INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

ATReSN-Net: Capturing Attentive Temporal Relations in Semantic Neighborhood for Acoustic Scene Classification

Liwen Zhang[1], *Jiqing Han*[1], *Ziqiang Shi*[2]

[1]School of Computer Science and Technology, Harbin Institute of Technology, China
[2]Information Technology Lab, Fujitsu Research & Development Center, Co., LTD.

lwzhang9161@126.com, jqhan@hit.edu.cn, shiziqiang@cn.fujitsu.com

Abstract

Convolutional Neural Networks (CNNs) have been widely investigated on Acoustic Scene Classification (ASC). Where the convolutional operation can extract useful semantic contents from a local receptive field in the input spectrogram within certain Manhattan distance, i.e., the kernel size. Although stacking multiple convolution layers can increase the range of the receptive field, without explicitly considering the temporal relations of different receptive fields, the increased range is limited around the kernel. In this paper, we propose a 3D CNN for ASC, named ATReSN-Net, which can capture temporal relations of different receptive fields from arbitrary time-frequency locations by mapping the semantic features obtained from the residual block into a semantic space. The ATReSN module has two primary components: first, a k-NN-based grouper for gathering a semantic neighborhood for each feature point in the feature maps. Second, an attentive pooling-based temporal relations aggregator for generating the temporal relations embedding of each feature point and its neighborhood. Experiments showed that our ATReSN-Net outperforms most of the state-of-the-art CNN models. We shared our code at ATReSN-Net.
Index Terms: Acoustic scene classification, attentive pooling, temporal relations, semantic neighborhood, ResNet.

1. Introduction

Acoustic Scene Classification (ASC) aims at classifying the real-life audio recordings into the predefined acoustic scene classes [1]. In the latest DCASE challenges [2, 3], the Convolutional Neural Network (CNN) based ASC methods have achieved promising performances. The ensembles of CNNs trained with multi-level features [4], and various classical CNN models [9, 8], e.g., VGG [5], ResNet [6], and AclNet [7], were utilized to deal with ASC. And the performance can be further improved by exploiting the teacher-student learning [10] based on the method in [4]. Followed by the SubSpectralNet [11], which aims at capturing the local relations in the frequency domain using band-wise crops of the spectrograms, the factorized CNN (FCNN) was proposed to learn the long-term ambient and short-term patterns in the acoustic scenes [12]. And it had been verified by the attention-based atrous CNN [13] that, a larger receptive field (RF) is more effective than the local pooling in learning time-frequency (T-F) representations.

These above works have extensively investigated the CNN-based ASC models. However, the temporal relations between different receptive fields (RFs) may not be considered explicitly. Recently, in our previous work, the pyramidal temporal pooling with discriminative mapping (DM-PTP) [14], we have verified that capturing temporal relations of the high-level features extracted from a group of ordered T-F RFs via a CNN, indeed benefits audio classification. However, the DM-PTP involves solving an argmin optimization problem [17] when performing temporal pooling [15], thus the CNN and DM-PTP model are trained separately. In this work, we attempt to capture the temporal relations between different RFs of the spectrogram in a more concise end-to-end framework, thus the temporal relations, CNN feature extractor, and classifier can be jointly optimized.

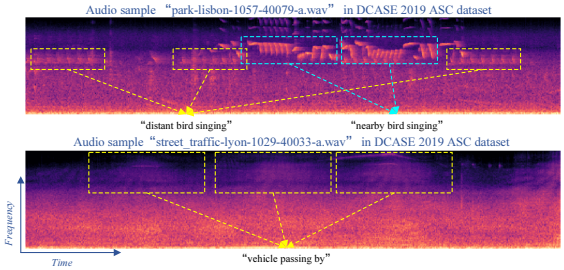

Figure 1: *The short-term sound patterns with similar semantic contents in real-life acoustic scenes.*

The first concern is which RFs should be selected for learning temporal relations. Real-life acoustic scenes are usually irregular and contain varieties of sound elements [16], the occurrences of the sound elements with similar semantic contents could be either periodic or randomly repeating (e.g., Figure 1). In [18], the nearest neighbor filter (NNF) [19] is used as the front-end of a CNN model, however, it may be insufficient to capture high-level temporal relations by directly grouping neighborhood on the front-end raw features. More recently, inspired by the PointNets [20, 21] based works, the Dynamic graph CNN [22] and correspondence proposals net [23] for computer vision tasks, we proposed the SeNoT-Net [24], which concentrates on learning temporal relations of different T-F RFs with similar semantic contents, i.e., semantic neighbors.

Results in [24] showed that, the performance of the CNN-based ASC model can be significantly improved by capturing the temporal relations of different feature points (i.e., T-F units) in the feature maps obtained from the 3D convolutional (Conv) block. However, for each feature point, the temporal relations within its semantic neighborhood are simply aggregated with a max/avg-pooling layer. As a hard aggregation, the max/avg-pooling would inevitably lead to leakage of information in a large RF. To better exploit the temporal relations in the semantic neighborhood, in this work, we utilize the attention mechanism to aggregate the temporal relations that would benefit the classification, and use a learnable MLP-based position encoding method to make full use of the positional information of the feature points. Then a 3D CNN-based network, ATReSN-Net, which aims at capturing the high-level **a**ttentive **t**emporal **re**lations of the feature points, and their **s**emantic **n**eighborhoods from different RFs, is proposed. The key mod-

http://dx.doi.org/10.21437/Interspeech.2020-1151

ule, ATReSN, consists of two components: 1) a k-NN-based semantic neighborhood grouper (SNG); 2) an attentive pooling-based temporal relations aggregator (ATRA). There are three major benefits by using this module in the CNN-based ASC model: 1) the obtained model can capture temporal relations of different RFs in arbitrary T-F locations by grouping neighborhood in the semantic space; 2) the ATRA helps the model to focus on those temporal relations that benefit the classification most; 3) multi-scale relations can be incorporated by inserting multiple ATReSNs in different layers of the network.

2. ATReSN-Net

The ATReSN-Net treats the 2D spectrogram of an audio sample as a 3D spectrogram sequence, which is formed by placing multiple smaller T-F segments with full frequency bins of the original spectrogram in temporal order. The 3D Conv is used to extract feature maps for the input sequence. For capturing temporal relations, the ATReSN first maps the feature points in the feature maps into a semantic space. Then for each feature point, the ATRA will obtain the temporal relations embedding of its self-centered semantic neighborhood, which are grouped by performing the k-NN-based SNG globally.

2.1. Semantic neighborhood grouping

In this work, we attempt to characterize an acoustic scene by aggregating the knowledge from the sound elements with similar semantic contents, i.e., semantic neighborhood. For the 2D Conv block, the local feature is extracted from a neighborhood defined by the Manhattan distance, such neighborhood can be easily grouped within a $k \times k$ square (here k is the kernel size), due to the regular structured input. For our ATReSN, the neighborhood is defined by the L_2 or *cosine* based semantic similarity, thus the input data is presented in an irregular form. Using the idea of our recent work [24], a k-NN-based grouper is designed to find the top-k semantic neighbors for each feature point in the feature maps obtained from the Conv block.

Specifically, given a Log Mel-Spectrogram of an audio sample, we first transform the 2D spectrogram into a 3D sequence with N smaller segments along the time bin. Using such sequence as input, a 3D Conv block with C channels can produce a set of feature maps, $\{\boldsymbol{X}_1, ..., \boldsymbol{X}_i, ..., \boldsymbol{X}_N\}$ with $\boldsymbol{X}_i \in \mathbb{R}^{F \times T \times C}$, where F and T are the output sizes in terms of the frequency and time bins, respectively. With these feature maps, the processing steps of SNG can be described as follows:

1) *Similarity matrix calculating.* For each pair of feature points, $\boldsymbol{x}_{i,f_i,t_i}$, $\boldsymbol{x}_{j,f_j,t_j} \in \mathbb{R}^C$ from $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$, their semantic similarity is calculated. Then a similarity matrix, $\boldsymbol{M} \in \mathbb{R}^{NFT \times NFT}$ with each row storing all the semantic distances for the corresponding center point, is obtained.

2) *Semantic neighbors searching.* To find the neighbors, a k-NN algorithm is performed on each row of $\boldsymbol{M}$, and the neighbor position indexes are obtained by searching the position index table according to the row indexes of the top-k similarities in $\boldsymbol{M}$. With these position indexes, the semantic neighborhood consisting of k neighbors for each feature point can be grouped.

2.2. Attentive temporal relations aggregation

As shown in Figure 2, the attentive temporal relations aggregator (ATRA) is performed on the neighborhood centered on each feature point grouped by the SNG. It consists of two units: 1) a pair-wise temporal relation encoder (PaTRE), 2) an attentive pooling-based aggregator.

2.2.1. Pair-wise temporal relation encoding

For each feature point and its k semantic neighbors, as well as their position indexes, the PaTRE unit aims at capturing the temporal relation between the center point and each of its semantic neighbor points. Since the semantic feature points are extracted from different T-F receptive fields, to capture their temporal relations, the encoder should be aware of their original positions in the feature maps. And it had been verified in [14] that the time indexes benefit the model for learning temporal relations in the audio signals. Hence, in this work, the feature points and their positional information are concatenated to present to the PaTRE unit. Specifically, this unit includes the following processing steps:

1) *Augmented position encoding.* To make full use of the positional information, we encode the position indexes of the center point and each of its k neighbors. For convenience, we use $\boldsymbol{x}_i, \boldsymbol{x}_{i(s)} \in \mathbb{R}^C$ ($i = 1, ..., NFT$ and $s = 1, ..., k$) to denote the i-th feature point and its semantic neighbor, respectively. The feature point position encoding is defined as

$$\hat{\boldsymbol{p}}_i^s = \gamma\left(\boldsymbol{p}_i \oplus \boldsymbol{p}_{i(s)} \oplus (\boldsymbol{p}_i - \boldsymbol{p}_{i(s)}) \oplus \|\boldsymbol{p}_i - \boldsymbol{p}_{i(s)}\|\right) \quad (1)$$

where $\boldsymbol{p}_i, \boldsymbol{p}_{i(s)} \in \mathbb{R}^3$ are the normalized position indexes (e.g., $\boldsymbol{p}_i = \{\frac{n_i}{N}, \frac{f_i}{F}, \frac{t_i}{T}\}$) of $\boldsymbol{x}_i$ and $\boldsymbol{x}_{i(s)}$, respectively, $\oplus$ denotes the concatenation operation, $\|\cdot\|$ is the Euclidean distance between two points, and $\gamma : \mathbb{R}^{10} \rightarrow \mathbb{R}^{16}$ stands for a shared MLP that generates the augmented positional embedding, $\hat{\boldsymbol{p}}_i^s$. It had been proved in [23, 25] that more robust local features can be obtained by using the relative positional information. Hence, except for the position indexes, $\hat{\boldsymbol{p}}_i^s$ also embeds the relative position, which characterizes the displacement changes of two feature points in time and frequency domains. Especially, for the time domain, there are two levels of displacement changes, i.e., the segment-level and point-level.

2) *Temporal relation encoding.* For each pair of the center point, $\boldsymbol{x}_i$, and its semantic neighbor, $\boldsymbol{x}_{i(s)}$, they are concatenated with their corresponding positional embedding, $\hat{\boldsymbol{p}}_i^s$, then presented to a shared MLP for the pair-wise temporal relation encoding, which is defined as follows:

$$\boldsymbol{e}_i^s = f(\boldsymbol{x}_i \oplus \boldsymbol{x}_{i(s)} \oplus \hat{\boldsymbol{p}}_i^s) \quad (2)$$

where $\boldsymbol{e}_i^s$ is the temporal relation embedding for $\boldsymbol{x}_i$ and $\boldsymbol{x}_{i(s)}$, and $f : \mathbb{R}^{2C+16} \rightarrow \mathbb{R}^C$ is the shared MLP with two hidden layers. The number of units in each hidden layer depends on the dimension of the center feature point, i.e., $C/4$ and $C/2$. Since the weights of the MLPs in our PaTRE unit are shared, the k pair-wise temporal relation embeddings for the semantic neighborhood of each feature point can be generated in parallel.

2.2.2. Attentive pooling-based aggregation

For local features aggregation, the max and average pooling seem to be the common choices [21, 23]. However, for the ASC task, it had been verified that learning audio representations in a larger receptive field is more effective [13]. Inspired by [26] and [25], we tend to use a more powerful attention-based pooling method that can automatically learn important local features from the neighborhood with more feature points. Our attentive pooling-based aggregator consists of the following steps:

1) *Attention weights computing.* Given the pair-wise temporal relation embeddings, $\{\boldsymbol{e}_i^1, ..., \boldsymbol{e}_i^k\}$, of $\boldsymbol{x}_i$, a shared MLP followed by a $softmax$ layer is used to calculate the attention weights. Specifically,

$$\boldsymbol{w}_i^s = softmax(g(\boldsymbol{e}_i^s)) \quad (3)$$

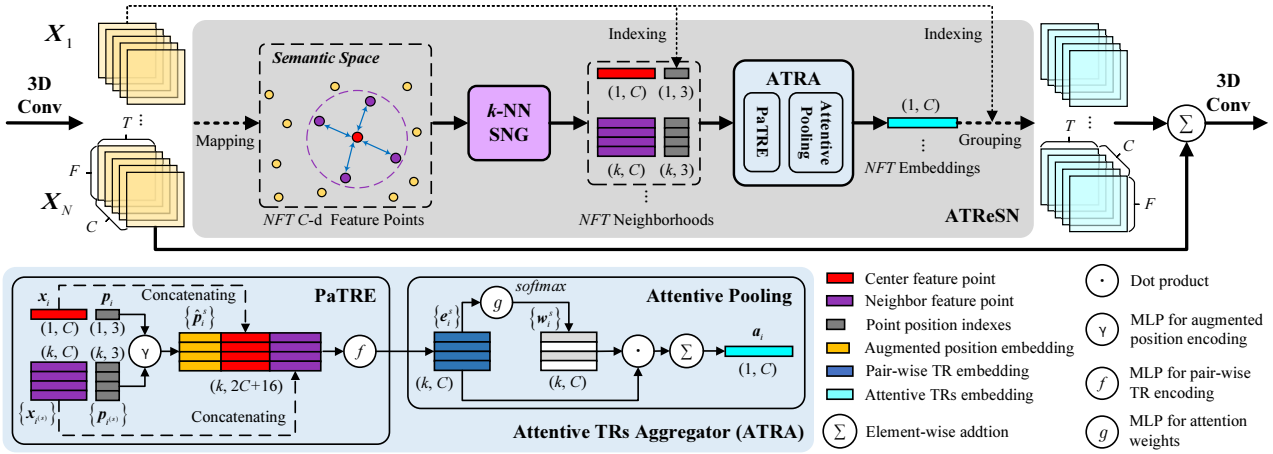

Figure 2: *Illustration of our ATReSN in a 3D convolutional architecture. TR is short for temporal relation, NFT means $N \times F \times T$.*

where $\boldsymbol{w}_i^s$ is the weight vector for $\boldsymbol{e}_i^s$, and $g : \mathbb{R}^C \rightarrow \mathbb{R}^C$ denotes the MLP with shared weights.

2) *Weighted summation.* By using the attention weight vectors, the attentive temporal relations embedding for each semantic neighborhood can be obtained as

$$\boldsymbol{a}_i = \sum_{s=1}^{k} \boldsymbol{w}_i^s \cdot \boldsymbol{e}_i^s. \tag{4}$$

Compared with max/avg-pooling, the attentive pooling can be regarded as a soft aggregating function with learnable parameters. It filters out those local temporal relations that should be concerned by the classifier, from the neighborhood of each feature point. Once all the attentive temporal relations embeddings, $\boldsymbol{a}_i$ with $i = 1, ..., NTF$ are obtained, they will be grouped to form N tensors of the same size with the feature maps, $\{\boldsymbol{X}_1, ..., \boldsymbol{X}_N\}$, according to the indexes of their corresponding feature points. Finally, an element-wise addition is conducted on both the target feature maps and embedding maps to form the input tensors for the following 3D Conv block.

2.3. ATReSN-Net architecture and implementation

The ATReSN works as an intermediate block like the Conv or residual (Res) block, and can be easily inserted into the classic CNN architectures. In this work, the ResNet [6] with a depth of 18 and its variant PreAct [27] are utilized as the backbones for our ATReSN-Net, respectively. We first train the backbones using the small segments of the entire Log Mel-Spectrogram of each audio sample in the training dataset. Then, all the Conv and pooling operations of the pre-trained backbone are transformed into their 3D forms. Finally, the ATReSN-Net is constructed by inserting the ATReSNs between two adjacent 3D residual blocks and removing the last FC layer in the backbone.

As shown in Table 1, for the Conv kernels in each 3D Res block, the additional dimension for the N spectrogram segments is 1, and the number of the channels is kept the same with the ResNet-18. Hence, each 3D Res block can be regarded as the combination of N 2D Res blocks with shared parameters in parallel, and there is no increase in the number of parameters. Moreover, since the weights of the ATRA block are also shared, the number is not changed as increasing the hyper-parameter, k for the SNG. Thus compared with the ResNet-18 (approximately 22.3M parameters), there is no significant increase of parameters for the ATReSN-Net.

Table 1: *The ATReSN-Net architecture with two ATReSNs using ResNet-18 as backbone, s denotes stride.*

Module	Description	Output Size
Input Log Mel-Spec sequence: $(N, 128, 80, 3)$		
3D Conv_0	$1 \times 3 \times 3$, 64	$(N, 128, 80, 64)$
3D Max-Pool_0	$1 \times 3 \times 3$, s: (1, 2, 2)	$(N, 64, 40, 64)$
3D Res_1_x	$\{1 \times 3 \times 3, 64\} \times 4$	$(N, 32, 20, 64)$
3D Res_2_1	$\{1 \times 3 \times 3, 128\} \times 2$	$(N, 16, 10, 128)$
ATReSN_1	3 shared MLPs	$(N, 16, 10, 128)$
3D Res_2_2	$\{1 \times 3 \times 3, 128\} \times 2$	$(N, 16, 10, 128)$
3D Res_3_1	$\{1 \times 3 \times 3, 256\} \times 2$	$(N, 8, 5, 256)$
ATReSN_2	3 shared MLPs	$(N, 8, 5, 256)$
3D Res_3_2	$\{1 \times 3 \times 3, 256\} \times 2$	$(N, 8, 5, 256)$
3D Res_4_x	$\{1 \times 3 \times 3, 512\} \times 4$	$(N, 4, 2, 512)$
Global Avg-Pool, 10-d FC, *softmax* (#parameters: 22.6M)		

3. Experiments and discussion

The experiments was conducted on the DCASE 2018 [2] and 2019 [3] ASC task 1a datasets. For each audio recording, the 3 channels 128 bands Log Mel energies that calculated from the original binaural signals and their differences with a frame size of 2048 samples (50% hop size) were extracted. Then each Log Mel-Spectrogram was split into 8 (i.e., $N = 8$) small segments of 80 frames long. For the backbone pre-training, the small Log Mel-Spectrogram segments in the training set were used to train the network. The SGD with Nesterov momentum [28] was used as the optimizer with the initial learning rate of 0.1, momentum of 0.9, weight decay of 0.0001, batch size of 128, and maximum epochs of 105. The learning rate decreased every 30 epochs exponentially with a rate of 0.1. For the ATReSN-Net fine-tuning, the ordered Log Mel-Spectrogram sequence of each training sample was used as the input. The SGD with momentum was used as the optimizer with the initial learning rate of 0.002, momentum of 0.9, weight decay of 0.0001, batch size of 16, and maximum epochs of 100. The learning rate decreased every 40 epochs exponentially with a rate of 0.1.

3.1. Comparisons with ResNet-baselines

To verify the effectiveness of our ATReSN-Net, two backbones, ResNet-18 and PreAct-18 were used as the baselines, and their test results were obtained by averaging the outputs of 8 seg-

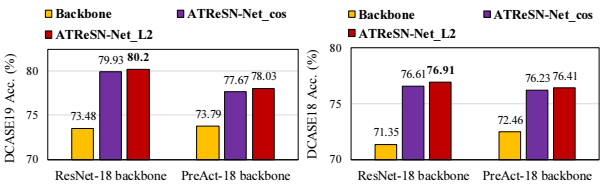

Figure 3: *Comparisons of ATReSN-Nets and ResNet backbones.*

ments in each testing sample. Based on each pre-trained backbone, two ATReSN-Nets with *cosine* and L_2 distances were fine-tuned. And they were all equipped with 4 ATReSNs locating in [2_1, 2_2] (i.e., between Res_2_1 and Res_2_2), [2_2, 3_1], [3_1, 3_2] and [3_2, 4_1]. The hyper-parameter, k for SNG was set to 4. As shown in Figure 3, the ATReSN-Nets significantly outperform their corresponding backbones by capturing temporal relations from different RFs. The best results are achieved by the ResNet-based ATReSN-Net using L_2 semantic distance.

3.2. Ablation studies

For the ablation experiments, the backbone was ResNet-18, the default k value was 8, and 4 ATReSNs with L_2 distance were used (the locations are consistent with Section 3.1).

a) Ablation on ATReSN. The ATReSN enables the network to explicitly aggregate important temporal relations in the semantic neighborhood. After removing ATReSN, the 3D ResNet would not extract local relations from different T-F RFs, and the aggregation is only performed on the global avg-pooling layer, which could inevitably result in the loss of information. Results in Table 2 show that our network significantly outperforms the 3D ResNet without the ATReSN.

Table 2: *Ablation experiments on ATReSN and augmented position encoding.*

Ablated Network ($k = 8$)	**DCASE19 Acc. (%)**
Network without ATReSN	75.86
Network with relative positions	79.98
Network with augmented positions	80.17
The Full Network (ATReSN-Net)	**80.68**

b) Ablation on attentive pooling. The attention mechanism helps our network to focus on the temporal relations that would benefit the classification. As a hard aggregation, max-pooling is usually effective for most cases. However, when performing on the larger receptive field, it would lead to leakage of useful information. For ablation on our attentive pooling aggregator, we designed an alternative version of ATReSN-Net, called TReSN-Net, by replacing the aggregator with a max-pooling layer. As shown in Figure 4, when $k \leq 4$, the performance of the two networks is rather close, and for the larger k, ATReSN-Net is better. This may indicate that the attentive pooling, as a soft aggregation, is more robust for aggregating information in larger RFs than the max-pooling.

c) Ablation on augmented position encoding. By considering the positions and relative displacements, our PaTRE unit can embed the local structure and temporal changes of the two semantic neighbors. A learnable MLP was used to encode the 10-d augmented positions for making full use of position information. Results in Table 2 show that, by encoding the augmented positions, a better performance can be further achieved.

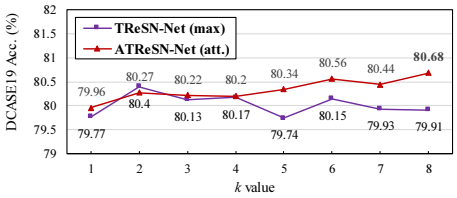

Figure 4: *Performance comparisons of networks using max-pooling and attentive pooling-based aggregating methods.*

Table 3: *Comparisons with other methods, simo is the strategy of a single model without ensemble or data augmentation. SpcAug denotes SpecAugment data augmentation.*

Method	**Strategy**	**DCASE Acc. (%)**	
		2018	**2019**
CNN baseline [2, 3]	*simo*	59.70	62.50
Jung et al. [4]	4 models	73.82	-
CNN-GRU_TS [10]	*simo*	74.26	-
Ensemble [10]	2 CNNs	77.36	-
CNN_NNF [18]	12 CNNs	69.30	-
Atrous CNN [13]	*simo*	72.70	-
SubSpectralNet [11]	*simo*	74.08	73.44
DM-PTP [14]	*simo*	75.80	76.91
FCNN-triplet [12]	*simo* + *SpcAug*	-	77.19
ResNet-50 [9]	*simo* + *SpcAug*	-	77.87
AclSincNet [9]	*simo*	-	76.08
SeNoT-Net [24]	*simo*	77.19	80.34
ATReSN-Net (ours)	*simo*	**77.87**	**80.68**

3.3. Comparisons with other methods

The results of the state-of-the-art ASC methods in Table 3 are from their corresponding articles. As shown in the table, our ATReSN-Net outperforms the CNN-based methods without learning temporal relations from different T-F RFs [2, 3, 4, 13, 9]. Compared with the sequential modeling methods, CNN-GRU_TS [10] and DM-PTP [14], our new method can also achieve higher accuracies, even for the ensemble model in [10]. Compared with SubSpectralNet [11] that only focus on extracting local relations from different frequency bins, our method shows better performance by capturing local relations from different RFs in both time and frequency domains. Furthermore, the ATReSN-Net can even surpass the FCNN with SpecAugment [29] data augmentation. By collecting the neighbors in high-level semantic feature maps instead of the raw input features, our method significantly outperforms the CNN_NNF [18]. And compare to our former work [24], the performances on both two datasets can be further improved by applying the attention-based aggregation for the temporal relations within the semantic neighborhoods.

4. Conclusions

In this paper, we propose a novel 3D CNN for ASC by capturing attentive temporal relations from a semantic neighborhood in the acoustic scene. We transform the ResNets into their 3D forms, and present the ATReSN-Net by inserting the ATReSNs into the 3D ResNets. With multiple ATReSNs in different layers, our network can learn temporal relations in multi-scales, and by aggregating the temporal relations from a neighborhood with the attentive pooling, the effective temporal relations embedding can be extracted from a large receptive field.

5. References

[1] A. Mesaros, T. Heittola, E. Benetos, P. Foster, M. Lagrange, T. Virtanen and M. D. Plumbley, "Detection and Classification of Acoustic Scenes and Events: Outcome of the DCASE 2016 Challenge," *IEEE/ACM Trans. Audio, Speech, and Language Processing*, vol. 26, no. 2, pp. 379-393, Feb. 2018, DOI. 10.1109/TASLP.2017.2778423.

[2] A. Mesaros, T. Heittola, and T. Virtanen, "A Multi-Device Dataset for Urban Acoustic Scene Classification," in *Proc. Detection and Classification of Acoustic Scenes and Events 2018 Workshop (DCASE2018)*, Nov. 2018, pp. 9-13.

[3] A. Mesaros, T. Heittola, and T. Virtanen, "Acoustic Scene Classification in DCASE 2019 Challenge: Closed and Open Set Classification and Data Mismatch Setups," in *Proc. Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)*, NY, USA, Oct. 2019, pp. 164-168.

[4] J. Jung, H. Heo, H. Shim and H. Yu, "DNN Based Multi-Level Feature Ensemble for Acoustic Scene Classification," in *Proc. Detection and Classification of Acoustic Scenes and Events 2018 Workshop (DCASE2018)*, Nov. 2018, pp. 113-117.

[5] K. Simonyan, and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *Computational and Biological Learning Society*, 2015, pp. 1-14.

[6] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016 pp. 770-778. DOI. 10.1109/CVPR.2016.90.

[7] J. Huang *et al*., "AclNet: Efficient End-to-End Audio Classification CNN", *CoRR*, 2018. [Online]. Available: http://arxiv.org/abs/1811.06669

[8] M. Octave, C. Matthieu and S. Oliver, "Exploring Deep Vision Models for Acoustic Scene Classification," in *Proc. Detection and Classification of Acoustic Scenes and Events 2018 Workshop (DCASE2018)*, Nov. 2018, pp. 103-107.

[9] J. Huang *et al*., "Acoustic Scene Classification Using Deep Learning-based Ensemble Averaging," in *Proc. Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)*, New York, NY, USA, Oct. 2019, pp. 94-98.

[10] H. Heo, J. Jung, H. Shim and H. Yu, "Acoustic Scene Classification Using Teacher-Student Learning with Soft-Labels," in *Proc. Interspeech*, Graz, Austria, 2019, pp. 614-618.

[11] S. Phaye, E. Benetos and Y. Wang, "Subspectralnet - Using Subspectrogram Based Convolutional Neural Networks for Acoustic Scene Classification," in *ICASSP*, Brighton, UK, 2019, pp. 825-829.

[12] J. Cho, S. Yun, H. Park, J. Eum and K. Hwang, "Acoustic Scene Classification Based on a Large-Margin Factorized CNN," in *Proc. Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)*, New York, NY, USA, Oct. 2019, pp. 45-49.

[13] Z. Ren, Q. Kong, J. Han, M. D. Plumbley and B. W. Schuller, "Attention-based Atrous Convolutional Neural Networks: Visualization and Understanding Perspectives of Acoustic Scenes," in *ICASSP*, Brighton, UK, 2019, pp. 56-60.

[14] L. Zhang, Z. Shi and J. Han, "Pyramidal Temporal Pooling with Discriminative Mapping for Audio Classification," in *IEEE/ACM Trans. Audio, Speech, and Language Processing*, vol. 28, pp. 770-784, 2020. DOI. 10.1109/TASLP.2020.2966868.

[15] L. Zhang, J. Han, and S. Deng, "Unsupervised Temporal Feature Learning Based on Sparse Coding Embedded BoAW," in *Proc. Interspeech*, Hyderabad, India, 2018, pp. 3284-3288.

[16] S. Chu, S. Narayanan and C. Jay Kuo, "Content Analysis for Acoustic Environment Classification in Mobile Robots," in *Proc. AAAI Fall Symposium* , Oct. 2006.

[17] H. Drucker, C. Burges, L. Kaufman, A. Smola and V. Vapnik, "Support Vector Regression Machines," *Advances in Neural Information Processing Systems*, vol. 9, pp. 155-161, 1997.

[18] N. Truc and P. Franz, "Acoustic Scene Classification Using a Convolutional Neural Network Ensemble and Nearest Neighbor Filters," in *Proc. Detection and Classification of Acoustic Scenes and Events 2018 Workshop (DCASE2018)*, Nov. 2018, pp. 34-38.

[19] Z. Rafii and B. Pardo, "Music/Voice Separation Using the Similarity Matrix," in *Proc. ISMIR*, pp. 583-588, 2012.

[20] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 2017, pp. 77-85. DOI. 10.1109/CVPR.2017.16.

[21] C. R. Qi, L. Yi, H. Su and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Conference on Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA, USA.

[22] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon. "Dynamic Graph CNN for Learning on Point Clouds," in *ACM Trans. Graph.*, vol. 38, no. 5, 2019. DOI. 10.1145/3326362.

[23] X. Liu, J. Lee and H. Jin, "Learning Video Representations From Correspondence Proposals," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, 2019. pp. 4268-4276. DOI. 10.1109/CVPR.2019.00440.

[24] L. Zhang, J. Han, and Z. Shi. "Learning Temporal Relations from Semantic Neighbors for Acoustic Scene Classification," *IEEE Signal Processing Letters*, vol. 27, 2020. pp. 950-954, DOI. 10.1109/LSP.2020.2996085.

[25] Q. Hu, B. Yang, L. Xie, et al. "RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds," *arXiv preprint arXiv:1911.11236v2*, 2020.

[26] B. Yang, S. Wang, A. Markham, et al., "Robust Attentional Aggregation of Deep Feature Sets for Multi-view 3D Reconstruction," *IJCV*, 128, pp. 53-73, (2020).

[27] K. He *et al*., "Identity Mappings in Deep Residual Networks," in *Proc. ECCV 2016*, 2016, pp. 630-645.

[28] Y. Nesterov, "Nonsmooth Convex Optimization" in *Introductory Lectures on Convex Optimization A Basic Course*, Boston, MA, USA: Springer, 2004, pp. 111-170.

[29] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A Simple Data Augmentation Method for Automatic Speech Recognition," *arXiv preprint arXiv:1904.08779*, 2019.