knowledge from teacher networks to student network with feature distillation method.

A. PRE-TRAINING OF TEACHER NETWORKS

For knowledge distillation a well-trained teacher is needed, firstly training the teacher networks in general supervised method. The following section described the general approach for audio tagging task using CNN, data processing and supervised learning.

1) DATA PREPROCESSING

In deep learning, audio tagging models generally use intercepted segments Mel-frequency Cepstral Coefficients (MFCC) and Log-Scale Mel-frequency Spectrum (Logmel) as input features. The Logmel had been considered as the best features of CNN training in acoustic tasks. To convert the raw data into Log Mel-spectrogram which used as the CNN input in training and testing. For the audio files of raw data, firstly downsample them with an appropriate sampling rate, then use Short-Time Fourier Transformation (STFT) method to analyze their frequency and phase. Then Log Mel-spectrogram can be obtained by applying the Mel filter bank, followed by a logarithm scaling, finally divided by the standard deviation and subtracting the mean value to normalize them. The delta and delta-delta features of original Logmel is calculated with a specific widow size. The original Logmel features, delta and delta-delta features form the three-channel Logmel samples. Each Logmel sample inherits the label of original audio.

2) PRE-TRAINING PROCESS OF TEACHER NETWORKS

For teacher networks, training by general supervised training method, using the Logmel as the inputs. The teacher network is trained by traditional supervised learning method. Minimizing the cross-entropy loss between predicted scores and ground-truth labels to optimize models by back propagation algorithm. Given a data set $D = (X, Y)$, where X are the Logmel form of the original dataset, and Y are corresponding one-hot target labels. We minimize the cross-entropy loss of model's classification probability $F_t(X)$ and target label Y. The cross-entropy loss function is:

$$L_C = -\frac{1}{n}\sum_{(x_i,y_i)\in D}[y_i \ln F_t(x_i,\theta_t) + (1-y_i)\ln(1-F_t(x_i,\theta_t))] \quad (1)$$

In the formula, n is the number of samples in dataset D, θ_t is the model's parameter. By minimizing the cross-entropy loss to optimize the model parameter θ_t by the backpropagation algorithm.

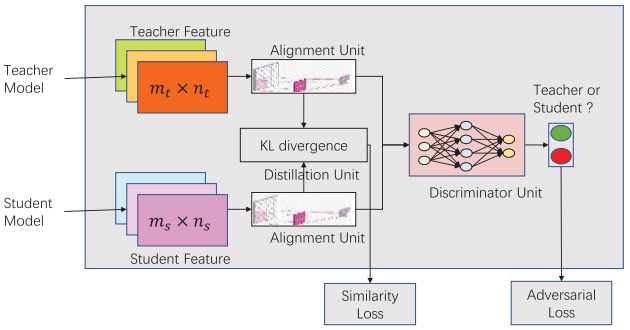


FIGURE 3. The modules for our distillation method.

B. THE DISTILLATION PROCESS

The modules for our feature distillation are shown in Figure 3, which contains alignment unit, distillation unit, and adversarial unit. The alignment unit is composed of the adaptive pooling layer which aligns feature maps of networks. The distillation unit guides the student to learn from the teacher by reducing the KL divergence loss between their feature maps. The adversarial unit (discriminator) used to force the student network producing features that confused with the teachers' features, which facilitate the transfer of knowledge.

For each distillation iteration, the teacher network and the student network get the same input. The two networks generate multi-level features in intermediate layers. The alignment unit reshapes features into vectors of the same dimension by stacked adaptive average pooling layers. The discriminator consists of a set of fully connected layers. The discriminator and the student network (generator) constitute a generative adversarial network (GAN). The GAN encourages the student network generating features similar to the teacher which can deceive the discriminator. In distillation units, the student network minimizes the KL loss of features between the teacher network and the student network to guide the training process. In addition, we use the sample mixup method to generate new samples in the neighborhood domain, which will enhance dataset generalization. We will introduce our approach in detail in the following sections.

1) ALIGNMENT UNITS ALIGN DIFFERENT FEATURES

The alignment units are responsible for converting the intermediate output features to the same shape, the features come from the student network and the teacher network. The similarity of the features of the two sources can be compared only if their dimensions are the same. The adaptive pooling layers as the alignment unit, which maintaining the features' structure of the model while the feature dimension changed and aligning the features of network with different structures. Unlike the general maximum pooling and average pooling, the adaptive pooling layer can produce the fixed size output from different scale inputs. Adaptive pooling adaptively calculates the pooling kernel size by the following formula based the input dimension and the output dimension:

$$kernel\_size = (input\_size + 2 \times padding) - (output\_size - 1) \times stride \quad (2)$$

For a teacher-student structure with different levels and feature map sizes, we take a feature layer alignment as an example. To distillation in the teacher feature layer f_t and the student feature layer f_s. According to the Formula 2, the kernel size of adaptive pooling layer is calculated for the teacher and the student respectively. The pooling operation converts f_t and f_s into the feature matrix of the same size, and then the feature matrix is averaged in channels to obtain a translated simplified feature vector.

The value of output size, padding and stride are determined before pooling operation, and kernel size is calculated based on the value of input size using the above formula. After alignment, we get vectors of a pre-set dimension both on student network and teacher network.

2) FEATURE DISTILLATION UNIT

The optimization of student network F_s does not depend on the one-hot target label. The student network's parameters θ_s are optimized by minimizing the similarity loss between the teacher network's features and student network's features. We calculate similarity loss of with Kullback Leibler divergence, which is usually used to measure the difference of probability distributions. We use it to measure the similarity loss when the student feature distribution $f_s(x)$ fits the feature distribution $f_t(x)$ of the teacher network. $f_s(x)$ is the aligned feature of the student network and $f_t(x)$ is the aligned feature of the teacher network. So the similarity loss between the feature maps of teacher and student calculated by Kullback Leibler divergence as follows:

$$\begin{aligned} L_{KL} &= -\frac{1}{m \times n} \sum_{x \in X} \sum_{i=1}^{m} f_t^i(x) \log \frac{f_t^i(x)}{f_s^i(x)} \\ &= \frac{1}{m \times n} \sum_{x \in X} \sum_{i=1}^{m} (-f_t^i(x) \log f_t^i(x) + f_t^i(x) \log f_s^i(x)) \end{aligned} \quad (3)$$

where m is the number of distillation features' layers, $f_t^i(x)$ and $f_s^i(x)$ are the i-th aligned feature layer in teacher network and student network. In the knowledge distillation process, the teacher network parameters are fixed. The second part of the KL loss is only related to the teacher network. It is a constant value, discarding it the similarity loss function would be:

$$L_{KL} = -\frac{1}{m \times n} \sum_{x \in X} \sum_{i=1}^{m} f_t^i(x) \log f_s^i(x) \quad (4)$$

For each of the inputs, the student network is guided by multiple feature layers of teacher network, so the student network is easier to learn the knowledge than the traditional distillation with only one layer of soft label guidance.

3) ADVERSARIAL UNITS LEARNING FEATURES' DISTRIBUTION

We use the adversarial units (discriminator) to supervise the training of student networks. The student network is the generator of generative adversarial net (GAN) which tried to imitate the teacher, and the teacher network features are adversarial samples. In our method, the discriminator is a classifier composed of fully connected layers. We spliced the teacher features $f_t(x)$ and student network features $f_s^i(x)$ as the input of discriminator, then the discriminator classify try to identify a feature from teacher or student. The target formula of generative adversarial net is as follows:

$$\begin{aligned} \min_{F_s} \max_{F_d} V(F_d, F_s) = \frac{1}{m} \sum_{i=0}^{m} [&E \log F_d(f_t^i(x)) \\ &+ E \log (1 - F_d(f_s^i(x)))] \end{aligned} \quad (5)$$

In the above formula, F_s is the student network (generator), F_d is the discriminator, V is the function about F_d and F_s, m is the selected feature layers' number for distillation. In the optimization phase, firstly optimize the discriminator. Maximize the output value of the discriminator when inputting the teacher feature, the target formula is:

$$\begin{aligned} \max_{F_d} V(F_d, F_s) = \frac{1}{m} \sum_{i=0}^{m} [&E \log F_d(f_t^i(x)) \\ &+ E \log (1 - F_d(f_s^i(x)))] \end{aligned} \quad (6)$$

And then optimize the student network (generator), minimize the output value of the discriminator when inputting the student feature. The teacher parameters are fixed, so delete the first item, the target formula is:

$$\min_{F_s} V(F_d, F_s) = \frac{1}{m} \sum_{i=0}^{m} E[\log (1 - F_d(f_s^i(x)))] \quad (7)$$

So the adversarial loss for student network is

$$L_A = \frac{1}{m} \sum_{i=0}^{m} E[\log (1 - F_d(f_s^i(x)))] \quad (8)$$

The power of the adversarial units is that it automatically learns the data distribution of the teacher features. Adversarial learning of generators and discriminators in generative adversarial net enhances students' ability to learn from trained model.

4) LOSS COMBINATION

Based on the similarity loss L_{KL} and adversarial loss L_A obtained from the previous analysis, we combine them for the distillation of student network. The final loss of student as follows:

$$L = L_{KL} + L_d \quad (9)$$

The final loss motivates the student network to mimic and absorb the teacher's knowledge.

C. ALGORITHM

The proposed distillation method can be divided into three cases depending on the network structure of the teacher and the student. In the first case, both the teacher and the student use the same structure. In the second case, the teacher and

Algorithm 1 Algorithm of the Proposed Adversarial Feature Distillation Method

Pre-training teacher networks:

First identifying and building a collection of teacher networks, then minimizing the cross-entropy loss $L_C^j(f_t(X), Y)$ with the standard supervised learning approach to train the teacher networks and saving them. In our experiment, teacher networks including VGGNet, ResNet and DenseNet.

Feature distillation:

Teacher zoo $T=\{T_0, T_1, T_2\}$, student network f_s, dataset $D = \{X, Y\}$.
for Each iteration **do**
  Alternate select teacher network f_t = Alternate(T),
  for Each distillation feature layer i **do**
    Align feature maps Align$(f_s(X), f_t(X))$,
    Similarity loss $L_{KL}(f_s(X), f_t(X))$,
    Adversarial loss $L_d(f_s(X), f_t(X))$,
    Back propagation optimization for i-th adversarial unit,
  end for
  The distillation loss $L = L_{KL} + L_d$,
  Back propagation optimization for student f_s.
end for

the student are randomly selected structures that are different from each other. The third case, multiple teacher networks correspond to one student network, alternately using the teacher networks to guide the distillation process of the student network in each iteration. In the multiple teacher distillation case, which implicitly ensemble the knowledge of multiple teachers into a single student. The procedure of our distillation method shown in Algorithm 1. The training process is divided into two phases, pre-training the teacher network and feature distillation. In the pre-training phase, minimizing the cross-entropy loss $L_C(f_t(X), Y)$ to optimize model parameters of teachers with the ground-truth label. The second step, at the adversarial feature distillation stage, minimizing the loss of adversarial loss $L_d(f_s(X), f_t(X))$ and similarity loss $L_{KL}(f_s(X), f_t(X))$ between the teacher networks and the student networks in each iteration.

IV. EVALUATION

We have step by step verified the effectiveness of the proposed method in improving model accuracy and model compression. The structure is implemented based on the Pytorch and evaluates on three large-scale audio classification tasks. Networks optimize with stochastic gradient descent (SGD) algorithm, and minibatch size is set as 64. For the student network, the learning rate is initiated to 0.01, and the decay and Nesterov momentum was 0.0001, and 0.9. For the adversarial unit, the initial learning rate is set to 0.001. The NVIDIA 1080 graphics cards were used for all the experience. To evaluate experimental results, the average mean accuracy (mAP@3) and accuracy indicator are typically used.

A. DATASET

To verify our method, three audio classification dataset were involved, two from the challenge of Detection and Classification of Acoustic Scenes and Events (DCASE) and the other one is the Google Speech Commands Dataset [31].

1) AUDIO SCENE CLASSIFICATION TASK

The challenge of Detection and Classification of Acoustic Scenes and Events 2018 (DCASE18) aims to enhance the development of audio computing scenarios and analysis methods. The DCASE18 task 1A [32] aims to classify audio scene of 10 predefined environment classes such as "park", "metro station". The dataset of DCASE18 task 1A is from the TUT Urban acoustic scenes 2018 dataset [33], which contains recordings from various acoustic scenes over six European cities. Each original record was 5-6 minutes of audio, which was divided into segment files of 10 seconds. The dataset consist of 10 acoustic scenes, "airport, shopping mall, metro station, street pedestrian, public square, street traffic, tram, bus, metro, park". Each acoustic scene having 864 files, all 8640 files, among the dataset 70% for training data and 30% for evaluation.

2) GENERAL AUDIO TAGGING TASK

The DCASE18 task 2 [34] is for general-purpose audio tagging task. The data samples come from Freesound [35] audio samples which labeled with AudioSet Ontology tags. And the dataset has 41 categories in all, while data samples were unreliable labeled. All the sample data sets are compressed to the pulse code modulation (PCM) 16 bits of 44.1KHz mono-channel audio. In the training set, uneven 9.5K samples among 41 categories were contained. The minimum number of audio for one category is 94 and the maximum number is 300. About 3.7K manual validation samples and about 5.8K unauthenticated samples contained in the training set. The test set contains about 1.6K manual validation samples and about 7.8K unauthenticated annotation samples. In our experiments, only the 1.6K manual validation data were used as the test set.

3) SPEECH COMMANDS CLASSFICATION TASK

The Google Speech Commands Dataset [31] is a dataset of one-second audio files, each contains a spoken English command word. In all about 65000 files, all of the samples are converted to 16-bit little-endian PCM-encoded wave files at a 16000 sample rate. In 30 categories, the core words include "Yes, No, Up, Down, Left, Right, On, Off, Stop, Go, Zero, One, Two, Three, Four, Five, Six, Seven, Eight, Nine" and ten auxiliary words include "Bed, Bird, Cat, Dog, Happy, House, Marvin, Sheila, Tree, Wow". In our experiment, only 6000 used training samples among the dataset.

In all experiments, samples were converted into Log Mel-spectrogram form for CNN training and testing. For all dataset, firstly using Short-Time Fourier Transformation (STFT) method to analyze the audio frequency and

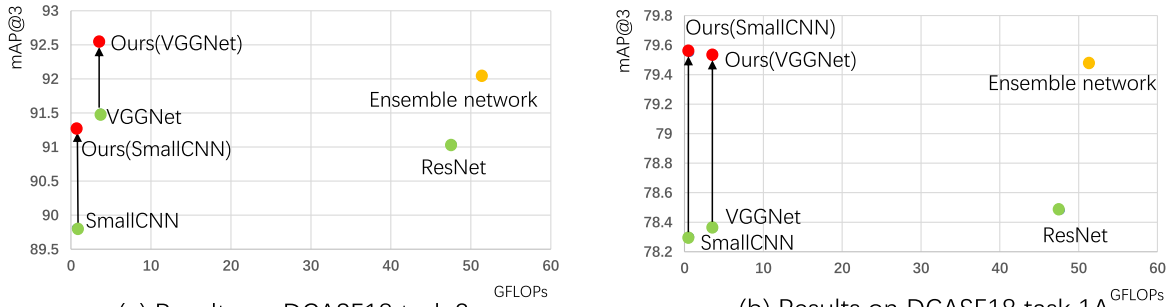


FIGURE 4. The result on DCASE18 task.

phase. Then Log Mel-spectrogram can be obtained by applying the Mel filter bank, 64 was set as the number of bandpass filters. Then converting Mel-spectrogram to a logarithmic scale, finally normalizing it by dividing by the standard deviation subtracting the mean value.

B. NETWORKS

We implemented experiments with several popular convolutional neural networks, VGGNet [36], ResNet [9], and DenseNet [37]. In addition, we designed a SmallCNN which was five convolutional layers and a classifier of two fully connected layers. The VGGNet uses continuous 3×3 convolution cores to replace the larger convolution cores ($11 \times 11, 7 \times 7, 5 \times 5$) in AlexNet, which ensures that the depth of the network is improved and better performance. The deep residual network ResNet introduced the residual structure, strengthened the ability of deep network feature transmission, and solved the problem that the accuracy of network deepening does not decrease. The DenseNet establishes dense connections with all the previous layers and the back layer, which solves the problem of gradient disappearance. DenseNet, ResNet and VGGNet have been widely adopted and achieved competitive results in image recognition task, image segmentation task, and speech recognition task. We select the network layer numbers of the three structures according to their best results. For VGGNet we use the 19-layer network VGGNet19, and for ResNet, we use the 101-layer network ResNet101. We used the 121-layer network DenseNet121.

C. THE RESULTS OF OUR DISTILLATION METHOD

Aggregating information from multiple networks enables more precise network, but the ensemble method liner increases the computational complexity and resource usage, which is only applicable to academic research and competitions. Our distillation method could implicit aggregate and transfer knowledge from multiple teacher networks into a student network. We compared the result of ensemble method and our distillation method in Table 1 and Table 2, and more intuitive results are shown in Figure 4. The result on DCASE18 task 1A and task 2 as respectively shown in Table 1 and Table 2, the first three lines are the baseline results of

TABLE 1. The results (%) and number of gigaFLOPS (GFLOPs),number of parameters (Para) for different network training on DCASE18 task 2 dataset. For 'Ours', we use two teacher networks, VGGNet and ResNet.

Model	GFLOPS	Para($\times 10^7$)	ACC	mAP@3
VGGNet	3.69	4.53	88.06	91.5
ResNet	47.69	4.26	88	91.04
SmallCNN	0.66	2.92	86.05	89.81
E(VGGNet+ResNet)	51.38	8.79	88.6	92.04
Ours(VGGNet)	3.69	4.53	89.31	92.54
Ours(SmallCNN)	0.66	2.92	87.68	91.28

TABLE 2. The results (%) and number of GFLOPs,number of parameters (Para) for different network training on DCASE18 task 1A dataset. For 'Ours', we use two teacher networks, VGGNet and ResNet.

Model	GFLOPS	Para($\times 10^7$)	ACC	mAP@3
VGGNet	3.69	4.53	68.19	78.36
ResNet	47.69	4.26	68.74	78.48
SmallCNN	0.66	2.92	67.1	78.3
E(VGGNet+ResNet)	51.38	8.79	69.28	79.47
Ours(VGGNet)	3.69	4.53	69.61	79.54
Ours(SmallCNN)	0.66	2.92	69.14	79.55

VGGNet, ResNet, and SmallCNN. The E(VGGNet+ResNet) means the traditional ensemble of VGGNet and ResNet, as shown in the table the ensemble accuracy is 88.6%, mAP@3 is 92.04% on DCASE18 task 2, and accuracy is 69.28%, mAP@3 is 79.47% on DCASE18 task 1A. The ensemble method improved model performance but caused the increase of model size, the number of parameters is 8.79×10^7 and the FLOPS of 5.14×10^7.

In our distillation method, we implicit ensemble two teacher networks to a single student network. Our distillation SmallCNN got the result of mAP@3 91.28% on DCASE18 task 2 and 79.55% on task 1A. Our distillation method decreased the amount of calculation and the number of parameters. After distillation on VGGNet and SmallCNN, the single model accuracy closed even higher than ensemble model. The compression ratio of FLOPS is 76:1 and the compression ratio of parameters is 3:1 on SmallCNN compare with the ensemble teacher VGGNet and ResNet. For VGGNet, our method got the best results, the mAP@3 of 92.54% on DCASE18 task 2 and 79.54% on task 1A.

TABLE 3. Ablation study results of VGGNet on DCASE18 task 2 dataset. The table shows the accuracy and mAP@3(%) of VGGNet in different experiment setting. '1-level' means traditional knowledge distillation, 'Multi-level' stands for multi-level feature distillation, 'Adversarial' represents adding the discriminator of adversarial learning strategies, and 'Mixup' represents the mixup data augmentation method.

1-level	Multi-level	Adversarial	Mixup	Accuracy	mAP@3
VGGNet baseline (Simonyan and Zisserman 2015)				86.18	89.76
√	—	—	—	86.43	90.15
√	√	—	—	87.19	90.39
√	√	√	—	87.45	90.96
—	—	—	√	88.06	91.5
√	√	—	√	89.16	92.24
√	√	√	√	89.27	92.45

D. ABLATION STUDY

To find out the influence of different modules in our structure, we designed ablation study of controlled experiments on the DCASE18 task 2 dataset. Our distillation structure contains modules of feature distillation unit, adversarial unit, and combined with the data mixup strategy. In order to distinguish the improvement of experimental results by each module, we perform a set of experiments on a different combination of modules on VGGNet. The results are presented in Table 3, and all experiments have the same settings as baseline except for the module mentioned in the table.

In order to distinguish the improvement of experimental results by each module, we perform a set of experiments on a different combination of modules on VGGNet. The experimental results are presented in Table 3, and all experiments have the same settings as baseline except for the module mentioned in the table. The VGGNet baseline mAP@3 trained under the traditional supervised learning method is 89.76%. In term of the multi-level knowledge distillation which contains five intermediate layers for distillation. The multi-level feature distillation mAP@3 is 90.39%. Further, the adversarial learning units are added in each knowledge distillation feature layer, achieved the mAP@3 of 90.96%. Using one of the three strategies alone, we can observe that the mAP@3 of the sample mixup method is best (91.5%). Combined with the multi-level feature distillation, adversarial learning, and data mixup, the network gets the best performance of accuracy 89.27% and the mAP@3 92.45%.

E. COMPARED WITH OTHER IMPLICIT KNOWLEDGE AGGREGATION METHOD

We conducted comparison experiments among the general supervised learning method, the traditional knowledge distillation method and our distillation method. The baseline is the results of the general supervised training method, the knowledge distillation (KD) [16] is the implicit knowledge aggregation method which learning from teacher network's soft labels which is suitable for small shallow networks. The data enhancement method (Mixup) [38] directly aggregates information of samples before training, which can effectively improve the generalization of the model. We use the sample mixup method to minimize neighborhood risk. In order to ensure the generality of the generated neighborhood samples, the random strategy is adopted in the mixup method, in which each time two samples are randomly selected from the original data to construct a new neighborhood sample by weighting sum them. The formula of mixup method as $x_{i,j} = \alpha * x_i + \beta * x_j, y_{i,j} = \alpha * y_i + \beta * y_j$, where x_i and x_j are samples from original data, y_i and y_j are corresponding labels of x_i and x_j, α and β are random numbers in the 0 to 1 interval, $\alpha + \beta = 1$, $x_{i,j}$ and $y_{i,j}$ are the generated neighborhood sample and the corresponding label. We experimented with the DCASE18 task 2 dataset and the speech commands dataset using VGGNet, ResNet and DenseNet. For each model, the four different training method were adopted. On the DCASE18 task 2 dataset, we used all of the released training data and the mAP@3 as evaluation criteria. In Table 4, we can observe that the mAP@3 values of VGGNet, ResNet, and DenseNet baseline on the DCASE18 task 2 data set are 89.76%, 89.42%, 89.65%. The knowledge distillation improved a slight boost in the model performance, but it is more suitable for shallow networks. Information fusion at the sample level (Mixup) yields better performance than KD. Our distillation get the biggest increment as FIGURE 4 shows, the mAP@3 of trained with our method compared with baseline increased 2.69%, 2.95% and 2.93% in VGGNet (89.76 VS 92.45), ResNet (89.42 VS 92.37) and DenseNet (89.65 VS 92.58).

TABLE 4. Results (mAP@3 %) on DCASE 2018 task 2 dataset for different training strategy. In the table "Baseline" is the traditional supervised learning method, "Mixup" is the data enhancement of mixup method, "KD" represents the traditional knowledge distillation method. And "Ours" represents the adversarial multi-level feature distillation with mixup mechanisms.

Model	Baseline	KD	Mixup	Ours
VGGNet	89.76	90.15	91.5	92.45
ResNet	89.42	90.07	91.32	92.37
DenseNet	89.65	90.28	91.66	92.58

The results of speech commands were shown in Table 5. We only used 6000 randomly selected training data for all the training, and the evaluation criteria are the accuracy rate. The results on the speech commands dataset show similar tends to the results on the DCASE18 task 2. The VGGNet, ResNet and DenseNet model used our method for training separately achieved 95.62%, 95.92%, 95.38% accuracy (the baseline is 95.4%, 95.74%, 93.6%).

The results show that in almost all cases, our knowledge distillation method achieved higher precision than the separate mixup method and the traditional knowledge distillation.

F. THE IMPACT OF DIFFERENT SIMILARITY LOSSES

We improved the CNN accuracy on acoustic tasks through the adversarial feature distillation method. The teacher network supervises student network's training with feature maps. Our distillation process based on KL divergence can significantly

TABLE 5. Accuracy (%) on speech commands dataset. In the table "Baseline" is the traditional supervised training method, "Mixup" is the data enhancement of mixup method, "KD" represents the traditional knowledge distillation method. And "Ours" represents the adversarial multi-level feature distillation with mixup mechanisms.

Model	Baseline	KD	Mixup	Ours
VGGNet	94.01	94.1	95.4	95.62
ResNet	95.74	95.73	95.83	95.92
DenseNet	93.6	94.16	94.85	95.38

TABLE 6. The mAP@3 (%) results on DCASE18 task 2 with VGGNet while distillation on different similarity loss (MAE, MSE, and KL loss).

Model	Baseline	MAE	MSE	KL
VGGNet	91.5	91.28	91.08	92.45
ResNet	91.04	91.42	91.48	92.37

improve the performance of the student network. The distilled model has a higher mAP@3 than model of standard training (92.45% VS 91.5%). But there are different types of loss functions can metric feature maps' distribution difference. We tested three common and effective similarity measurement functions of Mean Squared Error (MSE), Mean Absolute Error (MAE), and Kullback Leibler Divergence (KL). We compared the model accuracy of our feature distillation method under three different losses, the results are shown in Table 6.

In the experiment, the teacher network and student network are of the same structure (VGGNet or ResNet). With the KL loss, the distilled model get the best performance (the mAP@3 92.45% on VGGNet and 92.37% on ResNet), while with the MAE and MSE loss performance becomes worse. The KL loss is the most effective one on distillation. The MAE loss function is more robust to the outliers, but the derivative is not continuous making the process of optimal solution inefficient. The MSE loss is susceptible to local interference, but the optimization process is smoother and more stable. And KL loss is the logarithmic difference expectation of the probability between the raw distribution and the approximate distribution, which is robust and easy to optimize. It is more suitable for comparing the information loss between two distributions.

G. SEGMENT SCORES AGGREGATION

In the audio tagging task, usually a long audio file is divided into multiple segments. Aggregating the multiple segments score could effectively improve the classification accuracy. On the DCASE18 task 2 dataset, the samples of audio files duration range from 300ms to 30s. We divide the data into segments of each length is 1.5 seconds, and the insufficient length will be filled with constants. The audio-level score is the average of segment scores on an audio file. The audio-level score calculated as the formula $P_i = \frac{1}{T}\sum_{j=0}^{T} P_{i,j}$, where T is the number of segments divided from an audio, and P_i is the soft label of i-th audio file, $P_{i,j}$ is the soft label of

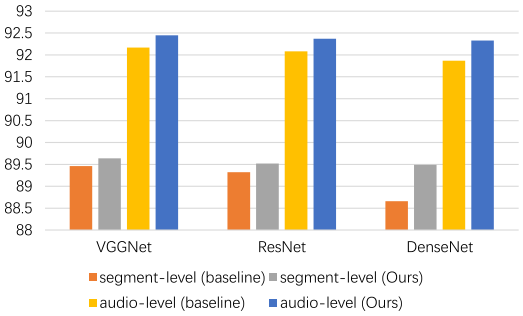


FIGURE 5. The comparison on segment-level score and audio-level score mAP@3 (%) on DCASE18 task 2 dataset.

TABLE 7. Accuracy (%) on speech commands dataset with cross-structure distillation. Improvement is compared to baseline VGGNet with accuracy of 95.4%.

Teacher	Accuracy	Student	Accuracy	Improvement
VGGNet	95.4	VGGNet	95.62	0.22
ResNet	95.74	VGGNet	95.86	0.46
DenseNet	93.6	VGGNet	95.92	0.52

j-th segment from i-th audio file. We compared the predicted scores of segment-level and audio-level in Figure 5. From the results, it can be found that the audio-level score is much more precise than the segment-level score. The reason may be that the tagging acoustic event occurred on an audio file only occurs within a certain range, and randomly intercepting the segment from the audio file may only contain part of the sound information or random noise. The classification of audio-level samples is clearly more scientific, reducing the interference of randomly selected segments noise.

H. DISTILLATION OF CROSS-STRUCTURE NETWORK

The cross-structure distillation would bring additional supplemental information. We experimented with cross-structure model on the speech commands dataset. Setting the VGGNet network as the student network, we compared the results, with VGGNet, ResNet and DenseNet separately as teacher network. The results shown in Table 7. As we mentioned before, VGGNet's baseline accuracy is 95.4%. In the experiment of VGGNet as the teacher, the target VGGNet training with our method obtained the accuracy of 95.62%, with ResNet as a teacher the target VGGNet accuracy is 95.86%, and with DenseNet as teacher network the target VGGNet accuracy is 95.92%. The cross-structure feature distillation achieved bigger improvement to the target network. That proves the differences in network structures bring more complementary knowledge for the target model, which leads to performance improvements.

I. DISTILLATION FROM BIG MODEL TO SMALL MODEL

It is easy for complex large networks to extract features from data, transferring knowledge from a complex model to the shallow network would be helpful. We verified the

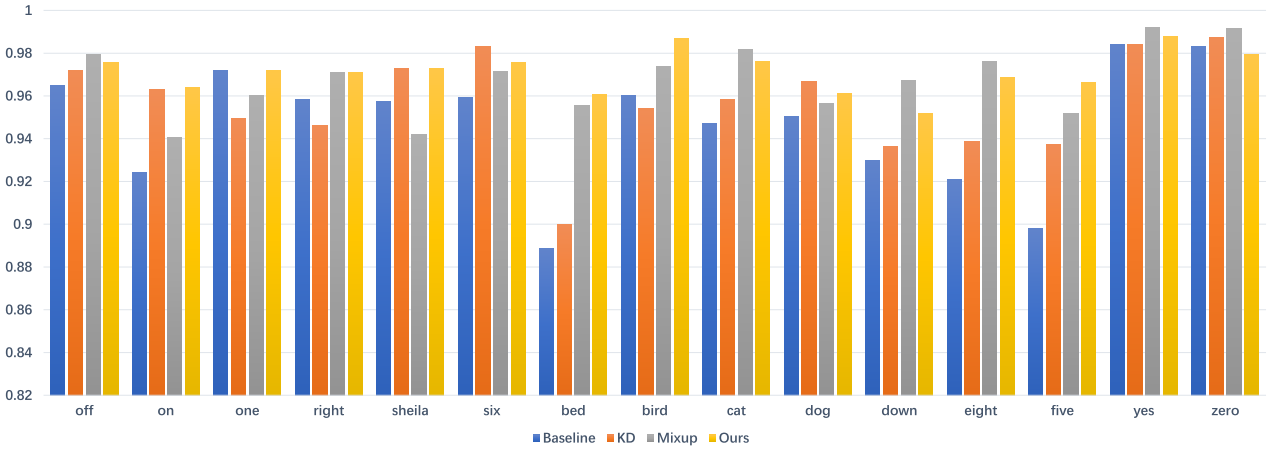


FIGURE 6. The classification accuracy comparison of VGGNet with different training methods on speech commands dataset. "Baseline" is the traditional supervised learning method, "Mixup" is mixup data enhancement, "Distillation" represents our adversarial multi-level distillation method without mixup and "Ours" represents the adversarial multi-level distillation with mixup data enhancement mechanisms.

TABLE 8. The accuracy (%) of cross-structure distillation on DCASE18 task 1A and task 2. The baseline SmallCNN get the accuracy and mAP@3 of 86% and 89.81% on DCASE18 task 2, and on DCASE18 task 1A.

Dataset	Teacher	Student	ACC	mAP@3
Task 1A	—	SmallCNN	67.1	78.3
	VGGNet	SmallCNN	68.56	79.05
	ResNet	SmallCNN	69.42	79.47
Task 2	—	SmallCNN	86.0	89.1
	VGGNet	SmallCNN	87.68	91.53
	ResNet	SmallCNN	86.88	90.5

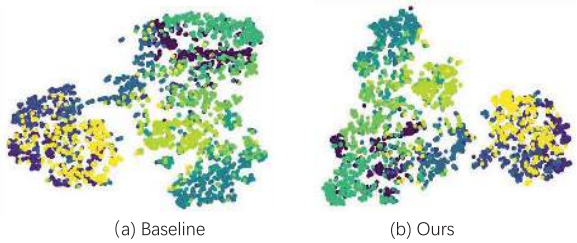


FIGURE 7. The t-sne diagram of SmallCNN which trained with baseline and our method.

compression and accuracy improving effect of our method, some classic CNN such as ResNet, VGGNet were used as the teachers of distillation, and the SmallCNN as student network. The results are shown in the Table 8, the FLOPS, the memory of networks can be found in Table 1. From Table 8, it can be found that both the accuracy and mAP@3 of the SmallCNN could benefit from a big model. The baseline SmallCNN achieve accuracy 67.1%, mAP@3 78.3% and accuracy 86.0%, mAP@3 89.1 on dataset DCASE18 task 1A and task 2. The powerful teacher network captured more features representing data relationships, and the feature distillation could transfer the information to a smaller student, which makes student network classification accuracy improved. In DCASE18 task 2, the SmallCNN get accuracy 87.68% and mAP@3 91.53% by distillation with teacher VGGNet. And in DCASE18 task 1A, the SmallCNN get the accuracy 69.42% and mAP@3 79.47% by feature distillation with ResNet as teacher.

J. THE T-SNE VISUALIZATION ANALYZES

Through visualization inspection, the improvement what we have made can be explored. utilizing the t-sne visual analysis method [39] to compare the embedded features of the baseline network and the embedded features of the network trained with our method. On the DCASE18 task 1A data set, for all 1200 test samples among 10 categories, the embed features of SmallCNN's last convolution layer were taken for t-sne analysis. The visualization results are shown in Figure 7, the similarities of our features are more compact, which proves that our features are better embedded.

K. THE ACCURACY COMPARISON ON CLASS LEVEL

We compared the accuracy of different training methods on 15 randomly selected categories. On the speech commands dataset, the categories contains "Off, On, One, Right, Sheila, Six, Bed, Bird, Cat, Dog, Squeak, Down, Eight, Five, Yes, Zero". The column chart of accuracy shown in Figure 6. Compared with the supervised learning method (baseline), the three other method achieved higher accuracy on almost all categories. The mixup method utilizes sample similarity at the sample level which was stronger than the general distillation. In the classes of "Bed, Bird, Cat, Eight, Five" the sample mixup method shows its advantage in improving model generalization, and training with our distillation method gets the best accuracy. Our method wins the first place in 7 of the 15 categories and won 6 second places. The results confirms that our distillation method is more stable and more effective than the other methods.

V. CONCLUSION

In this paper, for audio classification tasks, we have proposed a novel knowledge distillation method based on adversarial learning. Through multi-level feature distillation combined with adversarial units, our approach can compress model dramatically while improving the model's accuracy. We performed extensive experiments on the audio scene classification, general audio tagging and the speech commands recognition tasks. The experimental results demonstrate the effectiveness of the proposed method. For further research, we will focus on the application of knowledge distillation using multi-representations for the audio classification task.

REFERENCES

[1] M. Riley, E. Heinen, and J. Ghosh, "A text retrieval approach to content-based audio retrieval," in *Proc. Int. Symp. Music Inf. Retr. (ISMIR)*, 2008, pp. 295–300.

[2] G. Chen and B. Han, "Improve K-means clustering for audio data by exploring a reasonable sampling rate," in *Proc. 7th Int. Conf. Fuzzy Syst. Knowl. Discovery*, vol. 4, Aug. 2010, pp. 1639–1642.

[3] R. Cai, L. Lu, and A. Hanjalic, "Unsupervised content discovery in composite audio," in *Proc. 13th Annu. ACM Int. Conf. Multimedia*, 2005, pp. 628–637.

[4] X. Shao, C. Xu, and M. S. Kankanhalli, "Unsupervised classification of music genre using hidden Markov model," in *Proc. IEEE Int. Conf. Multimedia Expo*, vol. 3, Jun. 2004, pp. 2023–2026.

[5] G. Xia, D. Liang, R. B. Dannenberg, and M. J. Harvilla, "Segmentation, clustering, and display in a personal audio database for musicians," in *Proc. Int. Soc. Music Inf. Retr. Conf.*, 2011, pp. 139–144.

[6] A. Kumar and B. Raj, "Audio event detection using weakly labeled data," in *Proc. 24th ACM Int. Conf. Multimedia*, 2016, pp. 1038–1047.

[7] Y. Yin, R. R. Shah, and R. Zimmermann, "Learning and fusing multimodal deep features for acoustic scene categorization," in *Proc. ACM Int. Conf. Multimedia*, 2018, pp. 1892–1900.

[8] K. Xu, D. Feng, H. Mi, B. Zhu, D. Wang, L. Zhang, H. Cai, and S. Liu, "Mixup-based acoustic scene classification using multi-channel convolutional neural network," in *Proc. 19th Pacific Rim Conf. Multimedia*, Hefei, China, Sep. 2018, pp. 14–23.

[9] 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.

[10] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1–9.

[11] L. Deng and J. C. Platt, "Ensemble deep learning for speech recognition," in *Proc. 15th Annu. Conf. Int. Speech Commun. Assoc.*, 2014, pp. 1915–1919.

[12] X. Qiu, L. Zhang, Y. Ren, P. N. Suganthan, and G. Amaratunga, "Ensemble deep learning for regression and time series forecasting," in *Proc. IEEE Symp. Comput. Intell. Ensemble Learn.*, Dec. 2014, pp. 1–6.

[13] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *Proc. Brit. Mach. Vis. Conf.*, 2016, pp. 1–15.

[14] A. Canziani, A. Paszke, and E. Culurciello, "An analysis of deep neural network models for practical applications," 2016, *arXiv:1605.07678*. [Online]. Available: https://arxiv.org/abs/1605.07678

[15] J. Ba and R. Caruana, "Do deep nets really need to be deep?" in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2654–2662.

[16] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*. [Online]. Available: https://arxiv.org/abs/1503.02531

[17] Y. Cheng, D. Wang, P. Zhou, and T. Zhang, "A survey of model compression and acceleration for deep neural networks," 2017, *arXiv:1710.09282*. [Online]. Available: https://arxiv.org/abs/1710.09282

[18] N. Papernot, M. Abadi, U. Erlingsson, I. Goodfellow, and K. Talwar, "Semi-supervised knowledge transfer for deep learning from private training data," in *Proc. Int. Conf. Learn. Represent.*, 2017, pp. 1–16.

[19] M. Salvaris, D. Dean, and W. H. Tok, "Generative adversarial networks," in *Deep Learning With Azure*. Berkeley, CA, USA: Apress, 2018.

[20] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks: Training deep neural networks with weights and activations constrained to +1 or −1," 2016, *arXiv:1602.02830*. [Online]. Available: https://arxiv.org/abs/1602.02830

[21] H. Hu, R. Peng, Y.-W. Tai, and C.-K. Tang, "Network trimming: A data-driven neuron pruning approach towards efficient deep architectures," 2016, *arXiv:1607.03250*. [Online]. Available: https://arxiv.org/abs/1607.03250

[22] V. Lebedev and V. Lempitsky, "Fast convnets using group-wise brain damage," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 2554–2564.

[23] Z. Shen, Z. He, and X. Xue, "Meal: Multi-model ensemble via adversarial learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 4886–4893.

[24] R. Anil, G. Pereyra, A. Passos, R. Ormandi, G. E. Dahl, and G. E. Hinton, "Large scale distributed neural network training through online distillation," 2018, *arXiv:1804.03235*. [Online]. Available: https://arxiv.org/abs/1804.03235

[25] J. Kim, S. Park, and N. Kwak, "Paraphrasing complex network: Network compression via factor transfer," in *Proc. Neural Inf. Process. Syst.*, 2018, pp. 2760–2769.

[26] S. Zagoruyko and N. Komodakis, "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer," *CoRR*, vol. abs/1612.03928, 2017. [Online]. Available: http://arxiv.org/abs/1612.03928

[27] B. Heo, M. Lee, S. Yun, and J. Y. Choi, "Knowledge distillation with adversarial samples supporting decision boundary," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 3771–3778.

[28] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[29] L. E. Peterson, "K-nearest neighbor," *Scholarpedia*, vol. 4, no. 2, p. 1883, 2009.

[30] X. He and T.-S. Chua, "Neural factorization machines for sparse predictive analytics," in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2017, pp. 355–364.

[31] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," 2018, *arXiv:1804.03209*. [Online]. Available: https://arxiv.org/abs/1804.03209

[32] 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. [Online]. Available: https://arxiv.org/abs/1807.09840

[33] T. Heittola, A. Mesaros, and T. Virtanen, "TUT urban acoustic scenes 2018, development dataset," Audio Res. Group, Tampere Univ., Tampere, Finland, Tech. Rep., Apr. 2018. doi: 10.5281/zenodo.1228142.

[34] E. Fonseca, M. Plakal, F. Font, D. P. W. Ellis, X. Favory, J. Pons, and X. Serra, "General-purpose tagging of freesound audio with audioset labels: Task description, dataset, and baseline," in *Proc. Detection Classification Acoustic Scenes Events Workshop (DCASE)*, Nov. 2018, pp. 69–73. [Online]. Available: https://arxiv.org/abs/1807.09902

[35] E. Fonseca, J. Pons, X. Favory, F. Font, D. Bogdanov, A. Ferraro, S. Oramas, A. Porter, and X. Serra, "Freesound datasets: A platform for the creation of open audio datasets," in *Proc. 18th Int. Soc. Music Inf. Retr. Conf. (ISMIR)*, Suzhou, China, 2017, pp. 486–493.

[36] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: https://arxiv.org/abs/1409.1556

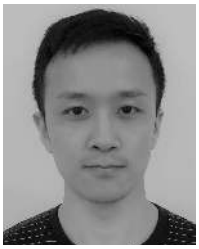
[37] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 2261–2269.

[38] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–13.

[39] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

LIANG GAO received the B.Sc. degree in computer science from the National University of Defense Technology, China, in 2017, where he is currently pursuing the master's degree. His research interests include computer vision, audio tagging, and deep learning.

HAIBO MI received the Ph.D. degree from the National Laboratory for Parallel and Distributed Processing, National University of Defense Technology (NUDT), Changsha, in 2012, where he is currently an Associate Professor. His research interests include distributed computing, cloud computing, and machine learning.



BOQING ZHU received the M.Sc. degree in computer science from the National University of Defense Technology, China, in 2019, where he is currently pursuing the Ph.D. degree. His research interests include machine learning and acoustics model.

DAWEI FENG received the Ph.D. degree from the University of Paris-Sud, France, in 2014. He is currently an Associate Professor with the College of Computer, National University of Defense Technology (NUDT). His research interests primarily focus on machine learning and distributed computing.

YICONG LI received the bachelor's degree in computer science and technology from the Hebei University of Technology and SUPINFO International University, in 2017. She is currently pursuing the master's degree in computer technology with the National University of Defense Technology. Her research interests include deep learning, natural language processing, and topic model.

YUXING PENG received the Ph.D. degree from the National University of Defense Technology (NUDT), China, in 1996, where he is currently a Professor with the College of Computer. His research interests primarily focus on machine learning and cloud computing.